



CMS32C030 User Manual

**32-bit microcontrollers based on ARM® Cortex®-M0+
V0.9.1**

Please note the following CMS IP policy

* China Micro Semicon Co., Ltd. (hereinafter referred to as the Company) has applied for patents and holds absolute legal rights and interests. The patent rights associated with the Company's MCUs or other products have not been authorized for use, and any company, organization, or individual who infringes the Company's patent rights through improper means will be subject to all possible legal actions taken by the Company to curb the infringement and to recover any damages suffered by the Company as a result of the infringement or any illegal benefits obtained by the infringer.

* The name and logo of Cmsemicon are registered trademarks of the Company.

* The Company reserves the right to further explain the reliability, functionality and design improvements of the products in the data sheet. However, the Company is not responsible for the use of the Specification Contents. The applications mentioned herein are for illustrative purposes only and the Company does not warrant and does not represent that these applications can be applied without further modification, nor does it recommend that its products be used in places that may cause harm to persons due to malfunction or other reasons. The Company's products are not authorized for use as critical components in lifesaving, life-sustaining devices or systems. The Company reserves the right to modify the products without prior notice. For the latest information, please visit the official website at www.mcu.com.cn.

Documentation Instructions

This manual is the technical reference manual for the CMS32C030 controller product. The technical reference manual provides application notes on how to use this series of products. It contains detailed information on the structure, functional descriptions, operating modes, and register configurations of each functional module, with a dedicated chapter for each functional module.

The technical reference manual provides descriptions of all the functional modules for this series of products. For detailed information on the specific features of a particular model (i.e., the functionality it supports), please refer to the corresponding datasheet.

The data sheet information is as follows:

CMS32C030: CMS32C030 Datasheet_vx.x.x.pdf

Typically, in the early stages of chip selection, the datasheet should be reviewed first to assess whether the product meets the functional requirements of the design. Once the required product is preliminarily selected, the technical reference manual should be consulted to confirm whether the operating modes of the functional modules meet the requirements. When the selection process moves into the programming and design phase, the technical reference manual should be studied in detail to understand the specific implementation methods and register configurations of each function. For hardware design, the datasheet can be referenced for information on voltage, current, drive capability, pin assignments, and other details.

For detailed information regarding the Cortex-M0+ core, SysTick timer, and NVIC (Nested Vectored Interrupt Controller), please refer to the corresponding ARM documentation.

Table of Contents

Documentation Instructions	2
Chapter 1 CPU	15
1.1 Overview	15
1.2 Cortex-M0+ Features.....	15
1.3 Debug Features	15
1.4 SWD Interface Pins and Debug Control.....	17
1.5 LOCKUP Control Register (LOCKCTL).....	19
1.6 Protection Register (PRCR).....	19
1.7 ARM Reference Documentation	19
Chapter 2 Pin Functions	20
2.1 Port Functions.....	20
2.2 Port Alternate Functions	20
2.3 Register Mapping.....	22
2.3.1 Control Function Register Mapping.....	22
2.3.2 I/O Alternate Function Register Mapping.....	23
2.3.3 Special Function Port RESINB Control Register Mapping	25
2.4 Register Description	26
2.4.1 Port Register (Px).....	27
2.4.2 Port Mode Register (PMx).....	29
2.4.3 Pull-Up Resistor Selection Register (PUx)	31
2.4.4 Pull-Down Resistor Selection Register (PDx)	33
2.4.5 Port Output Mode Register (POMx)	35
2.4.6 Port Mode Control Register (PMCx)	37
2.4.7 Port Set Control Register (PSETx)	39
2.4.8 Port Clear Control Register (PCLR _x)	41
2.4.9 Port Readback Register (PREAD _x)	43
2.4.10 Port Mode Selection Register (PMS).....	45
2.4.11 TTL and Schmitt Input Selection (PTTL _x).....	45
2.4.12 Port Output Alternate Function Configuration Register (PS_XX).....	47
2.4.13 Special Function Port RESINB Description (RSTM).....	57
Chapter 3 System Architecture	58
3.1 Overview	58
3.2 System Address Partitioning.....	59
Chapter 4 Clock Generation Circuit	61
4.1 Functions of Clock Generation Circuit.....	61
4.2 Structure of Clock Generation Circuit.....	62
4.3 Register Mapping.....	63
4.4 Register Description	64
4.4.1 System Clock Control Register (CKC)	64
4.4.2 Clock Operation Status Control Register (CSC).....	65
4.4.3 Clock Operation Mode Control Register (CMC).....	66
4.4.4 Peripheral Enable Registers (PER0, PER10, PER11, PER12, and PER13).....	67
4.4.5 15-Bit Interval Timer Operation Clock Selection Register (OSMC).....	70

4.4.6	High-Speed On-Chip Oscillator Frequency Selection Register (HOCODIV)	70
4.4.7	Low-speed Oscillator Clock Selection Register (SUBCKSEL)	71
4.5	System Clock Oscillation Circuit	72
4.5.1	XT1 Oscillation Circuit	72
4.5.2	High-Speed On-Chip Oscillator	72
4.5.3	Low-Speed On-Chip Oscillator	73
4.6	Operation of Clock Generation Circuit	74
4.7	Clock Control	75
4.7.1	Example of High-Speed On-Chip Oscillator Set-Up	75
4.7.2	Example of Setting XT1 Oscillation Circuit	76
4.7.3	State Transition Diagram of CPU Clock	77
4.7.4	Conditions Before Changing CPU Clock and Processing After Changing CPU Clock	79
4.7.5	Time Required to Switch CPU Clock and Main System Clock	80
4.7.6	Conditions Before Clock Oscillation Is Stopped	80
Chapter 5	15-Bit Interval Timer	81
5.1	Functions of 15-Bit Interval Timer	81
5.2	Structure of 15-Bit Interval Timer	81
5.3	Register Mapping	82
5.4	Register Description	83
5.4.1	Peripheral Enable Register 0 (PER0)	83
5.4.2	15-Bit Interval Timer Control Register (CON)	84
5.4.3	15-Bit Interval Timer Count Clock Control Register (LPTIMCL)	84
5.5	Operation of 15-Bit Interval Timer	85
5.5.1	Operation Timing of 15-Bit Interval Timer	85
5.5.2	Start of Count Operation and Re-Enter to Sleep Mode After Returned from Sleep Mode	86
Chapter 6	Advanced Control Timer (TIM1)	87
6.1	TIM1 Introduction	87
6.2	TIM1 Main Features	87
6.3	TIM1 Functional Description	89
6.3.1	Time-Base Unit	89
6.3.2	Counter Modes	91
6.3.3	Repetition Counter	100
6.3.4	Clock Selection	102
6.3.5	Capture/Compare Channels	105
6.3.6	Input Capture Mode	107
6.3.7	PWM Input Mode	108
6.3.8	Forced Output Mode	109
6.3.9	Output Compare Mode	110
6.3.10	PWM Mode	111
6.3.11	Complementary Output and Dead-Time Insertion	115
6.3.12	Using the Brake Function	117
6.3.13	Clearing the OCxREF Signal on An External Event	119
6.3.14	6-Step PWM Generation	120
6.3.15	One Pulse Mode	121
6.3.16	Encoder Interface Mode	123
6.3.17	Timer Input XOR Function	125

6.3.18	Interfacing With Hall Sensors	125
6.3.19	TIM1 and External Trigger Synchronization	127
6.3.20	Timer Synchronization.....	130
6.3.21	Debug Mode.....	131
6.4	Register Mapping.....	132
6.5	Register Description	133
6.5.1	TIM1 Control Register 1 (TIM1_CR1).....	133
6.5.2	TIM1 Control Register 2(TIM1_CR2).....	135
6.5.3	TIM1 Slave Mode Control Register (SMCR)	137
6.5.4	TIM1 DMA/Interrupt Enable Register (TIM1_DIER)	139
6.5.5	TIM1 Status Register (TIM1_SR).....	140
6.5.6	TIM1 Event Generation Register (TIM1_EGR)	142
6.5.7	TIM1 Capture/Compare Mode Register 1 (TIM1_CCMR1).....	143
6.5.8	TIM1 Capture/Compare Mode Register 1 (TIM1_CCMR2).....	146
6.5.9	TIM1 Capture/Compare Enable Register (TIM1_CCER)	148
6.5.10	TIM1 Counter (TIM1_CNT).....	150
6.5.11	TIM1 Prescaler (TIM1_PSC).....	150
6.5.12	TIM1 Auto-Reload Register (TIM1_ARR).....	150
6.5.13	TIM1 Repetition Counter Register (TIM1_RCR).....	150
6.5.14	TIM1 Capture/Compare Register 1 (TIM1_CCR1).....	151
6.5.15	TIM1 Capture/Compare Register 2 (TIM1_CCR2).....	151
6.5.16	TIM1 Capture/Compare Register 3 (TIM1_CCR3).....	151
6.5.17	TIM1 Capture/Compare Register 4 (TIM1_CCR4).....	152
6.5.18	TIM1 Brake and Dead-Time Register (TIM1_BDTR)	152
6.5.19	TIM1 DMA Control Register (TIM1_DCR).....	154
6.5.20	TIM1 Full Transfer DMA Address (TIM1_DMAR).....	154
Chapter 7	General-Purpose Timer (TIM3).....	155
7.1	TIM3 Introduction	155
7.2	TIM3 Main Features	155
7.3	TIM3 Functional Description	157
7.3.1	Time-Base Unit	157
7.3.2	Counter Modes.....	159
7.3.3	Clock Selection	170
7.3.4	Capture/Compare Channels	173
7.3.5	Input Capture Mode	175
7.3.6	PWM Input Mode	176
7.3.7	Forced Output Mode	177
7.3.8	Output Compare Mode.....	177
7.3.9	PWM Mode.....	179
7.3.10	One-Pulse Mode.....	182
7.3.11	Clearing the OCxREF Signal on an External Event.....	184
7.3.12	Encoder Interface Mode.....	185
7.3.13	Timer Input XOR Function	187
7.3.14	Timers and External Trigger Synchronization	187
7.3.15	Timer Synchronization.....	190
7.3.16	Debug Mode.....	197
7.4	Register Mapping.....	198

7.5	Register Description	199
7.5.1	TIM3 Control Register 1 (TIM3_CR1).....	199
7.5.2	TIM3 Control Register 2 (TIM3_CR2).....	201
7.5.3	TIM3 Slave Mode Control Register (TIM3_SMCR).....	202
7.5.4	TIM3 DMA/Interrupt Enable Register (TIM3_DIER)	203
7.5.5	TIM3 Status Register (TIM3_SR).....	204
7.5.6	TIM3 Event Generation Register (TIM3_EGR)	205
7.5.7	TIM3 Capture/Compare Mode Register 1 (TIM3_CCMR1)	206
7.5.8	TIM3 Capture/Compare Mode Register 2 (TIM3_CCMR2)	209
7.5.9	TIM3 Counter (TIM3_CNT).....	211
7.5.10	TIM3 Prescaler (TIM3_PSC).....	211
7.5.11	TIM3 Capture/Compare Enable Register (TIM3_CCER)	212
7.5.12	TIM3 Auto-Reload Register (TIM3_ARR).....	213
7.5.13	TIM3 Capture/Compare Register 1 (TIM3_CCR1).....	213
7.5.14	TIM3 Capture/Compare Register 2 (TIM3_CCR2).....	213
7.5.15	TIM3 Capture/Compare Register 3 (TIM3_CCR3).....	214
7.5.16	TIM3 Capture/Compare Register 4 (TIM3_CCR4).....	214
7.5.17	TIM3 DMA Control Register (TIM3_DCR).....	214
7.5.18	TIM3 Full Transfer DMA Address (TIM3_DMAR).....	215
Chapter 8	General-Purpose Timer (TIM14).....	216
8.1	TIM14 Introduction	216
8.2	TIM14 Main Features	216
8.3	TIM14 Functional Description	218
8.3.1	Time-Base Unit	218
8.3.2	Counter Modes.....	220
8.3.3	Clock Selection	223
8.3.4	Capture/Compare Channels	224
8.3.5	Input Capture Mode	226
8.3.6	Forced Output Mode	227
8.3.7	Output Compare Mode.....	228
8.3.8	PWM Mode.....	229
8.3.9	One Pulse Mode	230
8.3.10	Debug Mode.....	232
8.4	Register Mapping.....	233
8.5	Register Description	234
8.5.1	TIM14 Control Register 1 (TIM14_CR1).....	234
8.5.2	TIM14 Interrupt Enable Register (TIM14_DIER).....	235
8.5.3	TIM14 Status Register (TIM14_SR).....	235
8.5.4	TIM14 Event Generation Register (TIM14_EGR)	236
8.5.5	TIM14 Capture/Compare Mode Register 1 (TIM14_CCMR1)	236
8.5.6	TIM14 Capture/Compare Enable Register (TIM14_CCER)	238
8.5.7	TIM14 Counter (TIM14_CNT).....	238
8.5.8	TIM14 Prescaler (TIM14_PSC).....	239
8.5.9	TIM14 Auto-Reload Register (TIM14_ARR).....	239
8.5.10	TIM14 Capture/Compare Register 1 (TIM14_CCR1).....	239
8.5.11	TIM14 Option Register (TIM14_OR)	239
Chapter 9	General-Purpose Timers (TIM16/17).....	240

9.1	TIM16/TIM17 Main Features.....	240
9.2	TIM16/TIM17 Functional Description.....	242
9.2.1	Time-Base Unit	242
9.2.2	Counter Modes.....	244
9.2.3	Repetition Counter	248
9.2.4	Clock Sources	249
9.2.5	Capture/Compare Channels	249
9.2.6	Input Capture Mode	251
9.2.7	Forced Output Mode	252
9.2.8	Output Compare Mode.....	253
9.2.9	PWM Mode.....	254
9.2.10	Complementary Outputs and Dead-Time Insertion	255
9.2.11	Using the Break Function	257
9.2.12	One-Pulse Mode.....	260
9.3	Register Mapping.....	261
9.4	Register Description	262
9.4.1	TIM16/17 Control Register 1 (TIMx_CR1).....	262
9.4.2	TIM16/17 Control Register 2 (TIM16/17_CR2)	263
9.4.3	TIM16/17 DMA/Interrupt Enable Register (TIM16/17_DIER)	263
9.4.4	TIM16/17 Status Register (TIM16/17_SR)	264
9.4.5	TIM16/17 Event Generation Register (TIM16/17_EGR).....	265
9.4.6	TIM16/17 Capture/Compare Mode Register 1(TIM16/17_CCMR1).....	266
9.4.7	TIM16/17 Capture/Compare Enable Register (TIM16/17_CCER)	269
9.4.8	TIM16/17 Counter (TIM16/17_CNT)	270
9.4.9	TIM16/17 Prescaler (TIM16/17_PSC).....	270
9.4.10	TIM16/17 Auto-Reload Register (TIM16/17_ARR)	270
9.4.11	TIM16/17 Repetition Counter Register (TIM16/17_RCR).....	270
9.4.12	TIM16/17 Capture/Compare Register 1(TIM16/17_CCR1).....	271
9.4.13	TIM16/17 Brake and Dead Time Register (TIM16/17_BDTR).....	272
9.4.14	TIM16/17 DMA Control Register (TIM16/17_DCR).....	274
9.4.15	DMA address for Full Transfer (TIM16/17_DMAR)	274
Chapter 10	Windowed Watchdog Timer (WWDG).....	275
10.1	WWDG Introduction	275
10.2	WWDG Main Features	275
10.3	WWDG Functional Description	276
10.4	How To Program the Watchdog Timeout	278
10.5	Register Mapping.....	279
10.6	Register Description	279
10.6.1	WWDG Timer Control Register (WWDG_CR)	279
10.6.2	WWDG Configuration Register (WWDG_CFR)	280
10.6.3	WWDG Status Register (WWDG_SR).....	280
Chapter 11	Independent Watchdog (IWDG).....	281
11.1	Introduction	281
11.2	Features.....	281
11.3	Functional Description.....	281
11.3.1	Hardware Watchdog.....	281

11.3.2	Register Access Protection.....	281
11.3.3	Debug Mode.....	282
11.4	IWDG Register Mapping.....	283
11.5	IWDG Register Description	284
11.5.1	Key Register (IWDG_KR).....	284
11.5.2	Prescaler Register (IWDG_PR)	284
11.5.3	Reload Register (IWDG_RLR).....	285
11.5.4	Status Register (IWDG_SR)	285
Chapter 12	Universal Serial Communication Unit	286
12.1	Functions of Universal Serial Communication Unit.....	287
12.1.1	3-Wire Serial I/O (SSPI00)	287
12.1.2	Simplified SPI with Slave Chip Select Function	288
12.1.3	UART0.....	289
12.1.4	Simplified I ² C(IIC00)	290
12.2	Structure of Universal Serial Communication Unit.....	291
12.2.1	Shift Register	293
12.2.2	Serial Data Register mn (SDRmn).....	293
12.3	Registers for Controlling Universal Serial Communication Unit.....	295
12.3.1	Peripheral Enable Register (PER0).....	297
12.3.2	Serial Clock Select Register m (SPSm)	298
12.3.3	Serial Mode Register mn(SMRmn)	299
12.3.4	Serial Communication Operation Configuration Register mn (SCRmn).....	300
12.3.5	Serial Data Register mn (SDRmn).....	302
12.3.6	Serial Flag Clear Trigger Register mn (SDIRmn).....	303
12.3.7	Serial Status Register mn (SSRmn)	304
12.3.8	Serial Channel Start Register m (SSm)	306
12.3.9	Serial Channel Stop Register m (STm).....	307
12.3.10	Serial Channel Enable Status Register m (SEm).....	308
12.3.11	Serial Output Enable Register m (SOEm).....	309
12.3.12	Serial Output Register m (SOM)	310
12.3.13	Serial Output Level Register m (SOLm).....	311
12.3.14	Input Switch Control Register (ISC).....	312
12.3.15	Noise Filter Enable Register 0 (NFEN0)	313
12.3.16	Registers Controlling Port Functions of Serial I/O Pins	314
12.4	Run Stop Mode	315
12.4.1	Stopping the Operation by Units.....	315
12.4.2	Stopping the Operation by Channels.....	316
12.5	3-Wire Serial I/O (SSPI00) Communication	317
12.5.1	Master Transmission	318
12.5.2	Master Reception	327
12.5.3	Master Transmission and Reception	336
12.5.4	Slave Transmission	345
12.5.5	Slave Reception.....	353
12.5.6	Slave Transmission and Reception.....	360
12.5.7	Calculation of Transfer Clock Frequenc	369
12.5.8	Procedure for Handling Errors During 3-Wire Serial I/O Communication (SSPI00).....	371
12.6	Operation of Clock-Synchronous Serial Communication with Slave Selection Input Function	372

12.6.1	Slave Transmission	375
12.6.2	Slave Reception.....	385
12.6.3	Slave Transmission and Reception.....	392
12.6.4	Calculation of Transfer Clock Frequency	402
12.6.5	Procedure for Handling Errors in Clock-Synchronous Serial Communication with Slave Select Input Function	403
12.7	Operation of UART0 Communication.....	404
12.7.1	UART Transmission.....	405
12.7.2	UART Reception.....	414
12.7.3	Calculation of Baud Rate	421
12.7.4	Handling Steps When an Error Occurs During UART0 Communication	425
12.8	Simplified I ² C (IIC00) Communication Operation.....	426
12.8.1	Address Field Transmission	427
12.8.2	Data Transmission.....	432
12.8.3	Data Reception.....	435
12.8.4	Generation of Stop Condition	439
12.8.5	Calculation of Transfer Rate	440
Chapter 13	Serial Peripheral Interface (SPI)	444
13.1	Introduction	444
13.2	SPI	444
13.2.1	SPI Features	444
13.3	SPI Functional Description.....	445
13.3.1	General Description	445
13.3.2	Configuring SPI as a Slave Device	449
13.3.3	Configuring SPI as a Master Device	451
13.3.4	Configuring SPI for Half-Duplex Communication	453
13.3.5	Data Transmission and Reception Process.....	454
13.3.6	CRC Calculation	462
13.3.7	Status Flags	464
13.3.8	Procedure for Disabling the SPI.....	465
13.3.9	Using DMA (Direct Memory Access) for SPI Communication.....	466
13.3.10	Error Flags	469
13.3.11	SPI Interrupts	471
13.4	SPI Register Mapping.....	472
13.5	SPI Register Description.....	473
13.5.1	SPI Control Register 1 (SPI_CR1).....	473
13.5.2	SPI Control Register 2 (SPI_CR2).....	475
13.5.3	SPI Status Register (SPI_SR).....	476
13.5.4	SPI Data Register (SPI_DR).....	477
13.5.5	SPI CRC Polynomial Register (SPI_CRCPR).....	478
13.5.6	SPI RX CRC Register (SPI_RXCRCR)	478
13.5.7	SPI TX CRC Register (SPI_TXCRCR)	478
Chapter 14	I²C Interface.....	479
14.1	Introduction to I ² C	479
14.2	I ² C Main Features	479
14.3	I ² C Functional Description	481

14.3.1	Mode Selection	481
14.3.2	I ² C Slave Mode	483
14.3.3	I ² C Master Mode	486
14.3.4	Error Conditions	490
14.3.5	Programmable Noise Filter	492
14.3.6	SDA/SCL Line Control	492
14.3.7	SMBus	493
14.3.8	DMA Requests	496
14.3.9	Packet Error Checking (PEC)	498
14.4	I ² C Interrupts	499
14.5	I ² C Debug Mode	500
14.6	Register Mapping	500
14.7	Register Description	501
14.7.1	I ² C Control Register 1 (I2C_CR1)	501
14.7.2	I ² C Control Register 2 (I2C_CR2)	503
14.7.3	I ² C Own Address Register 1 (I2C_OAR1)	504
14.7.4	I ² C Own Address Register 2 (I2C_OAR2)	504
14.7.5	I ² C Data Register (I2C_DR)	505
14.7.6	I ² C Status Register 1 (I2C_SR1)	506
14.7.7	I ² C Status Register 2 (I2C_SR2)	509
14.7.8	I ² C Clock Control Register (I2C_CCR)	510
14.7.9	I ² C TRISE Register (I2C_TRISE)	511
14.7.10	I ² C FLTR Register (I2C_FLTR)	511
Chapter 15	Universal Synchronous/Asynchronous Receiver Transmitter (USART)	512
15.1	USART Introduction	512
15.2	USART Main Features	512
15.3	USART Functional Description	514
15.3.1	USART Character Description	516
15.3.2	Transmitter	517
15.3.3	Receiver	521
15.3.4	Generation of Decimal Baud Rate	526
15.3.5	Tolerance of the USART Receiver to Clock Deviation	529
15.3.6	Multiprocessor Communication	530
15.3.7	Parity Control	532
15.3.8	LIN (Local Interconnect Network) Mode	533
15.3.9	USART Synchronous Mode	536
15.3.10	Single-Wire Half-Duplex Communication	539
15.3.11	IrDA SIR ENDEC Module	540
15.3.12	2 Using DMA for Continuous Communication	542
15.3.13	Hardware Flow Control	545
15.4	USART Interrupts	547
15.5	USART Mode Configuration	548
15.6	Register Mapping	549
15.6.1	USART Status Register (USART_SR)	550
15.6.2	USART Data Register (USART_DR)	552
15.6.3	USART Baud Rate Register (USART_BRR)	552
15.6.4	USART Control Register 1 (USART_CR1)	553

15.6.5	USART Control Register 2 (USART_CR2).....	555
15.6.6	USART Control Register 3 (USART_CR3).....	557
15.6.7	USART Guard Time and Prescaler Register (USART_GTPR)	559
Chapter 16	Enhanced DMA	560
16.1	Functions of DMA.....	560
16.2	Structure of DMA.....	562
16.3	Register Mapping.....	563
16.4	Registers for Controlling DMA	564
16.4.1	Allocation of DMA Control Data Region and DMA Vector Table Region.....	564
16.4.2	Controlling Data Allocation	565
16.4.3	Vector Table	567
16.4.4	Peripheral Enable Register 10(PER10).....	568
16.4.5	DMA Channel Trigger Source Selection Register j (DMA_CHj_SEL) (j=0~1, 3).....	569
16.4.6	DMA Control Register j (DMACRj) (j=0~1, 3)	570
16.4.7	DMA Block Size Register j (DMBLSj) (j=0~1, 3)	571
16.4.8	DMA Transfer Count Register j (DMACTj) (j=0~1, 3).....	572
16.4.9	DMA Transfer Count Reload Register j (DMRLDj) (j=0~1, 3).....	573
16.4.10	DMA Source Address Register j (DMSARj) (j=0~1, 3)	573
16.4.11	DMA Target Address Register j (DMDARj) (j=0~1, 3).....	573
16.4.12	DMA Start Enable Register i(DMAEN0)	574
16.4.13	DMA Base Address Register (DMABAR).....	574
16.4.14	DMAEN0 Set Register (DMASET0).....	575
16.4.15	DMAEN0 Reset Register (DMACLR0)	575
16.4.16	Operation of DMA	576
16.4.17	Boot Source.....	576
16.4.18	Normal Mode	577
16.4.19	Repeat Mode	580
16.4.20	Chain Transfer.....	583
16.5	Cautions When Using DMA.....	585
16.5.1	DMA Control Data and Vector Table Setup	585
16.5.2	DMA Control Data Region and DMA Vector Table Region Allocation	585
16.5.3	DMA Execution Clock Counts.....	586
16.5.4	DMA Response Time	587
16.5.5	DMA Boot Sources	587
16.5.6	Operation in Standby Mode	587
Chapter 17	A/D Converter.....	588
17.1	Functions of A/D Converter.....	588
17.2	Register Mapping.....	591
17.3	Register Description	592
17.3.1	A/D Converter Mode Register 0 (ADM0)	592
17.3.2	A/D Converter Mode Register 1 (ADM1)	596
17.3.3	A/D Converter Mode Register 2 (ADM2)	597
17.3.4	A/D Converter Trigger Mode Register (ADTRG)	598
17.3.5	Analog Input Channel Specification Register (ADS)	599
17.3.6	A/D Internal Channel Selection Register (ADICHS).....	601
17.3.7	12-Bit A/D Conversion Channel Identifier Register (ADSCHS).....	602

17.3.8	12-Bit A/D Conversion Result Register (ADCR)	602
17.3.9	8-Bit A/D Conversion Result Register (ADCRH)	603
17.3.10	A/D Converter Sampling Time Extension Control Register (ADSMPWAIT).....	603
17.3.11	Conversion Result Comparison Upper Limit Configuration Register (ADUL).....	603
17.3.12	Conversion Result Comparison Lower Limit Configuration Register (ADLL).....	604
17.4	Input Voltage and Conversion Results	605
17.5	Operation Mode of A/D Converter	606
17.5.1	Software Trigger Mode (Select Mode, Sequential Conversion Mode)	606
17.5.2	Software Trigger Mode (Select Mode, Single Conversion Mode).....	607
17.5.3	Software Trigger Mode (Scan Mode, Sequential Conversion Mode)	608
17.5.4	Software Trigger Mode (Scan Mode, Single Conversion Mode).....	609
17.5.5	Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode)	610
17.5.6	Hardware Trigger No-Wait Mode (Select Mode, Single Conversion Mode).....	611
17.5.7	Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode.....	612
17.5.8	Hardware Trigger No-Wait Mode (Scan Mode, Single Conversion Mode).....	613
17.5.9	Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode)	614
17.5.10	Hardware Trigger Wait Mode (Select Mode, Single Conversion Mode)	615
17.5.11	Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode)	616
17.5.12	Hardware Trigger Wait Mode (Scan Mode, Single Conversion Mode).....	617
17.6	A/D Converter Setup Flowchart	618
17.6.1	Software Trigger Mode Settings	618
17.6.2	Hardware Trigger No-Wait Mode Settings	619
17.6.3	Hardware Trigger Wait Mode Settings.....	620
Chapter 18	Analog Comparators (ACMP0/1)	621
18.1	Overview	621
18.2	Features.....	621
18.3	Block Diagram.....	622
18.4	Register Mapping.....	623
18.5	Register Description	624
18.5.1	Analog Comparator 0 Control Register 0 (C0CON0).....	624
18.5.2	Analog Comparator 0 Control Register 1 (C0CON1).....	625
18.5.3	Analog Comparator 1 Control Register 0 (C1CON0).....	626
18.5.4	Analog Comparator 1 Control Register 1 (C1CON1).....	627
18.5.5	Analog Comparator Event Control Register (CEVCON)	628
18.5.6	Analog Comparator Reference Voltage Control Register (CVRCON)	629
18.5.7	Analog Comparator Write Enable Register (ACMPLOCK)	629
Chapter 19	Universal CRC	630
19.1	CRC Overview	630
19.2	CRC Features.....	630
19.3	Function Description	631
19.4	Register Mapping.....	631
19.5	Register Description	631
19.5.1	CRC Data Register (CRC_DR).....	631
19.5.2	CRC Independent Data Register (CRC_IDR).....	631
19.5.3	CRC Control Register (CR)	631
Chapter 20	Standby Function	632

20.1 Standby Function.....	632
20.2 Sleep Mode.....	633
20.2.1 Sleep Mode Configuration.....	633
20.2.2 Release of Sleep Mode.....	634
20.3 Deep Sleep Mode.....	635
20.3.1 Deep Sleep Mode Configuration.....	635
20.3.2 Release of Deep Sleep Mode	636
Chapter 21 Nested Vector Interrupt Controller (NVIC).....	637
21.1 Features.....	637
21.2 Exception Modes and System Interrupt Mapping	637
21.3 Vector Table	638
21.4 External Interrupt Structure	639
21.5 Register Mapping.....	640
21.6 Register Description	641
21.6.1 External Interrupt Rising Edge Enable Register (EGP0).....	641
21.6.2 External Interrupt Rising Edge Enable Register (EGP1).....	642
21.6.3 External Interrupt Falling Edge Enable Register (EGN0).....	643
21.6.4 External Interrupt Falling Edge Enable Register (EGN1).....	644
21.6.5 External Interrupt Mask Register 0 (E_IMR0)	646
21.6.6 External Interrupt Mask Register 1 (E_IMR1)	647
21.6.7 External Interrupt Pending Register 0 (E_PR0).....	648
21.6.8 External Interrupt Pending Register 1 (E_PR1).....	649
21.6.9 External Interrupt Filter Control Register (FILCON).....	650
Chapter 22 Reset Function	651
22.1 Reset Timing.....	653
22.2 Registers for Confirming the Reset Source	655
22.2.1 Register Mapping.....	655
22.2.2 Reset Control Flag Register (RESF).....	655
Chapter 23 Power-On Reset Circuit.....	657
23.1 Functions of Power-On Reset Circuit.....	657
23.2 Structure of Power-On Reset Circuit.....	658
23.3 Operation of Power-On Reset Circuit.....	659
Chapter 24 Voltage Detection Circuit	663
24.1 Functions of Voltage Detection Circuit.....	663
24.2 Structure of Voltage Detection Circuit.....	664
24.3 Register Mapping.....	664
24.4 Registers for Controlling Voltage Detection Circuit.....	665
24.4.1 Voltage Detection Register (LVIM)	665
24.4.2 Voltage Detection Level Register (LVIS).....	666
24.5 Operation of Voltage Detection Circuit	667
24.5.1 Reset Mode Configuration.....	667
24.6 Interrupt Mode Configuration.....	668
24.7 Interrupt & Reset Mode Configuration.....	670
24.8 Cautions for Voltage Detection Circuit.....	675

Chapter 25	Safety Functions	677
25.1	Overview of Safety Functions	677
25.2	Register Mapping.....	678
25.3	Operation of Safety Functions	679
25.3.1	Flash Memory CRC Operation Function (High-Speed CRC).....	679
25.3.2	SFR Guard Function	681
25.3.3	A/D Test Function.....	682
25.3.4	Digital Output Signal Level Detection Function for Input/Output Pins.....	683
25.3.5	Product Unique ID Register	684
Chapter 26	Option Byte	685
26.1	Functions of Option Bytes	685
26.1.1	User Option Bytes (000C0H~000C2H)	685
26.2	Flash Memory Data Protection Option Bytes (000C3H, 500004H).....	686
26.3	Register Mapping.....	686
26.4	User Option Bytes.....	687
26.4.1	User Option Byte (000C0H)	687
26.4.2	User Option Byte (000C1H)	687
26.4.3	User Option Byte (000C2H)	690
26.4.4	Flash Memory Data Protection Option Byte (000C3H).....	690
26.4.5	Flash Memory Data Protection Option Byte (500004H)	690
Chapter 27	Flash Control	691
27.1	Description of Flash Control	691
27.2	Structure of Flash Memory	691
27.3	Register Mapping.....	692
27.4	Register Description	693
27.4.1	Flash Operation Control Register (FLCMD)	693
27.4.2	Flash Key Register (FLKEY).....	693
27.4.3	Flash Data Register 0 (FLDATA0).....	693
27.4.4	Flash Address Register (FLADDR)	693
27.4.5	Flash Control Register (FLCR).....	694
27.4.6	Flash Status Register (FLSR).....	695
27.4.7	Flash Region Control Register (FLREGION).....	695
27.4.8	Flash Protection Register (FLPRT).....	696
27.5	How to Operate the Flash	698
27.5.1	Programming.....	698
27.5.2	Sector Erase	699
27.5.3	Chip Erase	700
27.6	Cautions for Flash Operation.....	701
Chapter 28	Appendix Revision History	702

Chapter 1 CPU

1.1 Overview

This section provides a brief introduction to the features and debugging capabilities of the ARM Cortex-M0+ core used in this product. For detailed information, please refer to the relevant ARM documentation.

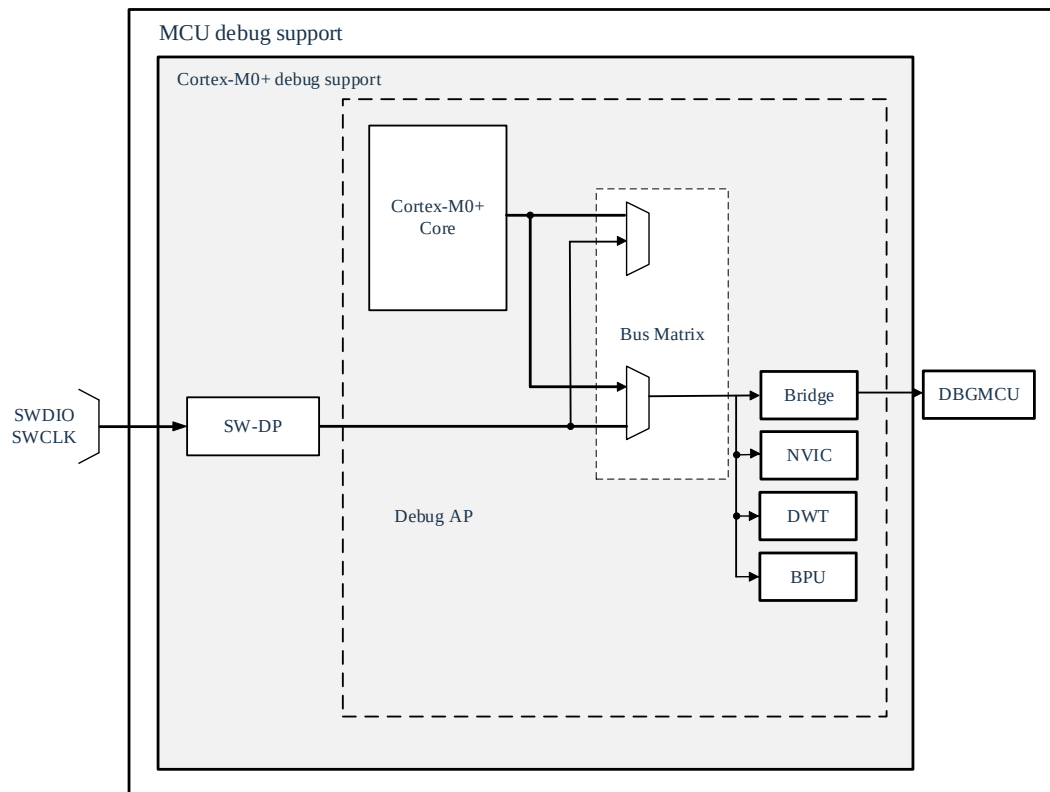
1.2 Cortex-M0+ Features

- The ARM Cortex-M0+ processor is a 32-bit RISC core with a 2-stage pipeline that only supports privileged mode
- 1-cycle hardware multiplier
- Nested Vector Interrupt Controller (NVIC)
 - 1 unshielded interrupt (NMI)
 - Supports 32 maskable interrupt requests (IRQ)
 - 4 interrupt priority levels
- The system timer SysTick is a 24-bit countdown timer that can be selected for F_{CLK} or F_{IL} count clocks
- Vector table offset register (VTOR)
 - The software can write VTOR to relocate the start address of the vector table to a different location
 - The default value for this register is 0x0000_0000. The lower 8 bits are ignored when writing, and reading returns zero, meaning the offset is 256-byte aligned.

1.3 Debug Features

- 2-wire SWD debug interface
- Supports for pausing, resuming, and single-step execution procedures
- Access processor's core registers and special function registers
- 4 hardware breakpoints (BPU)
- Unlimited software breakpoints (BKPT instruction)
- 2 data watch points (DWT)
- Memory access during kernel execution

Figure 1-1 Debug block diagram of Cortex-M0+



Caution: SWD cannot operate in Deep Sleep mode. Please perform debugging operations in Active or Sleep mode.

1.4 SWD Interface Pins and Debug Control

The two GPIOs of this product can be used as SWD interface pins, and these pins are available in all package variants.

Table 1-1 SWD debug port pin

SWD port name	Debug function	Pin assignment
SWCLK	Serial clock	PA14
SWDIO	Serial data input/output	PA13

When the SWD function is not in use, it can be disabled by setting the Debug Stop Control Register (DBGSTOPCR).

DBGSTOPCR address: 0x4001_B004

Bit	Symbol	Description	Reset value
31:25	-	Reserved	0
24	SWDIS	SWD debug interface disable 0: SWD debug interface enable. PA13/PA14 cannot be used as a GPIO in the state where the debugger is connected (because the ENO and DOUT for that IOBUF are now controlled by the debugger) 1: The SWD debug interface is disabled. PA13/PA14 can be used as a GPIO	0
23:14	-	Reserved	0
13	IICDB	In the state where the debugger is connected and the CPU is in the debug state (HALED=1), the IIC action/stop status 0: IIC acts 1: IIC stops	0
12	TIM17DB	In the state where the debugger is connected and the CPU is in the debug state (HALED=1), the TIM17 action/stop status 0: TIM17 acts 1: TIM17 stops	0
11	TIM16DB	In the state where the debugger is connected and the CPU is in the debug state (HALED=1), the TIM16 action/stop status 0: TIM16 acts 1: TIM16 stops	0
10	TIM1DB	In the state where the debugger is connected and the CPU is in the debug state (HALED=1), the TIM1 action/stop status 0: TIM1 acts 1: TIM1 stops	0
9:8	-	Reserved	0
7	TIM14DB	In the state where the debugger is connected and the CPU is in the debug state (HALED=1), the TIM14 action/stop status 0: TIM14 acts 1: TIM14 stops	0
6	TIM3DB	In the state where the debugger is connected and the CPU is in the debug state (HALED=1), the TIM3 action/stop status 0: TIM3 acts 1: TIM3 stops	0
5	WWDGDB	In the state where the debugger is connected and the CPU is in the debug state (HALED=1), the WWDG action/stop status 0: WWDG acts 1: WWDG stops	0
4	IWDGDB	In the state where the debugger is connected and the CPU is in the debug state (HALED=1), the IWDG action/stop status	0

		0: IWDG acts 1: IWDG stops	
3:2	-	Reserved	0
1	SAUDB	In the state where the debugger is connected and the CPU is in the debug state (HALED=1), the SAU action/stop status 0: SAU acts 1: SAU stops	0
0	LPTIMDB	In the state where the debugger is connected and the CPU is in the debug state (HALED=1), the LPTIM action/stop status 0: LPTIM acts 1: LPTIM stops	0

1.5 LOCKUP Control Register (LOCKCTL)

The LOCKCTL register is the configuration register for the Cortex-M0+ LockUp function, determining whether the brake inputs of TIM1/TIM16/TIM17 are enabled and whether the independent watchdog timer operates. PRCR is its write protection register.

The LOCKCTL register is set through an 8-bit memory manipulation instruction.

After generating a reset signal, the value of the LOCKCTL register becomes “01H”.

LOCKCTL address: 0x4002_0405

Bit	Symbol	Description	Reset value
7:2	-	Reserved	-
1	LOCKUP_BRKEN_TIM	LOCKUP function as TIM1/TIM16/TIM17 brake input enable control 0: Disable 1: Enable	0
0	LOCKUP_RST	LOCKUP function configuration 0: LOCKUP does not cause IWDG reset 1: LOCKUP causes IWDG reset	1

1.6 Protection Register (PRCR)

The LOCKCTL register is the configuration register for the Cortex-M0+ LockUp function, determining whether the brake inputs of TIM1/TIM16/TIM17 are enabled and whether the independent watchdog timer operates. PRCR is its write protection register.

The PRCR register is set through an 8-bit memory manipulation instruction.

After generating a reset signal, the value of the PRCR register becomes “00H”.

PRCR address: 0x4002_0406

Bit	Symbol	Description	Reset value
7:1	PRTKEY	PRCR write protection 78H: PRCR is writable Others: PRCR is not writable	0x0
0	PRCR	LOCKUP control register write protection 0: LOCKCTL register is not writable 1: LOCKCTL register is writable	0

1.7 ARM Reference Documentation

In the Cortex®-M0+ core, the built-in debug functionality is part of the ARM® CoreSight design suite. For related documentation, please refer to:

- Cortex®-M0+ Technical Reference Manual (TRM)
- ARM® Debug Interface V5
- ARM® CoreSight Design Kit Version r1p1 Technical Reference Manual
- ARM® CoreSight™ MTB-M0+ Technical Reference Manual

Chapter 2 Pin Functions

2.1 Port Functions

For detailed information, please refer to the datasheet of the corresponding product series.

2.2 Port Alternate Functions

For specific port alternate functions, please refer to the datasheet of each product series. For detailed port alternate functions, refer to Table 2-1 below.

Table 2-1 Port alternate function digital mapping table

	USART	SPI	SAU	I2C	TIM1	TIM3	TIM14	TIM16	TIM17
PA0	CTS1	MISO	SCK00/SCL00		CH1N/CH3				
PA1	RTS1	MOSI/SCK			CH2N/CH4				
PA2	TXD1	MOSI/SCK		SDA		CH1			
PA3	RXD1	MOSI	SDI00/SDA00/RXD0	SCL	CH1				
PA4	CK1	NSS	SDO00/TXD0			CH3	CH1		
PA5		SCK				CH2			
PA6	CK1	MISO			BKIN	CH1		CH1	
PA7	TXD1	MISO/MOSI		SDA	CH1N	CH2	CH1		CH1
PA8	RXD1/CK1	MOSI	SS00	SCL	CH1				
PA9	RXD1/TXD1	SCK	SDI00/SDA00/RXD0	SDA/SCL	BKIN/CH2				
PA10	TXD1/RXD1	NSS	SDO00/TXD0	SCL/SDA	CH3				BKIN
PA11	CTS1	MISO		SCL	CH4				
PA12	RTS1	MOSI		SDA	ETR				
PA13	RXD1	MISO			CH2				
PA14	TXD1								
PA15	RXD1	NSS							
PB0		NSS			CH2N	CH3			
PB1					CH3N	CH4	CH1		
PB2	RXD1		SCK00/SCL00						
PB3	RTS1	SCK			CH2				
PB4	CTS1	MISO				CH1			BKIN
PB5	CK1	MOSI		SMBA		CH2		BKIN	
PB6	TXD1		SDI00/SDA00/RXD0	SCL	CH3			CH1N	
PB7	RXD1		SDO00/TXD0	SDA					CHIN
PB8	TXD1		SCK00/SCL00/SS00	SDA				CH1	CH1
PF0	RXD1		SCK00/SCL00	SDA			CH1		
PF1	TXD1	NSS	SDI00/SDA00/RXD0	SCL			CH1		
PF2			SDO00/TXD0						
PF3	TXD1	NSS	SDI00/SDA00/RXD0			CH3			
PF4									

Note 1: The IO alternate function of this product requires the user to configure the PMC, PM, and other registers separately.

Note 2: When the IIC function is selected, the open-drain function is automatically enabled.

Note 3: For alternated functions, which can be used as both input and output, after selecting PS_XX, the input channel is automatically enabled.

Table 2-2: Analog function and special function pins

	ADC	ACMP	OSC	SWD	NRST
PA0	ANI0	C0N0/C0OUT0			
PA1	ANI1	C0P0			
PA2	ANI2	C1N0/C1OUT0			
PA3	ANI3	C1P1			
PA4	ANI4				
PA5	ANI5				
PA6	ANI6	C0OUT1			
PA7	ANI7	C1OUT1			
PA8	ANI24				
PA9	ANI25		OSC32_OUT		
PA10	ANI26		OSC32_IN		
PA11	ANI27	C0OUT3			
PA12	ANI28	C1OUT2			
PA13	ANI29			SWDIO	
PA14	ANI30			SWCLK	
PA15	ANI31				
PB0	ANI8	C0OUT2			
PB1	ANI9	C0N1			
PB2	ANI10	C0P1			
PB3	ANI11	C1N1			
PB4	ANI12	C1P2			
PB5	ANI13	C0OUT4			
PB6	ANI14	C1P3			
PB7	ANI15	C1N2			
PB8	ANI32	C0P2			
PF0	ANI16				
PF1	ANI17				
PF2	ANI18				NRST
PF3	ANI19	C1P0			
PF4	ANI20				

2.3 Register Mapping

2.3.1 Control Function Register Mapping

(Base address of the port control register = 0x4004_0000) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
PA	0x000	R/W	Register to set the port output latch value in 1-bit units; in input mode, reading this register gives the pin level; in output mode, reading this register gives the value of the port's output latch.	0x00
PB	0x002	R/W		0x00
PF	0x00A	R/W		0x00
PMA	0x020	R/W	When the port is used as a digital channel, set the port input or output register in 1-bit units.	0xFF
PMB	0x022	R/W		0xFF
PMF	0x02A	R/W		0xFF
PUA	0x030	R/W	Port internal pull-up resistor selection register	0x6000
PUB	0x032	R/W		0x00
PUF	0x03A	R/W		0x04
PDA	0x040	R/W	Port internal pull-down resistor selection register	0x00
PDB	0x042	R/W		0x00
PDF	0x04A	R/W		0x00
POMA	0x050	R/W	Open-drain mode register, the N-channel drain open circuit is only enabled when the port is configured as output mode.	0x00
POMB	0x052	R/W		0x00
POMF	0x05A	R/W		0x00
PMCA	0x060	R/W	Port mode register, set the port to use as digital or analog channels in 1-bit units.	0x9FFF
PMCB	0x062	R/W		0xFFFF
PMCF	0x06A	R/W		0x18
PSETA	0x070	WO	Register to set the port output latch register in 1-bit units	0x00
PSETB	0x072	WO		0x00
PSETF	0x07A	WO		0x00
PCLRA	0x080	WO	Register to clear the port output latch register in 1-bit units	0x00
PCLRB	0x082	WO		0x00
PCLRF	0x08A	WO		0x00
PREADA	0x090	RO	When the port is used as a digital GPIO, the corresponding port level can be obtained by reading this register.	0xx
PREADB	0x092	RO		0xx
PREADF	0x09A	RO		0xx
PTTLA	0x0A0	R/W	GPIO TTL/SIMT input selection, default is SMIT input.	0x00
PTTLF	0x0AA	R/W		

2.3.2 I/O Alternate Function Register Mapping

(Base address of the input/output alternate function register =0x40040800)

RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value	Condition	Priority
PS_INTP0	0x00	R/W	External interrupt 0 input port selection register	0x3	PMC=0, PM=1	High
PS_INTP1	0x01	R/W	External interrupt 1 input port selection register	0x3	PMC=0, PM=1	
PS_INTP2	0x02	R/W	External interrupt 2 input port selection register	0x3	PMC=0, PM=1	
PS_INTP3	0x03	R/W	External interrupt 3 input port selection register	0x3	PMC=0, PM=1	
PS_INTP4	0x04	R/W	External interrupt 4 input port selection register	0x3	PMC=0, PM=1	
PS_INTP5	0x05	R/W	External interrupt 5 input port selection register	0x3	PMC=0, PM=1	
PS_INTP6	0x06	R/W	External interrupt 6 input port selection register	0x3	PMC=0, PM=1	
PS_INTP7	0x07	R/W	External interrupt 7 input port selection register	0x3	PMC=0, PM=1	
PS_INTP8	0x08	R/W	External interrupt 8 input port selection register	0x3	PMC=0, PM=1	
PS_SCL	0x10	R/W	IIC SCL port selection register	0x7	PMC=0, PM=0	
PS_SDA	0x11	R/W	IIC SDA port selection register	0xf	PMC=0, PM=0	
PS_SMBA	0x12	R/W	IIC SMBA port selection register	0x1	PMC=0, PM=0	
PS_SCK	0x13	R/W	SPI SCK port selection register	0x7	SPI master: PMC=0, PM=0 SPI slave: PMC=0, PM=1	
PS_MISO	0x14	R/W	SPI MISO port selection register	0x7	SPI master: PMC=0, PM=1 SPI slave: PMC=0, PM=0	
PS_MOSI	0x15	R/W	SPI MOSI port selection register	0x7	SPI master: PMC=0, PM=0 SPI slave: PMC=0, PM=1	
PS_NSS	0x16	R/W	SPI NSS port selection register	0x7	SPI master: PMC=0, PM=0 SPI slave: PMC=0, PM=1	
PS_SCK00	0x17	R/W	SPI00 SCLK /IIC00 SCL port selection register	0x7	SPI00 master: PMC=0, PM=0 SPI00 slave: PMC=0, PM=1 IIC00 master/slave: PMC=0, PM=0	
PS_SDI00	0x18	R/W	SPI00 MISI/IIC00 SDA /UART0 RXD port selection register	0x7	PMC=0, PM=1	
PS_SDO00	0x19	R/W	SPI00 MOSI/UART0 TXD port selection register	0x7	PMC=0, PM=0	
PS_SS00	0x1A	R/W	SPI00 SS port selection register	0x3	SPI00 slave: PMC=0, PM=1	
PS_TXD1	0x20	R/W	USART TXD port selection register	0xf	PMC=0, PM=0	
PS_RXD1	0x21	R/W	USART RXD port selection register	0xf	PMC=0, PM=1	
PS_CTS1	0x22	R/W	USART CTS port selection register	0x3	PMC=0, PM=1	
PS_RTS1	0x23	R/W	USART RTS port selection register	0x3	PMC=0, PM=0	
PS_CK1	0x24	R/W	USART CK port selection register	0x7	PMC=0, PM=0	
PS_T1CH1	0x30	R/W	TIM1 CH1 port selection register	0x3	When inputting: PMC=0, PM=1 When outputting: PMC=0, PM=0	
PS_T1CH2	0x31	R/W	TIM1 CH2 port selection register	0x3	When inputting: PMC=0, PM=1 When outputting: PMC=0, PM=0	
PS_T1CH3	0x32	R/W	TIM1 CH3 port selection register	0x3	When inputting: PMC=0, PM=1 When outputting: PMC=0, PM=0	
PS_T1CH4	0x33	R/W	TIM1 CH4 port selection register	0x3	When inputting: PMC=0, PM=1 When outputting: PMC=0, PM=0	
PS_T1CH1N	0x34	R/W	TIM1 CH1N port selection register	0x3	PMC=0, PM=0	
PS_T1CH2N	0x35	R/W	TIM1 CH2N port selection register	0x3	PMC=0, PM=0	

PS_T1BK	0x36	R/W	TIM1 BKIN port selection register	0x3	PMC=0, PM=1	Low
PS_T3CH1	0x37	R/W	TIM3 CH1 port selection register	0x3	When inputting: PMC=0, PM=1 When outputting: PMC=0, PM=0	
PS_T3CH2	0x38	R/W	TIM3 CH2 port selection register	0x3	When inputting: PMC=0, PM=1 When outputting: PMC=0, PM=0	
PS_T3CH3	0x39	R/W	TIM3 CH3 port selection register	0x3	When inputting: PMC=0, PM=1 When outputting: PMC=0, PM=0	
PS_T14CH1	0x3A	R/W	TIM14 CH1 port selection register	0x7	When inputting: PMC=0, PM=1 When outputting: PMC=0, PM=0	
PS_T16CH1	0x3B	R/W	TIM16 CH1 port selection register	0x3	When inputting: PMC=0, PM=1 When outputting: PMC=0, PM=0	
PS_T17CH1	0x3C	R/W	TIM17 CH1 port selection register	0x3	When inputting: PMC=0, PM=1 When outputting: PMC=0, PM=0	
PS_T17BK	0x3D	R/W	TIM17 BKIN port selection register	0x3	PMC=0, PM=1	
PS_C0O	0x60	R/W	Comparator 0 port selection register	0x7	PMC=0, PM=0	
PS_C1O	0x61	R/W	Comparator 1 port selection register	0x3	PMC=0, PM=0	Low
PMS	0x7B	R/W	Port read mode selection register; when the port is in output mode, this register selects whether to read the port latch value or the pin output level.	0x00		

2.3.3 Special Function Port RESINB Control Register Mapping

(Register base address=0x4002_0400)

RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
RSTM	0x0B	R/W	Register to select RESINB (PF2) pin as external reset pin or GPIO pin	0x00

2.4 Register Description

The ports are controlled through the following registers.

- ◆ Port register (Px)
- ◆ Port mode register (PMx)
- ◆ Pull-up resistor selection register (PUx)
- ◆ Pull-down resistor selection register (PDx)
- ◆ Port output mode register (POMx)
- ◆ Port mode control register (PMCx)
- ◆ Port set control register (PSETx)
- ◆ Port clear control register (PCLR_x)
- ◆ Port input alternate function configuration register (PS_XX)
- ◆ Port level selection register (PTTL_x)
- ◆ Special function port RESINB control register (RSTM)

2.4.1 Port Register (Px)

This is the register (Px, where x = A, B, F) used to set the port output latch value in 1-bit units. In input mode, reading this register gives the pin level, while in output mode, reading it gives the value of the port output latch. After a reset signal is generated, the values of these registers are set to “00H”. The specific register descriptions are as follows:

Bit	Symbol	Description	Reset value
15	Px15	Port x mode register bit 15 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0
14	Px14	Port x mode register bit 14 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0
13	Px13	Port x mode register bit 13 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0
12	Px12	Port x mode register bit 12 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0
11	Px11	Port x mode register bit 11 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0
10	Px10	Port x mode register bit 10 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0
9	Px9	Port x mode register bit 9 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0
8	Px8	Port x mode register bit 8 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0
7	Px7	Port x mode register bit 7 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0
6	Px6	Port x mode register bit 6 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0
5	Px5	Port x mode register bit 5 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0
4	Px4	Port x mode register bit 4 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0
3	Px3	Port x mode register bit 3 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0
2	Px2	Port x mode register bit 2 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0
1	Px1	Port x mode register bit 1 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0

0	Px0	Port x mode register bit 0 0: In output mode, output “0”; in input mode, input low level. 1: In output mode, output “1”; in input mode, input high level.	0

Note: Be sure to set bits that are not mounted to their initial values.

2.4.2 Port Mode Register (PMx)

When the port is used as a digital channel, this register (PMx, where x = A, B, F) is used to set its input/output function in 1-bit units. After a reset signal is generated, all ports default to the input state. The specific register descriptions are as follows:

Bit	Symbol	Description	Reset value
15	PMx15	Port x mode register bit 15 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1
14	PMx14	Port x mode register bit 14 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1
13	PMx13	Port x mode register bit 13 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1
12	PMx12	Port x mode register bit 12 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1
11	PMx11	Port x mode register bit 11 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1
10	PMx10	Port x mode register bit 10 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1
9	PMx9	Port x mode register bit 9 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1
8	PMx8	Port x mode register bit 8 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1
7	PMx7	Port x mode register bit 7 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1
6	PMx6	Port x mode register bit 6 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1
5	PMx5	Port x mode register bit 5 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1
4	PMx4	Port x mode register bit 4 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1
3	PMx3	Port x mode register bit 3 0: Output mode (used as an output port with output buffer	1

		ON) 1: Input mode (used as an input port with output buffer OFF)	
2	PMx2	Port x mode register bit 2 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1
1	PMx1	Port x mode register bit 1 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1
0	PMx0	Port x mode register bit 0 0: Output mode (used as an output port with output buffer ON) 1: Input mode (used as an input port with output buffer OFF)	1

Note: Be sure to set bits that are not mounted to their initial values.

2.4.3 Pull-Up Resistor Selection Register (PUx)

The register for selecting internal pull-up resistors is PUx (where x = A, B, F). The pull-up resistor can only be enabled when the corresponding bit in the PMCx register is set to 0.

After a reset signal is generated, the pull-up function for the PA13 and PF2 ports is automatically enabled, while the pull-up function for other ports is disabled by default. The specific register descriptions are as follows:

Bit	Symbol	Description	Reset value
15	PUx15	Px pin internal pull-up resistor selection bit 15 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	0
14	PUx14	Px pin internal pull-up resistor selection bit 14 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	0
13	PUx13	Px pin internal pull-up resistor selection bit 13 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	When x = A, the reset value is 1; When x = other, the reset value is 0.
12	PUx12	Px pin internal pull-up resistor selection bit 12 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	0
11	PUx11	Px pin internal pull-up resistor selection bit 11 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	0
10	PUx10	Px pin internal pull-up resistor selection bit 10 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	0
9	PUx9	Px pin internal pull-up resistor selection bit 9 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	0
8	PUx8	Px pin internal pull-up resistor selection bit 8 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	0
7	PUx7	Px pin internal pull-up resistor selection bit 7 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	0
6	PUx6	Px pin internal pull-up resistor selection bit 6 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	0
5	PUx5	Px pin internal pull-up resistor selection bit 5 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	0
4	PUx4	Px pin internal pull-up resistor selection bit 4 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	0
3	PUx3	Px pin internal pull-up resistor selection bit 3 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	0

2	PUx2	Px pin internal pull-up resistor selection bit 2 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	When x = F, the reset value is 1; When x = other, the reset value is 0.
1	PUx1	Px pin internal pull-up resistor selection bit 1 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	0
0	PUx0	Px pin internal pull-up resistor selection bit 0 0: No internal pull-up resistor connected 1: Internal pull-up resistor connected	0

Note: Be sure to set bits that are not mounted to their initial values.

2.4.4 Pull-Down Resistor Selection Register (PDx)

The register for selecting internal pull-down resistors is PDx (where x = A, B, F). The pull-down resistor can only be enabled when the corresponding bit in the PMCx register is set to 0.

After a reset signal is generated, the pull-down function for the PA14 port is automatically enabled, while the pull-down function for other ports is disabled by default. The specific register descriptions are as follows:

Bit	Symbol	Description	Reset value
15	PDx15	Px pin internal pull-down resistor selection bit 15 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	0
14	PDx14	Px pin internal pull-down resistor selection bit 14 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	When x = A, the reset value is 1; When x = other, the reset value is 0.
13	PDx13	Px pin internal pull-down resistor selection bit 13 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	0
12	PDx12	Px pin internal pull-down resistor selection bit 12 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	0
11	PDx11	Px pin internal pull-down resistor selection bit 11 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	0
10	PDx10	Px pin internal pull-down resistor selection bit 10 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	0
9	PDx9	Px pin internal pull-down resistor selection bit 9 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	0
8	PDx8	Px pin internal pull-down resistor selection bit 8 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	0
7	PDx7	Px pin internal pull-down resistor selection bit 7 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	0
6	PDx6	Px pin internal pull-down resistor selection bit 6 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	0
5	PDx5	Px pin internal pull-down resistor selection bit 5 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	0
4	PDx4	Px pin internal pull-down resistor selection bit 4 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	0
3	PDx3	Px pin internal pull-down resistor selection bit 3 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	0
2	PDx2	Px pin internal pull-down resistor selection bit 2 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	0
1	PDx1	Px pin internal pull-down resistor selection bit 1 0: No internal pull-down resistor connected	0

		1: Internal pull-down resistor connected	
0	PDx0	Px pin internal pull-down resistor selection bit 0 0: No internal pull-down resistor connected 1: Internal pull-down resistor connected	0

Note: Be sure to set bits that are not mounted to their initial values.

2.4.5 Port Output Mode Register (POMx)

The port output mode register POMx (where x = A, B, F) is enabled only when configured in output mode with an N-channel open-drain. When the port is alternated for IIC functionality, the open-drain mode is forcibly enabled.

After a reset signal is generated, the values of these registers are set to “00H”.

Caution: For bits set to N-channel open-drain output mode (POMmn=1), the hardware does not automatically connect the internal pull-up resistor. If the user needs to enable the pull-up resistor, it must be manually enabled.

The specific register descriptions are as follows:

Bit	Symbol	Description	Reset value
15	POMx15	Px pin output mode selection bit 15 0: Normal output mode 1: N-channel open-drain output mode	0
14	POMx14	Px pin output mode selection bit 14 0: Normal output mode 1: N-channel open-drain output mode	0
13	POMx13	Px pin output mode selection bit 13 0: Normal output mode 1: N-channel open-drain output mode	0
12	POMx12	Px pin output mode selection bit 12 0: Normal output mode 1: N-channel open-drain output mode	0
11	POMx11	Px pin output mode selection bit 11 0: Normal output mode 1: N-channel open-drain output mode	0
10	POMx10	Px pin output mode selection bit 10 0: Normal output mode 1: N-channel open-drain output mode	0
9	POMx9	Px pin output mode selection bit 9 0: Normal output mode 1: N-channel open-drain output mode	0
8	POMx8	Px pin output mode selection bit 8 0: Normal output mode 1: N-channel open-drain output mode	0
7	POMx7	Px pin output mode selection bit 7 0: Normal output mode 1: N-channel open-drain output mode	0
6	POMx6	Px pin output mode selection bit 6 0: Normal output mode 1: N-channel open-drain output mode	0
5	POMx5	Px pin output mode selection bit 5 0: Normal output mode 1: N-channel open-drain output mode	0
4	POMx4	Px pin output mode selection bit 4 0: Normal output mode 1: N-channel open-drain output mode	0
3	POMx3	Px pin output mode selection bit 3 0: Normal output mode 1: N-channel open-drain output mode	0
2	POMx2	Px pin output mode selection bit 2 0: Normal output mode 1: N-channel open-drain output mode	0

1	POMx1	Px pin output mode selection bit 1 0: Normal output mode 1: N-channel open-drain output mode	0
0	POMx0	Px pin output mode selection bit 0 0: Normal output mode 1: N-channel open-drain output mode	0

Note: Be sure to set bits that are not mounted to their initial values.

2.4.6 Port Mode Control Register (PMCx)

The port mode register PMCx (where x = A, B, F) is used to set the port as either a digital (input/output) or analog (input) channel in 1-bit units.

After a reset signal is generated, PA13, PA14, PF2, PF0, and PF1 default to being used as digital channels, while other ports default to being used as analog channels, meaning the corresponding bit in PMCx is set to 1.

The specific register descriptions are as follows:

Bit	Symbol	Description	Reset value
15	PMCx15	Px pin digital (input/output) or analog (input) bit 15 0: Digital input/output (alternated functions other than analog input) 1: Analog input	1
14	PMCx14	Px pin digital (input/output) or analog (input) bit 14 0: Digital input/output (alternated functions other than analog input) 1: Analog input	When x = A, the reset value is 0; When x = other, the reset value is 1.
13	PMCx13	Px pin digital (input/output) or analog (input) bit 13 0: Digital input/output (alternated functions other than analog input) 1: Analog input	When x = A, the reset value is 0; When x = other, the reset value is 1.
12	PMCx12	Px pin digital (input/output) or analog (input) bit 12 0: Digital input/output (alternated functions other than analog input) 1: Analog input	1
11	PMCx11	Px pin digital (input/output) or analog (input) bit 11 0: Digital input/output (alternated functions other than analog input) 1: Analog input	1
10	PMCx10	Px pin digital (input/output) or analog (input) bit 10 0: Digital input/output (alternated functions other than analog input) 1: Analog input	1
9	PMCx9	Px pin digital (input/output) or analog (input) bit 9 0: Digital input/output (alternated functions other than analog input) 1: Analog input	1
8	PMCx8	Px pin digital (input/output) or analog (input) bit 8 0: Digital input/output (alternated functions other than analog input) 1: Analog input	1
7	PMCx7	Px pin digital (input/output) or analog (input) bit 7 0: Digital input/output (alternated functions other than analog input) 1: Analog input	1
6	PMCx6	Px pin digital (input/output) or analog (input) bit 6 0: Digital input/output (alternated functions other than analog input) 1: Analog input	1
5	PMCx5	Px pin digital (input/output) or analog (input) bit 5 0: Digital input/output (alternated functions other than analog input) 1: Analog input	1
4	PMCx4	Px pin digital (input/output) or analog (input) bit 4 0: Digital input/output (alternated functions other than analog input)	1

		1: Analog input	
3	PMCx3	Px pin digital (input/output) or analog (input) bit 3 0: Digital input/output (alternated functions other than analog input) 1: Analog input	1
2	PMCx2	Px pin digital (input/output) or analog (input) bit 2 0: Digital input/output (alternated functions other than analog input) 1: Analog input	When x = F, the reset value is 0; When x = other, the reset value is 1.
1	PMCx1	Px pin digital (input/output) or analog (input) bit 1 0: Digital input/output (alternated functions other than analog input) 1: Analog input	When x = F, the reset value is 0; When x = other, the reset value is 1.
0	PMCx0	Px pin digital (input/output) or analog (input) bit 0 0: Digital input/output (alternated functions other than analog input) 1: Analog input	When x = F, the reset value is 0; When x = other, the reset value is 1.

Note: Be sure to set bits that are not mounted to their initial values.

2.4.7 Port Set Control Register (PSETx)

This is the register PSETx (where x = A, B, F) used to set the port output latch value in 1-bit units.

After a reset signal is generated, the values of these registers are set to “00H”.

The specific register descriptions are as follows:

Bit	Symbol	Description	Reset value
15	PSETx15	Px pin set control bit 15 0: No operation 1: Set the corresponding Px15 to 1.	0
14	PSETx14	Px pin set control bit 14 0: No operation 1: Set the corresponding Px14 to 1.	0
13	PSETx13	Px pin set control bit 13 0: No operation 1: Set the corresponding Px13 to 1.	0
12	PSETx12	Px pin set control bit 12 0: No operation 1: Set the corresponding Px12 to 1.	0
11	PSETx11	Px pin set control bit 11 0: No operation 1: Set the corresponding Px11 to 1.	0
10	PSETx10	Px pin set control bit 10 0: No operation 1: Set the corresponding Px10 to 1.	0
9	PSETx9	Px pin set control bit 9 0: No operation 1: Set the corresponding Px9 to 1.	0
8	PSETx8	Px pin set control bit 8 0: No operation 1: Set the corresponding Px8 to 1.	0
7	PSETx7	Px pin set control bit 7 0: No operation 1: Set the corresponding Px7 to 1.	0
6	PSETx6	Px pin set control bit 6 0: No operation 1: Set the corresponding Px6 to 1.	0
5	PSETx5	Px pin set control bit 5 0: No operation 1: Set the corresponding Px5 to 1.	0
4	PSETx4	Px pin set control bit 4 0: No operation 1: Set the corresponding Px4 to 1.	0
3	PSETx3	Px pin set control bit 3 0: No operation 1: Set the corresponding Px3 to 1.	0
2	PSETx2	Px pin set control bit 2 0: No operation 1: Set the corresponding Px2 to 1.	0
1	PSETx1	Px pin set control bit 1 0: No operation 1: Set the corresponding Px1 to 1.	0
0	PSETx0	Px pin set control bit 0	0

		0: No operation 1: Set the corresponding Px0 to 1.	

Note: Be sure to set bits that are not mounted to their initial values.

2.4.8 Port Clear Control Register (PCLR_x)

This is the register PCLR_x (where x = A, B, F) used to clear the port output latch value in 1-bit units.

After a reset signal is generated, the values of these registers are set to “00H”.

The specific register descriptions are as follows:

Bit	Symbol	Description	Reset value
15	PCLR _x 15	Px pin clear control bit 15 0: No operation 1: Clear the corresponding Px15.	0
14	PCLR _x 14	Px pin clear control bit 14 0: No operation 1: Clear the corresponding Px14.	0
13	PCLR _x 13	Px pin clear control bit 13 0: No operation 1: Clear the corresponding Px13.	0
12	PCLR _x 12	Px pin clear control bit 12 0: No operation 1: Clear the corresponding Px12.	0
11	PCLR _x 11	Px pin clear control bit 11 0: No operation 1: Clear the corresponding Px11.	0
10	PCLR _x 10	Px pin clear control bit 10 0: No operation 1: Clear the corresponding Px10.	0
9	PCLR _x 9	Px pin clear control bit 9 0: No operation 1: Clear the corresponding Px9.	0
8	PCLR _x 8	Px pin clear control bit 8 0: No operation 1: Clear the corresponding Px8.	0
7	PCLR _x 7	Px pin clear control bit 7 0: No operation 1: Clear the corresponding Px7.	0
6	PCLR _x 6	Px pin clear control bit 6 0: No operation 1: Clear the corresponding Px6.	0
5	PCLR _x 5	Px pin clear control bit 5 0: No operation 1: Clear the corresponding Px5.	0
4	PCLR _x 4	Px pin clear control bit 4 0: No operation 1: Clear the corresponding Px4.	0
3	PCLR _x 3	Px pin clear control bit 3 0: No operation 1: Clear the corresponding Px3.	0
2	PCLR _x 2	Px pin clear control bit 2 0: No operation 1: Clear the corresponding Px2.	0
1	PCLR _x 1	Px pin clear control bit 1 0: No operation 1: Clear the corresponding Px1.	0
0	PCLR _x 0	Px pin clear control bit 0	0

		0: No operation 1: Clear the corresponding Px0.	

Note: Be sure to set bits that are not mounted to their initial values.

2.4.9 Port Readback Register (PREADx)

This is a read-only register PREADx (where x = A, B, F). When the port is configured as a digital GPIO, the corresponding port level can be read through this register.

Bit	Symbol	Description	Reset value
15	PREADx15	Px pin port level bit 15 0: Port is at low level 1: Port is at high level	x
14	PREADx14	Px pin port level bit 14 0: Port is at low level 1: Port is at high level	x
13	PREADx13	Px pin port level bit 13 0: Port is at low level 1: Port is at high level	x
12	PREADx12	Px pin port level bit 12 0: Port is at low level 1: Port is at high level	x
11	PREADx11	Px pin port level bit 11 0: Port is at low level 1: Port is at high level	x
10	PREADx10	Px pin port level bit 10 0: Port is at low level 1: Port is at high level	x
9	PREADx9	Px pin port level bit 9 0: Port is at low level 1: Port is at high level	x
8	PREADx8	Px pin port level bit 8 0: Port is at low level 1: Port is at high level	x
7	PREADx7	Px pin port level bit 7 0: Port is at low level 1: Port is at high level	x
6	PREADx6	Px pin port level bit 6 0: Port is at low level 1: Port is at high level	X
5	PREADx5	Px pin port level bit 5 0: Port is at low level 1: Port is at high level	x
4	PREADx4	Px pin port level bit 4 0: Port is at low level 1: Port is at high level	x
3	PREADx3	Px pin port level bit 3 0: Port is at low level 1: Port is at high level	x
2	PREADx2	Px pin port level bit 2 0: Port is at low level 1: Port is at high level	x

1	PREADx1	Px pin port level bit 1 0: Port is at low level 1: Port is at high level	x
0	PREADx0	Px pin port level bit 0 0: Port is at low level 1: Port is at high level	x

2.4.10 Port Mode Selection Register (PMS)

This register selects whether to read the output latch value of the port or the output level of the pin when the pin is configured in output mode (when the PMmn bit in the port mode register (PMm) is set to “0”).

The PMS register is set through an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register is set to “00H”.

Bit	Symbol	Description	Reset value
7:1	-	Reserved	-
0	PMS0	Data read selection when the pin is in output mode 0: Read the value of the Pmn register. 1: Read the digital output level of the pin.	0

2.4.11 TTL and Schmitt Input Selection (PTTLx)

This is a register PTTLx (where x = A, F) used to select the input level in 1-bit units.

After a reset signal is generated, the values of these registers are set to “00H”.

The specific register descriptions are as follows:

Bit	Symbol	Description	Reset value
15	PTTLx15	Px pin input level select control bit 15 0: Schmitt trigger input 1: TTL Input	0
14	PTTLx14	Px pin input level select control bit 14 0: Schmitt trigger input 1: TTL Input	0
13	PTTLx13	Px pin input level select control bit 13 0: Schmitt trigger input 1: TTL Input	0
12	PTTLx12	Px pin input level select control bit 12 0: Schmitt trigger input 1: TTL Input	0
11	PTTLx11	Px pin input level select control bit 11 0: Schmitt trigger input 1: TTL Input	0
10	PTTLx10	Px pin input level select control bit 10 0: Schmitt trigger input 1: TTL Input	0
9	PTTLx9	Px pin input level select control bit 9 0: Schmitt trigger input 1: TTL Input	0
8	PTTLx8	Px pin input level select control bit 8 0: Schmitt trigger input 1: TTL Input	0
7	PTTLx7	Px pin input level select control bit 7 0: Schmitt trigger input 1: TTL Input	0
6	PTTLx6	Px pin input level select control bit 6	0

		0: Schmitt trigger input 1: TTL Input	
5	PTTLx5	Px pin input level select control bit 5 0: Schmitt trigger input 1: TTL Input	0
4	PTTLx4	Px pin input level select control bit 4 0: Schmitt trigger input 1: TTL Input	0
3	PTTLx3	Px pin input level select control bit 3 0: Schmitt trigger input 1: TTL Input	0
2	PTTLx2	Px pin input level select control bit 2 0: Schmitt trigger input 1: TTL Input	0
1	PTTLx1	Px pin input level select control bit 1 0: Schmitt trigger input 1: TTL Input	0
0	PTTLx0	Px pin input level select control bit 0 0: Schmitt trigger input 1: TTL Input	0

Note: Be sure to set bits that are not mounted to their initial values. Only PA2/PA3, PF0/PF1 support TTL and Schmitt Trigger input selection.

2.4.12 Port Output Alternate Function Configuration Register (PS_XX)

The port output alternate function configuration register PS_XX allows mapping the output functions of some peripheral modules to the port. The specific mapping functions are detailed in Table 2-1. The reset value of the port output alternate function configuration register is “FFH”, at which point the port is set to its default alternated and GPIO functions. The setup steps are as follows:

- ◆ Set PMCmn = 0, select the digital function
- ◆ Set the input or output function according to the PS_XX register function (refer to the 2.3.2 list)
- ◆ Set the PS_XX register to select the corresponding pin
- ◆ Enable the corresponding module function (e.g., IIC, SPI, etc.)

Notes:

- (1) After setting the IIC_SCL/IIC_SDA function using the above steps, the open-drain function will be automatically enabled. If a pull-up resistor is needed, you can add an external pull-up or set the PUX register separately.
- (2) When selecting GPIO through the PS_XX register, it means that this function is unique and cannot be output by multiple GPIOs simultaneously.
- (3) When selecting different functions using PS_XX, the output has priority (PMx = 0). The priority order is described in the 2.3.2 table.

2.4.12.1 INTP0 Input Port Configuration Register (PS_INTP0)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_INTP0[1:0]	PS_INTP0 input selection 0x00: PA0's corresponding INTP0 input function 0x01: PB0's corresponding INTP0 input function 0x02: PF0's corresponding INTP0 input function Other: GPIO function	0x03

2.4.12.2 INTP1 Input Port Configuration Register (PS_INTP1)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_INTP1[1:0]	PS_INTP1 input selection 0x00: PA1's corresponding INTP1 input function 0x01: PB1's corresponding INTP1 input function 0x02: PF1's corresponding INTP1 input function Other: GPIO function	0x03

2.4.12.3 INTP2 Input Port Configuration Register (PS_INTP2)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_INTP2[1:0]	PS_INTP2 input selection 0x00: PA2's corresponding INTP2 input function 0x01: PB2's corresponding INTP2 input function 0x02: PF2's corresponding INTP2 input function Other: GPIO function	0x03

2.4.12.4 INTP3 Input Port Configuration Register (PS_INTP3)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_INTP3[1:0]	PS_INTP3 input selection 0x00: PA3's corresponding INTP3 input function 0x01: PB3's corresponding INTP3 input function 0x02: PF3's corresponding INTP3 input function Other: GPIO function	0x03

2.4.12.5 INTP4 Input Port Configuration Register (PS_INTP4)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_INTP4[1:0]	PS_INTP4 input selection 0x00: PA4's corresponding INTP4 input function 0x01: PB4's corresponding INTP4 input function 0x02: PF4's corresponding INTP4 input function Other: GPIO function	0x03

2.4.12.6 INTP5 Input Port Configuration Register (PS_INTP5)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_INTP5[1:0]	PS_INTP5 input selection 0x00: PA5's corresponding INTP5 input function 0x01: PB5's corresponding INTP5 input function Other: GPIO function	0x03

2.4.12.7 INTP6 Input Port Configuration Register (PS_INTP6)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_INTP6[1:0]	PS_INTP6 input selection 0x00: PA6's corresponding INTP6 input function 0x01: PB6's corresponding INTP6 input function Other: GPIO function	0x03

2.4.12.8 INTP7 Input Port Configuration Register (PS_INTP7)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_INTP7[1:0]	PS_INTP7 input selection 0x00: PA7's corresponding INTP7 input function 0x01: PB7's corresponding INTP7 input function Other: GPIO function	0x03

2.4.12.9 INTP8 Input Port Configuration Register (PS_INTP8)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_INTP8[1:0]	PS_INTP8 input selection 0x00: PA8's corresponding INTP8 input function 0x01: PB8's corresponding INTP8 input function Other: GPIO function	0x03

2.4.12.10 IIC_SCL Port Configuration Register (PS_SCL)

Bit	Symbol	Description	Reset value
7:3	--	Reserved	--
2:0	PS_SCL [2:0]	IIC communication SCL port selection 0x00: PA3's corresponding IIC SCL function 0x01: PA8's corresponding IIC SCL function 0x02: PA9's corresponding IIC SCL function 0x03: PA10's corresponding IIC SCL function 0x04: PA11's corresponding IIC SCL function 0x05: PB6's corresponding IIC SCL function 0x06: PF1's corresponding IIC SCL function Other: GPIO function	0x7

2.4.12.11 IIC_SDA Port Configuration Register (PS_SDA)

Bit	Symbol	Description	Reset value
7:4	--	Reserved	--
3:0	PS_SDA [3:0]	IIC communication SDA port selection 0x00: PA2's corresponding IIC SDA function 0x01: PA7's corresponding IIC SDA function 0x02: PA9's corresponding IIC SDA function 0x03: PA10's corresponding IIC SDA function 0x04: PA12's corresponding IIC SDA function 0x05: PB7's corresponding IIC SDA function 0x06: PB8's corresponding IIC SDA function 0x07: PF0's corresponding IIC SDA function Other: GPIO function	0xF

2.4.12.12 IIC_SMBA Port Configuration Register (PS_SMBA)

Bit	Symbol	Description	Reset value
7:1	--	Reserved	--
0	PS_SMBA	IIC communication SMBA port selection 0x00: PB5's corresponding IIC SMBA function Other: GPIO function	0x1

2.4.12.13 SPI_SCK SPI SCK Port Configuration Register (PS_SCK)

Bit	Symbol	Description	Reset value
7:3	--	Reserved	--
2:0	PS_SCK[2:0]	SPI communication SCK port selection 0x00: PA1's corresponding SPI_SCK function 0x01: PA2's corresponding SPI_SCK function 0x02: PA5's corresponding SPI_SCK function 0x03: PA9's corresponding SPI_SCK function 0x04: PB3's corresponding SPI_SCK function Other: GPIO function	0x7

2.4.12.14 SPI_MISO SPI Master Input/Slave Output Port Configuration Register (PS_MISO)

Bit	Symbol	Description	Reset value
7:3	--	Reserved	--
2:0	PS_MISO [2:0]	SPI communication MISO port selection 0x00: PA0's corresponding SPI_MISO function 0x01: PA6's corresponding SPI_MISO function 0x02: PA7's corresponding SPI_MISO function 0x03: PA11's corresponding SPI_MISO function 0x04: PA13's corresponding SPI_MISO function 0x05: PB4's corresponding SPI_MISO function Other: GPIO function	0x7

2.4.12.15 SPI_MOSI SPI Master Input/Slave Output Port Configuration Register (PS_MOSI)

Bit	Symbol	Description	Reset value
7:3	--	Reserved	--
2:0	PS_MOSI [2:0]	SPI communication MOSI port selection 0x00: PA1's corresponding SPI_MOSI function 0x01: PA2's corresponding SPI_MOSI function 0x02: PA3's corresponding SPI_MOSI function 0x03: PA7's corresponding SPI_MOSI function 0x04: PA8's corresponding SPI_MOSI function 0x05: PA12's corresponding SPI_MOSI function 0x06: PB5's corresponding SPI_MOSI function Other: GPIO function	0x7

2.4.12.16 SPI_NSS SPI Slave Chip Selection Signal Port Configuration Register (PS_NSS)

Bit	Symbol	Description	Reset value
7:3	--	Reserved	--
2:0	PS_NSS [2:0]	SPI communication NSS port selection 0x00: PA4's corresponding SPI_NSS function 0x01: PA10's corresponding SPI_NSS function 0x02: PA15's corresponding SPI_NSS function 0x03: PB0's corresponding SPI_NSS function 0x04: PF1's corresponding SPI_NSS function 0x05: PF3's corresponding SPI_NSS function Other: GPIO function	0x7

2.4.12.17 Serial Communication Unit SPI00 SCK/IIC00 SCL Port Configuration Register (PS_SCK00)

Bit	Symbol	Description	Reset value
7:3	--	Reserved	--
2:0	PS_SCK00[2:0]	SAU communication SCK/SCL port selection 0x00: PA0's corresponding SPI00 SCK, IIC00 SCL function 0x01: PB2's corresponding SPI00 SCK, IIC00 SCL function 0x02: PB8's corresponding SPI00 SCK, IIC00 SCL function 0x03: PF0's corresponding SPI00 SCK, IIC00 SCL function Other: GPIO function	0x7

2.4.12.18 Serial Communication Unit SPI00 MISI/ IIC00 SDA/ UART0 RXD Port Configuration Register (PS_SDI00)

Bit	Symbol	Description	Reset value
7:3	--	Reserved	--
2:0	PS_SDI00[2:0]	SAU communication MISI/SDA/RXD0 port selection 0x00: PA3's corresponding SPI00 MISI, IIC00 SDA, UART0 RXD function 0x01: PA9's corresponding SPI00 MISI, IIC00 SDA, UART0 RXD function 0x02: PB6's corresponding SPI00 MISI, IIC00 SDA, UART0 RXD function 0x03: PF1's corresponding SPI00 MISI, IIC00 SDA, UART0 RXD function 0x04: PF3's corresponding SPI00 MISI, IIC00 SDA, UART0 RXD function Other: GPIO function	0x7

2.4.12.19 Serial Communication Unit SPI00 MOSO/ UART0 TXD Port Configuration Register (PS_SDO00)

Bit	Symbol	Description	Reset value
7:3	--	Reserved	--
2:0	PS_SDO00[2:0]	SAU communication MOSO/TXD0 port selection 0x00: PA4's corresponding SPI00 MOSO, UART0 TXD function 0x01: PA10's corresponding SPI00 MOSO, UART0 TXD function 0x02: PB7's corresponding SPI00 MOSO, UART0 TXD function 0x03: PF2's corresponding SPI00 MOSO, UART0 TXD function Other: GPIO function	0x3

2.4.12.20 Serial Communication Unit SPI00 SS Port Configuration Register (PS_SS00)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_SS00 [1:0]	SAU communication SS00 port selection 0x00: PA8's corresponding SPI00 SS function 0x01: PB8's corresponding SPI00 SS function Other: GPIO function	0x3

2.4.12.21 USART_TXD1 USART Output Port Configuration Register (PS_TXD1)

Bit	Symbol	Description	Reset value
7:4	--	Reserved	--
3:0	PS_TXD1 [3:0]	USART communication TXD1 port selection 0x00: PA2's corresponding USART TXD1 function 0x01: PA7's corresponding USART TXD1 function 0x02: PA9's corresponding USART TXD1 function 0x03: PA10's corresponding USART TXD1 function 0x04: PA14's corresponding USART TXD1 function 0x05: PB6's corresponding USART TXD1 function 0x06: PB8's corresponding USART TXD1 function 0x07: PF1's corresponding USART TXD1 function 0x08: PF3's corresponding USART TXD1 function Other: GPIO function	0xF

2.4.12.22 USART_RXD1 USART Input Port Configuration Register (PS_RXD1)

Bit	Symbol	Description	Reset value
7:4	--	Reserved	--
3:0	PS_RXD1 [3:0]	USART communication RXD1 port selection 0x00: PA3's corresponding USART RXD1 function 0x01: PA8's corresponding USART RXD1 function 0x02: PA9's corresponding USART RXD1 function 0x03: PA10's corresponding USART RXD1 function 0x04: PA13's corresponding USART RXD1 function 0x05: PA15's corresponding USART RXD1 function 0x06: PB2's corresponding USART RXD1 function 0x07: PB7's corresponding USART RXD1 function 0x08: PF0's corresponding USART RXD1 function Other: GPIO function	0xF

2.4.12.23 USART_CTS1 USART Hardware Flow Input Port Configuration Register (PS_CTS1)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_CTS1 [1:0]	USART communication CTS1 port selection 0x00: PA0's corresponding USART CTS1 function 0x01: PA11's corresponding USART CTS1 function 0x02: PB4's corresponding USART CTS1 function Other: GPIO function	0x3

2.4.12.24 USART_RTS1 UART Hardware Flow Output Port Configuration Register (PS_RTS1)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_RTS1 [1:0]	USART communication RTS1 port selection 0x00: PA1's corresponding USART RTS1 function 0x01: PA12's corresponding USART RTS1 function 0x02: PB3's corresponding USART RTS1 function Other: GPIO function	0x3

2.4.12.25 USART_CK1 USART Synchronous Mode Clock Port Configuration Register (PS_CK1)

Bit	Symbol	Description	Reset value
7:3	--	Reserved	--
2:0	PS_CK1 [2:0]	USART communication CK1 port selection 0x00: PA4's corresponding USART CK1 function 0x01: PA6's corresponding USART CK1 function 0x02: PA8's corresponding USART CK1 function 0x03: PB5's corresponding USART CK1 function Other: GPIO function	0x7

2.4.12.26 TIM1_CH1 Port Configuration Register (PS_T1CH1)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_T1CH1 [1:0]	TIM1 CH1 port selection 0x00: PA3's corresponding TIM1 CH1 function 0x01: PA8's corresponding TIM1 CH1 function Other: GPIO function	0x3

2.4.12.27 TIM1_CH2 Port Configuration Register (PS_T1CH2)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_T1CH2 [1:0]	TIM1 CH2 port selection 0x00: PA9's corresponding TIM1 CH2 function 0x01: PA13's corresponding TIM1 CH2 function 0x02: PB3's corresponding TIM1 CH2 function Other: GPIO function	0x3

2.4.12.28 TIM1_CH3 Port Configuration Register (PS_T1CH3)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_T1CH3 [1:0]	TIM1 CH3 port selection 0x00: PA0's corresponding TIM1 CH3 function 0x01: PA10's corresponding TIM1 CH3 function 0x02: PB6's corresponding TIM1 CH3 function Other: GPIO function	0x3

2.4.12.29 TIM1 CH4 Port Configuration Register (PS_T1CH4)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_T1CH4 [1:0]	TIM1 CH4 port selection 0x00: PA1's corresponding TIM1 CH4 function 0x01: PA11's corresponding TIM1 CH4 function Other: GPIO function	0x3

2.4.12.30 TIM1 CH1N Port Configuration Register (PS_T1CH1N)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_T1CH1N[1:0]	TIM1 CH1N port selection 0x00: PA0's corresponding TIM1 CH1N function 0x01: PA7's corresponding TIM1 CH1N function Other: GPIO function	0x3

2.4.12.31 TIM1 CH2N Port Configuration Register (PS_T1CH2N)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_T1CH2N [1:0]	TIM1 CH2N port selection 0x00: PA1's corresponding TIM1 CH2N function 0x01: PB0's corresponding TIM1 CH2N function Other: GPIO function	0x3

2.4.12.32 TIM1 BKIN Port Configuration Register (PS_T1BK)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_T1BKIN [1:0]	TIM1BKIN port selection 0x00: PA6's corresponding TIM1 BKIN function 0x01: PA9's corresponding TIM1 BKIN function Other: GPIO function	0x3

2.4.12.33 TIM3 CH1 Port Configuration Register (PS_T3CH1)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_T3CH1 [1:0]	TIM3 CH1 port selection 0x00: PA2's corresponding TIM3 CH1 function 0x01: PA6's corresponding TIM3 CH1 function 0x02: PB4's corresponding TIM3 CH1 function Other: GPIO function	0x3

2.4.12.34 TIM3 CH2 Port Configuration Register (PS_T3CH2)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_T3CH2 [1:0]	TIM3 CH2 port selection 0x00: PA5's corresponding TIM3 CH2 function 0x01: PA7's corresponding TIM3 CH2 function 0x02: PB5's corresponding TIM3 CH2 function Other: GPIO function	0x3

2.4.12.35 TIM3 CH3 Port Configuration Register (PS_T3CH3)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_T3CH3 [1:0]	TIM3 CH3 port selection 0x00: PA4's corresponding TIM3 CH3 function 0x01: PB0's corresponding TIM3 CH3 function 0x02: PF3's corresponding TIM3 CH3 function Other: GPIO function	0x3

2.4.12.36 TIM14 CH1 Port Configuration Register (PS_T14CH1)

Bit	Symbol	Description	Reset value
7:3	--	Reserved	--
2:0	PS_T14CH1 [2:0]	TIM14 CH1 port selection 0x00: PA4's corresponding TIM14 CH1 function 0x01: PA7's corresponding TIM14 CH1 function 0x02: PB1's corresponding TIM14 CH1 function 0x03: PF0's corresponding TIM14 CH1 function 0x04: PF1's corresponding TIM14 CH1 function Other: GPIO function	0x7

2.4.12.37 TIM16 CH1 Input Port Configuration Register (PS_T16CH1)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_T16CH1 [1:0]	TIM16 CH1 port selection 0x00: PA6's corresponding TIM16 CH1 function 0x01: PB8's corresponding TIM16 CH1 function Other: GPIO function	0x3

2.4.12.38 TIM17 CH1 Port Configuration Register (PS_T17CH1)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_T17CH1 [1:0]	TIM17 CH1 port selection 0x00: PA7's corresponding TIM17 CH1 function 0x01: PB8's corresponding TIM17 CH1 function Other: GPIO function	0x3

2.4.12.39 TIM17 BKIN Port Configuration Register (PS_T17BKIN)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_BKIN [1:0]	TIM17 BKIN port selection 0x00: PA10's corresponding TIM17 BKIN function 0x01: PB4's corresponding TIM17 BKIN function Other: GPIO function	0x3

2.4.12.40 Comparator 0 Output Port Configuration Register (PS_C0O)

Bit	Symbol	Description	Reset value
7:3	--	Reserved	--
2:0	PS_C0O[2:0]	Comparator 0 output port selection 0x00: PA0's corresponding C0OUT0 function 0x01: PA6's corresponding C0OUT1 function 0x02: PB0's corresponding C0OUT2 function 0x03: PA11's corresponding C0OUT3 function 0x04: PB5's corresponding C0OUT4 function Other: GPIO function	0x7

2.4.12.41 Comparator 1 Output Port Configuration Register (PS_C1O)

Bit	Symbol	Description	Reset value
7:2	--	Reserved	--
1:0	PS_C1O[1:0]	Comparator 1 output port selection 0x00: PA2's corresponding C1OUT0 function 0x01: PA7's corresponding C1OUT1 function 0x02: PA12's corresponding C1OUT2 function Other: GPIO function	0x3

2.4.13 Special Function Port RESINB Description (RSTM)

By default, the RESINB (PF2) pin is active when the product is powered on. If you want to use this pin as a GPIO, you need to disable the reset function via a register. The register description is as follows:

Bit	Symbol	Description	Reset value
7:1	--	Reserved, set to 0.	0
0	RSTM	External reset function mask for RESINB pin 0: RESINB is an external reset pin 1: RESINB is used as a GPIO pin	0

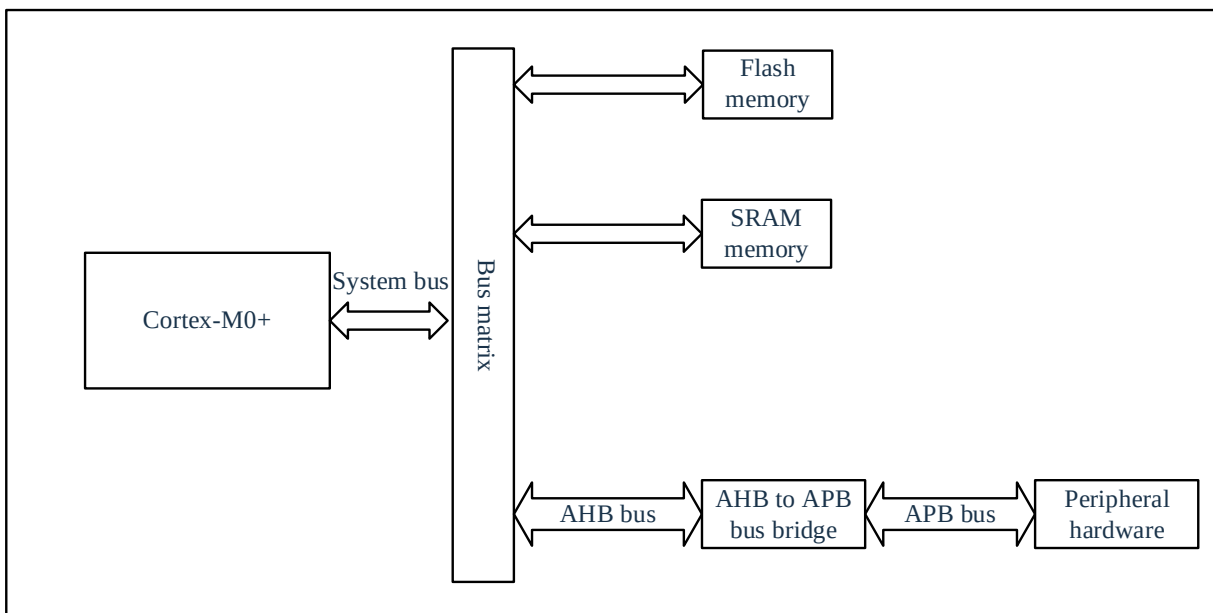
Chapter 3 System Architecture

3.1 Overview

This product system consists of the following components:

- 1 AHB bus (Master):
 - Cortex-M0+
- 3 AHB buses (Slaves):
 - Flash memory
 - SRAM memory
 - AHB to APB Bridge, contains all APB interface peripherals

Figure 3-1 System block diagram



System bus: This bus connects the system bus (peripheral bus) of the Cortex-M0+ core to the bus matrix

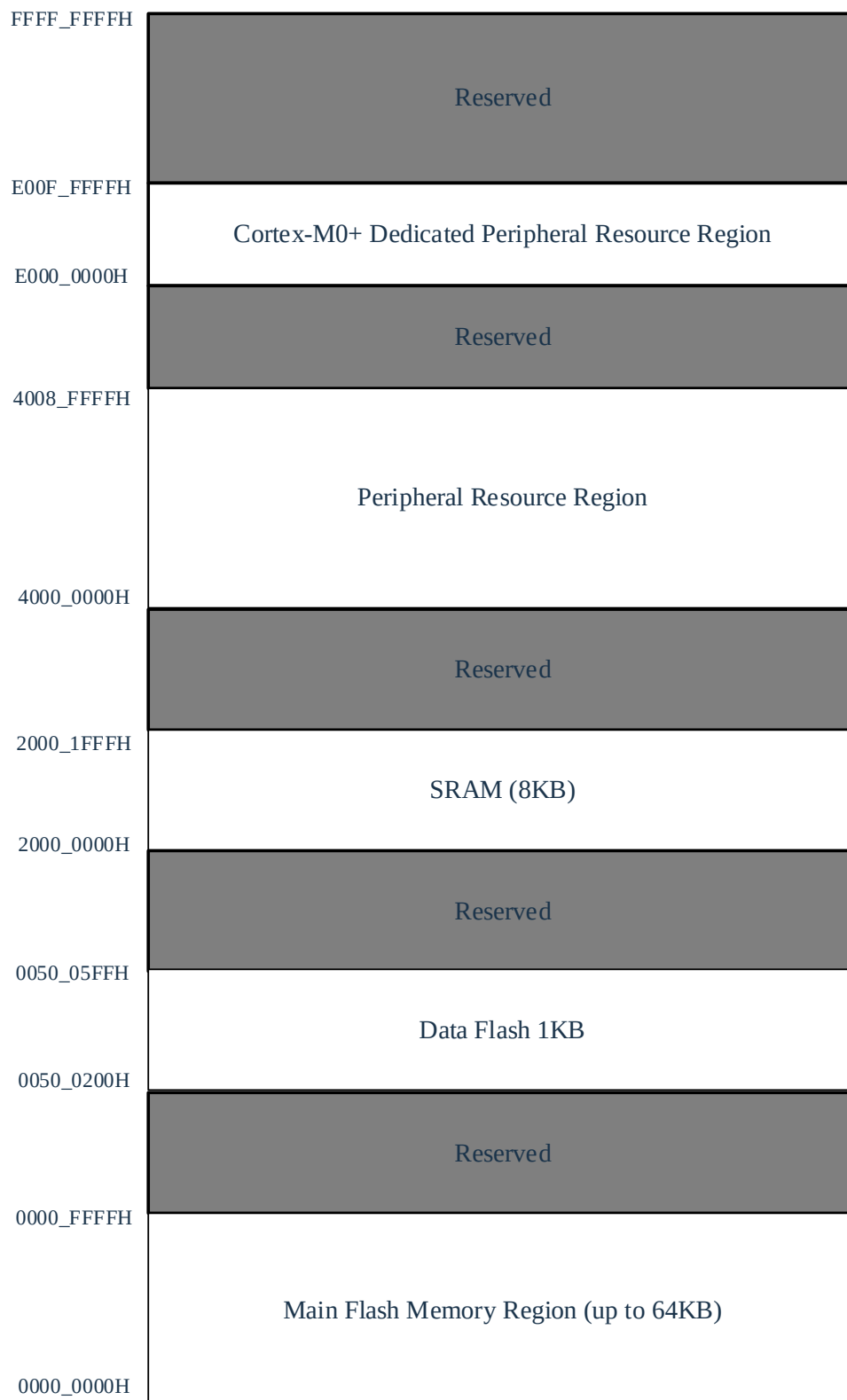
Bus matrix: The bus matrix coordinates the access of other buses to the core system bus.

AHB to APB Bridge: AHB to APB Bridge provides a synchronous connection between the AHB and APB buses.

Refer to Table 3-1 for address mapping of the different peripherals connected to each bridge.

3.2 System Address Partitioning

Figure 3-2 Map of address regions



Peripheral address allocation

Table 3-1 Peripheral register bank start address

Start address	Peripheral	Remark
0x4000_0000 - 0x4000_2FFF	Reserved	
0x4000_3000 - 0x4000_33FF	General-purpose CRC	
0x4000_5000 - 0x4000_5FFF	DMA	
0x4000_6000 - 0x4000_6FFF	Reserved	
0x4000_7000 - 0x4001_FFFF	Reserved	
0x4002_0000 - 0x4002_03FF	Flash control	
0x4002_0400 - 0x4002_0FFF	Clock control	
0x4002_1100 - 0x4002_17FF	Reserved	
0x4002_1800 - 0x4002_1BFF	High-speed CRC	See Safety Function
0x4002_1C00 - 0x4002_1FFF	Clock control	
0x4002_2000 - 0x4003_FFFF	Reserved	
0x4004_0000 - 0x4004_0F9F	GPIO	
0x4004_0FA5 - 0x4004_0FA6	Reserved	
0x4004_1500 - 0x4004_1543	Serial communications unit (SAU)	
0x4004_4B50 - 0x4004_4B50	LPTIME	
0x4004_5000 - 0x4004_53FF	AD converter	
0x4004_5400 - 0x4004_5AFF	Reserved	
0x4004_5B30 - 0x4004_5BFF	External interrupt control	
0x4004_5C00 - 0x4006_03FF	Reserved	
0x4006_0400 - 0x4006_07FF	TIM3	
0x4006_2000 - 0x4006_23FF	TIM14	
0x4006_2C00 - 0x4006_2FFF	WWDG	
0x4006_3000 - 0x4006_33FF	IWDG	
0x4006_3800 - 0x4006_3BFF	Reserved	
0x4006_4400 - 0x4006_47FF	Reserved	
0x4006_5400 - 0x4006_57FF	IIC	
0x4006_8200 - 0x4006_827F	ACMP	
0x4006_8360 - 0x4006_836F	-	-
0x4007_0000 - 0x4007_03FF	TIM1	
0x4007_1000 - 0x4007_13FF	USART	
0x4007_3000 - 0x4007_33FF	SPI	
0x4007_4400 - 0x4007_47FF	TIM16	
0x4007_4800 - 0x4007_4BFF	TIM17	
0x4006_9000 - 0x4006_901F	Reserved	
0x4008_0200 - 0xDFFF_FFFF	Reserved	

Chapter 4 Clock Generation Circuit

4.1 Functions of Clock Generation Circuit

The clock generation circuit is a circuit that generates a clock to the CPU and peripheral hardware. There are two types of system clock and clock oscillation circuits.

(1) Main system clock

High-Speed On-Chip Oscillator (high-speed OCO): The clock frequency is set to 48 MHz ($F_{HOCO} = 48 \text{ MHz}$). It can be divided down using an option byte (000C2H) to obtain a lower frequency. The clock is divided to generate the frequency F_{IH} . The CPU always begins operation using this high-speed oscillator after a reset. It is also possible to stop the oscillator by entering deep sleep mode or by setting the HIOSTOP bit (bit 0 in the CSC register). The frequency set by the option byte can be modified through the High-Speed On-Chip Oscillator Frequency Select Register (HOCODIV).

(2) Subsystem clock

XT1 oscillating circuit: By connecting a 32.768 KHz resonator to the XT1 and XT2 pins, a stable 32.768 KHz clock signal ($F_{XT} = 32.768 \text{ KHz}$) is generated. The oscillator can be stopped by setting the XTSTOP bit in the clock status control register (CSC, bit 6).

Additionally, a low-speed external clock can be provided via the EXCLKS/XT2/PA9 pin ($F_{EXS} = 32.768 \text{ KHz}$). The external clock input can be disabled by setting the XTSTOP bit.

Low-Speed On-Chip Oscillator (low-speed OCO): This on-chip oscillator operates at a typical frequency of 32 KHz ($F_{IL} = 32 \text{ KHz}$).

The low-speed oscillator can power only specific peripherals, such as:

- Independent Watchdog Timer (IWDT)
- 15-bit interval timer

Note: F_{HOCO} : High-speed on-chip oscillator clock frequency

F_{IH} : Clock frequency after the high-speed on-chip oscillator is divided

F_{IL} : Low-speed on-chip oscillator clock frequency

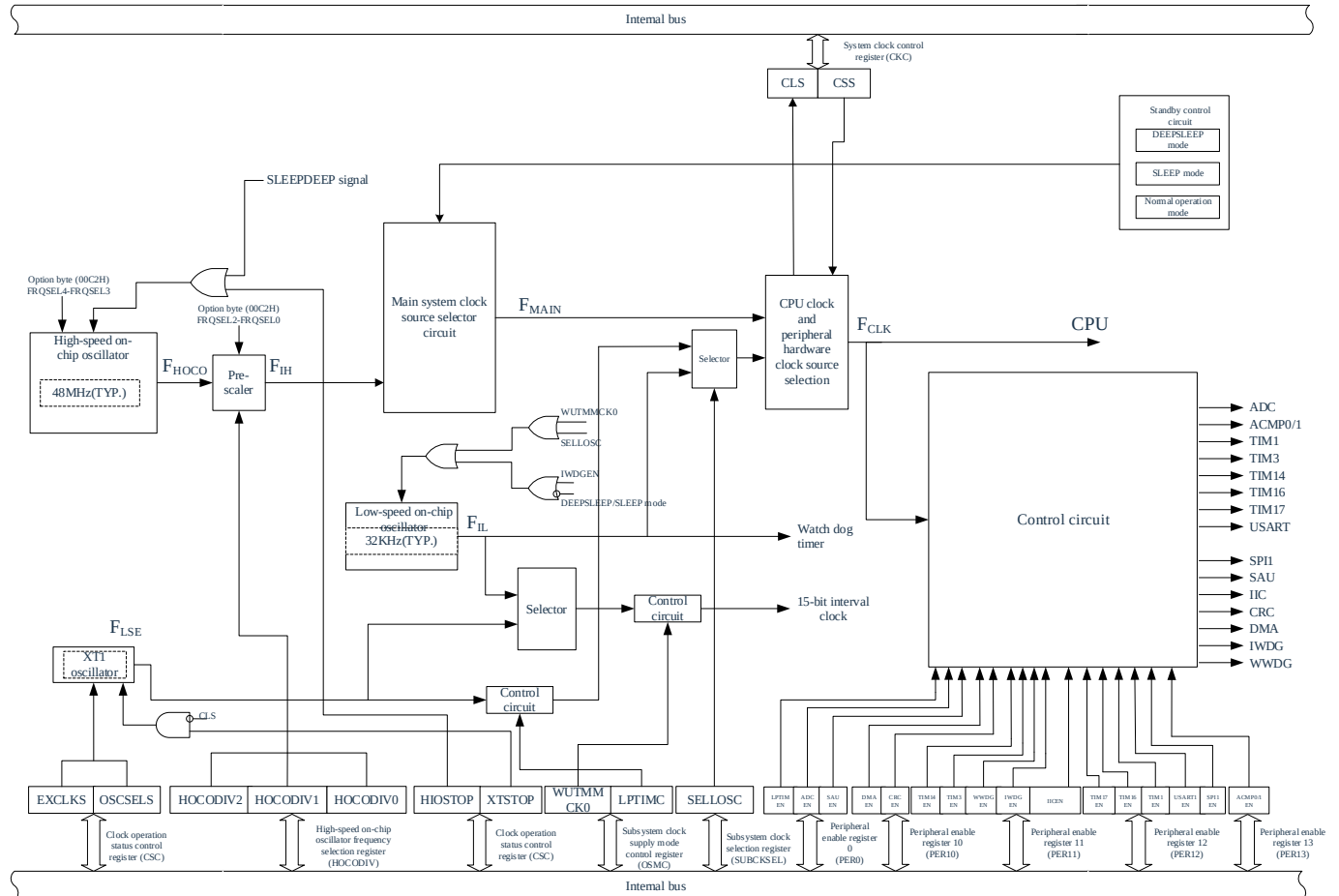
4.2 Structure of Clock Generation Circuit

The clock generation circuit is composed of the following hardware.

Table 4-1 Structure of clock generation circuit

Item	Structure
Control register	Clock operation mode control register (CMC) System clock control register (CKC) Clock operation status control register (CSC) Peripheral enable registers (PER0, PER10, PER11, PER12, PER13) High-speed on-chip oscillator frequency selection register (HOCODIV) Low-speed on-chip oscillator clock register (SUBCKSEL)
Oscillation circuit	High-speed on-chip oscillator Low-speed on-chip oscillator XT1 oscillation circuit

Figure 4-1 Block diagram of clock generation circuit



Note: F_{HOCO}: High-speed on-chip oscillator clock frequency

F_{IH}: Clock frequency after the high-speed on-chip oscillator is divided

F_{MAIN}: Main system clock frequency

F_{CLK}: CPU/peripheral hardware clock frequency

F_{IL}: Low-speed on-chip oscillator clock frequency

F_{LSE}: Low-speed external oscillator clock frequency

4.3 Register Mapping

(Register base address = 0x4002_0400) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
CMC	0x000	R/W	Clock operation mode control register	0x00
CSC	0x001	R/W	Clock operation status control register	0xC0
CKC	0x004	R/W	System clock control register	0x00
SUBCKSEL	0x007	R/W	Low-speed oscillator clock selection register	0x00
PER0	0x020	R/W	Peripheral enable register 0	0x00
OSMC	0x023	R/W	15-bit interval timer operation clock selection register	0x00

(Register base address = 0x4002_0810) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
PER10	0x000	R/W	Peripheral enable register 10	0x00
PER11	0x001	R/W	Peripheral enable register 11	0x00
PER12	0x002	R/W	Peripheral enable register 12	0x00
PER13	0x003	R/W	Peripheral enable register 13	0x00

(Register base address = 0x4002_1C00) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
HOCODIV	0x024	R/W	High-speed on-chip oscillator frequency selection register	Setting value of FRQSEL[2:0] bits of option byte (000C2H)

4.4 Register Description

4.4.1 System Clock Control Register (CKC)

This is a register that selects the CPU/peripheral hardware clock and the main system clock.

The CKC register is set by an 8-bit memory manipulation instruction.

Bit	Symbol	Description	Reset value
7	CLS ^{Note 1}	CPU/peripheral hardware clock (F _{CLK}) status 0: Main system clock (F _{MAIN}) 1: Subsystem clock (F _{SUB})	0
6	CSS	CPU/ peripheral hardware clock (F _{CLK}) selection 0: Main system clock (F _{MAIN}) 1: Subsystem clock (F _{SUB})	0
5:0	--	Reserved	0x0

Note 1: Bit 7 is a read-only bit.

Note 2: The clock provided to the CPU and peripheral hardware is set by the CSS bit. When the CPU clock is changed, the clock for the peripheral hardware is also changed (except for the 15-bit interval timer clock and the independent watchdog timer). Therefore, to change the CPU/peripheral hardware clock, the respective peripheral functions must be stopped.

4.4.2 Clock Operation Status Control Register (CSC)

This is the register that controls the operation of the high-speed on-chip oscillator clock. The CSC register is set by an 8-bit memory manipulation instruction.

After the reset signal is generated, the value of this register becomes “C0H”.

Bit	Symbol	Description	Reset value
7	--	Reserved	1
6	XTSTOP	XT1 oscillation clock operation control 0: XT1 oscillation circuit operates 1: XT1 oscillation circuit stops	1
5:1	--	Reserved	0x0
0	HIOSTOP	Operation control of high-speed on-chip oscillator clock 0: High-speed on-chip oscillator operates 1: High-speed on-chip oscillator stops	0

Caution:

After the reset is released, the CSC register must be set after configuring the Clock Operation Mode Control Register (CMC).

To start the XT1 oscillator by setting the XTSTOP bit, software must wait for the required oscillation stabilization time for the external low-speed clock.

The CSC register cannot be used to stop the clock that has been selected as the CPU/peripheral hardware clock (F_{CLK}).

For the register flag settings used to stop the clock oscillation (making external clock input invalid) and the conditions before stopping, please refer to Table 4-2.

Table 4-2 Clock stopping methods

Clock	Conditions before clock stopping (invalid external clock input)	Set CSC register flag
XT1 clock	CPU/peripheral hardware clock runs at a clock other than the subsystem clock (CLS=0)	XTSTOP=1
Low-speed external clock		
High-speed on-chip oscillator clock	CPU/peripheral hardware clock operates at a clock other than the high-speed on-chip oscillator clock (CLS=0 and MCS=1, or CLS=1)	HIOSTOP=1

4.4.3 Clock Operation Mode Control Register (CMC)

This is the register that sets the operating mode for the XT1 and XT2 pins.

After the reset is released, the CMC register can only be written once through an 8-bit memory manipulation instruction. This register can be read using an 8-bit memory manipulation instruction.

After the reset signal is generated, the value of this register becomes “00H”.

Bit	Symbol	Description	Reset value
7:4	--	Reserved	0
3:2	EXCLKS: OSCSELS	Selection of oscillation modes for XT1 oscillation circuit 00: XT1 and XT2 pins are used in port mode 01: XT1 Oscillator Mode: XT1/XT2 pins are connected to a crystal or ceramic resonator. 10: XT1 and XT2 pins are used in port mode 11: XT1 and XT2 pins are used in port mode: External clock input mode, where the XT2 pin is used for external clock input.	0
1	--	Reserved	0x0
0	LSEDRV	XT1 drive selection 0: Low drive mode 1: High drive mode	0

Note: The EXCLKS and OSCSELS bits are initialized only during power-on reset, and remain unchanged during other resets.

Caution: After the reset is released, the CMC register can only be written once through an 8-bit memory manipulation instruction. When the CMC register is used with its initial value (“00H”), in order to prevent incorrect actions when the program is out of control (if a value other than “00H” is written, it cannot be restored), the CMC register must be set to “00H” after the reset is released.

4.4.4 Peripheral Enable Registers (PER0, PER10, PER11, PER12, and PER13)

This is the register that sets whether to allow or prohibit providing clocks to various peripheral hardware. By stopping the clock to unused hardware, power consumption and noise can be reduced. When using peripheral functions controlled by these registers, the corresponding bit must be set to “1” before performing the initial setup of the peripheral function. Register function summary:

PER0	15-bit interval timer, ADC, SAU
PER10	DMA, CRC
PER11	TIM14, TIM3, WWDG, IWDG, IIC
PER12	TIM17, TIM16, TIM1, USART, SPI1
PER13	ACMP

The PER0 register and the PER10, PER11, PER12, and PER13 registers are set through 8-bit memory manipulation instructions.

After the reset signal is generated, the values of these registers become “00H”.

Peripheral Enable Register 0 (PER0)

Bit	Symbol	Description	Reset value
7	LPTIMEN	Control of real-time clock (RTC) and 15-bit interval timer input clock supply 0: Stop input clock supply, the SFR used by the real-time clock (RTC) and 15-bit interval timer cannot be written. 1: Enable input clock supply, the SFR used by the real-time clock (RTC) and 15-bit interval timer can be read and written.	0
6	-	Reserved	0
5	ADCEN	Control of A/D converter input clock supply 0: Stop to supply the input clock, ADC cannot run. 1: Supply the input clock, ADC can run.	0
4	-	Reserved	0
3	SAUEN	Control of serial communication unit input clock supply 0: Stop to supply the input clock, the SFR used by SAU cannot be written. 1: Supply the input clock, the SFR used by the SAU can be read and written.	0
2:0	-	Reserved	0

Peripheral Enable Register 10 (PER10)

Bit	Symbol	Description	Reset value
7:4	-	Reserved	0
3	DMAEN	Control of DMA input clock supply 0: Stop to supply the input clock, DMA cannot run. 1: Supply the input clock, DMA can run.	0
2	CRCEN	Control of CRC input clock supply 0: Stop to supply the input clock, CRC cannot run. 1: Supply the input clock, CRC can run.	0
1:0	-	Reserved	0

Peripheral Enable Register 11 (PER11)

Bit	Symbol	Description	Reset value
7	TIM14EN	Control of TIM14 input clock supply 0: Stop to supply the input clock, TIM14 cannot run. 1: Supply the input clock, TIM14 can run.	0
6	TIM3EN	Control of TIM3 input clock supply 0: Stop to supply the input clock, TIM3 cannot run. 1: Supply the input clock, TIM3 can run.	0
5	WWDGEN	Control of WWDG input clock supply 0: Stop to supply the input clock, WWDG cannot run. 1: Supply the input clock, WWDG can run.	0
4	IWDGEN	Control of IWDG input clock supply 0: Stop to supply the input clock, IWDG cannot run. 1: Supply the input clock, IWDG can run.	0
3:1	-	Reserved	0
0	IICEN	Control of IIC input clock supply 0: Stop to supply the input clock, IIC cannot run. 1: Supply the input clock, IIC can run.	0

Peripheral Enable Register 12 (PER12)

Bit	Symbol	Description	Reset value
7:5	--	Reserved	0
4	TIM17EN	Control of TIM17 input clock supply 0: Stop to supply the input clock, TIM17 cannot run. 1: Supply the input clock, TIM17 can run.	0
3	TIM16EN	Control of TIM16 input clock supply 0: Stop to supply the input clock, TIM16 cannot run. 1: Supply the input clock, TIM16 can run.	0
2	TIM1EN	Control of TIM1 input clock supply 0: Stop to supply the input clock, TIM1 cannot run. 1: Supply the input clock, TIM1 can run.	0
1	USART1EN	Control of USART input clock supply 0: Stop to supply the input clock, USART cannot run. 1: Supply the input clock, USART can run.	0
0	SPI1EN	Control of SPI1 input clock supply 0: Stop to supply the input clock, SPI1 cannot run. 1: Supply the input clock, SPI1 can run.	0

Peripheral Enable Register 13 (PER13)

Bit	Symbol	Description	Reset value
7:2	-	Reserved	0
1	ACMPEN	Control of ACMP input clock supply 0: Stop to supply the input clock, ACMP cannot run. 1: Supply the input clock, ACMP can run.	0
0	-	Reserved	0

4.4.5 15-Bit Interval Timer Operation Clock Selection Register (OSMC)

The OSMC register selects the clock for the 15-bit interval timer.

The OSMC register is set through an 8-bit memory manipulation instruction.

After the reset signal is generated, the value of this register becomes “00H”.

Bit	Symbol	Description	Reset value
7	LPTIMC	Settings for deep sleep mode and sleep mode with CPU running on external low-speed clock 0: Enable providing external low-speed clock to peripheral functions. 1: Stop providing external low-speed clock to peripheral functions, except for the 15-bit interval timer.	0
6:5	--	Reserved	0x0
4	WUTMMCK0	Selection of the run clock for the 15-bit interval timer: 0: Low-speed oscillator clock stops providing clock to the 15-bit interval timer. 1: Low-speed oscillator clock provides clock to the 15-bit interval timer.	0
3:0	--	Reserved	0x0

4.4.6 High-Speed On-Chip Oscillator Frequency Selection Register (HOCODIV)

This is the register that sets the frequency of the high-speed on-chip oscillator, configured by the option byte (000C2H).

The HOCODIV register is set through an 8-bit memory manipulation instruction.

After the reset signal is generated, the value of this register becomes the setting value of the FRQSEL[2:0] bits from the option byte (000C2H).

Bit	Symbol	Description	Reset value
7:3	--	Reserved	0
2:0	HOCODIV[2:0]	Selection of clock frequency for high-speed ob-chip oscillator 000: $F_{IH}=48\text{MHz}$ $F_{HOCO}=48\text{MHz}$ 001: $F_{IH}=24\text{MHz}$ $F_{HOCO}=48\text{MHz}$ 010: $F_{IH}=12\text{MHz}$ $F_{HOCO}=48\text{MHz}$ 011: $F_{IH}=6\text{MHz}$ $F_{HOCO}=48\text{MHz}$ 100: $F_{IH}=3\text{MHz}$ $F_{HOCO}=48\text{MHz}$ 101: $F_{IH}=1.5\text{MHz}$ $F_{HOCO}=48\text{MHz}$	Setting value of FRQSEL[2:0] bits of option byte (000C2H)

Note 1: The HOCODIV register must be set in a state where the high-speed on-chip oscillator clock (F_{IH}) is selected as the CPU/peripheral hardware clock (F_{CLK}).

Note 2: After changing the frequency through the HOCODIV register, the frequency switch is performed after the following transition time:

- Run up to 3 clocks at the frequency before the change.
- Wait for up to 3 CPU/peripheral hardware clocks at changed frequencies.

4.4.7 Low-speed Oscillator Clock Selection Register (SUBCKSEL)

The SUBCKSEL register is the register that selects the sub-system clock F_{SUB} .

The SUBCKSEL register is set through an 8-bit memory manipulation instruction.

After the reset signal is generated, the value of this register becomes “00H”.

Bit	Symbol	Description	Reset value
7:1	--	Reserved	0x0
0	SELLOCO	Low-speed oscillator clock selection 0: Select the low-speed external oscillator clock 1: Select the low-speed internal oscillator clock	0

4.5 System Clock Oscillation Circuit

4.5.1 XT1 Oscillation Circuit

The XT1 oscillation circuit oscillates by a crystal resonator (32.768kHz (TYP.)) connecting the XT1 pin and XT2 pin. When the XT1 oscillation circuit is used, the bit2 (OSCSELS) of the clock operation mode control register (CMC) must be set to “1” to input the external clock, and the EXCLKS pin must be input.

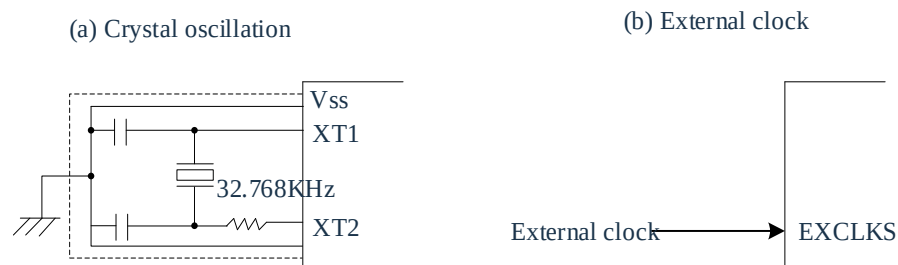
When using the XT1 oscillator circuit, the bit3 and bit2 (EXCLKS, OSCSELS) of the clock mode control register must be set:

Crystal oscillation: EXCLKS, OSCSELS=0, 1

External clock input: EXCLKS, OSCSELS=1, 1

When the XT1 oscillator circuit is not used, it must be set to the input port mode (EXCLKS, OSCSELS=0, 0). An example of an external circuit for the XT1 oscillation circuit is shown in Figure 4-2.

Figure 4-2 XT1 Example of an external circuit for XT1 oscillation circuit



Caution: When using the XT1 oscillator circuit, the dashed portions of Figure 4-2 must be routed in order to avoid the effects of wiring capacitance, etc., by the following method:

- Keep the wiring length as short as possible.
- Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flow.
- Always make the ground point of the oscillator capacitor the same potential as VSS. Do not ground the capacitor to a ground pattern through which a high current flow.
- Do not fetch signals from the oscillator.

4.5.2 High-Speed On-Chip Oscillator

The CMS32C030 has a built-in high-speed on-chip oscillator. Frequency can be selected from 48MHz, 24MHz, 12MHz, 6MHz, 3MHz, 1.5MHz and via option bytes (000C2H). The oscillation can be controlled by the bit0 (HIOSTOP) of the clock operation status control register (CSC).

After the reset is released, the high-speed on-chip oscillator automatically starts to oscillate.

4.5.3 Low-Speed On-Chip Oscillator

Low-Speed Internal Oscillator Clock is used as an external reference clock for the Independent Watchdog Timer, the 15-bit Interval Timer, and the SysTick Timer. It can also be used as the CPU clock and peripheral module clock.

When bit 4 (WUTMMCK0) of the 15-bit Interval Timer Operation Clock Selection Register (OSMC) and bit 1 (CKSEL) of the 15-bit Interval Timer Count Clock Control Register (LPTIMCL) are both set to “1”, or when bit 0 (SELLOCO) of the Sub-System Clock Selection Register (SUBCKSEL) is set to “1”, the low-speed on-chip oscillator will oscillate.

4.6 Operation of Clock Generation Circuit

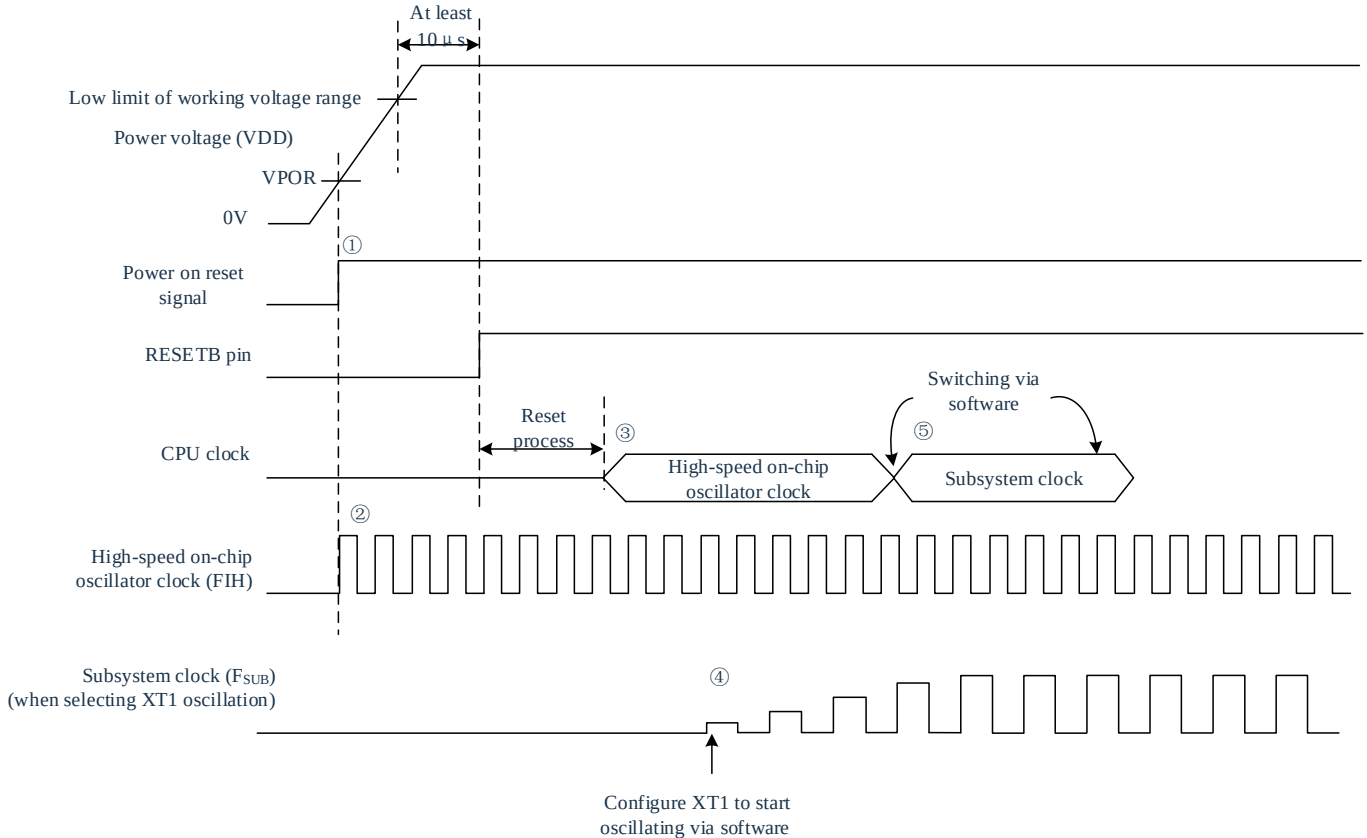
The clock generation circuit generates various clocks as shown below and controls the operation mode of the CPU such as the standby mode (refer to Figure 4-1).

F_{IH} : High-speed on-chip oscillator clock

F_{SUB} : Subsystem clock

After the CMS32C030 is released from reset, the CPU begins operation through the output of the high-speed on-chip oscillator. The operation of the clock generation circuit when the power is turned on is shown in Figure 4-3.

Figure 4-3 Operation of the clock generation circuit when the power is turned on



- (1) After power is turned on, an internal reset signal is generated through the power-on reset (POR) circuit. However, the reset state is maintained by a voltage detection circuit or an external reset until the operating voltage range shown in the AC characteristics of the datasheet is reached (the above figure shows an example when an external reset is used).
- (2) The high-speed on-chip oscillator starts oscillating automatically after the reset is released.
- (3) After the reset is released, voltage stabilization waiting and reset processing are performed, and then the CPU starts running with a high-speed on-chip oscillator clock.
- (4) The start of oscillation of the X1 clock or XT1 clock must be set by software (see “4.7.2 Example of setting XT1 oscillation circuit”).
- (5) If you want to switch the CPU clock to X1 clock or XT1 clock, you must set the switch by software after waiting for the clock oscillation to stabilize (see “4.7.2 Example of controlling XT1 oscillation clock”).

4.7 Clock Control

4.7.1 Example of High-Speed On-Chip Oscillator Set-Up

After the reset is released, the CPU/peripheral hardware clock (F_{CLK}) will definitely run on the high-speed on-chip oscillator clock. The frequency of the high-speed on-chip oscillator can be selected from 48MHz, 24MHz, 12MHz, 6MHz, 3MHz, and 1.5MHz using the FRQSEL0–FRQSEL4 bits of the option byte (000C2H). Additionally, the frequency can be changed through the high-speed on-chip oscillator frequency selection register (HOCODIV).

Option byte 000C2 settings

Bit	Symbol	Description			Reset value
7:5	--	Reserved			--
4:0	FRQSEL[4:0]	High-speed on-chip oscillator clock frequency selection			--
		FRQSEL[4:0] value	F _{HOCO}	F _{IH}	
		00000	48MHz	48MHz	
		00001	48MHz	24MHz	
		00010	48MHz	12MHz	
		00011	48MHz	6MHz	
		00100	48MHz	3MHz	
		00101	48MHz	1.5MHz	
		Other than the above	Prohibited settings.		

[HOCODIV register configuration]

Bit	Symbol	Description	Reset value
7:3	--	Reserved	0
2:0	HOCODIV[2:0]	High-speed on-chip oscillator clock frequency selection 000: $F_{IH} = 48\text{MHz}$ $F_{HOCO} = 48\text{MHz}$ 001: $F_{IH} = 24\text{MHz}$ $F_{HOCO} = 48\text{MHz}$ 010: $F_{IH} = 12\text{MHz}$ $F_{HOCO} = 48\text{MHz}$ 011: $F_{IH} = 6\text{MHz}$ $F_{HOCO} = 48\text{MHz}$ 100: $F_{IH} = 3\text{MHz}$ $F_{HOCO} = 48\text{MHz}$ 101: $F_{IH} = 1.5\text{MHz}$ $F_{HOCO} = 48\text{MHz}$	Setting value of FRQSEL[2:0] bits of option byte (000C2H)

4.7.2 Example of Setting XT1 Oscillation Circuit

After a reset release, the CPU/peripheral hardware clock (F_{CLK}) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the XT1 oscillation clock, set the oscillator and start oscillation by using the subsystem clock supply mode control register (OSMC), clock operation mode control register (CMC), and clock operation status control register (CSC), set the XT1 oscillation clock to F_{CLK} by using the system clock control register (CKC).

[Register settings] The registers must be set in the order of ① to ⑤.

- ① In deep sleep mode or sleep mode with the CPU running on an external low-speed clock, when the 15-bit interval timer is running on the external low-speed clock, the LPTIMC bit must be set to 1.

bit	7	6	5	4	3	2	1	0
OSMC	LPTIMC	0	0	WUTMMCK0	0	0	0	0

- ② Set the OSCSELS bit of the CMC register to 1 to operate the XT1 oscillator.

	7	6	5	4	3	2	1	0
CMC	0	0	0	0	EXCLKS	OSCSELS	0	LSEDRV

- ③ Clear the XTSTOP bit of the CSC register to start the XT1 oscillator circuit oscillating.

	7	6	5	4	3	2	1	0
CSC	1	XTSTOP	0	0	0	0	0	HIOSTOP

- ④ It is necessary to wait for the oscillation stabilization time required for the external low-speed clock through software and timer functions.

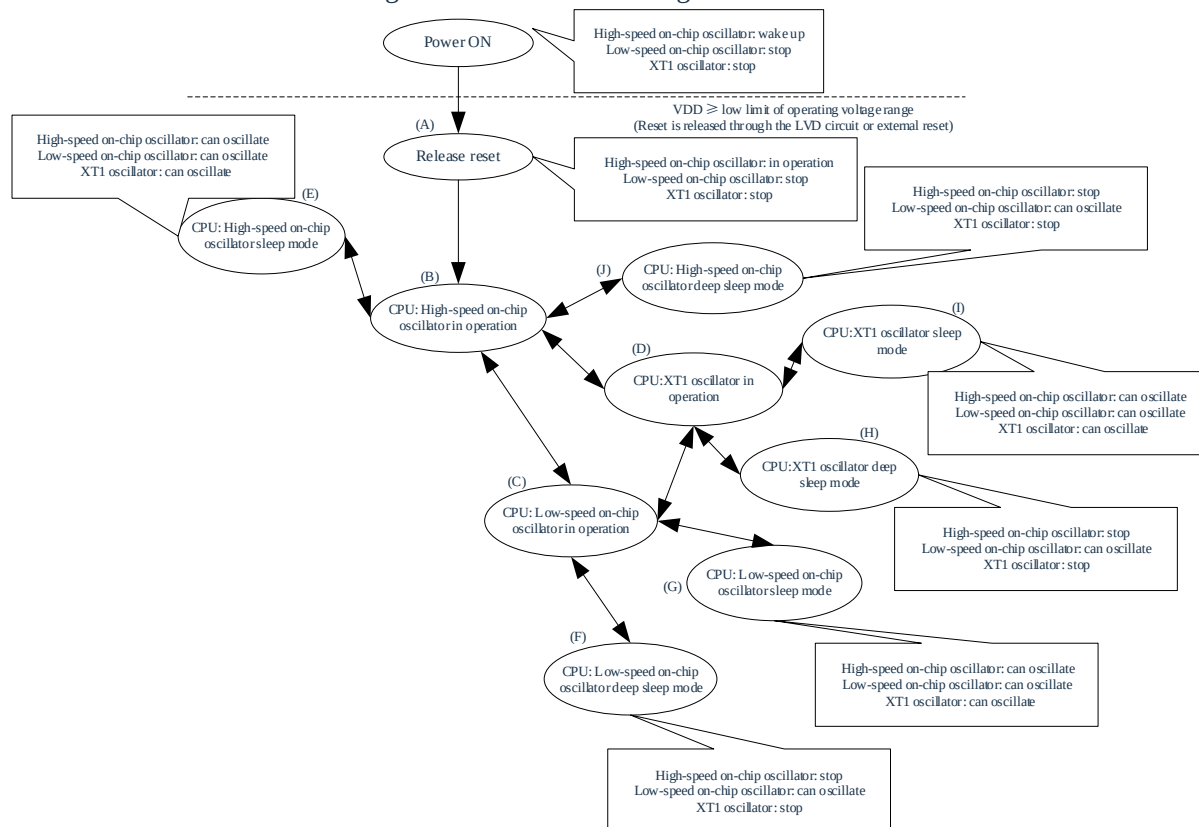
- ⑤ Set the XT1 oscillator clock as the CPU/peripheral hardware clock by using the CSS bit of the CKC register.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	0	0	0	0	0	0

4.7.3 State Transition Diagram of CPU Clock

The CPU clock state transition diagram of this product is as shown in Figure 4-4.

Figure 4-4 State transfer diagram of CPU clock



Examples of CPU clock transfer and SFR register setting are Table 4-3.

Table 4-3 Examples of CPU clock transfer and SFR register set-up (1/3)

(1) After the reset (A) is released, the CPU is shifted to the high-speed on-chip oscillator clock operation (B).

State transition	Settings for the SFR register
(A)→(B)	The SFR register (initial state after released) does not need to be set.

(2) After the reset (A) is released, the CPU is shifted to the low-speed on-chip oscillator clock operation (C).

State transition	Settings for the SFR register
(B)→(C) (D)→(C)	Order in which SFR registers are set:
	① SUBCKSEL register SELLOCO = 1
	② Configure the CKC register's CSS to 1 to select the low-speed on-chip oscillator clock as the clock for the CPU and peripheral circuits.
	③ Check the CKC register's CLS bit, and wait until it is set to 1.

(3) After the reset (A) is released, the CPU is shifted to the low-speed external oscillator clock operation (D).

State transition	Settings for the SFR register
(B)→(D) (C)→(D)	Order in which SFR registers are set:
	①SUBCKSEL register SELLOCO = 0
	②Configure the CKC register's CSS to 1 to select the low-speed external oscillator clock as the clock for the CPU and peripheral circuits.
	③Check the CKC register's CLS bit, and wait until it is set to 1.

(4) CPU transitions from another clock to run on the high-speed on-chip oscillator clock (B).

State transition	Settings for the SFR register
(C)→(B) (D)→(B)	Order in which SFR registers are set:
	Configure the CKC register's CSS to 0 to select the main clock as the clock for the CPU and peripheral circuits.
	Check the CKC register's CLS bit, and wait until it is set to 0.

Note 1: (A) to (J) in Table 4-3 correspond to (A) to (J) in Figure 4-4.

Note 2: The oscillation accuracy stability wait time of the high-speed on-chip oscillator clock varies due to temperature conditions and during deep sleep mode.

Table 4-3 Examples of CPU clock transfer and SFR register set-up (2/3)

(5) CPU transitions from high-speed on-chip clock operation (B) to sleep mode (E).

CPU transitions from low-speed on-chip clock operation (C) to sleep mode (G).

CPU transitions from low-speed external clock operation (D) to sleep mode (I).

State transition	Setting contents
(B)→(E) (C)→(G) (D)→(I)	Execute the WFI instruction.

Note: (A) to (J) in Table 4-3 correspond to (A) to (J) in Figure 4-4.

Table 4-3 Examples of CPU clock transfer and SFR register set-up (3/3)

(6) CPU transitions from high-speed on-chip clock operation (B) to deep sleep mode (J).

CPU transitions from low-speed on-chip clock operation (C) to deep sleep mode (F).

CPU transitions from low-speed external clock operation (D) to deep sleep mode (H).

State transition	Setting contents	
(B)→(J) (C)→(F) (D)→(H)	Stop the peripheral functions that cannot run in deep sleep mode.	Set bit 2 (SLEEPDEEP) of the SCR register to 1, and execute the WFI instruction.

Note: (A) to (J) in Table 4-3 correspond to (A) to (J) in Figure 4-4.

4.7.4 Conditions Before Changing CPU Clock and Processing After Changing CPU Clock

Conditions before changing the CPU clock and processing after changing the CPU clock are shown below.

Table 4-4 Changing CPU clock

CPU clock		Conditions before change	Processing after change
Before change	After change		
High-speed on-chip oscillator clock	Low-speed on-chip oscillator clock	Select the low-speed on-chip oscillator clock SELLOCO=1	If the high-speed on-chip oscillator is stopped (HIOSTOP=1), the operating current can be reduced.
	XT1 oscillator clock	XT1 oscillation is stable. • OSCSELS=1,EXCLKS=0,XTSTOP=0 • After elapse of oscillation stabilization time	
Low-speed on-chip oscillator clock	High-speed on-chip oscillator clock	Enable the high-speed on-chip oscillator oscillation • HIOSTOP=0 • After elapse of oscillation stabilization time	If the independent watchdog and LPTIM are not running, the low-speed on-chip clock oscillator can be turned off (SELLOCO = 0) to reduce operating current.
	XT1 oscillator clock	XT1 oscillation is stable. • OSCSELS=1,EXCLKS=0,XTSTOP=0 • After elapse of oscillation stabilization time	
XT1 oscillator clock	High-speed on-chip oscillator clock	Enable the high-speed on-chip oscillator oscillation • HIOSTOP=0 • After elapse of oscillation stabilization time	Can stop the oscillation of XT1 (XTSTOP=1)
	Low-speed on-chip oscillator clock	Select the low-speed on-chip oscillator clock SELLOCO=1	

4.7.5 Time Required to Switch CPU Clock and Main System Clock

The CPU clock can be switched by setting bit 6 (CSS) of the system clock control register (CKC) between the main system clock and the subsystem clock (i.e., high-speed on-chip oscillator clock, low-speed on-chip oscillator clock, and XT1 oscillator clock).

After rewriting the CKC register, the actual switch does not occur immediately. Instead, the system continues to run on the previous clock for several cycles after changing the CKC register (refer to Table 4-5).

Bit 7 (CLS) of the CKC register can be used to determine the CPU's running clock.

If the CPU clock is switched, the peripheral hardware clock will also be switched simultaneously.

Table 4-5 Number of clocks required

Switched clock Clock before switch	High-speed on-chip oscillator clock (F _{IH})	Low-speed on-chip oscillator clock (F _{IL})	XT1 oscillator clock
High-speed on-chip oscillator clock (F _{IH})		2.5 F _{IH} / F _{IL} clocks	2.5 F _{IH} / F _{XT} clocks
Low-speed on-chip oscillator clock (F _{IL})	2 clocks		2 clocks
XT1 oscillator clock	2 clocks	2 clocks	

Note 1: The number of clocks in Table 4-5 is the number of CPU clocks before the switch.

Note 2: The number of clocks in Table 4-5 is the number of clocks rounded to the decimal portion.

Example when switching the CPU from the main system clock to the subsystem clock (oscillation with F_{MAIN}=F_{IH}=1.5MHz, F_{LSE}= 32.768KHz)

$$2.5F_{\text{MAIN}}/F_{\text{LSE}}=2.5(1500/32.768)=144.4\approx 115 \text{ clocks}$$

4.7.6 Conditions Before Clock Oscillation Is Stopped

The register flags for stopping clock oscillation and the conditions before stopping are as follows.

Table 4-6 Condition and flag settings before clock oscillation is stopped

Clock	Condition before clock stops (invalid external clock input)	Flag setting of SFR register
High-speed on-chip oscillator clock	CLS=1 (The CPU is operating on a clock other than the high-speed on-chip oscillator clock)	HIOSTOP=1
Low-speed on-chip oscillator clock	CLS=0, WUMMCK0=0 (The CPU is operating on a clock other than the low-speed on-chip oscillator clock)	SELLOCO=0
XT1 oscillator clock	CLS=0 (The CPU is operating on a clock other than the low-speed external clock)	XTSTOP=1

Chapter 5 15-Bit Interval Timer

5.1 Functions of 15-Bit Interval Timer

An interrupt is generated at any previously specified time interval. It can be utilized for wakeup from deep sleep mode.

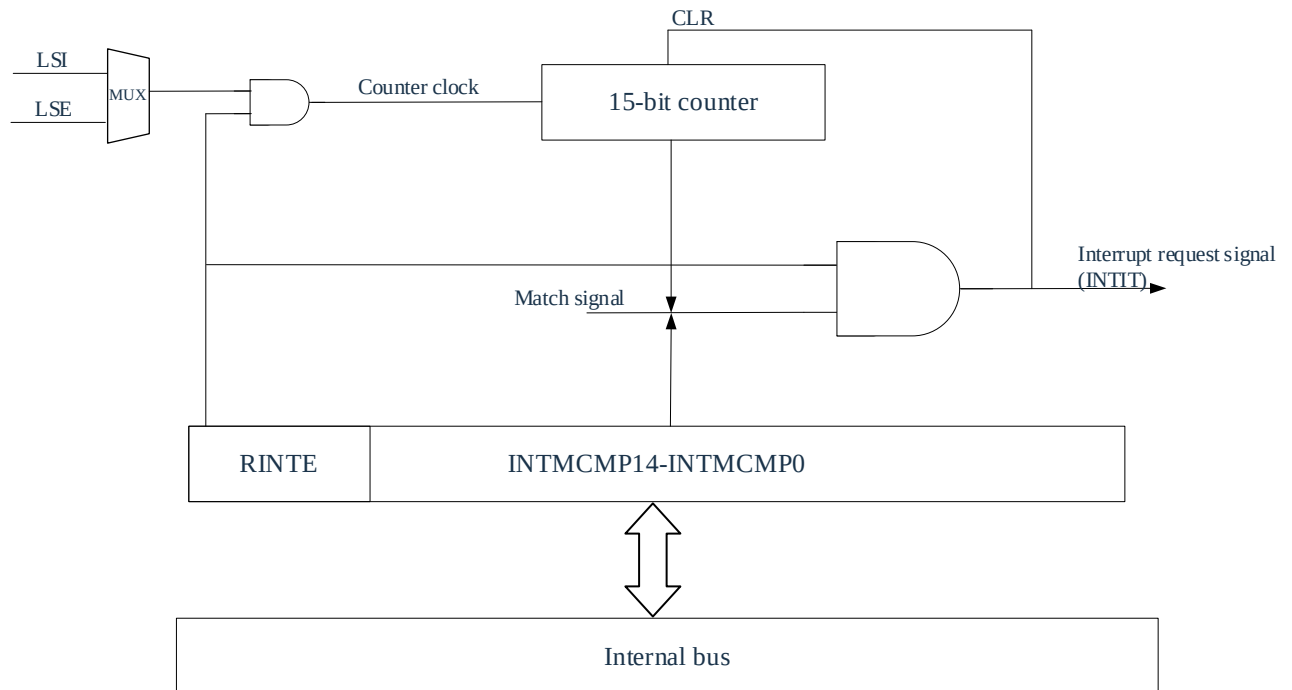
5.2 Structure of 15-Bit Interval Timer

The 15-bit interval timer consists of the following hardware.

Table 5-1 Structure of 15-bit interval timer

Item	Structure
Counter	15-bit counter
Control register	Peripheral enable register 0 (PER0)
	15-bit interval timer count clock control register (LPTIMCL)
	15-bit interval timer control register (CON)

Figure 5-1 Block diagram of 15-bit interval timer



5.3 Register Mapping

(Base address = 0x4002_0400) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
PER0	0x020	R/W	Peripheral enable register 0	0x00

(Base address = 0x4004_4B50) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
CON	0x000	R/W	15-bit interval timer control register	0x7FFF

(Base address = 0x4004_0470) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
LPTIMCL	0x00C	R/W	15-bit interval timer count clock control register	0x00

5.4 Register Description

5.4.1 Peripheral Enable Register 0 (PER0)

The PER0 register is used to enable or disable the clock supply to various peripheral hardware. By stopping the clock to unused hardware, power consumption and noise can be reduced.

To use the 15-bit interval timer, bit 7 (LPTIMEN) must be set to “1”. The PER0 register is set through an 8-bit memory manipulation instruction. After generating a reset signal, the value of this register becomes “00H”.

Bit	Symbol	Description	Reset value
7	LPTIMEN	Control of 15-bit interval timer input clock supply 0: Stop to supply the input clock, the SFR used by the 15-bit interval timer cannot be written. 1: Supply the input clock, the SFR used by the 15-bit interval timer can be read and written.	0
6	-	Reserved	0
5	ADCEN	Control of ADC input clock supply 0: Stop to supply the input clock, ADC cannot run. 1: Supply the input clock, ADC can run.	0
4	-	Reserved	0
3	SAUEN	Control of serial communication unit input clock supply 0: Stop to supply the input clock, the SFR used by the SAU cannot be written. 1: Supply the input clock, the SFR used by the SAU can be read and written.	0
2:0	-	Reserved	0

5.4.2 15-Bit Interval Timer Control Register (CON)

This register is used to set up the starting and stopping of the 15-bit interval timer operation and to specify the timer compare value. The ITMC register is set by a 16-bit memory manipulation instruction.

After a reset signal is generated, the value of this register becomes “7FFFH”.

Bit	Symbol	Description	Reset value
15	RINTE	15-bit interval timer operation control 0: Count operation stopped (count clear) 1: Count operation started	0x0
14:0	ITCMP	Specification of the 15-bit interval timer compare values	0x7FFF

ITCMP14~ITCMP0	Specification of the 15-bit interval timer compare values
0001H	These bits generate “an interrupt at the fixed cycle, count clock cycles×(ITCMP set value+1)”.
• • •	
7FFFH	
0000H	Settings are disabled.

Example interrupt cycles when 001H or FFFH is specified for ITCMP14~ITCMP0

- ITCMP14~ITCMP0=0001H, count clock: $F_{LSE}=32.768\text{KHz}$
 $1/32.768[\text{KHz}] \times (1+1) = 0.06103515625[\text{ms}] \approx 61.03[\mu\text{s}]$
- ITCMP15~ITCMP0=7FFFH, count clock: $F_{LSE}=32.768\text{KHz}$
 $1/32.768[\text{KHz}] \times (32767 + 1) = 1[\text{S}]$

Cautions:

1. To change the RINTE bit from 1 to 0, it must be rewritten after disabling interrupt handling by setting LPTIM in the interrupt mask flag register. To resume operation (from 0 to 1), interrupt handling must be enabled after clearing the interrupt flag.
2. The value read from the RINTE bit is reflected one count clock after setting the RINTE bit.
3. After transitioning from sleep mode to normal operating mode, if the CON register needs to be set and the system is to return to sleep mode, the transition to sleep mode must occur only after confirming that the value written to the CON register is reflected, or after at least one count clock has elapsed since setting the CON register.
4. To change the settings of the ITCMP14 to ITCMP0 bits, this must be done when the RINTE bit is 0.

However, the ITCMP14 to ITCMP0 bits can be changed simultaneously when changing the RINTE bit from 0 to 1 or from 1 to 0.

5.4.3 15-Bit Interval Timer Count Clock Control Register (LPTIMCL)

Bit	Symbol	Description	Reset value
7:1	--	Reserved	0
0	CKSEL	Selection of clock source for the 15-bit interval timer's counting clock: 0: Select low-speed external oscillator clock (LSE) 1: Select low-speed internal oscillator clock (LSI)	0

5.5 Operation of 15-Bit Interval Timer

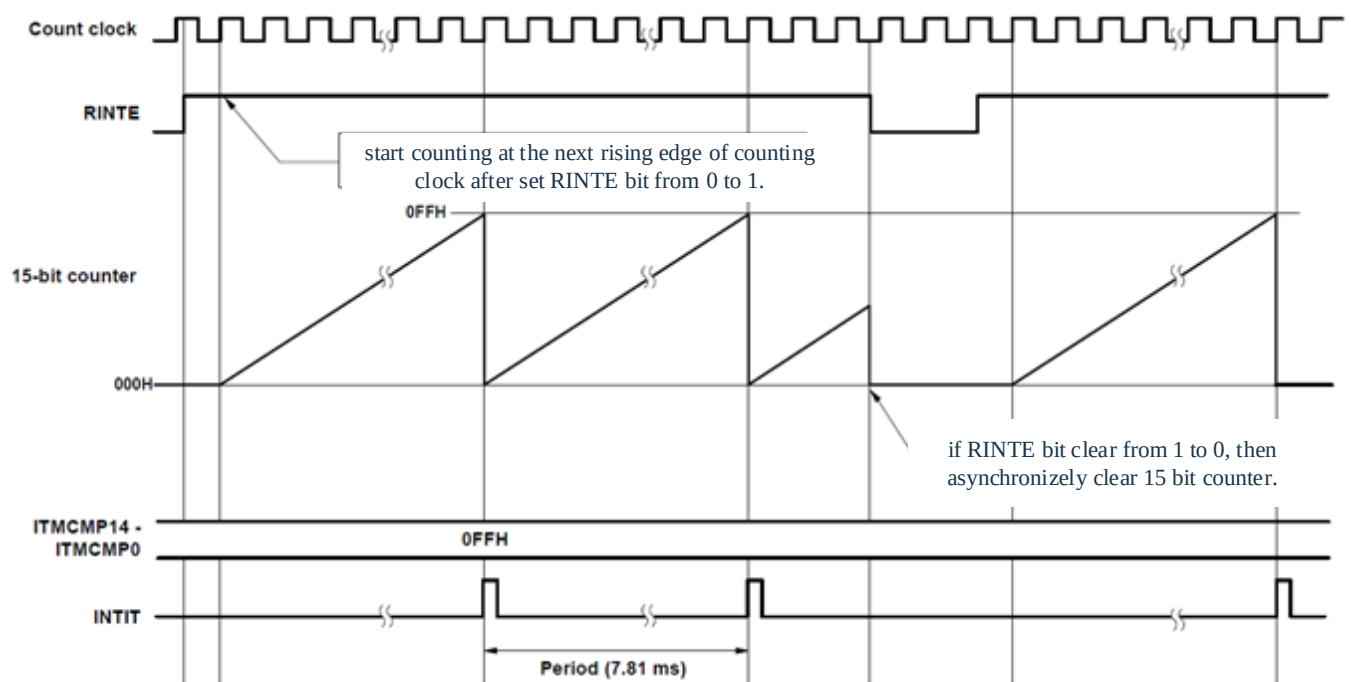
5.5.1 Operation Timing of 15-Bit Interval Timer

The 15-bit interval timer operates with the count value set by the ITCMP14 to ITCMP0 bits as the interval, generating repeated interrupt requests. If the RINTE bit is set to 1, the 15-bit counter begins counting.

When the 15-bit count value matches the value set in the ITCMP14 to ITCMP0 bits, the 15-bit count value is cleared to 0 and counting continues, simultaneously generating an interrupt request signal.

The basic operation of the 15-bit interval timer is shown in Figure 5-2.

Figure 5-2 Operation timing of 15-bit interval timer
(ITCMP14~ITCMP0=0FFH, count clock: $F_{LSE}=32.768\text{KHz}$)

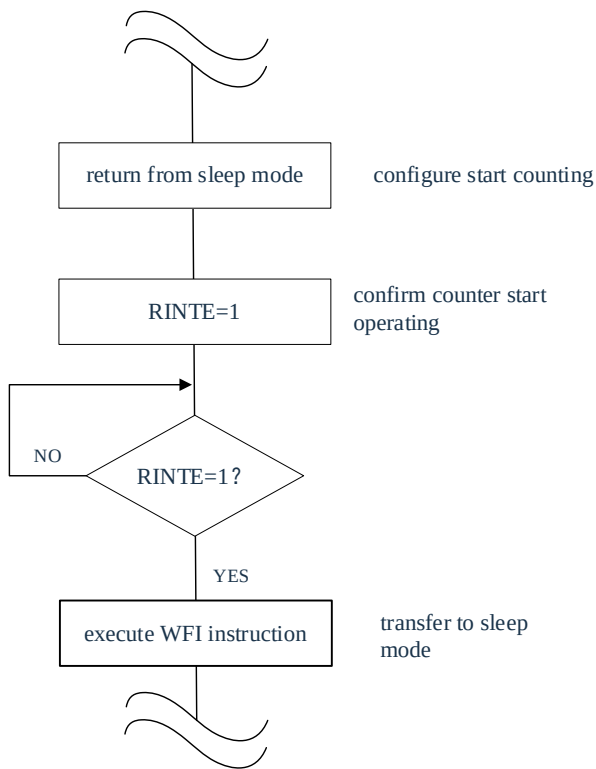


5.5.2 Start of Count Operation and Re-Enter to Sleep Mode After Returned from Sleep Mode

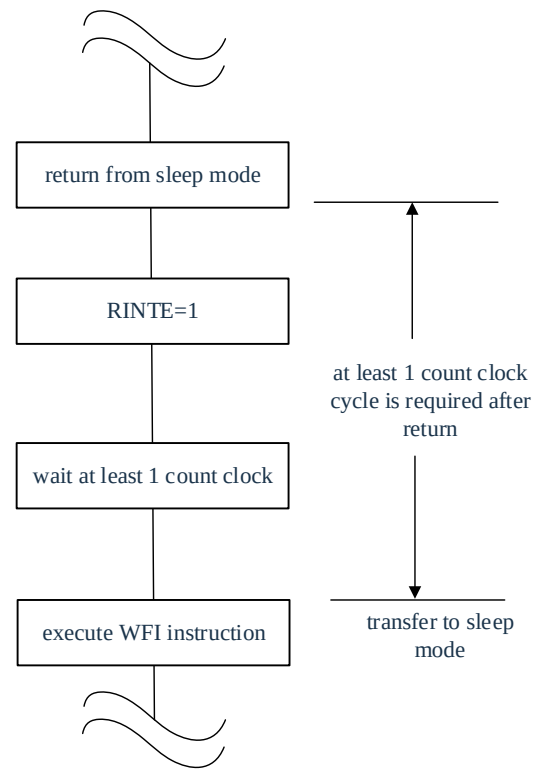
When setting the RINTE bit after returned from sleep mode and entering sleep mode again, write 1 to the RINTE bit, and confirm the written value of the RINTE bit is reflected or wait for at least one cycle of the count clock. Then, enter sleep mode.

- After setting RINTE to 1, confirm by polling that the RINTE bit has become 1, and then enter sleep mode (see Example 1 in the figure below).
- After setting RINTE to 1, wait for at least one cycle of the count clock and then enter sleep mode (see Example 2 in the figure below).

Example 1



Example 2



Chapter 6 Advanced Control Timer (TIM1)

6.1 TIM1 Introduction

The advanced control timer TIM1 includes a 16-bit auto-reload counter, which is driven by a programmable prescaler. This timer can be used for various purposes, including measuring the pulse width of input signals (input capture) or generating output waveforms (output compare, PWM, and complementary PWM with dead-time insertion).

By using the timer prescaler, pulse widths and waveform periods can be modulated from a few microseconds to a few milliseconds.

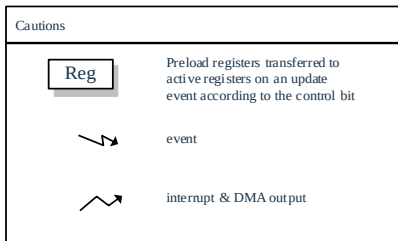
The advanced control timer (TIM1) and general-purpose (TIMx) timers are completely independent and do not share any resources. They can be synchronized.

6.2 TIM1 Main Features

The TIM1 timer has the following features:

- 16-bit up, down, and up/down auto-reload counters.
- 16-bit programmable prescaler to divide the counter clock frequency (i.e., runtime modification), with a division factor between 1 and 65536.
- Up to 4 independent channels, which can be used for:
 - Input capture
 - Output compare
 - PWM generation (edge-aligned and center-aligned modes)
 - Single pulse mode output
- Complementary output with programmable dead-time insertion.
- External signal-controlled timer with synchronization circuitry allowing multiple timers to be interconnected.
- Repetition counter, which updates the timer register after counting a specified number of periods.
- Brake input to reset the timer's output to a known state.
- Generates interrupt/DMA requests under the following events:
 - Update: counter overrun/underrun, counter initialization (via software or internal/external trigger)
 - Trigger events (counter start, stop, initialization, or count triggered by internal/external signal)
 - Input capture
 - Output compare
 - Brake input
- Supports incremental (quadrature) encoder and Hall sensor circuits.
- Trigger input for external clock or current management per cycle.

Figure 6-1 TIM1 block diagram



6.3 TIM1 Functional Description

6.3.1 Time-Base Unit

The main module of the programmable advanced control timer is a 16-bit counter and its associated auto-reload register. The counter can count up, down, or up and down in both directions. The clock for the counter can be divided by the prescaler.

The counter, auto-reload register, and prescaler register can be read or written through software. Read and write operations can be executed even while the counter is running.

The time base unit includes:

- Counter register (TIM1_CNT)
- Prescaler register (TIM1_PSC)
- Auto-reload register (TIM1_ARR)
- Repetition counter register (TIM1_RCR)

The auto-reload register is preloaded, and writing or reading the auto-reload register accesses the preloaded register. Based on the auto-reload preload enable bit (ARPE) in the TIM1_CR1 register, the contents of the preloaded register are transferred to the shadow register immediately or at every update event (UEV). When the counter reaches an overrun (or underrun condition in the case of a down counter) and when the UDIS bit in the TIM1_CR1 register equals 0, an update event is generated. Update events can also be triggered by software.

The counter is driven by the clock output CK_CNT from the prescaler, and CK_CNT is only valid when the counter enable bit (CEN) in the TIM1_CR1 register is set.

Note that after one clock cycle with the CEN bit set in the TIM1_CR register, the counter begins counting.

6.3.1.1 Prescaler Description

The prescaler can divide the counter clock by any value between 1 and 65535. It is controlled by a 16-bit counter based on a 16-bit register (in the TIM1_PSC register). Since this control register has a buffer, it can be changed during operation. The new prescaler parameters are adopted at the next update event.

Figure 6-2 Counter timing diagram with prescaler division change from 1 to 2

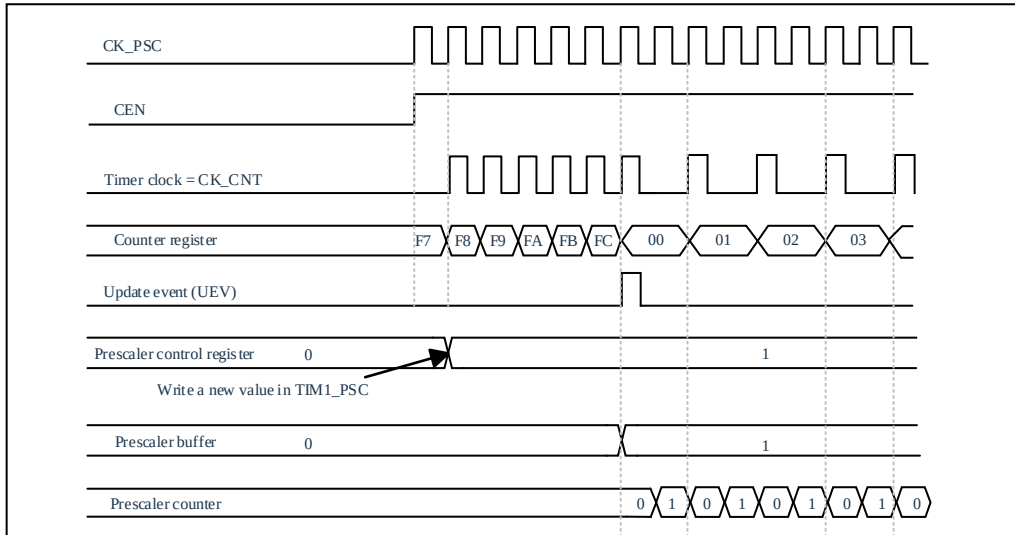
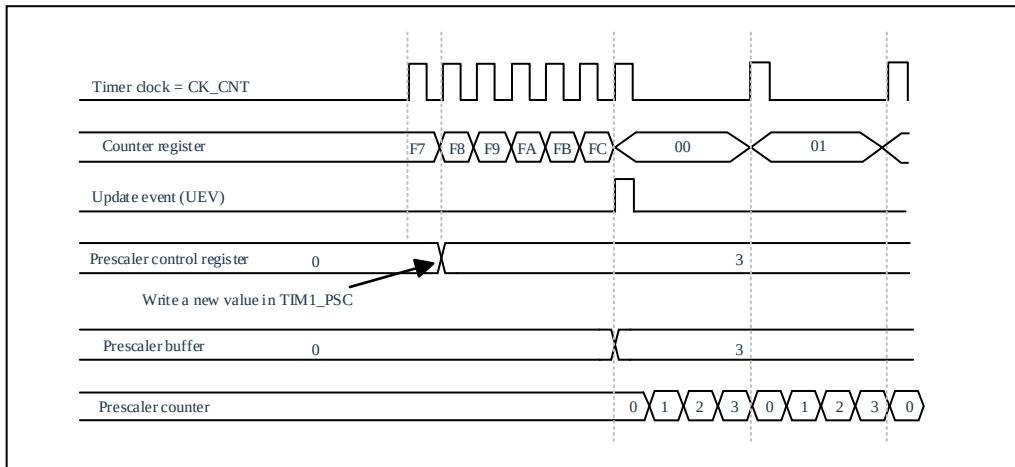


Figure 6-3 Counter timing diagram with prescaler division change from 1 to 4



6.3.2 Counter Modes

6.3.2.1 Up-Counting Mode

In up-counting mode, the counter counts from 0 to the auto-reload value and then restarts from 0, generating a counter overrun event. If the repetition counter is used, an update event is generated after the counter counts up a specified number of times (programmable for the repetition counter). Otherwise, an update event is generated on each counter overrun.

An update event can also be generated by setting the UG bit in the TIM1_EGR register (either through software or using the mode controller). Setting the UDIS bit in the TIM1_CR1 register disables update events; this also prevents the shadow register from updating when new values are written to the preload register. Update events will not be generated until the UDIS bit is cleared. Even so, when an update event should occur, the counter will still reset to 0, and the prescaler counter will also reset to 0 (but the prescaler value itself remains unchanged). Additionally, if the URS bit in the TIM1_CR1 register is set (for update requests), setting the UG bit will generate an update event (UEV), but the hardware will not set the UIF flag (meaning no interrupt or DMA request will occur). This is to prevent simultaneous update and capture interrupts when clearing the counter in capture mode.

When an update event occurs, all registers are updated, and the hardware sets the update flag (UIF bit in the TIM1_SR register) based on the URS bit.

- The repetition counter is reloaded with the value in the TIM1_RCR register.
- The auto-reload shadow register is reloaded with the value from the preload register (TIM1_ARR).
- The prescaler buffer is reloaded with the value from the preload register (TIM1_PSC).

The following figures illustrate the behavior of the counter under different clock frequencies when TIM1_ARR = 0x36.

Figure 6-4 Counter timing diagram, internal clock divided by 1

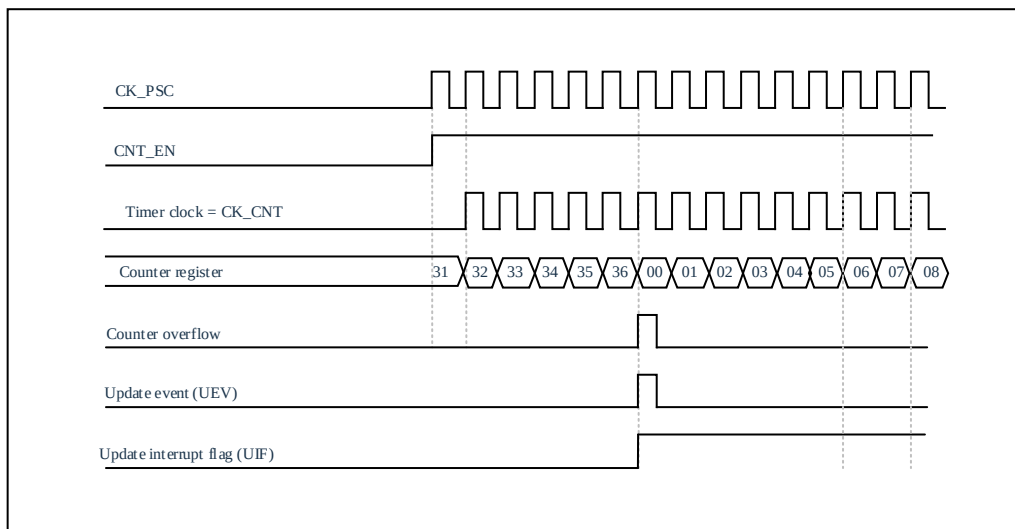


Figure 6-5 Counter timing diagram, internal clock divided by 2

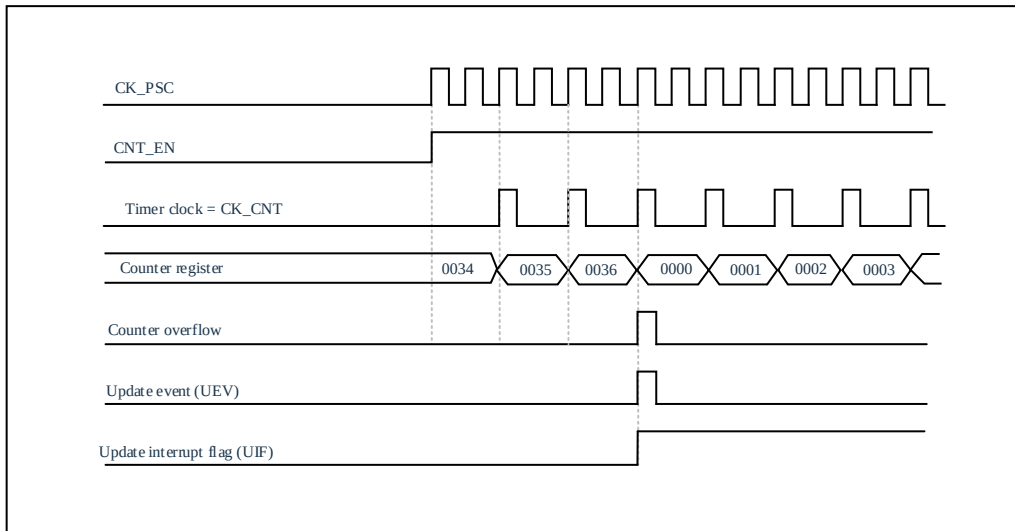


Figure 6-6 Counter timing diagram, internal clock divided by 4

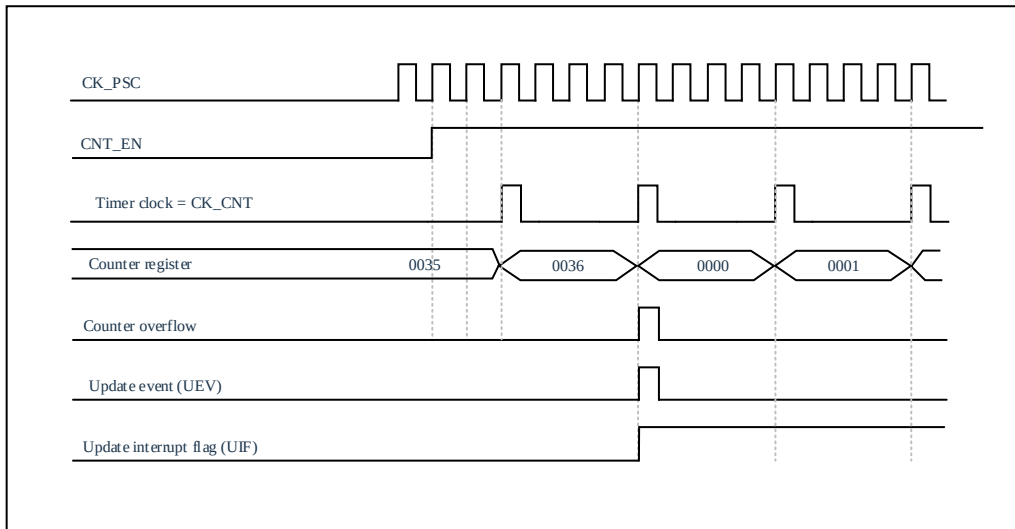


Figure 6-7 Counter timing diagram, internal clock divided by N

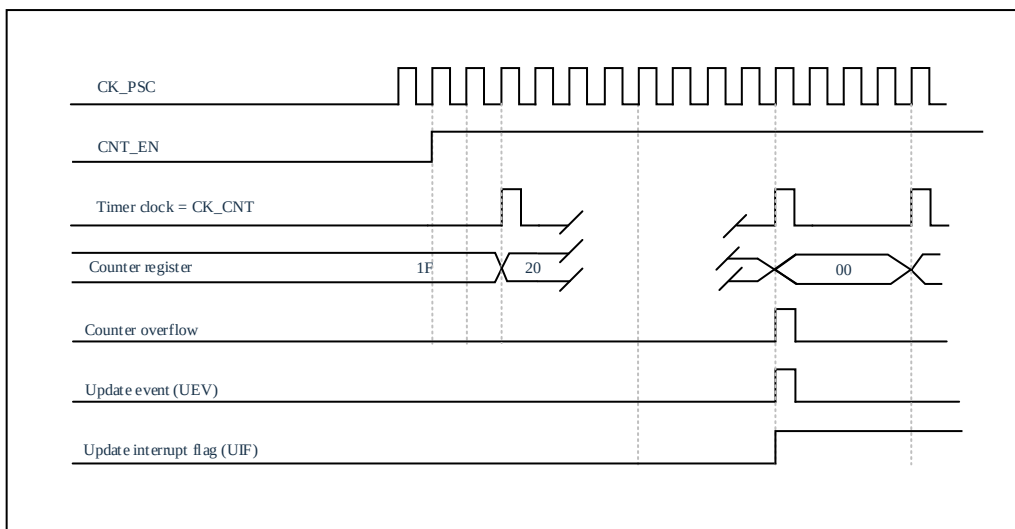


Figure 6-8 Counter timing diagram, update event when ARPE=0 (TIM1_ARR not preloaded)

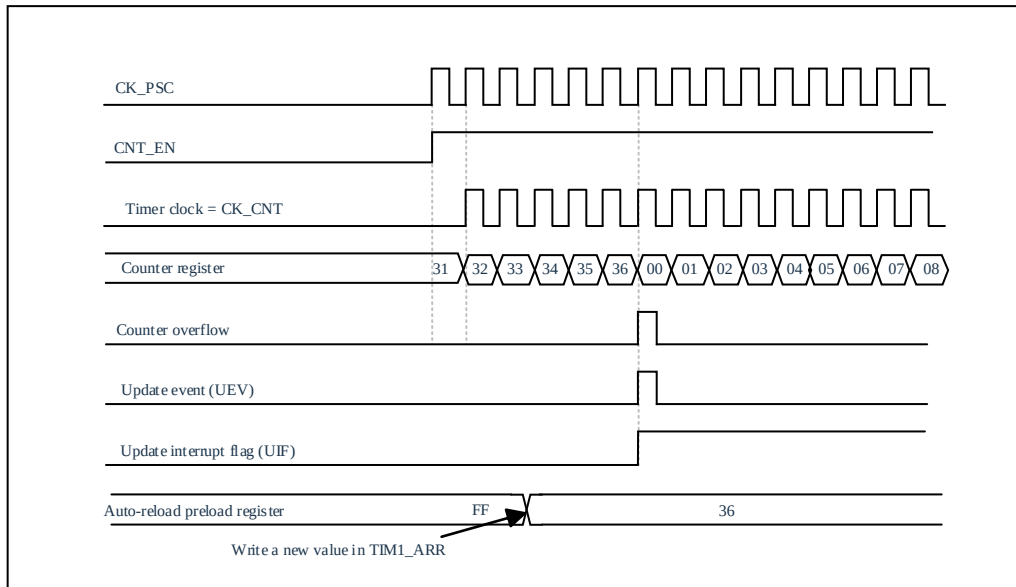
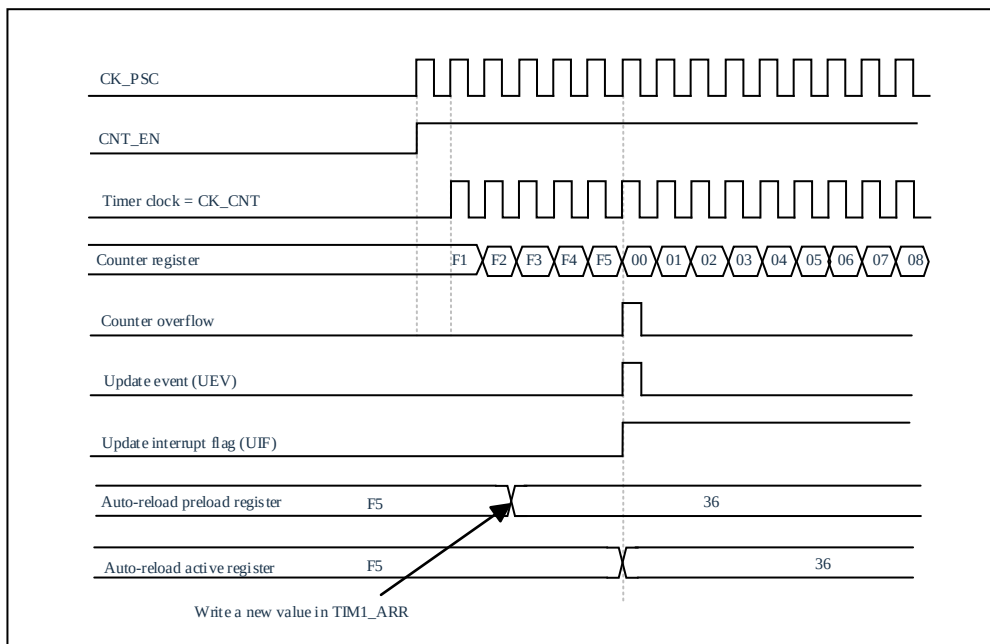


Figure 6-9 Counter timing diagram, update event when ARPE=1 (TIM1_ARR preloaded)



6.3.2.2 Down-Counting Mode

In down-counting mode, the counter starts from the auto-reload value and counts down to 0, then restarts counting down from the auto-reload value, generating a counter underrun event.

If the repetition counter is used, an update event (UEV) will be generated after the counter has counted down the number of times set in the repetition counter (TIM1_RCR), otherwise, an update event is generated only when the counter underruns.

An update event can also be generated by setting the UG bit in the TIM1_EGR register (either through software or using the mode controller).

Setting the UDIS bit in the TIM1_CR1 register can disable UEV events. This prevents the shadow register from being updated when new values are written to the preload register. As a result, no update event will occur until the UDIS bit is cleared. However, the counter will still restart counting from the current auto-reload value, and the prescaler counter will restart from 0 (but the prescaler factor remains unchanged).

Additionally, if the URS bit in the TIM1_CR1 register is set (for update requests), setting the UG bit will generate an update event (UEV) without setting the UIF flag (therefore no interrupt or DMA request will be generated). This is to avoid generating both update and capture interrupts when a capture event occurs and the counter is cleared.

When an update event occurs, all registers are updated, and the update flag (UIF bit in the TIM1_SR register) is set based on the URS bit configuration.

- The repetition counter is reloaded with the value in the TIM1_RCR register.
- The prescaler buffer is loaded with the preload value (TIM1_PSC register value).
- The current auto-reload register is updated with the preload value (TIM1_ARR register value).

Note: The auto-reload is updated before the counter is reloaded, so the next cycle will reflect the expected value.

The following figures illustrate the behavior of the counter under different clock frequencies when TIM1_ARR = 0x36.

Figure 6-10 Counter timing diagram, internal clock divided by 1

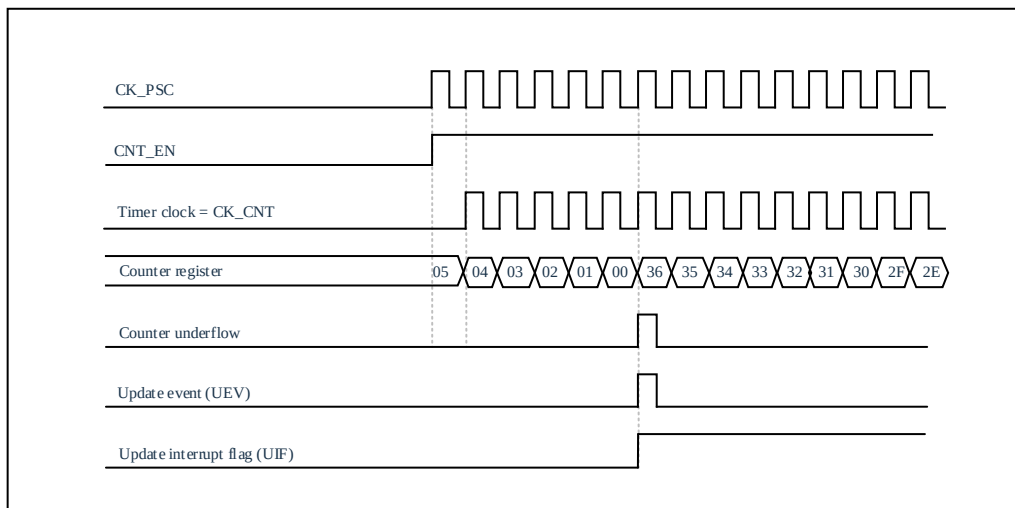


Figure 6-11 Counter timing diagram, internal clock divided by 2

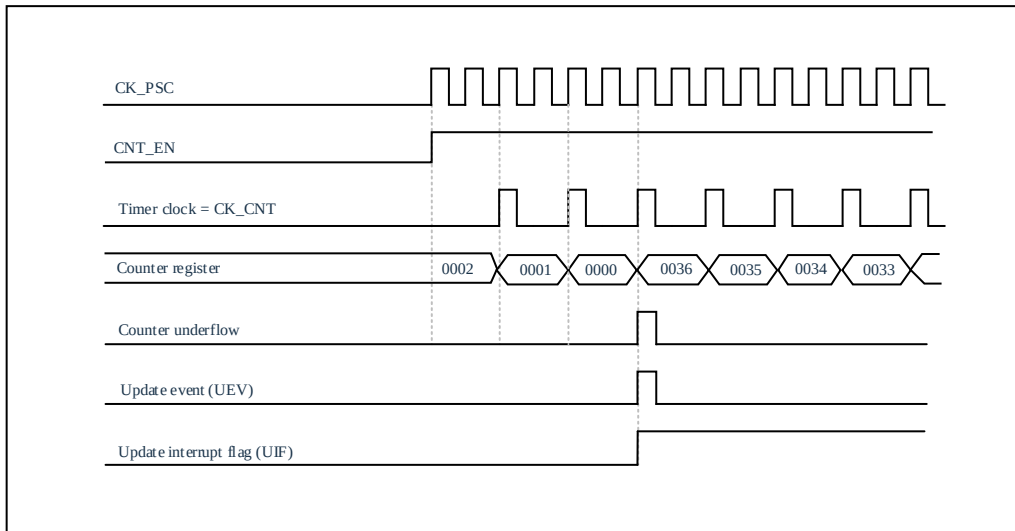


Figure 6-12 Counter timing diagram, internal clock divided by 4

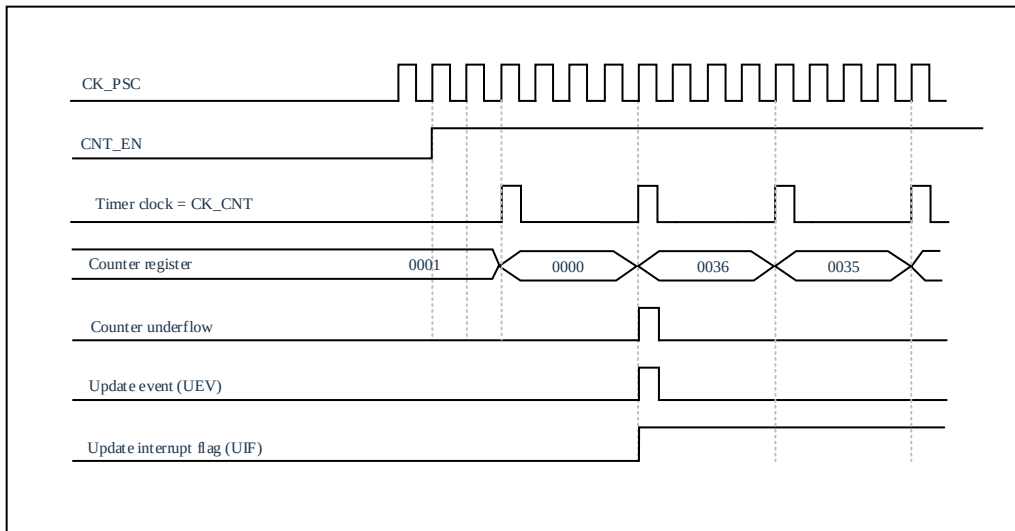


Figure 6-13 Counter timing diagram, internal clock divided by N

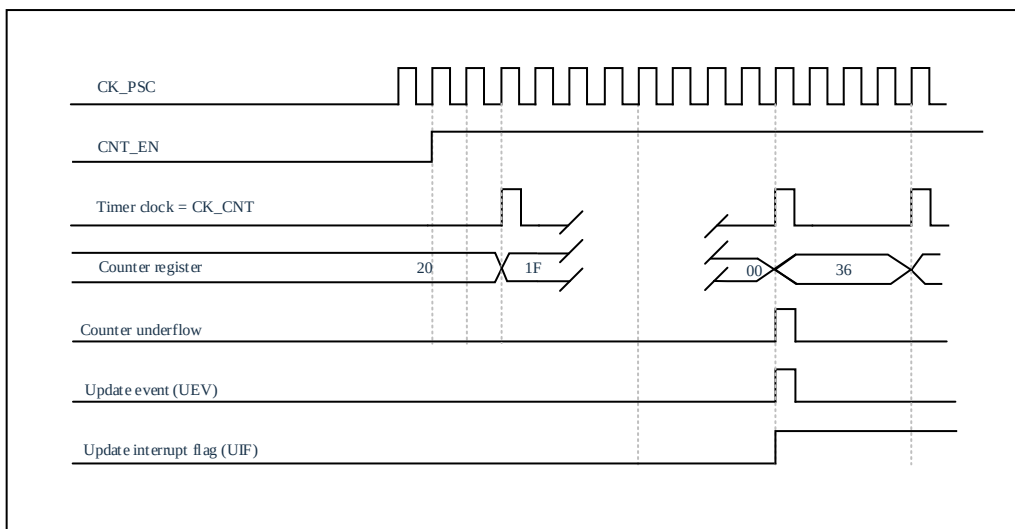
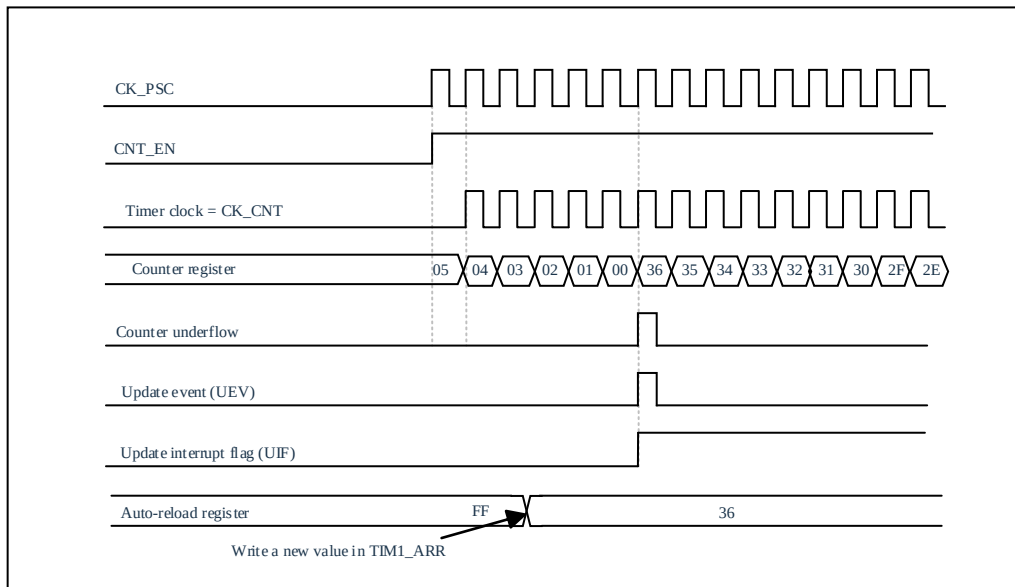


Figure 6-14 Counter timing diagram, update event when repetition counter is not used



6.3.2.3 Center-Aligned Mode (Up/Down Counting)

In center-aligned mode, the counter starts from 0, counts up to the auto-reload value (TIM1_ARR register) - 1, generating a counter overrun event, then counts down to 1 and generates a counter underrun event; after that, it restarts counting from 0.

Center-aligned mode is active when CMS in the TIM1_CR1 register is not equal to 0. When the channel is configured in output mode, the output compare interrupt flag will be set when: counting down (center-aligned mode 1, CMS = 01), counting up (center-aligned mode 2, CMS = 10), or counting up and down (center-aligned mode 3, CMS = 11).

In this mode, the DIR direction bit in the TIM1_CR1 register cannot be written. It is updated by hardware and indicates the current counting direction.

Update events can be generated on every counter overrun and every counter underrun. Additionally, an update event can be generated by setting the UG bit in the TIM1_EGR register (either via software or using the mode controller). Afterward, the counter restarts counting from 0, and the prescaler counter also restarts from 0.

Setting the UDIS bit in the TIM1_CR1 register can disable UEV events. This prevents the shadow register from being updated when new values are written to the preload register. Therefore, no update event will occur until the UDIS bit is cleared. However, the counter will continue counting up or down based on the current auto-reload value.

Furthermore, if the URS bit in the TIM1_CR1 register is set (selecting update requests), setting the UG bit will generate an update event (UEV) without setting the UIF flag (so no interrupt or DMA request will occur). This is to prevent generating both update and capture interrupts when a capture event occurs and the counter is cleared.

When an update event occurs, all registers are updated, and the update flag (UIF bit in the TIM1_SR register) is set according to the URS bit configuration.

- The repetition counter is reloaded with the value in the TIM1_RCR register.
- The prescaler buffer is loaded with the preload value (TIM1_PSC register value).
- The current auto-reload register is updated with the preload value (TIM1_ARR register value).

Note: If an update occurs due to counter overrun, the auto-reload will be updated before the counter is reloaded, so the next cycle will reflect the expected value (the counter will be loaded with the new value).

The following figures show some examples of the counter behavior for different clock frequencies.

Figure 6-15 Counter timing diagram, internal clock divided by 1, TIM1_ARR = 0x06C

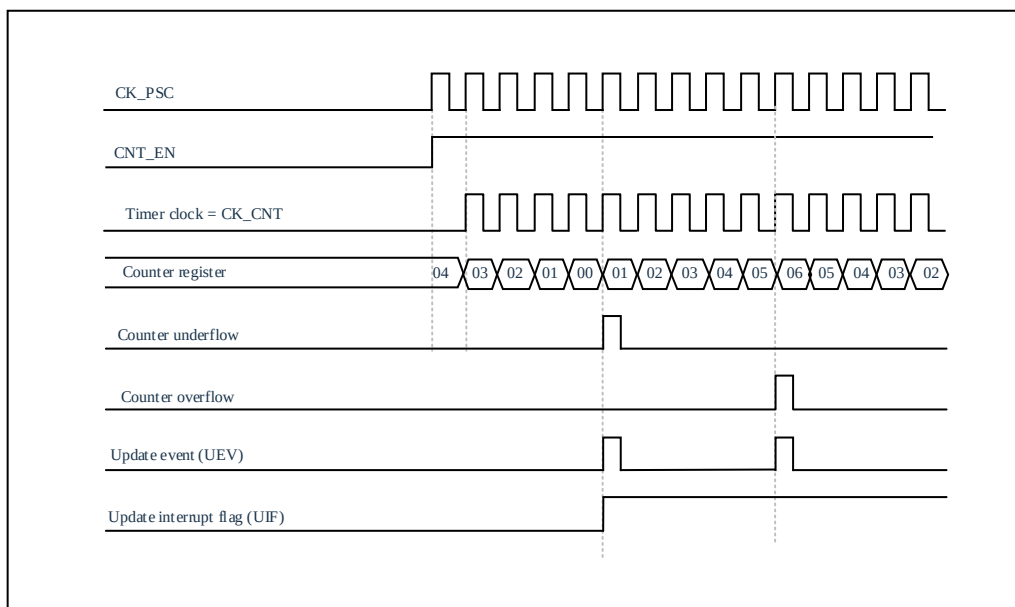


Figure 6-16 Counter timing diagram, internal clock divided by 2

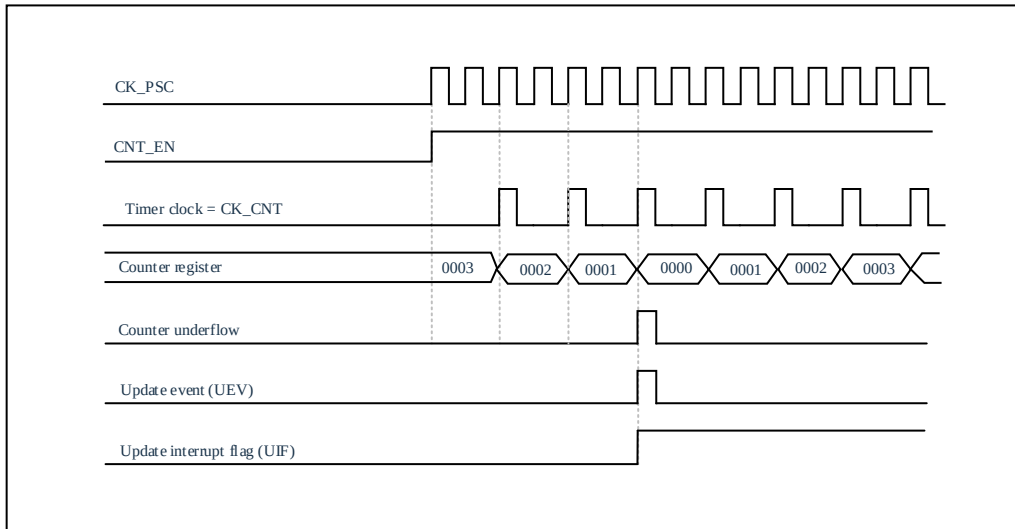


Figure 6-17 Counter timing diagram, internal clock divided by 4, TIM1_ARR=0x36

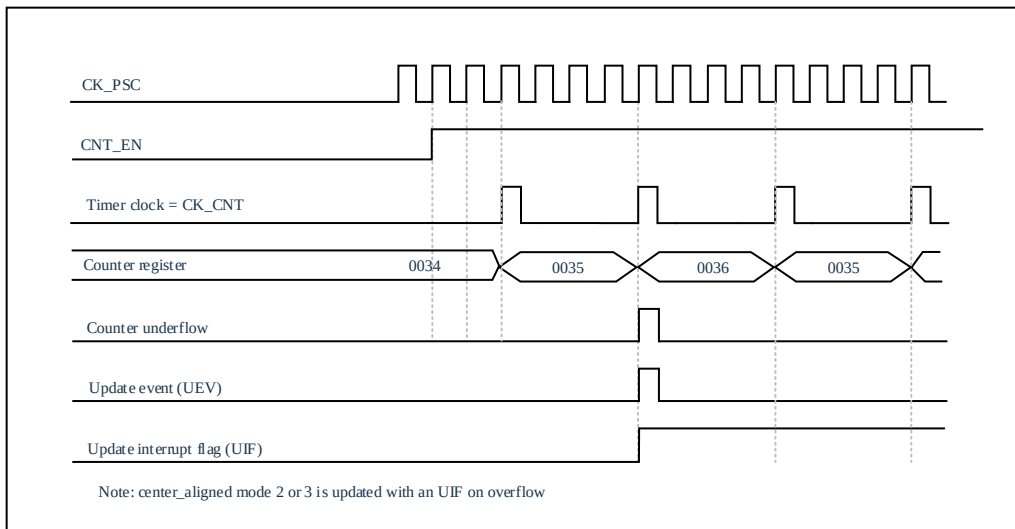


Figure 6-18 Counter timing diagram, internal clock divided by N

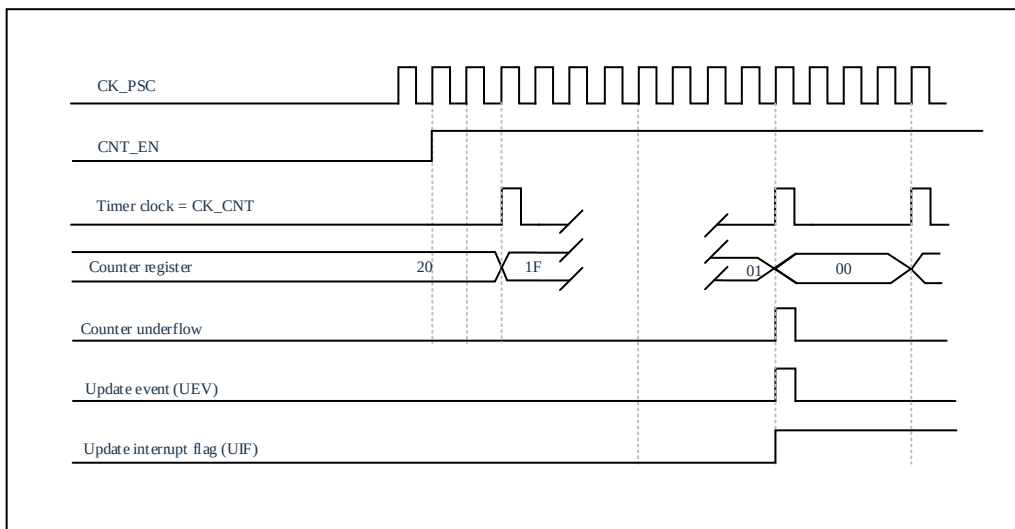


Figure 6-19 Counter timing diagram, update event with ARPE=1 (counter underrun)

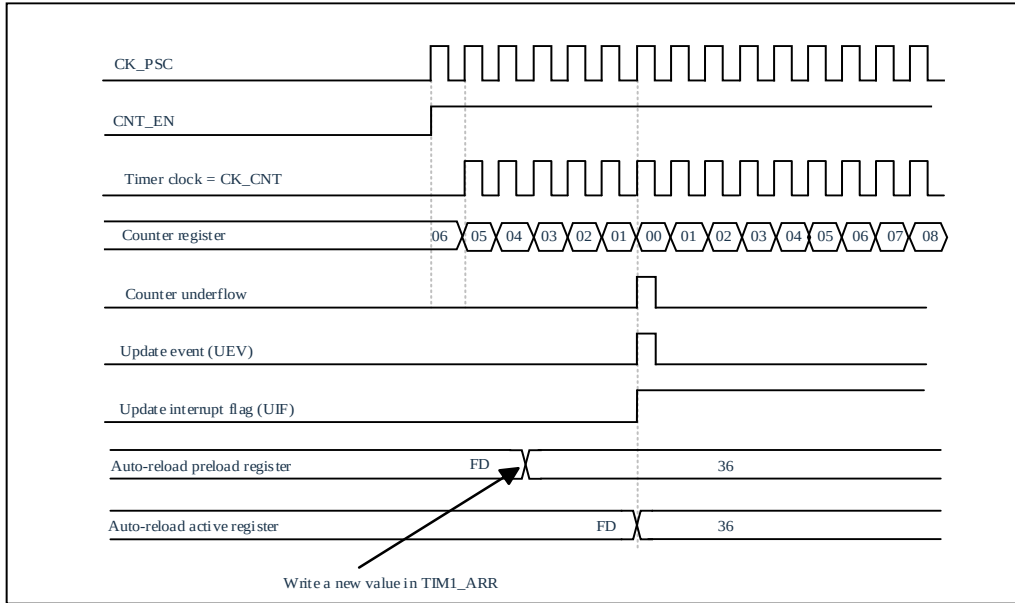
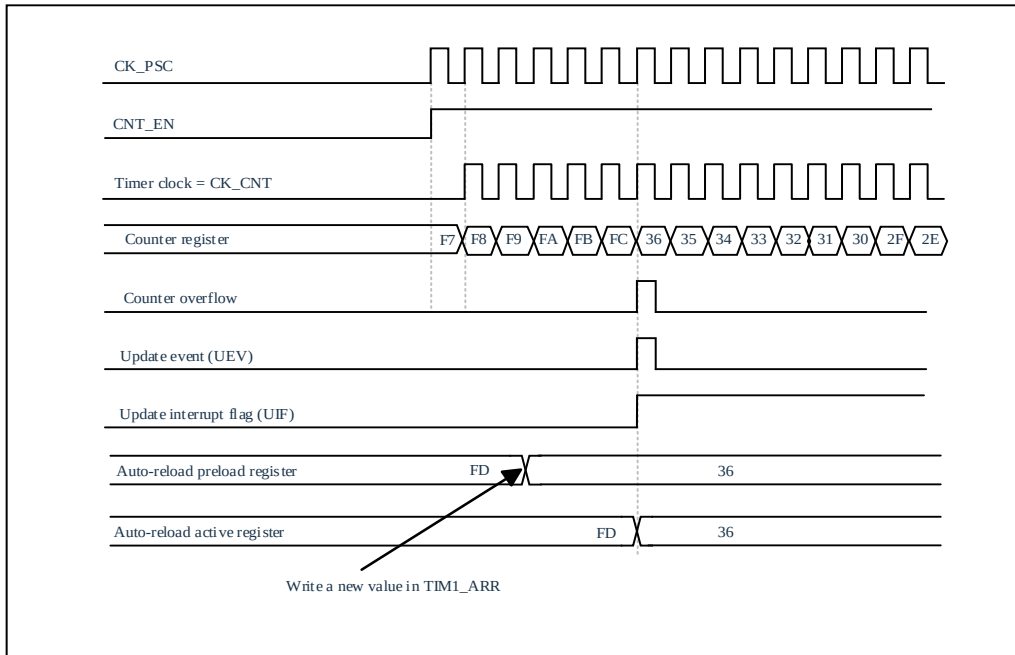


Figure 6-20 Counter timing diagram, Update event with ARPE=1 (counter overrun)



6.3.3 Repetition Counter

The time base unit describes how update events are generated when the counter overruns or underruns in the up and down counting modes. It only generates an event when the repetition counter counts down to zero. This is also useful when generating PWM signals.

This means that after every N overruns or underruns, data is transferred from the preload register to the shadow registers (TIM1_ARR auto-reload register, TIM1_PSC preload register, and the capture/compare registers TIM1_CCRx in compare mode), where N is the value in the TIM1_RCR repetition counter register.

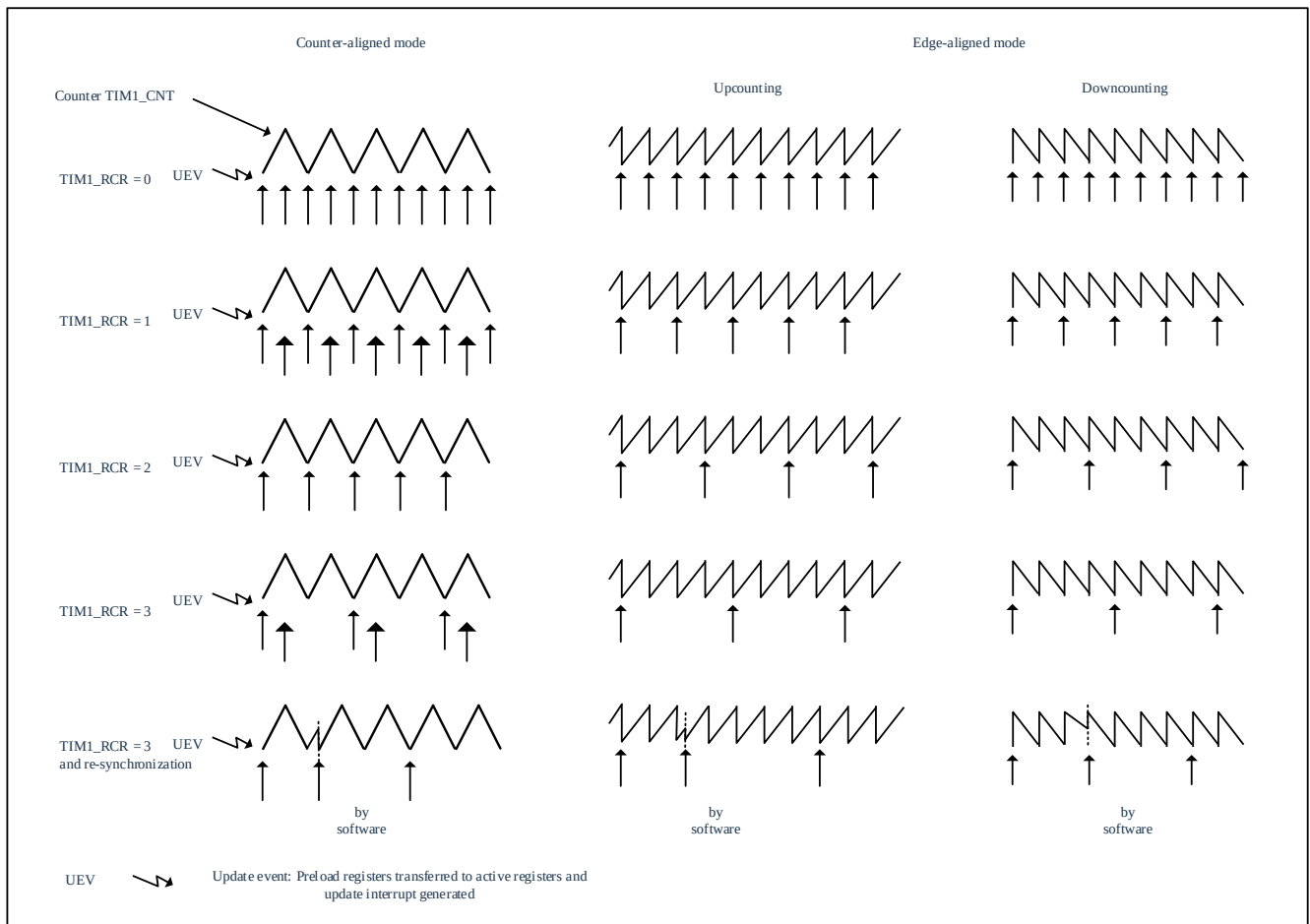
The repetition counter decrements when any of the following conditions are met:

- ◆ Each time the counter overruns in up-counting mode.
- ◆ Each time the counter underruns in down-counting mode.
- ◆ Each time there is an overrun or underrun in center-aligned mode. Although this limits the maximum PWM cycle to 128 bits, it allows the duty cycle to be updated twice in each PWM cycle. In center-aligned mode, because the waveform is symmetrical, the maximum resolution is $2 \cdot T_{ck}$ if the compare register is updated only once per PWM cycle.

The repetition counter is automatically loaded, and the repetition rate is defined by the value in the TIM1_RCR register. When an update event is generated by software (by setting the UG bit in the TIM1_EGR register) or by hardware from the mode controller, an update event occurs immediately, regardless of the value of the repetition counter, and the content of the TIM1_RCR register is reloaded into the repetition counter.

In center-aligned mode, for odd values of RCR, the update event is generated depending on when the RCR register is written and when the counter starts, whether an overrun or underrun occurs. If the RCR is written before starting the counter, the update event is generated on the overrun. For example, when $RCR = 3$, the update event is generated on the 4th overrun or underrun event (depending on when RCR is written).

Figure 6-21 Update rate examples depending on mode and TIM1_RCR register settings



6.3.4 Clock Selection

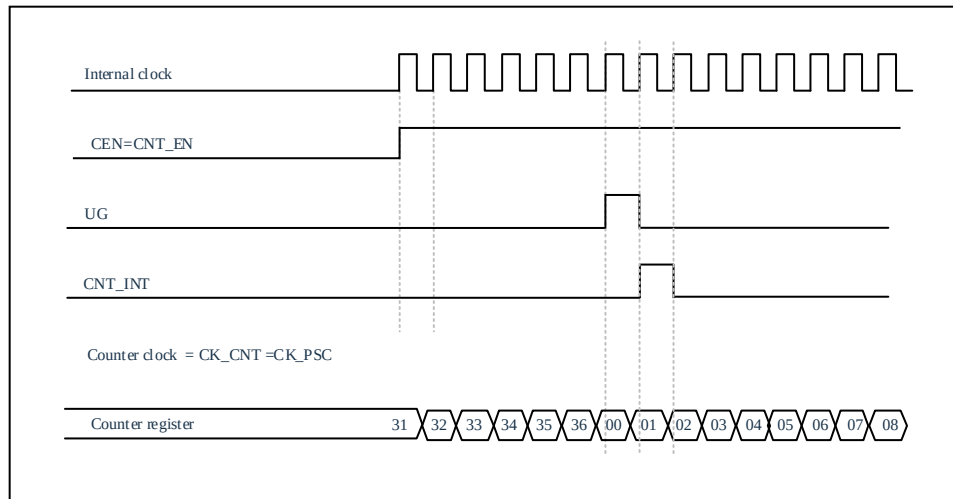
The clock for the counter can be provided by the following clock sources:

- ◆ Internal clock (CK_INT)
- ◆ External clock mode 1: External input pin
- ◆ External clock mode 2: External trigger input (ETR)
- ◆ Internal trigger input (ITRx): Using one timer as a prescaler for another timer. For example, TIM1 can be configured as a prescaler for another timer, TIM3.

6.3.4.1 Internal Clock Source (CK_INT)

If the mode controller is disabled, the CEN, DIR (TIM1_CR1 register), and UG bits (TIM1_EGR register) are effectively control bits and can only be modified by software. As long as the CEN bit is set to 1, the clock for the prescaler is provided by the internal clock CK_INT.

Figure 6-22 Control circuit in normal mode, internal clock divided by 1



6.3.4.2 External Clock Source Mode 1

This mode is selected when SMS=111 in the TIM1_SMCR register. The counter can count at each rising or falling edge on a selected input.

Figure 6-23 TI2 external clock connection example

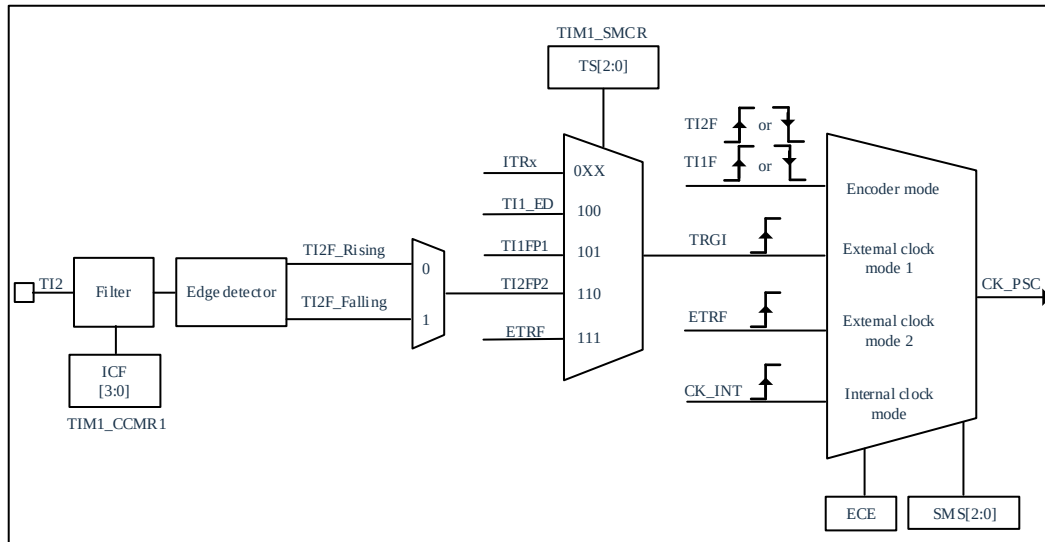
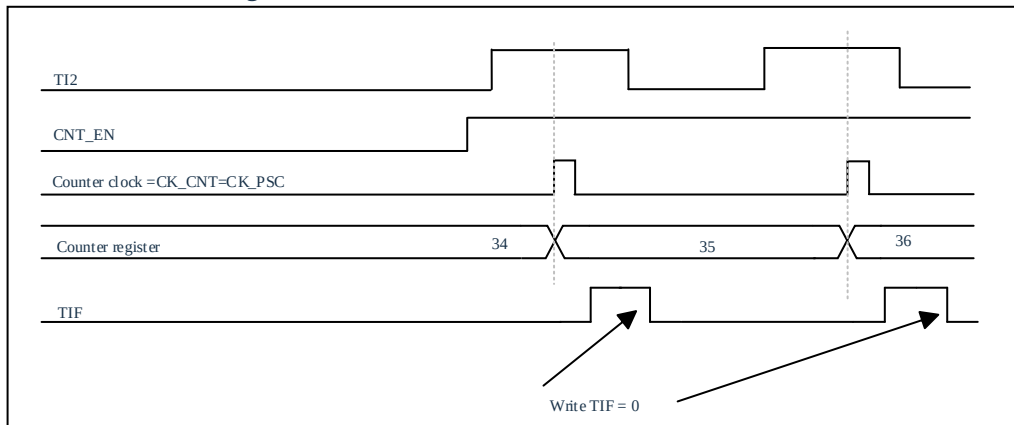


Figure 6-24 Control circuit in external clock mode 1



6.3.4.3 External Clock Source Mode 2

By setting the ECE bit of the TIM1_SMCR register to 1, this mode is selected. The counter is able to count on every rising or falling edge of the external trigger ETR.

Figure 6-25 External trigger input block

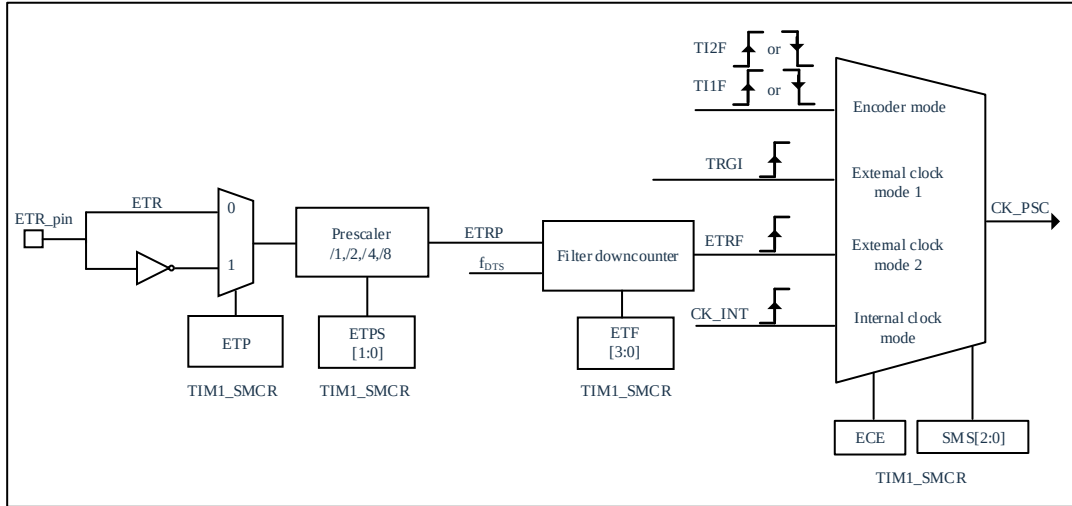
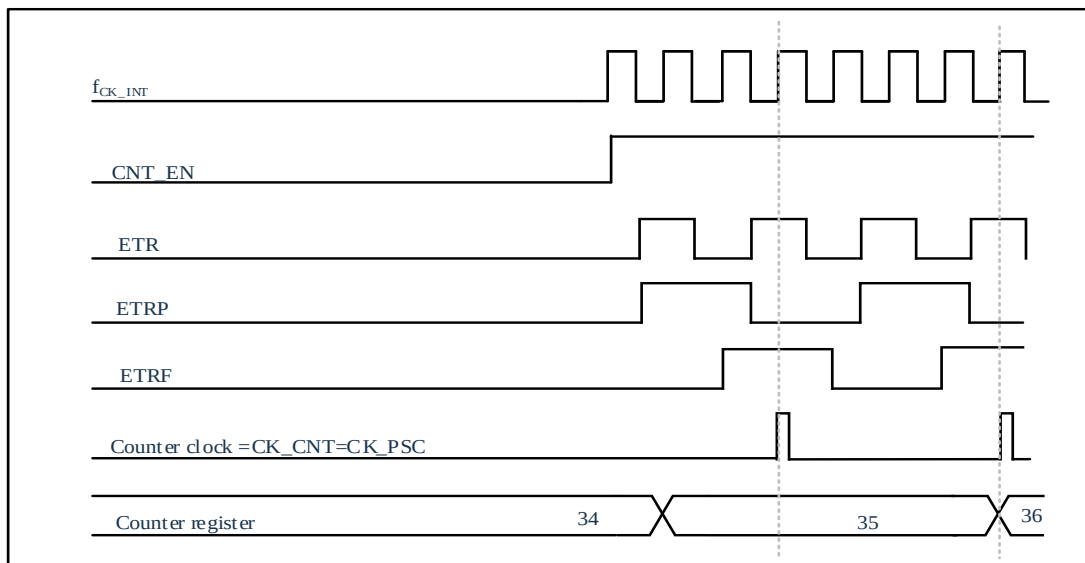


Figure 6-26 Control circuit in external clock mode 2

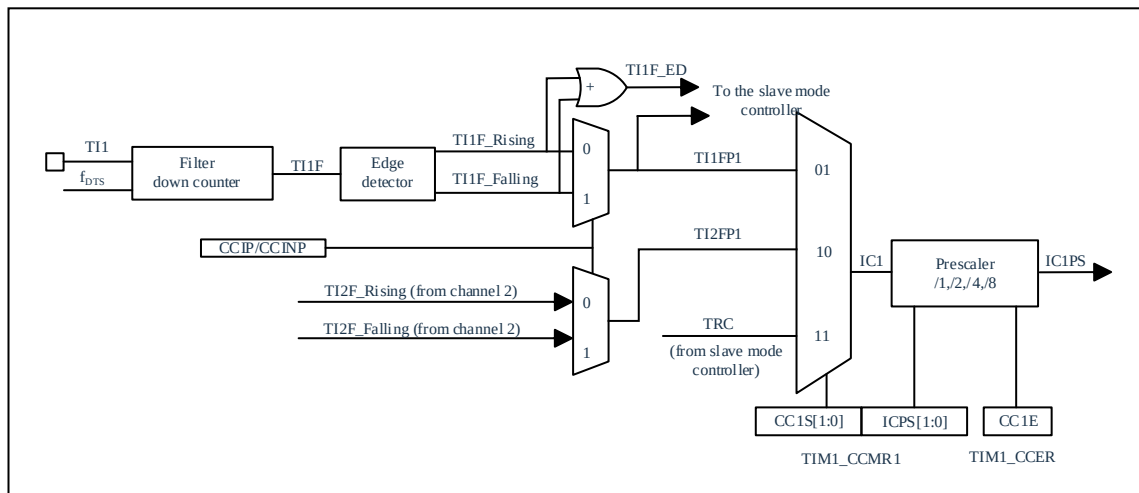


6.3.5 Capture/Compare Channels

Each capture/compare channel is built around a capture/compare register (which includes a shadow register), and consists of an input section (input filtering, multiplexing, and prescaler) and an output section (comparator and output control).

The input section samples the corresponding TIX input signal and generates a filtered signal (TIXF). Then, an edge detector with polarity selection generates a signal (TIXFPx), which can be used as a trigger input to the mode controller or as a capture control. This signal is then prescaled and enters the capture register (ICxPS).

Figure 6-27: Capture/compare channel (example: channel 1 input stage)



The output stage generates an intermediate waveform which is then used for reference: OCxREF (active high). The polarity acts at the end of the chain.

Figure 6-28 Capture/compare channel 1 main circuit

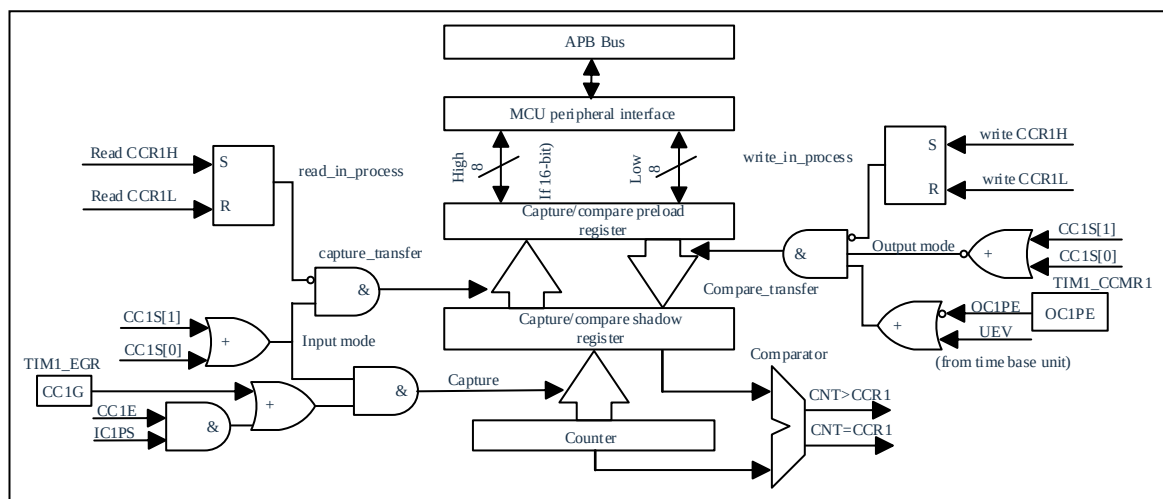


Figure 6-29: Output stage of capture/compare channel (channel 1 to 3)

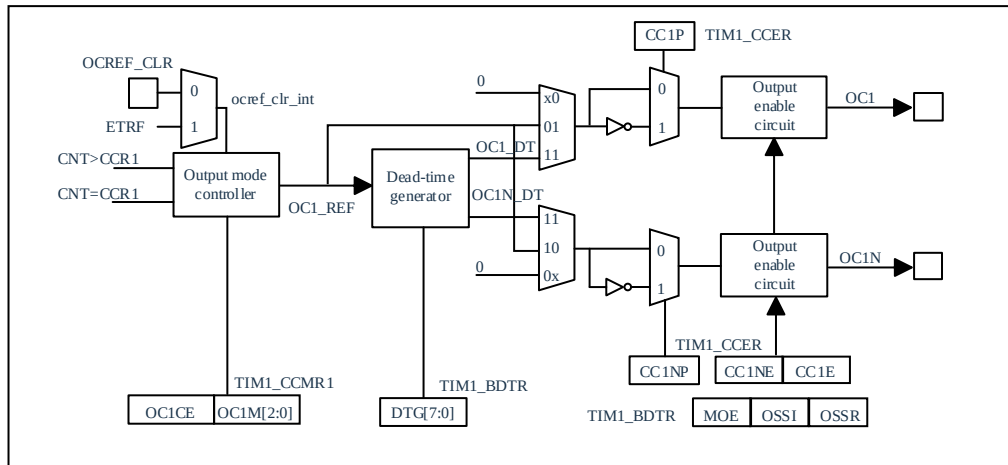
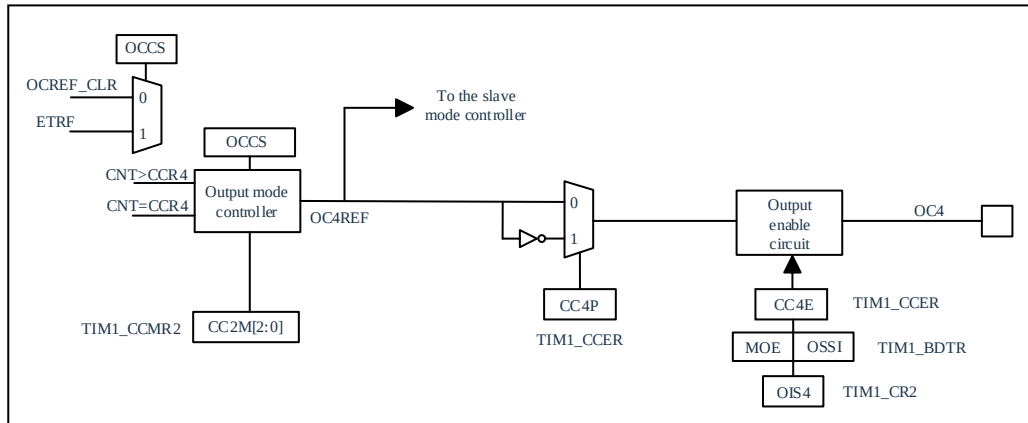


Figure 6-30 Output stage of capture/compare channel (channel 4)



The capture/compare block is made of one preload register and one shadow register. Write and read always access the preload register.

In capture mode, captures are actually done in the shadow register, which is copied into the preload register.

In compare mode, the content of the preload register is copied into the shadow register which is compared to the counter.

6.3.6 Input Capture Mode

In input capture mode, when a corresponding edge of the ICx signal is detected, the current value of the counter is latched into the capture/compare register. When a capture event occurs, the corresponding CCxIF flag (TIM1_SR register) is set to 1. If interrupts or DMA operations are enabled, an interrupt or DMA request will be generated. If the CCxIF flag is already set when a capture event occurs, the capture overrun flag CCxOF (TIM1_SR register) will be set to 1. Writing CCxIF = 0 clears the CCxIF flag, or reading the captured data stored in the TIM1_CCRx register also clears CCxIF. Writing CCxOF = 0 clears CCxOF.

The following example demonstrates how to capture the counter value into the TIM1_CCR1 register on the rising edge of the TI1 input. The steps are as follows:

- ◆ Select valid input: TIM1_CCMR1 must be connected to the TI1 input, so write CC1S = 01 in the TIM1_CCMR1 register. As long as CC1S is not 00, the channel is configured as input, and the TIM1_CCR1 register becomes read-only.
- ◆ Configure input filter: Based on the characteristics of the input signal, configure the input filter for the required bandwidth (i.e., the input filter control bit is the ICxF bit of the TIM1_CCMRx register). Suppose the input signal has jitter within a maximum of 5 internal clock cycles, we must configure the filter bandwidth to be longer than 5 clock cycles. For example, we can sample continuously 8 times (at Fck_int frequency) to confirm the last real edge transition on TI1, i.e., write IC1F = 0011 in the TIM1_CCMR1 register.
- ◆ Select the valid edge for the TI1 channel: Write CC1P = 0 (rising edge) in the TIM1_CCER register.
- ◆ Configure input prescaler: In this example, we want to capture on every valid level transition, so the prescaler is disabled (write IC1PS = 00 in the TIM1_CCMR1 register).
- ◆ Enable capture: Set CC1E = 1 in the TIM1_CCER register to allow capturing the counter value into the capture register.
- ◆ Enable interrupt or DMA request: If necessary, enable the related interrupt request by setting the CC1IE bit in the TIM1_DIER register, or allow DMA requests by setting the CC1DE bit in the TIM1_DIER register.

When an input capture occurs:

- ◆ The counter value is transferred to the TIM1_CCR1 register on a valid edge transition.
- ◆ The CC1IF flag is set (interrupt flag). If at least two consecutive captures occur and CC1IF is not cleared, CC1OF will also be set to 1.
- ◆ If the CC1IE bit is set, an interrupt will occur.
- ◆ If the CC1DE bit is set, a DMA request will also be triggered.

To handle capture overrun, it is recommended to read the capture data before reading the capture overrun flag to avoid missing potential overrun information that may occur between reading the overrun flag and reading the capture data.

Note: Setting the corresponding CCxG bit in the TIM1_EGR register can generate input capture interrupts and/or DMA requests via software.

6.3.7 PWM Input Mode

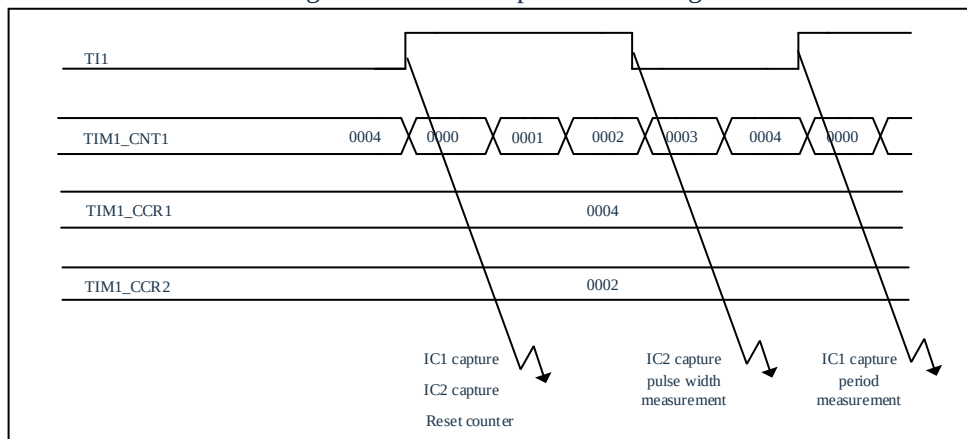
This mode is a special case of input capture mode. Except for the following differences, the operation is the same as in input capture mode:

- ◆ Two ICx signals are mapped to the same TIx input.
- ◆ These two ICx signals are active on edges with opposite polarity.
- ◆ One of the TIxFP signals is used as the trigger input signal and is configured in reset mode by the mode controller.

For example, when measuring the pulse width (in the TIM1_CCR1 register) and the duty cycle (in the TIM1_CCR2 register) of a PWM signal input to TI1, the specific steps are as follows (depending on the CK_INT frequency and the prescaler value):

- ◆ Select valid input for TIM1_CCR1: Set CC1S = 01 in the TIM1_CCMR1 register (select TI1).
- ◆ Select the valid polarity for TI1FP1 (used to capture data into TIM1_CCR1 and reset the counter): Set CC1P = 0 (rising edge active).
- ◆ Select valid input for TIM1_CCR2: Set CC2S = 10 in the TIM1_CCMR1 register (select TI1).
- ◆ Select the valid polarity for TI1FP2 (to capture data into TIM1_CCR2): Set CC2P = 1 (falling edge active).
- ◆ Select valid trigger input signal: Set TS = 101 in the TIM1_SMCR register (select TI1FP1).
- ◆ Configure the mode controller to reset mode: Set SMS = 100 in the TIM1_SMCR register.
- ◆ Enable capture: Set CC1E = 1 and CC2E = 1 in the TIM1_CCER register.

Figure 6-31 PWM input mode timing



6.3.8 Forced Output Mode

In output mode (when $CCxS = 00$ in the TIM1_CCMRx register), the output compare signal (OCxREF and the corresponding OCx/OCxN) can be directly forced into an active or inactive state by software, without relying on the comparison result between the output compare register and the counter. By setting the corresponding OCxM = 101 in the TIM1_CCMRx register, the output compare signal (OCxREF/OCx) is forced into an active state. This forces OCxREF to a high level (OCxREF is always active at a high level), while OCx gets a signal with the opposite polarity to that of CCxP.

For example: if $CCxP = 0$ (OCx is active high), then OCx is forced to a high level. By setting OCxM = 100 in the TIM1_CCMRx register, the OCxREF signal can be forced to a low level.

In this mode, the comparison between the TIM1_CCRx shadow register and the counter still takes place, and the corresponding flags will be updated. As a result, interrupts and DMA requests can still be triggered. This will be further explained in the next section on output compare mode.

6.3.9 Output Compare Mode

This function is used to control an output waveform or indicate that a given time has been reached. When the counter matches the contents of the capture/compare register, the output compare function performs the following actions:

- ◆ Outputs the value defined by the output compare mode (OCxM in the TIM1_CCMRx register) and output polarity (CCxP in the TIM1_CCER register) to the corresponding pin. At the comparison match, the output pin can either maintain its level (OCxM=000), set to active level (OCxM=001), set to inactive level (OCxM=010), or toggle (OCxM=011).
- ◆ Sets the flag in the interrupt status register (CCxIF in the TIM1_SR register).
- ◆ If the corresponding interrupt mask (CCxIE in the TIM1_DIER register) is set, an interrupt will be generated.
- ◆ If the corresponding enable bit (CCxDE in the TIM1_DIER register, CCDS in the TIM1_CR2 register for DMA request functionality) is set, a DMA request will be generated.

The OCxPE bit in TIM1_CCMRx determines whether the TIM1_CCRx register uses the preload register. In output compare mode, the update event (UEV) has no effect on OCxREF and OCx output.

The synchronization accuracy can reach one count cycle of the counter. Output compare mode (in single pulse mode) can also be used to output a single pulse.

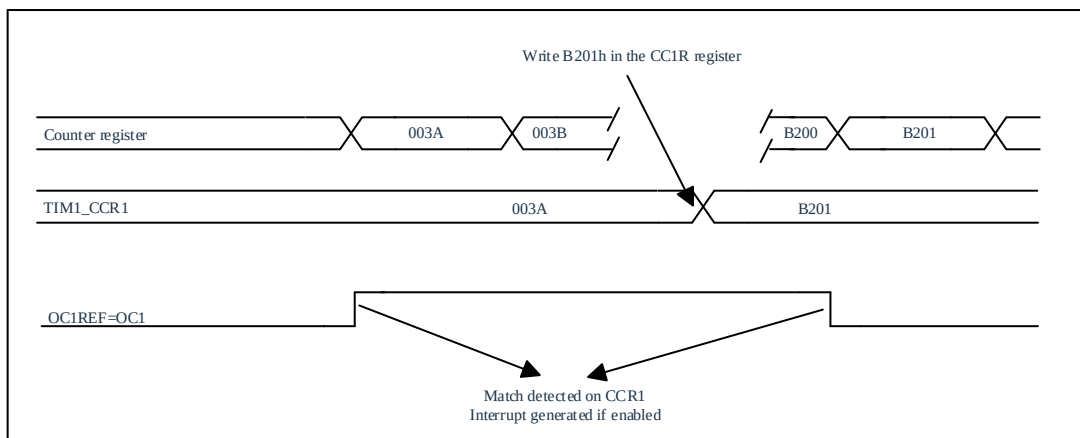
Steps for configuring output compare mode:

1. Select the counter clock (internal, external, prescaler).
2. Write the appropriate data into the TIM1_ARR and TIM1_CCRx registers.
3. If an interrupt request is to be generated, set the CCxIE bit.
4. Select the output mode, for example:
 - To toggle the OCx output pin when the counter matches CCRx, set OCxM=011
 - Set OCxPE = 0 to disable the preload register
 - Set CCxP = 0 to select active high polarity
 - Set CCxE = 1 to enable output

Set the CEN bit in the TIM1_CR1 register to start the counter.

The TIM1_CCRx register can be updated at any time by software to control the output waveform, provided the preload register is not used (OCxPE = “0”; otherwise, the shadow register of TIM1_CCRx can only be updated when the next update event occurs). The figure below provides an example.

Figure 6-32 Output compare mode, toggle on OC1



6.3.10 PWM Mode

The Pulse Width Modulation (PWM) mode allows generating a signal with a frequency determined by the TIM1_ARR register and a duty cycle determined by the TIM1_CCRx register.

When the OCxM bits in the TIM1_CCMRx register are written with “110” (PWM Mode 1) or “111” (PWM Mode 2), each OCx output channel can independently generate a PWM signal. The corresponding preload register must be enabled by setting the OCxPE bit in the TIM1_CCMRx register, and then the ARPE bit in the TIM1_CR1 register must be set to enable the automatic reloading of the preload register in up-counting or center-aligned modes.

The preload register can only be transferred to the shadow register when an update event occurs. Therefore, before the counter begins counting, all registers must be initialized by setting the UG bit in the TIM1_EGR register.

The polarity of OCx can be configured via software by setting the CCxP bit in the TIM1_CCER register, which can be set to active high or active low. The output enable of OCx is controlled by a combination of the CCxE, CCxNE, MOE, OSSI, and OSSR bits in the TIM1_CCER and TIM1_BDTR registers. For further details, refer to the description of the TIM1_CCER register.

In PWM mode (either Mode 1 or Mode 2), TIM1_CNT and TIM1_CCRx are always compared (depending on the counting direction of the counter) to determine if $TIM1\_CCR_x \leq TIM1\_CNT$ or $TIM1\_CNT \leq TIM1\_CCR_x$.

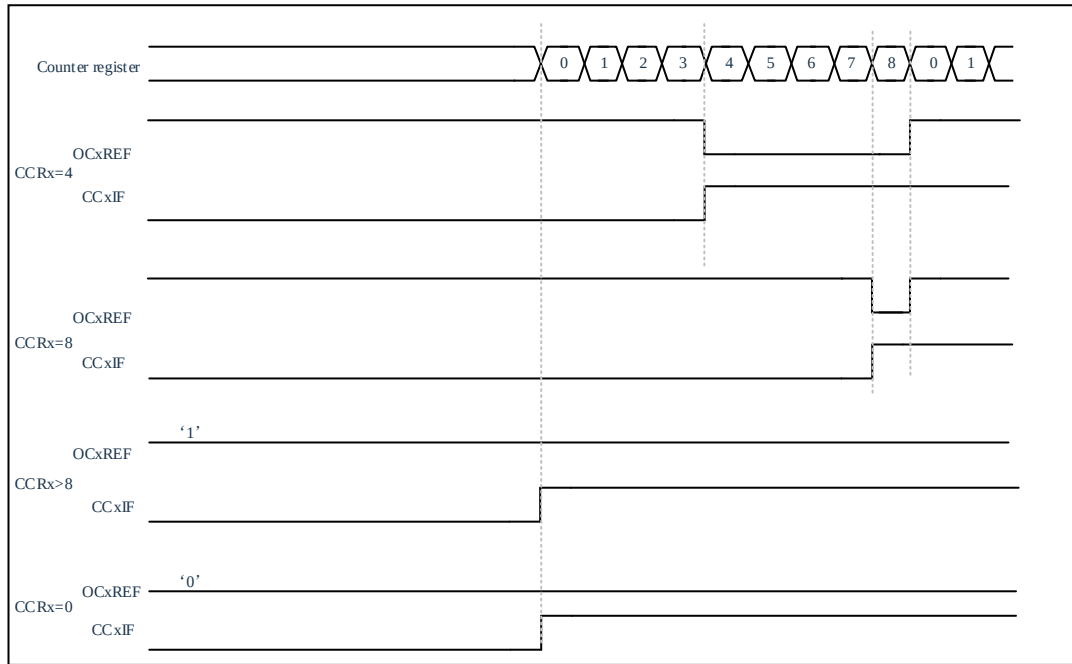
Depending on the state of the CMS bits in the TIM1_CR1 register, the timer can generate either edge-aligned or center-aligned PWM signals.

6.3.10.1 PWM Edge-Aligned Mode

- Up-counting configuration

When the DIR bit in the TIM1_CR1 register is low, up-counting is executed. Below is an example of PWM Mode 1. When $TIM1\_CNT < TIM1\_CCR_x$, the PWM reference signal OCxREF is high; otherwise, it is low. If the comparison value in TIM1_CCRx is greater than the auto-reload value (TIM1_ARR), then OCxREF stays at '1'. If the comparison value is 0, then OCxREF stays at '0'. The figure below shows an example of an edge-aligned PWM waveform when $TIM1\_ARR = 8$.

Figure 6-33 Edge-aligned PWM waveforms (ARR=8)



- Down-counting configuration

When the DIR bit in the TIM1_CR1 register is high, down-counting is executed.

In PWM Mode 1, when $TIM1\_CNT > TIM1\_CCR_x$, the reference signal OCxREF is low; otherwise, it is high. If the comparison value in TIM1_CCRx is greater than the auto-reload value in TIM1_ARR, then OCxREF stays at '1'. This mode cannot generate a 0% PWM waveform.

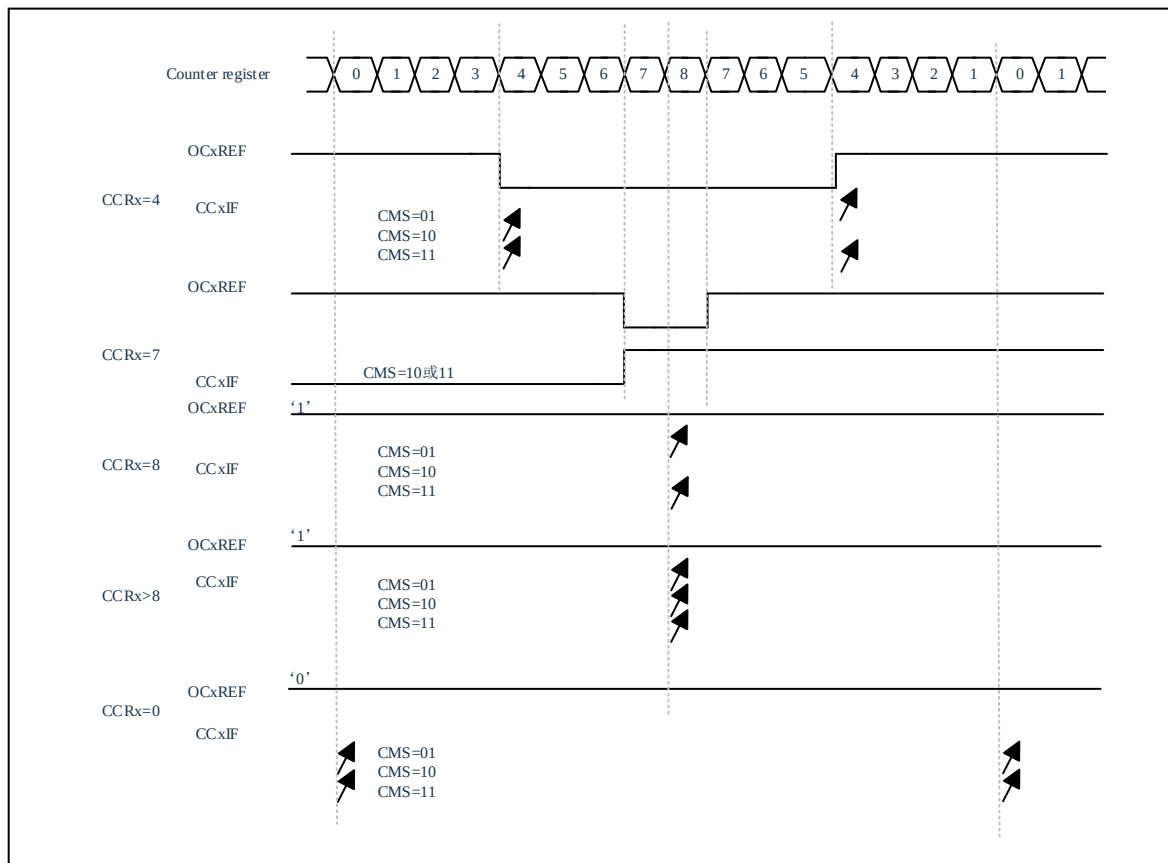
6.3.10.2 PWM Center-Aligned Mode

When the CMS bits in the TIM1_CR1 register are not set to '00', the center-aligned mode is enabled (all other configurations have the same effect on the OCxREF/OCx signal). Depending on the CMS bit setting, the compare flag can be set to 1 when the counter is counting up, when the counter is counting down, or during both up and down counting. The counter direction bit (DIR) in the TIM1_CR1 register is updated by hardware, so it should not be modified by software.

The figure below shows some examples of PWM waveforms in center-aligned mode:

- ◆ TIM1_ARR = 8
- ◆ PWM Mode 1
- ◆ CMS = 01 in the TIM1_CR1 register. In center-aligned mode, the compare flag is set when the counter is counting down.

Figure 6-34 Center-aligned PWM waveforms (ARR=8)



Tips for using center-aligned mode:

- ◆ When entering center-aligned mode, the current up/down counting configuration is used. This means whether the counter counts up or down depends on the current value of the DIR bit in the TIM1_CR1 register. Additionally, software cannot modify both the DIR and CMS bits at the same time.
- ◆ It is not recommended to rewrite the counter when operating in center-aligned mode, as this may produce unpredictable results. Specifically:
 - If a value written to the counter is greater than the auto-reload value (TIM1_CNT > TIM1_ARR), the direction will not be updated.
 For example, if the counter is counting up, it will continue to count up.

- If 0 or the TIM1_ARR value is written to the counter, the direction is updated, but no update event (UEV) is generated.
- ◆ The safest method when using center-aligned mode is to generate a software update (set the UG bit in the TIM1_EGR register) before starting the counter, and avoid modifying the counter value during the counting process.

6.3.11 Complementary Output and Dead-Time Insertion

The Advanced Control Timer (TIM1) is capable of generating two complementary signals and managing the instantaneous turning off and on of the outputs. This time is commonly referred to as dead-time. Users should adjust the dead-time based on the characteristics of the connected output devices (such as level transition delays, power switch delays, etc.).

By configuring the CCxP and CCxNP bits in the TIM1_CCER register, the polarity of each output (main output OCx or complementary output OCxN) can be independently selected.

The complementary signals OCx and OCxN are controlled through the following combination of control bits: CCxE and CCxNE bits in the TIM1_CCER register, MOE, OISx, OISxN, OSSI, and OSSR bits in the TIM1_BDTR and TIM1_CR2 registers. See Table 7-3 for the control bits of complementary output channels OCx and OCxN with brake functionality. Notably, the dead-time is activated when transitioning to the IDLE state (when MOE goes low).

Simultaneously setting both CCxE and CCxNE bits will insert the dead-time. If a brake circuit is present, the MOE bit must also be set. Each channel has an 8-bit dead-time generator DTG[7:0]. The reference signal OCxREF can generate two outputs, OCx and OCxN. If OCx and OCxN are active high:

- ◆ The OCx output signal is the same as the reference signal, but with a delay on the rising edge relative to the rising edge of the reference signal.
- ◆ The OCxN output signal is the opposite of the reference signal, but with a delay on the rising edge relative to the falling edge of the reference signal.

If the delay exceeds the current effective output width (OCx or OCxN), no corresponding pulse will be generated. The following figures illustrate the relationship between the output signal of the dead-time generator and the current reference signal OCxREF (assuming CCxP = 0, CCxNP = 0, MOE = 1, CCxE = 1, and CCxNE = 1).

Figure 6-35 Complementary output with dead-time insertion

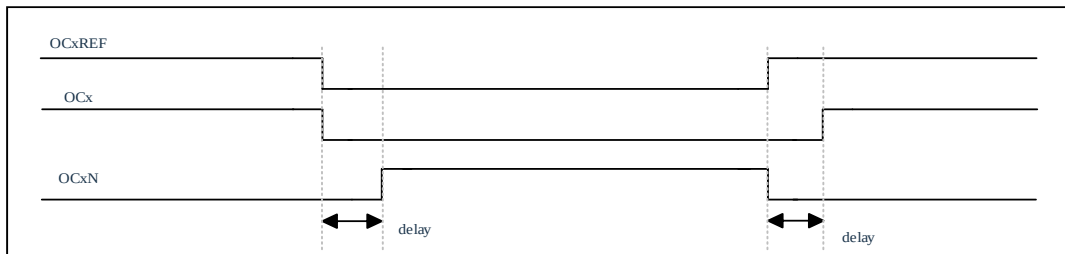


Figure 6-36 Dead-time waveforms with delay greater than the negative pulse

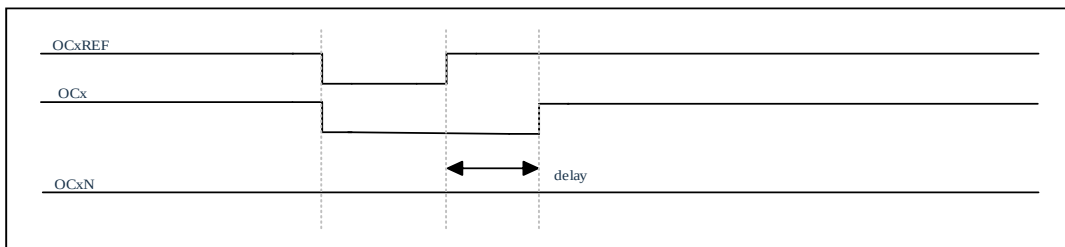
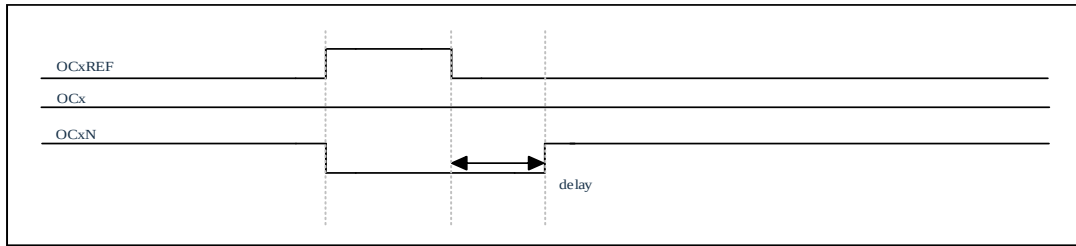


Figure 6-37 Dead-time waveforms with delay greater than the positive pulse



Each channel's dead-time delay is the same and is configured by programming the DTG bits in the TIM1_BDTR register.

6.3.11.1 Redirecting OCxREF to OCx or OCxN

In output modes (forced, output compare, or PWM), the OCxREF signal can be redirected to either the OCx or OCxN output by configuring the CCxE and CCxNE bits in the TIM1_CCER register. This functionality allows for a special waveform (such as PWM or static high level) to be sent to a particular output when the complementary output is at an invalid level. Another use case is to set both outputs to the invalid level simultaneously or to provide complementary outputs with dead-time while one output is at a valid level.

Note: When only OCxN is enabled (CCxE=0, CCxNE=1), it is not complemented and becomes active as soon as OCxREF is high. For example, if CCxNP=0 then OCxN=OCxREF. On the other hand, when both OCx and OCxN are enabled (CCxE=CCxNE=1) OCx becomes active when OCxREF is high whereas OCxN is complemented and becomes active when OCxREF is low.

6.3.12 Using the Brake Function

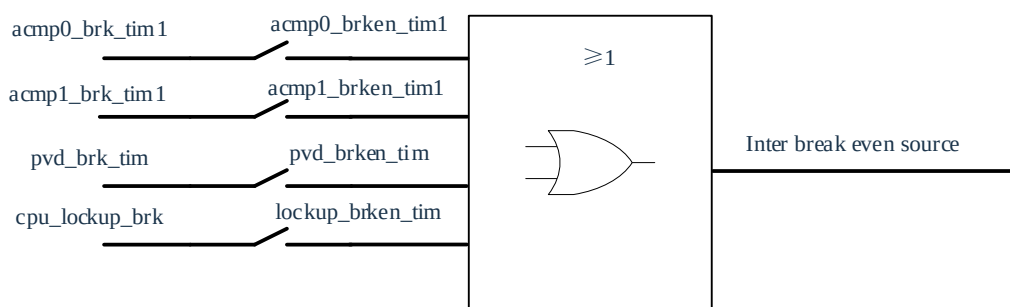
When using the brake function, both the output enable signal and the invalid level signal are modified based on additional control bits. In any case, the OCx and OCxN outputs cannot be at a valid level simultaneously.

The brake source can be either the brake input pin or the following internal sources:

- ◆ CPU LOCKUP output
- ◆ PVD output
- ◆ Comparator output

The block diagram for the internal brake sources is shown below:

Figure 6-38 Internal brake sources block diagram



After a system reset, the brake circuit is disabled, and the MOE bit is set to low. The brake function can be enabled by setting the BKE bit in the TIM1_BDTR register, and the polarity of the brake input signal can be selected by configuring the BKP bit in the same register. Both the BKE and BKP bits can be modified simultaneously. When writing to the BKE and BKP bits, there will be a delay of one APB clock cycle before the bits are actually written, so it is necessary to wait for one APB clock cycle before reading back the written bits correctly.

Because the falling edge of the MOE bit can be asynchronous, a resynchronization circuit is set between the actual signal (applied to the output) and the synchronized control bits (in the TIM1_BDTR register). This resynchronization circuit introduces a delay between the asynchronous and synchronized signals. Specifically, if MOE=1 is written while it is low, a delay (dummy instruction) must be inserted before reading it to obtain the correct value. This is because the written signal is asynchronous, while the read signal is synchronous.

When a brake event occurs (when the selected level appears on the brake input pin), the following actions take place:

- ◆ The MOE bit is asynchronously cleared, placing the output in an invalid state, idle state, or reset state (depending on the OSS1 bit). This feature is still effective when the MCU's oscillator is off.
- ◆ Once MOE=0, each output channel's output is set to the level defined by the OISx bits in the TIM1_CR2 register. If OSS1=0, the timer releases the enable output; otherwise, the enable output remains high.
- ◆ When using complementary outputs:
 - If the timer clock is still active, the dead-time generator will be reactivated. After the dead-time, the output ports are driven according to the levels indicated by the OISx and OISxN bits. Even in this case, OCx and OCxN cannot both be driven to valid levels at the same time. Note: Due to resynchronizing MOE, the dead-time will be longer than usual (approximately two ck_tim clock cycles).
 - If OSS1=0, the timer releases the enable output; otherwise, it keeps the enable output high, or once either CCxE or CCxNE is set high, the enable output will become high.

- ◆ If the BIE bit is set in the TIM1_DIER register, an interrupt will be triggered when the brake status flag (BIF bit in the TIM1_SR register) is '1'.
- ◆ If the AOE bit is set in the TIM1_BDTR register, the MOE bit will be automatically set high at the next update event (UEV); for example, this can be used for shaping. Otherwise, the MOE bit will remain low until set to '1' again. This feature can be used for safety purposes, such as connecting the brake input to an alarm output from the power drive, a thermal sensor, or other safety devices.

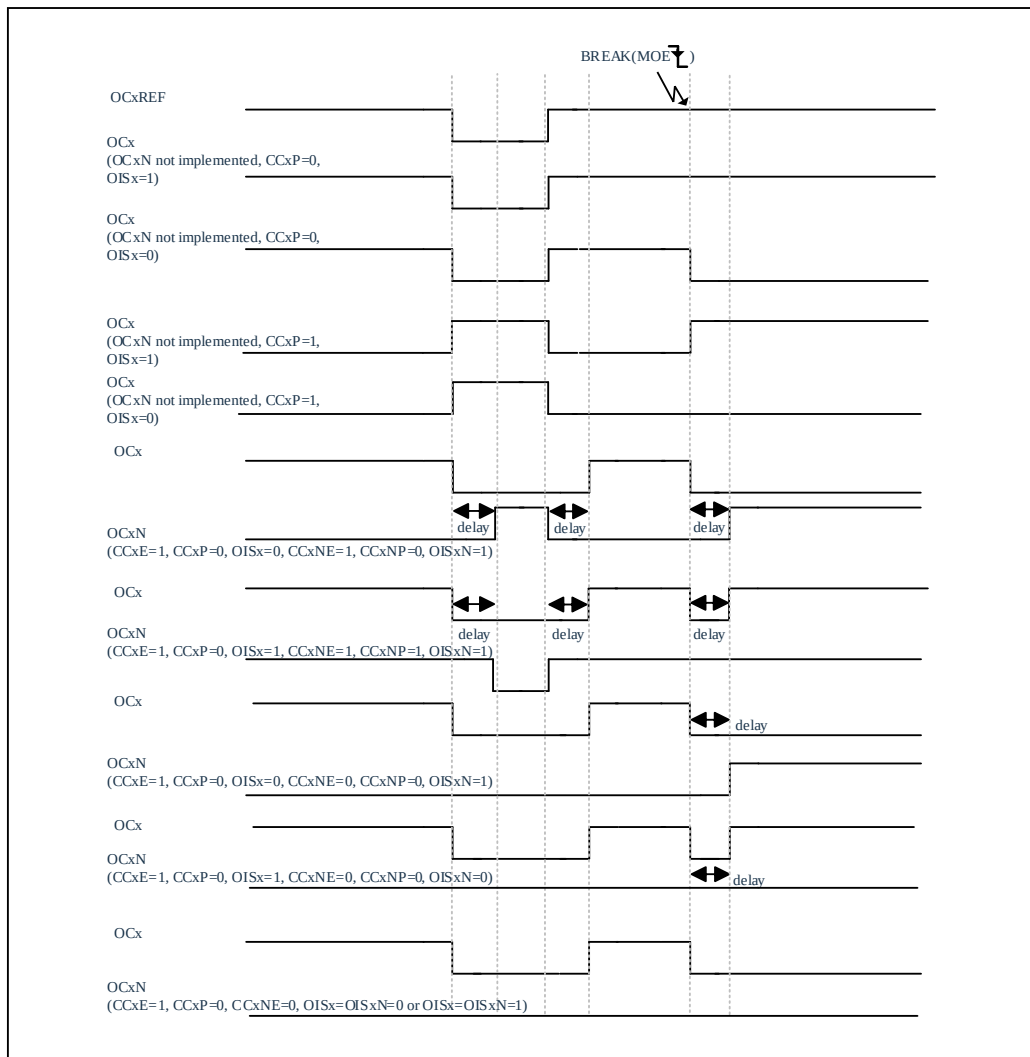
Note: The break inputs is acting on level. Therefore, when the brake input is active, MOE cannot be set simultaneously (automatically or through software). Also, the BIF status flag cannot be cleared.

The brake can be generated by the BRK input, and its active polarity is programmable, enabled by setting the BKE bit in the TIM1_BDTR register.

In addition to brake input and output management, the brake circuit also implements write protection to ensure the safety of the application. It allows the user to freeze several configuration parameters (dead-time length, OCx/OCxN polarity and disabled states, OCxM configuration, brake enable, and polarity). The user can choose one of the three levels of protection via the LOCK bit in the TIM1_BDTR register. After a system reset, the LOCK bit can only be modified once.

The following diagram shows an example of behavior of the outputs in response to a break.

Figure 6-39 Output behavior in response to a break



6.3.13 Clearing the OCxREF Signal on An External Event

For a given channel, setting the corresponding OCxCE bit in the TIM1_CCMRx register to 1 allows the OCxREF signal to be pulled low by a high level on the ETRF input pin. The OCxREF signal will remain low until the next update event (UEV).

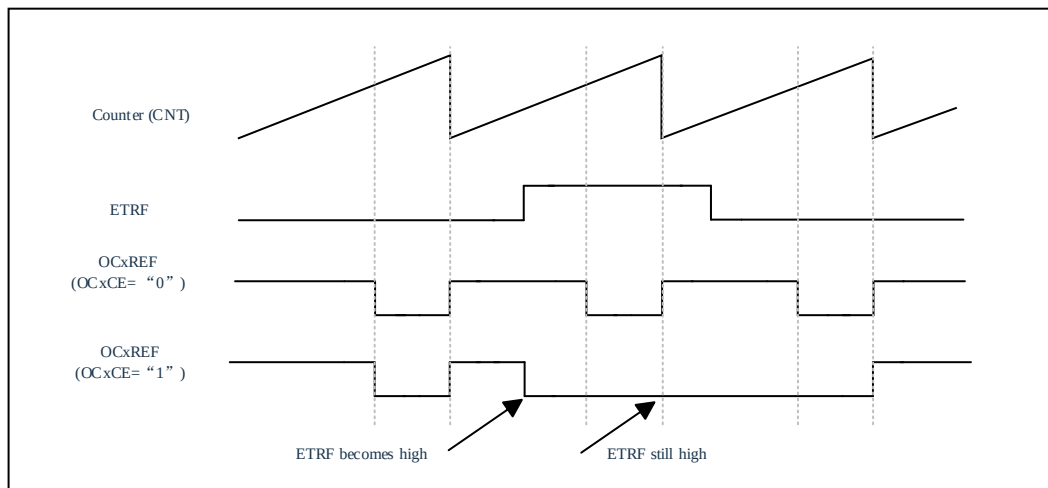
This feature can only be used in output compare and PWM modes, and cannot be used in forced mode.

For example, the OCxREF signal can be connected to the output of a comparator to control current. In this case, ETR must be configured as follows:

1. The external trigger prescaler must be turned off: TIM1_SMCR register, ETPS[1:0] = 00.
2. External clock mode 2 must be disabled: TIM1_SMCR register, ECE = 0.
3. The external trigger polarity (ETP) and external trigger filter (ETF) can be configured as needed.

The following diagram shows the action of the OCxREF signal when the ETRF input becomes high, for different values of OCxCE. In this example, the timer TIM1 is in PWM mode.

Figure 6-40: Clearing the OCxREF Signal in TIM1



Caution: In case of a PWM with a 100% duty cycle (if CCRx>ARR), then OCxREF is enabled again at the next counter overrun.

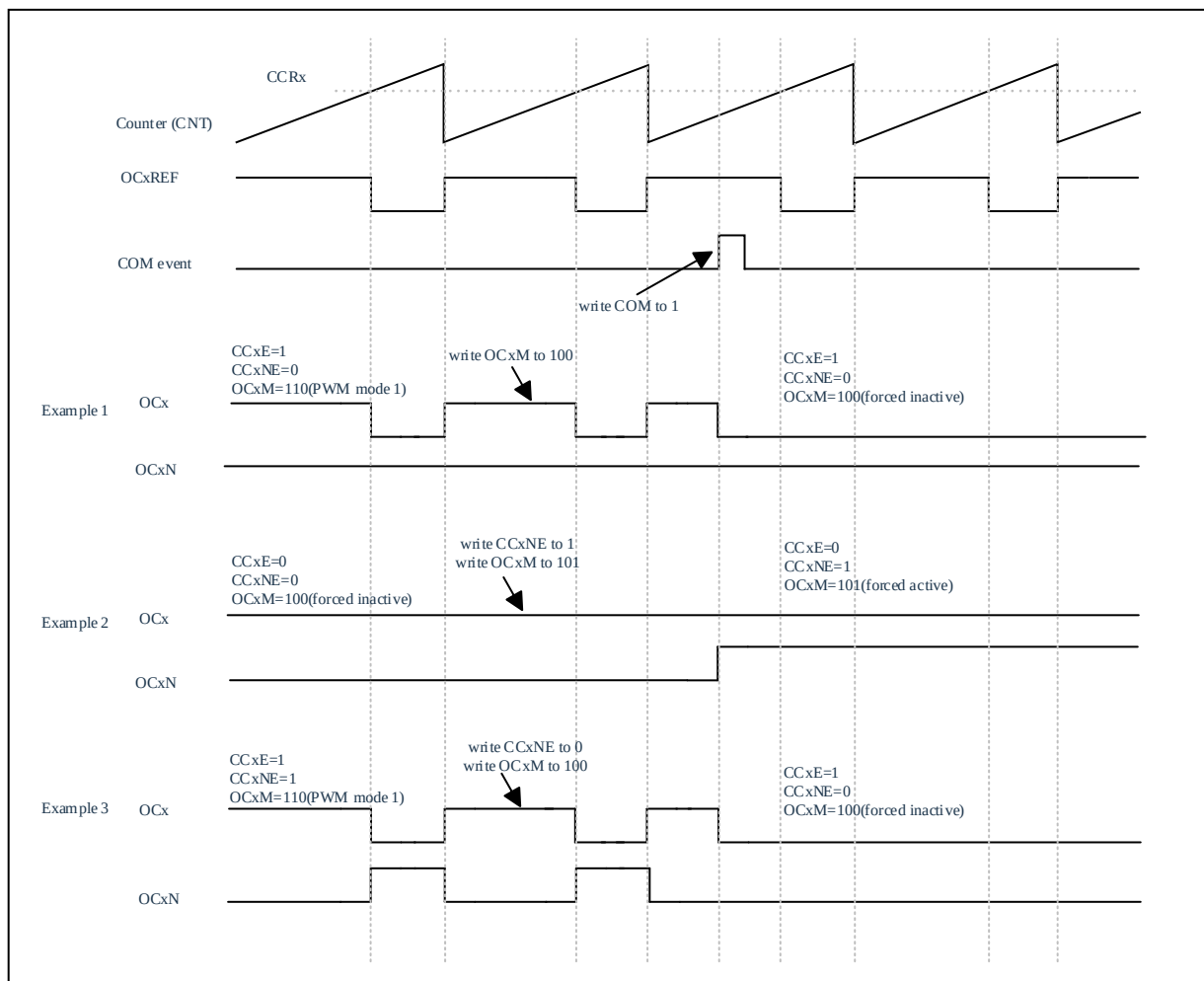
6.3.14 6-Step PWM Generation

When complementary outputs are needed on a channel, the preload bits OCxM, CCxE, and CCxNE are used. During a COM commutation event, these preload bits are transferred to shadow register bits. This allows for the next step configuration to be pre-set, and all channel configurations can be modified simultaneously at the same time. The COM event can be generated by software by setting the COM bit in the TIM1_EGR register, or by hardware on the rising edge of the TRGI.

When the COM event occurs, a flag is set (COMIF bit in the TIM1_SR register). If the COMIE bit in the TIM1_DIER register is set, an interrupt is generated; if the COMDE bit in the TIM1_DIER register is set, a DMA request is generated.

The following diagram shows the OCx and OCxN outputs under three different configurations when a COM event occurs.

Figure 6-41 6-step generation, COM example (OSSR=1)



6.3.15 One Pulse Mode

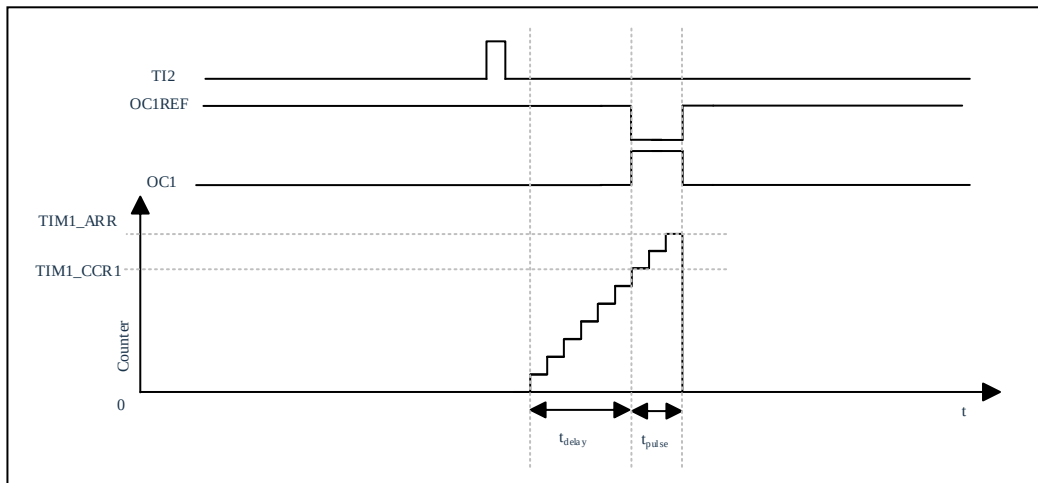
One Pulse Mode (OPM) is a special case of the many modes previously described. This mode allows the counter to respond to a trigger and, after a program-controlled delay, generate a pulse with a pulse width that can be controlled by the program.

Waveforms can be generated in output compare mode or PWM mode by starting the counter from the mode controller. Setting the OPM bit in the TIM1_CR1 register selects One Pulse Mode, which causes the counter to automatically stop when the next update event (UEV) is generated.

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

- ◆ Upcounting mode: $CNT < CCRx \leq ARR$ (specifically, $0 < CCRx$)
- ◆ Downcounting mode: $CNT > CCRx$

Figure 6-42 Example of one pulse mode



For example, when a rising edge on the TI2 input pin needs to be detected, and after a delay of t_{DELAY} , a positive pulse of length t_{PULSE} is generated on OC1:

Using TI2FP2 as Trigger 1:

- ◆ Set $CC2S = 01$ in the TIM1_CCMR1 register to map TI2FP2 to TI2.
- ◆ Set $CC2P = 0$ in the TIM1_CCER register to enable detection of the rising edge on TI2FP2.
- ◆ Set $TS = 110$ in the TIM1_SMCR register to use TI2FP2 as the trigger (TRGI) for the slave mode controller.
- ◆ Set $SMS = 110$ in the TIM1_SMCR register (trigger mode), so TI2FP2 is used to start the counter.

The waveform for OPM is determined by the values written to the compare registers (considering clock frequency and counter prescaler):

- ◆ t_{DELAY} is defined by the value in the TIM1_CCR1 register.
- ◆ t_{PULSE} is defined by the difference between the auto-reload value and the compare value ($TIM1\_ARR - TIM1\_CCR1$).
- ◆ It is assumed that a waveform from 0 to 1 is generated when a compare match occurs, and a waveform from 1 to 0 is generated when the counter reaches the preload value.

First, set OC1M = 111 in the TIM1_CCMR1 register to enter PWM mode 2. Optionally enable the preload register: set OC1PE = 1 in the TIM1_CCMR1 register and ARPE in the TIM1_CR1 register. Then, write the compare value to the TIM1_CCR1 register, the auto-reload value to the TIM1_ARR register, and set the UG bit to generate an update event. Afterward, wait for an external trigger event on TI2. In this example, CC1P = 0.

In this example, the DIR and CMS bits in the TIM1_CR1 register should be cleared.

Since only one pulse is needed, set the OPM = 1 in the TIM1_CR1 register, so the counter stops on the next update event (when the counter rolls over from the auto-reload value to 0).

Particular case: OCx fast enable:

In Single Pulse Mode, the edge detection logic on the TIx input pin sets the CEN bit to start the counter. Then, the comparison between the counter and compare value generates the output transition. However, these operations require some clock cycles, which limits the minimum delay time t_{DELAY} .

To output a waveform with the minimum delay, you can set the OCxFE bit in the TIM1_CCMRx register. At this point, OCxREF (and OCx) responds directly to the trigger without relying on the compare result, and the output waveform matches the waveform at the time of compare match. OCxFE only works when the channel is configured in PWM1 or PWM2 mode.

6.3.16 Encoder Interface Mode

The method for selecting the encoder interface mode is as follows:

If the counter counts only on the edge of TI2, set SMS = 001 in the TIM1_SMCR register.

If the counter counts only on the edge of TI1, set SMS = 010 in the TIM1_SMCR register.

If the counter counts on both TI1 and TI2 edges, set SMS = 011.

By setting the CC1P and CC2P bits in the TIM1_CCER register, the polarity of TI1 and TI2 can be selected. Input filters can also be programmed if necessary.

Both inputs TI1 and TI2 are used as the interface for an incremental encoder. Refer to Table 6-1. Assuming the counter is already enabled (CEN = 1 in the TIM1_CR1 register), the counter is driven by each valid transition on TI1FP1 or TI2FP2. TI1FP1 and TI2FP2 are the signals from TI1 and TI2 after passing through the input filters and polarity control; if no filtering or phase-shifting is applied, then TI1FP1 = TI1 and TI2FP2 = TI2. Based on the transition order of the two input signals, the counting pulse and direction signal are generated. Depending on the transition order of the two input signals, the counter counts up or down, and the hardware sets the DIR bit in the TIM1_CR1 register accordingly. Regardless of whether the counter relies on TI1, TI2, or both to count, any transition on either input (TI1 or TI2) will recalculate the DIR bit.

The encoder interface mode is essentially equivalent to using an external clock with direction selection. This means the counter only counts continuously between 0 and the TIM1_ARR register's auto-reload value (counting from 0 to ARR or from ARR to 0, depending on the direction). Therefore, the TIM1_ARR register must be configured before counting starts. Similarly, features such as the capture unit, comparator, prescaler, repetition counter, and trigger output still operate as usual. Encoder mode is not compatible with external clock mode 2, and they cannot operate simultaneously. In this mode, the counter is automatically adjusted based on the speed and direction of the incremental encoder, so the counter's content always indicates the encoder's position. The counting direction corresponds to the rotation direction of the connected sensor. The following table lists all possible combinations, assuming TI1 and TI2 do not change simultaneously.

Table 6-1 Counting direction versus encoder signals

Active edge	Level on opposite signal (TI1FP1 for TI2, TI2FP2 for TI1)	TI1FP1 signal		TI1FP2 signal	
		Rising	Falling	Rising	Falling
Counting on TI1 only	High	Down	Up	No Count	No Count
	Low	Up	Down	No Count	No Count
Counting on TI2 only	High	No Count	No Count	Up	Down
	Low	No Count	No Count	Down	Up
Counting on TI1 and TI2	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

An external incremental encoder can be connected directly to the MCU without external interface logic. However, a comparator is generally used to convert the differential output of the encoder to a digital signal, which greatly enhances resistance to noise interference. The third signal from the encoder represents the mechanical zero point, which can be connected to an external interrupt input to trigger a counter reset.

The diagram below shows an example of counter operation, illustrating how counting signals are generated and how direction control works. It also shows how input jitter is suppressed when dual-edge detection is selected. Jitter may occur when the sensor's position is near a transition point. In this example, we assume the following configuration:

- ◆ CC1S='01'(TIM1_CCMR1 register, TI1FP1 mapped on TI1)
- ◆ CC2S='01'(TIM1_CCMR2 register, TI1FP2 mapped on TI2)
- ◆ CC1P='0'(TIM1_CCER register, TI1FP1 non-inverted, TI1FP1=TI1)
- ◆ CC2P='0'(TIM1_CCER register, TI1FP2 non-inverted, TI1FP2=TI2)
- ◆ SMS='011'(TIM1_SMCR register, both inputs are active on both rising and falling edges)
- ◆ CEN='1'(TIM1_CR1 register, counter enabled)

Figure 6-43 Example of counter operation in encoder interface mode

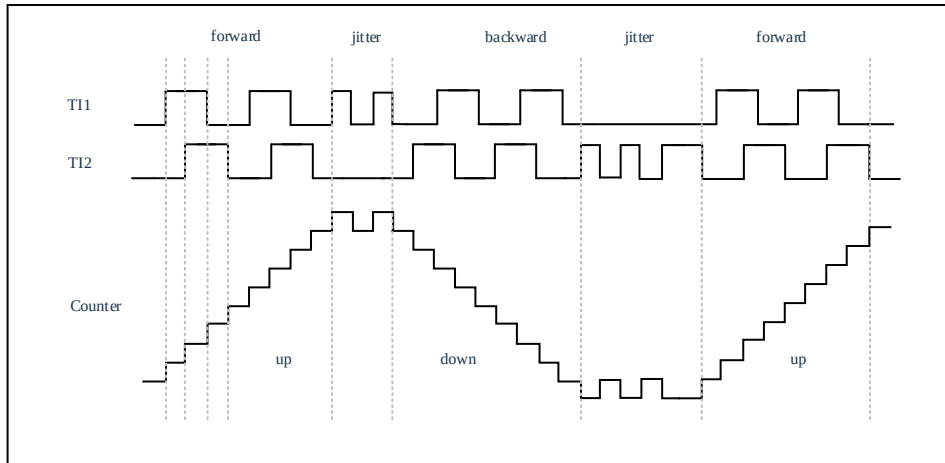
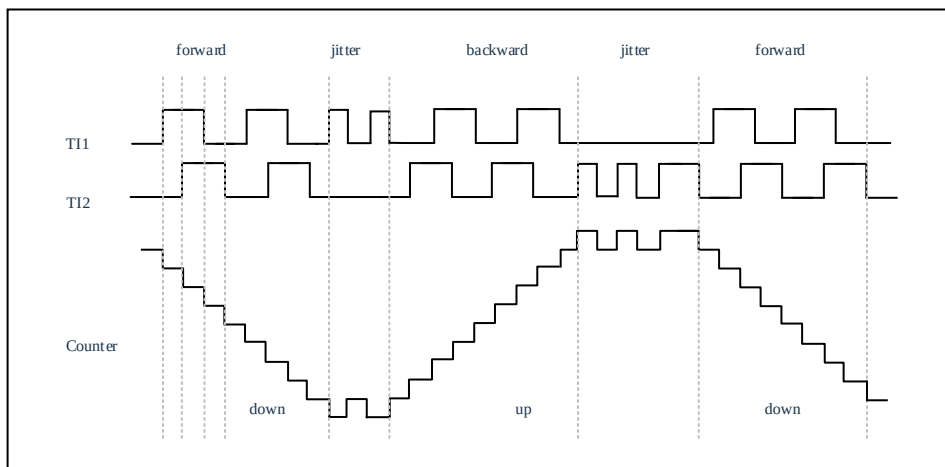


Figure 6-44 Example of encoder interface mode with TI1FP1 polarity inverted



When the timer is configured in encoder interface mode, it provides the current position information of the sensor. By using the second configuration in capture mode of the timer, the interval between two encoder events can be measured, obtaining dynamic information (speed, acceleration, deceleration). The encoder output indicating the mechanical zero point can be used for this purpose. Based on the interval between the two events, the counter can be read out at fixed times. If possible, the counter value can be latched into the third input capture register (the capture signal must be periodic and can be generated by another timer); it can also be read via a DMA request generated by the real-time clock.

6.3.17 Timer Input XOR Function

The TI1S bit of the TIM1_CR2 register allows the input filter of channel 1 to be connected to the output of an XOR gate, with three inputs to the XOR gate being TIM1_CH1, TIM1_CH2, and TIM1_CH3.

The XOR output can be used for input functions of all timers, such as triggering or input capture.

6.3.18 Interfacing With Hall Sensors

When using an advanced control timer (TIM1) to generate PWM signals to drive a motor, another general-purpose timer (TIM3) can be used as an interface timer to interface with the Hall sensor. The timer input pins (CC1, CC2, CC3) of TIM3 are connected to the TI1 input channel through an XOR gate (selected by setting the TI1S bit in the TIM3_CR2 register). The interface timer captures this signal.

The mode controller is configured to reset mode, and the input is TI1F_ED. Every time one of the three inputs changes, the counter starts counting from 0 again. This generates a time base triggered by any change at the Hall input.

The capture/compare channel 1 of the interface timer is configured to capture mode, and the capture signal is TRC. The capture value reflects the time delay between two input changes, providing information about the motor speed.

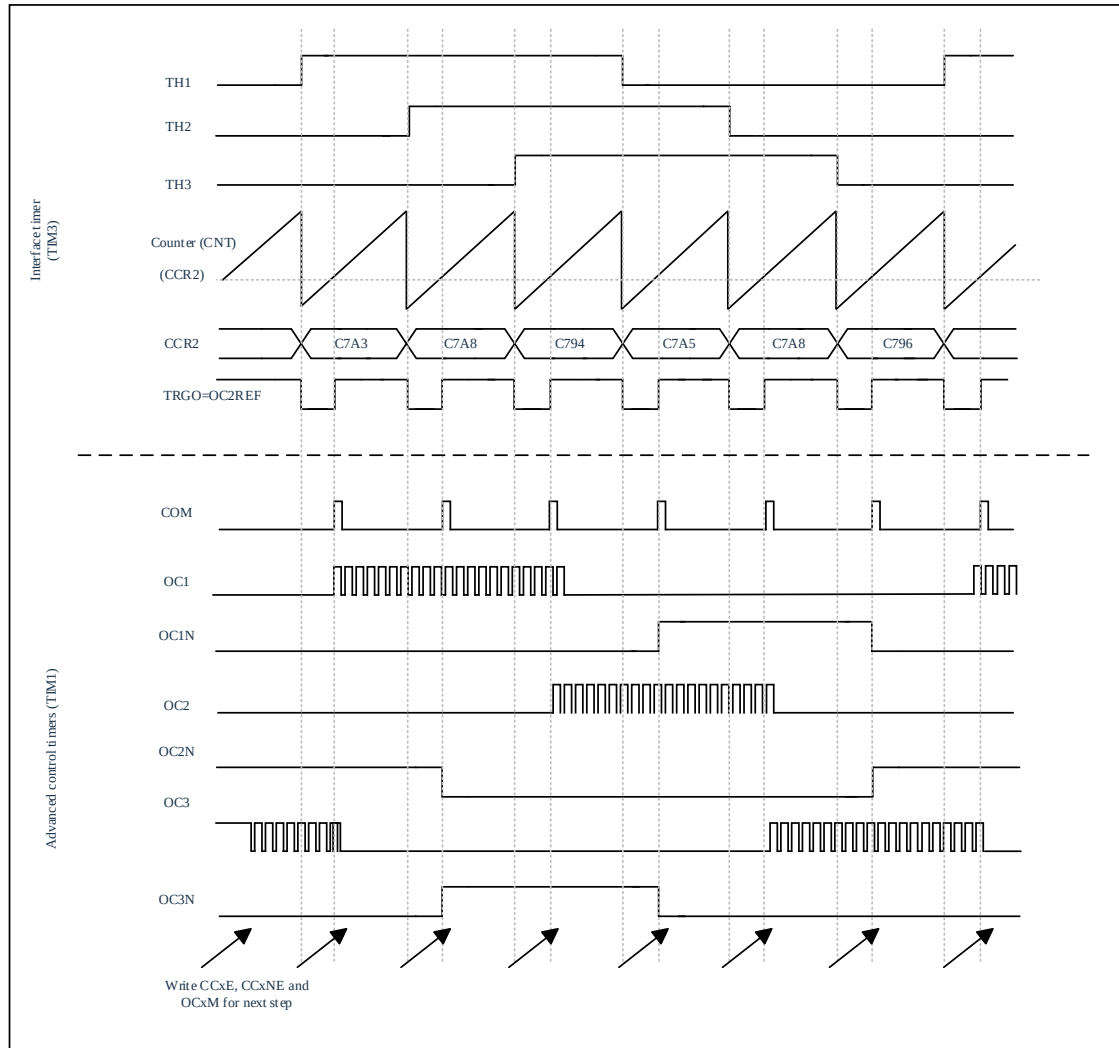
The interface timer can be used to generate a pulse in the output mode, which can (by triggering a COM event) change the properties of the channels of the advanced control timer TIM1, which generates PWM signals to drive the motor. Therefore, the interface timer channel must be programmed to generate a positive pulse after a specified delay (output compare or PWM mode), which is sent to the advanced control timer TIM1 via the TRGO output.

Example: The Hall input is connected to the TIM3 timer, and after each change on any Hall input, the PWM configuration of the advanced control timer TIM1 needs to be changed at a specified moment.

- ◆ Set the TI1S bit of the TIM3_CR2 register to '1' to configure three timer inputs XORed to the TI1 input.
- ◆ Time base programming: Set TIM3_ARR to its maximum value (the counter must be cleared by changes on TI1). Set the prescaler to achieve the longest counter period, which is longer than the time interval between two changes on the sensor.
- ◆ Set channel 1 to capture mode (select TRC): Set CC1S = 01 in the TIM3_CCMR1 register, and if needed, configure the digital filter.
- ◆ Set channel 2 to PWM2 mode, with the required delay: Set OC2M = 111 and CC2S = 00 in the TIM3_CCMR1 register.
- ◆ Select OC2REF as the trigger output on TRGO: Set MMS = 101 in the TIM3_CR2 register.

In the advanced-control timer TIM1, the right ITR input must be selected as trigger input, the timer is programmed to generate PWM signals, the capture/compare control signals are preloaded (CCPC=1 in the TIM1_CR2 register) and the COM event is controlled by the trigger input (CCUS=1 in the TIM1_CR2 register). The PWM control bits (CCxE, OCxM) are written after a COM event for the next step (this can be done in an interrupt subroutine generated by the rising edge of OC2REF).

Figure 6-45 Example of hall sensor interface



6.3.19 TIM1 and External Trigger Synchronization

The TIM1 timer can synchronize with an external trigger in various modes: reset mode, gated mode, and trigger mode.

6.3.19.1 Slave Mode: Reset Mode

When a trigger input event occurs, the counter and its prescaler can be re-initialized. Additionally, if the URS bit in the TIM1_CR1 register is low, an update event (UEV) is generated, and all preload registers (TIM1_ARR, TIM1_CCRx) are updated.

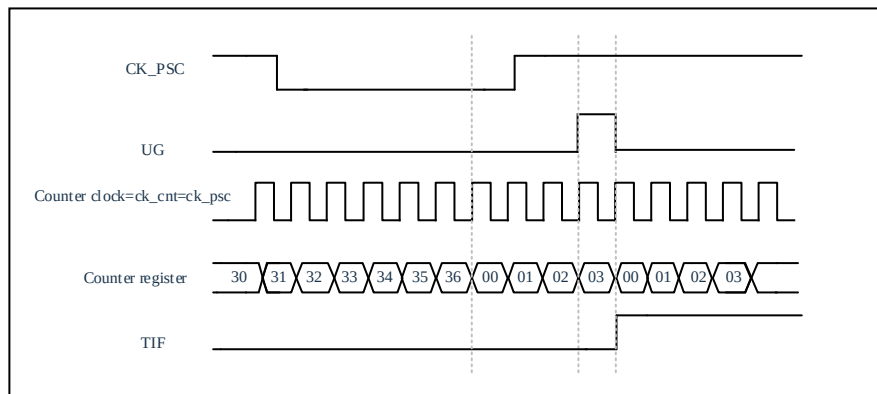
In the following example, the rising edge of the TI1 input causes the up counter to be reset:

- ◆ Configure channel 1 to detect the rising edge of TI1. Configure the input filter bandwidth (in this example, no filter is needed, so keep IC1F = 0000). No capture prescaler is used in the trigger operation, so no configuration is needed. The CC1S bit only selects the input capture source, i.e., CC1S = 01 in the TIM1_CCMR1 register. Set CC1P = 0 in the TIM1_CCER register to determine polarity (only detecting rising edge).
- ◆ Set SMS = 100 in the TIM1_SMCR register to configure the timer for reset mode; set TS = 101 in the TIM1_SMCR register to select TI1 as the input source.
- ◆ Set CEN = 1 in the TIM1_CR1 register to start the counter.

The counter begins counting based on the internal clock and then operates normally until a rising edge on TI1 occurs; at this point, the counter is reset and starts counting from 0. Simultaneously, the trigger flag (TIF bit in the TIM1_SR register) is set, and an interrupt request or DMA request is generated depending on the settings of the TIE (interrupt enable) bit and TDE (DMA enable) bit in the TIM1_DIER register.

The following diagram shows the operation when the auto-reload register TIM1_ARR = 0x36. The delay between the rising edge of TI1 and the actual reset of the counter depends on the re-synchronization circuit of the TI1 input.

Figure 6-46 Control circuit in reset mode



6.3.19.2 Slave Mode: Gated Mode

The counter is enabled according to the level of the selected input.

In the following example, the counter counts upwards only when TI1 is low:

- ◆ Configure channel 1 to detect the low level on TI1. Configure the input filter bandwidth (in this example, no filter is needed, so keep IC1F = 0000).

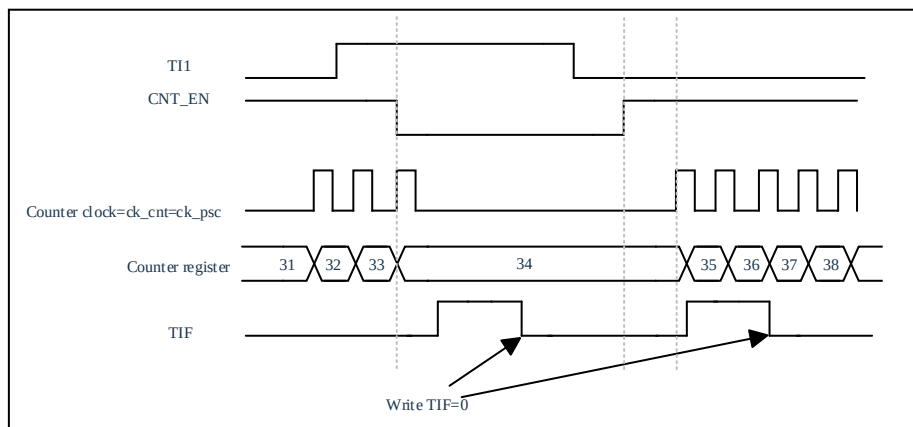
No capture prescaler is used in the trigger operation, so no configuration is needed. The CC1S bit is used to select the input capture source, i.e., CC1S = 01 in the TIM1_CCMR1 register. Set CC1P = 1 in the TIM1_CCER register to determine the polarity (only detecting low level).

- ◆ Set SMS = 101 in the TIM1_SMCR register to configure the timer for gated mode; set TS = 101 in the TIM1_SMCR register to select TI1 as the input source.
- ◆ Set CEN = 1 in the TIM1_CR1 register to start the counter. In gated mode, if CEN = 0, the counter cannot start, regardless of the trigger input level.

As long as TI1 is low, the counter starts counting based on the internal clock. Once TI1 becomes high, the counting stops. The TIF flag in the TIM1_SR register is set both when the counter starts and when it stops.

The delay between the rising edge of TI1 and the actual stop of the counter depends on the re-synchronization circuit of the TI1 input.

Figure 6-47 Control circuit in gated mode



6.3.19.3 Slave Mode: Trigger Mode

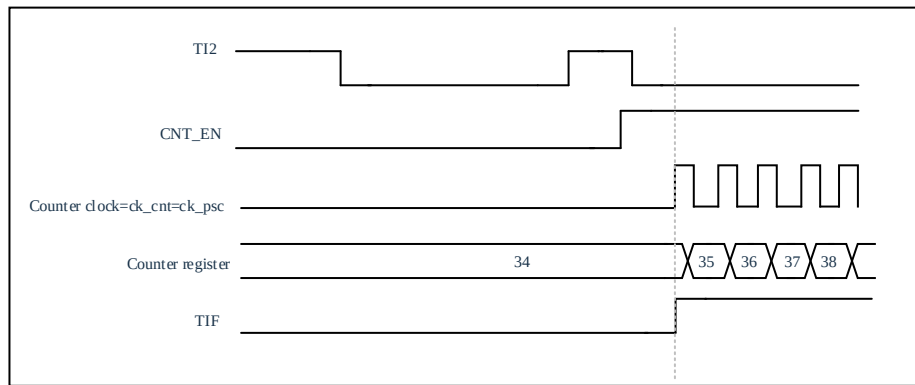
The counter is enabled by the selected event on the input.

In the following example, the counter starts counting upwards on the rising edge of the TI2 input:

- ◆ Configure channel 2 to detect the rising edge of TI2. Configure the input filter bandwidth (in this example, no filter is needed, so keep IC2F = 0000). No capture prescaler is used in the trigger operation, so no configuration is needed. The CC2S bit is used to select the input capture source, i.e., CC2S = 01 in the TIM1_CCMR1 register. Set CC2P = 1 in the TIM1_CCER register to determine the polarity (only detecting the rising edge).
- ◆ Set SMS = 110 in the TIM1_SMCR register to configure the timer for trigger mode; set TS = 110 in the TIM1_SMCR register to select TI2 as the input source.

When a rising edge appears on TI2, the counter starts counting under the internal clock's control, and the TIF flag is set simultaneously. The delay between the rising edge of TI2 and the actual start of counting depends on the re-synchronization circuit of the TI2 input.

Figure 6-48 Control circuit in trigger mode



6.3.19.4 Slave Mode: External Clock Mode 2 + Trigger Mode

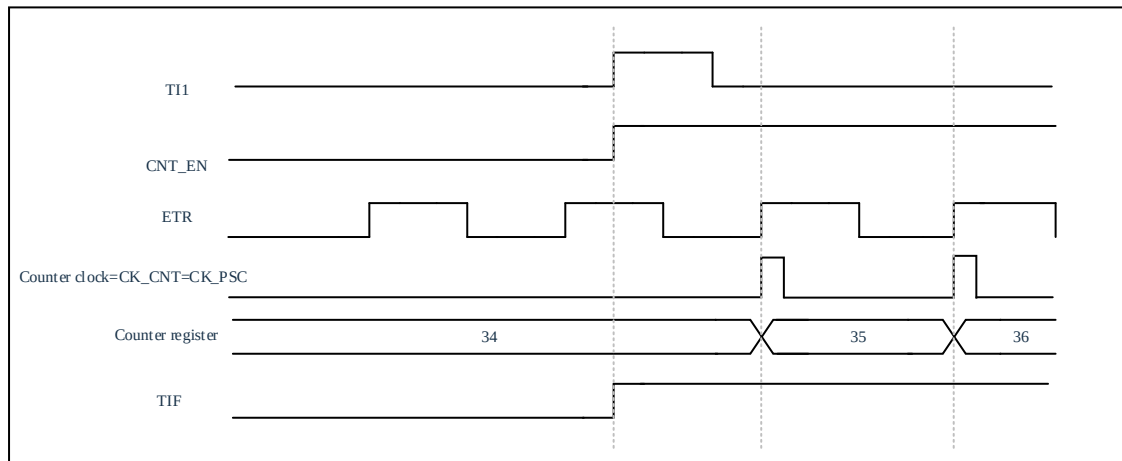
External Clock Mode 2 can be used with another slave mode (except External Clock Mode 1 and Encoder Mode). In this case, the ETR signal is used as the input for the external clock, and either reset mode, gated mode, or trigger mode can select another input as the trigger input. It is not recommended to use the TS bit in the TIM1_SMCR register to select ETR as TRGI.

In the following example, once a rising edge occurs on TI1, the counter counts upwards on every rising edge of the ETR signal:

1. Configure the external trigger input circuit through the TIM1_SMCR register:
 - ETF=0000: No filtering
 - ETPS=00: No prescaler
 - ETP=0: Detect rising edge of ETR, set ECE = 1 to enable External Clock Mode 2.
2. Configure channel 1 to detect the rising edge of TI1:
 - IC1F=0000: No filtering
 - No capture prescaler is used in the trigger operation, so no configuration is needed.
 - Set CC1S = 01 in the TIM1_CCMR1 register to select the input capture source.
 - Set CC1P = 0 in the TIM1_CCER register to determine the polarity (only detect the rising edge).
3. Set SMS = 110 in the TIM1_SMCR register to configure the timer for trigger mode. Set TS = 101 in the TIM1_SMCR register to select TI1 as the input source.

When a rising edge appears on TI1, the TIF flag is set, and the counter begins counting on each rising edge of the ETR signal. The delay between the rising edge of the ETR signal and the actual counter reset depends on the re-synchronization circuit of the ETRP input.

Figure 6-49 Control circuit in external clock mode 2 + trigger mode



6.3.20 Timer Synchronization

The TIM timers are internally connected for synchronization or chaining. When a timer is in master mode, it can reset, start, stop, or clock the counter of another timer in slave mode. For more details, refer to Section 8.3.15: Timer Synchronization.

6.3.21 Debug Mode

When the chip enters debug mode, the TIM1 counter can either continue to operate normally or stop, depending on the settings in Section 1.4.

6.4 Register Mapping

(TIM1 base address = 0x4007_0000)

RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Address	R/W	Description	Reset value
CR1	0x00	R/W	TIM1 Control Register 1	0x00
CR2	0x04	R/W	TIM1 Control Register 2	0x00
SMCR	0x08	R/W	TIM1 Slave Mode Control Register	0x00
DIER	0x0C	R/W	TIM1 DMA/Interrupt Enable Register	0x00
SR	0x10	R/W	TIM1 Status Register	0x00
EGR	0x14	R/W	TIM1 Event Generation Register	0x00
CCMR1	0x18	R/W	TIM1 Capture/Compare Mode Register 1	0x00
CCMR2	0x1C	R/W	TIM1 Capture/Compare Mode Register 2	0x00
CCER	0x20	R/W	TIM1 Capture/Compare Enable Register	0x00
CNT	0x24	R/W	TIM1 Counter	0x00
PSC	0x28	R/W	TIM1 Prescaler	0x00
ARR	0x2C	R/W	TIM1 Auto-Reload Register	0xFFFF
RCR	0x30	R/W	TIM1 Repetition Counter Register	0x00
CCR1	0x34	R/W	TIM1 Capture/Compare Register 1	0x00
CCR2	0x38	R/W	TIM1 Capture/Compare Register 2	0x00
CCR3	0x3C	R/W	TIM1 Capture/Compare Register 3	0x00
CCR4	0x40	R/W	TIM1 Capture/Compare Register 4	0x00
BDTR	0x44	R/W	TIM1 Brake and Dead-Time Register	0x00
DCR	0x48	R/W	TIM1 DMA Control Register	0x00
DMAR	0x4C	R/W	TIM1 Full Transfer DMA Address	0x00

6.5 Register Description

6.5.1 TIM1 Control Register 1 (TIM1_CR1)

Bit	Symbol	Description	Reset value
31:10	--	Reserved, set to 0.	0x0
9:8	CKD[1:0]	Clock division This bit-field indicates the division ratio between the timer clock (CK_INT) frequency and the dead-time and sampling clock (t_{DTS}) used by the dead-time generators and the digital filters (ETR, TIx), 00: t_{DTS} = t_{CK_INT} 01: t_{DTS} = 2 * t_{CK_INT} 10: t_{DTS} = 4 * t_{CK_INT} 11: Reserved, do not program this value	0x0
7	ARPE	Auto-reload preload enable 0: TIM1_ARR register is not buffered 1: TIM1_ARR register is buffered	0
6:5	CMS[1:0]	Center-aligned mode selection 00: Edge-aligned mode. The counter counts up or down depending on the direction bit (DIR). 01: Center-aligned mode 1. The counter counts up and down alternatively. Output compare interrupt flags of channels configured in output (CCxS=00 in TIM1_CCMRx register) are set only when the counter is counting down. 10: Center-aligned mode 2. The counter counts up and down alternatively. Output compare interrupt flags of channels configured in output (CCxS=00 in TIM1_CCMRx register) are set only when the counter is counting up. 11: Center-aligned mode 3. The counter counts up and down alternatively. Output compare interrupt flags of channels configured in output (CCxS=00 in TIM1_CCMRx register) are set both when the counter is counting up or down. Note: It is not allowed to switch from edge-aligned mode to center-aligned mode as long as the counter is enabled (CEN=1).	0x0
4	DIR	Direction 0: Counter used as upcounter 1: Counter used as downcounter Note: This bit is read only when the timer is configured in Center-aligned mode or Encoder mode.	0
3	OPM	One pulse mode 0: Counter is not stopped at update event 1: Counter stops counting at the next update event (clearing the bit CEN)	0
2	URS	Update request source This bit is set and cleared by software to select the UEV event sources. 0: Any of the following events generate an update interrupt or DMA request if enabled. These events can be: - An update interrupt or DMA request - Counter overrun/underrun - Setting the UG bit - Update generation through the slave mode controller 1: Only counter overrun/underrun generates an update interrupt or DMA request if enabled.	0
1	UDIS	Update disable This bit is set and cleared by software to enable/disable UEV event generation. 0: UEV enabled. The Update (UEV) event is generated by one of the following events: - Counter overrun/underrun - Setting the UG bit - Update generation through the slave mode controller Buffered registers are then loaded with their preload values. 1: UEV disabled. The Update event is not generated, shadow registers keep their value (ARR, PSC, CCRx). However the	0

		counter and the prescaler are reinitialized if the UG bit is set or if a hardware reset is received from the slave mode controller.	
0	CEN counter enable	Counter enable 0: Counter disabled 1: Counter enabled Note: External clock, gated mode and encoder mode can work only if the CEN bit has been previously set by software. However trigger mode can set the CEN bit automatically by hardware.	0

6.5.2 TIM1 Control Register 2(TIM1_CR2)

Bit	Symbol	Description	Reset value
31:15	--	Reserved, set to 0.	0x0
14	OIS4	Output Idle state 4 (OC4 output), refer to OIS1 bit	0
13	OIS3N	Output Idle state 3 (OC3N output), refer to OIS1N bit	0
12	OIS3	Output Idle state 3 (OC3 output), refer to OIS1 bit	0
11	OIS2N	Output Idle state 2 (OC2N output), refer to OIS1N bit	0
10	OIS2	Output Idle state 2 (OC2 output), refer to OIS1 bit	0
9	OIS1N	Output Idle state 1 (OC1N output) 0: OC1N=0 after a dead-time when MOE=0 1: OC1N=1 after a dead-time when MOE=0 Note: This bit cannot be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIM1_BDTR register).	0
8	OIS1	Output Idle state 1 (OC1 output) 0: OC1=0 (after a dead-time if OC1N is implemented) when MOE=0 1: OC1=1 (after a dead-time if OC1N is implemented) when MOE=0 Note: This bit cannot be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIM1_BDTR register).	0
7	TI1S	TI1 selection 0: The TIM1_CH1 pin is connected to TI1 input 1: The TIM1_CH1, CH2 and CH3 pins are connected to the TI1 input (XOR combination)	0
6:4	MMS[1:0]	Master mode selection These bits allow to select the information to be sent in master mode to slave timers for synchronization (TRGO). The combination is as follows: 000: Reset - the UG bit from the TIM1_EGR register is used as trigger output (TRGO). If the reset is generated by the trigger input (slave mode controller configured in reset mode) then the signal on TRGO is delayed compared to the actual reset. 001: Enable - the Counter Enable signal CNT_EN is used as trigger output (TRGO). It is useful to start several timers at the same time or to control a window in which a slave timer is enable. The Counter Enable signal is generated by a logic OR between CEN control bit and the trigger input when configured in gated mode. When the Counter Enable signal is controlled by the trigger input, there is a delay on TRGO, except if the master/slave mode is selected (refer to the MSM bit description in TIM1_SMCR register). 010: Update - The update event is selected as trigger output (TRGO). For instance a master timer can then be used as a prescaler for a slave timer. 011: Compare Pulse - The trigger output send a positive pulse when the CC1IF flag is to be set (even if it was already high), as soon as a capture or a compare match occurred. (TRGO). 100: Compare - OC1REF signal is used as trigger output (TRGO) 101: Compare - OC2REF signal is used as trigger output (TRGO) 110: Compare - OC3REF signal is used as trigger output (TRGO) 111: Compare - OC4REF signal is used as trigger output (TRGO)	0x0
3	CCDS	Capture/compare DMA selection 0: CCx DMA request sent when CCx event occurs 1: CCx DMA requests sent when update event occurs	0
2	CCUS	Capture/compare control update selection 0: When capture/compare control bits are preloaded (CCPC=1), they are updated by setting the COMG bit only 1: When capture/compare control bits are preloaded (CCPC=1), they are updated by setting the COMG bit or when an rising edge occurs on TRGI Note: This bit acts only on channels that have a complementary output.	0

1	--	Reserved, set to 0.	0
0	CCPC	Capture/compare preloaded control 0: CCxE, CCxNE and OCxM bits are not preloaded 1: CCxE, CCxNE and OCxM bits are preloaded, after having been written, they are updated only when a communication event (COM) occurs. Note: This bit acts only on channels that have a complementary output.	0

6.5.3 TIM1 Slave Mode Control Register (SMCR)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15	ETR	External trigger polarity This bit selects whether ETR or ETRF is used for trigger operations 0: ETR is non-inverted, active at high level or rising edge. 1: ETR is inverted, active at low level or falling edge.	0
14	ECE	External clock enable, this bit enables External clock mode 2. 0: External clock mode 2 disabled 1: External clock mode 2 enabled. The counter is clocked by any active edge on the ETRF signal.	0
13:12	ETPS[1:0]	External trigger prescaler External trigger signal ETRP frequency must be at most 1/4 of TIM1CLK frequency. A prescaler can be enabled to reduce ETRP frequency. It is useful when inputting fast external clocks. 00: Prescaler OFF 01: ETRP frequency divided by 2 10: ETRP frequency divided by 4 11: ETRP frequency divided by 8	0x0
11:8	ETF[3:0]	External trigger filter This bit-field then defines the frequency used to sample ETRP signal and the length of the digital filter applied to ETRP. The digital filter is made of an event counter in which N consecutive events are needed to validate a transition on the output: 0000: No filter, sampling is done at f_{DTS} 0001: $f_{SAMPLING}=f_{CK\_INT}$, $N=2$ 0010: $f_{SAMPLING}=f_{CK\_INT}$, $N=4$ 0011: $f_{SAMPLING}=f_{CK\_INT}$, $N=8$ 0100: $f_{SAMPLING}=f_{DTS}/2$, $N=6$ 0101: $f_{SAMPLING}=f_{DTS}/2$, $N=8$ 0110: $f_{SAMPLING}=f_{DTS}/4$, $N=6$ 0111: $f_{SAMPLING}=f_{DTS}/4$, $N=8$ 1000: $f_{SAMPLING}=f_{DTS}/8$, $N=6$ 1001: $f_{SAMPLING}=f_{DTS}/8$, $N=8$ 1010: $f_{SAMPLING}=f_{DTS}/16$, $N=5$ 1011: $f_{SAMPLING}=f_{DTS}/16$, $N=6$ 1100: $f_{SAMPLING}=f_{DTS}/16$, $N=8$ 1101: $f_{SAMPLING}=f_{DTS}/32$, $N=5$ 1110: $f_{SAMPLING}=f_{DTS}/32$, $N=6$ 1111: $f_{SAMPLING}=f_{DTS}/32$, $N=8$	0x0
7	MSM	Master/slave mode 0: No action 1: The effect of an event on the trigger input (TRGI) is delayed to allow a perfect synchronization between the current timer and its slaves (through TRGO). It is useful if we want to synchronize several timers on a single external event.	0
6:4	TS[2:0]	Trigger selection, this bit-field selects the trigger input to be used to synchronize the counter. 000: Reserved(ITR0) 001: Reserved(ITR1) 010: TIM3 (ITR2) 011: TIM17 (ITR3) 100: TI1 Edge Detector (TI1F_ED) 101: Filtered Timer Input 1 (TI1FP1) 110: Filtered Timer Input 2 (TI2FP2) 111: External Trigger input (ETRF) Note: These bits must be changed only when they are not used to avoid wrong edge detections at the transition.	0x0
3	OCCS	OCREF clear selection, this bit is used to select the OCREF clear source. 0: OCREF_CLR_INT is connected to the	0

		OCREF_CLR input 1: OCREF_CLR_INT is connected to ETRF	
2:0	SMS	Slave mode selection When external signals are selected the active edge of the trigger signal (TRGI) is linked to the polarity selected on the external input (refer to Input Control register and Control Register description). 000: Slave mode disabled - if CEN = '1' then the prescaler is clocked directly by the internal clock. 001: Encoder mode 1 - Counter counts up/down on TI1FP1 edge depending on TI2FP2 level. 010: Encoder mode 2 - Counter counts up/down on TI2FP2 edge depending on TI1FP1 level. 011: Encoder mode 3 - A combination of Mode 1 and Mode 2. 100: Reset Mode - Rising edge of the selected trigger input (TRGI) reinitializes the counter and generates an update of the registers. 101: Gated Mode - The counter clock is enabled when the trigger input (TRGI) is high. The counter stops (but is not reset) as soon as the trigger becomes low. Both start and stop of the counter are controlled. 110: Trigger Mode - The counter starts at a rising edge of the trigger TRGI (but it is not reset). Only the start of the counter is controlled. 111: External Clock Mode 1 - Rising edges of the selected trigger (TRGI) clock the counter. Note: The gated mode must not be used if TI1F_EN is selected as the trigger input (TS='100'). Indeed, TI1F_ED outputs 1 pulse for each transition on TI1F, whereas the gated mode checks the level of the trigger signal.	0x0

Table 7-2 TIMx internal trigger connection

Slave TIM	ITRO(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM1	Reserved	Reserved	TIM3	TIM17

6.5.4 TIM1 DMA/Interrupt Enable Register (TIM1_DIER)

Bit	Symbol	Description	Reset value
31:15	--	Reserved, set to 0.	0x0
14	TDE	Trigger DMA request enable 0: Trigger DMA request disabled 1: Trigger DMA request enabled	0
13	COMDE	COM DMA request enable 0: COM DMA request disabled 1: COM DMA request enabled	0
12	CC4DE	Capture/Compare 4 DMA request enable 0: CC4 DMA request disabled 1: CC4 DMA request enabled	0
11	CC3DE	Capture/Compare 3 DMA request enable 0: CC3 DMA request disabled 1: CC3 DMA request enabled	0
10	CC2DE	Capture/Compare 2 DMA request enable 0: CC2 DMA request disabled 1: CC2 DMA request enabled	0
9	CC1DE	Capture/Compare 1 DMA request enable 0: CC1 DMA request disabled 1: CC1 DMA request enabled	0
8	UDE	Update DMA request enable 0: Update DMA request disabled 1: Update DMA request enabled	0
7	BIE	Break interrupt enable 0: Break interrupt disabled 1: Break interrupt enabled	0
6	TIE	Trigger interrupt enable 0: Trigger interrupt disabled 1: Trigger interrupt enabled	0
5	COMIE	COM interrupt enable 0: COM interrupt disabled 1: COM interrupt enabled	0
4	CC4IE	CC4IE: Capture/Compare 4 interrupt enable 0: CC4 interrupt disabled 1: CC4 interrupt enabled	0
3	CC3IE	CC3IE: Capture/Compare 3 interrupt enable 0: CC3 interrupt disabled 1: CC3 interrupt enabled	0
2	CC2IE	CC2IE: Capture/Compare 2 interrupt enable 0: CC2 interrupt disabled 1: CC2 interrupt enabled	0
1	CC1IE	CC1IE: Capture/Compare 1 interrupt enable 0: CC1 interrupt disabled 1: CC1 interrupt enabled	0
0	UIE	Update interrupt enable 0: Update interrupt disabled 1: Update interrupt enabled	0

6.5.5 TIM1 Status Register (TIM1_SR)

Bit	Symbol	Description	Reset value
31:13	--	Reserved, set to 0.	0x0
12	CC4OF	Capture/Compare 4 overcapture flag refer to CC1OF description	0
11	CC3OF	Capture/Compare 3 overcapture flag refer to CC1OF description	0
10	CC2OF	Capture/Compare 2 overcapture flag refer to CC1OF description	0
9	CC1OF	Capture/Compare 1 overcapture flag This flag is set by hardware only when the corresponding channel is configured in input capture mode. It is cleared by software by writing it to '0'. 0: No overcapture has been detected. 1: The counter value has been captured in TIM1_CCR1 register while CC1IF flag was already set	0
8	--	Reserved, set to 0.	0
7	BIF	Break interrupt flag This flag is set by hardware as soon as the break input goes active. It can be cleared by software if the break input is not active. 0: No break event occurred. 1: An active level has been detected on the break input.	0
6	TIF	Trigger interrupt flag This flag is set by hardware on trigger event (active edge detected on TRGI input when the slave mode controller is enabled in all modes but gated mode. It is cleared by software. 0: No trigger event occurred. 1: Trigger interrupt pending.	0
5	COMIF	COM interrupt flag This flag is set by hardware on COM event (when Capture/compare Control bits - CCxE, CCxNE, OCxM - have been updated). It is cleared by software by writing it to '0'. 0: No COM event occurred. 1: COM interrupt pending.	0
4	CC4IF	CC4IF: Capture/Compare 4 interrupt flag refer to CC1IF description	0
3	CC3IF	CC3IF: Capture/Compare 3 interrupt flag refer to CC1IF description	0
2	CC2IF	CC2IF: Capture/Compare 2 interrupt flag refer to CC1IF description	0
1	CC1IF	If channel CC1 is configured as output: This flag is set by hardware when the counter matches the compare value, with some exception in center-aligned mode (refer to the CMS bits in the TIM1_CR1 register description). It is cleared by software. 0: No match. 1: The content of the counter TIM1_CNT matches the content of the TIM1_CCR1 register. If channel CC1 is configured as input: This bit is set by hardware on a capture. It is cleared by software or by reading the TIM1_CCR1 register. 0: No input capture occurred 1: The counter value has been captured in TIM1_CCR1 register (An edge has been detected on IC1 which matches the selected polarity)	0
0	UIF	Update interrupt flag This bit is set by hardware on an update event. It is cleared by software. 0: No update occurred. 1: Update interrupt pending. This bit is set by hardware when the registers are updated:	0

		- At overrun or underrun regarding the repetition counter value (update if repetition counter = 0) and if the UDIS=0 in the TIM1_CR1 register. - When CNT is reinitialized by software using the UG bit in TIM1_EGR register, if URS=0 and UDIS=0 in the TIM1_CR1 register. - When CNT is reinitialized by a trigger event (TIM1 slave mode control register (TIM1_SMCR)), if URS=0 and UDIS=0 in the TIM1_CR1 register.	

6.5.6 TIM1 Event Generation Register (TIM1_EGR)

Bit	Symbol	Description	Reset value
31x:8	--	Reserved, set to 0.	0x0
7	BG	Break generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0: No action 1: A break event is generated. MOE bit is cleared and BIF flag is set. Related interrupt or DMA transfer can occur if enabled.	0
6	TG	Trigger generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0: No action 1: The TIF flag is set in TIM1_SR register. Related interrupt or DMA transfer can occur if enabled.	0
5	COMG	Capture/Compare control update generation This bit can be set by software, it is automatically cleared by hardware. 0: No action 1: When CCPC bit is set, it allows to update CCxE, CCxNE and OCxM bits Note: This bit acts only on channels having a complementary output.	0
4	CC4G	CC4G: Capture/Compare 4 generation Refer to CC1G description	0
3	CC3G	CC3G: Capture/Compare 3 generation Refer to CC1G description	0
2	CC2G	CC2G: Capture/Compare 2 generation Refer to CC1G description	0
1	CC1G	CC1G: Capture/Compare 1 generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0: No action 1: A capture/compare event is generated on channel 1 If channel CC1 is configured as output: CC1IF flag is set, Corresponding interrupt or DMA request is sent if enabled. If channel CC1 is configured as input: The current value of the counter is captured in TIM1_CCR1 register. The CC1IF flag is set, the corresponding interrupt or DMA request is sent if enabled. The CC1OF flag is set if the CC1IF flag was already high.	0
0	UG	Update generation This bit can be set by software, it is automatically cleared by hardware. 0: No action 1: Reinitialize the counter and generates an update of the registers. Note that the prescaler counter is cleared too (anyway the prescaler ratio is not affected). The counter is cleared if the center-aligned mode is selected or if DIR=0 (upcounting), else it takes the auto-reload value (TIM1_ARR) if DIR=1 (downcounting).	0

6.5.7 TIM1 Capture/Compare Mode Register 1 (TIM1_CCMR1)

The channels can be used in input (capture mode) or in output (compare mode). The direction of a channel is defined by configuring the corresponding CCxS bits. All the other bits of this register have a different function in input and in output mode. For a given bit, OCxx describes its function when the channel is configured in output, ICxx describes its function when the channel is configured in input. So one must take care that the same bit can have a different meaning for the input stage and for the output stage.

Output compare mode

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15	OC2CE	Output Compare 2 clear enable, refer to OC1CE description	0
14:12	OC2M[2:0]	Output Compare 2 mode, refer to OC1M description	0x0
11	OC2PE	Output Compare 2 preload enable, refer to OC1PE description	0
10	OC2FE	Output Compare 2 fast enable, refer to OC1FE description	0
9:8	CC2S[1:0]	Capture/Compare 2 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC2 channel is configured as output 01: CC2 channel is configured as input, IC2 is mapped on TI2 10: CC2 channel is configured as input, IC2 is mapped on TI1 11: CC2 channel is configured as input, IC2 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIM1_SMCR register) Note: CC2S bits are writable only when the channel is OFF (CC2E = '0' in TIM1_CCER).	0x0
7	OC1CE	Output Compare 1 clear enable 0: OC1REF is not affected by the ETRF Input 1: OC1Ref is cleared as soon as a High level is detected on ETRF input	0
6:4	OC1M	These bits define the behavior of the output reference signal OC1REF from which OC1 and OC1N are derived. OC1REF is active high whereas OC1 and OC1N active level depends on CC1P and CC1NP bits. 000: Frozen - The comparison between the output compare register TIM1_CCR1 and the counter TIM1_CNT has no effect on the outputs. 001: Set channel 1 to active level on match. OC1REF signal is forced high when the counter TIM1_CNT matches the capture/compare register 1 (TIM1_CCR1). 010: Set channel 1 to inactive level on match. OC1REF signal is forced low when the counter TIM1_CNT matches the capture/compare register 1 (TIM1_CCR1). 011: Toggle - OC1REF toggles when TIM1_CNT=TIM1_CCR1. 100: Force inactive level - OC1REF is forced low. 101: Force active level - OC1REF is forced high. 110: PWM mode 1 - In upcounting, channel 1 is active as long as TIM1_CNT<TIM1_CCR1 else inactive. In downcounting, channel 1 is inactive (OC1REF='0') as long as TIM1_CNT>TIM1_CCR1 else active (OC1REF='1'). 111: PWM mode 2 - In upcounting, channel 1 is inactive as long as TIM1_CNT<TIM1_CCR1 else active. In downcounting, channel 1 is active as long as TIM1_CNT>TIM1_CCR1 else inactive. Note 1: These bits can not be modified as long as LOCK level 3 has been programmed (LOCK bits in TIM1_BDTR register) and CC1S='00' (the channel is configured in output). Note 2: In PWM mode 1 or 2, the OC1REF level changes only when the result of the comparison changes or when the output compare mode switches from "frozen" mode to "PWM" mode.	0x0
3	OC1PE	Output Compare 1 preload enable 0: Preload register on TIM1_CCR1 disabled. TIM1_CCR1 can be written at anytime, the new value is taken in account	0

		immediately. 1: Preload register on TIM1_CCR1 enabled. Read/Write operations access the preload register. TIM1_CCR1 preload value is loaded in the active register at each update event. Note: These bits can not be modified as long as LOCK level 3 has been programmed (LOCK bits in TIM1_BDTR register) and CC1S= '00' (the channel is configured in output). Note 2: In single pulse mode, PWM mode can be used without confirming the preload register; otherwise, its behavior is undefined.	
2	OC1FE	Output Compare 1 fast enable This bit is used to accelerate the effect of an event on the trigger in input on the CC output. 0: CC1 behaves normally depending on counter and CCR1 values even when the trigger is ON. The minimum delay to activate CC1 output when an edge occurs on the trigger input is 5 clock cycles. 1: An active edge on the trigger input acts like a compare match on CC1 output. Then, OC is set to the compare level independently from the result of the comparison. Delay to sample the trigger input and to activate CC1 output is reduced to 3 clock cycles. OCFE acts only if the channel is configured in PWM1 or PWM2 mode.	0
1:0	CC1S	Capture/Compare 1 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC1 channel is configured as output 01: CC1 channel is configured as input, IC1 is mapped on TI1 10: CC1 channel is configured as input, IC1 is mapped on TI2 11: CC1 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS bit (TIM1_SMCR register) Note: CC1S bits are writable only when the channel is OFF (CC1E = '0' in TIM1_CCER).	0x0

Input capture mode

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:12	IC2F	Input capture 2 filter, refer to IC1F description	0x0
11:10	IC2PSC[1:0]	Input capture 2 prescaler, refer to IC1PSC description	0x0
9:8	CC2S	CC2S: Capture/Compare 2 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC2 channel is configured as output 01: CC2 channel is configured as input, IC2 is mapped on TI2 10: CC2 channel is configured as input, IC2 is mapped on TI1 11: CC2 channel is configured as input, IC2 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS bit (TIM1_SMCR register) Note: CC2S bits are writable only when the channel is OFF (CC2E = '0' in TIM1_CCER).	0x0
7:4	IC1F[3:0]	Input capture 1 filter This bit-field defines the frequency used to sample TI1 input and the length of the digital filter applied to TI1. The digital filter is made of an event counter in which N consecutive events are needed to validate a transition on the output: 0000: No filter, sampling is done at f_{DTS} 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5	0x0

		1011: $f_{\text{SAMPLING}} = f_{\text{DTS}}/16, N=6$ 1100: $f_{\text{SAMPLING}} = f_{\text{DTS}}/16, N=8$ 1101: $f_{\text{SAMPLING}} = f_{\text{DTS}}/32, N=5$ 1110: $f_{\text{SAMPLING}} = f_{\text{DTS}}/32, N=6$ 1111: $f_{\text{SAMPLING}} = f_{\text{DTS}}/32, N=8$	
3:2	IC1PSC	IC1PSC: Input capture 1 prescaler This bit-field defines the ratio of the prescaler acting on CC1 input (IC1). The prescaler is reset as soon as CC1E= '0' (TIM1_CCER register). 00: no prescaler, capture is done each time an edge is detected on the capture input. 01: capture is done once every 2 events 10: capture is done once every 4 events 11: capture is done once every 8 events	0x0
1:0	CC1S	Capture/Compare 1 selection 00: CC1 channel is configured as output 01: CC1 channel is configured as input, IC1 is mapped on TI1 10: CC1 channel is configured as input, IC1 is mapped on TI2 11: CC1 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS bit (TIM1_SMCR register) Note: CC1S bits are writable only when the channel is OFF (CC1E = '0' in TIM1_CCER).	0x0

6.5.8 TIM1 Capture/Compare Mode Register 1 (TIM1_CCMR2)

Output compare mode

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15	OC4CE	Output compare 4 clear enable, refer to OC1CE description	0x0
14:12	OC4M[2:0]	Output compare 4 mode, refer to OC1M description	0x0
11	OC4PE	Output compare 4 preload enable, refer to OC1PE description	0
10	OC4FE	Output compare 4 fast enable, refer to OC1FE description	0
9:8	CC4S[1:0]	Capture/Compare 4 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC4 channel is configured as output 01: CC4 channel is configured as input, IC4 is mapped on TI4 10: CC4 channel is configured as input, IC4 is mapped on TI3 11: CC4 channel is configured as input, IC4 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS bit (TIM1_SMCR register) Note: CC4S bits are writable only when the channel is OFF (CC4E = '0' in TIM1_CCER).	0x0
7	OC3CE	Output compare 3 clear enable, refer to OC1CE description	0
6:4	OC3M	Output compare 3 mode, refer to OC1M description	0x0
3	OC3PE	Output compare 3 preload enable, refer to OC1PE description	0
2	OC3FE	Output compare 3 fast enable, refer to OC1FE description	0
1:0	CC3S	Capture/Compare 3 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC3 channel is configured as output 01: CC3 channel is configured as input, IC3 is mapped on TI3 10: CC3 channel is configured as input, IC3 is mapped on TI4 11: CC3 channel is configured as input, IC3 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS bit (TIM1_SMCR register) Note: CC3S bits are writable only when the channel is OFF (CC3E = '0' in TIM1_CCER).	0x0

Input capture mode

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:12	IC4F	Input capture 4 filter, refer to IC1F description	0x0
11:10	IC4PSC	Input capture 4 prescaler, refer to IC1PSC description	0x0
9:8	CC4S[1:0]	Capture/Compare 4 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC4 channel is configured as output 01: CC4 channel is configured as input, IC4 is mapped on TI4 10: CC4 channel is configured as input, IC4 is mapped on TI3 11: CC4 channel is configured as input, IC4 is mapped on TRC. 11: This mode is working only if an internal trigger input is selected through TS bit (TIM1_SMCR register) Note: CC4S bits are writable only when the channel is OFF (CC4E = '0' in TIM1_CCER).	0x0
7:4	IC3F	Input capture 3 filter, refer to IC1F description	0x0
3:2	IC3PSC	Input capture 3 prescaler, refer to IC1PSC description	0x0
1:0	CC3S	Capture/compare 3 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC3 channel is configured as output 01: CC3 channel is configured as input, IC3 is mapped on TI3 10: CC3 channel is configured as input, IC3 is mapped on TI4 11: CC3 channel is configured as input, IC3 is mapped on TRC. 11: This mode is working only if an internal trigger input is selected through TS bit (TIM1_SMCR register) Note: CC3S bits are writable only when the channel is OFF (CC3E = '0' in TIM1_CCER).	0x0

6.5.9 TIM1 Capture/Compare Enable Register (TIM1_CCER)

Bit	Symbol	Description	Reset value
31:14	--	Reserved, set to 0.	0x0
13	CC4P	Capture/Compare 4 output polarity, refer to CC1P description	0
12	CC4E	Capture/Compare 4 output enable, refer to CC1E description	0
11	CC3NP	Capture/Compare 3 complementary output polarity, refer to CC1NP description	0
10	CC3NE	Capture/Compare 3 complementary output enable, refer to CC1NE description	0
9	CC3P	Capture/Compare 3 output polarity, refer to CC1P description	0
8	CC3E	Capture/Compare 3 output enable, refer to CC1E description	0
7	CC2NP	Capture/Compare 2 complementary output polarity, refer to CC1NP description	0
6	CC2NE	Capture/Compare 2 complementary output enable, refer to CC1NE description	0
5	CC2P	Capture/Compare 2 output polarity, refer to CC1P description	0
4	CC2E	Capture/Compare 2 output enable, refer to CC1E description	0
3	CC1NP	Capture/Compare 1 complementary output polarity 0: OC1N active high. 1: OC1N active low. Note: This bit is not writable as soon as LOCK level 2 or 3 has been programmed (LOCK bits in TIM1_BDTR register) and CC1S= "00" (the channel is configured in output).	0
2	CC1NE	Capture/Compare 1 complementary output enable 0: Off - OC1N is not active. OC1N level is then function of MOE, OSS1, OSSR, OIS1, OIS1N and CC1E bits. 1: On - OC1N signal is output on the corresponding output pin depending on MOE, OSS1, OSSR, OIS1, OIS1N and CC1E bits.	0
1	CC1P	Capture/Compare 1 output polarity CC1 channel configured as output: 0: OC1 active high 1: OC1 active low CC1 channel configured as input: This bit selection determines whether IC1 or the inverted signal of IC1 is used as the trigger or capture signal. 0: Non-inverted: Capture occurs on the rising edge of IC1; when used as an external trigger, IC1 is non-inverted. 1: Inverted: Capture occurs on the falling edge of IC1; when used as an external trigger, IC1 is inverted. Note: Once the LOCK level (LCK bit in the TIM1_BDTR register) is set to 3 or 2, this bit cannot be modified.	0
0	CC1E	Capture/Compare 1 output enable When CC1 channel is configured as output: 0: Disabled: OC1 output is disabled, so the output level of OC1 depends on the values of MOE, OSS1, OSSR, OIS1, OIS1N, and CC1NE bits. 1: Enabled: OC1 signal is output to the corresponding output pin, and its output level depends on the values of MOE, OSS1, OSSR, OIS1, OIS1N, and CC1NE bits. When CC1 channel is configured as input: This bit determines whether the counter value can be captured into input capture/compare register 1 (TIM1_CCR1). 0: Capture disabled 1: Capture enabled	0

Note: The state of the external I/O pins connected to the complementary channels OCx and OCxN depends on the states of the OCx and OCxN channels as well as the GPIO registers.

Table 6-3: Output control bits for complementary OCx and OCxN channels with interrupt feature

Control bits					Output states	
MOE bit	OSSI bit	OSSR bit	CCxE bit	CCxNE bit	OCx output state	OCxN output state
1	X	0	0	0	Output Disabled (not driven by the timer) OCx=0, OCx_EN=0	Output Disabled (not driven by the timer) OCxN=0, OCxN_EN=0
		0	1	0	Output Disabled (not driven by the timer) OCx=0, OCx_EN=0	OCxREF+Polarity OCxN=OCxREF xor CCxNP, OCxN_EN=1
		0	1	0	OCxREF+Polarity OCx=OCxREF xor CCxP, OCx_EN=1	Output Disabled (not driven by the timer) OCxN=0, OCxN_EN=0
		0	1	1	OCREF+Polarity+ dead-time OCx_EN=1	Complementary to OCREF (not OCREF)+Polarity+ dead-time OCxN_EN=1
		1	0	0	Output Disabled (not driven by the timer) OCx=0, OCx_EN=0	Output Disabled (not driven by the timer) OCxN=0, OCxN_EN=0
		1	0	1	Off-State (output enabled with inactive state) OCx=CCxP, OCx_EN=1	OCxREF+Polarity OCxN=OCxREF xor CCxNP, OCxN_EN=1
		1	1	0	OCxREF+Polarity OCx=OCxREF xor CCxP, OCx_EN=1	Off-State (output enabled with inactive state) OCxN=CCxNP, OCxN_EN=1
		1	1	1	OCREF+Polarity+ dead-time OCx_EN=1	Complementary to OCREF (not OCREF)+ Polarity + dead-time OCxN_EN=1
0	0	X	0	0	Output Disabled (not driven by the timer) OCx=CCxP, OCx_EN=0	Output Disabled (not driven by the timer) OCxN=OCxNP, OCxN_EN=0
	0		0	1	Output Disabled (not driven by the timer) Asynchronously: OCx=CCxP, OCxEN=0, OCxN=CCxNP, OCxN_EN=0 Then if the clock is present: OCx=OISx and OCxN=OISxN after a dead-time, assuming that OISx and OISxN do not correspond to OCX and OCxN both in active state.	
	0		1	0		
	0		1	1		
	1		0	0	Output Disabled (not driven by the timer) OCx=0, OCx_EN=0	Output Disabled (not driven by the timer) OCxN=0, OCxN_EN=0
	1		0	1	Off-State (output enabled with inactive state) Asynchronously: OCx=CCxP, OCxEN=1, OCxN=CCxNP, OCxN_EN=1 Then if the clock is present: OCx=OISx and OCxN=OISxN after a dead-time, assuming that OISx and OISxN do not correspond to OCX and OCxN both in active state.	
	1		1	0		
	1		1	1		

6.5.10 TIM1 Counter (TIM1_CNT)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	CNT[15:0]	Counter value	0x0

6.5.11 TIM1 Prescaler (TIM1_PSC)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	PSC[15:0]	Prescaler value The counter clock frequency (CK_CNT) is equal to $f_{CK\_PSC} / (PSC[15:0] + 1)$. PSC contains the value to be loaded in the active prescaler register at each update event (including when the counter is cleared through UG bit of TIM_EGR register or through trigger controller when configured in “reset mode”).	0x0

6.5.12 TIM1 Auto-Reload Register (TIM1_ARR)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	ARR[15:0]	Auto-reload value ARR is the value to be loaded in the actual auto-reload register. The counter is blocked while the auto-reload value is null.	0xFFFF

6.5.13 TIM1 Repetition Counter Register (TIM1_RCR)

Bit	Symbol	Description	Reset value
31:8	--	Reserved, set to 0.	0x0
7:0	REP[7:0]	Repetition counter value These bits allow the user to set-up the update rate of the compare registers (i.e. periodic transfers from preload to active registers) when preload registers are enable, as well as the update interrupt generation rate, if this interrupt is enable. Each time the REP_CNT related downcounter reaches zero, an update event is generated and it restarts counting from REP value. As REP_CNT is reloaded with REP value only at the repetition update event U_RC, any write to the TIM1_RCR register is not taken in account until the next repetition update event. It means in PWM mode (REP+1) corresponds to: - the number of PWM periods in edge-aligned mode - the number of half PWM period in center-aligned mode.	

6.5.14 TIM1 Capture/Compare Register 1 (TIM1_CCR1)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	CCR1[15:0]	Capture/Compare 1 value If channel CC1 is configured as output: CCR1 is the value to be loaded in the actual capture/compare 1 register (preload value). It is loaded permanently if the preload feature is not selected in the TIM1_CCMR1 register (bit OC1PE). Else the preload value is copied in the active capture/compare 1 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIM1_CNT and signaled on OC1 output. If channel CC1 is configured as input: CCR1 is the counter value transferred by the last input capture 1 event (IC1).	0x0

6.5.15 TIM1 Capture/Compare Register 2 (TIM1_CCR2)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	CCR2[15:0]	Capture/Compare 2 value If channel CC2 is configured as output: CCR2 is the value to be loaded in the actual capture/compare 2 register (preload value). It is loaded permanently if the preload feature is not selected in the TIM1_CCMR2 register (bit OC2PE). Else the preload value is copied in the active capture/compare 2 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIM1_CNT and signalled on OC output. If channel CC2 is configured as input: CCR2 is the counter value transferred by the last input capture 2 event (IC2).	0x0

6.5.16 TIM1 Capture/Compare Register 3 (TIM1_CCR3)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	CCR3[15:0]	Capture/Compare value If channel CC3 is configured as output: CCR3 is the value to be loaded in the actual capture/compare 3 register (preload value). It is loaded permanently if the preload feature is not selected in the TIM1_CCMR3 register (bit OC3PE). Else the preload value is copied in the active capture/compare 3 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIM1_CNT and signalled on OC output. If channel CC3 is configured as input: CCR3 is the counter value transferred by the last input capture 3 event (IC3).	0x0

6.5.17 TIM1 Capture/Compare Register 4 (TIM1_CCR4)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	CCR4[15:0]	Capture/Compare 4 value If channel CC4 is configured as output: CCR4 is the value to be loaded in the actual capture/compare 4 register (preload value). It is loaded permanently if the preload feature is not selected in the TIM1_CCMR4 register (bit OC4PE). Else the preload value is copied in the active capture/compare 4 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIM1_CNT and signalled on OC output. If channel CC4 is configured as input: CCR4 is the counter value transferred by the last input capture 4 event (IC4).	0x0

6.5.18 TIM1 Brake and Dead-Time Register (TIM1_BDTR)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15	MOE	Main output enable This bit is cleared asynchronously by hardware as soon as the break input is active. It is set by software or automatically depending on the AOE bit. It is acting only on the channels which are configured in output. 0: OC and OCN outputs are disabled or forced to idle state. 1: OC and OCN outputs are enabled if their respective enable bits are set (CCxE, CCxNE in TIM1_CCER register).	0
14	AOE	Automatic output enable 0: MOE can be set only by software 1: MOE can be set by software or automatically at the next update event (if the break input is not be active) Note: This bit cannot be modified as long as LOCK level 1 has been programmed (LOCK bits in TIM1_BDTR register).	0
13	BKP	Break polarity 0: Break input BRK is active low 1: Break input BRK is active high Note: This bit can not be modified as long as LOCK level 1 has been programmed (LOCK bits in TIM1_BDTR register).	0
12	BKE	Break enable 0: Break inputs disabled 1: Break inputs enabled Note: This bit cannot be modified when LOCK level 1 has been programmed (LOCK bits in TIM1_BDTR register).	0
11	OSSR	Off-state selection for Run mode This bit is used when MOE=1 on channels having a complementary output which are configured as outputs. OSSR is not implemented if no complementary output is implemented in the timer. See OC/OCN enable description for more details (capture/compare enable register (TIM1_CCER)). 0: When inactive, OC/OCN outputs are disabled (OC/OCN enable output signal=0). 1: When inactive, OC/OCN outputs are enabled with their inactive level as soon as CCxE=1 or CCxNE=1 .then,OC/OCN enable output singal=1 Note: This bit can not be modified as soon as the LOCK level 2 has been programmed (LOCK bits in TIM1_BDTR register).	0
10	OSSI	Off-state selection for Idle mode This bit is used when MOE=0 on channels configured as outputs.	0

		See OC/OCN enable description for more details (capture/compare enable register (TIM1_CCER)). 0: When inactive, OC/OCN outputs are disabled (OC/OCN enable output signal=0). 1: When inactive, OC/OCN outputs are forced first with their idle level as soon as CCxE=1 or CCxNE=1. OC/OCN enable output signal=1) Note: This bit can not be modified as soon as the LOCK level 2 has been programmed (LOCK bits in TIM1_BDTR register).	
9:8	LOCK	Lock configuration These bits offer a write protection against software errors. 00: LOCK OFF - No bit is write protected. 01: LOCK Level 1 = DTG bits in TIM1_BDTR register, OISx and OISxN bits in TIM1_CR2 register and BKE/BKP/AOE bits in TIM1_BDTR register can no longer be written. 10: LOCK Level 2 = LOCK Level 1 + CC Polarity bits (CCxP/CCxNP bits in TIM1_CCER register, as long as the related channel is configured in output through the CCxS bits) as well as OSSR and OSSI bits can no longer be written. 11: LOCK Level 3 = LOCK Level 2 + CC Control bits (OCxM and OCxPE bits in TIM1_CCMRx registers, as long as the related channel is configured in output through the CCxS bits) can no longer be written. Note: The LOCK bits can be written only once after the reset. Once the TIM1_BDTR register has been written, their content is frozen until the next reset.	0x0
7:0	DTG	Dead-time generator setup This bit-field defines the duration of the dead-time inserted between the complementary outputs. DT correspond to this duration. DTG[7:5]=0xx => DT=DTG[7:0]*t_{dtg}, with $t_{dtg}=t_{DTS}$, DTG[7:5]=10x => DT=(64+DTG[5:0])*t_{dtg}, with $T_{dtg}=2xt_{DTS}$. DTG[7:5]=110 => DT=(32+DTG[4:0])*t_{dtg}, with $T_{dtg}=8xt_{DTS}$. DTG[7:5]=111 => DT=(32+DTG[4:0])*t_{dtg}, with $T_{dtg}=16xt_{DTS}$. Example if TDTS=125 ns (8 MHz), dead-time possible values are: 0 to 15875 ns by 125 ns steps, 16 us to 31750 ns by 250 ns steps, 32 us to 63 us by 1 us steps, 64 us to 126 us by 2 us steps Note: This bit-field can not be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIM1_BDTR register).	0x0

6.5.19 TIM1 DMA Control Register (TIM1_DCR)

Bit	Symbol	Description	Reset value
31:13	-	Reserved	0x0
12:8	DBL	DMA burst length This 5-bit vector defines the number of DMA transfers (the timer recognizes a burst transfer when a read or a write access is done to the TIM1_DMAR address) 00000: 1 transfer 00001: 2 transfers 00010: 3 transfers 00011: 4 transfers 10001: 18 transfers	0x0
7:5	-	Reserved	0x0
4:0	DBA	DMA base address This 5-bit vector defines the base-address for DMA transfers (when read/write access are done through the TIM1_DMAR address). DBA is defined as an offset starting from the address of the TIM1_CR1 register.	0x0

6.5.20 TIM1 Full Transfer DMA Address (TIM1_DMAR)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	DMAB	DMA register for burst accesses A read or write operation to the DMAR register accesses the register located at the address $(\text{TIM1\_CR1 address}) + (\text{DBA} + \text{DMA index}) * 4$ where TIM1_CR1 address is the address of the control register 1, DBA is the DMA base address configured in TIM1_DCR register, DMA index is automatically controlled by the DMA transfer, and ranges from 0 to DBL (DBL configured in TIM1_DCR).	0x0

Chapter 7 General-Purpose Timer (TIM3)

7.1 TIM3 Introduction

The general-purpose timers contain a 16-bit auto-reload counter, which is driven by a programmable prescaler.

They can be used for various purposes, including measuring the pulse width of an input signal (input capture) or generating output waveforms (output compare and PWM).

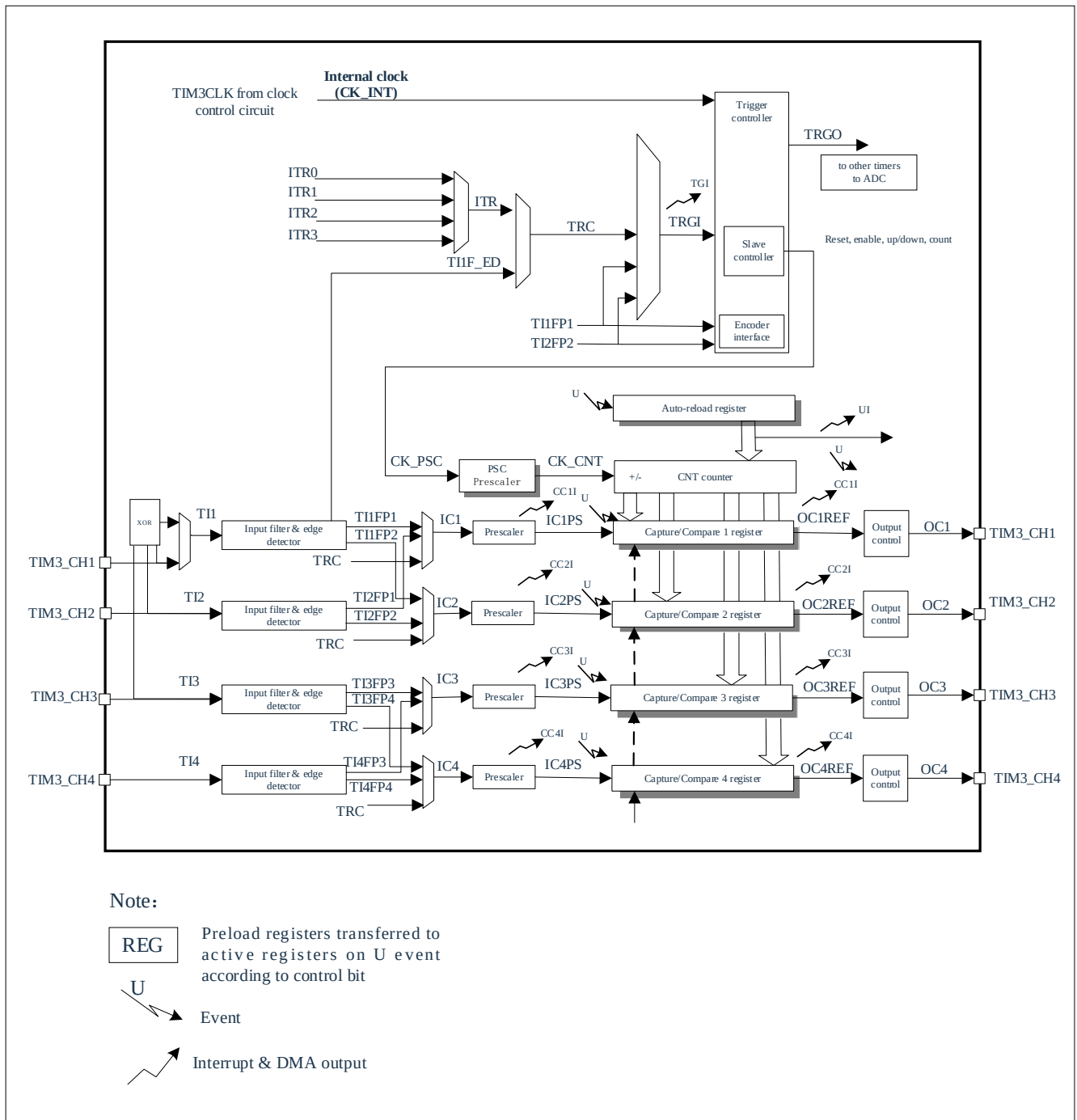
By using the timer prescaler, the pulse width and waveform period can be modulated from a few microseconds to a few milliseconds.

7.2 TIM3 Main Features

The general-purpose TIM3 timer has the following features:

- 16-bit TIM3 up, down, and up/down auto-reload counter.
- 16-bit programmable prescaler for dividing the counter clock frequency (i.e., modifying at runtime), with a prescaler value ranging from 1 to 65536.
- Up to 4 independent channels, which can be used for:
 - Input capture
 - Output compare
 - PWM generation (edge and center-aligned modes)
 - One-pulse mode output
- Synchronization circuits that allow the timer to be controlled by external signals and enable multiple timers to be interconnected.
- Interrupt/DMA requests are generated when the following events occur:
 - Update: Counter overrun/underrun, counter initialization (via software or internal/external triggers)
 - Trigger events (counter start, stop, initialize, or counting via internal/external triggers)
 - Input capture
 - Output compare
- Supports incremental (quadrature) encoder and Hall sensor circuits
- External clock trigger input or cycle-by-cycle current management

Figure 7-1 General-purpose timer block diagram (TIM3)



7.3 TIM3 Functional Description

7.3.1 Time-Base Unit

The main module of the programmable timer consists of a 16-bit counter and its associated auto-reload register. The counter can count upwards, downwards, or up/down bidirectionally. The clock for the counter can be divided by the prescaler.

The counter, auto-reload register, and prescaler register can be read and written via software. Read/write operations can be performed even while the counter is running.

The time base unit includes:

- Counter register (TIM3_CNT)
- Prescaler register (TIM3_PSC)
- Auto-reload register (TIM3_ARR)

The auto-reload register is preloaded. When writing or reading from the auto-reload register, the preload register is accessed. The content of the preload register can either be directly transferred to the shadow register or transferred on every update event (UEV), depending on the auto-reload preload enable bit (ARPE) in the TIM3_CR1 register. When the counter reaches the overrun value (or the underrun value in the case of down-counting) and the UDIS bit in the TIM3_CR1 register is 0, an update event will be triggered. This update event can also be generated by software. The generation of update events for various configurations will be detailed below.

The counter is clocked by the prescaler output CK_CNT, and the counter will only start when the counter enable bit (CEN) in the TIM3_CR1 register is set to 1 (for more details on counter enabling, refer to the related descriptions from the mode controller).

Note that the actual counter enable signal CNT_EN will be set to 1 one clock cycle after CEN is set to 1.

The prescaler can divide the counter clock frequency, with the division factor ranging from 1 to 65536. The prescaler is controlled by a 16-bit register (TIM3_PSC), which allows real-time changes. The new prescaler ratio will be applied at the next update event.

Figure 7-2 Counter timing diagram with prescaler division change from 1 to 2

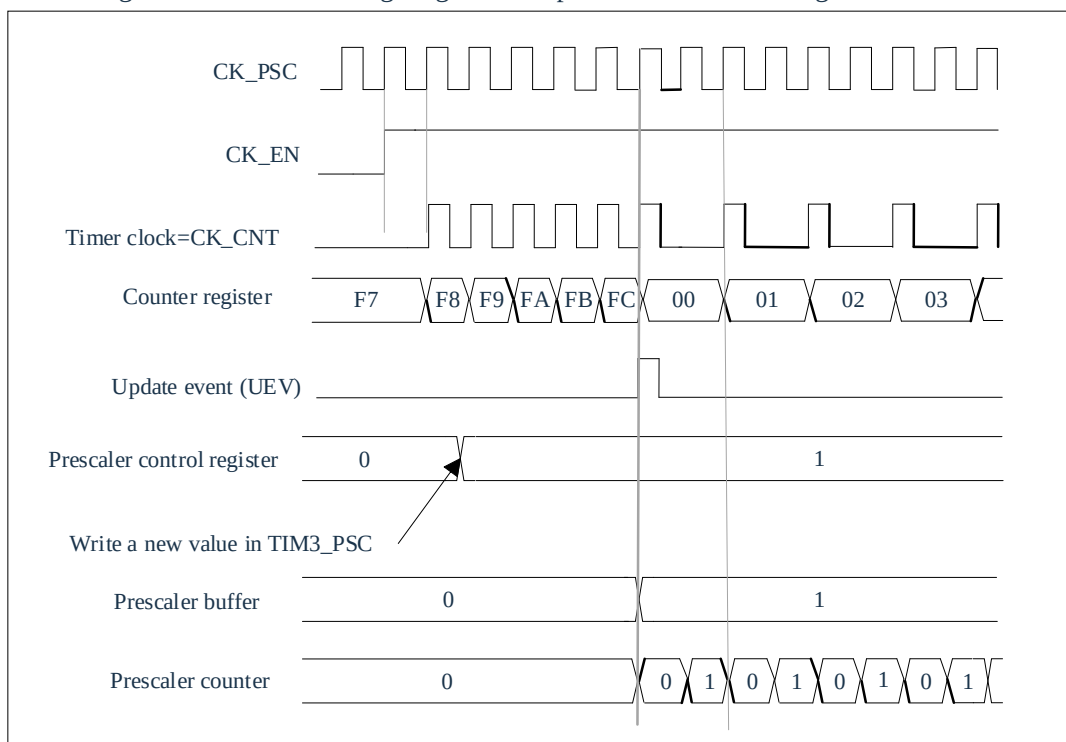
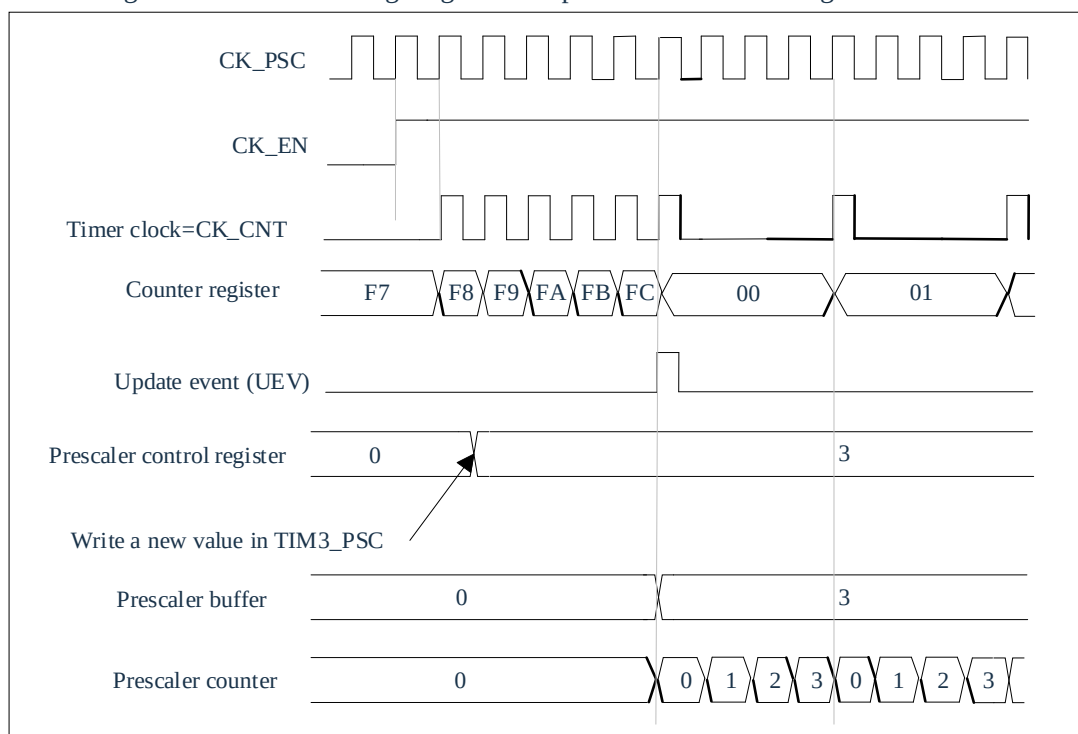


Figure 7-3 Counter timing diagram with prescaler division change from 1 to 4



7.3.2 Counter Modes

7.3.2.1 Up-Counting Mode

In the up-counter mode, the counter counts from 0 to the auto-reload value (the content of the TIM3_ARR register), and then it starts counting from 0 again, generating a counter overrun event.

Each time a counter overrun occurs, an update event will be generated, or the UG bit in the TIM3_EGR register can be set to 1 (via software or using the mode controller) to generate an update event.

By setting the UDIS bit to 1 in the TIM3_CR1 register via software, the UEV event can be disabled. This prevents the shadow register from being updated when new values are written to the preload register. No update event will occur until the UDIS bit is written to 0. However, both the counter and the prescaler counter will restart counting from 0 (while the prescaler ratio remains unchanged). Additionally, if the URS bit (update request selection) in the TIM3_CR1 register is set to 1, setting the UG bit to 1 will generate an update event (UEV) but will not set the UIF flag to 1 (thus, no interrupt or DMA request will be triggered). This ensures that if the counter is reset during a capture event, an update interrupt and capture interrupt will not be triggered simultaneously.

When an update event occurs, all registers will be updated, and the update flag (the UIF bit in the TIM3_SR register) will be set to 1 (depending on the URS bit):

- The preload value (the content of the TIM3_PSC register) will be reloaded into the prescaler's buffer.
- The auto-reload shadow register will be updated with the preload value.

The following figures illustrate the behavior of the counter at different clock frequencies when TIM3_ARR = 0x36.

Figure 7-4 Counter timing diagram, internal clock divided by 1

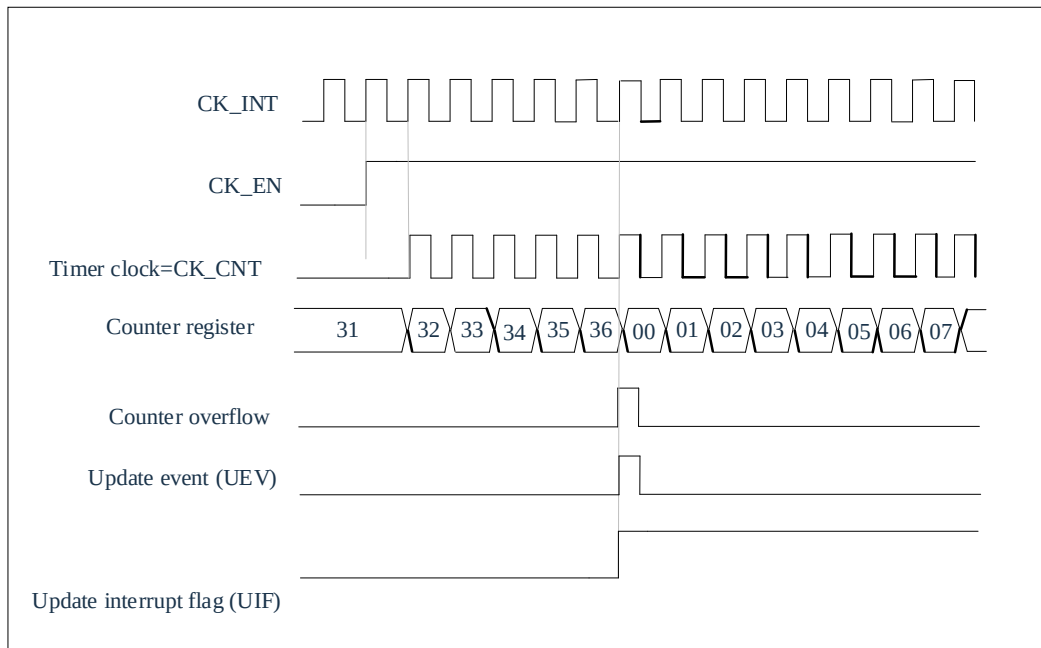


Figure 7-5 Counter timing diagram, internal clock divided by 2

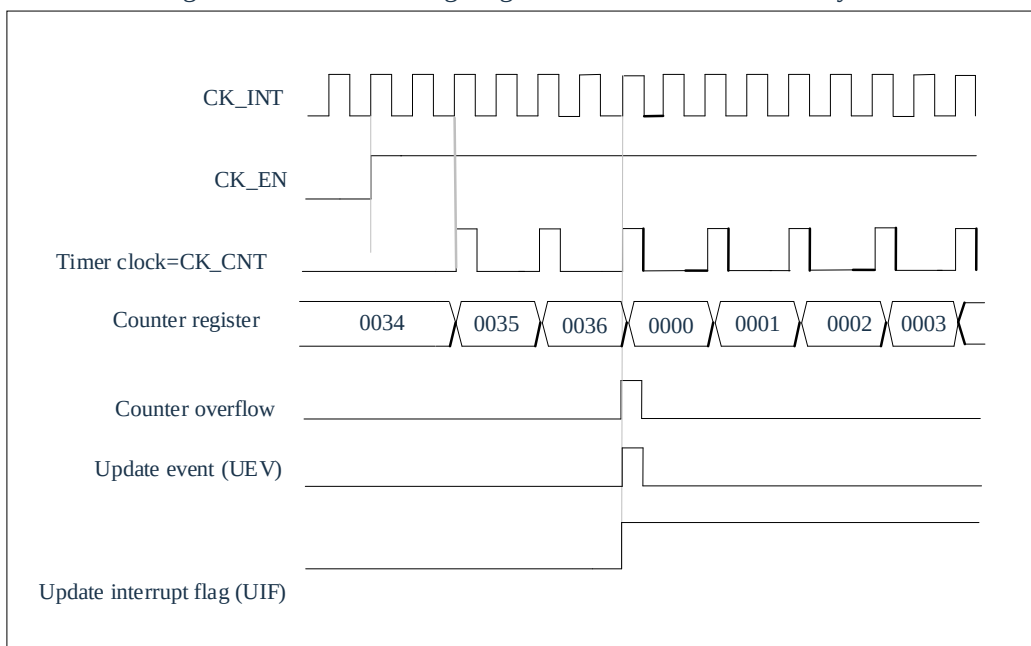


Figure 7-6 Counter timing diagram, internal clock divided by 4

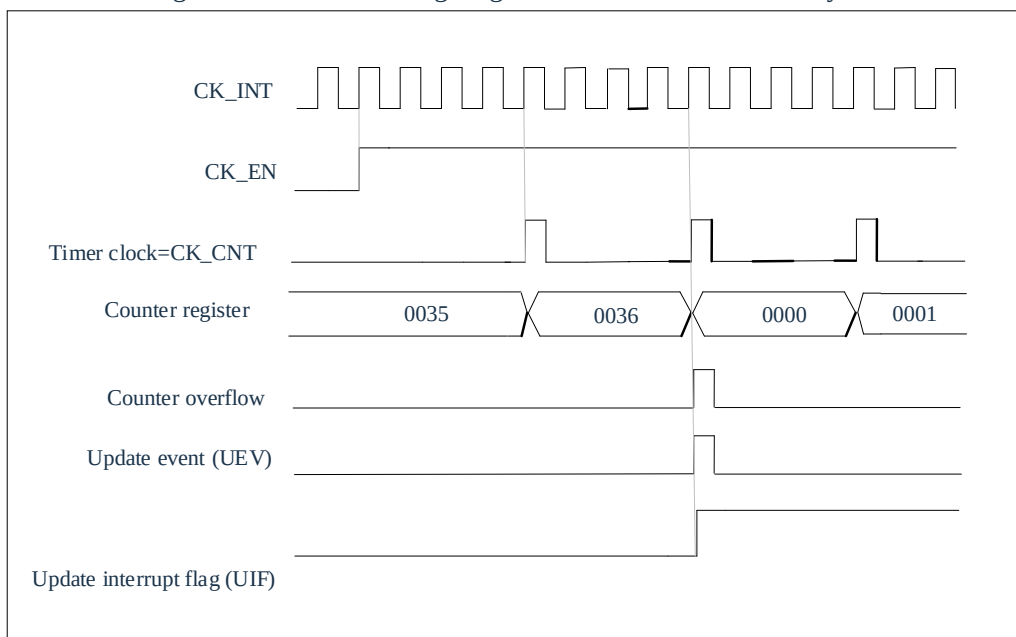


Figure 7-7 Counter timing diagram, internal clock divided by N

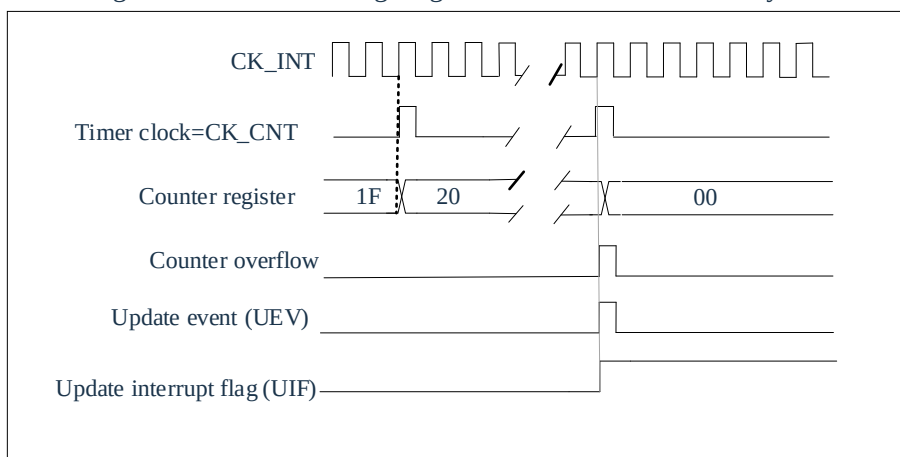


Figure 7-8 Counter timing diagram, update event when ARPE=0 (TIM3_ARR not preloaded)

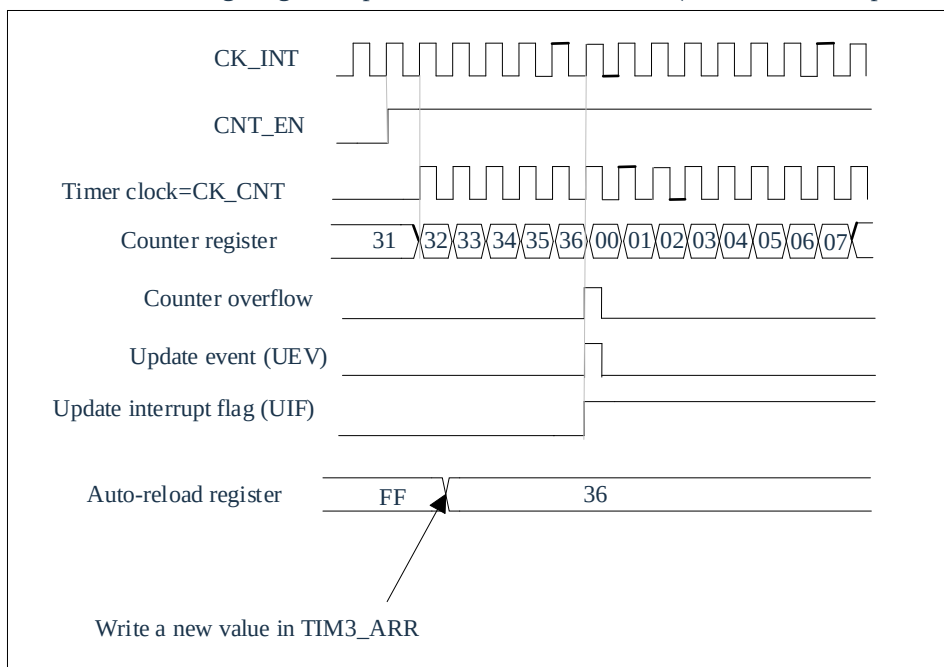
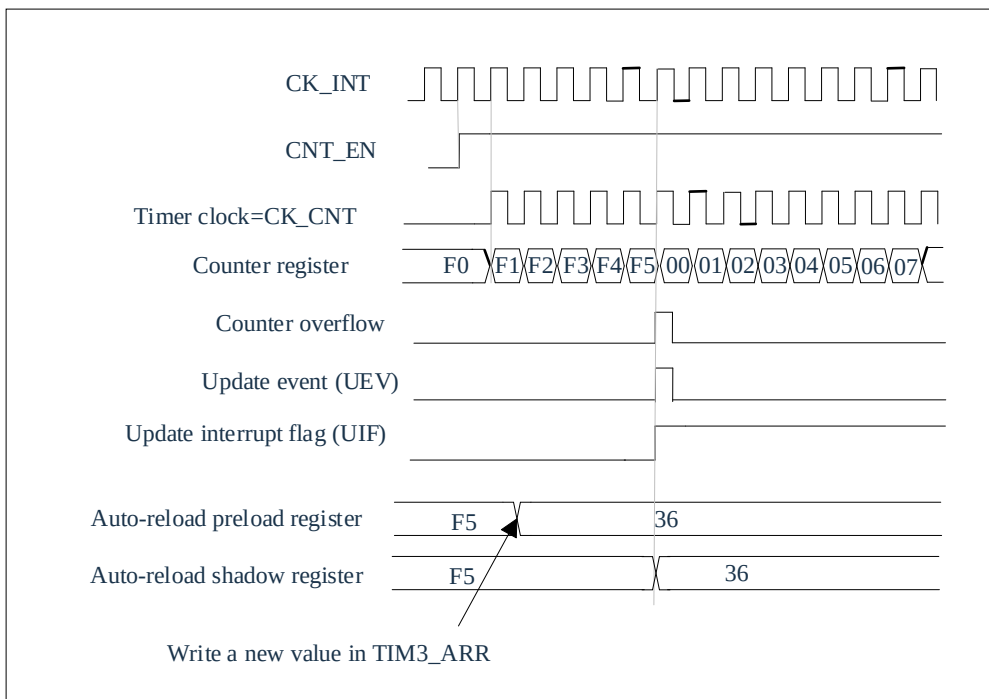


Figure 7-9 Counter timing diagram, update event when ARPE=1 (TIM3_ARR preloaded)



7.3.2.2 Down-Counting Mode

In down-counting mode, the counter counts down from the auto-reload value to 0, then restarts counting from the auto-reload value and generates a counter underrun event. Every time a counter underrun occurs, an update event is generated, or the UG bit in the TIM3_EGR register can be set to 1 (either by software or using the mode controller) to generate an update event.

By setting the UDIS bit to 1 in the TIM3_CR1 register through software, the UEV update event can be disabled. This prevents updates to the shadow register when new values are written to the preload register. No update event will occur until the UDIS bit is written to 0. However, the counter will restart counting from the current auto-reload value, and the prescaler counter will restart counting from 0 (though the prescaler value remains unchanged).

Additionally, if the URS bit (Update Request Selection) in the TIM3_CR1 register is set to 1, setting the UG bit to 1 will generate the update event (UEV), but the UIF flag will not be set to 1 (so no interrupts or DMA requests will be triggered). This way, if the counter is cleared during a capture event, no update interrupt or capture interrupt will occur simultaneously.

When an update event occurs, all registers are updated, and the update flag (UIF bit in the TIM3_SR register) is set to 1 (depending on the URS bit):

- The preload value (content of the TIM3_PSC register) will be reloaded into the prescaler's buffer.
- The auto-reload active register will be updated with the preload value (content of the TIM3_ARR register).

Note: The auto-reload register is updated before the counter is reloaded, so the next counting cycle will use the new desired period.

Figure 7-10 Counter timing diagram, internal clock divided by 1

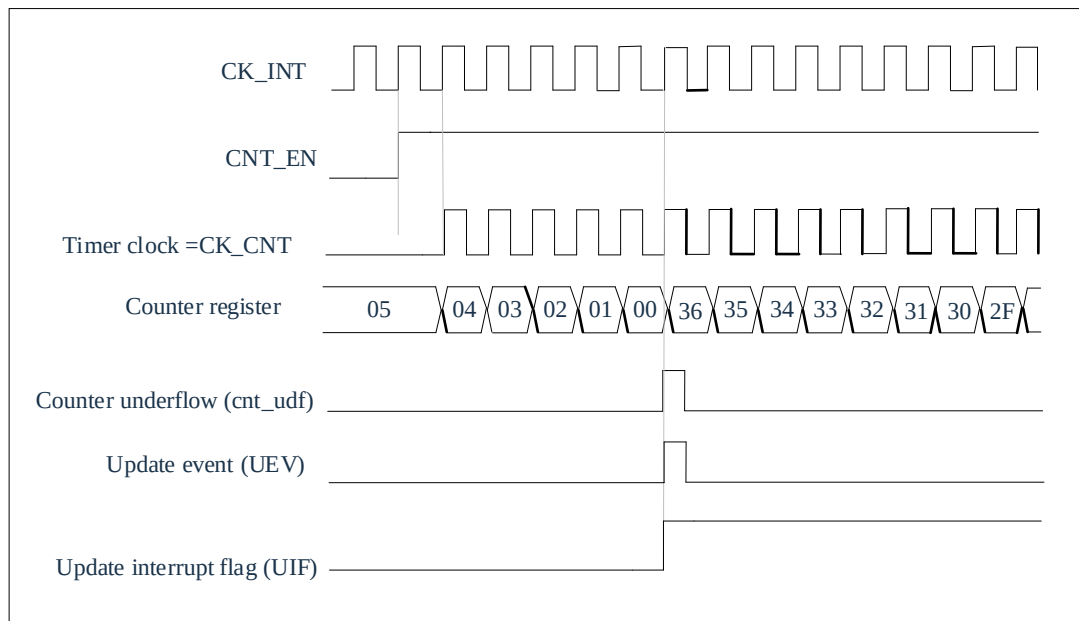


Figure 7-11 Counter timing diagram, internal clock divided by 2

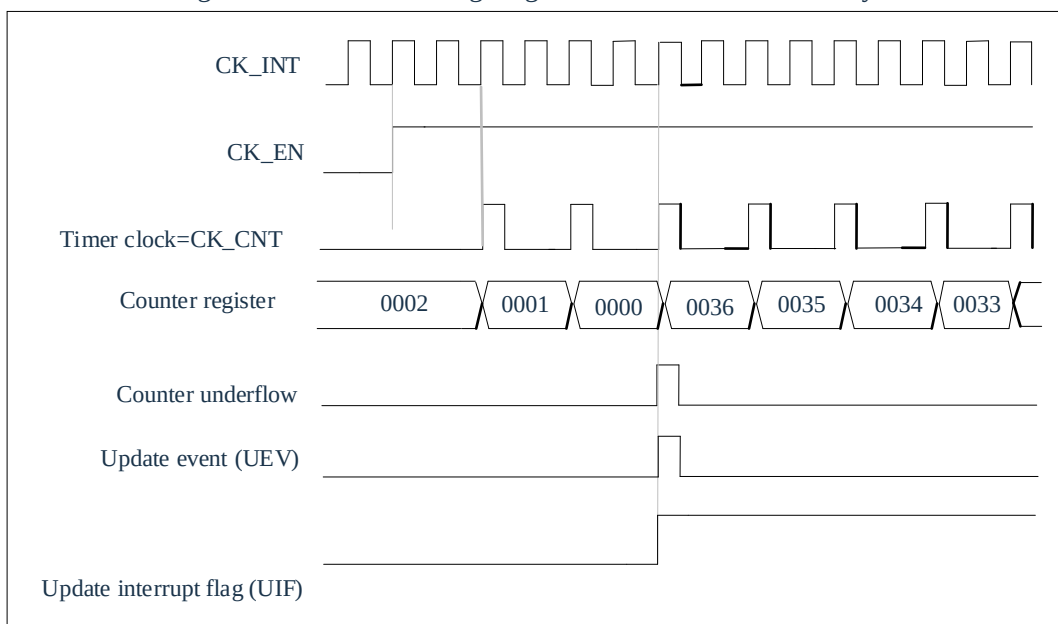


Figure 7-12 Counter timing diagram, internal clock divided by 4

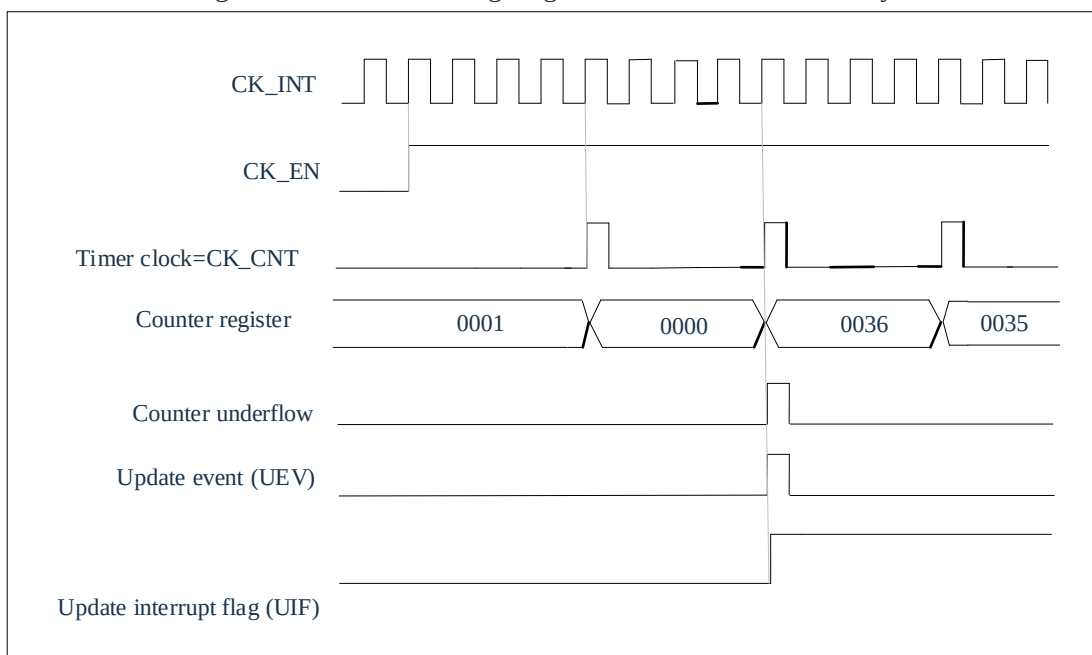


Figure 7-13 Counter timing diagram, internal clock divided by N

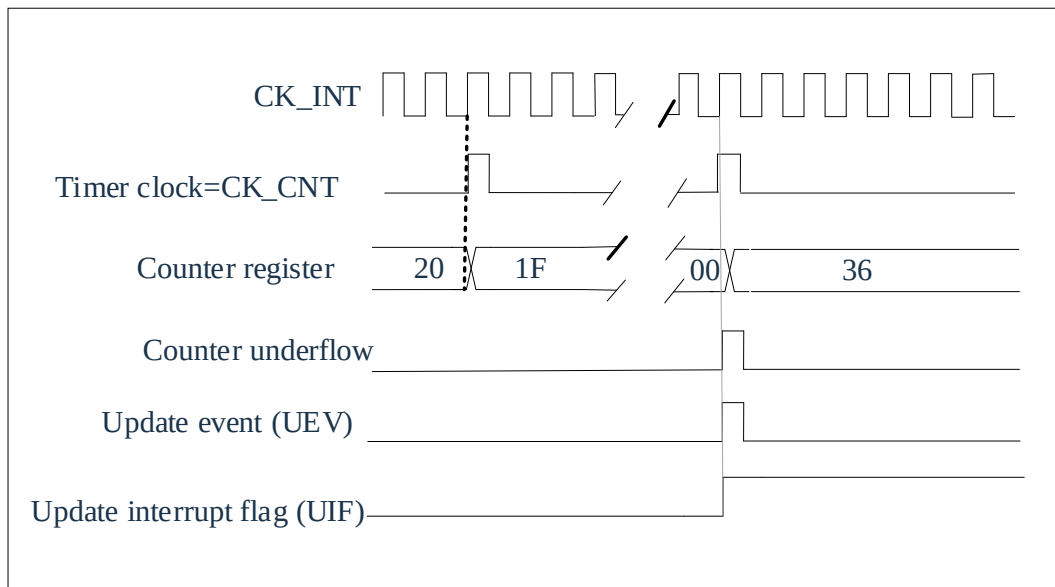
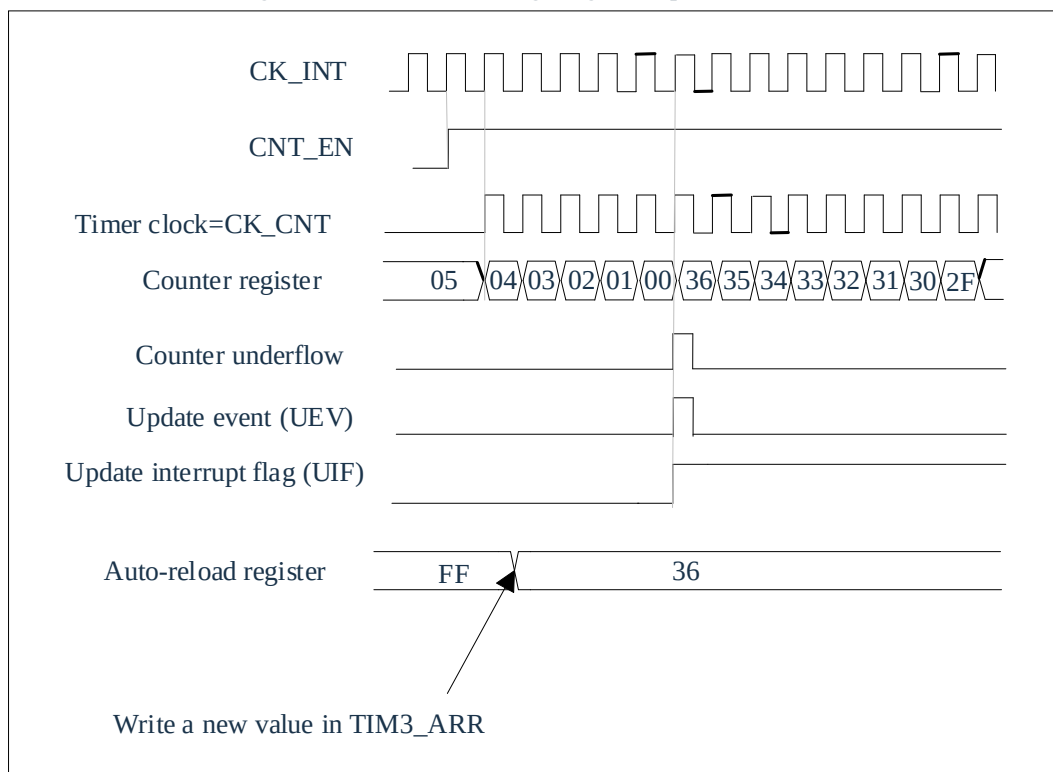


Figure 7-14 Counter timing diagram, update event



7.3.2.3 Center-Aligned Mode (Up/Down Counting)

In center-aligned mode, the counter starts from 0 and counts up to the auto-reload value (content of the TIM3_ARR register) minus 1, generating a counter overrun event. It then counts down from the auto-reload value to 1 and generates a counter underrun event. After that, it restarts counting from 0.

When the CMS bit in the TIM3_CR1 register is not “00”, center-aligned mode is enabled. When the channel is configured in output mode, its output compare interrupt flag will be set to 1 in the following modes: counter down-counting (center-aligned mode 1, CMS = “01”), counter up-counting (center-aligned mode 2, CMS = “10”), and counter up/down counting (center-aligned mode 3, CMS = “11”).

In this mode, the direction bit (DIR bit in the TIM3_CR1 register) cannot be written. It is updated by hardware to indicate the current counter direction.

An update event is generated every time a counter overrun or underrun occurs, or the UG bit in the TIM3_EGR register can be set to 1 (either by software or using the mode controller) to generate an update event. In this case, both the counter and the prescaler counter will restart from 0.

By setting the UDIS bit to 1 in the TIM3_CR1 register through software, the UEV update event can be disabled. This prevents updates to the shadow register when new values are written to the preload register. No update event will occur until the UDIS bit is written to 0. However, the counter will still count up and down according to the current auto-reload value.

Additionally, if the URS bit (Update Request Selection) in the TIM3_CR1 register is set to 1, setting the UG bit to 1 will generate the update event (UEV), but the UIF flag will not be set to 1 (so no interrupts or DMA requests will be triggered). This way, if the counter is cleared during a capture event, no update interrupt or capture interrupt will occur simultaneously.

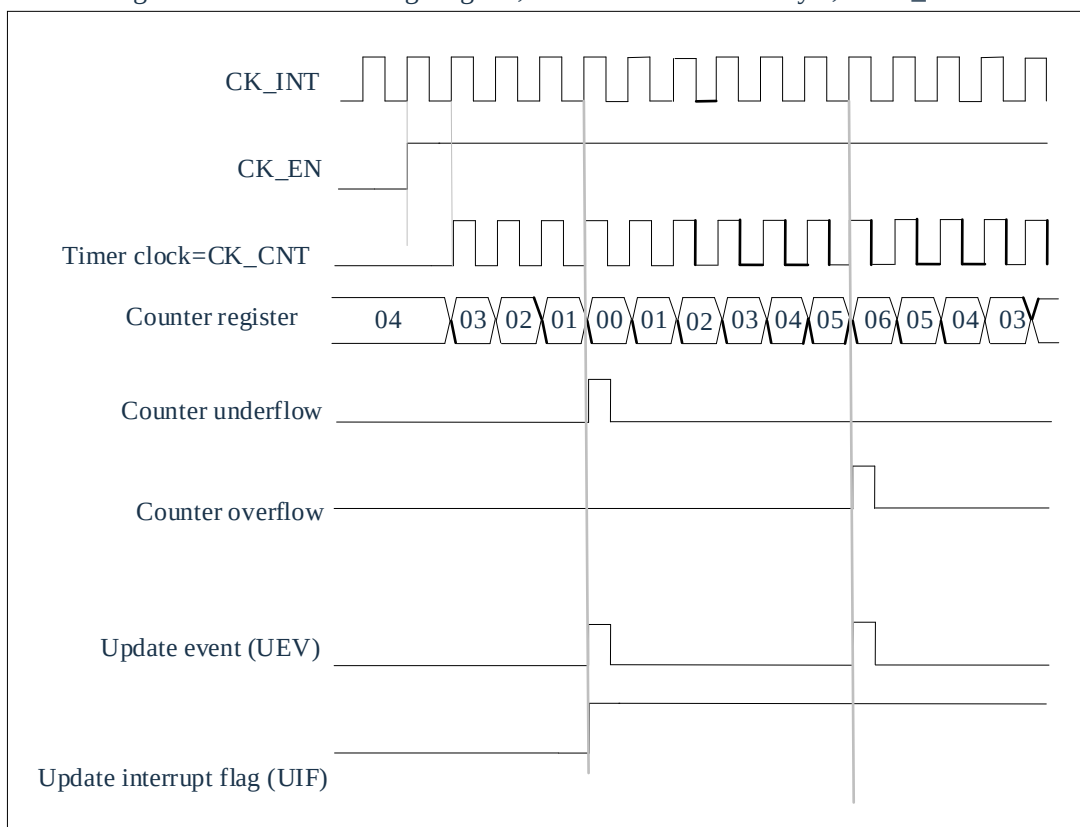
When an update event occurs, all registers are updated, and the update flag (UIF bit in the TIM3_SR register) is set to 1 (depending on the URS bit):

- The preload value (content of the TIM3_PSC register) will be reloaded into the prescaler’s buffer.
- The auto-reload active register will be updated with the preload value (content of the TIM3_ARR register).

Note: If the update operation is triggered by a counter overrun, the auto-reload register is updated before reloading the counter. Therefore, the next counting cycle will use the new desired period (the counter is reloaded with the new value).

The following figures illustrate the behavior of the counter at different clock frequencies.

Figure 7-15 Counter timing diagram, internal clock divided by 1, TIM3_ARR=0x6



Note: Center-aligned mode 1 is used.

Figure 7-16 Counter timing diagram, internal clock divided by 2

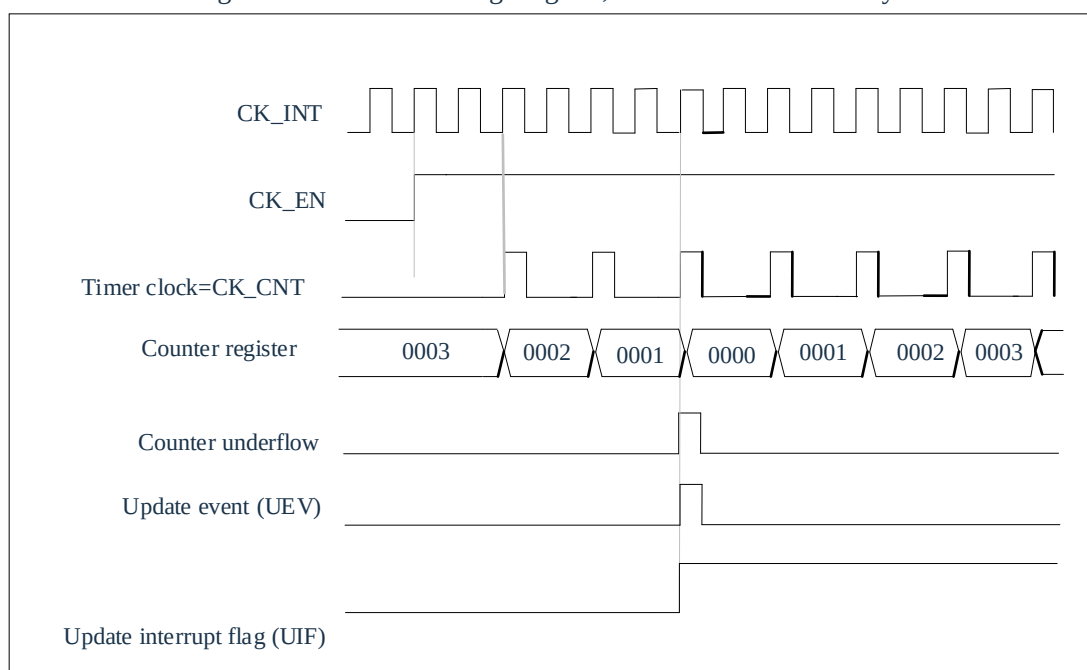
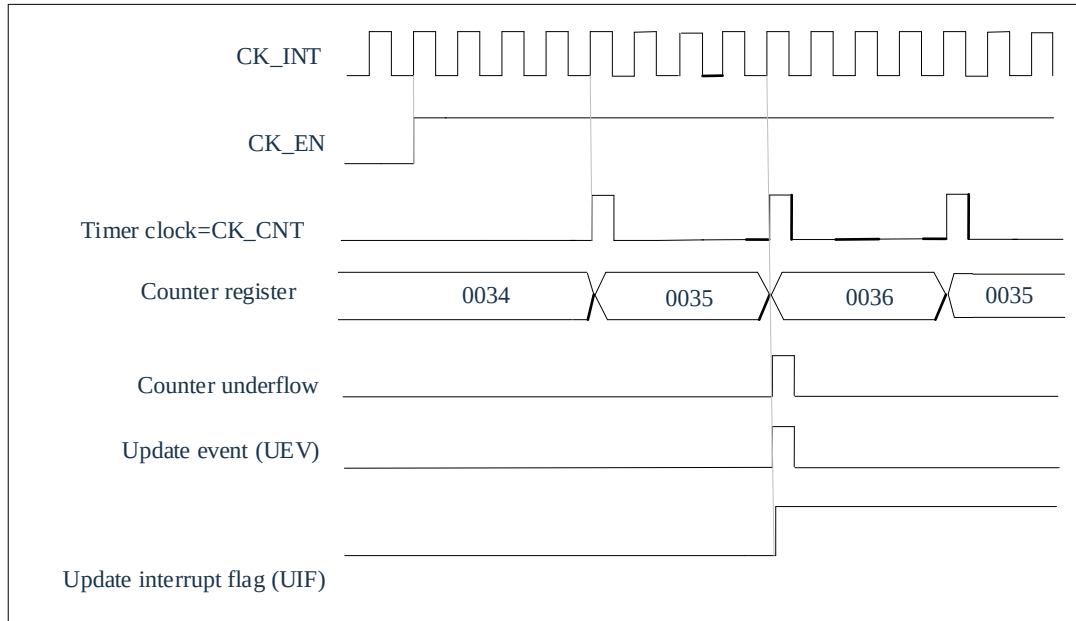


Figure 7-17 Counter timing diagram, internal clock divided by 4, TIM3_ARR=0x36



Note: Center-aligned mode 2 or 3 is used with an UIF on overrun.

Figure 7-18 Counter timing diagram, internal clock divided by N

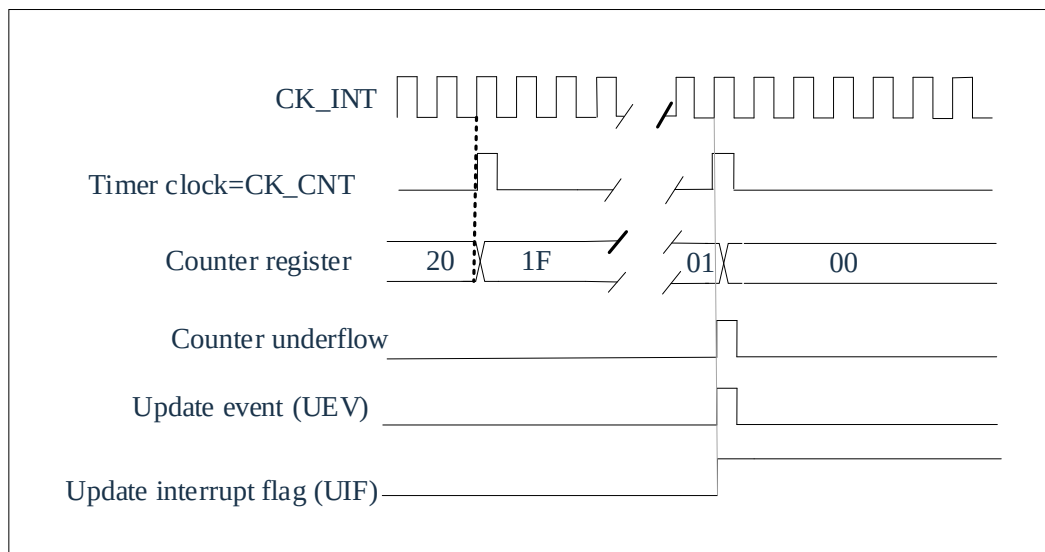


Figure 7-19 Counter timing diagram, update event with ARPE=1 (counter underrun)

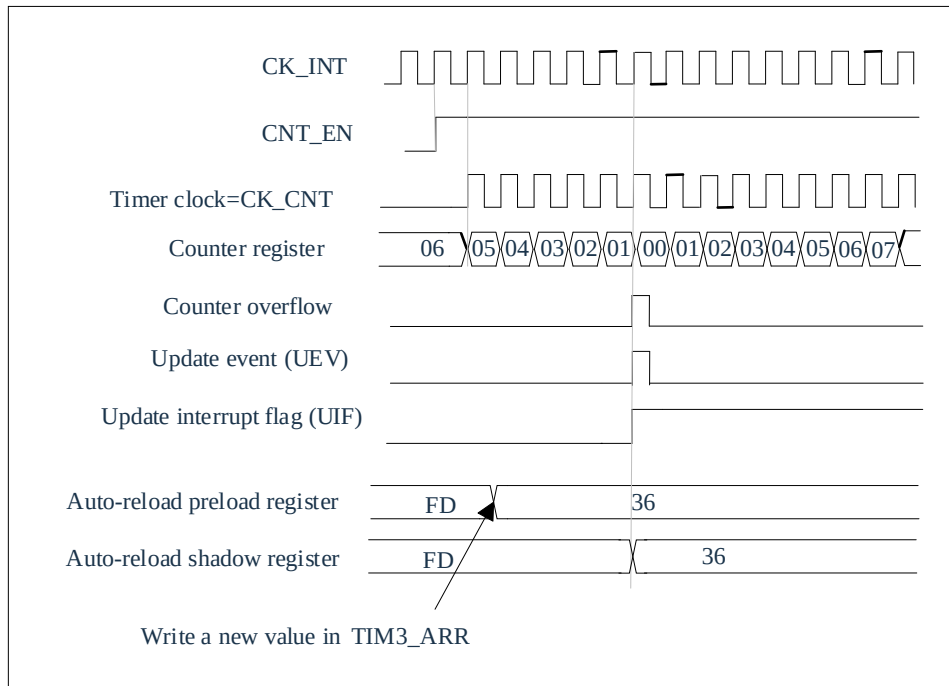
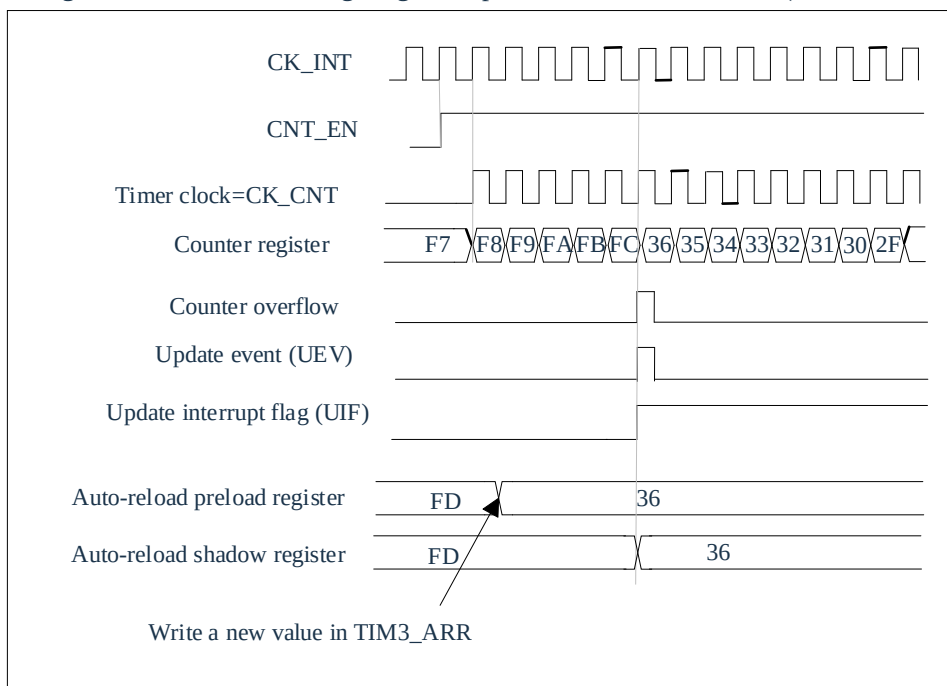


Figure 7-20 Counter timing diagram, update event with ARPE=1 (counter overrun)



7.3.3 Clock Selection

The counter clock can be provided by the following clock sources:

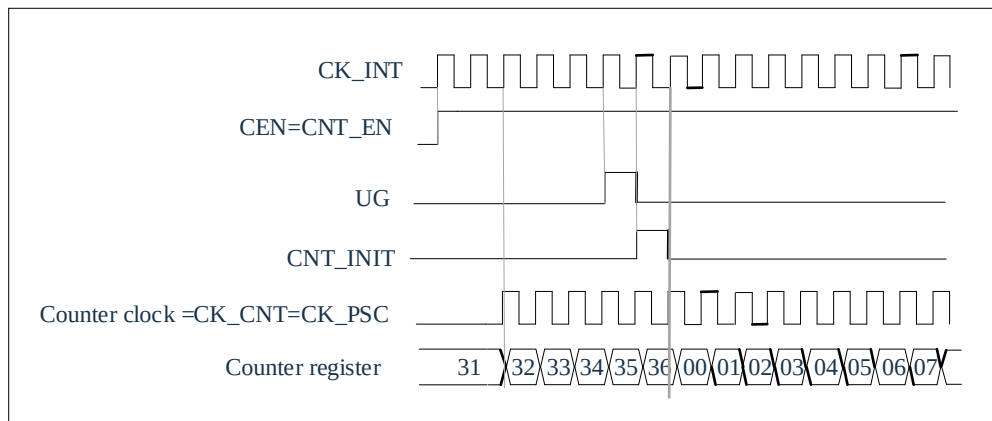
- Internal clock (CK_INT)
- External clock mode 1: External input pin
- Internal trigger input (ITRx): Using one timer as a prescaler for another timer, for example, Timer 1 can be set as the prescaler for Timer 3.

7.3.3.1 Internal Clock Source (CK_INT)

If disabled from the mode controller, the CEN bit, DIR bit (in the TIM3_CR1 register), and UG bit (in the TIM3_EGR register) are the actual control bits and can only be modified by software. When the CEN bit is written to 1, the prescaler clock is provided by the internal clock CK_INT.

Figure 7-21 shows the behavior of the control circuit and incrementing counter in normal mode (without prescaling).

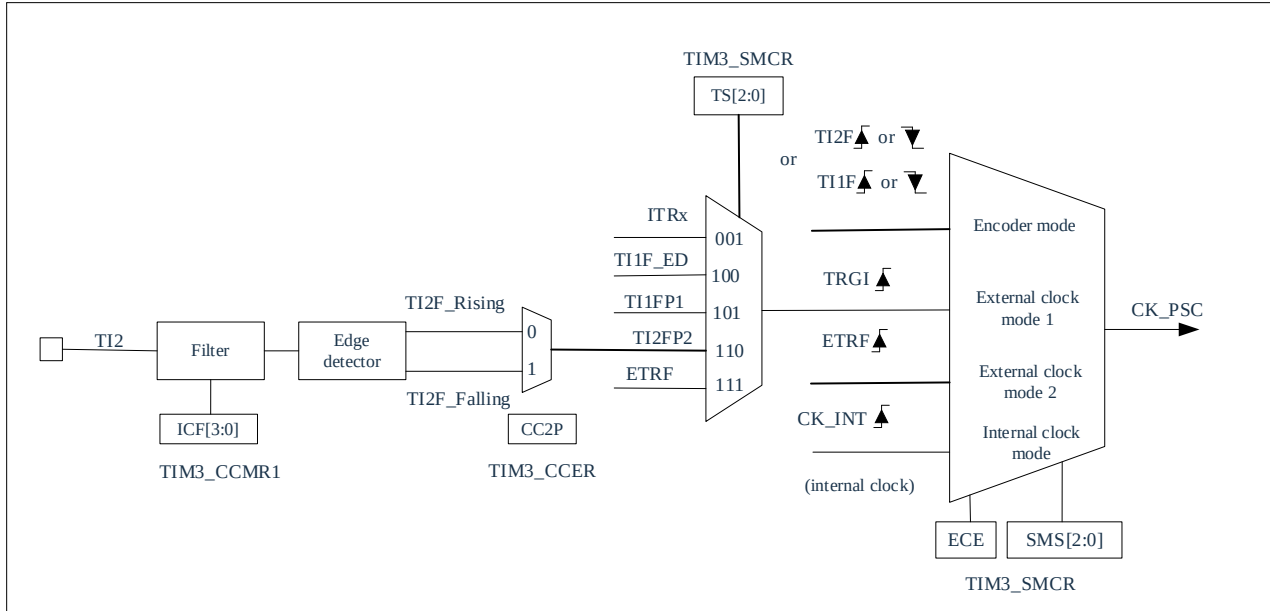
Figure 7-21 Control circuit in normal mode, internal clock divided by 1



7.3.3.2 External Clock Source Mode 1

When the SMS in the TIM3_SMCR register is set to 111, this mode can be selected. The counter will count on the rising or falling edge of the selected input signal.

Figure 7-22 TI2 external clock connection example



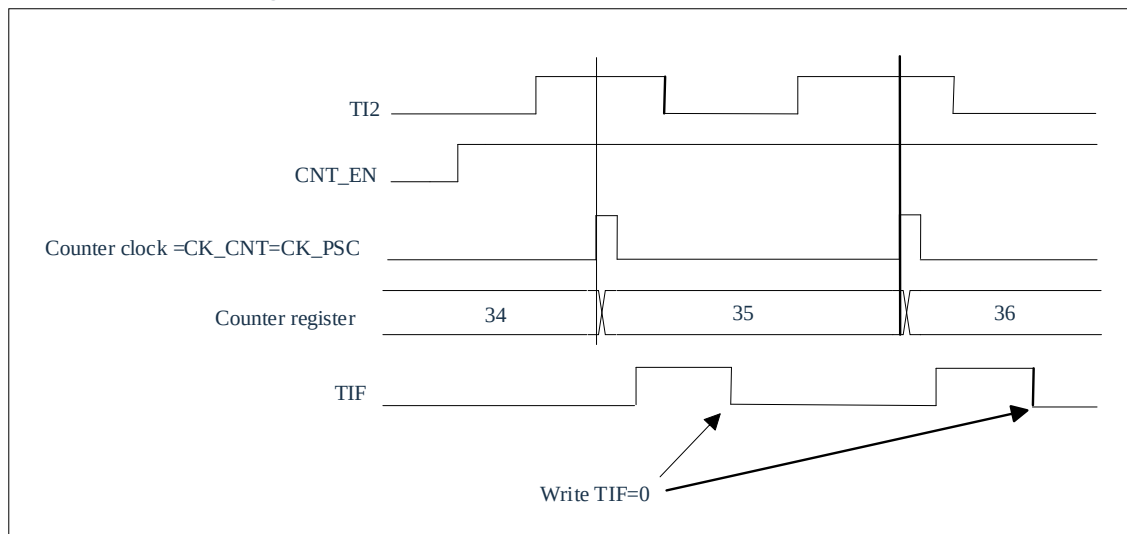
For example, to make the incrementing counter count on the rising edge of the TI2 input, follow these steps:

1. Configure Channel 2 to detect the rising edge of the TI2 input by writing CC2S = “01” in the TIM3_CCMR1 register.
2. Configure the input filter time by writing the IC2F[3:0] bits in the TIM3_CCMR1 register (if no filtering is needed, keep IC2F = 0000).
3. Select the rising edge polarity by writing CC2P = 0 and CC2NP = 0 in the TIM3_CCER register.
4. Enable the timer to work in External Clock Mode 1 by writing SMS = 111 in the TIM3_SMCR register.
5. Select TI2 as the input source by writing TS = 110 in the TIM3_SMCR register.
6. Enable the counter by writing CEN = 1 in the TIM3_CR1 register.

When the rising edge of TI2 occurs, the counter will increment once, and the TIF flag will be set to 1.

The delay between the rising edge of TI2 and the actual counter clock is caused by the resynchronization circuitry of the TI2 input.

Figure 7-23 Control circuit in external clock mode 1



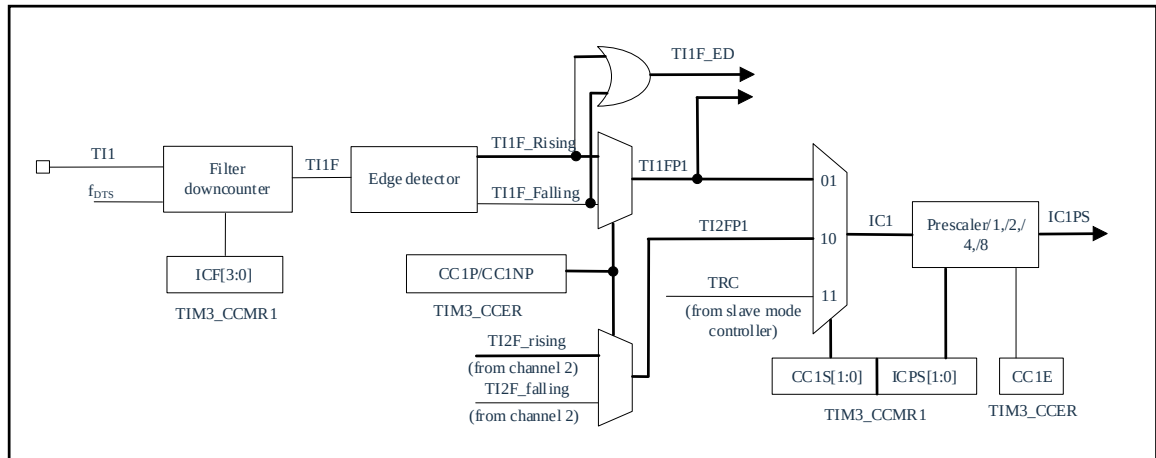
7.3.4 Capture/Compare Channels

Each capture/compare channel is composed of a capture/compare register (including a shadow register), a capture input stage (digital filter, multiplexing, and prescaler), and an output stage (comparator and output control).

The diagram below provides an overview of a capture/compare channel.

The input stage samples the corresponding TIx input and generates a filtered signal TIxF. Then, an edge detector with polarity selection generates a signal (TIxFPx), which can serve as a trigger input for the mode controller or as a capture command. This signal is first prescaled (ICxPS) and then enters the capture register.

Figure 7-24 Capture/compare channel (example: channel 1 input stage)



The output stage generates an intermediate waveform which is then used for reference: OCxRef (active high). The polarity acts at the end of the chain.

Figure 7-25 Capture/compare channel 1 main circuit

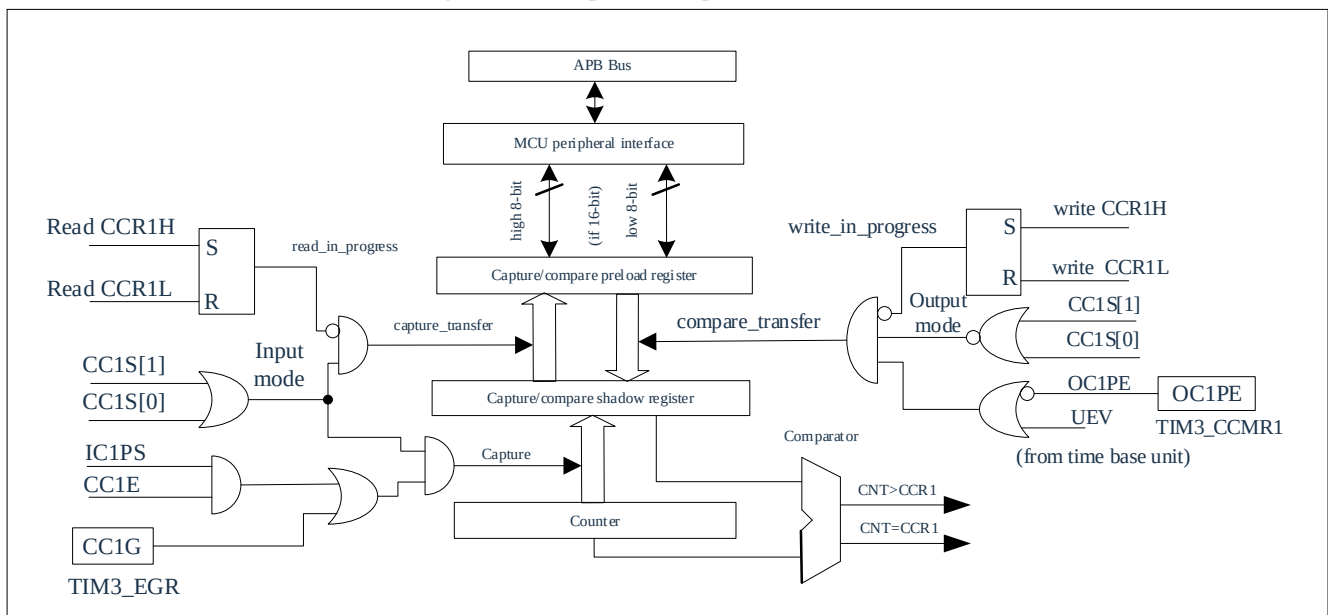
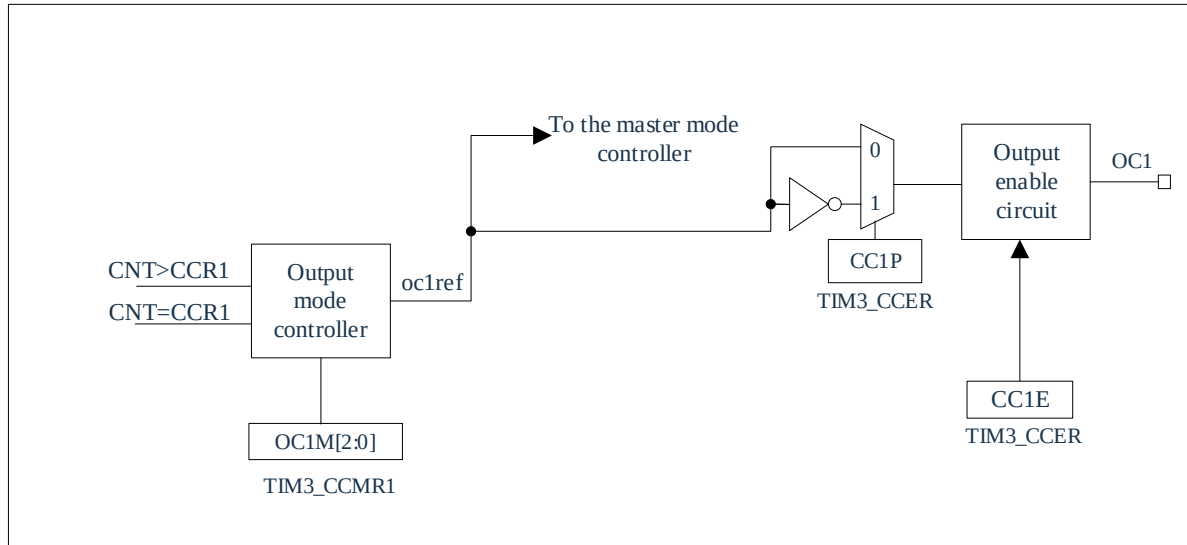


Figure 7-26 Output stage of capture/compare channel (channel 1)



The capture/compare block is made of one preload register and one shadow register. Write and read always access the preload register.

In capture mode, captures are actually done in the shadow register, which is copied into the preload register.

In compare mode, the content of the preload register is copied into the shadow register which is compared to the counter.

7.3.5 Input Capture Mode

In input capture mode, when a corresponding ICx signal detects a transition edge, the capture/compare register is used to latch the value of the counter. When a capture event occurs, the corresponding CCxIF flag (in the TIM3_SR register) is set to 1, and an interrupt or DMA request can be sent if enabled. If the CCxIF flag is already high when a capture event occurs, the repeat capture flag CCxOF (in the TIM3_SR register) will be set to 1. You can clear the CCxIF flag by writing 0 to it via software or by reading the captured data stored in the TIM3_CCRx register. Writing 0 to CCxOF clears it.

The following example illustrates how to capture the counter value into TIM3_CCR1 when a rising edge appears on the TI1 input. The specific steps are as follows:

- Select the valid input: TIM3_CCR1 must be connected to the TI1 input. Therefore, write “01” to the CC1S bit in the TIM3_CCMR1 register. As long as CC1S is not equal to 00, the channel will be configured in input mode, and the TIM3_CCR1 register will be read-only.
- Configure the input filter based on the signal connected to the timer for the required bandwidth (i.e., when the input is TIx, the input filter control bits are in the TIM3_CCMRx register’s ICxF bits). Suppose the input signal may jitter for up to 5 internal clock cycles when it changes. We need to set the filter time to be longer than 5 internal clock cycles. The transition edge on TI1 can then be confirmed after 8 consecutive samples with new levels (sampled at the fDTS frequency). Write “0011” to the IC1F bit in the TIM3_CCMR1 register.
- Select the valid transition edge by writing 0 to both CC1P and CC1NP bits in the TIM3_CCER register (in this case, the rising edge).
- Configure the input prescaler: In this case, we want the capture operation to occur on every valid transition, so we need to disable the prescaler (write “00” to the IC1PS bits in the TIM3_CCMR1 register).
- Enable the capture: Write 1 to the CC1E bit in the TIM3_CCER register to allow the counter value to be captured into the capture register.
- Optional: Enable the related interrupt request by writing 1 to the CC1IE bit in the TIM3_DIER register, or enable the DMA request by writing 1 to the CC1DE bit in the same register.

When input capture occurs:

- The counter value will be captured into TIM3_CCR1 when a valid transition edge occurs.
- The CC1IF flag is set to 1 (interrupt flag). If at least two consecutive captures happen, but the CC1IF flag is not cleared, the CC1OF overrun flag will be set to 1.
- An interrupt is generated based on the CC1IE bit.
- A DMA request is generated based on the CC1DE bit.

To handle repeated captures, it is recommended to read the data before reading the capture overrun flag to avoid missing repeated captures that may occur between reading the capture overrun flag and reading the data.

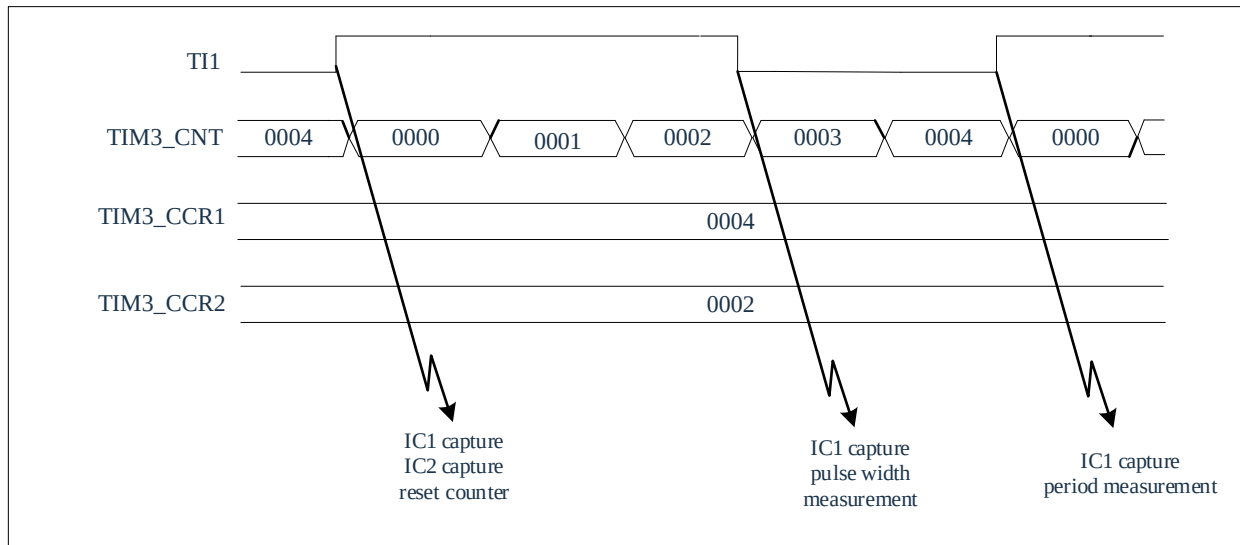
Note: To generate IC interrupts and/or DMA requests, write the corresponding CCxG bit in the TIM3_EGR register to 1 via software.

7.3.6 PWM Input Mode

This mode is a special case of the input capture mode. The implementation steps are essentially the same as those for input capture mode, with the following differences:

- Two ICx signals are mapped to the same TIx input.
- These two ICx signals are active at the edges, but with opposite polarities.
- Choose one of the two TIxFP signals as the trigger input and configure the mode controller to reset mode.
- For example, the following steps can be used to measure the period (located in TIM3_CCR1 register) and duty cycle (located in TIM3_CCR2 register) of a PWM applied to TI1 (depending on CK_INT frequency and the prescaler value):
- Select the valid input for TIM3_CCR1: Write “01” to the CC1S bit in the TIM3_CCMR1 register (select TI1).
- Select the valid polarity for TI1FP1 (for capture and counter clearing in TIM3_CCR1): Write “0” to both the CC1P and CC1NP bits (rising edge active).
- Select the valid input for TIM3_CCR2: Write “10” to the CC2S bits in the TIM3_CCMR1 register (select TI1).
- Select the valid polarity for TI1FP2 (for capture in TIM3_CCR2): Write “1” to the CC2P bit (falling edge active).
- Select the valid trigger input: Write “101” to the TS bits in the TIM3_SMCR register (select TI1FP1).
- Configure the mode controller to reset mode: Write “100” to the SMS bits in the TIM3_SMCR register.
- Enable capture: Write “1” to both the CC1E and CC2E bits in the TIM3_CCER register.

Figure 7-27 PWM input mode timing



7.3.7 Forced Output Mode

In output mode (when CCxS bits in the TIM3_CCMRx register are set to 00), each output compare signal (OCxREF and OCx) can be directly forced by software to either the active or inactive level, without considering any comparison results between the output compare register and the counter.

To force the output compare signal (OCxREF/OCx) to the active level, simply write “101” to the corresponding OCxM bits in the TIM3_CCMRx register. This will force OCxREF to be high (OCxREF is always high active), while OCx will get the opposite value of the CCxP polarity bit.

For example: CCxP=0 (OCx active high) => OCx is forced to high.\

To force the OCxREF signal to the low level, write “100” to the OCxM bits in the TIM3_CCMRx register. However, the comparison between the TIMx_CCRx shadow register and the counter will still be performed, and the flag can still be set to 1. As a result, the corresponding interrupt and DMA requests can be sent. The output compare mode section covers this in detail.

7.3.8 Output Compare Mode

This feature is used to control the output waveform or indicate that a certain time period has passed. When a match occurs between the capture/compare register and the counter, the output compare function:

- Outputs the value defined by the output compare mode (OCxM bits in TIM3_CCMRx register) and output polarity (CCxP bits in TIM3_CCER register) to the corresponding pin. On compare match, the output pin can either maintain its level (OCxM=000), be set to active level (OCxM=001), inactive level (OCxM=010), or toggle (OCxM=011).
- Sets the flag in the interrupt status register to 1 (CCxIF bit in the TIM3_SR register).
- If the corresponding interrupt enable bit (CCxIE bit in the TIM3_DIER register) is set to 1, an interrupt is generated.
- If the corresponding DMA enable bit (CCxDE bit in the TIM3_DIER register, CCDS bit in the TIM3_CR2 register for selecting DMA requests) is set to 1, a DMA request is sent.

Using the OCxPE bit in the TIM3_CCMRx register, the TIM3_CCRx register can be configured with or without a preload register.

In output compare mode, the update event (UEV) has no effect on OCxREF and OCx output. The synchronization precision can reach one counter cycle. Output compare mode can also be used for outputting an one pulse (in one pulse mode).

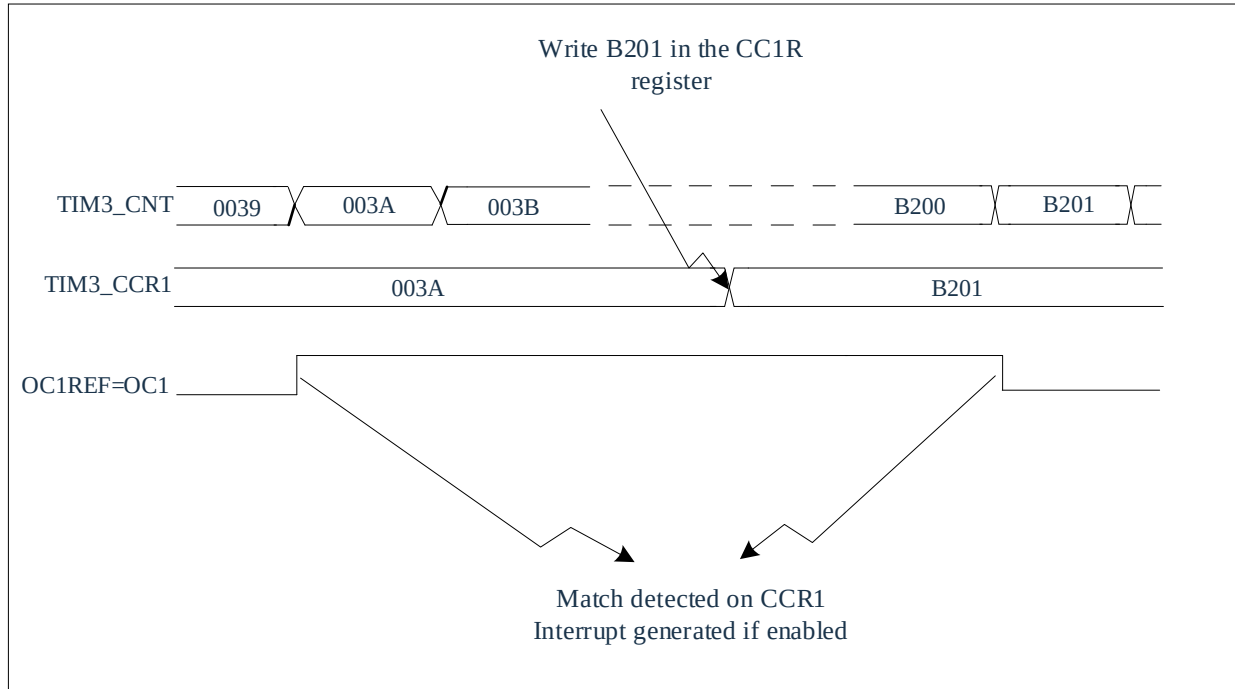
Steps:

1. Select the counter clock (internal, external, prescaler).
2. Write the desired data to the TIM3_ARR and TIM3_CCRx registers.
3. If generating interrupts and/or DMA requests is required, set the CCxIE and/or CCxDE bits to 1.
4. Select the output mode. For example:
 - To toggle OCx output pin when CNT matches CCRx, set OCxM=011.
 - Set OCxPE=0 to disable the preload register.
 - Set CCxP=0 to select active high polarity.
 - Set CCxE=1 to enable output.
5. Enable the counter by setting the CEN bit to 1 in the TIM3_CR1 register.

The TIM3_CCRx register can be updated at any time by software to control the output waveform, provided the preload register is not enabled (OCxPE=0; otherwise, the TIM3_CCRx shadow register will only be updated on the next update event (UEV)).

An example is given in Figure 7-28.

Figure 7-28 Output compare mode, toggle on OC1



7.3.9 PWM Mode

Pulse Width Modulation (PWM) mode can generate a signal whose frequency is determined by the TIM3_ARR register value, and its duty cycle is determined by the TIM3_CCRx register value.

By writing 110 (PWM mode 1) or 111 (PWM mode 2) to the OCxM bits in the TIM3_CCMRx register, the PWM mode for each channel (each OCx output corresponds to a PWM) can be independently selected.

To enable the corresponding preload register, the OCxPE bit in the TIM3_CCMRx register must be set to 1. Finally, to enable the automatic reload preload register, the ARPE bit in the TIM3_CR1 register must be set to 1.

Since the preload register will only transfer to the shadow register when an update event occurs, all registers must be initialized before starting the counter by setting the UG bit in the TIM3_EGR register to 1.

The OCx polarity can be programmed using the CCxP bit in the TIM3_CCER register. It can be set to active high or active low. The OCx output is enabled by setting the CCxE bit in the TIM3_CCER register. For more detailed information, refer to the TIM3_CCER register description.

In PWM mode (1 or 2), the TIM3_CNT is always compared with TIM3_CCRx to determine whether $TIM3\_CCR_x \leq TIM3\_CNT$ or $TIM3\_CNT \leq TIM3\_CCR_x$ (depending on the counter counting direction). However, to align with the ETRF (External Trigger), the OCxREF signal will only be active under the following conditions before the next PWM cycle:

- When the comparison result changes
- When the output compare mode (OCxM bits in TIM3_CCMRx register) switches from the "frozen" configuration (no comparison, OCxM = "000") to either of the PWM modes (OCxM = "110" or "111"). PWM output can be forced by software during the timer's operation.

Based on the CMS bit status in the TIM3_CR1 register, the timer can generate either edge-aligned or center-aligned PWM signals.

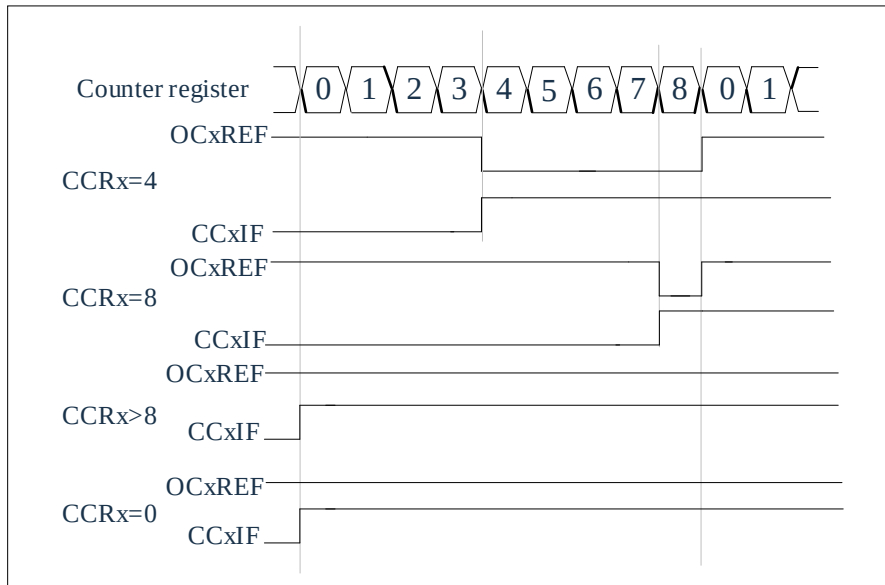
7.3.9.1 PWM Edge-Aligned Mode

Up-Counting Configuration

When the DIR bit in the TIM3_CR1 register is low, up-counting is performed.

The following describes PWM mode 1 as an example. As long as $TIM3\_CNT < TIM3\_CCR_x$, the PWM reference signal $OCxREF$ will be high, otherwise, it will be low. If the comparison value in $TIM3\_CCR_x$ is greater than the auto-reload value (in $TIM3\_ARR$), $OCxREF$ remains at “1”. If the comparison value is 0, $OCxREF$ remains at “0”. Figure 7-29 illustrates some PWM waveforms in edge-aligned mode ($TIM3\_ARR = 8$).

Figure 7-29 Edge-aligned PWM waveforms ($TIM3\_ARR=8$)



Down-Counting Configuration

When the DIR bit in the TIM3_CR1 register is high, down-counting is performed.

In PWM mode 1, as long as $TIM3\_CNT > TIM3\_CCR_x$, the reference signal $OCxREF$ will be low, otherwise, it will be high. If the comparison value in $TIM3\_CCR_x$ is greater than the auto-reload value in $TIM3\_ARR$, $OCxREF$ remains at “1”. In this mode, it is not possible to generate a 0% PWM waveform.

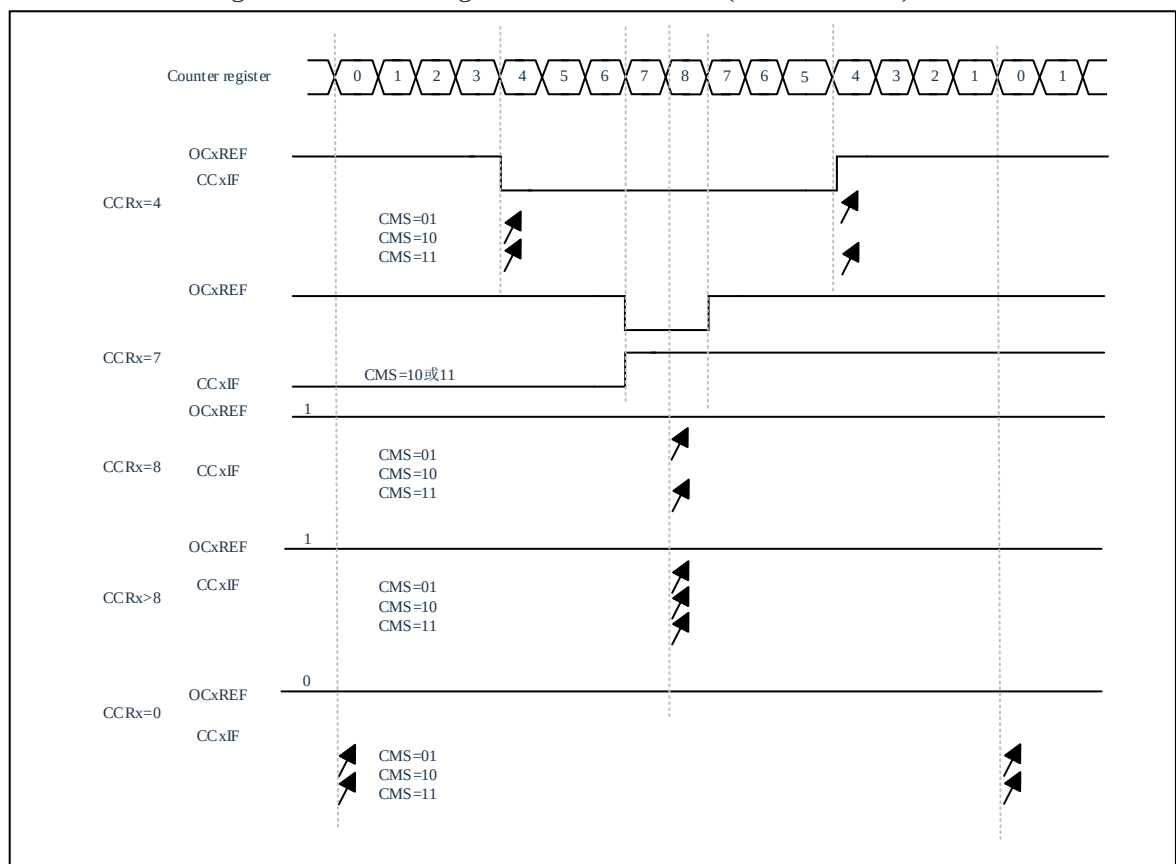
7.3.9.2 PWM Center-Aligned Mode

When the CMS bit in the TIM3_CR1 register is not “00” (with all other configurations having the same effect on OCxREF/OCx signal), center-aligned mode becomes effective. Depending on the CMS bit configuration, the comparison flag can be set to 1 when the counter is counting up, counting down, or counting both up and down. The direction bit (DIR) in the TIM3_CR1 register is updated by hardware and cannot be changed by software.

Figure 7-30 shows the PWM waveform in center-aligned mode. In this example:

- TIM3_ARR=8,
- PWM mode is PWM Mode 1,
- In the center-aligned mode 1 chosen based on CMS = 01 in the TIM3_CR1 register, the comparison flag is set to 1 when the counter counts down.

Figure 7-30 Center-aligned PWM waveforms (TIM3_ARR=8)



Hints on using center-aligned mode:

- When starting in center-aligned mode, the current up-down configuration is used. It means that the counter counts up or down depending on the value written in the DIR bit in the TIM3_CR1 register. Moreover, the DIR and CMS bits must not be changed at the same time by the software.
- Writing to the counter while running in center-aligned mode is not recommended as it can lead to unexpected results. In particular:
 - The direction is not updated if a value greater than the auto-reload value is written in the counter (TIM3_CNT>TIM3_ARR). For example, if the counter was counting up, it continues to count up.
 - The direction is updated if 0 or the TIM3_ARR value is written in the counter but no Update Event UEV is generated.

- The safest way to use center-aligned mode is to generate an update by software (setting the UG bit in the TIM3_EGR register) just before starting the counter and not to write the counter while it is running.

7.3.10 One-Pulse Mode

One-pulse mode (OPM) is a particular case of the previous modes. It allows the counter to be started in response to a stimulus and to generate a pulse with a programmable length after a programmable delay.

Starting the counter can be controlled through the slave mode controller. Generating the waveform can be done in output compare mode or PWM mode. One-pulse mode is selected by setting the OPM bit in the TIM3_CR1 register. This makes the counter stop automatically at the next update event UEV.

A pulse can be correctly generated only if the compare value is different from the counter initial value. Before starting (when the timer is waiting for the trigger), the configuration must be:

- In upcounting: $CNT < CCRx \leq ARR$ (in particular, $0 < CCRx$),
- In downcounting: $CNT > CCRx$.

Example of one-pulse mode

For example one may want to generate a positive pulse on OC1 with a length of t_{PULSE} and after a delay of t_{DELAY} as soon as a positive edge is detected on the TI2 input pin.

Use TI2FP2 as trigger 1:

- Map TI2FP2 on TI2 by writing $CC2S=01$ in the TIM3_CCMR1 register.
- TI2FP2 must detect a rising edge, write $CC2P=0$ and $CC2NP= '0'$ in the TIM3_CCER register.
- Configure TI2FP2 as trigger for the slave mode controller (TRGI) by writing $TS=110$ in the TIM3_SMCR register.
- TI2FP2 is used to start the counter by writing SMS to '110' in the TIM3_SMCR register (trigger mode).

The OPM waveform is defined by writing the compare registers (taking into account the clock frequency and the counter prescaler).

- The t_{DELAY} is defined by the value written in the TIM3_CCR1 register.
- The t_{PULSE} is defined by the difference between the auto-reload value and the compare value ($TIM3\_ARR - TIM3\_CCR1$).
- Let's say one want to build a waveform with a transition from '0' to '1' when a compare match occurs and a transition from '1' to '0' when the counter reaches the auto-reload value. To do this PWM mode 2 must be enabled by writing $OC1M=111$ in the TIM3_CCMR1 register. Optionally the preload registers can be enabled by writing $OC1PE=1$ in the TIM3_CCMR1 register and $ARPE$ in the TIM3_CR1 register. In this case one has to write the compare value in the TIM3_CCR1 register, the auto-reload value in the TIM3_ARR register, generate an update by setting the UG bit and wait for external trigger event on TI2. $CC1P$ is written to '0' in this example.

In our example, the DIR and CMS bits in the TIMx_CR1 register should be low.

Since only 1 pulse (Single mode) is needed, a 1 must be written in the OPM bit in the TIM3_CR1 register to stop the counter at the next update event (when the counter rolls over from the auto-reload value back to 0).

Particular case: OCx fast enable:

In One-pulse mode, the edge detection on T_{IX} input set the CEN bit which enables the counter. Then the comparison between the counter and the compare value makes the output toggle. But several clock cycles are needed for these operations and it limits the minimum delay $t_{\text{DELAY min}}$ we can get.

If one wants to output a waveform with the minimum delay, the OCxFE bit can be set in the TIM3_CCMRx register. Then OCxRef (and OCx) are forced in response to the stimulus, without taking in account the comparison. Its new level is the same as if a compare match had occurred. OCxFE acts only if the channel is configured in PWM1 or PWM2 mode.

7.3.11 Clearing the OCxREF Signal on an External Event

When a high level is applied to the ETRF input for a given channel (with the corresponding OCxCE enable bit in the TIM3_CCMRx register set to “1”), the OCxREF signal will become low. The OCxREF signal will remain low until the next update event (UEV).

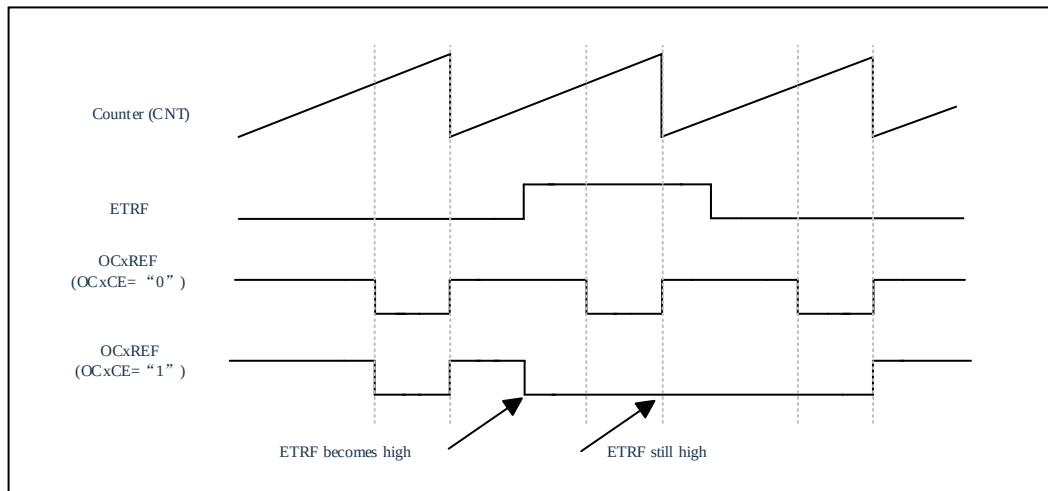
This feature can only be used in output compare mode and PWM mode, and is not applicable in forced output mode.

For example, the ETR signal can be connected to the output of a comparator to control current. In this case, the ETR must be configured as follows:

1. The external trigger prescaler must be turned off: The ETPS[1:0] bits in the TIM3_SMCR register must be set to “00”.
2. External clock mode 2 must be disabled: The ECE bit in the TIM3_SMCR register must be set to “0”.
3. The external trigger polarity (ETP) and external trigger filter (ETF) can be configured based on the application’s needs.

Figure 7-31 compares the behavior of the OCxREF signal when the OCxCE enable bit is in different states, showing how the OCxREF signal behaves when the ETRF input becomes high. In this example, the TIM3 timer is programmed in PWM mode.

Figure 7-31 Clearing TIMx OCxREF



In case of a PWM with a 100% duty cycle (if $CCR_x > ARR$), OCxREF is enabled again at the next counter overrun.

7.3.12 Encoder Interface Mode

To select Encoder Interface mode write SMS=001 in the TIM3_SMCR register if the counter is counting on TI2 edges only, SMS=010 if it is counting on TI1 edges only and SMS=011 if it is counting on both TI1 and TI2 edges.

Select the TI1 and TI2 polarity by programming the CC1P and CC2P bits in the TIM3_CCER register. CC1NP and CC2NP must be kept cleared. When needed, the input filter can be programmed as well.

The two inputs TI1 and TI2 are used to interface to an incremental encoder. Refer to Table 7-1. The counter is clocked by each valid transition on TI1FP1 or TI2FP2 (TI1 and TI2 after input filter and polarity selection, TI1FP1=TI1 and TI2FP2=TI2 if not filtered and not inverted). The sequence of transitions of the two inputs is evaluated and generates count pulses as well as the direction signal. Depending on the sequence the counter counts up or down, the DIR bit in the TIM3_CR1 register is modified by hardware accordingly. The DIR bit is calculated at each transition on any input (TI1 or TI2), whatever the counter is counting on TI1 only, TI2 only or both TI1 and TI2.

Encoder interface mode acts simply as an external clock with direction selection. This means that the counter just counts continuously between 0 and the auto-reload value in the TIM3_ARR register (0 to ARR or ARR down to 0 depending on the direction). So the TIM3_ARR must be configured before starting. In the same way, the capture, compare, prescaler, trigger output features continue to work as normal.

In this mode, the counter is modified automatically following the speed and the direction of the incremental encoder and its content, therefore, always represents the encoder's position. The count direction correspond to the rotation direction of the connected sensor. The table summarizes the possible combinations, assuming TI1 and TI2 do not switch at the same time.

Table 7-1 Counting direction versus encoder signals

Active edge	Level on opposite signal (TI1FP1 for TI2, TI2FP2 for TI1)	TI1FP1 signal		TI2FP2 signal	
		Rising	Falling	Rising	Falling
Counting on TI1 only	High	Down	Up	No Count	No Count
	Low	Up	Down	No Count	No Count
Counting on TI2 only	High	No Count	No Count	Up	Down
	Low	No Count	No Count	Down	Up
Counting on TI1 and TI2	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

An external incremental encoder can be connected directly to the MCU without external interface logic. However, comparators are normally be used to convert the encoder's differential outputs to digital signals. This greatly increases noise immunity. The third encoder output which indicate the mechanical zero position, may be connected to an external interrupt input and trigger a counter reset.

Figure 7-32 gives an example of counter operation, showing count signal generation and direction control. It also shows how input jitter is compensated where both edges are selected. This might occur if the sensor is positioned near to one of the switching points. For this example we assume that the configuration is the following:

- CC1S="01"(TIM3_CCMR1 register, IC1FP1 mapped on TI1)
- CC2S="01"(TIM3_CCMR2 register, IC1FP2 mapped on TI2)
- CC1P="0", CC1NP="0", IC1F="0000"(TIM3_CCER register, IC1FP1 noninverted, IC1FP1=TI1)
- CC2P="0", CC2NP="0", IC2F="0000"(TIM3_CCER register, IC2FP2 noninverted, IC1FP2=TI2)
- SMS="011"(TIMx_SMCR register, both inputs are active on both rising and falling edges)

- CEN = 1(TIM3_CR1 register, Counter is enabled)

Figure 7-32 Example of counter operation in encoder interface mode

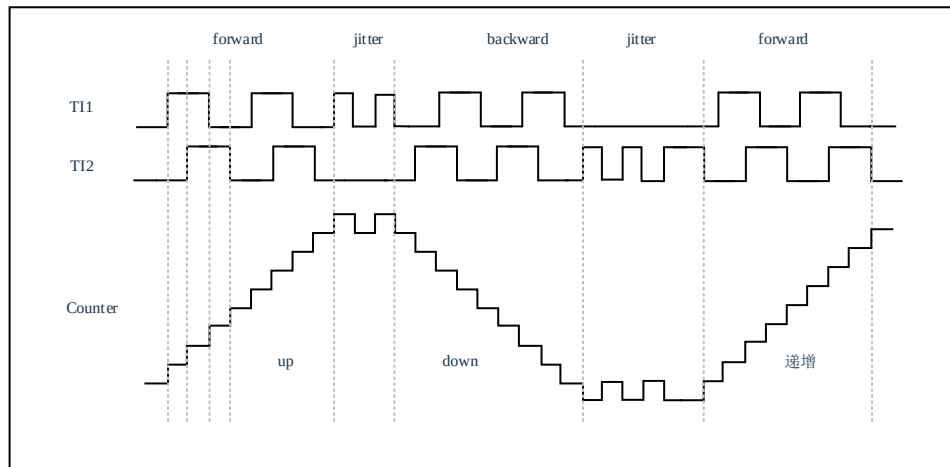
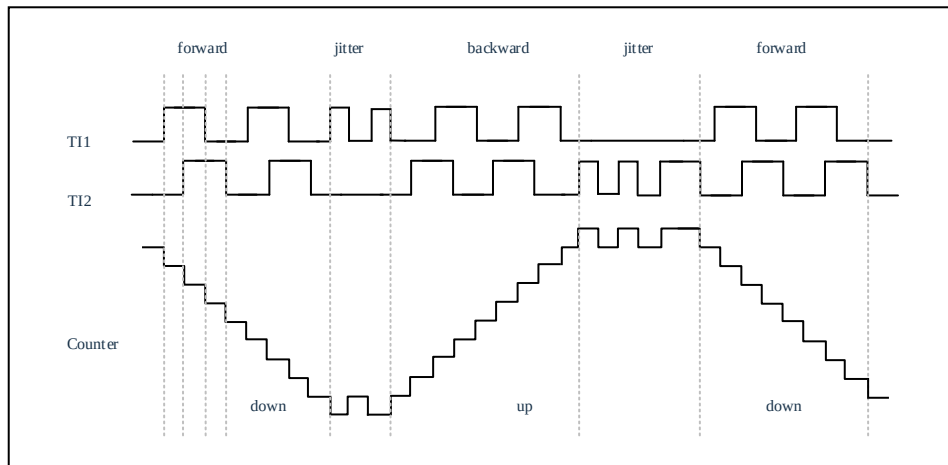


Figure 7-33 Example of encoder interface mode with TI1FP1 polarity inverted



The timer, when configured in Encoder Interface mode provides information on the sensor's current position. Dynamic information can be obtained (speed, acceleration, deceleration) by measuring the period between two encoder events using a second timer configured in capture mode. The output of the encoder which indicates the mechanical zero can be used for this purpose. Depending on the time between two events, the counter can also be read at regular times. This can be done by latching the counter value into a third input capture register if available (then the capture signal must be periodic and can be generated by another timer). when available, it is also possible to read its value through a DMA request generated by a Real-Time clock.

7.3.13 Timer Input XOR Function

The TI1S bit in the TIM3_CR2 register, allows the input filter of channel 1 to be connected to the output of a XOR gate, combining the three input pins TIM3_CH1 to TIM3_CH3.

The XOR output can be used with all the timer input functions such as trigger or input capture.

7.3.14 Timers and External Trigger Synchronization

The TIM3 Timers can be synchronized with an external trigger in several modes: Reset mode, Gated mode and Trigger mode.

7.3.14.1 Slave Mode: Reset Mode

The counter and its prescaler can be reinitialized in response to an event on a trigger input. Moreover, if the URS bit from the TIM3_CR1 register is low, an update event UEV is generated. Then all the preloaded registers (TIM3_ARR, TIM3_CCRx) are updated.

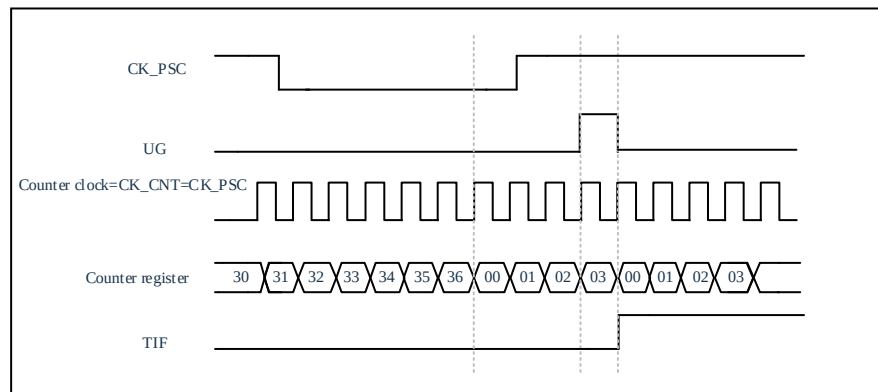
In the following example, the upcounter is cleared in response to a rising edge on TI1 input:

- Configure the channel 1 to detect rising edges on TI1. Configure the input filter duration (in this example, we do not need any filter, so we keep IC1F=0000). The capture prescaler is not used for triggering, so it does not need to be configured. The CC1S bits select the input capture source only, CC1S = 01 in the TIM3_CCMR1 register. Write CC1P=0 and CC1NP=0 in TIM3_CCER register to validate the polarity (and detect rising edges only).
- Configure the timer in reset mode by writing SMS=100 in TIM3_SMCR register. Select TI1 as the input source by writing TS=101 in TIM3_SMCR register.
- Start the counter by writing CEN=1 in the TIM3_CR1 register.

The counter starts counting on the internal clock, then behaves normally until TI1 rising edge. When TI1 rises, the counter is cleared and restarts from 0. In the meantime, the trigger flag is set (TIF bit in the TIM3_SR register) and an interrupt request, or a DMA request can be sent if enabled (depending on the TIE and TDE bits in TIM3_DIER register).

The following figure shows this behavior when the auto-reload register TIM3_ARR=0x36. The delay between the rising edge on TI1 and the actual reset of the counter is due to the resynchronization circuit on TI1 input.

Figure 7-34 Control circuit in reset mode



7.3.14.2 Slave Mode: Gated Mode

The counter can be enabled depending on the level of a selected input.

In the following example, the upcounter counts only when TI1 input is low:

- Configure the channel 1 to detect low levels on TI1. Configure the input filter duration (in this example, we do not need any filter, so we keep IC1F=0000). The capture prescaler is not used for triggering, so it does not need to be configured. The CC1S bits select the input capture source only, CC1S=01 in TIM3_CCMR1 register. Write CC1P=1 and CC1NP=0 in TIM3_CCER register to validate the polarity (and detect low level only).
- Configure the timer in gated mode by writing SMS=101 in TIM3_SMCR register. Select TI1 as the input source by writing TS=101 in TIM3_SMCR register.
- Enable the counter by writing CEN=1 in the TIM3_CR1 register (in gated mode, the counter doesn't start if CEN=0, whatever is the trigger input level).

The counter starts counting on the internal clock as long as TI1 is low and stops as soon as TI1 becomes high. The TIF flag in the TIM3_SR register is set both when the counter starts or stops.

The delay between the rising edge on TI1 and the actual stop of the counter is due to the resynchronization circuit on TI1 input.

7.3.14.3 Slave Mode: Trigger Mode

The counter can start in response to an event on a selected input.

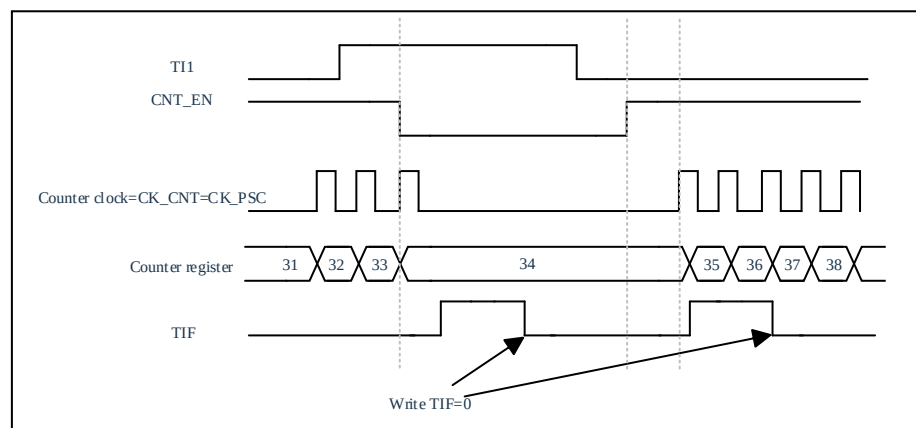
In the following example, the upcounter starts in response to a rising edge on TI2 input:

- Configure the channel 2 to detect rising edges on TI2. Configure the input filter duration (in this example, we do not need any filter, so we keep IC2F=0000). The capture prescaler is not used for triggering, so it does not need to be configured. CC2S bits are selecting the input capture source only, CC2S=01 in TIM3_CCMR1 register. Write CC2P=1 and CC2NP=0 in TIM3_CCER register to validate the polarity (and detect low level only).
- Configure the timer in trigger mode by writing SMS=110 in TIM3_SMCR register. Select TI2 as the input source by writing TS=110 in TIM3_SMCR register.

When a rising edge occurs on TI2, the counter starts counting on the internal clock and the TIF flag is set.

The delay between the rising edge on TI2 and the actual start of the counter is due to the resynchronization circuit on TI2 input.

Figure 7-35 Control circuit in trigger mode



7.3.14.4 Slave Mode: External Clock Mode 2 + Trigger Mode

The external clock mode 2 can be used in addition to another slave mode (except external clock mode 1 and encoder mode). In this case, the ETR signal is used as external clock input, and another input can be selected as trigger input when operating in reset mode, gated mode or trigger mode. It is recommended not to select ETR as TRGI through the TS bits of TIM3_SMCR register.

In the following example, the upcounter is incremented at each rising edge of the ETR signal as soon as a rising edge of TI1 occurs:

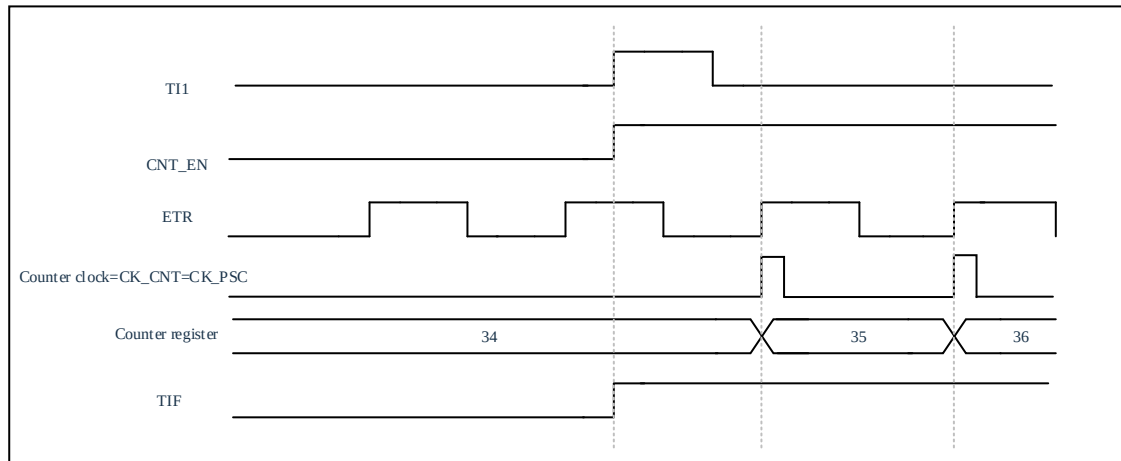
1. Configure the external trigger input circuit by programming the TIM3_SMCR register as follows:
 - ETF = 0000: no filter
 - ETPS = 00: prescaler disabled
 - ETP = 0: detection of rising edges on ETR and ECE=1 to enable the external clock mode 2.
2. Configure the channel 1 as follows, to detect rising edges on TI:
 - IC1F = 0000: no filter
 - The capture prescaler is not used for triggering and does not need to be configured.
 - CC1S = 01 in TIM3_CCMR1 register to select only the input capture source
 - CC1P = 0 in TIM3_CCER register to validate the polarity (and detect rising edge only).
3. Configure the timer in trigger mode by writing SMS=110 in TIM3_SMCR register. Select TI1 as the input source by writing TS=101 in TIM3_SMCR register.

A rising edge on TI1 enables the counter and sets the TIF flag. The counter then counts on ETR rising edges.

The delay between the rising edge of the ETR signal and the actual reset of the counter is due to the resynchronization circuit on ETRP input.

Note: This chip does not support this mode.

Figure 7-36 Control circuit in trigger mode



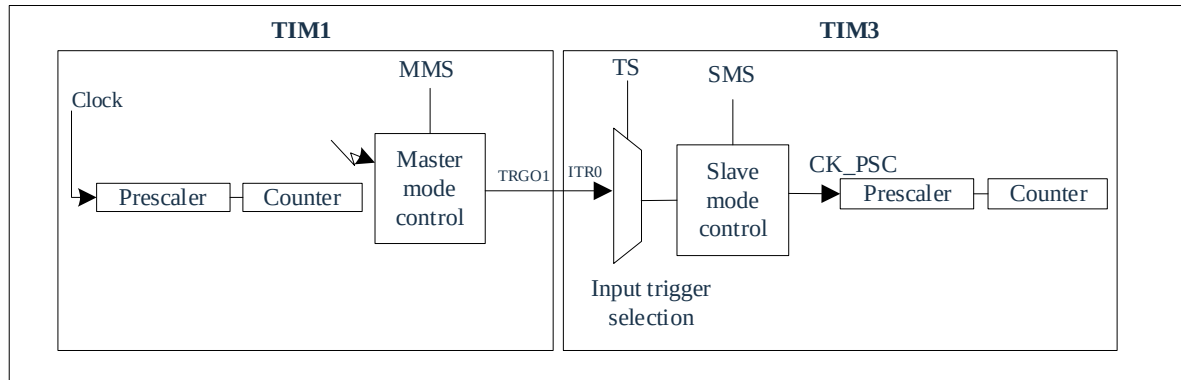
7.3.15 Timer Synchronization

The TIM3 timers are linked together internally for timer synchronization or chaining. When one Timer is configured in Master Mode, it can reset, start, stop or clock the counter of another Timer configured in Slave Mode.

Figure 7-37: Master/Slave timer example presents an overview of the trigger selection and the master mode selection blocks.

Using one timer as prescaler for another

Figure 7-37 Master/Slave timer example



For example, Timer 1 can be configured to act as a prescaler for Timer 3. Refer to Figure 7-37. To do this:

- Configure Timer 1 in master mode so that it outputs a periodic trigger signal on each update event UEV. If MMS=010 is written in the TIM1_CR2 register, a rising edge is output on TRGO1 each time an update event is generated.
- To connect the TRGO1 output of Timer 1 to Timer 3, Timer 3 must be configured in slave mode using ITR1 as internal trigger. This is selected through the TS bits in the TIM3_SMCR register (writing TS=000).
- Then the Timer2's slave mode controller should be configured in external clock mode 1 (write SMS=111 in the TIM3_SMCR register). This causes Timer 3 to be clocked by the rising edge of the periodic Timer 1 trigger signal (which correspond to the timer 1 counter overrun).
- Finally both timers must be enabled by setting their respective CEN bits within their respective TIMx_CR1 registers. Make sure to enable Timer3 before enabling Timer1.

Caution: If OCx is selected on Timer 1 as trigger output (MMS=1xx), its rising edge is used to clock the counter of timer 3.

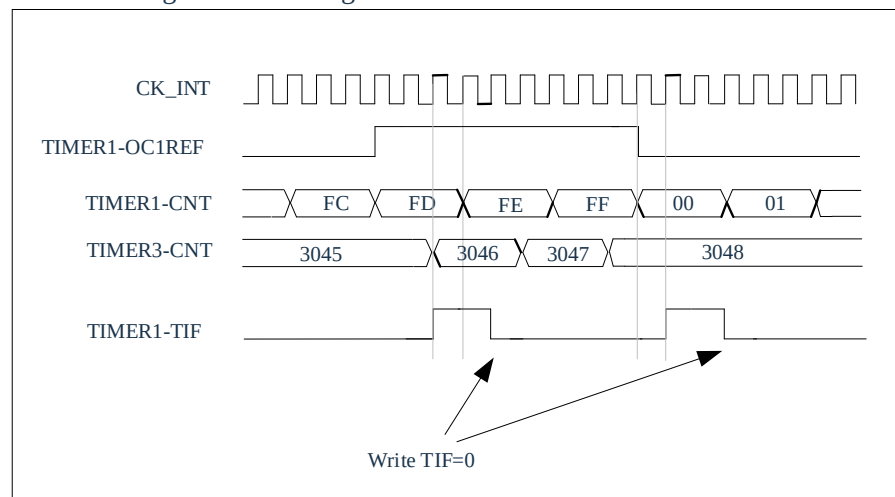
7.3.15.1 Using One Timer to Enable Another Timer

In this example, we control the enable of Timer 3 with the output compare 1 of Timer 1. Refer to Figure 7-37 for connections. Timer 3 counts on the divided internal clock only when OC1REF of Timer 1 is high. Both counter clock frequencies are divided by 3 by the prescaler compared to CK_INT ($f_{CK\_CNT} = f_{CK\_INT}/3$).

- Configure Timer 1 master mode to send its Output Compare 1 Reference (OC1REF) signal as trigger output (MMS=100 in the TIM1_CR2 register).
- Configure the Timer 1 OC1REF waveform (TIM1_CCMR1 register).
- Configure Timer 3 to get the input trigger from Timer 1 (TS=000 in the TIM3_SMCR register).
- Configure Timer 3 in gated mode (SMS=101 in TIM3_SMCR register).
- Enable Timer 3 by writing 1 in the CEN bit (TIM3_CR1 register).
- Start Timer 1 by writing 1 in the CEN bit (TIM1_CR1 register).

Note: The counter 3 clock is not synchronized with counter 1, this mode only affects the Timer 3 counter enable signal.

Figure 7-38 Gating timer 3 with OC1REF of timer 1



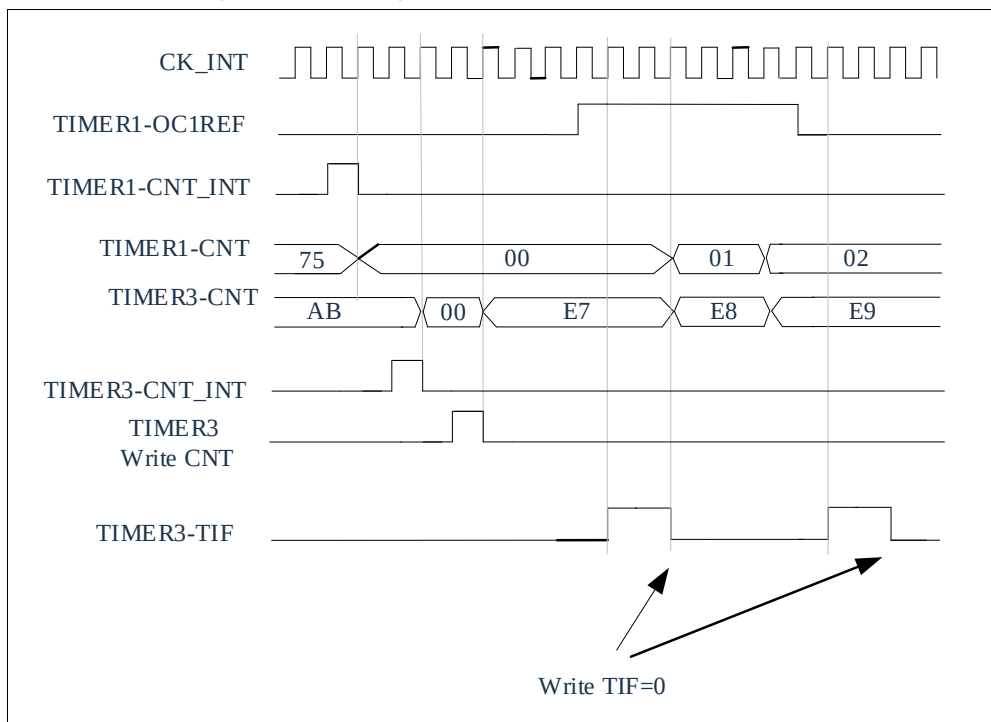
In the example in Figure 7-38, the Timer 3 counter and prescaler are not initialized before being started. So they start counting from their current value. It is possible to start from a given value by resetting both timers before starting Timer 1. Then any value can be written in the timer counters. The timers can easily be reset by software using the UG bit in the TIMx_EGR registers.

In the next example, we synchronize Timer 1 and Timer 3. Timer 1 is the master and starts from 0. Timer 3 is the slave and starts from 0xE7. The prescaler ratio is the same for both timers. Timer 3 stops when Timer 1 is disabled by writing 0 to the CEN bit in the TIM1_CR1 register:

- Configure Timer 1 master mode to send its Counter Enable signal (CNT_EN) as a trigger output (MMS=001 in the TIM1_CR2 register).
- Configure the Timer 1 OC1REF waveform (TIM1_CCMR1 register).
- Configure Timer 3 to get the input trigger from Timer 1 (TS=000 in the TIM3_SMCR register).
- Configure Timer 3 in gated mode (SMS=101 in TIM3_SMCR register).
- Reset Timer 1 by writing 1 in UG bit (TIM1_EGR register).
- Reset Timer 3 by writing 1 in UG bit (TIM3_EGR register).
- Initialize Timer 3 to 0xE7 by writing '0xE7' in the timer 3 counter (TIM3_CNT).
- Enable Timer 3 by writing 1 in the CEN bit (TIM3_CR1 register).
- Start Timer 1 by writing 1 in the CEN bit (TIM1_CR1 register).

- Stop Timer 1 by writing 0 in the CEN bit (TIM1_CR1 register).

Figure 7-39 Gating timer 3 with enable of timer 1

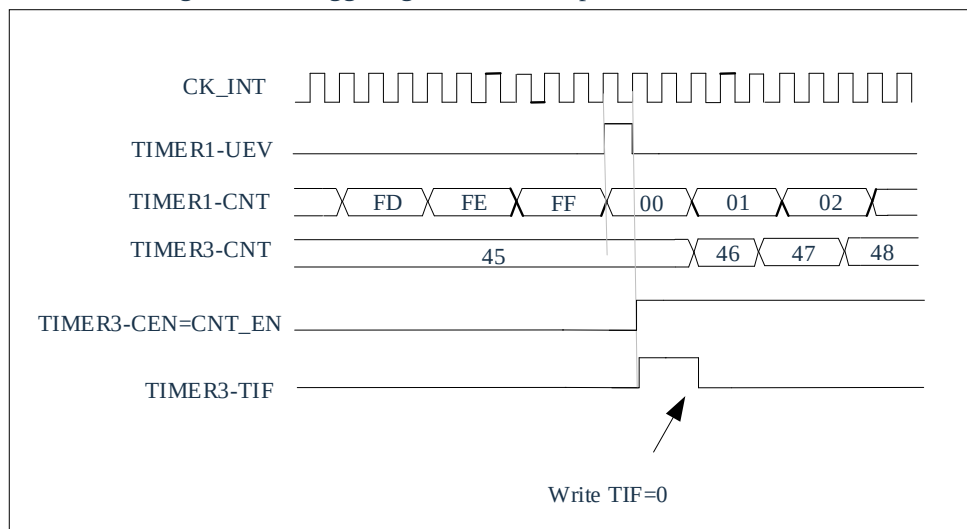


7.3.15.2 Using One Timer to Start Another Timer

In this example, we set the enable of Timer 3 with the update event of Timer 1. Refer to Figure 7-40 for connections. Timer 3 starts counting from its current value (which can be nonzero) on the divided internal clock as soon as the update event is generated by Timer 1. When Timer 3 receives the trigger signal its CEN bit is automatically set and the counter counts until we write 0 to the CEN bit in the TIM3_CR1 register. Both counter clock frequencies are divided by 3 by the prescaler compared to CK_INT ($f_{CK\_CNT} = f_{CK\_INT}/3$).

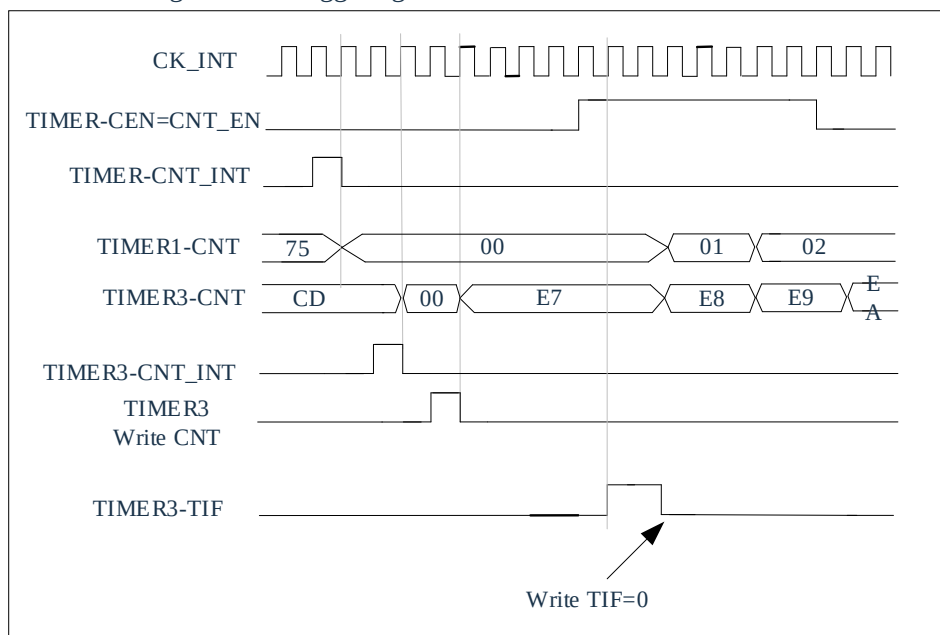
- Configure Timer 1 master mode to send its Update Event (UEV) as trigger output (MMS=010 in the TIM1_CR2 register).
- Configure the Timer 1 period (TIM1_ARR registers).
- Configure Timer 3 to get the input trigger from Timer 1 (TS=000 in the TIM3_SMCR register).
- Configure Timer 3 in trigger mode (SMS=110 in TIM3_SMCR register).
- Start Timer 1 by writing 1 in the CEN bit (TIM1_CR1 register).

Figure 7-40 Triggering timer 3 with update of timer 1



As in the previous example, both counters can be initialized before starting counting. Figure 7-41 shows the behavior with the same configuration as in Figure 7-40 but in trigger mode instead of gated mode (SMS=110 in the TIM3_SMCR register).

Figure 7-41 Triggering timer 3 with enable of timer 1



7.3.15.3 Using One Timer as a Prescaler for Another Timer

For example, Timer 1 can be configured as a prescaler for Timer 3. The relevant connection diagram is shown in Figure 7-37. To achieve this:

- Configure Timer 1 in master mode, sending its update event (UEV) as a trigger output (MMS = 010 in the TIM1_CR2 register). This will output a periodic signal each time the counter overruns.
- Configure the period of Timer 1 (TIM1_ARR register).
- Configure Timer 3 to receive an input trigger from Timer 1 (TS = 000 in the TIM3_SMCR register).
- Configure Timer 3 in external clock mode (SMS = 111 in the TIM3_SMCR register).
- Start Timer 3 by writing 1 to the CEN bit (TIM3_CR1 register).
- Start Timer 1 by writing 1 to the CEN bit (TIM1_CR1 register).

7.3.15.4 Starting 2 Timers Synchronously in Response to an External Trigger

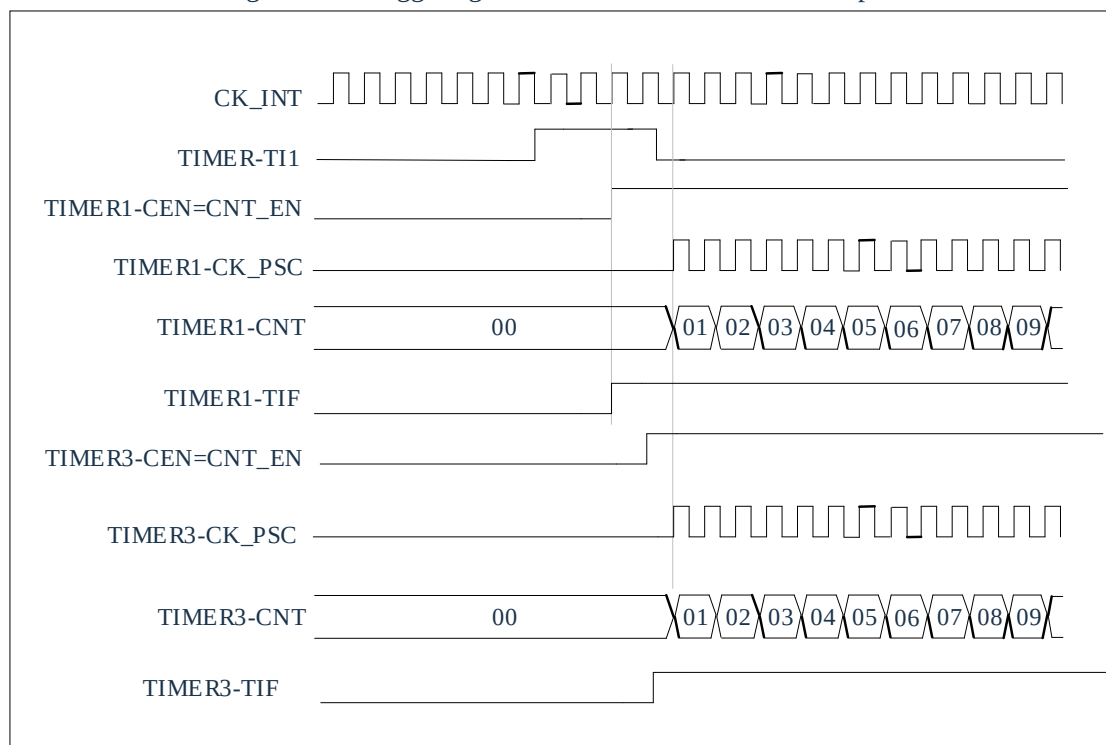
In this example, we set the enable of timer 1 when its TI1 input rises, and the enable of Timer 3 with the enable of Timer 1. Refer to Figure 7-37 for connections. To ensure the counters are aligned, Timer 1 must be configured in Master/Slave mode (slave with respect to TI1, master with respect to Timer 3):

- Configure Timer 1 master mode to send its Enable as trigger output (MMS=001 in the TIM1_CR2 register).
- Configure Timer 1 slave mode to get the input trigger from TI1 (TS=100 in the TIM1_SMCR register).
- Configure Timer 1 in trigger mode (SMS=110 in the TIM1_SMCR register).
- Configure the Timer 1 in Master/Slave mode by writing MSM=1 (TIM1_SMCR register).
- Configure Timer 3 to get the input trigger from Timer 1 (TS=000 in the TIM3_SMCR register).
- Configure Timer 3 in trigger mode (SMS=110 in the TIM3_SMCR register).

When a rising edge occurs on TI1 (Timer 1), both counters starts counting synchronously on the internal clock and both TIF flags are set.

Note: In this example both timers are initialized before starting (by setting their respective UG bits). Both counters starts from 0, but an offset can easily be inserted between them by writing any of the counter registers (TIMx_CNT). One can see that the master/slave mode insert a delay between CNT_EN and CK_PSC on timer 1.

Figure 7-42 Triggering timer 1 and 3 with timer 1 TI1 input



7.3.16 Debug Mode

When the chip enters debug mode, according to the settings in section 1.4, the TIM3 counter can either continue to operate normally or stop.

7.4 Register Mapping

(TIM3 base address = 0x4006_0400)

RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Address	R/W	Description	Reset value
CR1	0x00	R/W	TIM3 Control Register 1	0x00
CR2	0x04	R/W	TIM3 Control Register 2	0x00
SMCR	0x08	R/W	TIM3 Slave Mode Control Register	0x00
DIER	0x0C	R/W	TIM3 DMA/Interrupt Enable Register	0x00
SR	0x10	R/W	TIM3 Status Register	0x00
EGR	0x14	R/W	TIM3 Event Generation Register	0x00
CCMR1	0x18	R/W	TIM3 Capture/Compare Mode Register 1	0x00
CCMR2	0x1C	R/W	TIM3 Capture/Compare Mode Register 2	0x00
CCER	0x20	R/W	TIM3 Capture/Compare Enable Register	0x00
CNT	0x24	R/W	TIM3 Counter	0x00
PSC	0x28	R/W	TIM3 Prescaler	0x00
ARR	0x2C	R/W	TIM3 Auto-Reload Register	0xFFFF
CCR1	0x34	R/W	TIM3 Capture/Compare Register 1	0x00
CCR2	0x38	R/W	TIM3 Capture/Compare Register 2	0x00
CCR3	0x3C	R/W	TIM3 Capture/Compare Register 3	0x00
CCR4	0x40	R/W	TIM3 Capture/Compare Register 4	0x00
DCR	0x48	R/W	TIM3 DMA Control Register	0x00
DMAR	0x4C	R/W	TIM3 Full Transfer DMA Address	0x00

7.5 Register Description

7.5.1 TIM3 Control Register 1 (TIM3_CR1)

Bit	Symbol	Description	Reset value
31:10	--	Reserved, set to 0.	0x0
9:8	CKD[1:0]	Clock division This bit field indicates the division ratio between the timer clock (CK_INT) frequency and the dead-time generator as well as the sampling clock (tDTS) used by the digital filter (ETR, TIx). 00: tDTS = tCK_INT 01: tDTS = 2 * tCK_INT 10: tDTS = 4 * tCK_INT 11: Reserved, setting this value is not allowed.	0x0
7	ARPE	Auto-reload preload enable 0: TIM3_ARR register is not buffered 1: TIM3_ARR register is buffered	0
6:5	CMS[1:0]	Center-aligned mode selection: 00: Edge-aligned mode. The counter counts up or down depending on the direction bit (DIR). 01: Center-aligned mode 1. The counter counts up and down alternatively. Output compare interrupt flags of channels configured in output (CCxS=00 in TIM3_CCMRx register) are set only when the counter is counting down. 10: Center-aligned mode 2. The counter counts up and down alternatively. Output compare interrupt flags of channels configured in output (CCxS=00 in TIM3_CCMRx register) are set only when the counter is counting up. 11: Center-aligned mode 3. The counter counts up and down alternatively. Output compare interrupt flags of channels configured in output (CCxS=00 in TIM3_CCMRx register) are set both when the counter is counting up or down. Note: It is not allowed to switch from edge-aligned mode to center-aligned mode as long as the counter is enabled (CEN=1)	0x0
4	DIR	Direction: 0: Counter used as upcounter 1: Counter used as downcounter Note: This bit is read only when the timer is configured in Center-aligned mode or Encoder mode.	0
3	OPM	One-pulse mode: 0: Counter is not stopped at update event 1: Counter stops counting at the next update event (clearing the bit CEN)	0
2	URS	Update request source: This bit is set and cleared by software to select the UEV event sources. 0: Any of the following events generate an update interrupt or DMA request if enabled. These events can be: - Counter overrun/underrun - Setting the UG bit - Update generation through the slave mode controller 1: Only counter overrun/underrun generates an update interrupt or DMA request if enabled.	0
1	UDIS	Update disable: This bit is set and cleared by software to enable/disable UEV event generation. 0: UEV enabled. The Update (UEV) event is generated by one of the following events: - Counter overrun/underrun - Setting the UG bit - Update generation through the slave mode controller Buffered registers are then loaded with their preload values.	0

		1: UEV disabled. The Update event is not generated, shadow registers keep their value (ARR, PSC, CCRx). However the counter and the prescaler are reinitialized if the UG bit is set or if a hardware reset is received from the slave mode controller.	
0	CEN	Counter enable: 0: Counter disabled 1: Counter enabled Note: External clock, gated mode and encoder mode can work only if the CEN bit has been previously set by software. However trigger mode can set the CEN bit automatically by hardware.	0

7.5.2 TIM3 Control Register 2 (TIM3_CR2)

Bit	Symbol	Description	Reset value
31:8	--	Reserved, set to 0.	0
7	TI1S	TI1 selection: 0: The TIM3_CH1 pin is connected to TI1 input 1: The TIM3_CH1, CH2 and CH3 pins are connected to the TI1 input (XOR combination)	0
6:4	MMS[1:0]	Master mode selection: These bits allow to select the information to be sent in master mode to slave timers for synchronization (TRGO). The combination is as follows: 000: Reset - the UG bit from the TIM3_EGR register is used as trigger output (TRGO). If the reset is generated by the trigger input (slave mode controller configured in reset mode) then the signal on TRGO is delayed compared to the actual reset. 001: Enable - the Counter enable signal, CNT_EN, is used as trigger output (TRGO). It is useful to start several timers at the same time or to control a window in which a slave timer is enabled. The Counter Enable signal is generated by a logic OR between CEN control bit and the trigger input when configured in gated mode. When the Counter Enable signal is controlled by the trigger input, there is a delay on TRGO, except if the master/slave mode is selected (see the MSM bit description in TIM3_SMCR register). 010: Update - The update event is selected as trigger output (TRGO). For instance a master timer can then be used as a prescaler for a slave timer. 011: Compare Pulse - The trigger output send a positive pulse when the CC1IF flag is to be set (even if it was already high), as soon as a capture or a compare match occurred. (TRGO) 100: Compare - OC1REF signal is used as trigger output (TRGO) 101: Compare – OC2REF signal is used as trigger output (TRGO) 110: Compare – OC3REF signal is used as trigger output (TRGO) 111: Compare – OC4REF signal is used as trigger output (TRGO)	0x0
3	CCDS	Capture/compare DMA selection: 0: CCx DMA request sent when CCx event occurs 1: CCx DMA requests sent when update event occurs	0
2:0	--	Reserved, set to 0.	0x0

7.5.3 TIM3 Slave Mode Control Register (TIM3_SMCR)

Bit	Symbol	Description	Reset value
31:8	--	Reserved, set to 0.	0x0
7	MSM	Master/slave mode: 0: No action 1: The effect of an event on the trigger input (TRGI) is delayed to allow a perfect synchronization between the current timer and its slaves (through TRGO). It is useful if we want to synchronize several timers on a single external event.	0
6:4	TS[2:0]	Trigger selection: This bit-field selects the trigger input to be used to synchronize the counter. 000: Internal Trigger 0(ITR0) 001: Internal Trigger 1(ITR1) 010: Internal Trigger 2(ITR2) 011: Internal Trigger 3(ITR3) 100: TI1 Edge Detector (TI1F_ED) 101: Filtered Timer Input 1 (TI1FP1) 110: Filtered Timer Input 2 (TI2FP2) 111: External Trigger input (ETRF) Note: These bits must be changed only when they are not used to avoid wrong edge detections at the transition.	0x0
3	OCCS	OCREF clear selection: (This bit is used to select the OCREF clear source) 0: OCREF_CLR_INT is connected to the OCREF_CLR input 1: OCREF_CLR_INT is connected to ETRF	0
2:0	SMS	Slave mode selection: When external signals are selected the active edge of the trigger signal (TRGI) is linked to the polarity selected on the external input (see Input Control register and Control Register description). 000: Slave mode disabled - if CEN = 1 then the prescaler is clocked directly by the internal clock. 001: Encoder mode 1 - Counter counts up/down on TI1FP1 edge depending on TI2FP2 level. 010: Encoder mode 2 - Counter counts up/down on TI2FP2 edge depending on TI1FP1 level. 011: Encoder mode 3 - Counter counts up/down on both TI1FP1 and TI2FP2 edges depending on the level of the other input. 100: Reset Mode - Rising edge of the selected trigger input (TRGI) reinitializes the counter and generates an update of the registers. 101: Gated Mode - The counter clock is enabled when the trigger input (TRGI) is high. The counter stops (but is not reset) as soon as the trigger becomes low. Both start and stop of the counter are controlled. 110: Trigger Mode - The counter starts at a rising edge of the trigger TRGI (but it is not reset). Only the start of the counter is controlled. 111: External Clock Mode 1 - Rising edges of the selected trigger (TRGI) clock the counter. Note: The gated mode must not be used if TI1F_ED is selected as the trigger input (TS=100). Indeed, TI1F_ED outputs 1 pulse for each transition on TI1F, whereas the gated mode checks the level of the trigger signal.	0x0

Table 7-2 TIM3 internal trigger connection

Slave TIM	ITT0(TS=000)	ITT0(TS=001)	ITT0(TS=010)	ITT0(TS=011)
TIM3	TIM1	Reserved	Reserved	TIM14 OC1

7.5.4 TIM3 DMA/Interrupt Enable Register (TIM3_DIER)

Bit	Symbol	Description	Reset value
31:15	--	Reserved, set to 0.	0x0
14	TDE	Trigger DMA request enable 0: Trigger DMA request disabled 1: Trigger DMA request enabled	0
13	--	Reserved, set to 0.	0
12	CC4DE	Capture/Compare 4 DMA request enable 0: CC4 DMA request disabled. 1: CC4 DMA request enabled.	0
11	CC3DE	Capture/Compare 3 DMA request enable 0: CC3 DMA request disabled. 1: CC3 DMA request enabled.	0
10	CC2DE	Capture/Compare 2 DMA request enable 0: CC2 DMA request disabled. 1: CC2 DMA request enabled.	0
9	CC1DE	Capture/Compare 1 DMA request enable 0: CC1 DMA request disabled. 1: CC1 DMA request enabled.	0
8	UDE	Update DMA request enable 0: Update DMA request disabled. 1: Update DMA request enabled.	0
7	--	Reserved, set to 0.	0
6	TIE	Trigger interrupt enable 0: Trigger interrupt disabled. 1: Trigger interrupt enabled.	0
5	--	Reserved, set to 0.	0
4	CC4IE	CC4IE: Capture/Compare 4 interrupt enable 0: CC4 interrupt disabled. 1: CC4 interrupt enabled.	0
3	CC3IE	CC3IE: Capture/Compare 3 interrupt enable 0: CC3 interrupt disabled. 1: CC3 interrupt enabled.	0
2	CC2IE	CC2IE: Capture/Compare 2 interrupt enable 0: CC2 interrupt disabled. 1: CC2 interrupt enabled.	0
1	CC1IE	CC1IE: Capture/Compare 1 interrupt enable 0: CC1 interrupt disabled. 1: CC1 interrupt enabled.	0
0	UIE	Update interrupt enable 0: Update interrupt disabled 1: Update interrupt enabled	0

7.5.5 TIM3 Status Register (TIM3_SR)

Bit	Symbol	Description	Reset value
31:13	--	Reserved, set to 0.	0x0
12	CC4OF	Capture/Compare 4 overcapture flag Refer to CC1OF description	0
11	CC3OF	Capture/Compare 3 overcapture flag Refer to CC1OF description	0
10	CC2OF	Capture/Compare 2 overcapture flag Refer to CC1OF description	0
9	CC1OF	Capture/Compare 1 overcapture flag This flag is set by hardware only when the corresponding channel is configured in input capture mode. It is cleared by software by writing it to 0. 0: No overcapture has been detected 1: The counter value has been captured in TIMx_CCR1 register while CC1IF flag was already set	0
8:7	--	Reserved, set to 0.	0x0
6	TIF	Trigger interrupt flag This flag is set by hardware on trigger event (active edge detected on TRGI input when the slave mode controller is enabled in all modes but gated mode. It is set when the counter starts or stops when gated mode is selected. It is cleared by software. 0: No trigger event occurred 1: Trigger interrupt pending	0
5	--	Reserved, set to 0.	0
4	CC4IF	CC4IF: Capture/Compare 4 interrupt flag Refer to CC1IF description	0
3	CC3IF	CC3IF: Capture/Compare 3 interrupt flag Refer to CC1IF description	0
2	CC2IF	CC2IF: Capture/Compare 2 interrupt flag Refer to CC1IF description	0
1	CC1IF	If channel CC1 is configured as output: This flag is set by hardware when the counter matches the compare value, with some exception in center-aligned mode (refer to the CMS bits in the TIM3_CR1 register description). It is cleared by software. 0: No match 1: The content of the counter TIM3_CNT matches the content of the TIM3_CCR1 register. If channel CC1 is configured as input: This bit is set by hardware on a capture. It is cleared by software or by reading the TIM3_CCR1 register. 0: No input capture occurred 1: The counter value has been captured in TIM1_CCR1 register (An edge has been detected on IC1 which matches the selected polarity) Note: When CEN is enabled, this bit will also be set.	0
0	UIF	Update interrupt flag This bit is set by hardware on an update event. It is cleared by software. 0: No update occurred. 1: Update interrupt pending. This bit is set by hardware when the registers are updated: – At overrun or underrun and if UDIS=0 in the TIM3_CR1 register. – When CNT is reinitialized by software using the UG bit in TIM3_EGR register, if URS=0 and UDIS=0 in the TIM3_CR1 register. – When CNT is reinitialized by a trigger event (refer to the synchro control register description), if URS=0 and UDIS=0 in the TIM3_CR1 register.	0

7.5.6 TIM3 Event Generation Register (TIM3_EGR)

Bit	Symbol	Description	Reset value
31:7	--	Reserved, set to 0.	0x0
6	TG	Trigger generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0: No action 1: The TIF flag is set in TIM1_SR register. Related interrupt or DMA transfer can occur if enabled.	0
5	--	Reserved, set to 0.	0
4	CC4G	CC4G: Capture/compare 4 generation Refer to CC1G description	0
3	CC3G	CC3G: Capture/compare 3 generation Refer to CC1G description	0
2	CC2G	CC2G: Capture/compare 2 generation Refer to CC1G description	0
1	CC1G	CC1G: Capture/compare 1 generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0: No action. 1: A capture/compare event is generated on channel 1 If channel CC1 is configured as output: CC1IF flag is set, Corresponding interrupt or DMA request is sent if enabled. If channel CC1 is configured as input: The current value of the counter is captured in TIM3_CCR1 register. The CC1IF flag is set, the corresponding interrupt or DMA request is sent if enabled. The CC1OF flag is set if the CC1IF flag was already high.	0
0	UG	Update generation: (This bit can be set by software, it is automatically cleared by hardware.) 0: No action 1: Re-initialize the counter and generates an update of the registers. Note that the prescaler counter is cleared too (anyway the prescaler ratio is not affected). The counter is cleared if the center-aligned mode is selected or if DIR=0 (upcounting), else it takes the auto-reload value (TIM3_ARR) if DIR=1 (downcounting).	0

7.5.7 TIM3 Capture/Compare Mode Register 1 (TIM3_CCMR1)

Similar to TIM1, the functions of all other bits in this register differ in input mode and output mode.

Output compare mode

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15	OC2CE	Output compare 2 clear enable, refer to OC1CE description	0
14:12	OC2M[2:0]	Output compare 2 mode, refer to OC1M description	0x0
11	OC2PE	Output compare 2 preload enable, refer to OC1PE description	0
10	OC2FE	Output compare 2 fast enable, refer to OC1FE description	0
9:8	CC2S[1:0]	Capture/Compare 2 selection: This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC2 channel is configured as output 01: CC2 channel is configured as input, IC2 is mapped on TI2 10: CC2 channel is configured as input, IC2 is mapped on TI1 11: CC2 channel is configured as input, IC2 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIM3_SMCR register) Note: CC2S bits are writable only when the channel is OFF (CC2E = 0 in TIM3_CCER).	0x0
7	OC1CE	Output compare 1 clear enable: 0: OC1Ref is not affected by the ETRF input 1: OC1Ref is cleared as soon as a High level is detected on ETRF input	0
6:4	OC1M	These bits define the behavior of the output reference signal OC1REF from which OC1 and OC1N are derived. OC1REF is active high whereas OC1 and OC1N active level depends on CC1P and CC1NP bits. 000: Frozen - The comparison between the output compare register TIM3_CCR1 and the counter TIM3_CNT has no effect on the outputs. 001: Set channel 1 to active level on match. OC1REF signal is forced high when the counter TIM3_CNT matches the capture/compare register 1 (TIM3_CCR1). 010: Set channel 1 to inactive level on match. OC1REF signal is forced low when the counter TIM3_CNT matches the capture/compare register 1 (TIM3_CCR1). 011: Toggle - OC1REF toggles when TIM3_CNT=TIM3_CCR1. 100: Force inactive level - OC1REF is forced low. 101: Force active level - OC1REF is forced high. 110: PWM mode 1 - In upcounting, channel 1 is active as long as TIM3_CNT<TIM3_CCR1 else inactive. In downcounting, channel 1 is inactive (OC1REF=0) as long as TIM3_CNT>TIM3_CCR1 else active (OC1REF=1). 111: PWM mode 2 - In upcounting, channel 1 is inactive as long as TIM3_CNT<TIM3_CCR1 else active. In downcounting, channel 1 is active as long as TIM3_CNT>TIM3_CCR1 else inactive. Note: 1: These bits can not be modified as long as LOCK level 3 has been programmed (LOCK bits in TIMx_BDTR register) and CC1S=00 (the channel is configured in output). 2: In PWM mode 1 or 2, the OCREF level changes only when the result of the comparison changes or when the output compare mode switches from “frozen” mode to “PWM” mode.	0x0
3	OC1PE	Output compare 1 preload enable: 0: Preload register on TIM3_CCR1 disabled. TIM3_CCR1 can be written at anytime, the new value is taken in	0

		account immediately. 1: Preload register on TIM3_CCR1 enabled. Read/Write operations access the preload register. TIM3_CCR1 preload value is loaded in the active register at each update event. Note 1: The PWM mode can only be used in single pulse mode without confirmation of the preload register; otherwise, its behavior is uncertain.	
2	OC1FE	Output compare 1 fast enable: This bit is used to accelerate the effect of an event on the trigger in input on the CC output. 0: CC1 behaves normally depending on counter and CCR1 values even when the trigger is ON. The minimum delay to activate CC1 output when an edge occurs on the trigger input is 5 clock cycles. 1: An active edge on the trigger input acts like a compare match on CC1 output. Then, OC is set to the compare level independently from the result of the comparison. Delay to sample the trigger input and to activate CC1 output is reduced to 3 clock cycles. OCFE acts only if the channel is configured in PWM1 or PWM2 mode.	0
1:0	CC1S	Capture/Compare 1 selection: This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC1 channel is configured as output. 01: CC1 channel is configured as input, IC1 is mapped on TI1. 10: CC1 channel is configured as input, IC1 is mapped on TI2. 11: CC1 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS bit (TIMx_SMCR register) Note: CC1S bits are writable only when the channel is OFF (CC1E = 0 in TIMx_CCER).	0x0

Input capture mode

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:12	IC2F	Input capture 2 filter, refer to IC1F description	0x0
11:10	IC2PSC[1:0]	Input capture 2 prescaler, refer to IC1PSC description	0x0
9:8	CC2S	Capture/compare 2 selection: This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC2 channel is configured as output. 01: CC2 channel is configured as input, IC2 is mapped on TI2. 10: CC2 channel is configured as input, IC2 is mapped on TI1. 11: CC2 channel is configured as input, IC2 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS bit (TIM3_SMCR register) Note: CC2S bits are writable only when the channel is OFF (CC2E = 0 in TIM3_CCER).	0x0
7:4	IC1F[3:0]	Input capture 1 filter: This bit-field defines the frequency used to sample TI1 input and the length of the digital filter applied to TI1. The digital filter is made of an event counter in which N consecutive events are needed to validate a transition on the output: 0000: No filter, sampling is done at f_{DTS} 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 1111: $f_{SAMPLING}=f_{DTS}/32$, N=8	0x0
3:2	IC1PSC	Input capture 1 prescaler: This bit-field defines the ratio of the prescaler acting on CC1 input (IC1). The prescaler is reset as soon as CC1E=0 (TIM3_CCER register). 00: no prescaler, capture is done each time an edge is detected on the capture input 01: capture is done once every 2 events 10: capture is done once every 4 events 11: capture is done once every 8 events	0x0
1:0	CC1S	Capture/Compare 1 selection: 00: CC1 channel is configured as output 01: CC1 channel is configured as input, IC1 is mapped on TI1 10: CC1 channel is configured as input, IC1 is mapped on TI2 11: CC1 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS bit (TIM3_SMCR register) Note: CC1S bits are writable only when the channel is OFF (CC1E = 0 in TIM3_CCER).	0x0

7.5.8 TIM3 Capture/Compare Mode Register 2 (TIM3_CCMR2)

Output compare mode

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15	OC4CE	Output compare 4 clear enable, refer to OC1CE description	0
14:12	OC4M[2:0]	Output compare 4 mode, refer to OC1M description	0x0
11	OC4PE	Output compare 4 preload enable, refer to OC1PE description	0
10	OC4FE	Output compare 4 fast enable, refer to OC1FE description	0
9:8	CC4S[1:0]	Capture/Compare 4 selection: This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC4 channel is configured as output 01: CC4 is configured as input, IC4 is mapped on TI4 10: CC4 is configured as input, IC4 is mapped on TI3 11: CC4 channel is configured as input, IC4 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS bit (TIM3_SMCR register) Note: CC4S bits are writable only when the channel is OFF (CC4E = 0 in TIM3_CCER).	0x0
7	OC3CE	Output compare 3 clear enable, refer to OC1CE description	0
6:4	OC3M	Output compare 3 mode, refer to OC1M description	0x0
3	OC3PE	Output compare 3 preload enable, refer to OC1PE description	0
2	OC3FE	Output compare 3 fast enable, refer to OC1FE description	0
1:0	CC3S	Capture/Compare 3 selection: This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC3 channel is configured as output 01: CC3 channel is configured as input, IC3 is mapped on TI3 10: CC3 channel is configured as input, IC3 is mapped on TI34 11: CC3 channel is configured as input, IC3 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS bit (TIM3_SMCR register) Note: CC3S bits are writable only when the channel is OFF (CC3E = 0 in TIM3_CCER).	0x0

Input capture mode

Bit	Symbol	Description	Reset value
31:12	IC4F	Input capture 4 filter, refer to IC1F description	0x0
11:10	IC4PSC	Input capture 4 prescaler, refer to IC1PSC description	0x0
9:8	CC4S[1:0]	Capture/Compare 4 selection: This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC4 channel is configured as output 01: CC4 channel is configured as input, IC4 is mapped on TI4 10: CC4 channel is configured as input, IC4 is mapped on TI3 11: CC4 channel is configured as input, IC4 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS bit (TIM3_SMCR register) Note: CC4S bits are writable only when the channel is OFF (CC4E = 0 in TIM3_CCER).	0x0
7:4	IC3F	Input capture 3 filter: These bits define the sampling frequency of the TI3 input and the length of the digital filter. The digital filter consists of an event counter, which generates an output transition after counting N events. 0000: No filter, sampling is done at the f_{DTS} frequency. 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8	0x0

		1010: $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=5 1011: $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=6 1100: $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=8 1101: $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=5 1110: $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=6 1111: $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=8	
3:2	IC3PSC	Input capture 3 prescaler: This bit field defines the prescaler ratio for the CC3 input (IC3). As long as CC3E = 0 (in the TIM3_CCER register), the prescaler is immediately reset. 00: No prescaler, capture occurs on every edge detected on the capture input. 01: Capture occurs every 2 events. 10: Capture occurs every 4 events. 11: Capture occurs every 8 events.	0x0
1:0	CC3S	Capture/Compare 3 selection: This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC3 channel is configured as output 01: CC3 channel is configured as input, IC3 is mapped on TI3 10: CC3 channel is configured as input, IC3 is mapped on TI4 11: CC3 channel is configured as input, IC3 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS bit (TIM1_SMCR register) Note: CC3S bits are writable only when the channel is OFF (CC3E = 0 in TIM3_CCER).	0x0

7.5.9 TIM3 Counter (TIM3_CNT)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	CNT[15:0]	Counter value	0x0

7.5.10 TIM3 Prescaler (TIM3_PSC)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	PSC[15:0]	Prescaler value The counter clock frequency (CK_CNT) is equal to $f_{CK\_PSC}/(PSC[15:0] + 1)$. PSC contains the value to be loaded into the active prescaler register whenever an update event occurs.	0x0

7.5.11 TIM3 Capture/Compare Enable Register (TIM3_CCER)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15	CC4NP	Capture/Compare 4 output polarity, refer to CC1PN description	0
14	--	Reserved, set to 0.	0
13	CC4P	Capture/Compare 4 output polarity, refer to CC1P description	0
12	CC4E	Capture/Compare 4 output enable, refer to CC1E description	0
11	CC3NP	Capture/Compare 3 complementary output polarity, refer to CC1PN description	0
10	--	Reserved, set to 0.	0
9	CC3P	Capture/Compare 3 output polarity, refer to CC1P description	0
8	CC3E	Capture/Compare 3 output enable, refer to CC1E description	0
7	CC2NP	Capture/Compare 2 complementary output polarity, refer to CC1PN description	0
6	--	Reserved, set to 0.	0
5	CC2P	Capture/Compare 2 output polarity, refer to CC1P description	0
4	CC2E	Capture/Compare 2 output enable, refer to CC1E description	0
3	CC1NP	CC1 channel configured as output: set to 0. CC1 channel configured as input: This bit is used in conjunction with CC1P	0
2	--	Reserved, set to 0.	0
1	CC1P	Capture/Compare 1 output polarity CC1 channel configured as output: 0: OC1 active high 1: OC1 active low CC1 channel configured as input: CC1NP/CC1P bits select TI1FP1 and TI2FP1 polarity for trigger or capture operations. 00: noninverted/rising edge: Circuit is sensitive to TIxFP1 rising edge (capture, trigger in reset, external clock or trigger mode), TIxFP1 is not inverted (trigger in gated mode, encoder mode). inverted/falling edge 01: Circuit is sensitive to TIxFP1 falling edge (capture, trigger in reset, external clock or trigger mode), TIxFP1 is inverted (trigger in gated mode, encoder mode). 10: Reserved, do not use this configuration. 11: noninverted/both edges	0
0	CC1E	Capture/Compare 1 output enable CC1 channel configured as output: 0: Off - OC1 is not active 1: On - OC1 signal is output on the corresponding output pin CC1 channel configured as input: This bit determines if a capture of the counter value can actually be done into the input capture/compare register 1 (TIM3_CCR1) or not. 0: Capture disabled 1: Capture enabled	0

Table 7-3 Output control bit for standard OCx channels

CCx bit	OCx output state
0	Output disabled (OCx=0, OCx_EN=0)
1	OCx=OCxREF+ Polarity, OCx_EN=1)

Note: The state of the external IO pins connected to the standard OCx channels depends on the OCx channel state and the GPIO registers.

7.5.12 TIM3 Auto-Reload Register (TIM3_ARR)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	ARR[15:0]	Low Auto-reload value ARR is the value to be loaded in the actual auto-reload register. Refer to Section Timer-Base Unit: Time-base unit for more details about ARR update and behavior. The counter is blocked while the auto-reload value is null.	0xFFFF

7.5.13 TIM3 Capture/Compare Register 1 (TIM3_CCR1)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	CCR1[15:0]	Capture/Compare 1 value If channel CC1 is configured as output: CCR1 is the value to be loaded in the actual capture/compare 1 register (preload value). It is loaded permanently if the preload feature is not selected in the TIM3_CCMR1 register (bit OC1PE). Otherwise the preload value is copied in the active capture/compare 1 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIM3_CNT and signaled on OC1 output. If channel CC1 is configured as input: CCR1 is the counter value transferred by the last input capture 1 event (IC1).	0x0

7.5.14 TIM3 Capture/Compare Register 2 (TIM3_CCR2)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	CCR2[15:0]	Capture/Compare 2 value If channel CC2 is configured as output: CCR2 is the value to be loaded in the actual capture/compare 2 register (preload value). It is loaded permanently if the preload feature is not selected in the TIM3_CCMR1 register (bit OC1PE). Otherwise the preload value is copied in the active capture/compare 1 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIM3_CNT and signaled on OC2 output. If channel CC2 is configured as input: CCR2 is the counter value transferred by the last input capture 2 event (IC2).	0x0

7.5.15 TIM3 Capture/Compare Register 3 (TIM3_CCR3)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	CCR3[15:0]	Capture/Compare 3 value If channel CC3 is configured as output: CCR3 is the value to be loaded in the actual capture/compare 3 register (preload value). It is loaded permanently if the preload feature is not selected in the TIM3_CCMR2 register (bit OC3PE). Otherwise the preload value is copied in the active capture/compare 1 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on OC3 output. If channel CC3 is configured as input: CCR3 is the counter value transferred by the last input capture 3 event (IC3).	0x0

7.5.16 TIM3 Capture/Compare Register 4 (TIM3_CCR4)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	CCR4[15:0]	Capture/Compare 4 value If channel CC4 is configured as output: CCR4 is the value to be loaded in the actual capture/compare 4 register (preload value). It is loaded permanently if the preload feature is not selected in the TIM3_CCMR2 register (bit OC4PE). Otherwise the preload value is copied in the active capture/compare 1 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIM3_CNT and signaled on OC4 output. If channel CC4 is configured as input: CCR4 is the counter value transferred by the last input capture 4 event (IC4).	0x0

7.5.17 TIM3 DMA Control Register (TIM3_DCR)

Bit	Symbol	Description	Reset value
31:13	--	Reserved	0x0
12:8	DBL	DMA burst length This 5-bit vector defines the number of DMA transfers (the timer recognizes a burst transfer when a read or a write access is done to the TIM3_DMAR address). 00000: 1 transfer 00001: 2 transfers 00010: 3 transfers 00011: 4 transfers: 10001: 18 transfers	0x0
7:5	--	Reserved	0x0
4:0	DBA	DMA base address This 5-bit vector defines the base-address for DMA transfers (when read/write access are done through the TIMx_DMAR address). DBA is defined as an offset starting from the address of the TIM3_CR1 register. 00000: TIM1_CR1 00001: TIM1_CR2 00010: TIM1_SCMR:	0x0

7.5.18 TIM3 Full Transfer DMA Address (TIM3_DMAR)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	DMAB	DMA register for burst accesses A read or write operation to the DMAR register accesses the register located at the address $(\text{TIM3\_CR1 address}) + (\text{DBA} + \text{DMA index}) \times 4$ where TIM3_CR1 address is the address of the control register 1, DBA is the DMA base address configured in TIM3_DCR register, DMA index is automatically controlled by the DMA transfer, and ranges from 0 to DBL (DBL configured in TIM3_DCR).	0x0

Chapter 8 General-Purpose Timer (TIM14)

8.1 TIM14 Introduction

The TIM14 general-purpose timer consists of a 16-bit auto-reload counter driven by a programmable prescaler.

It may be used for a variety of purposes, including measuring the pulse lengths of input signals (input capture) or generating output waveforms (output compare, PWM).

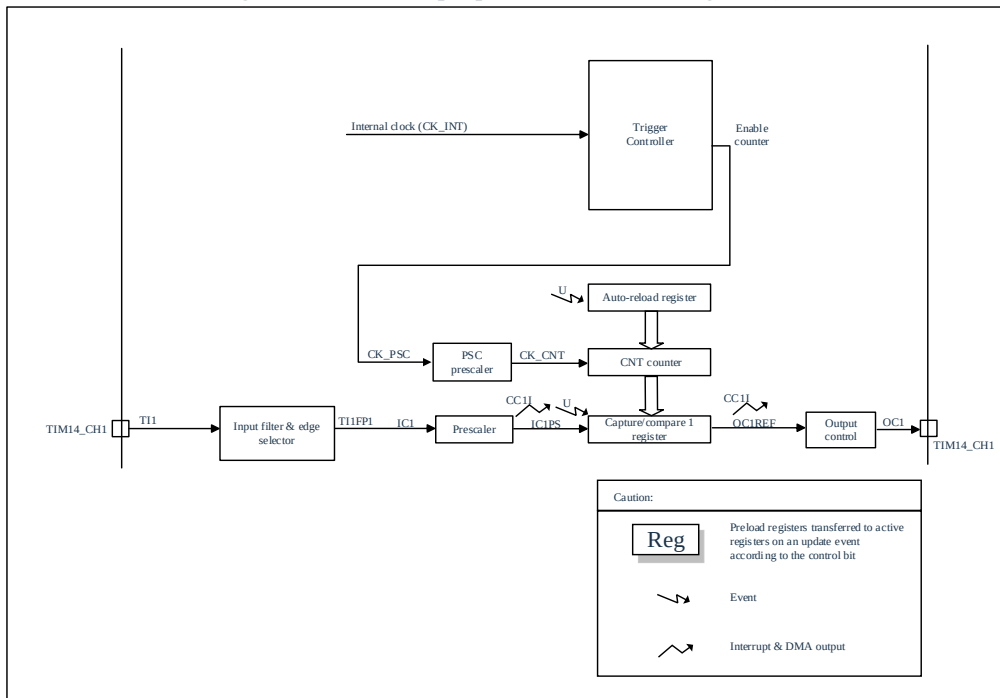
Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the RCC clock controller prescalers.

8.2 TIM14 Main Features

TIM14 has the following features:

- 16-bit auto-reload upcounter
- 16-bit programmable prescaler used to divide the counter clock frequency by any factor between 1 and 65535 (can be changed “on the fly”)
- independent channel for:
 - Input capture
 - Output compare
 - PWM generation (edge-aligned mode)
 - One pulse mode output
- Interrupt generation on the following events:
 - Update: counter overrun, counter initialization (by software)
 - Input capture
 - Output compare

Figure 8-1 General-purpose timer block diagram (TIM14)



8.3 TIM14 Functional Description

8.3.1 Time-Base Unit

The main block of the programmable general-purpose timer is a 16-bit upcounter with its related auto-reload register. The counter clock can be divided by a prescaler.

The counter, the auto-reload register and the prescaler register can be written or read by software. This is true even when the counter is running.

The time-base unit includes:

- Counter register (TIM14_CNT)
- Prescaler register (TIM14_PSC)
- Auto-reload register (TIM14_ARR)

The auto-reload register is preloaded. Writing to or reading from the auto-reload register accesses the preload register. The content of the preload register are transferred into the shadow register permanently or at each update event (UEV), depending on the auto-reload preload enable bit (ARPE) in TIM14_CR1 register. The update event is sent when the counter reaches the overrun and if the UDIS bit equals 0 in the TIM14_CR1 register. It can also be generated by software. The generation of the update event is described in detailed for each configuration.

The counter is clocked by the prescaler output CK_CNT, which is enabled only when the counter enable bit (CEN) in TIM14_CR1 register is set (refer also to the slave mode controller description to get more details on counter enabling).

Note that the counter starts counting 1 clock cycle after setting the CEN bit in the TIM14_CR1 register.

8.3.1.1 Prescaler Description

The prescaler can divide the counter clock frequency by any factor between 1 and 65536. It is based on a 16-bit counter controlled through a 16-bit register (in the TIM14_PSC register). It can be changed on the fly as this control register is buffered. The new prescaler ratio is taken into account at the next update event.

Figures below give some examples of the counter behavior when the prescaler ratio is changed on the fly.

Figure 8-2 Counter timing diagram with prescaler division change from 1 to 2

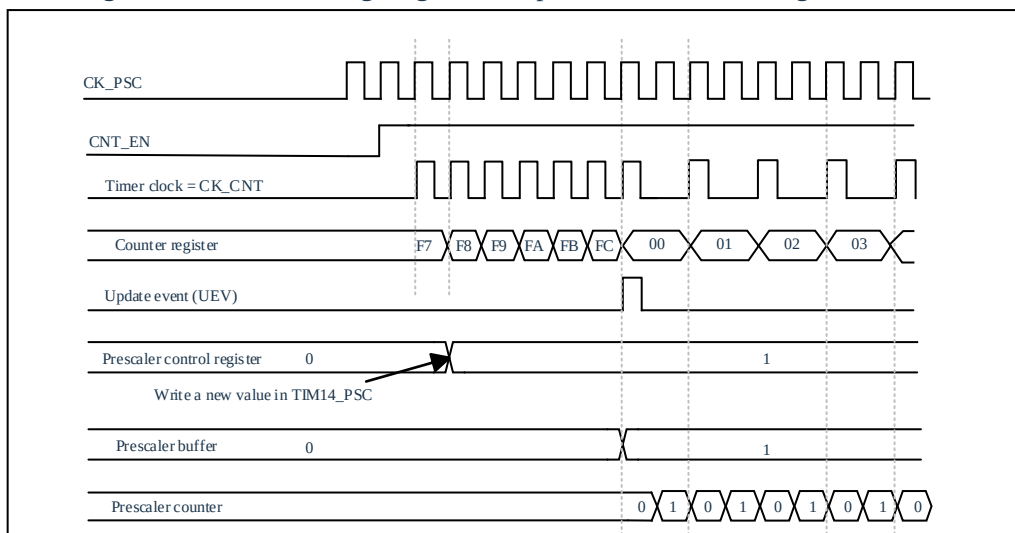
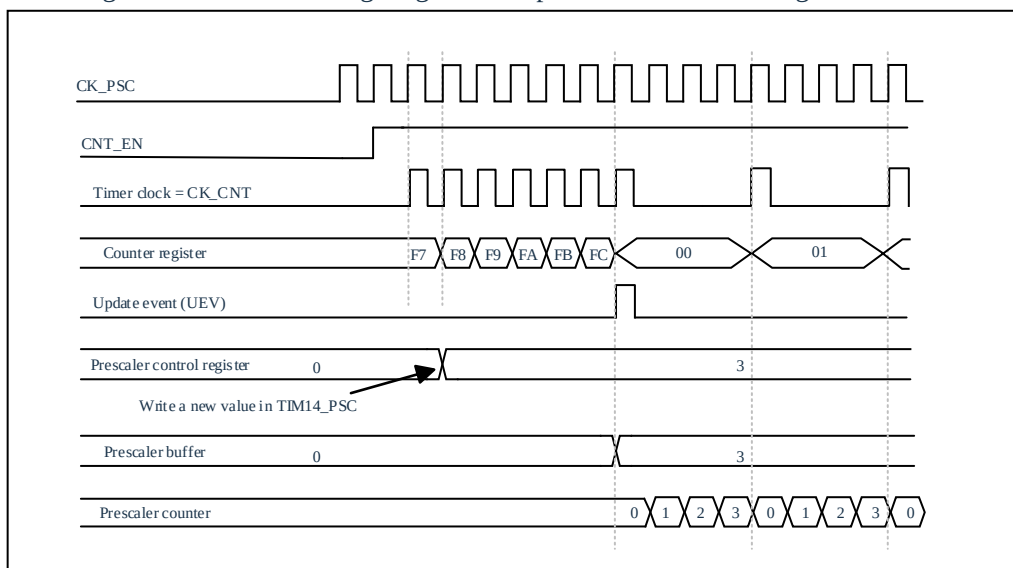


Figure 8-3 Counter timing diagram with prescaler division change from 1 to 4



8.3.2 Counter Modes

8.3.2.1 Up-Counting Mode

The counter counts from 0 to the auto-reload value (content of the TIM14_ARR register), then restarts from 0 and generates a counter overrun event.

Setting the UG bit in the TIM14_EGR register also generates an update event.

The UEV event can be disabled by software by setting the UDIS bit in the TIM14_CR1 register. This is to avoid updating the shadow registers while writing new values in the preload registers. Then no update event occurs until the UDIS bit has been written to 0. However, the counter restarts from 0, as well as the counter of the prescaler (but the prescale rate does not change). In addition, if the URS bit (update request selection) in TIM14_CR1 register is set, setting the UG bit generates an update event UEV but without setting the UIF flag (thus no interrupt is sent). This is to avoid generating both update and capture interrupts when clearing the counter on the capture event.

When an update event occurs, all the registers are updated and the update flag (UIF bit in TIM14_SR register) is set (depending on the URS bit):

- The auto-reload shadow register is updated with the preload value (TIM14_ARR),
- The buffer of the prescaler is reloaded with the preload value (content of the TIM14_PSC register).

The following figures show some examples of the counter behavior for different clock frequencies when TIM14_ARR=0x36.

Figure 8-4 Counter timing diagram, internal clock divided by 1

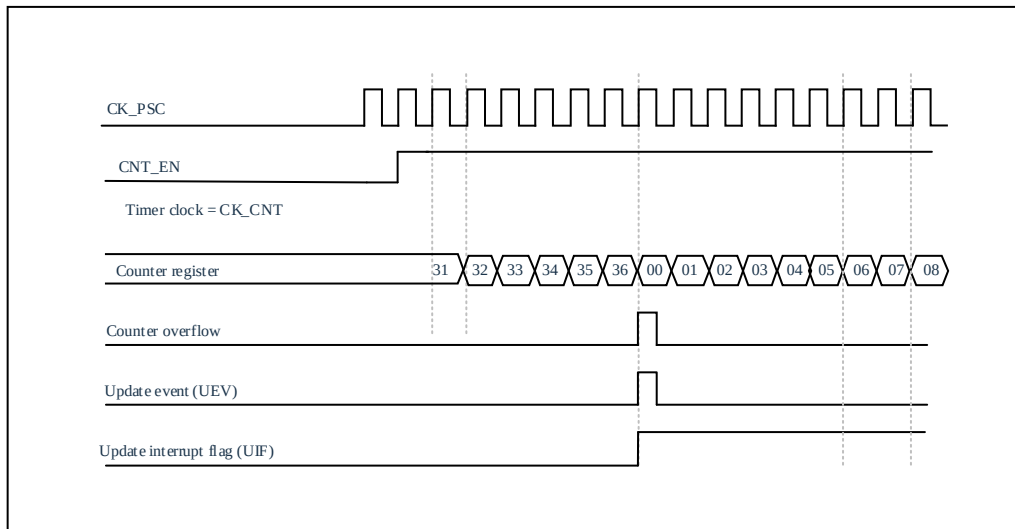


Figure 8-5 Counter timing diagram, internal clock divided by 2

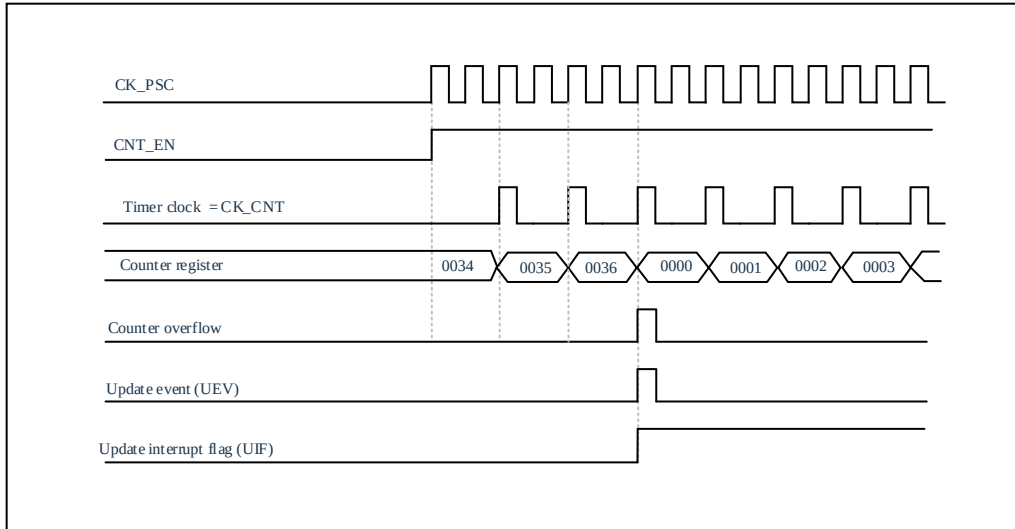


Figure 8-6 Counter timing diagram, internal clock divided by 4

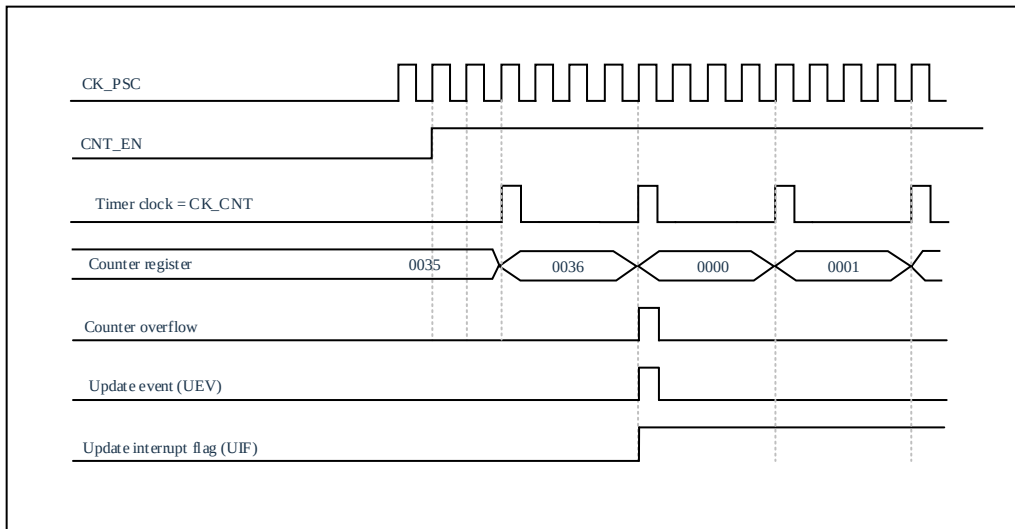


Figure 8-7 Counter timing diagram, internal clock divided by N

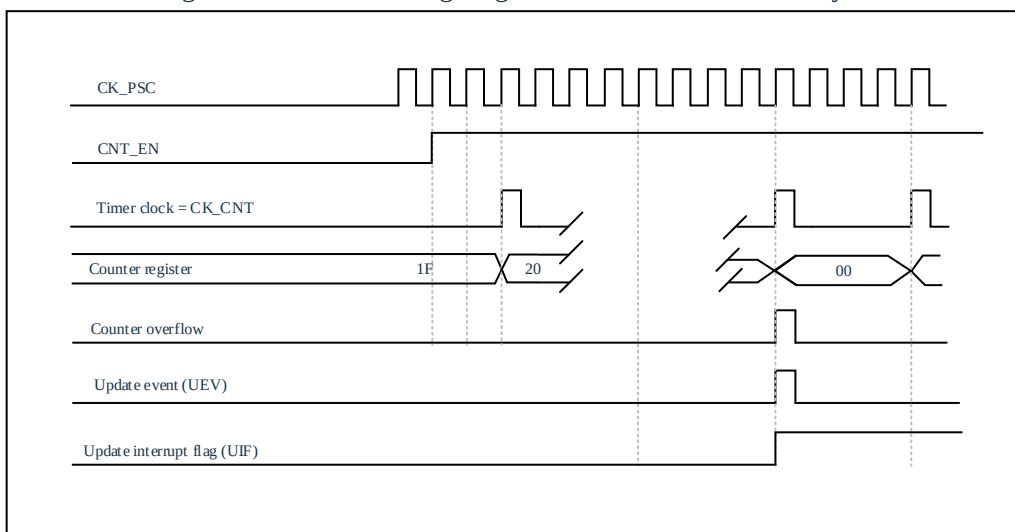


Figure 8-8 Counter timing diagram, update event when ARPE=0 (TIM14_ARR not preloaded)

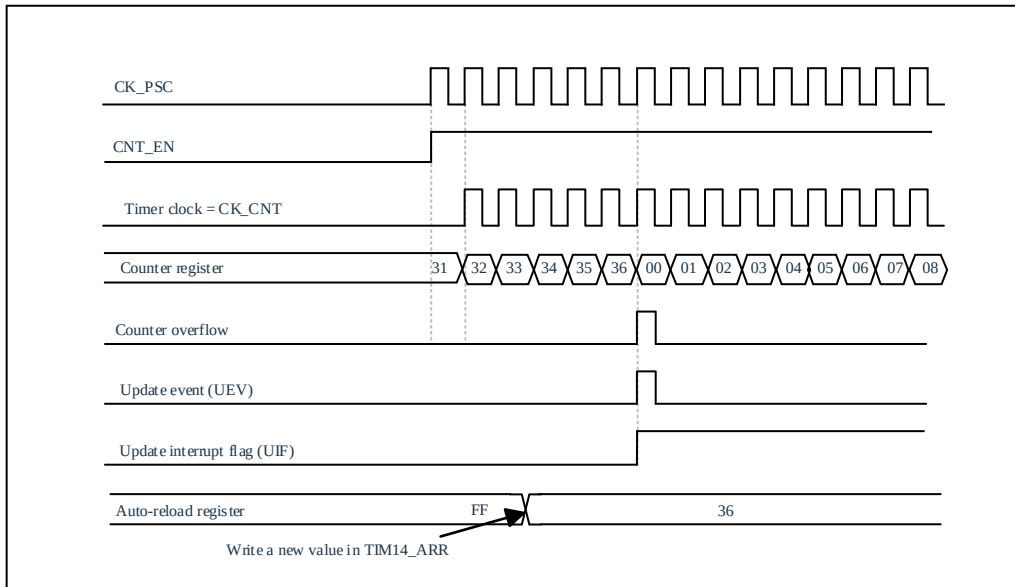
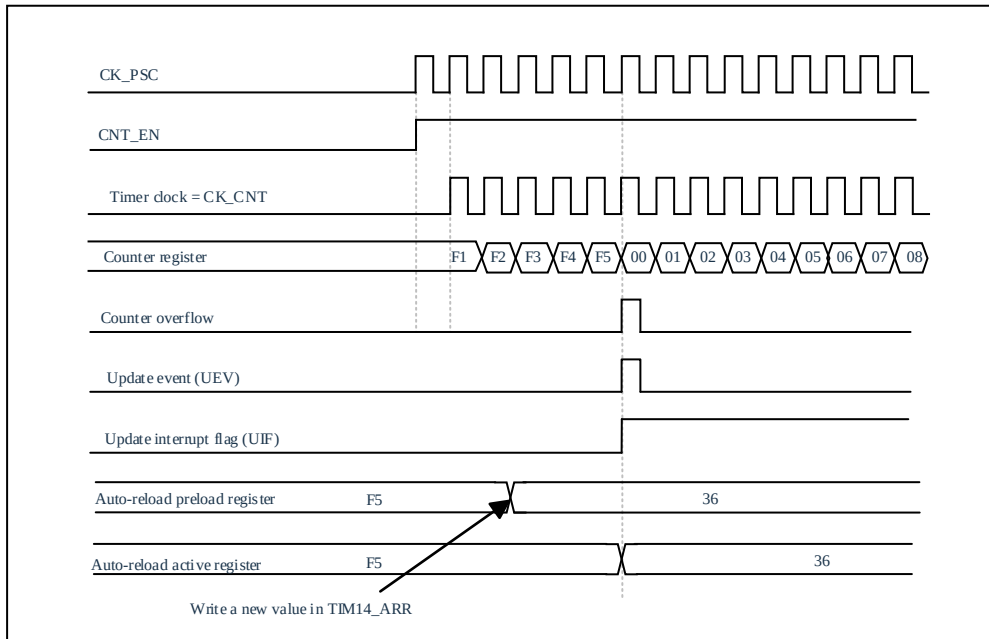


Figure 8-9 Counter timing diagram, update event when ARPE=1 (TIM14_ARR preloaded)

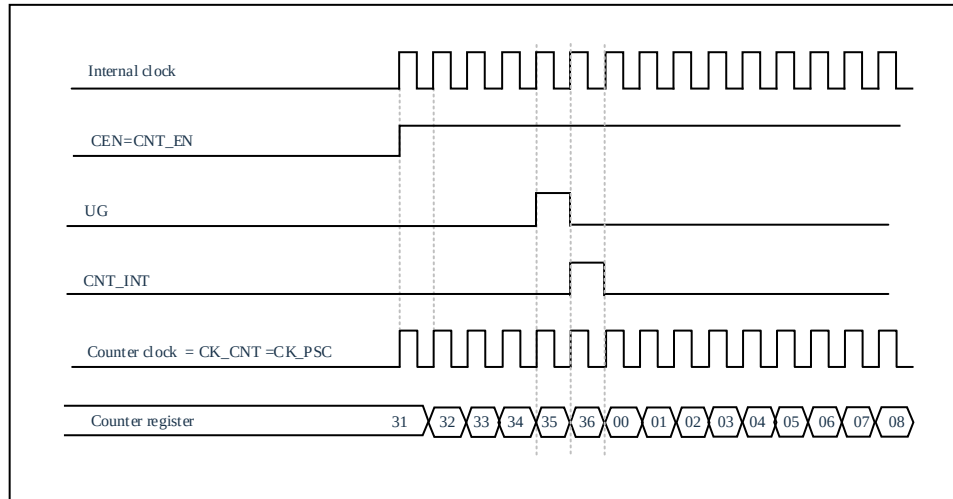


8.3.3 Clock Selection

The counter clock is provided by the Internal clock (CK_INT) source. The CEN (in the TIM14_CR1 register) and UG bits (in the TIM14_EGR register) are actual control bits and can be changed only by software (except for UG that remains cleared automatically). As soon as the CEN bit is written to 1, the prescaler is clocked by the internal clock CK_INT.

Figure 8-10 shows the behavior of the control circuit and the upcounter in normal mode, without prescaler.

Figure 8-10 Control circuit in normal mode, internal clock divided by 1



8.3.4 Capture/Compare Channels

Each Capture/Compare channel is built around a capture/compare register (including a shadow register), a input stage for capture (with digital filter, multiplexing and prescaler) and an output stage (with comparator and output control).

Figure 8-11 to Figure 8-13 give an overview of one capture/compare channel.

The input stage samples the corresponding TI1 input to generate a filtered signal TI1F. Then, an edge detector with polarity selection generates a signal (TI1FP1) which can be used as trigger input by the slave mode controller or as the capture command. It is prescaled before the capture register (IC1PS).

Figure 8-11 Capture/compare channel (example: channel 1 input stage)

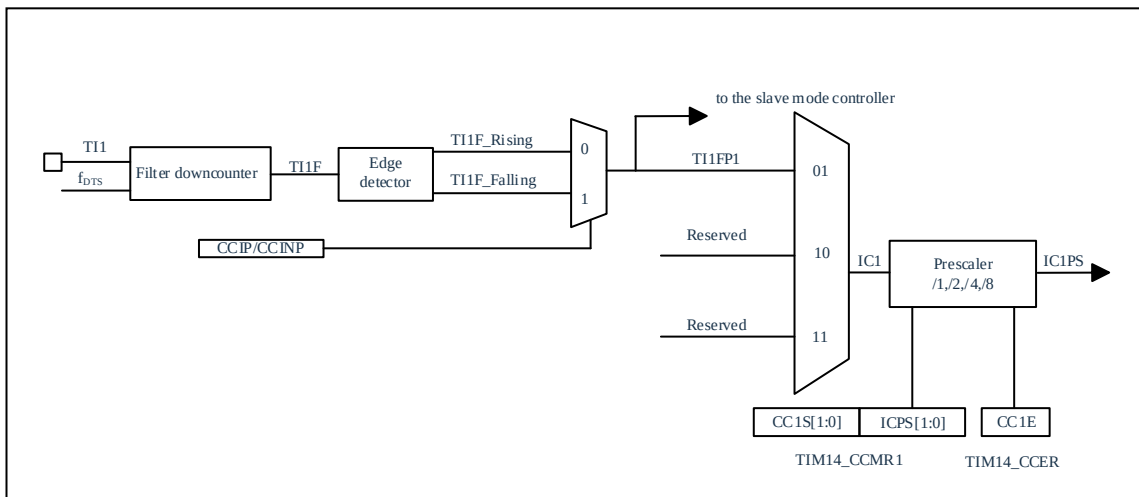


Figure 8-12 Capture/compare channel 1 main circuit

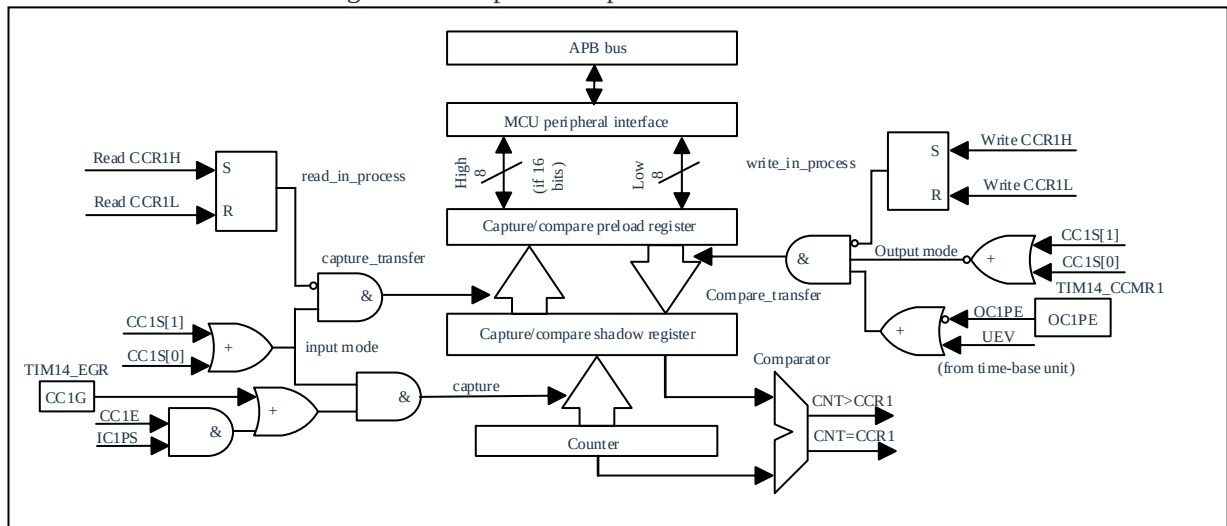
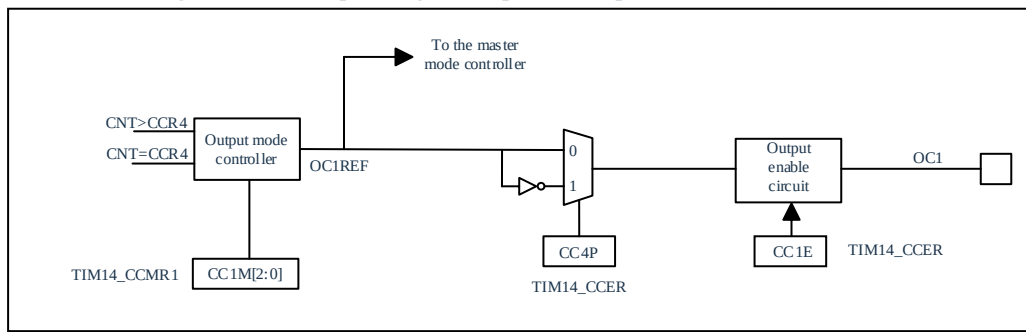


Figure 8-13 Output stage of capture/compare channel (channel 1)



The capture/compare block is made of one preload register and one shadow register. Write and read always access the preload register.

In capture mode, captures are actually done in the shadow register, which is copied into the preload register.

In compare mode, the content of the preload register is copied into the shadow register which is compared to the counter.

8.3.5 Input Capture Mode

In Input capture mode, the Capture/Compare Registers (TIM14_CCR1) are used to latch the value of the counter after a transition detected by the corresponding IC1 signal. When a capture occurs, the corresponding CC1IF flag (TIM14_SR register) is set and an interrupt or a DMA request can be sent if they are enabled. If a capture occurs while the CC1IF flag was already high, then the over-capture flag CC1OF (TIM14_SR register) is set. CC1IF can be cleared by software by writing it to '0' or by reading the captured data stored in the TIM14_CCR1 register. CC1OF is cleared when it is written with 0.

The following example shows how to capture the counter value in TIM14_CCR1 when TI1 input rises. To do this, use the following procedure:

1. Select the active input: TIM14_CCR1 must be linked to the TI1 input, so write the CC1S bits to '01' in the TIM14_CCMR1 register. As soon as CC1S becomes different from '00', the channel is configured in input mode and the TIM14_CCR1 register becomes read-only.
2. Based on the characteristics of the input signal, configure the input filter to the required width (if the input is one of the TI1 inputs, then program the IC1F bits in the TIM14_CCMR1 register). Assuming the input signal may experience jitter within a maximum of 5 internal clock cycles, the filter time must be set to greater than 5 internal clock cycles. It can be configured to confirm the transition edge on TI1 after detecting 8 consecutive samples with new levels (sampled at the f_{DTS} frequency). Then, write "0011" to the IC1F bits in the TIM14_CCMR1 register.
3. Select the edge of the active transition on the TI1 channel by programming CC1P and CC1NP bits to '00' in the TIM14_CCER register (rising edge in this case).
4. Program the input prescaler. In our example, we wish the capture to be performed at each valid transition, so the prescaler is disabled (write IC1PS bits to '00' in the TIM14_CCMR1 register).
5. Enable capture from the counter into the capture register by setting the CC1E bit in the TIM14_CCER register. If needed, enable the related interrupt request by setting the CC1IE bit in the TIM14_DIER register.

When an input capture occurs:

- The TIM14_CCR1 register gets the value of the counter on the active transition.
- CC1IF flag is set (interrupt flag). CC1OF is also set if at least two consecutive captures occurred whereas the flag was not cleared.
- An interrupt is generated depending on the CC1IE bit.

In order to handle the overcapture, it is recommended to read the data before the overcapture flag. This is to avoid missing an overcapture which could happen after reading the flag and before reading the data.

Note: IC interrupt requests can be generated by software by setting the corresponding CC1G bit in the TIM14_EGR register.

8.3.6 Forced Output Mode

In output mode (CC1S bits = '00' in the TIM14_CCMR1 register), each output compare signal (OC1REF and then OC1) can be forced to active or inactive level directly by software, independently of any comparison between the output compare register and the counter.

To force an output compare signal (OC1REF/OC1) to its active level, one just needs to write '101' in the OC1M bits in the corresponding TIM14_CCMR1 register. Thus OC1REF is forced high (OC1REF is always active high) and OC1 get opposite value to CC1P polarity bit.

For example: CC1P= '0' (OC1 active high) => OC1 is forced to high level.

The OC1REF signal can be forced low by writing the OC1M bits to '100' in the TIM14_CCMR1 register.

The comparison between the TIM14_CCR1 shadow register and the counter is still performed and allows the flag to be set. Interrupt requests can be sent accordingly. This is described in the output compare mode section below.

8.3.7 Output Compare Mode

This function is used to control an output waveform or to indicate when a period of time has elapsed.

When a match is found between the capture/compare register and the counter, the output compare function:

1. Assigns the corresponding output pin to a programmable value defined by the output compare mode (OC1M bits in the TIM14_CCMR1 register) and the output polarity (CC1P bit in the TIM14_CCER register). The output pin can keep its level (OC1M= '000'), be set active (OC1M= '001'), be set inactive (OC1M= '010') or can toggle (OC1M= '011') on match.
2. Sets a flag in the interrupt status register (CC1IF bit in the TIM14_SR register).
3. Generates an interrupt if the corresponding interrupt mask is set (CC1IE bit in the TIM14_DIER register).

The TIM14_CCR1 registers can be programmed with or without preload registers using the OC1PE bit in the TIM14_CCMR1 register.

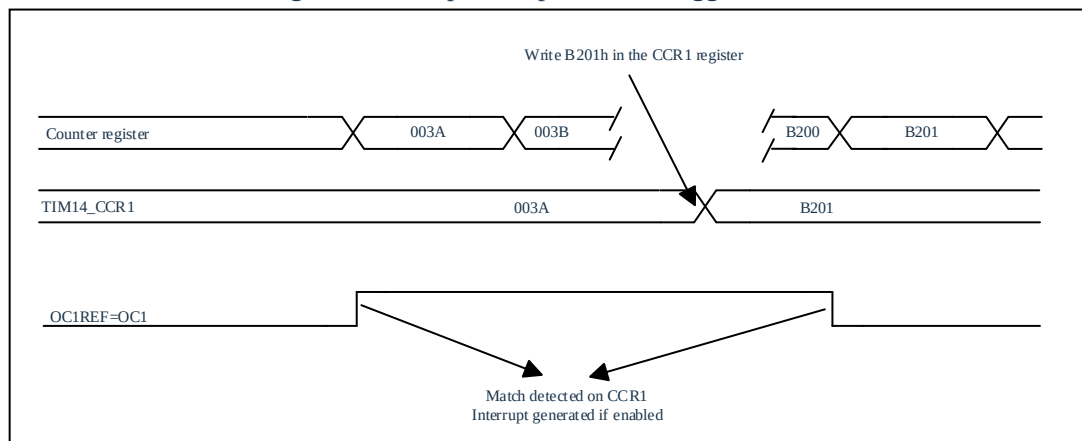
In output compare mode, the update event (UEV) has no effect on the OC1REF and OC1 output. The synchronization accuracy can reach one counter cycle.

Procedure:

1. Select the counter clock (internal, external, prescaler).
2. Write the desired data in the TIM14_ARR and TIM14_CCR1 registers.
3. Set the CC1IE bit if an interrupt request is to be generated.
4. Select the output mode. For example:
 - Write OC1M = '011' to toggle OC1 output pin when CNT matches CCR1
 - Write OC1PE = '0' to disable preload register
 - Write CC1P = '0' to select active high polarity
 - Write CC1E = '1' to enable the output
5. Enable the counter by setting the CEN bit in the TIM14_CR1 register.

The TIM14_CCR1 register can be updated at any time by software to control the output waveform, provided that the preload register is not enabled (OC1PE='0', else TIM14_CCR1 shadow register is updated only at the next update event UEV). An example is given in Figure 8-14.

Figure 8-14 Output compare mode, toggle on OC1



8.3.8 PWM Mode

Pulse Width Modulation mode allows to generate a signal with a frequency determined by the value of the TIM14_ARR register and a duty cycle determined by the value of the TIM14_CCR1 register.

By writing 110 (PWM mode 1) or 111 (PWM mode 2) to the OC1M bits in the TIM14_CCMR1 register, you can independently select the PWM mode for each channel (each OC1 output corresponds to one PWM). The corresponding preload register must be enabled by setting the OC1PE bit in the TIM14_CCMR1 register to 1. Finally, enable the auto-reload preload register by setting the ARPE bit in the TIM14_CR1 register to 1 (in up-counting or center-aligned mode).

As the preload registers are transferred to the shadow registers only when an update event occurs, before starting the counter, all registers must be initialized by setting the UG bit in the TIM14_EGR register.

The OC1 polarity can be programmed using the CC1P bit in the TIM14_CCER register. It can be configured to either active high or active low. The OC1 output is enabled by setting the CC1E bit in the TIM14_CCER register to 1.

In PWM mode (either 1 or 2), TIM14_CNT is always compared with TIM14_CCR1 to determine whether $TIM14\_CNT \leq TIM14\_CCR1$.

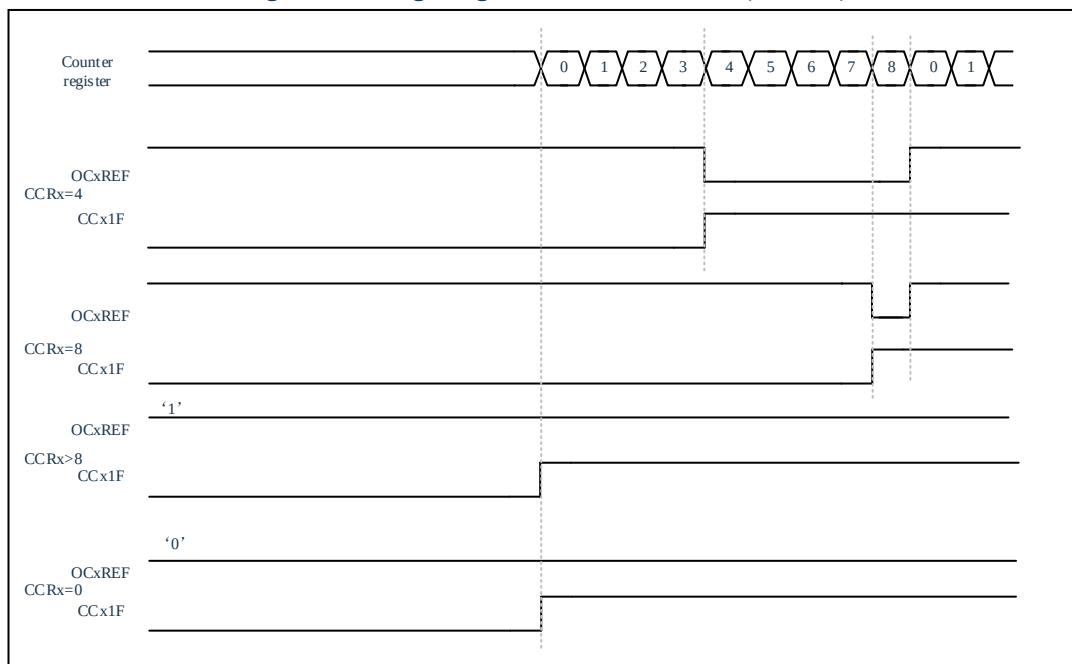
The timer is able to generate PWM in edge-aligned mode only since the counter is upcounting.

8.3.8.1 PWM Edge-Aligned Mode

In the following example, we consider PWM mode 1. The reference PWM signal OC1REF is high as long as $TIM14\_CNT < TIM14\_CCR1$ else it becomes low. If the compare value in TIM14_CCR1 is greater than the auto-reload value (in TIM14_ARR) then OC1REF is held at '1'. If the compare value is 0 then OC1Ref is held at '0'.

Figure 8-15 shows some edge-aligned PWM waveforms in an example where TIM14_ARR=8.

Figure 8-15 Edge-aligned PWM waveforms (ARR=8)



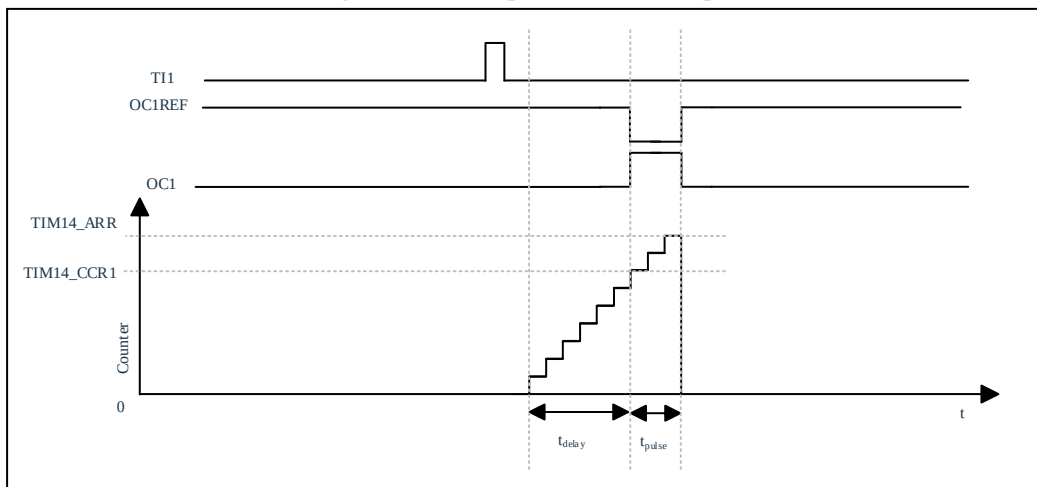
8.3.9 One Pulse Mode

One Pulse Mode (OPM) is a special case of the modes mentioned above. In this mode, the counter can be triggered by an excitation signal and can generate a pulse with a programmable width after a programmable delay.

The counter can be started by the mode controller. A waveform can be generated in either output compare mode or PWM mode. One Pulse Mode can be selected by setting the OPM bit in the TIM14_CR1 register to 1. In this case, the counter will automatically stop when the next update event (UEV) occurs.

A pulse can only be correctly generated when the compare value is different from the initial counter value. Before starting (when the timer is waiting for the trigger), the following configuration must be made: $CNT < CCR1 \leq ARR$ (Note: $0 < CCR1$).

Figure 8-16 One pulse mode example



The OPM waveform is defined by writing to the compare register (considering the clock frequency and counter prescaler).

- t_{DELAY} is defined by the value written to the TIM14_CCR1 register.
- t_{PULSE} is defined by the difference between the auto-reload value and the compare value ($TIM14\_ARR - TIM14\_CCR1$).
- If you want to generate a waveform where the signal changes from 0 to 1 at the compare match and changes back from 1 to 0 when the counter reaches the auto-reload value, you should write $OC1M = "111"$ in the TIM14_CCMR1 register to enable PWM mode 2. If needed, you can choose to enable the preload register by writing 1 to the OC1PE bit in the TIM14_CCMR1 register and the ARPE bit in the TIM14_CR1 register. In this case, the compare value should be written in the TIM14_CCR1 register, and the auto-reload value in the TIM14_ARR register. After generating an update by setting the UG bit to 1, wait for the external trigger event on TI1. In this example, the value of CC1P is 0.

Since only one pulse is needed (one pulse mode), the OPM bit in the TIM14_CR1 register should be set to 1 so that the counter stops counting when the next update event (when the counter returns from the auto-reload value to 0) occurs. When the OPM bit in the TIM14_CR1 register is 0, the repeat mode is selected.

Special Case: OC1 fast enable

In one pulse mode, the edge detection on the TI1 input will set the CEN bit to 1, which enables the counter. Then, when a comparison occurs between the counter value and the compare value, the output will toggle. However, completing these operations requires several clock cycles, which limits the minimum possible delay (minimum t_{DELAY}).

To output a waveform with the shortest possible delay, you can set the OC1FE bit in the TIM14_CCMR1 register to 1. This will force OC1REF (and OC1) to respond to the excitation signal without considering the comparison result. The new level will be the same as the level when the compare match occurs. The OC1FE bit only takes effect when the channel is configured in PWM1 or PWM2 mode.

8.3.10 Debug Mode

When the chip enters debug mode, as per the settings in section 1.4, the TIM14 counter can either continue to operate normally or stop operating.

8.4 Register Mapping

(TIM14 base address = 0x4006_2000)

RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Address	R/W	Description	Reset value
CR1	0x00	R/W	TIM14 Control Register 1	0x00
DIER	0x0C	R/W	TIM14 DMA/Interrupt Enable Register	0x00
SR	0x10	R/W	TIM14 Status Register	0x00
EGR	0x14	R/W	TIM14 Event Generation Register	0x00
CCMR1	0x18	R/W	TIM14 Capture/Compare Mode Register 1	0x00
CCER	0x20	R/W	TIM14 Capture/Compare Enable Register	0x00
CNT	0x24	R/W	TIM14 Counter	0x00
PSC	0x28	R/W	TIM14 Prescaler	0x00
ARR	0x2C	R/W	TIM14 Auto-Reload Register	0xFFFF
CCR1	0x34	R/W	TIM14 Capture/Compare Register 1	0x00
OR	0x50	R/W	TIM14 Option Register	0x00

8.5 Register Description

8.5.1 TIM14 Control Register 1 (TIM14_CR1)

Bit	Symbol	Description	Reset value
31:10	--	Reserved, set to 0.	0x0
9:8	CKD[1:0]	Clock division This bitfield indicates the prescaler ratio between the timer clock (CK_INT) frequency and the dead-time generator as well as the digital filters (ETR, TIX) used by the dead-time and sampling clock (t_{DTS}). 00: $t_{DTS} = t_{CK\_INT}$ 01: $t_{DTS} = 2 * t_{CK\_INT}$ 10: $t_{DTS} = 4 * t_{CK\_INT}$ 11: Reserved, do not set to this value.	0x0
7	ARPE	Auto-reload preload enable 0: TIM14_ARR register is not buffered 1: TIM14_ARR register is buffered	0
6:4	--	Reserved, set to 0.	0x0
3	OPM	One pulse mode: 0: The counter does not stop counting when an update event occurs. 1: The counter stops when the next update event occurs (clearing the CEN bit).	0
2	URS	Update request source This bit is set and cleared by software to select the update interrupt (UEV) sources. 0: If update interrupts or DMA requests are enabled, either of the following events will generate an update interrupt or DMA request: — Counter overrun/underrun — Setting the UG bit 1: If update interrupts or DMA requests are enabled, only counter overrun/underrun will generate an update interrupt or DMA request.	0
1	UDIS	Update disable This bit is set and cleared by software to enable/disable update interrupt (UEV) event generation. 0: UEV enabled. An UEV is generated by one of the following events: — Counter overrun/underrun — Setting the UG bit Buffered registers are then loaded with their preload values. 1: Disable UEV. No update event is generated, and the shadow registers (ARR, PSC, CCRx) retain their values. If the UG bit is set or a hardware reset is issued from the mode controller, the counter and prescaler are reinitialized.	0
0	CEN counter enable	Counter enable 0: Counter disabled 1: Counter enabled Note: After the CEN bit is set by software, the external clock, gated mode, and encoder mode can function. Trigger mode can automatically set the CEN bit through hardware.	0

8.5.2 TIM14 Interrupt Enable Register (TIM14_DIER)

Bit	Symbol	Description	Reset value
31:2	--	Reserved, set to 0.	0x0
1	CC1IE	CC1IE: Capture/Compare 1 interrupt enable 0: CC1 interrupt disabled 1: CC1 interrupt enabled	0
0	UIE	Update interrupt enable 0: Update interrupt disabled 1: Update interrupt enabled	0

8.5.3 TIM14 Status Register (TIM14_SR)

Bit	Symbol	Description	Reset value
31:10	--	Reserved, set to 0.	0x0
9	CC1OF	Capture/Compare 1 overcapture flag This flag is set by hardware only when the corresponding channel is configured in input capture mode. It is cleared by software by writing it to '0'. 0: No overcapture has been detected. 1: The counter value has been captured in TIM14_CCR1 register while CC1IF flag was already set	0
8:2	--	Reserved, set to 0.	0x0
1	CC1IF	Capture/compare 1 interrupt flag Channel CC1 is configured as output: When the counter matches the compare value, this flag is set to 1 by hardware. It is cleared by software. 0: No match. 1: The content of the counter TIM14_CNT matches the content of the TIM14_CCR1 register. Channel CC1 is configured as input: This bit is set to 1 by hardware when a capture event occurs. It is cleared by software or by reading the TIM14_CCR1 register. 0: No input capture occurred. 1: The counter value has been captured in TIM14_CCR1 register (an edge has been detected on IC1 which matches the selected polarity).	0
0	UIF	Update interrupt flag This bit is set by hardware on an update event. It is cleared by software. 0: No update occurred. 1: Update interrupt pending. This bit is set by hardware when the registers are updated: - At overrun and if UDIS= '0' in the TIM14_CR1 register. - When CNT is reinitialized by software using the UG bit in TIM14_EGR register, if URS= '0' and UDIS= '0' in the TIM14_CR1 register.	0

8.5.4 TIM14 Event Generation Register (TIM14_EGR)

Bit	Symbol	Description	Reset value
31:2	--	Reserved, set to 0.	0x0
1	CC1G	Capture/compare 1 generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0: No action. 1: A capture/compare event is generated on channel 1: Channel CC1 is configured as output CC1IF flag is set, the corresponding interrupt and DMA are sent if enabled. Channel CC1 is configured as input The current value of the counter is captured in TIM14_CCR1 register. The CC1IF flag is set, the corresponding interrupt is sent if enabled. The CC1OF flag is set if the CC1IF flag was already high.	0
0	UG	Update generation This bit can be set by software, it is automatically cleared by hardware. 0: No action 1: Re-initialize the counter and generates an update of the registers. Note that the prescaler counter is cleared too (anyway the prescaler ratio is not affected). The counter is cleared.	0

8.5.5 TIM14 Capture/Compare Mode Register 1 (TIM14_CCMR1)

Output compare mode

Bit	Symbol	Description	Reset value
31:7	--	Reserved, set to 0.	0x0
6:4	OC1M	Output compare 1 mode These bits define the behavior of the output reference signal OC1REF for OC1 and OC1N. OC1REF is active high, while the active level for OC1 and OC1N depends on the CC1P and CC1NP bits. 000: Frozen. The comparison between the output compare register TIM14_CCR1 and the counter TIM14_CNT has no effect on the outputs. 001: Set channel 1 to active level on match. OC1REF signal is forced high when the counter TIM14_CNT matches the capture/compare register 1 (TIM14_CCR1). 010: Set channel 1 to inactive level on match. OC1REF signal is forced low when the counter TIM14_CNT matches the capture/compare register 1 (TIM14_CCR1). 011: Toggle - OC1REF toggles when TIM1_CNT = TIM1_CCR1. 100: Force inactive level - OC1REF is forced low. 101: Force active level - OC1REF is forced high. 110: PWM mode 1 - Channel 1 is active as long as TIM14_CNT < TIM14_CCR1 else inactive. 111: PWM mode 2 - Channel 1 is inactive as long as TIM14_CNT < TIM14_CCR1 else active. Note 1: In PWM Mode 1 or PWM Mode 2, the OC1REF level only changes when the compare result changes, or when switching from frozen mode to PWM mode in output compare mode.	0x0
3	OC1PE	Output compare 1 preload enable 0: Preload register on TIM14_CCR1 disabled. TIM14_CCR1 can be written at anytime, the new value is taken in account immediately. 1: Preload register on TIM14_CCR1 enabled. Read/Write operations access the preload register. TIM14_CCR1	0

		preload value is loaded in the active register at each update event. Note 1: The PWM mode can be used without validating the preload register only in one pulse mode. Else the behavior is not guaranteed.	
2	OC1FE	Output compare 1 fast enable This bit is used to accelerate the effect of an event on the trigger in input on the CC output. 0: CC1 behaves normally depending on counter and CCR1 values even when the trigger is ON. The minimum delay to activate CC1 output when an edge occurs on the trigger input is 5 clock cycles. 1: An active edge on the trigger input acts like a compare match on CC1 output. OC is then set to the compare level independently of the result of the comparison. Delay to sample the trigger input and to activate CC1 output is reduced to 3 clock cycles. OC1FE acts only if the channel is configured in PWM1 or PWM2 mode.	0
1:0	CC1S	Capture/Compare 1 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC1 channel is configured as output. 01: CC1 channel is configured as input, IC1 is mapped on TI1. 10: Reserved 11: Reserved Note: CC1S bits are writable only when the channel is OFF (CC1E = 0 in TIM14_CCER).	0

Input capture mode

Bit	Symbol	Description	Reset value
31:8	--	Reserved, set to 0.	0x0
7:4	IC1F[3:0]	Input capture 1 filter This bit-field defines the frequency used to sample TI1 input and the length of the digital filter applied to TI1. The digital filter is made of an event counter in which N events are needed to validate a transition on the output: 0000: No filter, sampling is done at f_{DTS} 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 1111: $f_{SAMPLING}=f_{DTS}/32$, N=8	0x0
3:2	IC1PSC	Input capture 1 prescaler This bit-field defines the ratio of the prescaler acting on CC1 input (IC1). The prescaler is reset as soon as CC1E= '0' (TIM14_CCER register). 00: no prescaler, capture is done each time an edge is detected on the capture input 01: capture is done once every 2 events 10: capture is done once every 4 events 11: capture is done once every 8 events	0x0
1:0	CC1S	Capture/Compare 1 selection 00: CC1 channel is configured as output 01: CC1 channel is configured as input, IC1 is mapped on TI1 10: Reserved 11: Reserved Note: CC1S bits are writable only when the channel is OFF (CC1E = 0 in TIM14_CCER).	0x0

8.5.6 TIM14 Capture/Compare Enable Register (TIM14_CCER)

Bit	Symbol	Description	Reset value
31:4	--	Reserved, set to 0.	0x0
3	CC1NP	Capture/Compare 1 complementary output polarity CC1 channel configured as output: CC1NP must be kept cleared CC1 channel configured as input: CC1NP bit is used in conjunction with CC1P to define TI1FP1 polarity (refer to CC1P description).	0
2	---	Reserved, set to 0.	0
1	CC1P	Capture/Compare 1 output polarity CC1 channel configured as output: 0: OC1 active high 1: OC1 active low CC1 channel configured as input: The CC1NP/CC1P bits can select the active polarity of TI1FP1 for trigger or capture operations. 00: noninverted/rising edge The circuit is sensitive to the rising edge of TI1FP1 (capture or trigger operation in reset mode, external clock mode, or trigger mode). TI1FP1 is non-inverted (trigger operation in gated mode or encoder mode). 01: inverted/falling edge trigger The circuit is sensitive to the falling edge of TI1FP1 (capture or trigger operation in reset mode, external clock mode, or trigger mode). TI1FP1 is inverted (trigger operation in gated mode or encoder mode). 10: Reserved, do not use this configuration 11: non-inverted/edge trigger The circuit is sensitive to both the rising and falling edges of TI1FP1 (capture or trigger operation in reset mode, external clock mode, or trigger mode). TI1FP1 is non-inverted (trigger operation in gated mode).	0
0	CC1E	Capture/Compare 1 output enable CC1 channel configured as output: 0: Off - OC1 is not active 1: On - OC1 signal is output on the corresponding output pin CC1 channel configured as input: This bit determines if a capture of the counter value can actually be done into the input capture/compare register 1 (TIM14_CCR1) or not. 0: Capture disabled 1: Capture enabled	0

Table 8-1 Output control bit for standard OC1 channels

CC1E bit	OC1 output state
0	Output disabled (OC1="0", OC1NE="0")
1	OC1=OC1REF + Polarity, OC1EN = "1"

8.5.7 TIM14 Counter (TIM14_CNT)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	CNT[15:0]	Counter value	0x0

8.5.8 TIM14 Prescaler (TIM14_PSC)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	PSC[15:0]	Prescaler value The counter clock frequency CK_CNT is equal to $f_{CK\_PSC}/(PSC[15:0] + 1)$. PSC contains the value to be loaded in the active prescaler register at each update event.	0x0

8.5.9 TIM14 Auto-Reload Register (TIM14_ARR)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	ARR[15:0]	Auto-reload value ARR is the value to be loaded in the actual auto-reload register. The counter is blocked while the auto-reload value is null.	0xFFFF

8.5.10 TIM14 Capture/Compare Register 1 (TIM14_CCR1)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	CCR1[15:0]	Capture/Compare 1 value Channel CC1 is configured as output CCR1 is the value to be loaded in the actual capture/compare 1 register (preload value). It is loaded permanently if the preload feature is not selected in the TIM14_CCMR1 register (bit OC1PE). Else the preload value is copied in the active capture/compare 1 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIM14_CNT and signaled on OC1 output. Channel CC1 is configured as input CCR1 is the counter value transferred by the last input capture 1 event (IC1).	0x0

8.5.11 TIM14 Option Register (TIM14_OR)

Bit	Symbol	Description	Reset value
31:2	--	Reserved, set to 0.	0x0
1:0	TI1_RMP	Timer input 1 remap Set and cleared by software. 00: TIM14 Channel1 is connected to the GPIO. Refer to the alternate function mapping in the device datasheets. Other: Reserved	0x0

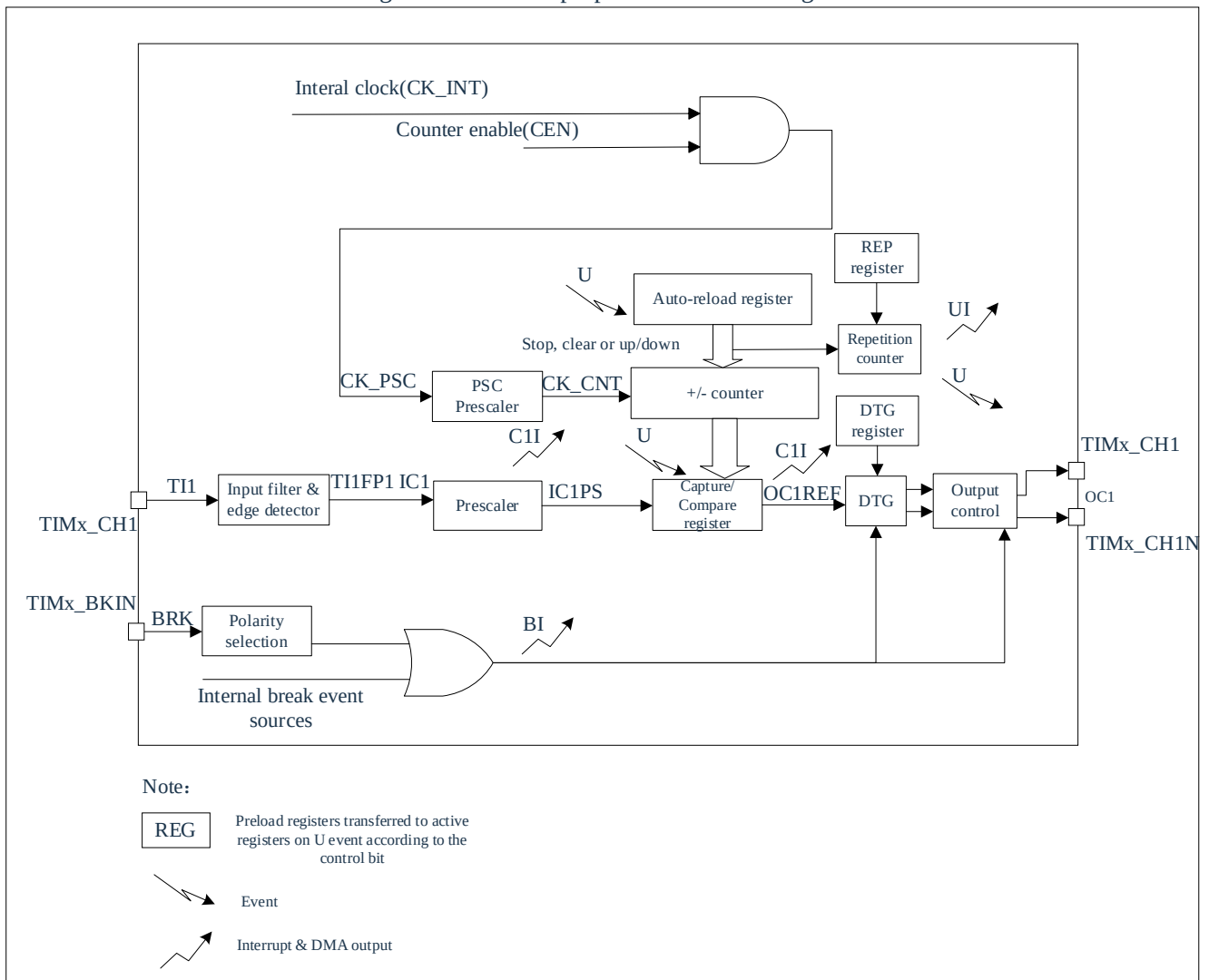
Chapter 9 General-Purpose Timers (TIM16/17)

TIM16 and TIM17 have identical functions.

9.1 TIM16/TIM17 Main Features

- ◆ Bit auto-load upcounter
- ◆ 16-bit programmable prescaler used to divide (also “on the fly”) the counter clock frequency by any factor between 1 and 65535
- ◆ 1 independent channel for:
 - Input capture
 - Output compare
 - PWM generation (Edge-aligned mode)
 - One-pulse mode output
- ◆ Complementary outputs with programmable dead-time
- ◆ Repetition counter to update the timer registers only after a given number of cycles of the counter
- ◆ Break input to put the timer’s output signals in the reset state or a known state
- ◆ Interrupt/DMA generation on the following events:
 - Update: counter overrun
 - Input capture
 - Output compare
 - Break signal input

Figure 9-1 General-purpose timer block diagram



9.2 TIM16/TIM17 Functional Description

9.2.1 Time-Base Unit

The main block of the programmable general purpose timer is a 16-bit upcounter with its related auto-reload register. The counter clock can be divided by a prescaler.

The counter, the auto-reload register and the prescaler register can be written or read by software. This is true even when the counter is running. The time-base unit includes:

- Counter register (TIMx_CNT)
- Prescaler register (TIMx_PSC)
- Auto-reload register (TIMx_ARR)
- Repetition counter register (TIMx_RCR)

The auto-reload register is preloaded. Writing to or reading from the auto-reload register accesses the preload register. The content of the preload register are transferred into the shadow register permanently or at each update event (UEV), depending on the auto-reload preload enable bit (ARPE) in TIMx_CR1 register. The update event is sent when the counter reaches the overrun (or underrun when downcounting) and if the UDIS bit equals 0 in the TIMx_CR1 register. It can also be generated by software.

The counter is clocked by the prescaler output CK_CNT, which is enabled only when the counter enable bit (CEN) in TIMx_CR1 register is set.

Note that the counter starts counting 1 clock cycle after setting the CEN bit in the TIMx_CR1 register.

Prescaler description:

The prescaler can divide the counter clock frequency by any factor between 1 and 65536. It is based on a 16-bit counter controlled through a 16-bit register (in the TIMx_PSC register). It can be changed on the fly as this control register is buffered. The new prescaler ratio is taken into account at the next update event.

The figure below give some examples of the counter behavior when the prescaler ratio is changed on the fly:

Figure 9-2 Counter timing diagram with prescaler division change from 1 to 2

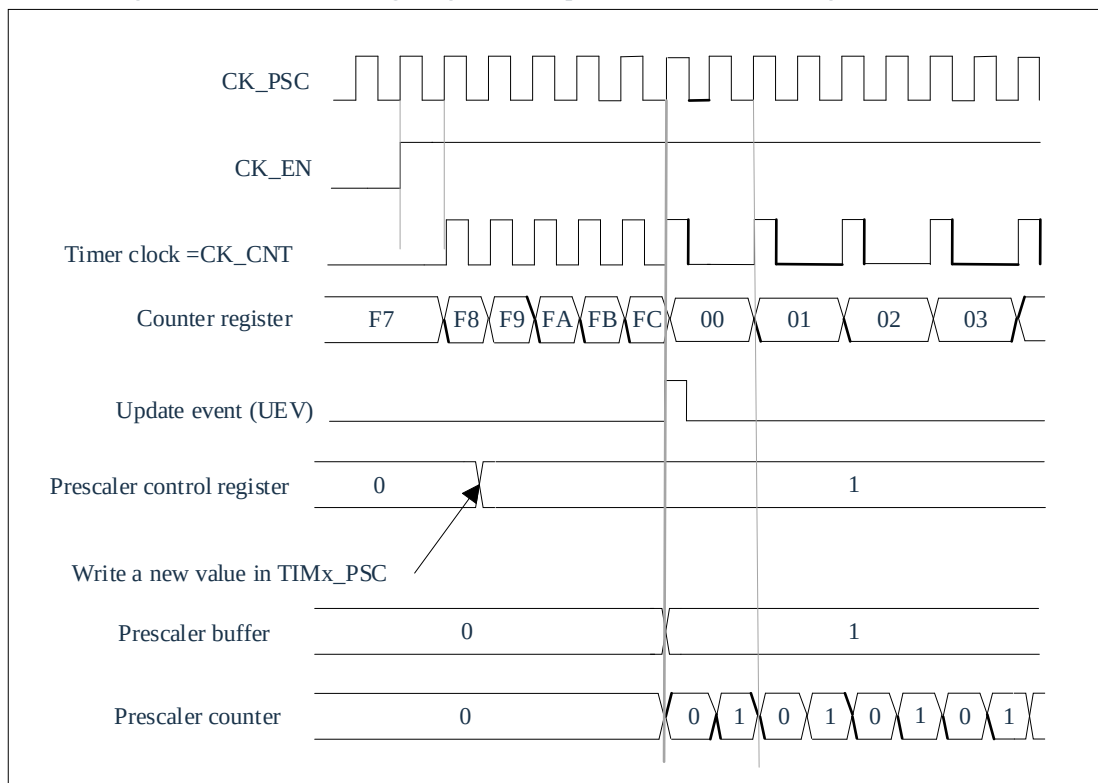
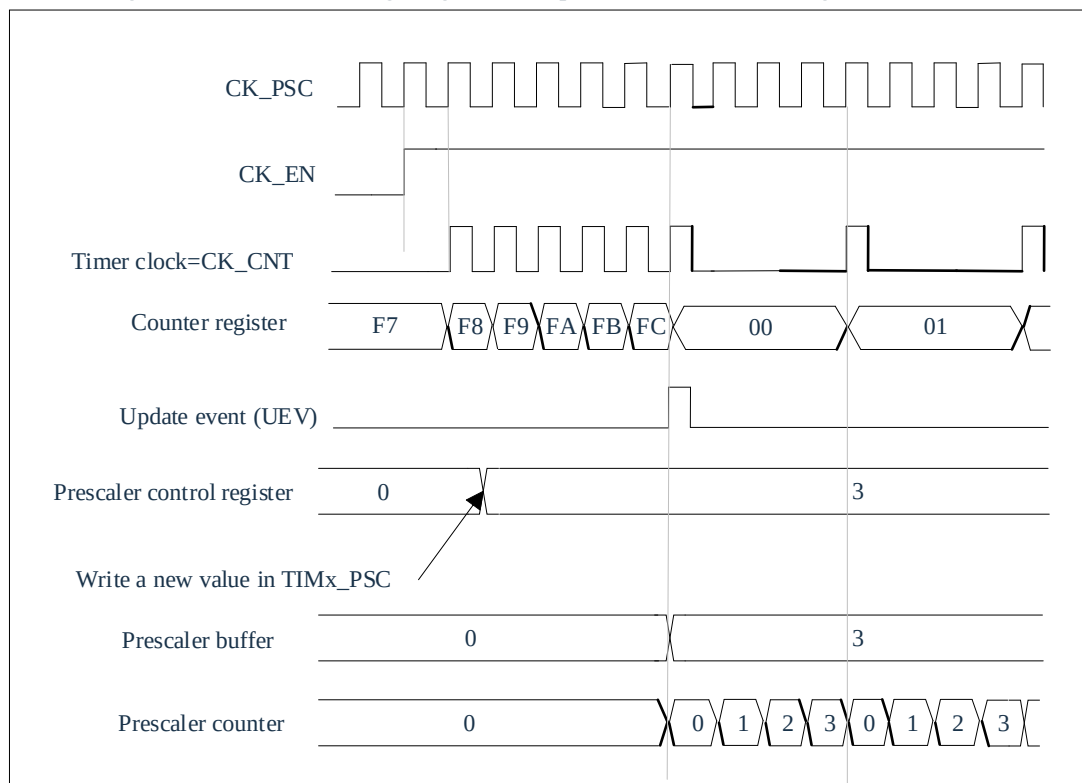


Figure 9-3 Counter timing diagram with prescaler division change from 1 to 4



9.2.2 Counter Modes

The counter counts from 0 to the auto-reload value (content of the TIMx_ARR register), then restarts from 0 and generates a counter overflow event.

If the repetition counter is used, the update event (UEV) is generated after upcounting is repeated for the number of times programmed in the repetition counter register. Else the update event is generated at each counter overflow.

Setting the UG bit in the TIMx_EGR register (by software or by using the slave mode controller) also generates an update event. The UEV event can be disabled by software by setting the UDIS bit in the TIMx_CR1 register. This is to avoid updating the shadow registers while writing new values in the preload registers. Then no update event occurs until the UDIS bit has been written to 0. However, the counter restarts from 0, as well as the counter of the prescaler (but the prescale rate does not change). In addition, if the URS bit (update request selection) in TIMx_CR1 register is set, setting the UG bit generates an update event UEV but without setting the UIF flag (thus no interrupt or DMA request is sent). This is to avoid generating both update and capture interrupts when clearing the counter on the capture event. When an update event occurs, all the registers are updated and the update flag (UIF bit in TIMx_SR register) is set (depending on the URS bit):

- The repetition counter is reloaded with the content of TIMx_RCR register,
- The auto-reload shadow register is updated with the preload value (TIMx_ARR),
- The buffer of the prescaler is reloaded with the preload value (content of the TIMx_PSC register).

The following figures show some examples of the counter behavior for different clock frequencies when TIMx_ARR=0x36.

Figure 9-4 Counter timing diagram, internal clock divided by 1

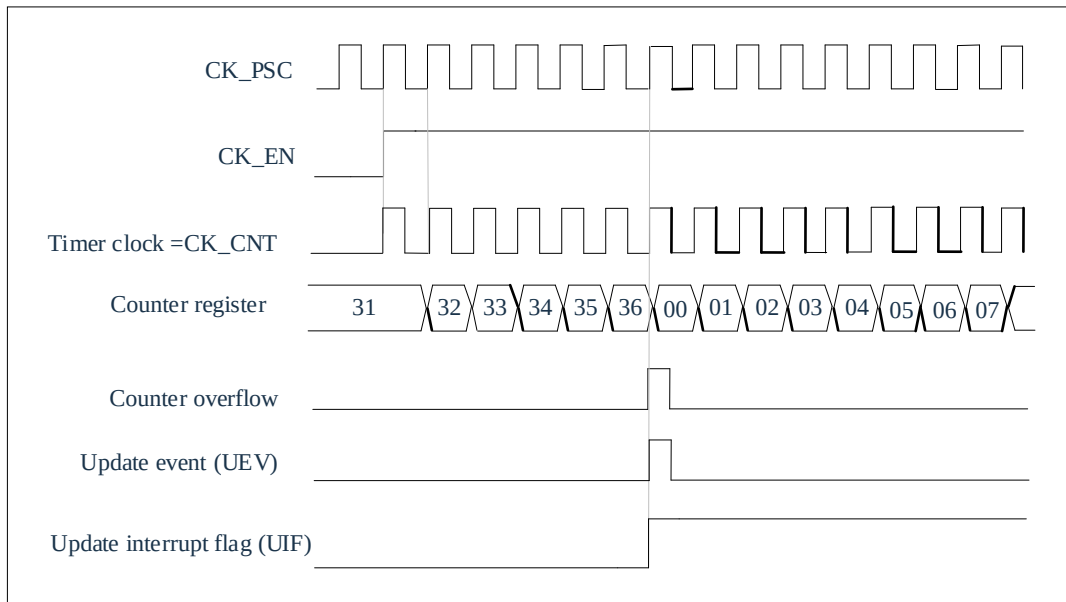


Figure 9-5 Counter timing diagram, internal clock divided by 2

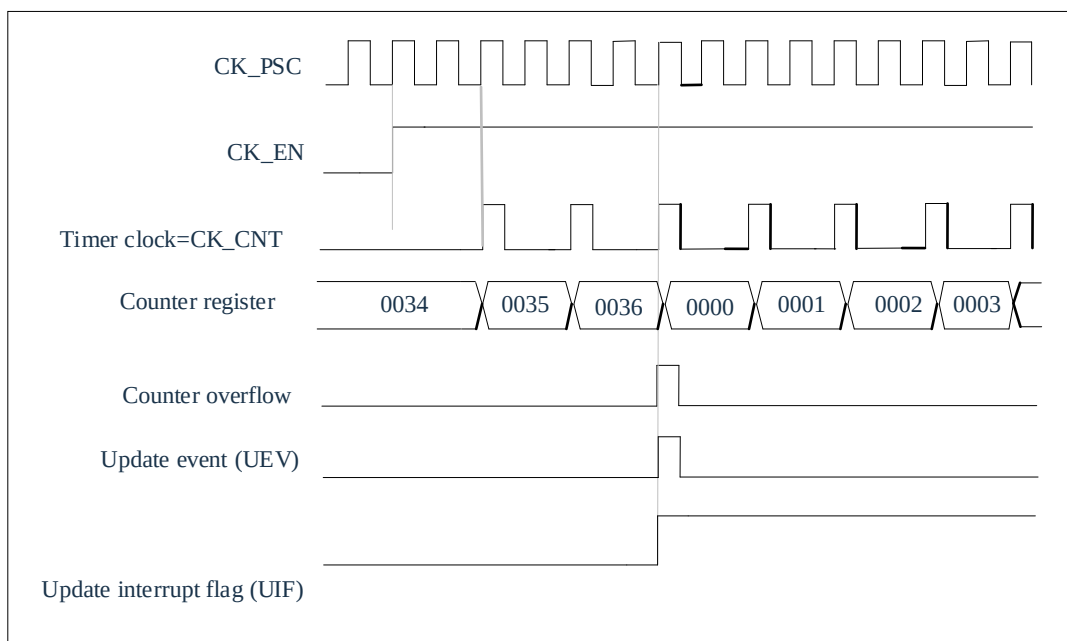


Figure 9-6 Counter timing diagram, internal clock divided by 4

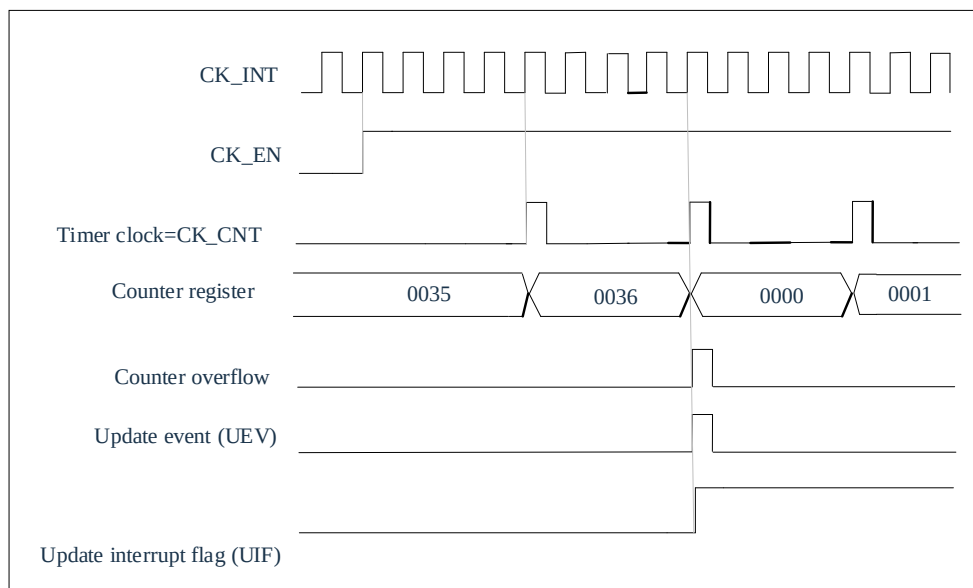


Figure 9-7 Counter timing diagram, internal clock divided by N

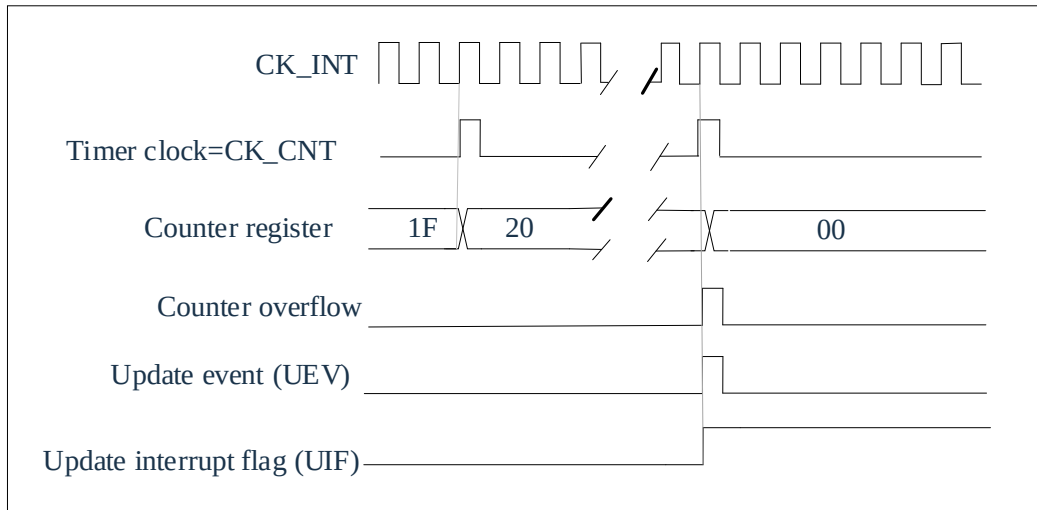


Figure 9-8 Counter timing diagram, update event when ARPE=0 (TIMx_ARR not preloaded)

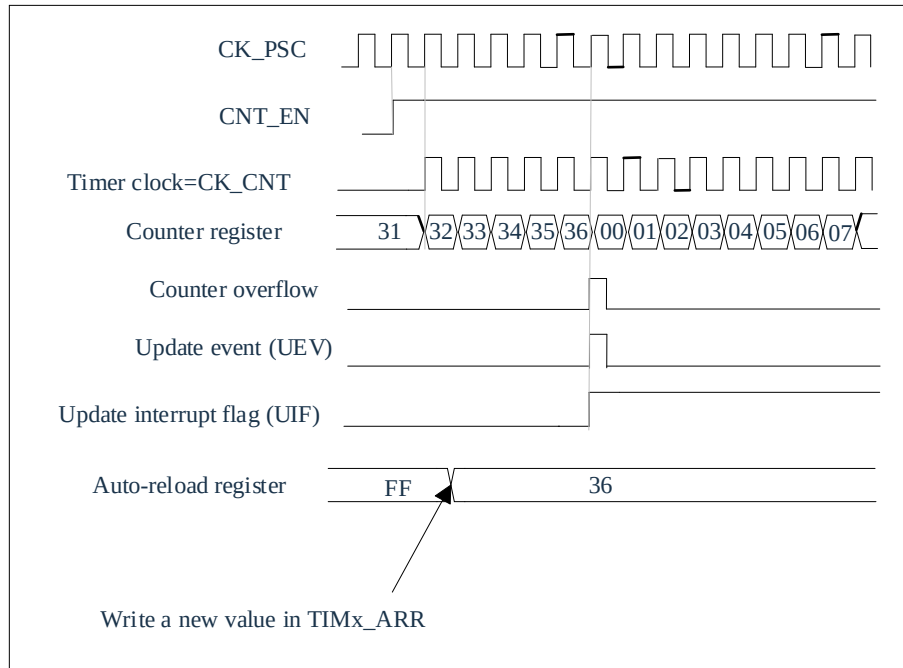
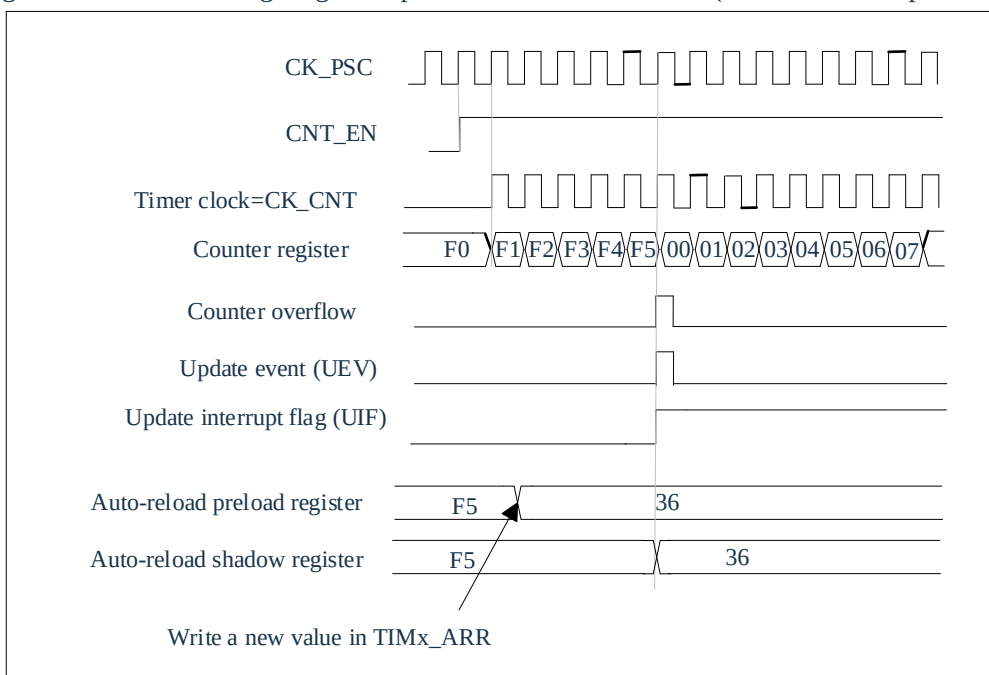


Figure 9-9 Counter timing diagram, update event when ARPE=1 (TIMx_ARR not preloaded)



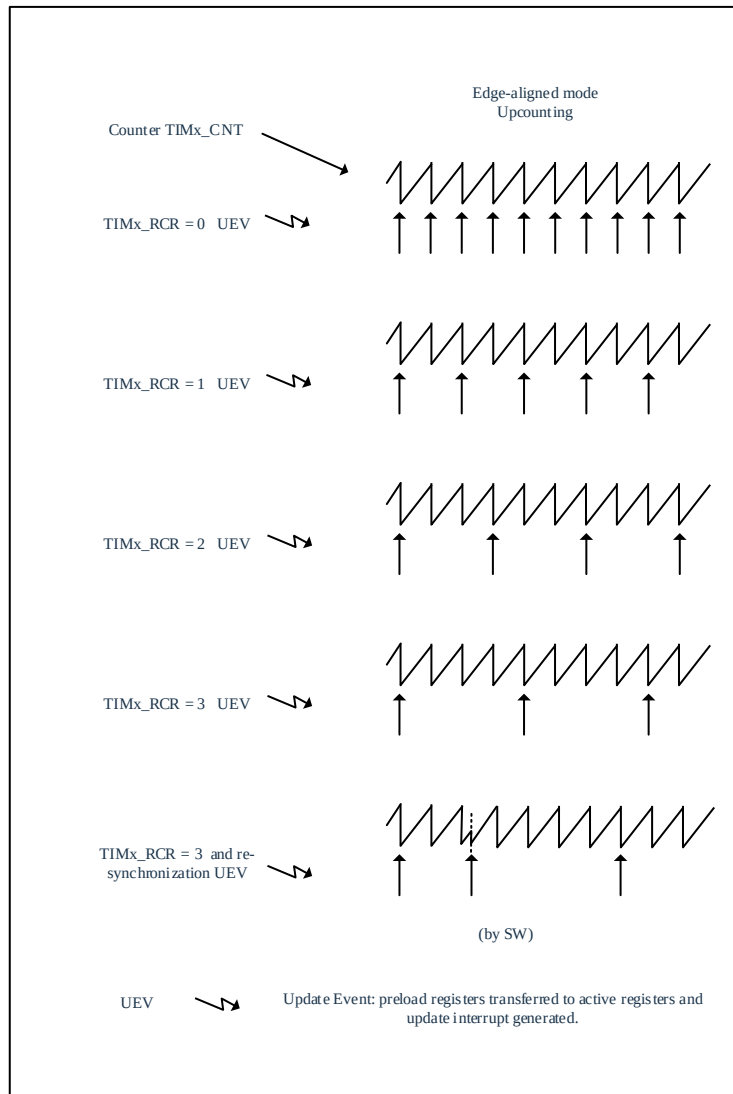
9.2.3 Repetition Counter

Section time-base unit describes how the update event (UEV) is generated with respect to the counter overruns/underruns. It is actually generated only when the repetition counter has reached zero. This can be useful when generating PWM signals.

This means that data are transferred from the preload registers to the shadow registers (TIMx_ARR auto-reload register, TIMx_PSC prescaler register, but also TIMx_CCRx capture/compare registers in compare mode) every N counter overruns or underruns, where N is the value in the TIMx_RCR repetition counter register. The repetition counter is decremented at each counter overrun in upcounting mode.

The repetition counter is an auto-reload type; the repetition rate is maintained as defined by the TIMx_RCR register value. When the update event is generated by software (by setting the UG bit in TIMx_EGR register) or by hardware through the slave mode controller, it occurs immediately whatever the value of the repetition counter is and the repetition counter is reloaded with the content of the TIMx_RCR register.

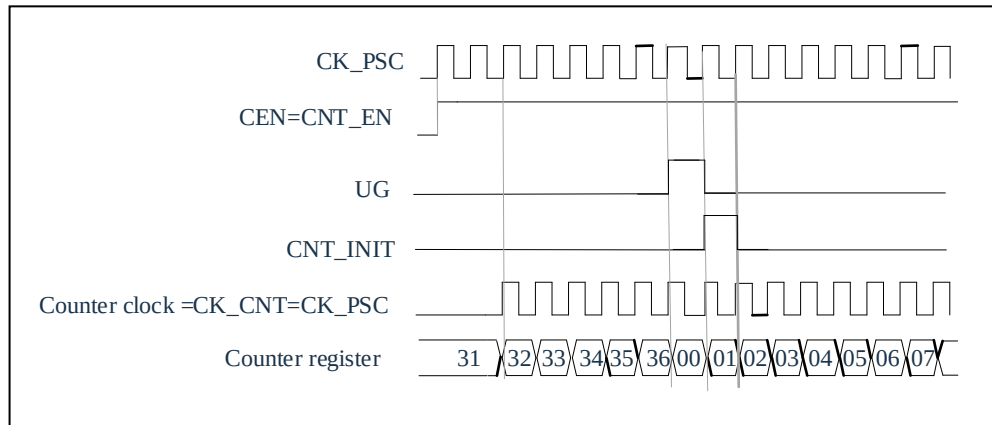
Figure 9-10 Update rate examples depending on mode and TIMx_RCR register settings



9.2.4 Clock Sources

The counter clock is provided by the internal clock (CK_INT). The CEN bit in the TIMx_CR1 register and the UG bit in the TIMx_EGR register are the actual control bits (except that the UG bit is automatically cleared) and can only be changed by software. Once the CEN bit is set to 1, the internal clock provides the clock to the prescaler.

Figure 9-11 Control circuit in normal mode, internal clock divided by 1



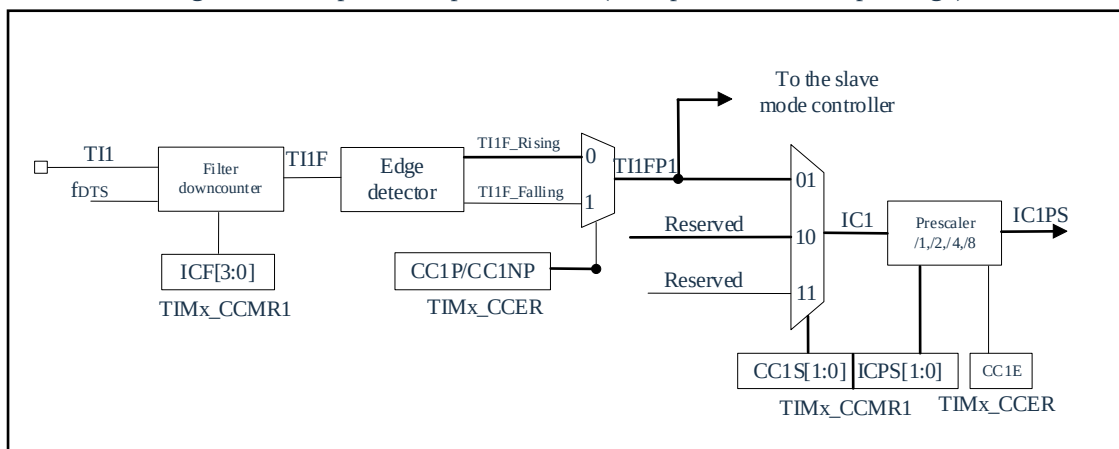
9.2.5 Capture/Compare Channels

Each Capture/Compare channel is built around a capture/compare register (including a shadow register), a input stage for capture (with digital filter, multiplexing and prescaler) and an output stage (with comparator and output control).

The figures below give an overview of one Capture/Compare channel.

The input stage samples the corresponding TI1 input to generate a filtered signal TI1F. Then, an edge detector with polarity selection generates a signal (TI1FP1) which can be used as trigger input by the slave mode controller or as the capture command. It is prescaled before the capture register (IC1PS).

Figure 9-12 Capture/compare channel (example: channel 1 input stage)



[illegible][illegible]

In capture mode, captures are actually done in the shadow register, which is copied into the preload register. In compare mode, the content of the preload register is copied into the shadow register which is compared to the counter.

9.2.6 Input Capture Mode

In Input capture mode, the Capture/Compare Registers (TIMx_CCR1) are used to latch the value of the counter after a transition detected by the corresponding ICx signal. When a capture occurs, the corresponding CCxIF flag (TIMx_SR register) is set. If a capture occurs while the CCxIF flag was already high, then the over-capture flag CCxOF (TIMx_SR register) is set. CCxIF can be cleared by software by writing it to '0' or by reading the captured data stored in the TIMx_CCRx register. CCxOF is cleared when it is written with 0.

The following example shows how to capture the counter value in TIMx_CCR1 when TI1 input rises. To do this, use the following procedure:

- Select the active input: TIMx_CCR1 must be linked to the TI1 input, so write the CC1S bits to 01 in the TIMx_CCMR1 register. As soon as CC1S becomes different from 00, the channel is configured in input and the TIMx_CCR1 register becomes read-only.
- Program the appropriate input filter duration in relation with the signal connected to the timer (when the input is one of the TIx (ICxF bits in the TIMx_CCMRx register). Let's imagine that, when toggling, the input signal is not stable during at most 5 internal clock cycles. We must program a filter duration longer than these 5 clock cycles. We can validate a transition on TI1 when 8 consecutive samples with the new level have been detected (sampled at f_{DTS} frequency). Then write IC1F bits to 0011 in the TIMx_CCMR1 register.
- Select the edge of the active transition on the TI1 channel by writing CC1P bit to 0 in the TIMx_CCER register (rising edge in this case).
- Program the input prescaler. In our example, we wish the capture to be performed at each valid transition, so the prescaler is disabled (write IC1PS bits to '00' in the TIMx_CCMR1 register).
- Enable capture from the counter into the capture register by setting the CC1E bit in the TIMx_CCER register.

If needed, enable the related interrupt request by setting the CC1IE bit in the TIMx_DIER register, and/or the DMA request by setting the CC1DE bit in the TIMx_DIER register.

When an input capture occurs:

- The TIMx_CCR1 register gets the value of the counter on the active transition.
- CC1IF flag is set (interrupt flag). CC1OF is also set if at least two consecutive captures occurred whereas the flag was not cleared.
- An interrupt is generated depending on the CC1IE bit.
- A DMA request is generated depending on the CC1DE bit.

In order to handle the overcapture, it is recommended to read the data before the overcapture flag. This is to avoid missing an overcapture which could happen after reading the flag and before reading the data.

Note: IC interrupt and/or DMA requests can be generated by software by setting the corresponding CCxG bit in the TIMx_EGR register.

9.2.7 Forced Output Mode

In output mode (CCxS bits = 00 in the TIMx_CCMR1 register), each output compare signal (OC1REF and then OC1) can be forced to active or inactive level directly by software, independently of any comparison between the output compare register and the counter.

To force an output compare signal (OC1REF/OC1) to its active level, one just needs to write 101 in the OC1M bits in the corresponding TIMx_CCMR1 register. Thus OC1REF is forced high (OC1REF is always active high) and OC1 get opposite value to CC1P polarity bit.

For example: CC1P=0 (OC1 active high) => OC1 is forced to high level.

The OC1REF signal can be forced low by writing the OC1M bits to 100 in the TIMx_CCMR1 register.

Anyway, the comparison between the TIMx_CCR1 shadow register and the counter is still performed and allows the flag to be set. Interrupt and DMA requests can be sent accordingly. This is described in the output compare mode section below.

9.2.8 Output Compare Mode

This function is used to control an output waveform or indicating when a period of time has elapsed.

When a match is found between the capture/compare register and the counter, the output compare function:

- Assigns the corresponding output pin to a programmable value defined by the output compare mode (OC1M bits in the TIMx_CCMR1 register) and the output polarity (CC1P bit in the TIMx_CCER register). The output pin can keep its level (OC1M=000), be set active (OC1M=001), be set inactive (OC1M=010) or can toggle (OC1M=011) on match.
- Sets a flag in the interrupt status register (CC1IF bit in the TIMx_SR register).
- Generates an interrupt if the corresponding interrupt mask is set (CC1IE bit in the TIMx_DIER register). The OC1PE bit in TIMx_CCMR1 selects whether the TIMx_CCR1 register needs to use the preload register.

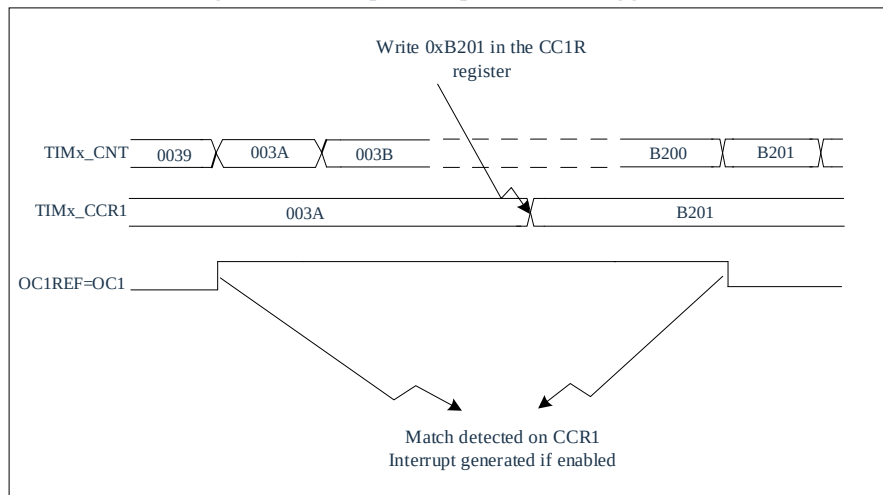
In output compare mode, the update event UEV has no effect on OC1REF and OC1 output. The timing resolution is one count of the counter. Output compare mode can also be used to output a single pulse (in One-pulse mode).

Procedure:

1. Select the counter clock (internal, external, prescaler).
2. Write the desired data in the TIMx_ARR and TIMx_CCR1 registers.
3. Set the CC1IE bit if an interrupt request is to be generated.
4. Select the output mode. For example:
 - Write OC1M = 011 to toggle OC1 output pin when CNT matches CCR1
 - Write OC1PE = 0 to disable preload register
 - Write CC1P = 0 to select active high polarity
 - Write CC1E = 1 to enable the output
5. Enable the counter by setting the CEN bit in the TIMx_CR1 register.

The TIMx_CCR1 register can be updated at any time by software to control the output waveform, provided that the preload register is not enabled (OCxPE='0', else TIMx_CCR1 shadow register is updated only at the next update event UEV). An example is given in Figure 9-15.

Figure 9-15 Output compare mode, toggle on OC1



9.2.9 PWM Mode

Pulse Width Modulation mode allows to generate a signal with a frequency determined by the value of the TIMx_ARR register and a duty cycle determined by the value of the TIMx_CCR1 register.

The PWM mode can be selected independently on each channel (one PWM per OCx output) by writing '110' (PWM mode 1) or '111' (PWM mode 2) in the OC1M bits in the TIMx_CCMR1 register. The corresponding preload register must be enabled by setting the OC1PE bit in the TIMx_CCMR1 register, and eventually the auto-reload preload register by setting the ARPE bit in the TIMx_CR1 register.

As the preload registers are transferred to the shadow registers only when an update event occurs, before starting the counter, all registers must be initialized by setting the UG bit in the TIMx_EGR register.

The polarity of OC1 can be configured via software by setting the CC1P bit in the TIMx_CCER register. This bit allows you to choose whether the output is active high or active low. The CC1E bit in the TIMx_CCER register controls the output enable for OC1.

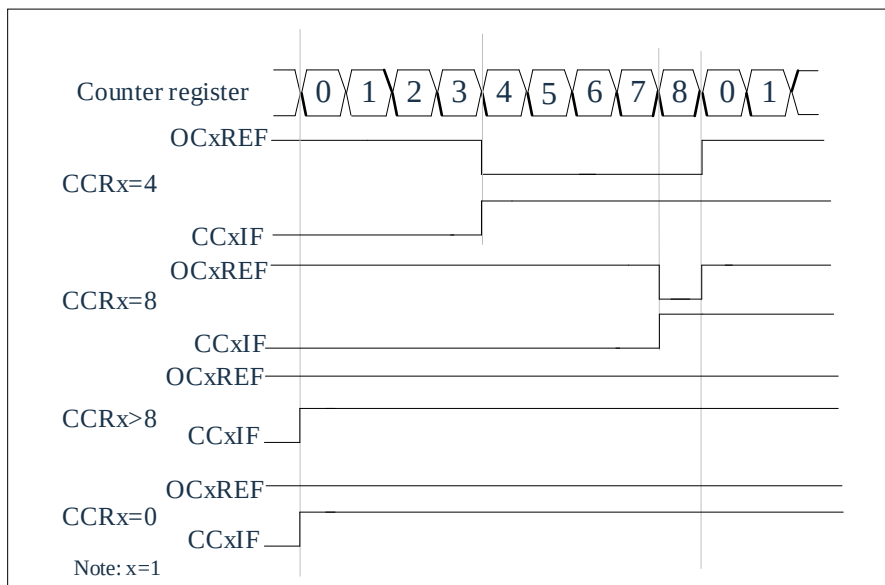
In PWM mode (Mode 1 or Mode 2), TIMx_CNT and TIMx_CCR1 are always compared to determine whether $TIMx\_CNT \leq TIMx\_CCR1$.

The timer can only generate edge-aligned PWM when the counter is counting upwards.

9.2.9.1 PWM Edge-Aligned Mode

Below is an example of PWM Mode 1. When $TIMx\_CNT < TIMx\_CCR1$, the PWM reference signal OCxREF is high; otherwise, it is low. If the comparison value in TIMx_CCR1 is greater than the auto-reload value TIMx_ARR, OC1REF stays at '1'. If the comparison value is 0, OC1REF stays at '0'. The diagram below shows the edge-aligned PWM waveform when $TIMx\_ARR = 8$.

Figure 9-16 Edge-aligned PWM waveforms (ARR=8)



9.2.10 Complementary Outputs and Dead-Time Insertion

TIM16/17 can output two complementary signals and manage the instantaneous switching off and on of the outputs. This time is typically referred to as the dead-time, and users should adjust the dead-time based on the characteristics of the connected output devices (such as level-shifting delay, power switch delay, etc.).

By configuring the CC1P and CC1NP bits in the TIMx_CCER register, the polarity can be independently selected for each output (main output OC1 or complementary output OC1N).

The complementary signals OC1 and OC1N are controlled by a combination of control bits, including the CC1E and CC1NE bits in the TIMx_CCER register, and the MOE, OIS1, OIS1N, OSS1, and OSSR bits in the TIMx_BDTR and TIMx_CR2 registers. These are detailed in Table 9-1, which shows the control bits for the complementary output channels OC1 and OC1N with brake functionality. Notably, the dead-time is activated when transitioning to the IDLE state (when MOE drops to 0).

Simultaneously setting the CC1E and CC1NE bits will insert the dead-time, and if a brake circuit is present, the MOE bit must also be set. Each channel has an 8-bit dead-time generator DTG[7:0]. The reference signal OC1REF can generate two outputs: OC1 and OC1N. If OC1 and OC1N are active high:

- The OC1 output signal is the same as the reference signal, except that its rising edge is delayed relative to the rising edge of the reference signal.
- The OC1N output signal is the opposite of the reference signal, except that its rising edge is delayed relative to the falling edge of the reference signal.

If the delay is greater than the current active output width (OC1 or OC1N), no corresponding pulse will be generated.

The following diagrams show the relationship between the output signal from the dead-time generator and the current reference signal OC1REF. (Assuming CC1P = 0, CC1NP = 0, MOE = 1, CC1E = 1, and CC1NE = 1)

Figure 9-17 Complementary outputs with dead-time insertion

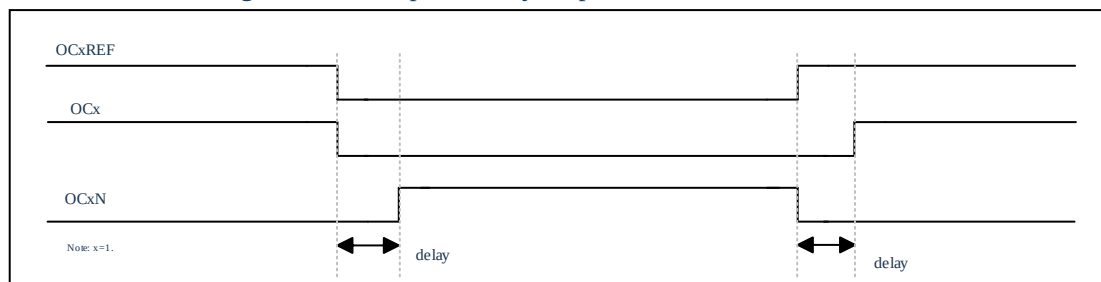


Figure 9-18 Dead-time waveforms with delay greater than the negative pulse

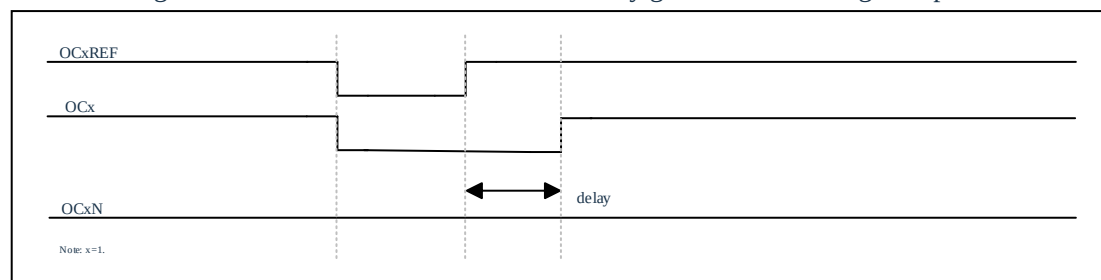
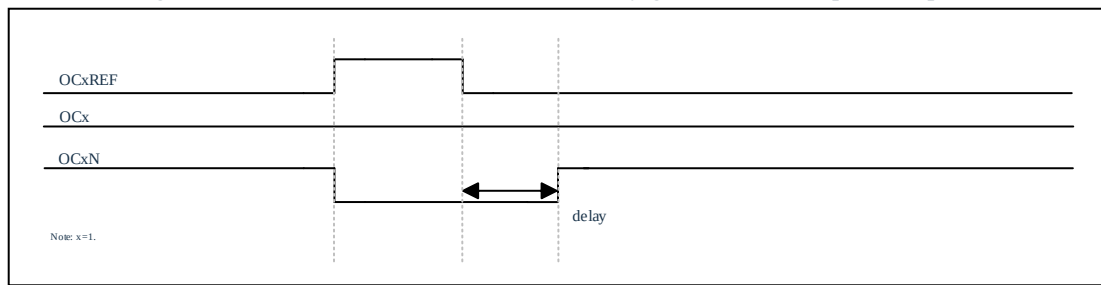


Figure 9-19 Dead-time waveforms with delay greater than the positive pulse



The dead-time delay is the same for each of the channels and is programmable with the DTG bits in the TIMx_BDTR register.

Re-directing OC1REF to OC1 or OC1N

In output mode (forced, output compare or PWM), OC1REF can be re-directed to the OC1 output or to OC1N output by configuring the CC1E and CC1NE bits in the TIMx_CCER register. This allows to send a specific waveform (such as PWM or static active level) on one output while the complementary remains at its inactive level. Other alternative possibilities are to have both outputs at inactive level or both outputs active and complementary with dead-time.

Note: When only OC1N is enabled (CC1E=0, CC1NE=1), it is not complemented and becomes active as soon as OC1REF is high. For example, if CC1NP=0 then OC1N=OC1REF. On the other hand, when both OC1 and OC1N are enabled (CC1E=CC1NE=1) OCx becomes active when OC1REF is high whereas OC1N is complemented and becomes active when OC1REF is low.

9.2.11 Using the Break Function

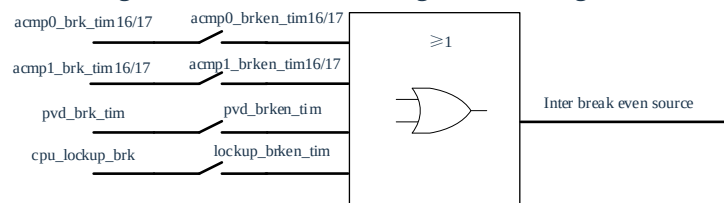
When using the break function, the output enable signals and inactive levels are modified according to additional control bits (MOE, OSS1 and OSSR bits in the TIMx_BDTR register, OIS1 and OIS1N bits in the TIMx_CR2 register). In any case, the OC1 and OC1N outputs cannot be set both to active level at a given time.

The source for break (BRK) channel can be an external source connected to the BKIN pin or one of the following internal sources:

- the core LOCKUP output
- the PVD output
- the CMP output

The block diagram of the internal brake signal is shown below.

Figure 9-20 Internal brake signal block diagram.



When exiting from reset, the break circuit is disabled and the MOE bit is low. The break function can be enabled by setting the BKE bit in the TIMx_BDTR register. The break input polarity can be selected by configuring the BKP bit in the same register. BKE and BKP can be modified at the same time. When the BKE and BKP bits are written, a delay of 1 APB clock cycle is applied before the writing is effective. Consequently, it is necessary to wait 1 APB clock period to correctly read back the bit after the write operation.

Because MOE falling edge can be asynchronous, a resynchronization circuit has been inserted between the actual signal (acting on the outputs) and the synchronous control bit (accessed in the TIMx_BDTR register). It results in some delays between the asynchronous and the synchronous signals. In particular, if MOE is set to 1 whereas it was low, a delay must be inserted (dummy instruction) before reading it correctly. This is because the write acts on the asynchronous signal whereas the read reflects the synchronous signal.

When a break occurs (selected level on the break input):

- The MOE bit is cleared asynchronously, putting the outputs in inactive state, idle state or in reset state (selected by the OSS1 bit). This feature functions even if the MCU oscillator is off.
- Each output channel is driven with the level programmed in the OIS1 bit in the TIMx_CR2 register as soon as MOE=0. If OSS1=0 then the timer releases the enable output else the enable output remains high.
- When complementary outputs are used:
 - The outputs are first put in reset state inactive state (depending on the polarity). This is done asynchronously so that it works even if no clock is provided to the timer.
 - If the timer clock is still present, then the dead-time generator is reactivated in order to drive the outputs with the level programmed in the OIS1 and OIS1N bits after a dead-time. Even in this case, OC1 and OC1N cannot be driven to their active level together. Note that because of the resynchronization on MOE, the dead-time duration is a bit longer than usual (around 2 ck_tim clock cycles).

- If OSSI=0 then the timer releases the enable outputs else the enable outputs remain or become high as soon as one of the CC1E or CC1NE bits is high.

- If the BIE bit in the TIMx_DIER register is set, an interrupt will be generated when the brake status flag (BIF bit in the TIMx_SR register) is '1'. If the BDE bit in the TIMx_DIER register is set, a DMA request will be generated.
- If the AOE bit in the TIMx_BDTR register is set, the MOE bit is automatically set again at the next update event UEV. This can be used to perform a regulation, for instance. Else, MOE remains low until it is written with 1 again. In this case, it can be used for security and the break input can be connected to an alarm from power drivers, thermal sensors or any security components.

Note: The break inputs is acting on level. Thus, the MOE cannot be set while the break input is active (neither automatically nor by software). In the meantime, the status flag BIF cannot be cleared.

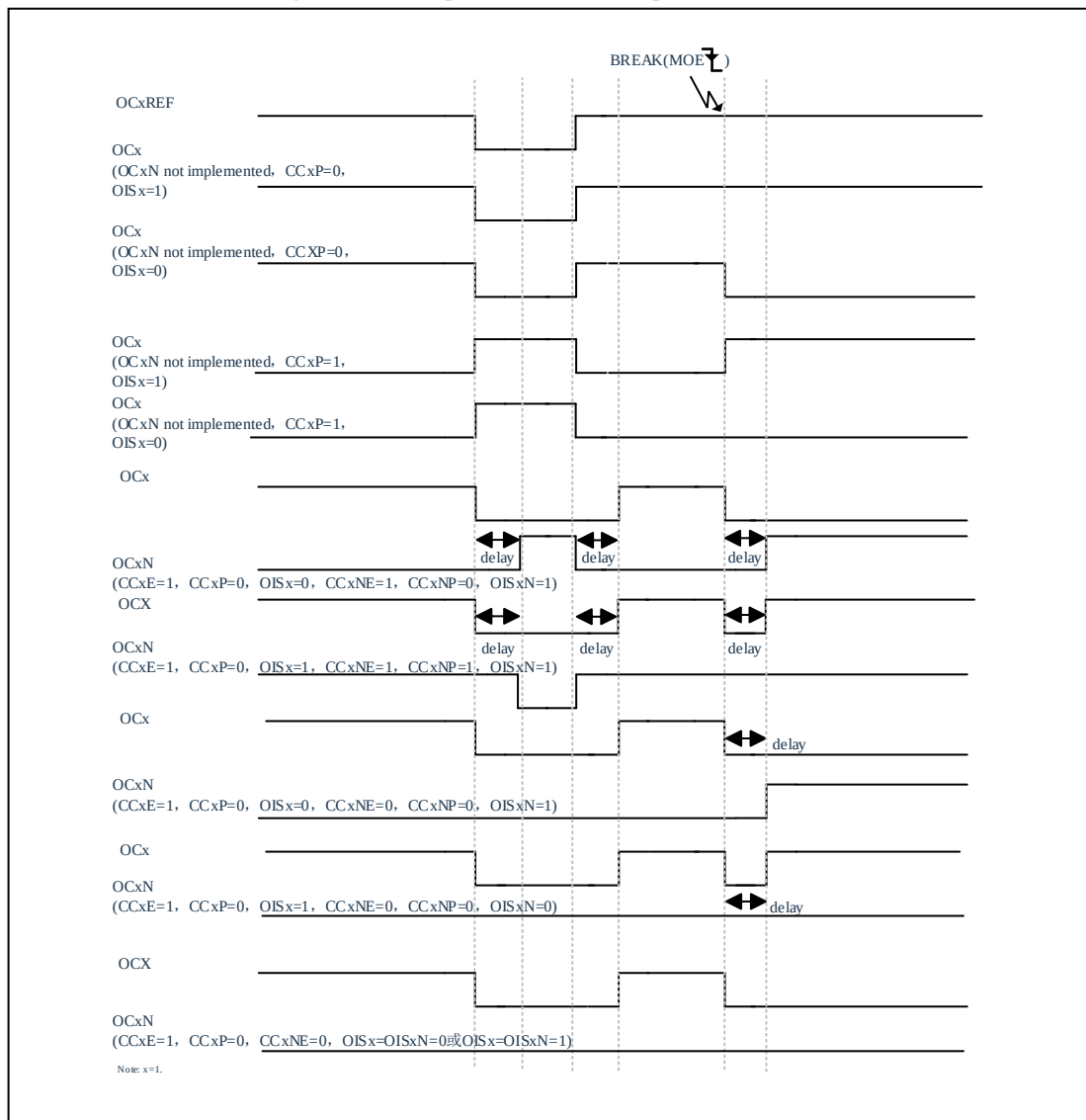
The break can be generated by the BRK input which has a programmable polarity and an enable bit BKE in the TIMx_BDTR Register. There are two ways to trigger braking:

- Through the programmable polarity BKR input, while enabling BKE in the TIMx_BDTR register.
- By setting the BG bit in the TIMx_EGR register via software.

In addition to the break input and the output management, a write protection has been implemented inside the break circuit to safeguard the application. It allows to freeze the configuration of several parameters (dead-time duration, OC1/OC1N polarities and state when disabled, OC1M configurations, break enable and polarity). The protection can be selected among 3 levels with the LOCK bits in the TIMx_BDTR register. The user can select one of the three protection levels through the LOCK bit in the TIMx_BDTR register, as described in the TIMx Brake and Dead-Time Registers (TIMx_BDTR). After an MCU reset, the LOCK bit can only be modified once.

The Figure below shows an example of behavior of the outputs in response to a break.

Figure 9-21 Output behavior in response to a break



9.2.12 One-Pulse Mode

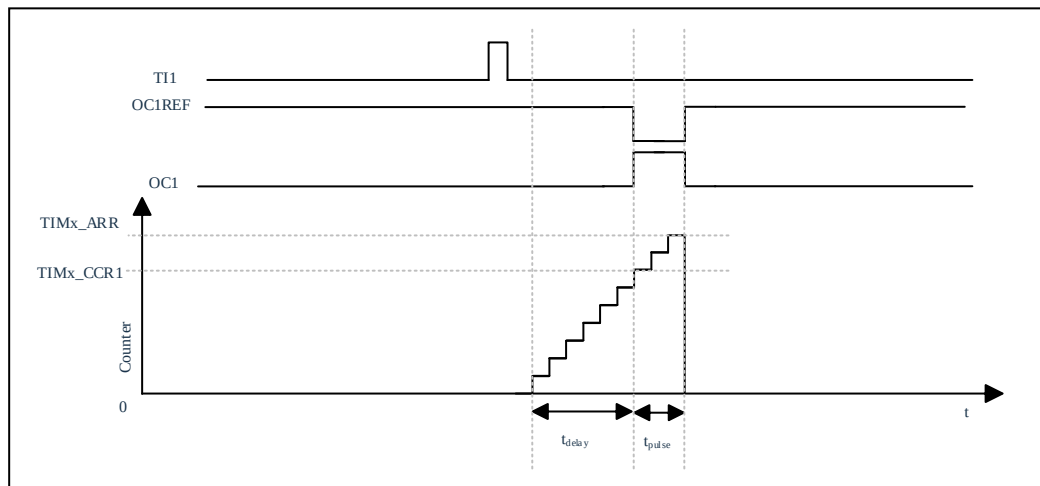
One-pulse mode (OPM) is a particular case of the previous modes. It allows the counter to be started in response to a stimulus and to generate a pulse with a programmable length after a programmable delay.

Starting the counter can be controlled through the slave mode controller. Generating the waveform can be done in output compare mode or PWM mode. One-pulse mode is selected by setting the OPM bit in the TIMx_CR1 register. This makes the counter stop automatically at the next update event UEV.

A pulse can be correctly generated only if the compare value is different from the counter initial value. Before starting (when the timer is waiting for the trigger), the configuration must be:

- In upcounting: $CNT < CCR1 \leq ARR$ (in particular, $0 < CCR1$)
- In downcounting: $CNT > CCR1$

Figure 9-22 Example of one-pulse mode



Particular case: OC1 fast enable

In One-pulse mode, the edge detection on TI11 input set the CEN bit which enables the counter. Then the comparison between the counter and the compare value makes the output toggle. But several clock cycles are needed for these operations and it limits the minimum delay $t_{DELAY\ min}$ we can get. If one wants to output a waveform with the minimum delay, the OC1FE bit can be set in the TIMx_CCMR1 register. Then OC1Ref (and OC1) are forced in response to the stimulus, without taking in account the comparison. Its new level is the same as if a compare match had occurred. OC1FE acts only if the channel is configured in PWM1 or PWM2 mode.

9.3 Register Mapping

(TIM16 base address = 0x4007_4400)

RO: Read-Only, WO: Write-Only, R/W: Read/Write

(TIM17 base address = 0x4007_4800)

RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Address	R/W	Description	Reset value
CR1	0x00	R/W	TIMx Control Register 1	0x00
CR2	0x04	R/W	TIMx Control Register 2	0x00
DIER	0x0C	R/W	TIMx DMA/Interrupt Enable Register	0x00
SR	0x10	R/W	TIMx Status Register	0x00
EGR	0x14	R/W	TIMx Event Generation Register	0x00
CCMR1	0x18	R/W	TIMx Capture/Compare Mode Register 1	0x00
CCER	0x20	R/W	TIMx Capture/Compare Enable Register	0x00
CNT	0x24	R/W	TIMx Counter	0x00
PSC	0x28	R/W	TIMx Prescaler	0x00
ARR	0x2C	R/W	TIMx Auto-Reload Register	0xFFFF
RCR	0x30	R/W	TIMx Repetition Counter Register	0x00
CCR1	0x34	R/W	TIMx Capture/Compare Register 1	0x00
DCR	0x48	R/W	TIMx DMA Control Register	0x00
DMAR	0x4C	R/W	TIMx Full Transfer DMA Address	0x00

9.4 Register Description

9.4.1 TIM16/17 Control Register 1 (TIMx_CR1)

Bit	Symbol	Description	Reset value
31:10	--	Reserved, set to 0.	0x0
9:8	CKD[1:0]	Clock division: This bit field indicates the division ratio between the timer clock (CK_INT) frequency and the dead-time and sampling clock (tDTS) used by the dead-time generators and the digital filters (ETR, TI1) (TlX) 00: tDTS = tCK_INT 01: tDTS = 2 * tCK_INT 10: tDTS = 4 * tCK_INT 11: Reserved, do not program this value	0x0
7	ARPE	Auto-reload preload enable: 0: TIMx_ARR register is not buffered 1: TIMx_ARR register is buffered	0
6:4	--	Reserved, set to 0.	0x0
3	OPM	One-pulse mode: 0: Counter is not stopped at update event 1: Counter stops counting at the next update event (clearing the bit CEN)	0
2	URS	Update request source: This bit is set and cleared by software to select the UEV event sources. 0: Any of the following events generate an update interrupt or DMA request if enabled. These events can be: — Counter overrun/underrun — Setting the UG bit — Update generation through the slave mode controller 1: Only counter overrun/underrun generates an update interrupt or DMA request if enabled	0
1	UDIS	Update disable: This bit is set and cleared by software to enable/disable UEV event generation. 0: UEV enabled. The Update (UEV) event is generated by one of the following events: — Counter overrun/underrun — Setting the UG bit — Update generation through the slave mode controller Buffered registers are then loaded with their preload values. 1: UEV disabled. The Update event is not generated, shadow registers keep their value (ARR, PSC, CCR1). However the counter and the prescaler are reinitialized if the UG bit is set or if a hardware reset is received from the slave mode controller.	0
0	CEN counter enable	Counter enable: 0: Counter disabled 1: Counter enabled Note: After the CEN bit is set by software, external clock, gated mode, and encoder mode can operate. The trigger mode can automatically set the CEN bit through hardware.	0

9.4.2 TIM16/17 Control Register 2 (TIM16/17_CR2)

Bit	Symbol	Description	Reset value
31:10	--	Reserved, set to 0.	0x0
9	OIS1N	Output Idle state 1 (OC1N output): 0: OC1N=0 after a dead-time when MOE=0 1: OC1N=1 after a dead-time when MOE=0 Note: This bit can not be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIMx_BDTR register).	0
8	OIS1	Output Idle state 1 (OC1 output): 0: OC1=0 (after a dead-time if OC1N is implemented) when MOE=0 1: OC1=1 (after a dead-time if OC1N is implemented) when MOE=0 Note: This bit can not be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIMx_BDTR register).	0
7:4	--	Reserved, set to 0.	0x0
3	CCDS	Capture/compare DMA selection: 0: CC1 DMA request sent when CC1 event occurs 1: CC1 DMA requests sent when update event occurs	0
2:1	--	Reserved, set to 0.	0x0
0	CCPC	Capture/compare preloaded control: 0: CC1E, CC1NE and OC1M are not preloaded 1: CC1E, CC1NE and OC1M are preloaded, after having been written, they are updated only when COM bit is set. Note: This bit acts only on channels that have a complementary output.	0

9.4.3 TIM16/17 DMA/Interrupt Enable Register (TIM16/17_DIER)

Bit	Symbol	Description	Reset value
31:10	--	Reserved, set to 0.	0x0
9	CC1DE	Capture/Compare 1 DMA request enable: 0: CC1 DMA request disabled 1: CC1 DMA request enabled	0
8	UDE	Update DMA request enable: 0: Update DMA request disabled 1: Update DMA request enabled	0
7	BIE	Break interrupt enable: 0: Break interrupt disabled 1: Break interrupt enabled	0
6	--	Reserved, set to 0.	0
5	COMIE	COM interrupt enable: 0: COM interrupt disabled 1: COM interrupt enabled	0
4:2	--	Reserved, set to 0.	0x0
1	CC1IE	Capture/Compare 1 interrupt enable: 0: CC1 interrupt disabled 1: CC1 interrupt enabled	0
0	UIE	Update interrupt enable: 0: Update interrupt disabled 1: Update interrupt enabled	0

9.4.4 TIM16/17 Status Register (TIM16/17_SR)

Bit	Symbol	Description	Reset value
31:10	--	Reserved, set to 0.	0x0
9	CC1OF	Capture/Compare 1 overcapture flag (This flag is set by hardware only when the corresponding channel is configured in input capture mode. It is cleared by software by writing it to '0'.): 0: No overcapture has been detected 1: The counter value has been captured in TIMx_CCR1 register while CC1IF flag was already set	0
8	--	Reserved, set to 0.	0
7	BIF	Break interrupt flag: (This flag is set by hardware as soon as the break input goes active. It can be cleared by software if the break input is not active.) 0: No break event occurred 1: An active level has been detected on the break input	0
6	--	Reserved, set to 0.	0
5	COMIF	COM interrupt flag: This flag is set by hardware on a COM event (once the capture/compare control bits –CC1E, CC1NE, OC1M– have been updated). It is cleared by software by writing it to '0'. 0: No COM event occurred 1: COM interrupt pending	0
4:2	--	Reserved, set to 0.	0x0
1	CC1IF	Channel CC1 is configured as output: The flag is set to 1 by hardware when the counter matches the compare value, except in center-aligned mode (see the description of the CMS bit in the TIMx_CR1 register). However, it needs to be cleared by software. 0: No match. 1: The content of the counter TIMx_CNT matches the content of the TIMx_CCR1 register. If channel CC1 is configured as input This bit is set by hardware on a capture. It is cleared by software or by reading the TIMx_CCR1 register. 0: No input capture occurred 1: The counter value has been captured in TIMx_CCR1 register (An edge has been detected on IC1 which matches the selected polarity)	0
0	UIF	Update interrupt flag: This bit is set by hardware on an update event. It is cleared by software. 0: No update occurred. 1: Update interrupt pending. This bit is set by hardware when the registers are updated: – At overrun regarding the repetition counter value (update if repetition counter = 0) and if the UDIS=0 in the TIMx_CR1 register. – When CNT is reinitialized by software using the UG bit in TIMx_EGR register, if URS=0 and UDIS=0 in the TIMx_CR1 register.	0

9.4.5 TIM16/17 Event Generation Register (TIM16/17_EGR)

Bit	Symbol	Description	Reset value
31:8	--	Reserved, set to 0.	0x0
7	BG	Break generation event: This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0: No action. 1: A break event is generated. MOE bit is cleared and BIF flag is set. Related interrupt or DMA transfer can occur if enabled.	0
6	--	Reserved, set to 0.	0
5	COMG	Capture/Compare control update generation: This bit can be set by software, it is automatically cleared by hardware. 0: No action 1: When the CCPC bit is set, it is possible to update the CC1E, CC1NE and OC1M bits Note: This bit acts only on channels that have a complementary output.	0
4:2	--	Reserved, set to 0.	0x0
1	CC1G	CC1G: Capture/Compare 1 generation: This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0: No action. 1: A capture/compare event is generated on channel 1 Channel CC1 is configured as output: CC1IF flag is set, Corresponding interrupt or DMA request is sent if enabled. Channel CC1 is configured as input: The current value of the counter is captured in TIMx_CCR1 register. The CC1IF flag is set, the corresponding interrupt or DMA request is sent if enabled. The CC1OF flag is set if the CC1IF flag was already high.	0
0	UG	Update generation: This bit can be set by software, it is automatically cleared by hardware. 0: No action 1: Reinitialize the counter and generates an update of the registers. Note that the prescaler counter is cleared too (anyway the prescaler ratio is not affected).	0

9.4.6 TIM16/17 Capture/Compare Mode Register 1(TIM16/17_CCMR1)

Output compare mode

Bit	Symbol	Description	Reset value
31:7	--	Reserved, set to 0.	0x0
6:4	OC1M	These bits define the behavior of the output reference signal OC1REF from which OC1 and OC1N are derived. OC1REF is active high whereas OC1 and OC1N active level depends on CC1P and CC1NP bits. 000: Frozen - The comparison between the output compare register TIMx_CCR1 and the counter TIMx_CNT has no effect on the OC1REF. 001: Set channel 1 to active level on match. OC1REF signal is forced high when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1). 010: Set channel 1 to inactive level on match. OC1REF signal is forced low when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1). 011: Toggle - OC1REF toggles when TIMx_CNT=TIMx_CCR1. 100: Force inactive level - OC1REF is forced low. 101: Force active level - OC1REF is forced high. 110: PWM mode 1 - channel 1 is active(OC1REF=0) as long as TIMx_CNT<TIMx_CCR1 else inactive (OC1REF=1). 111: PWM mode 2 - channel 1 is active as long as TIMx_CNT>TIMx_CCR1 else inactive. Note 1: These bits can not be modified as long as LOCK level 3 has been programmed (LOCK bits in TIMx_BDTR register) and CC1S=00 (the channel is configured in output). Note 2: In PWM mode 1 or 2, the OC1REF level changes only when the result of the comparison changes or when the output compare mode switches from “frozen” mode to “PWM” mode.	0x0
3	OC1PE	Output Compare 1 preload enable: 0: Preload register on TIMx_CCR1 disabled. TIMx_CCR1 can be written at anytime, the new value is taken in account immediately. 1: Preload register on TIMx_CCR1 enabled. Read/Write operations access the preload register. TIMx_CCR1 preload value is loaded in the active register at each update event. Note 1: These bits can not be modified as long as LOCK level 3 has been programmed (LOCK bits in TIMx_BDTR register) and CC1S=00 (the channel is configured in output). Note 2: Only in single-pulse mode, the PWM mode can be used without confirming the preload register, otherwise, its operation is undefined.	0
2	OC1FE	Output compare 1 fast enable: This bit is used to accelerate the effect of an event on the trigger in input on the CC output. 0: CC1 behaves normally depending on counter and CCR1 values even when the trigger is ON. The minimum delay to activate CC1 output when an edge occurs on the trigger input is 5 clock cycles. 1: An active edge on the trigger input acts like a compare match on CC1 output. Then, OC is set to the compare level independently of the result of the comparison. Delay to sample the trigger input and to activate CC1 output is reduced to 3 clock cycles. OC1FE acts only if the channel is configured in PWM1 or PWM2 mode.	0
1:0	CC1S	Capture/Compare 1 selection: This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC1 channel is configured as output 01: CC1 channel is configured as input, IC1 is mapped on	0x0

		TI1 10: Reserved 11: Reserved Note: CC1S bits are writable only when the channel is OFF (CC1E = '0' in TIMx_CCER).	

Input capture mode

Bit	Symbol	Description	Reset value
31:7	--	Reserved, set to 0.	0x0
7:4	IC1F[3:0]	Input capture 1 filter: This field defines the sampling frequency for the TI1 input and the digital filter bandwidth applied to TI1. The digital filter consists of an event counter, and every N events are considered a valid edge. 0000: No filter, sampling at f_{DTS} frequency 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 1111: $f_{SAMPLING}=f_{DTS}/32$, N=8	0x0
3:2	IC1PSC	Input capture 1 prescaler: This field defines the prescaler for the CC1 input (IC1). The prescaler is immediately reset when CC1E = 0 (in the TIMx_CCER register). 00: No prescaler, capture is executed on every detected edge. 01: Capture is executed every 2 events. 10: Capture is executed every 4 events. 11: Capture is executed every 8 events.	0x0
1:0	CC1S	Capture/compare 1 selection: 00: CC1 channel configured as output 01: CC1 channel configured as input, IC1 mapped to TI1 10: Reserved 11: Reserved Note: Data can only be written to the CC1S bit when the channel is disabled (CC1E = 0 in the TIMx_CCER register).	0x0

Table 9-1 Output control bits for complementary OCx and OCxN channels with break feature

Control bits					Output states	
MOE	OSSI	OSSR	CCxE	CCxNE	OCx output state	OCxN output state
1	X	0	0	0	Output disabled (not driven by the timer) OCx=0, OCx_EN=0	Output disabled (not driven by the timer) OCxN=0, OCxN_EN=0
		0	0	1	Output disabled (not driven by the timer) OCx=0, OCx_EN=0	OCxPEF+Polarity OCxN=OCxPEF xor OCxNP OCxN_EN=1
		0	1	0	OCxPEF+Polarity OCx=OCxPEF xor OCxP OCx_EN=1	Output disabled (not driven by the timer) OCxN=0, OCxN_EN=0
		0	1	1		
		1	0	0		
		1	0	1		
		1	1	0		
0	X	1	1	1	OCxPEF+Polarity+dead time OCx_EN=1	Complementary to OCPEF (not OC-REF)
		0	0	0		
		0	0	1		
		0	1	0		
		0	1	1		
		1	0			
		1	0	0		
		1	1	0		
		1	1	1		

Note: x=1.

9.4.7 TIM16/17 Capture/Compare Enable Register (TIM16/17_CCER)

Bit	Symbol	Description	Reset value
31:4	--	Reserved, set to 0.	0x0
3	CC1NP	Capture/Compare 1 complementary output polarity: 0: OC1N active high 1: OC1N active low Note: This bit is not writable as soon as LOCK level 2 or 3 has been programmed (LOCK bits in TIMx_BDTR register) and CC1S=00 (the channel is configured in output).	0
2	CC1NE	Capture/Compare 1 complementary output enable: 0: Off - OC1N is not active. OC1N level is then function of MOE, OSS1, OSSR, OIS1, OIS1N and CC1E bits. 1: On - OC1N signal is output on the corresponding output pin depending on MOE, OSS1, OSSR, OIS1, OIS1N and CC1E bits.	0
1	CC1P	Capture/Compare 1 output polarity: CC1 channel configured as output: 0: OC1 active high 1: OC1 active low CC1 channel configured as input: The CC1NP/CC1P bits select the polarity of TI1FP1 for capture operation. 00: Non-inverted/Rising Edge Trigger The circuit is sensitive to the rising edge of TI1FP1 (capture or trigger operation is performed in reset mode, external clock mode, or trigger mode), and TI1FP1 is non-inverted (trigger operation is performed in gated mode or encoder mode). 01: Inverted/Falling Edge Trigger The circuit is sensitive to the falling edge of TI1FP1 (capture or trigger operation is performed in reset mode, external clock mode, or trigger mode), and TI1FP1 is inverted (trigger operation is performed in gated mode or encoder mode). 10: Reserved, do not use this configuration. 11: Non-inverted/Edge Trigger The circuit is sensitive to both rising and falling edges of TI1FP1 (capture or trigger operation is performed in reset mode, external clock mode, or trigger mode), and TI1FP1 is non-inverted (trigger operation is performed in gated mode). This configuration must not be used in encoder mode. Note: This bit is not writable as soon as LOCK level 2 or 3 has been programmed (LOCK bits in TIMx_BDTR register)	0
0	CC1E	Capture/Compare 1 output enable: CC1 channel configured as output: 0: Off - OC1 is not active. OC1 level is then function of MOE, OSS1, OSSR, OIS1, OIS1N and CC1NE bits. 1: On - OC1 signal is output on the corresponding output pin depending on MOE, OSS1, OSSR, OIS1, OIS1N and CC1NE bits. CC1 channel configured as input: This bit determines if a capture of the counter value can actually be done into the input capture/compare register 1 (TIMx_CCR1) or not. 0: Capture disabled 1: Capture enabled	0

9.4.8 TIM16/17 Counter (TIM16/17_CNT)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	CNT[15:0]	Counter value	0x0

9.4.9 TIM16/17 Prescaler (TIM16/17_PSC)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	PSC[15:0]	Prescaler value The counter clock frequency (CK_CNT) is equal to $f_{CK\_PSC} / (PSC[15:0] + 1)$. The PSC contains the value that is loaded into the active prescaler register each time an update event occurs (including when the counter is cleared via the UG bit in the TIMx_EGR register, or when cleared by the trigger controller in "reset mode").	0x0

9.4.10 TIM16/17 Auto-Reload Register (TIM16/17_ARR)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	ARR[15:0]	Auto-reload value ARR is the value to be loaded in the actual auto-reload register. The counter is blocked while the auto-reload value is null.	0xFFFF

9.4.11 TIM16/17 Repetition Counter Register (TIM16/17_RCR)

Bit	Symbol	Description	Reset value
31:8	--	Reserved, set to 0.	0x0
7:0	REP[7:0]	REP[7:0]: Repetition counter value These bits allow the user to set-up the update rate of the compare registers (i.e. periodic transfers from preload to active registers) when preload registers are enable, as well as the update interrupt generation rate, if this interrupt is enable. Each time the REP_CNT related downcounter reaches zero, an update event is generated and it restarts counting from REP value. As REP_CNT is reloaded with REP value only at the repetition update event U_RC, any write to the TIMx_RCR register is not taken in account until the next repetition update event. It means in PWM mode (REP+1) corresponds to: - the number of PWM periods in edge aligned mode. - the number of PWM half-cycles in center-aligned mode	0x0

9.4.12 TIM16/17 Capture/Compare Register 1(TIM16/17_CCR1)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	CCR1[15:0]	Capture/Compare 1 value If channel CC1 is configured as output: CCR1 is the preload value of the Capture/Compare Register 1. If the preload function is not enabled via the OC1PE bit in the TIMx_CCMR1 register, the written value will be directly transferred to the current register. Otherwise, the value will take effect only when an update event occurs (copied to the actual active Capture/Compare Register 1). The actual Capture/Compare Register contains the value to be compared with the counter TIMx_CNT, and a signal will be generated on the OC1 output. If channel CC1 is configured as input: CCR1 is the counter value transferred by the last input capture 1 event (IC1).	0x0

9.4.13 TIM16/17 Brake and Dead Time Register (TIM16/17_BDTR)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15	MOE	Main output enable: This bit is cleared asynchronously by hardware as soon as the break input is active. It is set by software or automatically depending on the AOE bit. It is acting only on the channels which are configured in output. 0: OC and OCN outputs are disabled or forced to idle state 1: OC and OCN outputs are enabled if their respective enable bits are set (CC1E, CC1NE in TIMx_CCER register)	0
14	AOE	Automatic output enable: 0: MOE can be set only by software 1: MOE can be set by software or automatically at the next update event (if the break input is not be active) Note: This bit can not be modified as long as LOCK level 1 has been programmed (LOCK bits in TIMx_BDTR register).	0
13	BKP	Break polarity: 0: Break input BRK is active low 1: Break input BRK is active high Note: This bit can not be modified as long as LOCK level 1 has been programmed (LOCK bits in TIMx_BDTR register).	0
12	BKE	Break enable: 0: Brake input disabled (BRK and BRK_ACTH) 1: Brake input enabled (BRK and BRK_ACTH) Note: After programming the LOCK level 1 (in the LOCK bit of the TIMx_BDTR register), this bit cannot be modified.	0
11	OSSR	Off-state selection for Run mode: This bit is used when MOE=1 on channels having a complementary output which are configured as outputs. OSSR is not implemented if no complementary output is implemented in the timer. See OC/OCN enable description for more details 0: When inactive, OC/OCN outputs are disabled (OC/OCN enable output signal=0) 1: When inactive, OC/OCN outputs are enabled with their inactive level as soon as CCxE=1 or CCxNE=1. Then, OC/OCN enable output signal=1 Note: This bit can not be modified as soon as the LOCK level 2 has been programmed (LOCK bits in TIMx_BDTR register).	0
10	OSSI	Off-state selection for Idle mode: This bit is used when MOE=0 on channels configured as outputs. See OC/OCN enable description for more details 0: When inactive, OC/OCN outputs are disabled (OC/OCN enable output signal=0) 1: When inactive, OC/OCN outputs are forced first with their idle level as soon as CCxE=1 or CCxNE=1. OC/OCN enable output signal=1) Note: This bit can not be modified as soon as the LOCK level 2 has been programmed (LOCK bits in TIMx_BDTR register).	0
9:8	LOCK	Lock configuration: These bits offer a write protection against software errors. 00: LOCK OFF - No bit is write protected 01: LOCK Level 1 = OIS1 and OIS1N bits in TIMx_CR2 register and DTG/BKE/BKP/AOE bits in TIMx_BDTR register can no longer be written.	0x0

		10: LOCK Level 2 = LOCK Level 1 + CC Polarity bits (CC1P/CC1NP bits in TIMx_CCER register, as long as the related channel is configured in output through the CC1S bits) as well as OSSR and OSSI bits can no longer be written. 11: LOCK Level 3 = LOCK Level 2 + CC Control bits (OC1M and OC1PE bits in TIMx_CCMR1 registers, as long as the related channel is configured in output through the CC1S bits) can no longer be written. Note: The LOCK bits can be written only once after the reset. Once the TIMx_BDTR register has been written, their content is frozen until the next reset.	
7:0	DTG	Dead-time generator setup This bit-field defines the duration of the dead-time inserted between the complementary outputs. DT correspond to this duration. DTG[7:5]=0xx => DT=DTG[7:0]x t_{dtg}, with $t_{dtg}=t_{DTS}$. DTG[7:5]=10x => DT=(64+DTG[5:0])xt_{dtg}, with $T_{dtg}=2xt_{DTS}$. DTG[7:5]=110 => DT=(32+DTG[4:0])xt_{dtg}, with $T_{dtg}=8xt_{DTS}$. DTG[7:5]=111 => DT=(32+DTG[4:0])xt_{dtg}, with $T_{dtg}=16xt_{DTS}$. Example if TDTS=125ns (8MHz), dead-time possible values are: 0 to 15875 ns by 125 ns steps, 16 μs to 31750 ns by 250 ns steps, 32 μs to 63 μs by 1 μs steps, 64 μs to 126 μs by 2 μs steps Note: This bit-field can not be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIMx_BDTR register).	0x0

9.4.14 TIM16/17 DMA Control Register (TIM16/17_DCR)

Bit	Symbol	Description	Reset value
31:13	-	Reserved	0x0
12:8	DBL	DMA burst length This 5-bit vector defines the length of DMA transfers (the timer recognizes a burst transfer when a read or a write access is done to the TIMx_DMAR address). 00000: 1 transfer 00001: 2 transfers 00010: 3 transfers 00011: 4 transfers 10001: 18 transfers	0x0
7:5	-	Reserved	0x0
4:0	DBA	DMA base address This 5-bits vector defines the base-address for DMA transfers (when read/write access are done through the TIMx_DMAR address). DBA is defined as an offset starting from the address of the TIMx_CR1 register. Example: 00000: TIMx_CR1, 00001: TIMx_CR2, 00010: Reserved ...	0x0

9.4.15 DMA address for Full Transfer (TIM16/17_DMAR)

Bit	Symbol	Description	Reset value
31:16	--	Reserved, set to 0.	0x0
15:0	DMAB	DMA register for burst accesses A read or write access to the DMAR register accesses the register located at the address: “(TIMx_CR1 address) + (DBA+DMA index) x 4” in which: TIMx_CR1 address is the address of the control register 1, DBA is the DMA base address configured in TIMx_DCR register, DMA index is the offset automatically controlled by the DMA transfer, with a range between 0 and DBL (as configured in the DBL field of the TIMx_DCR register).	0x0

Chapter 10 Windowed Watchdog Timer (WWDG)

10.1 WWDG Introduction

The window watchdog is typically used to monitor software failures caused by external interference or unforeseen logical conditions that cause the application to deviate from its normal execution sequence. Unless the decrementing counter value is refreshed before the T6 bit becomes 0, the watchdog circuit will trigger an MCU reset when the preset time period is reached.

If the 7-bit decrementing counter value in the control register is refreshed before the decrementing counter reaches the window register value, it will also trigger an MCU reset. This means the counter must be refreshed within a specified time window.

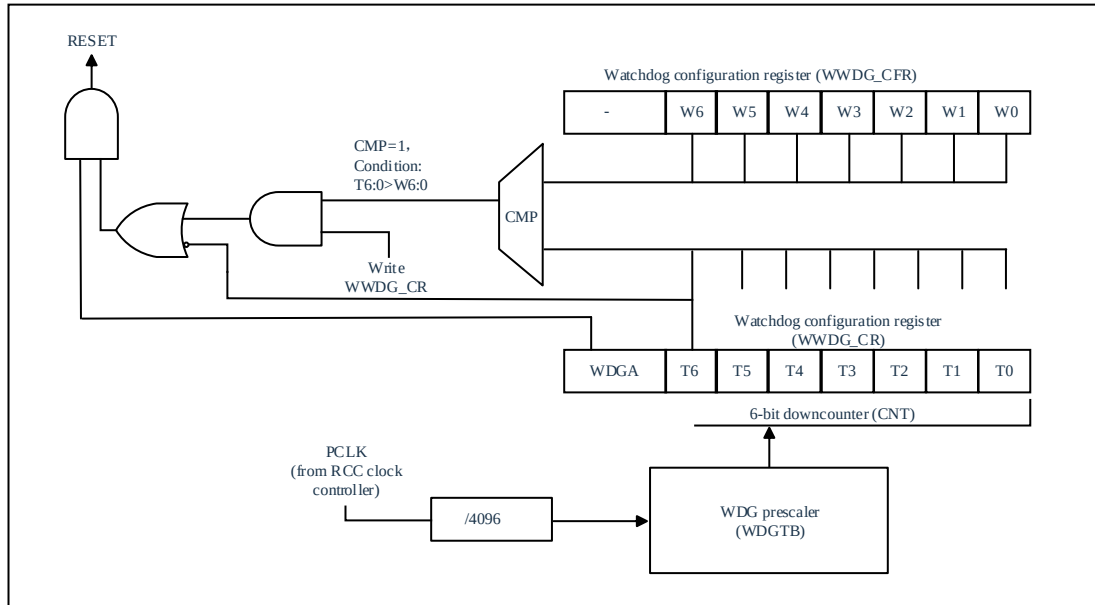
10.2 WWDG Main Features

- Programmable free-running down-counter
- Conditional reset
 - Reset (if watchdog activated) when the down-counter value becomes lower than 0x40
 - Reset (if watchdog activated) if the down-counter is reloaded outside the window
- Early wake-up interrupt (EWI): triggered (if enabled and the watchdog activated) when the down-counter is equal to 0x40

10.3 WWDG Functional Description

If the watchdog is activated (i.e., the WDGA bit in the WWDG_CR register is set to 1), a reset will be triggered when the 7-bit decrementing counter (T[6:0] bits) rolls from 0x40 to 0x3F (when T6 is cleared). If the counter value is greater than the value stored in the window register, a reset will occur if the software reloads the counter.

Figure 10-1 Watchdog block diagram



The application must regularly write to the WWDG_CR register during normal operation to prevent the MCU from resetting. This can only be done when the counter value is lower than the value stored in the window register. The value stored in the WWDG_CR register must be between 0xFF and 0xC0.

10.3.1.1 Enabling the Watchdog

The watchdog is always disabled after a reset. It is enabled by setting the WDGA bit in the WWDG_CR register, then it cannot be disabled again except by a reset.

10.3.1.2 Controlling the Down-Counter

This down-counter is free-running, counting down even if the watchdog is disabled. When the watchdog is enabled, the T6 bit must be set to prevent generating an immediate reset.

The T[5:0] bits contain the number of increments that represent the time delay before the watchdog produces a reset. The timing varies between a minimum and a maximum value, due to the unknown status of the prescaler when writing to the WWDG_CR register. The WWDG configuration register (WWDG_CFR) contains the high limit of the window: to prevent a reset, the down-counter must be reloaded when its value is lower than or equal to the window register value, and greater than 0x3F. It describes the window watchdog process.

Note: The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

10.3.1.3 Advanced Features of Watchdog Interrupt

If certain safety operations or data logging must be performed before triggering an actual reset, the Early Wake-up Interrupt (EWI) can be used. By setting the EWI bit in the WWDG_CFR register, the EWI interrupt is enabled. When the decrementing counter reaches a value of 0x40, the EWI interrupt will be generated. Before the device is reset, the corresponding Interrupt Service Routine (ISR) can be used to trigger specific actions (e.g., communication or data logging).

In some applications, the EWI interrupt can be used to manage software system checks and/or system recovery/function degradation without triggering a WWDG reset. In such cases, the corresponding ISR can be used to reload the WWDG counter to prevent a WWDG reset, and then trigger the desired operation.

The EWI interrupt can be cleared by writing 0 to the EWIF bit in the WWDG_SR register.

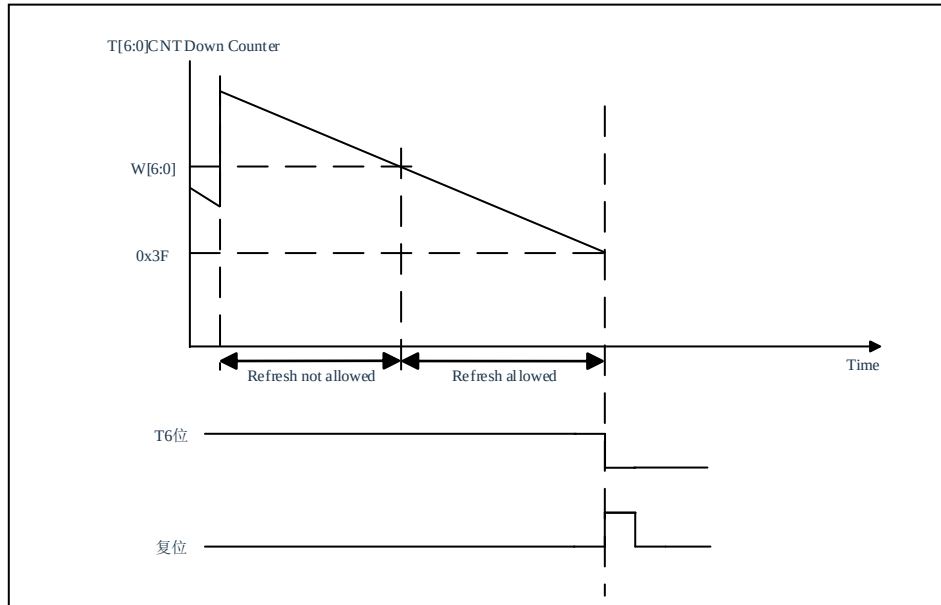
Caution: If the EWI interrupt cannot be used due to a system lock in a higher-priority task, the WWDG reset will eventually occur.

10.4 How To Program the Watchdog Timeout

Use the formula in Figure 10-2 to calculate the WWDG timeout.

Caution: When writing to the WWDG_CR register, always write 1 in the T6 bit to avoid generating an immediate reset.

Figure 10-2 Windowed watchdog timing diagram



The formula to calculate the timeout value is given by:

$$t_{\text{WWDG}} = t_{\text{PCLK}} * 4096 * 2^{\text{WDGTB}} * (t[5:0] + 1) \text{ (ms)}$$

where:

t_{WWDG} : WWDG timeout

t_{PCLK} : APB clock period measured in ms

Refer to the below table for the minimum and maximum values of T_{WWDG} .

Table 10-1 24MHz (F_{PCLK}) timeout values

Prescaler	WDGTB	Minimum timeout (us) $T[5:0] = 0x00$	Maximum timeout (ms) $T[5:0] = 0x3F$
1	0	170.67	10.92
2	1	341.34	21.85
4	2	682.67	43.69
8	3	1365.34	87.39

10.5 Register Mapping

(WWDG base address = 0x4006_2C00)

RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
WWDG_CR	0x000	R/W	WWDG Control Register	0x7f
WWDG_CFR	0x004	R/W	WWDG Configuration Register	0x7f
WWDG_SR	0x008	R/W	WWDG Status Register	0x0

10.6 Register Description

10.6.1 WWDG Timer Control Register (WWDG_CR)

Bit	Symbol	Description	Reset value
31:8	-	Reserved	0x0
7	WDGA	Activation bit 0: Watchdog disabled 1: Watchdog enabled This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.	0
6:0	T[6:0]	7-bit counter (MSB to LSB) These bits contain the value of the watchdog counter, decremented every $(4096 \times 2^{\text{WDGTB}[1:0]})$ PCLK cycles. A reset is produced when it is decremented from 0x40 to 0x3F (T6 becomes cleared).	0x7f

10.6.2 WWDG Configuration Register (WWDG_CFR)

Bit	Symbol	Description	Reset value
31:10	-	Reserved	0x0
9	EWI	Early wakeup interrupt Set by software and cleared by hardware after a reset. When set, an interrupt occurs whenever the counter reaches the value 0x40.	0
8:7	WDGTB	The time base of the prescaler can be modified as follows: 00: CK counter clock (PCLK div 4096) div 1 01: CK counter clock (PCLK div 4096) div 2 10: CK counter clock (PCLK div 4096) div 4 11: CK counter clock (PCLK div 4096) div 8	0
6:0	W	7-bit window value These bits contain the window value to be compared with the down-counter.	0x7f

10.6.3 WWDG Status Register (WWDG_SR)

Bit	Symbol	Description	Reset value
31:1	-	Reserved	0x0
0	EWIF	Early wakeup interrupt flag This bit is set by hardware when the counter has reached the value 0x40. It must be cleared by software by writing 0. Writing 1 has no effect. This bit is also set if the interrupt is not enabled.	0

Chapter 11 Independent Watchdog (IWDG)

11.1 Introduction

The Independent Watchdog (IWDG) is driven by its dedicated low-speed clock (LSI), so it remains operational even if the main clock fails.

When a system generates a LOCKUP, the IWDG counter enable is forced on, and after the LOCKUP is cleared, the IWDG counter enable is turned off.

The IWDG is best suited for applications that require the watchdog to run as a totally independent process outside the main application, but have lower timing accuracy constraints.

11.2 Features

- Free-running downcounter
- Clocked from an independent RC oscillator (can operate in Sleep and Deep Sleep modes)
- Reset (if watchdog activated) when the downcounter value becomes lower than 0x000

11.3 Functional Description

Figure 11-1 shows the functional blocks of the independent watchdog module.

When the independent watchdog is started by writing the value 0xCCCC in the IWDG key register (IWDG_KR), the counter starts counting down from the reset value of 0xFFFF. When it reaches the end of count value (0x000) a reset signal is generated (IWDG reset). Whenever the key value 0xAAAA is written in the IWDG key register (IWDG_KR), the IWDG_RLR value is reloaded in the counter and the watchdog reset is prevented.

11.3.1 Hardware Watchdog

If the “Hardware watchdog” feature is enabled through the device option bits, the watchdog is automatically enabled at power-on, and generates a reset unless the IWDG key register (IWDG_KR) is written by the software before the counter reaches end of count or if the downcounter is reloaded inside the window.

11.3.2 Register Access Protection

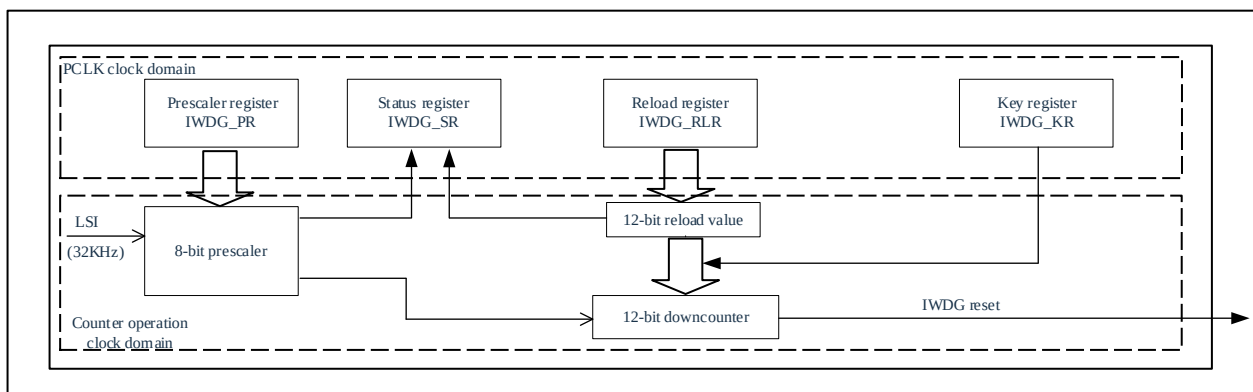
Write access to IWDG prescaler register (IWDG_PR) and IWDG reload register (IWDG_RLR) is protected. To modify them, the user must first write the code 0x5555 in the IWDG key register (IWDG_KR). A write access to this register with a different value breaks the sequence and register access is protected again. This is the case of the reload operation (writing 0xAAAA).

A status register is available to indicate that an update of the prescaler or of the downcounter reload value or of the window value is ongoing.

11.3.3 Debug Mode

When the device enters Debug mode (core halted), the IWDG counter either continues to work normally or stops, depending on the configuration bits set in section 1.4.

Figure 11-1 Independent watchdog block diagram



Caution: The watchdog function is located in the VDD voltage domain, still functional in Stop and Standby modes.

Table 11-1 32KHz (LSI) frequency IWDG timeout periods minimum/maximum values

Prescaler	PR[2:0] bits	Minimum timeout (ms) RL[11:0] = 0x000	Maximum timeout (ms) RL[11:0] = 0xFFFF
/4	0	0.125	512
/8	1	0.25	1024
/16	2	0.5	2048
/32	3	1	4096
/64	4	2	8192
/128	5	4	16384
/256	6	8	32768

Caution: These times are provided for a 32KHz clock. In practice, the MCU's internal RC frequency will vary between 20KHz and 40KHz. Furthermore, even if the RC oscillator's frequency is accurate, the exact timing still depends on the phase difference between the APB interface clock and the RC oscillator clock, meaning there will always be an uncertain full RC cycle.

11.4 IWDG Register Mapping

(IWDG base address = 0x4006_3000)

RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
IWDG_KR	0x000	R/W	IWDG Control Register	0x0
IWDG_PR	0x004	R/W	IWDG Prescaler Register	0x0
IWDG_RLR	0x008	R/W	IWDG Reload Register	0xff
IWDG_SR	0x00C	R/W	IWDG Status Register	0X0

11.5 IWDG Register Description

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

11.5.1 Key Register (IWDG_KR)

Bit	Symbol	Description	Reset value
31:16	-	Reserved	0x0
15:0	KEY	Key value (write only, read 0000h): These bits must be written by software at regular intervals with the key value AAAAh, otherwise the watchdog generates a reset when the counter reaches 0. 0x5555: Enable access to the IWDG_PR and IWDG_RLR registers 0xCCCC: Enable the watchdog (except if the hardware watchdog option is selected)	0

11.5.2 Prescaler Register (IWDG_PR)

Bit	Symbol	Description	Reset value
31:3	-	Reserved	0x0
2:0	PR	Prescaler divider: 000: Divided by 4 001: Divided by 8 010: Divided by 16 011: Divided by 32 100: Divided by 64 101: Divided by 128 110: Divided by 256 111: Divided by 256 These bits are write access protected see Section 12.3.2: Register access protection. They are written by software to select the prescaler divider feeding the counter clock. PVU bit of the IWDG status register (IWDG_SR) must be set to 0 in order to be able to change the prescaler divider.	0x0

Note: Reading this register returns the prescaler value from the VDD voltage domain. This value may not be up to date/valid if a write operation to this register is ongoing. For this reason the value read from this register is valid only when the PVU bit in the IWDG status register (IWDG_SR) is set to 0.

11.5.3 Reload Register (IWDG_RLR)

Bit	Symbol	Description	Reset value
31:12	-	Reserved	0x0
11:0	RL	Watchdog counter reload value These bits are write access protected see Register access protection. They are written by software to define the value to be loaded in the watchdog counter each time the value AAAAh is written in the IWDG key register (IWDG_KR). The watchdog counter counts down from this value. The timeout period is a function of this value and the clock prescaler. The RVU bit in the IWDG status register (IWDG_SR) must be set to 0 to be able to change the reload value.	0xfff

Caution: Reading this register returns the reload value from the VDD voltage domain. This value may not be up to date/valid if a write operation to this register is ongoing on it. For this reason the value read from this register is valid only when the RVU bit in the IWDG status register (IWDG_SR) is set to 0.

11.5.4 Status Register (IWDG_SR)

Bit	Symbol	Description	Reset value
31:2	-	Reserved	0x0
1	RVU	Watchdog counter reload value update This bit is set by hardware to indicate that an update of the reload value is ongoing. It is reset by hardware when the reload value update operation is completed in the VDD voltage domain (takes up to five RC 32 kHz cycles). Reload value can be updated only when RVU bit is set to 0.	0
0	PVU	Watchdog prescaler value update This bit is set by hardware to indicate that an update of the prescaler value is ongoing. It is reset by hardware when the prescaler update operation is completed in the VDD voltage domain (takes up to five RC 32 kHz cycles). Prescaler value can be updated only when PVU bit is set to 0.	0

Caution: If several reload, prescaler, or window values are used by the application, it is mandatory to wait until RVU bit is reset before changing the reload value, to wait until PVU bit is reset before changing the prescaler value. However, after updating the prescaler and/or the reload/window value it is not necessary to wait until RVU or PVU is reset before continuing code execution except in case of low-power mode entry.

Chapter 12 Universal Serial Communication Unit

This product is equipped with one universal serial communication unit. Unit 0 has two serial channels, each capable of supporting 3-wire serial (SSPI), UART, and simplified I²C communication functions. The channel function allocation is as follows:

Table 12-1: Universal serial channel unit channel function allocation

Unit	Channel	Used as SSPI	Used as UART	Used as Simplified I ² C
0	0	SSPI00	UART0	IIC00
	1	-		-

Cautions:

1. “-” indicates that this function is not supported in this model;
2. When using UART0 on Channel 0 and Channel 1 of Unit 0, SSPI00 and IIC00 cannot be used.

12.1 Functions of Universal Serial Communication Unit

The functions of each serial interface supported by this product are shown below.

12.1.1 3-Wire Serial I/O (SSPI00)

Data is transmitted and received synchronously with a serial clock (SCLK) output from the master control device.

This is a clock synchronization function communicating using 1 SCLK, 1 SDO and 1 SDI.

For specific set-up examples, refer to “12.5 3-wire serial I/O (SSPI00) communication operation”.

[Data transmission and reception]

- 7-bit or 8-bit data length
- Phase control of transmitting and receiving data
- MSB/LSB first
- Level setting of data transmission and reception

[Clock control]

- Master or slave selection
- Phase control of input/output clocks
- Set the transfer period generated by the prescaler and the in-channel counter.
- Maximum transfer rate

Master communication: Max.6MHz

Slave communication: Max.6MHz

[Interrupt function]

- Transfer end interrupt, buffer empty interrupt

[Error detection flag]

- Overflow error

Caution: It must be used within the scope of the SCLK Cycle Time (t_{KCY}) characteristics. Please refer to the data sheet for details.

12.1.2 Simplified SPI with Slave Chip Select Function

It supports SPI serial communication interface with slave chip select input functionality. This is a clock-synchronized communication interface using one slave chip select input (SS), one serial clock (SCLK), one serial data output (SDO), and one serial data input (SDI) for a total of four communication lines.

[Data Transmission and Reception]

- 7-bit and 8-bit data length
- Phase control for data transmission and reception
- MSB/LSB priority selection
- Level setting for data transmission and reception

[Clock control]

- Phase control of input/output clocks
- Set the transfer period generated by the prescaler and the in-channel counter
- Maximum transfer rate

Slave communication: 6MHz

[Interrupt function]

- Transfer end interrupt, buffer empty interrupt

[Error detection flag]

- Overflow error

12.1.3 UART0

This is the function for asynchronous communication using two lines: serial data transmit (TxD) and serial data receive (RxD). With these two communication lines, data is transmitted and received asynchronously with other communication parties using a data frame (composed of a start bit, data, parity bit, and stop bit) and an internal baud rate. Full-duplex UART communication can be achieved by using two dedicated channels, one for transmission (even channel) and one for reception (odd channel).

For specific setup examples, refer to 12.7 UART0 Communication Operation.

[Data transmission and reception]

- 7-bit and 8-bit data length
- MSB/LSB priority selection
- Level setting for data transmission and reception, with inversion selection
- Appending of parity bit, parity function
- Appending of stop bits

[Interrupt function]

- Transfer end interrupt, buffer empty interrupt
- Error interrupts triggered by framing errors, parity errors, or overrun errors

[Error detection flag]

- Framing errors, parity errors, overrun errors

12.1.4 Simplified I²C(IIC00)

This is the function for clock-synchronized communication with multiple devices using two lines: serial clock (SCL) and serial data (SDA). Since this simplified I²C is designed for single-device communication, such as with flash memory, A/D converters, etc., it is used solely as the master device.

For the start and stop conditions, AC specifications must be followed, and software handling is required while operating the control registers. For specific setup examples, refer to 12.8 Simplified I²C (IIC00) Communication Operation.

[Data transmission and reception]

- Master sends, master receives (master function for single master only)
- ACK output function ^{Note}, ACK detection function
- 8-bit data length (when sending an address, the high 7 bits represent the address, and the lowest bit controls R/W)
- Start and stop conditions generated via software

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- ACK error, Overflow error

[Unsupported Simplified I²C Functions]

- Slave transmission, slave reception
- Arbitration failure detection function
- Wait detection function

Note: When receiving the last data, if 0 is written to the SOEm (Serial Output Enable register m) to stop serial data output, ACK will not be output. For detailed information, refer to 12.8.3(2) Handling Process.

Remark: For using a fully functional I²C bus, refer to Chapter 14 I²C Interface.

12.2 Structure of Universal Serial Communication Unit

The universal serial communication unit consists of the following hardware.

Table 12-2: Structure of universal serial communication unit

Item	Structure
Shift register	16-bit
Buffer register	Serial data register mn(SDRmn) ^{Note}
Serial clock input/output	SCLKOI00 pin (for 3-wire serial I/O), SCL00 pin (for simplified I ² C)
Serial data input	SDI00 pin (for 3-wire serial I/O), RxD0 pin (for UART)
Serial data output	SDO00 pin (for 3-wire serial I/O), TxD0 pin (for UART)
Serial data input/output	SDA00 pin (for simplified I ² C)
Slave select input	SAU0_SS pin (for serial communication unit 0 used as SPI slave select input function)
Control registers	< Register of unit setting section > • Peripheral enable register 0 (PER0) • Serial clock selection register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOM) • Serial output level register m (SOLm) • Input switch control register (ISC) • Noise filter enable register 0 (NFEN0)
	<Register for each channel> • Serial data register mn (SDRmn) • Serial mode register mn (SMRmn) • Serial communication operation setting register mn (SCRmn) • Serial status register mn (SSRmn) • Serial flag clear trigger register mn (SIRmn)
	• Port alternate function configuration register (PS_xx) • Port output mode register (POMxx) • Port mode register (PMxx) • Port register (Pxx)

Note: when SEmn=1;

Remark: m: unit number (m=0);

n: channel number (n=0, 1);

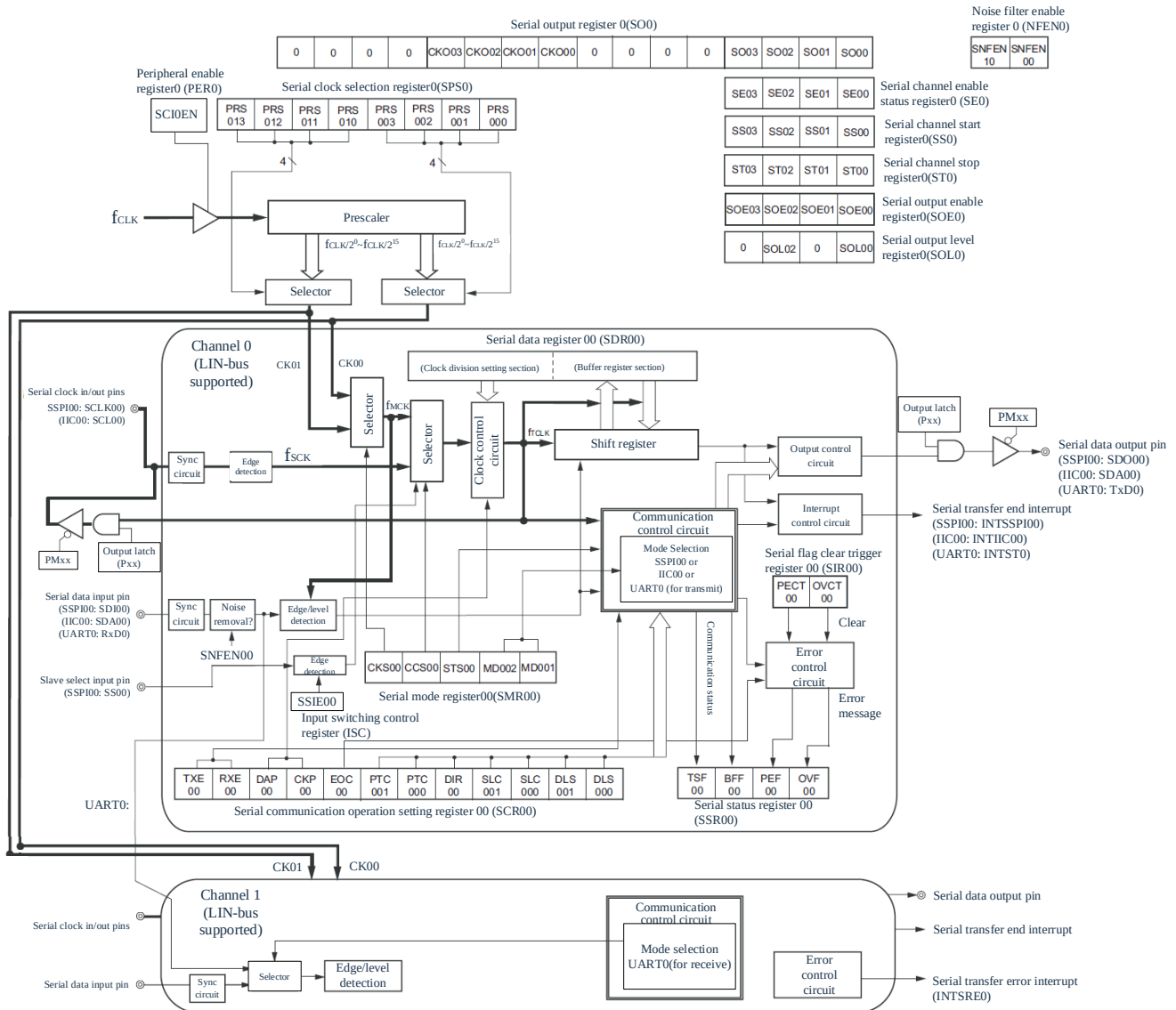
p: SSPI number (p=00);

q: UART number (q=0);

r: IIC number (r=00).

A block diagram of universal serial communication unit 0 is shown in Figure 12-1: (Unit 0)

Figure 12-1: Diagram of universal serial communication unit 0



12.2.1 Shift Register

This is a 16-bit register used for parallel-to-serial and serial-to-parallel conversion.

When receiving data, the input data from the serial input pin is converted to parallel data; when sending data, the value transmitted to this register is output as serial data from the serial output pin. The shift register cannot be directly operated by the program.

To read/write shift register data, use the serial data register mn (SDRmn) during operation (SEmn=1).

Table 12-3: Format of shift register

Bit	Symbol	Description	Reset value
15:0	SDRmn	Serial data shift register	0x0

12.2.2 Serial Data Register mn (SDRmn)

The SDRmn register is the send and receive data register (16 bits) for channel n.

When operation is stopped (SEmn=0), bits 15-9 are used as the frequency division register for the operating clock (FMCK). During operation (SEmn=1), bits 15-9 are used as the transmit/receive buffer register.

When receiving data, parallel data converted from the shift register is saved to the SDRmn register. When sending data, the data transmitted to the shift register is sent from the SDRmn register.

For data output sequence, based on the serial communication operation setting register mn (SCRmn) bits 1-0 (DLSmn1~DLSmn0), data saved in the SDRmn register is as follows:

- 7-bit data length (saved in SDRmn bits 0-6)
- 8-bit data length (saved in SDRmn bits 0-7)

The SDRmn register can be read/written in 16-bit units.

When SEMn=1, the low 8 bits of the SDRmn register can be read/written as SDRmnL in 8-bit units.^{Note}

Based on the communication mode, the following SFR names can be used to read/write the SDRmnL register:

- SSPIp communication.....SDIOpL
- UARTq reception.....RXDq(UARTq receive data register)
- UARTq transmission.....TXDq(UARTq transmit data register)
- IICr communication.....SDIOr(IICr data register)

After generating a reset signal, the value of the SDRmn register becomes 0000H.

Note: When the operation is stopped (SEmn=0), writing to SDRmn[7:0] through an 8-bit memory manipulation instruction is prohibited (otherwise, SDRmn[15:9] will be cleared to 0).

Remark: m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00);

q: UART number (q=0);

r: IIC number (r=00).

Table 12-4: Format of serial data register mn (SDRmn)

Bit	Symbol	Description	Reset value
15:0	SDRmn	Serial Data Register	0x0

Remark:

- For the function of the higher 7 bits of the SDRmn register, please refer to 12.3 Registers for Controlling the General Serial Communication Unit.
- m: unit number (m=0);
n: channel number (n=0, 1).

12.3 Registers for Controlling Universal Serial Communication Unit

The registers for controlling the Universal Serial Communication Unit are as follows:

1. Peripheral Enable Register 0 (PER0)
2. Serial Clock Select Register m (SPSm)
3. Serial Mode Register mn (SMRmn)
4. Serial Communication Operation Configuration Register mn (SCRmn)
5. Serial Data Register mn (SDRmn)
6. Serial Flag Clear Trigger Register mn (SDIRmn)
7. Serial Status Register mn (SSRmn)
8. Serial Channel Start Register m (SSm)
9. Serial Channel Stop Register m (STm)
10. Serial Channel Enable Status Register m (SEm)
11. Serial Output Enable Register m (SOEm)
12. Serial Output Level Register m (SOLm)
13. Serial Output Register m (SOM)
14. Input Switch Control Register (ISC)
15. Noise Filter Enable Register 0 (NFEN0)
16. Port Alternate Configuration Register (PS_{xx})
17. Port Output Mode Register (POM_x)
18. Port Mode Register (PM_x)
19. Port Register (P_x)

Remark: m: unit number (m=0);

n: channel number (n=0, 1).

List of Serial Communication Unit Registers:

Register base address: 0x40041500

Table 12-5: Register list

Offset address	Register name	R/W	Reset value
0x000	SSRm0	R	0000H
0x002	SSRm1	R	0000H
0x008	SIRm0	R/W	0000H
0x00A	SIRm1	R/W	0000H
0x010	SMRm0	R/W	0020H
0x012	SMRm1	R/W	0020H
0x018	SCRm0	R/W	0083H
0x01A	SCRm1	R/W	0083H
0x020	SEm	R/O	0000H
0x022	SSm	R/W	0000H
0x024	STm	R/W	0000H
0x026	SPSm	R/W	0000H
0x028	SOM	R/W	0303H
0x02A	SOEm	R/W	0000H
0x034	SOLm	R/W	0000H
0x040	SDRm0	R/W	0000H
0x042	SDRm1	R/W	0000H

Remark: unit number m=0.

(Register base address = 0x4004_0470) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
NFEN0	0x00	R/W	Noise Filter Enable Register 0	0x00
ISC	0x03	R/W	Input Switch Control Register	0x00

12.3.1 Peripheral Enable Register (PER0)

PER0 is the register for enabling or disabling the clock supply to peripheral hardware. By stopping the clock supply to unused hardware, power consumption and noise can be reduced.

To use Universal Serial Communication Unit 0, the bit 3 (SAUEN) of PER0 must be set to 1.

Table 12-6: Format of peripheral enable register0 (PER0)

Bit	Symbol	Description	Reset value
7	LPTIMEN	Control of 15-bit interval timer input clock supply 0: Stop to supply the input clock, the SFR used by the 15-bit interval timer cannot be written. 1: Supply the input clock, the SFR used by the 15-bit interval timer can be read and written.	0
6	-	Reserved	0
5	ADCEN	Control of ADC module input clock supply 0: Stop to supply the input clock, ADC cannot run. 1: Supply the input clock, ADC can run.	0
4	-	Reserved	0
3	SAUEN	Control of serial communication unit input clock supply 0: Stop to supply the input clock, the SFR used by the SAU cannot be written. 1: Supply the input clock, the SFR used by the SAU can be read and written.	0
2:0	-	Reserved	0

Caution: To set the universal serial communication unit m, the following registers must first be set when the SAUmEN bit is “1”. When the SAUmEN bit is “0”, the write operation of the control register of the universal serial communication unit m is ignored, and the read value is the initial value (except for input switching control register (ISC), noise filter enable register 0 (NFEN0), port alternate function configuration register (PS_xx), and port output mode register (POMx), port mode registers (PMx), port mode control registers (PMCx), and port registers (Px).

1. Serial clock selection register m (SPSm)
2. Serial mode register mn (SMRmn)
3. Serial communication operation configuration register mn (SCRmn)
4. Serial data register mn (SDRmn)
5. Serial flag clear trigger register mn (SIRmn)
6. Serial status register mn (SSRmn)
7. Serial channel start register m (SSm)
8. Serial channel stop register m (STm)
9. Serial channel enable status register m (SEm)
10. Serial output enable register m (SOEm)
11. Serial output level register m (SOLm)
12. Serial output register m (SOM)

12.3.2 Serial Clock Select Register m (SPSm)

The SPSm register is a 16-bit register that selects the two common runtime clocks (CKm0, CKm1) provided to each channel. The CKm1 clock is selected using bits 7 to 4, and the CKm0 clock is selected using bits 3 to 0.

It is prohibited to rewrite the SPSm register during operation (when SEMn = 1).

The SPSm register is set by a 16-bit memory manipulation instruction.

The low 8 bits of the SPSm register can be set using SPSmL and 8-bit memory manipulation instructions.

After the reset signal is generated, the value of the SPSm register changes to “0000H”.

Table 12-7: Format of serial clock selection register m (SPSm)

Bit	Symbol	Description	Reset value
15:8	-	Reserved	0x0
7:4	PRSm13~PRSm10	Operation clock CKm1 selection 0000: F _{CLK} 0001: F _{CLK} /2 0010: F _{CLK} /2 ² 0011: F _{CLK} /2 ³ 0100: F _{CLK} /2 ⁴ 0101: F _{CLK} /2 ⁵ 0110: F _{CLK} /2 ⁶ 0111: F _{CLK} /2 ⁷ 1000: F _{CLK} /2 ⁸ 1001: F _{CLK} /2 ⁹ 1010: F _{CLK} /2 ¹⁰ 1011: F _{CLK} /2 ¹¹ 1100: F _{CLK} /2 ¹² 1101: F _{CLK} /2 ¹³ 1110: F _{CLK} /2 ¹⁴ 1111: F _{CLK} /2 ¹⁵	0x0
3:0	PRSm03~PRSm00	Operation clock CKm0 selection 0000: F _{CLK} 0001: F _{CLK} /2 0010: F _{CLK} /2 ² 0011: F _{CLK} /2 ³ 0100: F _{CLK} /2 ⁴ 0101: F _{CLK} /2 ⁵ 0110: F _{CLK} /2 ⁶ 0111: F _{CLK} /2 ⁷ 1000: F _{CLK} /2 ⁸ 1001: F _{CLK} /2 ⁹ 1010: F _{CLK} /2 ¹⁰ 1011: F _{CLK} /2 ¹¹ 1100: F _{CLK} /2 ¹² 1101: F _{CLK} /2 ¹³ 1110: F _{CLK} /2 ¹⁴ 1111: F _{CLK} /2 ¹⁵	0x0

Note: To change the clock selected as F_{CLK} (change the value of the system clock control register (CKC)) during the operation of the Universal Serial Communication Unit (SCI), the operation of the SCI must be stopped (serial channel stop register m (STm)=000FF) and then make the change.

Caution: Bits 15~8 must be set to “0”.

Remark:

F_{CLK}: CPU/peripheral hardware clock frequency;

m: unit number (m=0);

k=0, 1.

12.3.3 Serial Mode Register mn(SMRmn)

The SMRmn register is used to configure the operating mode of channel n. It allows you to select the operating clock (f_{MCK}), specify whether the serial clock (f_{SCLK}) input can be used, set the start trigger, configure the operation mode (SSPI, UART, or simplified I²C), and select the interrupt source. Additionally, in UART mode only, it allows the configuration of the inverted level for received data.

It is prohibited to rewrite the SMRmn register during operation (when SEmn = 1). However, the MDmn0 bit can be rewritten during operation.

The SMRmn register is set by a 16-bit memory manipulation instruction.

After the reset signal is generated, the value of the SMRmn register changes to “0020H”.

Table 12-8: Format of serial mode register mn (SMRmn)

Bit	Symbol	Description	Reset value
15	CKSmn	Selection of channel n operating clock (f_{MCK}) 0: The operating clock CKm0 set by the SPSm register 1: The operating clock CKm1 set by the SPSm register The operating clock (f_{MCK}) is used for edge detection circuit. The transfer clock (f_{TCLK}) is generated by setting the CCSmn bit and the high 7 bits of the SDRmn register.	0
14	CCSmn	Selection of channel n transmit clock (f_{TCLK}) 0: The operation clock f_{MCK} division clock specified by CKSmn bit 1: The input clock f_{SCLK} from the SCLKp pin (slave transfer in SSPI mode). The transfer clock f_{TCLK} is used for shift registers, communication control circuits, output controllers, interrupt control circuits, and error control circuits. When the CCSmn bit is “0”, the operation clock (f_{MCK}) is divided by the high 7 bits of the SDRmn register.	0
13:10	-	-	-
8	STSmn ^{Note 1}	Selection of start triggering sources 0: Only software triggers are valid (selected in SSPI, UART transmit, Simplified I ² C) 1: The effective edge of the RxDq pin (selected in UART reception) The transfer begins when the above conditions are met after setting the SSm register to “1”.	0
7	-	-	-
6	SISmn0 ^{Note 1}	Level inversion control for receiving data on channel n in UART mode 0: Detect the falling edge as the start bit. Do not invert the input communication data. 1: Detect the rising edge as the start bit. Invert the input communication data.	0x0
5:3	-	-	0x4
2:1	MDmn2~MDmn1	Channel n operating mode settings 00: SSPI mode 01: UART mode 10: Simplified I ² C mode 11: Settings are disabled.	0x0
0	MDmn0	Channel n interrupt source selection 0: Transfer end interrupt 1: Buffer empty interrupt: Triggered when data is transferred from the Shift Data Register (SDRmn) to the shift register. If the MDmn0 bit is 1 and the SDRmn register is empty, the next transmission data is written for continuous transmission.	0

Note 1: Limited to SMR01, SMR03, SMR11, SMR21, SMR31, SMR41 registers;

Caution: Set bits 13~9, 7, 4, 3(bits 13~6, 4, 3 of SMR00, SMR02, SMR10, SMR20, SMR30, SMR40) to “0”, and set bit5 to 1”.

Remark: m: unit number (m=0); n: channel number (n=0, 1); p: SSPI number (p=00);

q: UART number (q=0); r: IIC number (r=00).

12.3.4 Serial Communication Operation Configuration Register mn (SCRmn)

The SCRmn register is the communication operation control register for channel n. It is used to configure settings such as data transmission and reception modes, data and clock phases, error signal masking, parity bits, start bits, stop bits, and data length.

Rewriting the SCRmn register is prohibited during operation (when SEMn = 1).

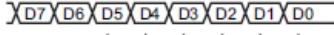
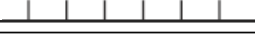
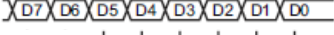
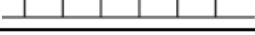
The SCRmn register is set by a 16-bit memory manipulation instruction.

After the reset signal is generated, the value of the SCRmn register changes to “0083H”.

Table 12-9: Format of serial communication operation configuration register mn (SCRmn)

Bit	Symbol	Description	Reset value
15:14	TXEmn,RXEmn	Setting of the channel n operating mode 00: Disable communication. 01: Receive only. 10: Transmit only. 11: Enable transmission and reception.	0x0
13:12	DAPmn,CKPmn	Data and clock phase selection in SSPI mode 00: Type 1 01: Type 2 10: Type 3 11: Type 4 In UART mode, set DAPmn and CKPmn to 0	0x0
11	-	-	0
10	EOCmn	Error interrupt signal (INTSREx(x=0~1)) mask control 0: Disable generating error interrupts INTSREx(generating INTSRx). 1: Enable generating error interrupts INTSREx(No INTSRx is generated during error). The EOCmn bit must be set to 0 in SSPI mode and Simplified I2C mode, or when transmitting the UART ^{Note 2}	0x0
9:8	PTCmn1~PTCmn0	Setting of parity bits in UART mode Transmission 00: No parity bits are output 01: Output 0 parity ^{Note 3} 10: Output even check 11: Output odd check Reception No parity is received Parity is not judged Judged as even parity Judged as odd parity Be sure to set PTCmn1, PTCmn0 = 0, 0 in the SSPI mode and simplified I2C mode.	0x0
7	DIRmn	Selection of data transfer sequence in SSPI and UART modes 0: Perform MSB-first input/output. 1: Perform LSB-first input/output. In the simplified I2C mode, the DIRmn bit must be set to 0.	0x1
6	-	-	0x0
5:4	SLCmn1~SLCmn0	Setting of the stop bit in UART mode 00: No stop bit 01: Stop bit length = 1 bit 10: Stop bit length = 2 bits (Limited to mn=00) 11: Prohibit setting. If the transfer end interrupt is selected, the interrupt is generated after all stop bits have been transferred. It must be set to 1 stop bit (SLCmn1, SLCmn0=0, 1) during UART receive or in simplified I2C mode. In SSPI mode, it must be set to no stop bit (SLCmn1, SLCmn0 = 0, 0). It must be set to 1 bit (SLCmn1, SLCmn0=0, 1) or 2 bits (SLCmn1, SLCmn0=1, 0) when UART is sent.	0x0
3:2	-	Reserved	-
1:0	DLSmn1~DLSmn0	Data length setting in SSPI mode 10: Data length of 7 bits (saved in bit0~6 of SDRmn register) Corresponding serial function SSPI UART IIC ○ ○ ×	0x3

		11: Data length of 8 bits (saved in bit0~7 of SDRmn register) ○ ○ ○	
		Other Settings are disabled In simplified I ² C mode, DLSmn1~DLSmn0=11B.	

DAPmn	CKPmn	Data and clock phase selection in SSPI mode	Type
0	0	SCLKp  SDOp  SDIp input timing 	1
0	1	SCLKp  SDOp  SDIp input timing 	2
1	0	SCLKp  SDOp  SDIp input timing 	3
1	1	SCLKp  SDOp  SDIp input timing 	4
in UART mode and simplified I ² C mode, set the DAPmn bit and CKPmn bit to 0.			

Note 1: Limited to SCR00 register;

Note 2: When the EOCmn bit is 0 and the SSPImn is not used, it is possible to generate an error interrupt INTSREn.

Caution: Bits 6, 10 and 11 must be set to 0 (bits 5 of the SCR01 register must be set to 0).

Remark: m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00).

12.3.5 Serial Data Register mn (SDRmn)

The SDRmn register is the data register (16 bits) for channel n transmission and reception.

When the operation stops (SEmn=0), bit15~9 is used as a prescaler setting register for the operating clock (F_{MCK}). During operation (SEmn=1), bit15~9 is used as a transmit and receive buffer register.

If the CCSmn bit of the serial mode register mn (SMRmn) is set to 0, the frequency division clock of the running clock set by bit15~9 (High 7 bits) of the SDRmn register is used as the transfer clock.

The SIRmn register is set by a 16-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “0000H”.

Table 12-10: Format of serial data register mn (SDRmn)

Bit	Symbol	Description	Reset value
15:9		Transmit clock setting for operating clock 7'h0: F_{MCK} 7'h1: $F_{MCK}/2$ 7'h2: $F_{MCK}/3$ 7'h3: $F_{MCK}/4$ • • 7'h7E: $F_{MCK}/127$ 7'h7F: $F_{MCK}/128$ When the operation is stopped (SEmn = 0), bits 15-9 are used as the clock division setting register for the operation clock (F_{MCK}). During operation (SEmn = 1), bits 15~9 are used as the transmit and receive buffer register.	0x0
8:0		During operation (SEmn = 1), bits 8:0 are used as the transmit and receive buffer register.	0x0

Cautions:

1. When operation is stopped (SEmn=0), bit8~0 must be cleared to zero.
2. When using UART, it is prohibited to set SDRmn[15:9] to 0000000B and 0000001B.
3. When using Simplified I²C, it is prohibited to set SDRmn[15:9] to 0000000B, and the setting value of SDRmn[15:9] must be greater than or equal to 0000001B.
4. When operation is stopped (SEmn=0), it is prohibited to rewrite SDRmn[7:0] by 8-bit memory manipulation instructions (otherwise, SDRmn[15:9] are cleared).

Remark:

For the function of the SDRmn register during operation, please refer to “12.2 Structure of universal serial communication unit”.

m: unit number (m=0);

n: channel number (n=0, 1).

12.3.6 Serial Flag Clear Trigger Register mn (SDIRmn)

This is the trigger register used to clear the error flags of channel n.

If each bit (FECTmn, PECTmn, OVCTmn) is set to 1, the corresponding bit (FEFmn, PEFmn, OVFmn) of the serial status register mn (SSRmn) is cleared to 0. Since the SDIRmn register is a trigger register, clearing the corresponding bit of the SSRmn register also clears the SDIRmn register immediately.

The SIRmn register is set by a 16-bit memory manipulation instruction.

It can use SIRmnL and set the lower 8 bits of the SIRmn register with an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of the SIRmn register changes to “0000H”.

Table 12-11: Format of serial flag clear trigger register mn (SIRmn)

Bit	Symbol	Description	Reset value
15:3	-	-	0x0
2	FECTmn ^{Note 1}	Clear trigger of framing error flag of channel n 0: Not cleared 1: Clear the FEFmn bit of the SSRmn register to 0.	0
1	PECTmn	Clear trigger of parity error flag of channel n 0: Not cleared 1: Clear the PEFmn bit of the SSRmn register to 0.	0
0	OVCTmn	Clear trigger of overrun error flag of channel n 0: Not cleared 1: Clear the OVFmn bit of the SSRmn register to 0.	0

Note 1: Limited to SIR01 register;

Caution: The bits 15~3 (SIR00 and SIR10 registers are bits 15~2) must be set to 0.

Remark:

m: unit number (m=0);

n: channel number (n=0, 1);

The read value of the SIRmn register is always “0000H”.

12.3.7 Serial Status Register mn (SSRmn)

The SSRmn register is a register that indicates the communication status and error occurrence status of channel n. The errors indicated by this register are a framing error, parity error, and overrun error. The SSRmn register is read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSRmn register can be read with an 8-bit memory manipulation instruction with SSRmnL.

After a reset signal is generated, the value of the SSRmn register changes to “0000H”.

Table 12-12: Format of serial status register mn (SSRmn) (1/2)

Bit	Symbol	Description	Reset value
15:7	-	-	0x0
6	TSMn	Communication status indication flag of channel n 0: Communication is stopped or suspended. 1: Communication is in progress. [Clear condition] • When STMn of STM register is set to 1 (communication is stopped) or SSMn bit of SSM register is set to 1 (communication is suspended) • When the communication ends [Set condition] • When communication starts	0x0
5	BFFmn	Buffer register status indication flag of channel n 0: Valid data is not stored in the SDRmn register. 1: Valid data is stored in the SDRmn register. [Clear condition] • When the transmit data from the SDRmn register to the shift register is completed during the transmit process • When the received data is read from the SDRmn register during the receive process • When the STMn bit of STM register is set to 1 (communication is stopped) or the SSMn bit of the SSM register is set to 1 (communication is enabled) [Set condition] • When writing and transmitting data to the SDRmn register in the state where the TXEmn bit of the SCRmn register is 1 (the transmit mode, transmit and receive mode in each communication mode). • When saving the receive data to the SDRmn register in the state where the RXEmn bit of the SCRmn register is 1 (receive mode, transmit and receive mode in each communication mode). • When a receive error occurs	0x0
4:3	-	-	0x0
2	FEFmn ^{Note 1}	Detection flag for channel n frame error 0: No error occurred. 1: An error occurred (in UART reception) [Clear condition] • When writing 1 to the FECTmn bit of the SIRmn register [Set condition] • When no stop bit is detected at the end of UART reception	0x0
1	PEFmn	Detection flag for channel n parity error 0: No error occurred. 1: An error occurred (in UART reception) or ACK was not detected (in I ² C transmission) [Clear condition] • When writing 1 to the PECTmn bit of the SIRmn register [Set condition] • When sending data with different parity and parity bits (parity errors) at the end of UART reception • At the time of I ² C transmission and at the time of ACK reception the slave did not return an ACK signal (no ACK detected)	0x0 0x0
0	OVFmn	Detection flag for channel n overrun error 0: No error occurred. 1: An error occurred. [Clear condition]	0x0

		• When writing 1 to the OVCTmn bit of the SIRmn register [Set condition] • In the SCRmn register, the RXEmn bit is 1 (reception mode, transmission and reception mode in each communication mode), and when the send data is written without reading the received data or when the next receive data is written, even though the receive data is stored in the SDRmn register. • Data is not ready to be sent during a slave transmission or slave transmission and reception in SSPI mode	

Note 1: Limited to SSR01 register.

Remark: If the BFFmn bit is set to 1 when writing to the SDRmn register, the saved transmit or receive data will be discarded, and an overrun error (OVEmn = 1) will be detected.

m: unit number (m=0)

n: channel number (n=0, 1)

12.3.8 Serial Channel Start Register m (SSm)

The SSm register is a trigger register that is used to enable starting communication/counting of each channel.

If each bit (SSmn) is set to 1, the corresponding bit (SEmn) of the serial channel enable status register m (SEm) is set to 1 (operation enable state). Since the SSmn bit is a trigger bit, the SSmn bit is cleared immediately if the SEmn bit is 1.

The SSm register is set by a 16-bit memory manipulation instruction.

The SSmL can be used and the lower 8 bits of the SSm register can be set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of the SSm register changes to “0000H”.

Table 12-13: Format of serial channel start register m (SSm)

Bit	Symbol	Description	Reset value
15:2	-	-	0x0
1	SSm1	Operation start trigger of channel n 0: No trigger operation. 1: Set the SEmn bit to 1 and enter communication wait status ^{Note} .	0
0	SSm0	Operation start trigger of channel n 0: No trigger operation. 1: Set the SEmn bit to 1 and enter communication wait status ^{Note} .	0

Note: If the SSmn bit is set to 1 during communication, communication is stopped and the standby state is entered. At this time, the control register and shift register values, the SCLKmn pin and SDOmn pin, the FEFmn flag, the PEFmn flag, and the OVFn flag remain in state.

Caution: Bits 15 to 2 of SS0 register must be set to 0.

Remark:

m: unit number (m=0);

n: channel number (n=0, 1);

The read value of the SSm register is always 0000H.

12.3.9 Serial Channel Stop Register m (STm)

The STm register is a trigger register that is used to enable stopping communication/counting of each channel.

If each bit (STmn) is set to 1, the corresponding bit (SEmn) of the serial channel enable status register m (SEm) is cleared to 0 (operation stop state). Since the STmn bit is a trigger bit, if the SEmn bit is 0, the STmn bit is cleared immediately.

The STm register is set by a 16-bit memory manipulation instruction.

The STmL can be used and the lower 8 bits of the STm register can be set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of the STm register changes to “0000H”.

Table 12-14: Format of serial channel stop register m (STm)

Bit	Symbol	Description	Reset value
15:2	-	-	0x0
1	STm1	Operation stop trigger of channel n 0: No trigger operation. 1: Clear the SEmn bit to 0 and stop the communication operation ^{Note} .	0
0	STm0	Operation stop trigger of channel n 0: No trigger operation. 1: Clear the SEmn bit to 0 and stop the communication operation ^{Note} .	0

Note: The control register and shift register values, the SCLKmn pin and SDOmn pin, and the FEFmn flag, PEFmn flag, and OVFn flag remain in state.

Caution: Be sure to clear bits 15 to 2 of ST0 register.

Remark:

m: unit number (m=0);

n: channel number (n=0, 1);

The read value of STm register is always “0000H”.

12.3.10 Serial Channel Enable Status Register m (SEm)

The SEm register indicates whether data transmission/reception operation of each channel is enabled or stopped.

If each bit of the serial start enable register m (SSm) is set to 1, the corresponding bit is set to 1. If each bit of the serial channel stop register m (STm) is set to 1, its corresponding bit is cleared to 0.

Channel n that is enabled to operate cannot rewrite by software the value of the CKOmn bit (serial clock output of channel n) of serial output register m (SOM) to be described below, and a value reflected by a communication operation is output from the serial clock pin.

Channel n that stops operation can set the value of the CKOmn bit of the SOM register by software and output its value from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SEm register is read by a 16-bit memory manipulation instruction.

The SEmL can be used and the lower 8 bits of the SEm register can be read with an 8-bit memory manipulation instruction.

The value of the SEm register changes to “0000H” after a reset signal is generated.

Table 12-15: Format of serial channel enable status register m (SEm)

Bit	Symbol	Description	Reset value
15:2	-	-	0x0
1	SEm1	Indication of operation enable/stop status of channel n 0: Operation stopped 1: Operation enabled	0
0	SEm0	Indication of operation enable/stop status of channel n 0: Operation stopped 1: Operation enabled	0

Remark: m: unit number (m=0);

n: channel number (n=0, 1).

12.3.11 Serial Output Enable Register m (SOEm)

The SOEm register is a register that is used to enable or stop output of the serial communication operation of each channel.

For channel n that allows serial output, the value of the SOMn bit of the serial output register m (SOM) described later cannot be rewritten by software, and the value reflected by the communication operation is output from the serial data output pin.

For channel n where serial output is stopped, the value of the SOMn bit of the SOM register can be set by software, and the value is output from the serial data output pin. As a result, arbitrary waveforms such as start conditions or stop conditions can be generated by software.

The SOEm register is set by a 16-bit memory manipulation instruction.

The SOEmL can be used and the lower 8 bits of the SOEm register can be set by an 8-bit memory manipulation instruction.

When a reset signal is generated, the value of the SOEm register changes to “0000H”.

Table 12-16: Format of serial output enable register m (SOEm)

Bit	Symbol	Description	Reset value
15:2	-	-	0x0
1	SOEm1	Serial output enable/stop of channel n 0: Stop output by serial communication operation. 1: Enable output by serial communication operation.	0
0	SOEm0	Serial output enable/stop of channel n 0: Stop output by serial communication operation. 1: Enable output by serial communication operation.	0

Remark: Be sure to clear bits 15 to 2 of SOE0 register.

m: unit number (m=0);

n: channel number (n=0, 1).

12.3.12 Serial Output Register m (SOM)

The SOM register is a buffer register for serial output of each channel.

The value of the SOMn bit of this register is output from the serial data output pin of channel n.

The value of the CKOm bit of this register is output from the serial clock output pin of channel n.

The SOMn bit of this register can be overwritten by software only if serial output is prohibited (SOEmn=0). When serial output (SOEmn=1) is allowed, the value of the SOMn bit of this register can only be changed by serial communication, ignoring the rewriting of the software.

The CKOm bit of this register can be overridden by software only if the channel is stopped running (SEmn=0). When the channel is allowed to run (SEmn=1), the value of the CKOm bit of this register can only be changed by serial communication, ignoring the rewriting of the software.

To use a serial interface pin as a non-serial interface function such as a port function, the corresponding CKOm bit and SOMn bit must be set to 1.

The SOM register is set by a 16-bit memory manipulation instruction.

After a reset signal is generated, the value of the SOM register changes to “0303H”.

Table 12-17: Format of serial output register m (SOM)

Bit	Symbol	Description	Reset value
15:10	-	-	0x0
9	CKOm1	Serial clock output of channel n 0: The output value of the serial clock is 0. 1: The output value of the serial clock is 1.	1
8	CKOm0	Serial clock output of channel n 0: The output value of the serial clock is 0. 1: The output value of the serial clock is 1.	1
7:2			0x0
1	SOM1	Serial data output of channel n 0: The output value of serial data is 0. 1: The output value of serial data is 1.	1
0	SOM0	Serial data output of channel n 0: The output value of serial data is 0. 1: The output value of serial data is 1.	1

Caution: Bits 15~10 and bits 7~2 of the SO0 register must be set to 0.

Remark: m: unit number (m=0);

n: channel number (n=0, 1).

12.3.13 Serial Output Level Register m (SOLm)

The SOLm register is a register that sets the reverse phase of the data output level of each channel.

This register can only be set in UART mode. In SSPI mode and Simplified I2C mode, the corresponding bit must be set to 0. Only when serial output is allowed (SOEmn=1), the inverted setting of each channel n of this register is reflected to the pin output. When serial output is prohibited (SOEmn=0), the value of the SOMn bit is output directly. Rewriting the SOLm register during operation (SEmn=1) is prohibited.

The SOLm register is set by a 16-bit memory manipulation instruction.

Set the lower 8 bits of the SOLm register with an 8-bit memory manipulation instruction with SOLmL.

After a reset signal is generated, the value of the SOLm register changes to “0000H”.

Table 12-18: Format of serial output level register m(SOLm)

Bit	Symbol	Description	Reset value
15:1	-	-	0x0
0	SOLm0	Select inversion of the level of the transmit data of channel n in UART mode 0: The communication data is output directly. 1: Invert the communication data.	0x0

Caution: Bits 15~1 of SOL0 register must be set to 0.

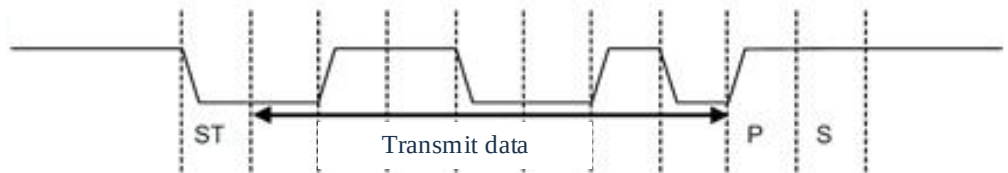
Remark: m: unit number (m=0);

n: channel number (n=0, 1).

When performing UART transmission, an example of level inversion for the transmitted data would be:

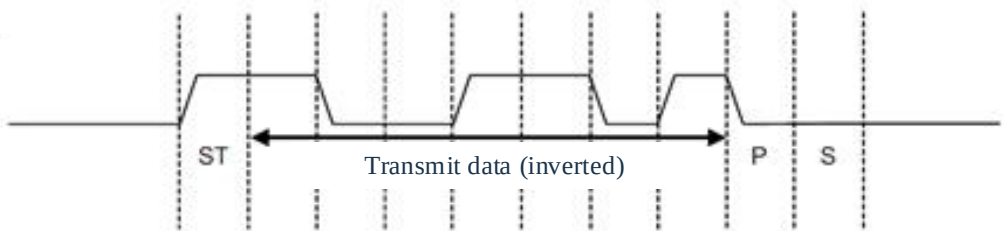
(a) Normal output (SOLmn=0)

SOLmn=0 output
TxDq



(b) Inverted output (SOLmn=1)

SOLmn=1 output
TxDq



Remark: m: unit number (m=0);

n: channel number (n=0, 1).

12.3.14 Input Switch Control Register (ISC)

When LIN-bus communication is implemented via UART0, the ISC1 bit and ISC0 bit of the ISC register are used for external interrupt and timer array unit coordination. If bit0 is set to 1, the input signal of the serial data input (RxD0) pin is selected as the input for the external interrupt (INTP0), so that the wake-up signal can be detected by the INTP0 interrupt.

If bit1 is set to 1, the input signal of the serial data input (RxD0) pin is selected as the input to the timer, so that the wake-up signal can be detected by the timer, and the width of the low level of the break field and the pulse width of the sync field can be measured.

The SS1E00 bit controls the SS00 pin input of channel 0 in the slave mode of SSPI00 communication. During the period when a high level is input to the SS00 pin, transmission and reception are not performed even if a serial clock is input; during the period when a low level is input to the SS00 pin, transmission and reception are performed if a serial clock is input, depending on the settings of each mode.

The ISC register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of the ISC register changes to “00H”.

Table 12-19: Format of input switch control register (ISC)

Bit	Symbol	Description	Reset value
7	--	Reserved	0
6	SSIE00	Setting of SS00 input for channel 0 in slave mode of SSPI00 communication 0: SS00 pin input is invalid. 1: SS00 pin input is valid.	0
5:0	--	Reserved	0

Caution: Bits 7 and 5~0 must be set to 0.

12.3.15 Noise Filter Enable Register 0 (NFEN0)

The NFEN0 register is used to set whether the noise filter can be used for the input signal from the serial data input pin to each channel.

Disable the noise filter of the pin used for SSPI or simplified I²C communication, by clearing the corresponding bit of this register to 0. Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

When the noise filter is valid, the consistency of the 2 clocks is detected after synchronization via the operation clock (F_{MCK}) of the target channel; when the noise filter is invalid, synchronization is performed only via the operation clock (F_{MCK}) of the target channel.

The NFEN0 register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of the NFEN0 register changes to “00H”.

Table 12-20: Format of noise filter enable register 0 (NFEN0)

Bit	Symbol	Description	Reset value
7:3	-	Reserved	0x0
2	SNFEN10	Use of noise filter of RXD0 pin 0: Noise filter OFF 1: Noise filter ON Set the SNFEN10 bit to 1 to use the RXD0 pin. Clear the SNFEN10 bit to 0 to use the function other than RxD0 pin.	0x0
1:0	-	Reserved	0x0

Caution: Bits 7~3, Bits 1~0 must be set to 0.

12.3.16 Registers Controlling Port Functions of Serial I/O Pins

When using the Universal Serial Communication Unit, the control registers for the alternate port function (Port Mode Register (PMxx), Port Alternate Function Configuration Register (PxxCFG), Port Output Mode Register (POMxx), and Port Mode Control Register (PMCxx)) must be set.

For details, please refer to Chapter 2 Port Functions.

When using the alternated port of serial data output pin or serial clock output pin as serial data output or serial clock output, the bit of Port Mode Control Register (PMCxx) and the bit of Port Mode Register (PMxx) corresponding to each port must be set to 0. In this case, the Port Register (Pxx) bit can be 0 or 1.

In addition, when used in N-channel open-drain output mode, the bit of the port output mode register (POMxx) corresponding to each port must be set to 1.

When using the alternated port of the serial data input pin or serial clock input pin as serial data input or serial clock input, you must set the bit of the Port Mode Register (PMxx) corresponding to each port to 1 and set the bit of the Port Mode Control Register (PMCxx) to 0. In this case, the Port Register (Pxx) bits can be 0 or 1.

12.4 Run Stop Mode

Each serial interface of the universal serial communication unit has a run stop mode. Serial communication is not possible in the run stop mode, so power consumption can be reduced. In addition, pin used for serial interface can be used as port function in idle mode.

12.4.1 Stopping the Operation by Units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

The PER0 register is used to enable or disable supplying the clock to the peripheral hardware. Reduce power consumption and noise by stopping clocks to unused hardware.

To stop Universal Serial Communication Unit 0, the bit3 (SAUEN) of the PER0 must be set to 0.

Table 12-21: Setting of peripheral enable register 0 (PER0) in unit-based stop operation

Bit	Symbol	Description	Reset value
7	LPTIMEN	Control of 15-bit interval timer input clock supply 0: Stop to supply the input clock, the SFR used by the 15-bit interval timer cannot be written. 1: Supply the input clock, the SFR used by the 15-bit interval timer can be read and written.	0
6	-	Reserved	0
5	ADCEN	Control of ADC input clock supply 0: Stop to supply the input clock, ADC cannot run. 1: Supply the input clock, ADC can run.	0
4	-	Reserved	0
3	SAUEN	Control of serial communication unit input clock supply 0: Stop to supply the input clock, the SFR used by the SAU cannot be written. 1: Supply the input clock, the SFR used by the SAU can be read and written.	0
2:0	-	Reserved	0

Remark:

- When the SAUmEN bit is 0, the write operation of the control register of the universal serial communication unit m is ignored and the read values are initial. However, the following registers are excluded:
 - Input switch control register (ISC)
 - Noise filter enable register 0 (NFEN0)
 - Port alternate function configuration register (PS_xx)
 - Port output mode register (POMx)
 - Port mode register (PMx)
 - Port register (Px)
- ×: This is the unused bit of the universal serial communication unit (depending on the setting of other peripheral functions).
0/1: Set to 0 or 1 according to the user.

12.4.2 Stopping the Operation by Channels

Stop operation by channel via each of the following register settings.

(1) Serial channel stop register m (STm).....This is a register that sets the communication/stop count for each channel.

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	STm3	STm2	STm1	STm0

- ① Clear the SEMn bit to “0” and stop communication operation
- ② Because the STmn bit is a trigger bit, clear the STmn bit immediately if the SEMn bit is 0.

(2) Serial channel enable state register m (SEm).....The register indicates the running or stopping state of data transmission and reception of each channel.

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	SEm3	SEm2	SEm1	SEm0

- ① 0: Operation stop state
- ② The SEm register is a read-only state register, which stops running through the STm register. For a channel that has stopped running, the value of the CKOmn bit of the SOM register can be set by software.

(3) Serial output enable register m(SOEm).....This is a register that sets the enable or stop of serial communication output for each channel.

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3	SOEm2	SOEm1	SOEm0

- ① 0: Stop outputs by serial communication operation
- ② For channels that have stopped serial output, the value of the SOmn bit of the SOM register can be set by software.

(4) Serial output register m (SOM).....This is the buffer register for the serial output of each channel.

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM	0	0	0	0	CKOm3	CKOm2	CKOm1	CKOm0	0	0	0	0	SOM3	SOM2	SOM1	SOM0

- ① CKOm3/2/1/0: 1: serial clock output value = “1”; 0: serial clock output value = “0”
- ② SOM3/2/1/0: 1: serial data output value = “1”; 0: serial data output value = “0”
- ③ When the pin corresponding to each channel is used as a port function, the corresponding CKOmn bit and SOMn bit must be set to 1.

Remark: Limited to universal serial communication unit 0;

m: unit number (m=0);

n: channel number (n=0, 1);

0/1: Set 0 or 1 according to the user’s purpose.

12.5 3-Wire Serial I/O (SSPI00) Communication

This is a clock synchronization communication function implemented by three wires of serial clock (SCLK) and serial data (SDI and SDO).

[Data transmission and reception]

- Data length of 7 bits, 8 bits
- Phase control for transmitting and receiving data
- MSB/LSB FIRST
- Level setting for transmitting and receiving data

[Clock control]

- Master or slave selection
- Phase control of input/output clocks
- Set the transfer period generated by the prescaler and the in-channel counter.
- Maximum transfer rate
- Master communication: Max.6MHz
- Slave communication: Max.6MHz

[Interrupt function]

- Transfer end interrupt, buffer empty interrupt

[Error detection flag]

- Overflow error

Channel 0 of SCIO supports 3-wire serial I/O (SSPI00).

There are six communication modes for 3-wire serial I/O (SSPI00):

- Master Transmission (see Section 12.5.1)
- Master Reception (see Section 12.5.2)
- Master Transmission and Reception (see Section 12.5.3)
- Slave Transmission (see Section 12.5.4)
- Slave Reception (see Section 12.5.5)
- Slave Transmission and Reception (see Section 12.5.6)

Caution: It must be used within the range specified by the SCLK cycle time (T_{KCY}) characteristics. For more details, refer to the datasheet.

12.5.1 Master Transmission

Master transmission refers to the operation of this product output transmission clock and sending data to other devices.

Table 12-22: Master transmission

3-wire serial I/O	SSPI00
Target channel	Channel of SCIO
Pin used	SCLKOI00, SDO00
Interrupt	INTSSPI00 Transfer end interrupt (in single transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	None
Transfer data length	7 bits, 8 bits
Transfer rate ^{Note}	Max.6M[Hz] Min. $F_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] F_{CLK} : system clock frequency
Data phase	It can be selected by the DAPmn bit of the SCRmn register. DAPmn=0: Start data output when the serial clock starts running. DAPmn=1: Start data output half a clock before the serial clock starts running.
Clock phase	It can be selected by the CKPmn bit of the SCRmn register. CKPmn=0: Non-reverse CKPmn=1: Reverse
Data direction	MSB or LSB first

Note: It must be used within the scope of peripheral functional characteristics that meet this condition and meet the electrical characteristics (see data sheet).

m: unit number (m=0);

n: channel number (n=0, 1);

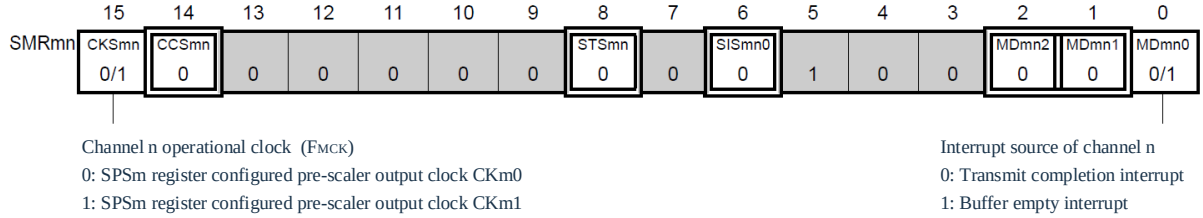
mn=00~01.

(1) Register settings

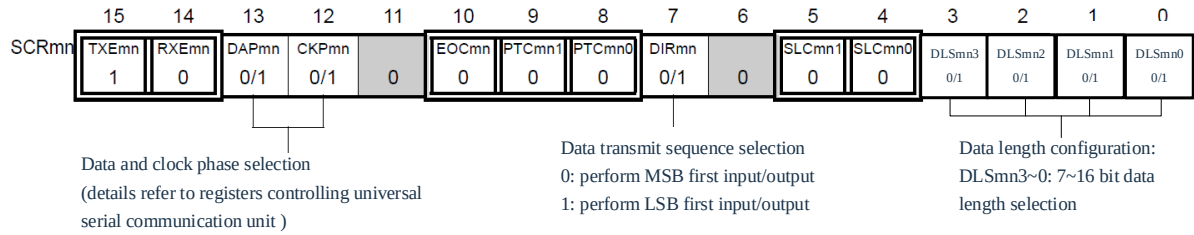
Figure 12-4: 3-wire Serial I/O (SSPI00)

Example of register setting content in master transmission

(a) Serial mode register mn (SMRmn)

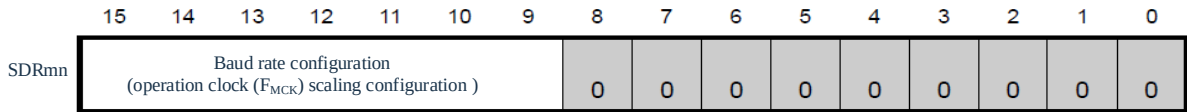


(b) Serial communication operation configuration register mn(SCRmn)

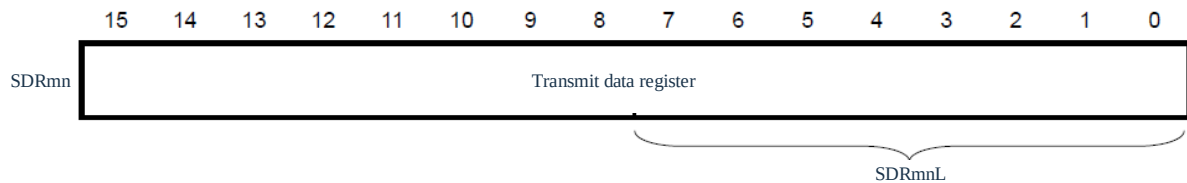


(c) Serial data register mn(SDRmn)

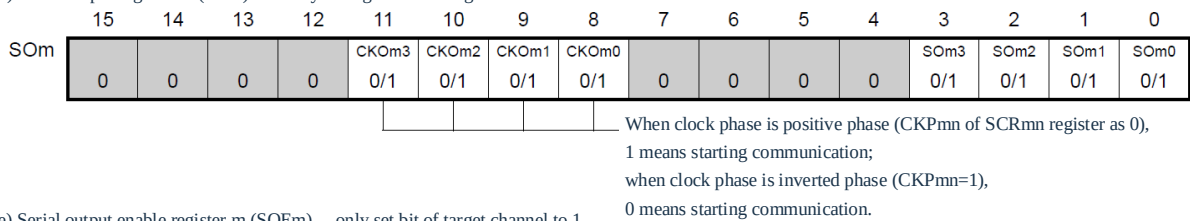
(1) When operation stops (Semn=0)



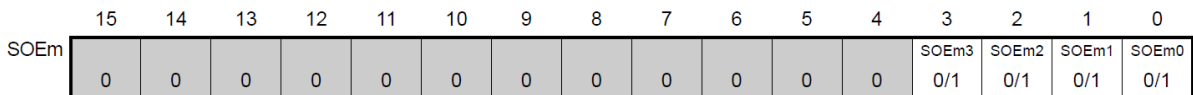
(2) During operation (SEmn=1) (lower 8 bits: SDRmnL)



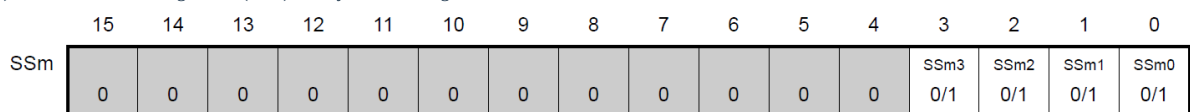
(d) Serial output register m (SOM) Only configure bit of target channel



(e) Serial output enable register m (SOEm).....only set bit of target channel to 1.



(f) Serial channel start register m (SSm).....only set bit of target channel to 1.



Remark: m: unit number (m=0) n: channel number (n=0, 1)

□: Cannot set (set initial value). 0/1: Set 0 or 1 according to the user's purpose.

(2) Procedure

Figure 12-5: Initial setup steps for master transmission



Figure 12-6: Stop steps for master transmission

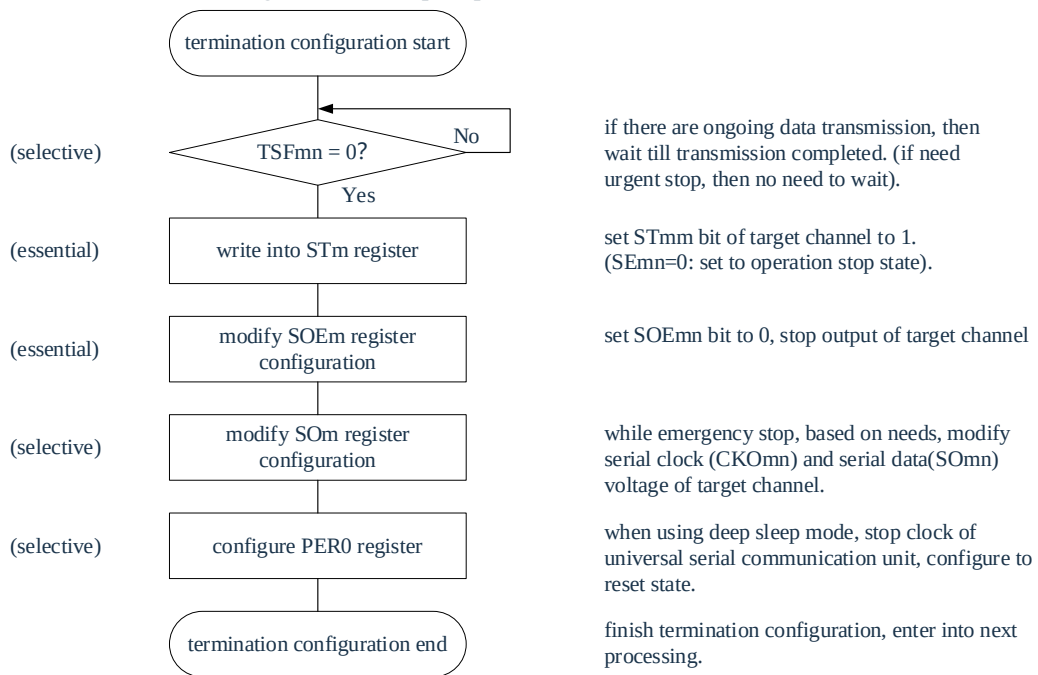
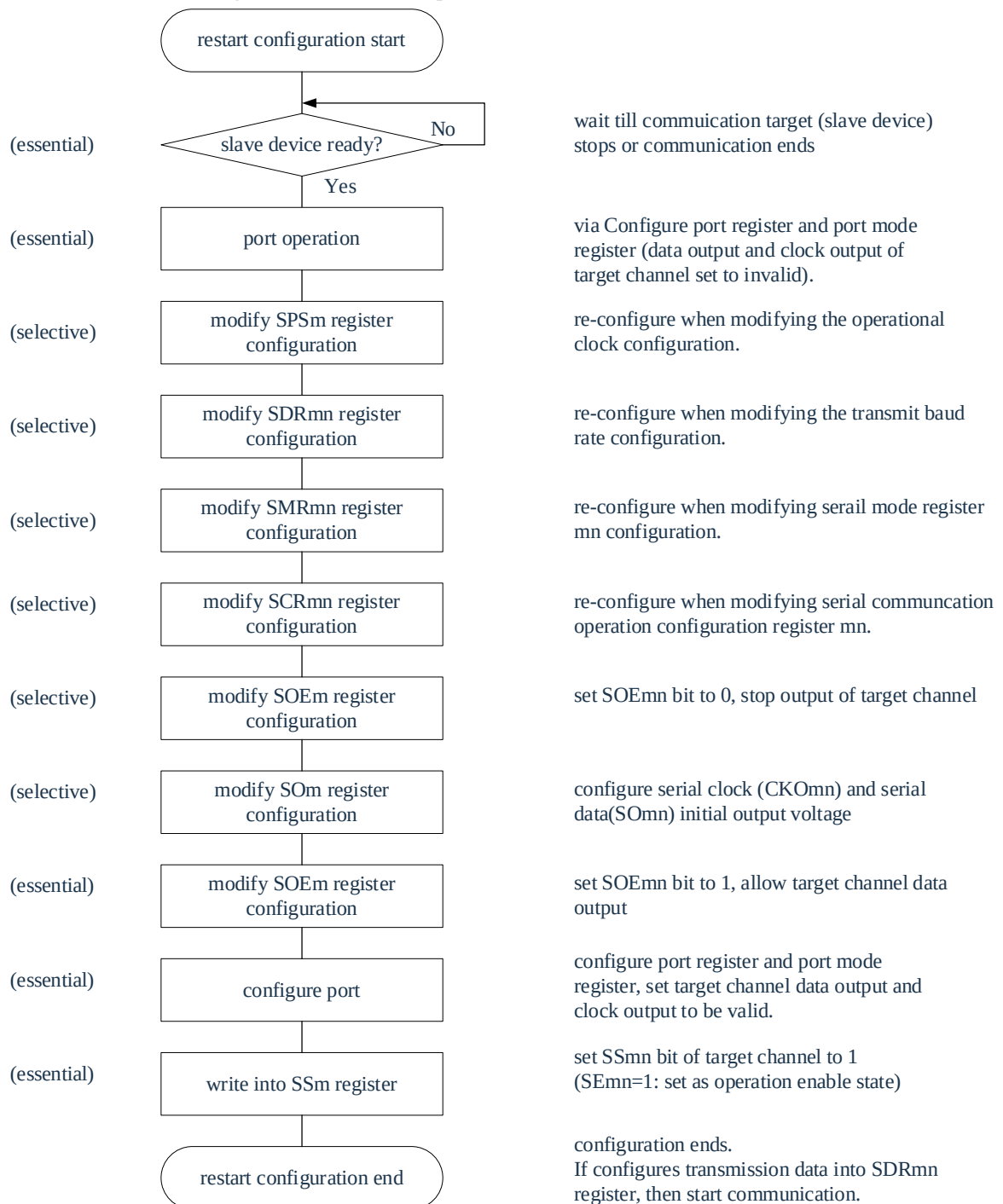


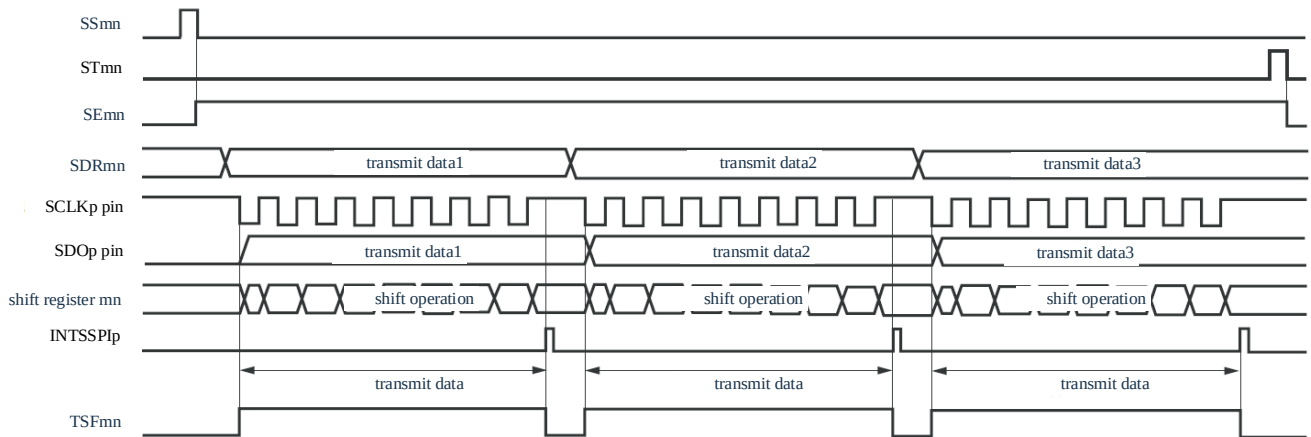
Figure 12-7: Restart steps for master transmission



Caution: If you override PER0 in the abort setting to stop the clock, you must wait until the communication target (slave device) stops or the communication is over to make the initial setting instead of starting the setting again.

(3) Processing flow (single transmit mode)

Figure 12-8: Timing diagram of master transmission (single transmit mode) (type 1: DAPmn=0, CKPmn=0)

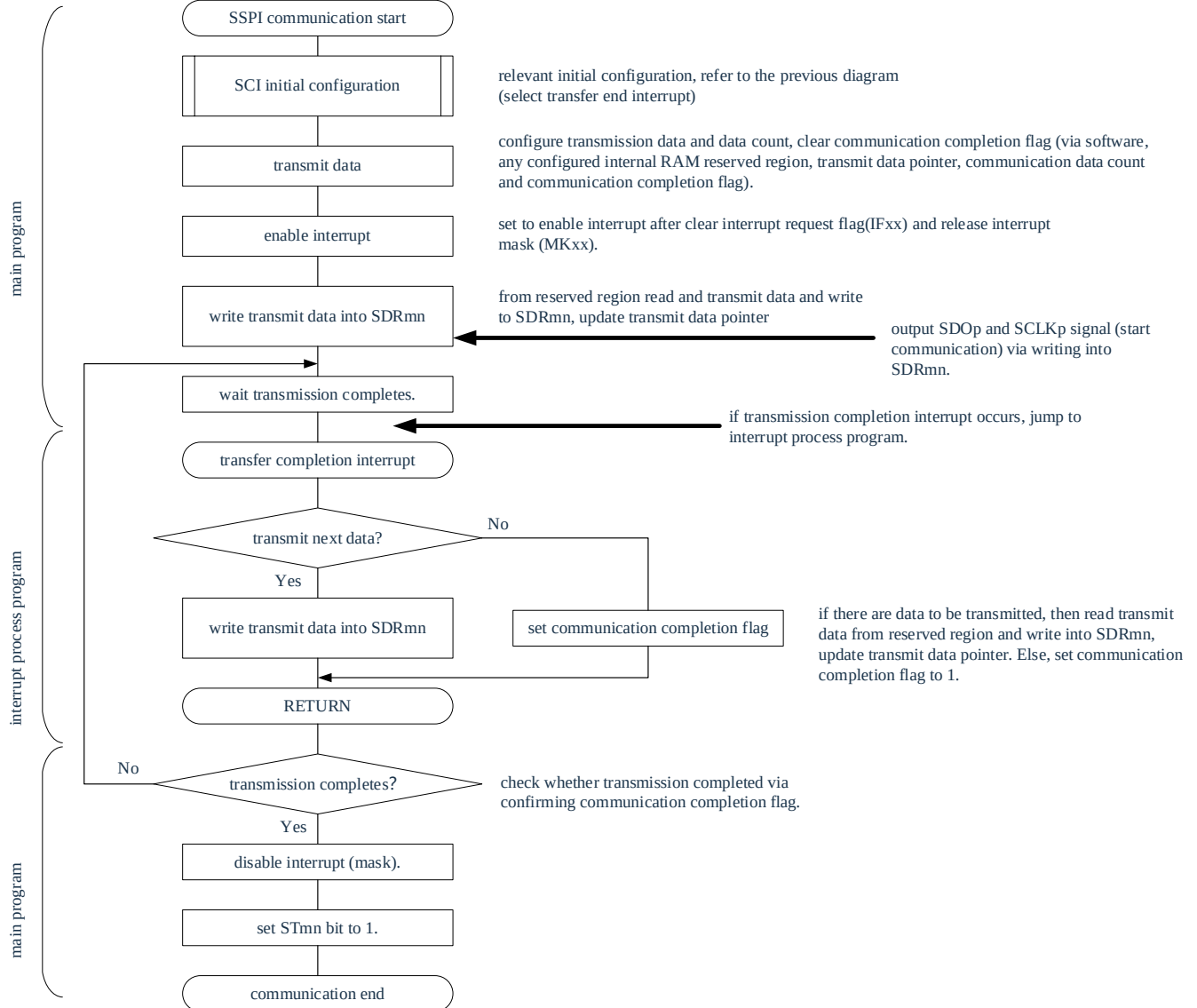


Remark: m: unit number (m=0);

n: channel number (n=0, 1);

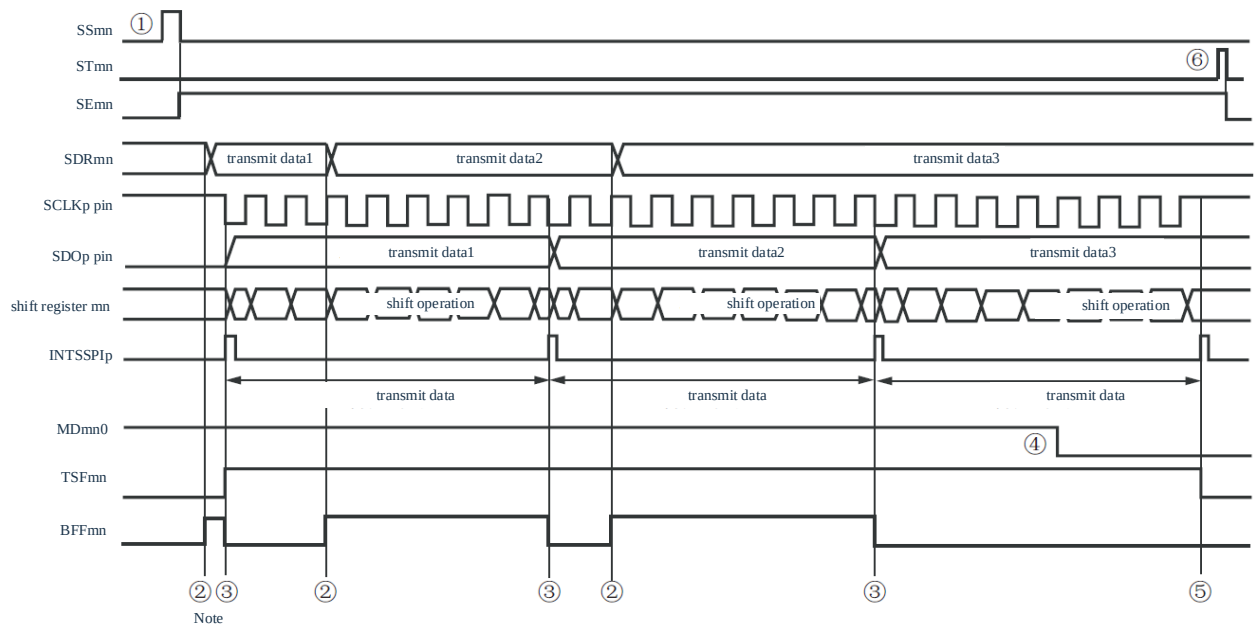
p: SSPI number (p=00).

Figure 12-9: Flow of master transmission (single transmit mode)



(4) Processing flow (continuous transmit mode)

Figure 12-10: Timing diagram of the master transmission (continuous transmit mode)
(type1: DAPmn=0, CKPmn=0)



Note: If the BFFmn bit of the Serial Status Register mn (SSRmn) is 1 (when valid data is stored in the Serial Data Register mn (SDRmn)), writing data to the SDRmn register will overwrite the transmitted data.

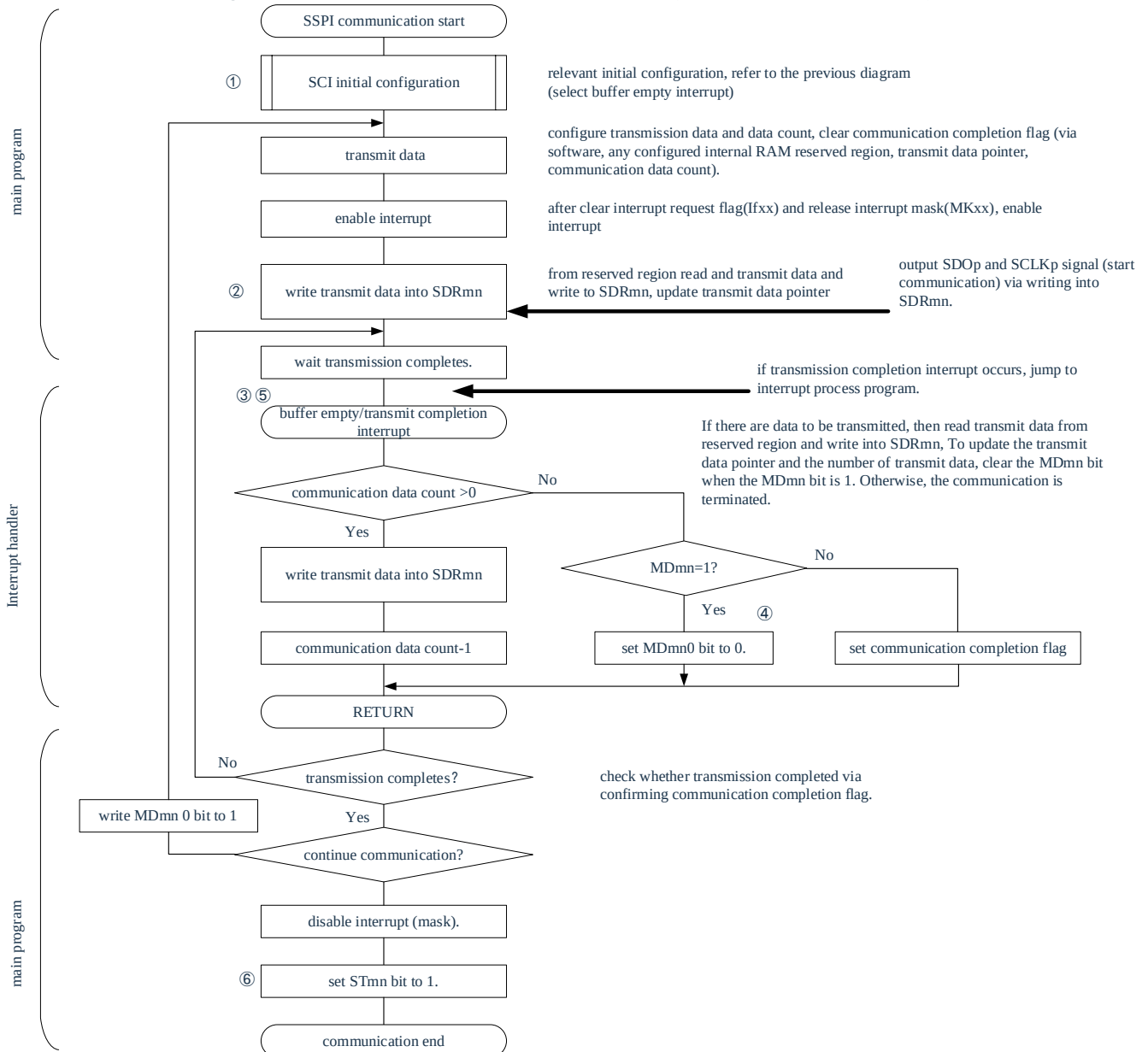
Caution: The MDmn0 bit of the Serial Mode Register mn (SMRmn) can be rewritten even during operation. However, to ensure the last transmitted data is completed before the transmission end interrupt, the rewrite must be done before starting to transmit the last bit.

Remark: m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00).

Figure 12-11: Flowchart of master transmission (continuous transmit mode)



Remark: ① to ⑥ correspond to ① to ⑥ in “Figure 12-10 Timing diagram of the master transmission (continuous transmit mode)”.

12.5.2 Master Reception

Master reception refers to the operation of this product outputs transfer clock and receives data from other devices.

Table 12-23: Master reception

3-wire serial I/O	SSPI00
Target channel	Channel of SCIO
Pin used	SCLKOI00, SDI00
Interrupt	INTSSPI00
	Transfer end interrupt (in single transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Only the overrun error detection flag (OVFmn)
Transfer data length	7 or 8 bits
Transfer rate ^{Note}	Max.6M[Hz] Min. $F_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] F_{CLK} : system clock frequency
Data phase	It can be selected by the DAPmn bit of the SCRmn register. DAPmn=0: Start data output when the serial clock starts running. DAPmn=1: Start data output half a clock before the serial clock starts running.
Clock phase	It can be selected by the CKPmn bit of the SCRmn register. CKPmn=0: Non-reverse CKPmn=1: Reverse
Data direction	MSB or LSB first

Caution: It must be used within the scope of the peripheral functional characteristics that meet this condition and meet the electrical characteristics (refer to the data sheet).

Remark: m: unit number (m=0);

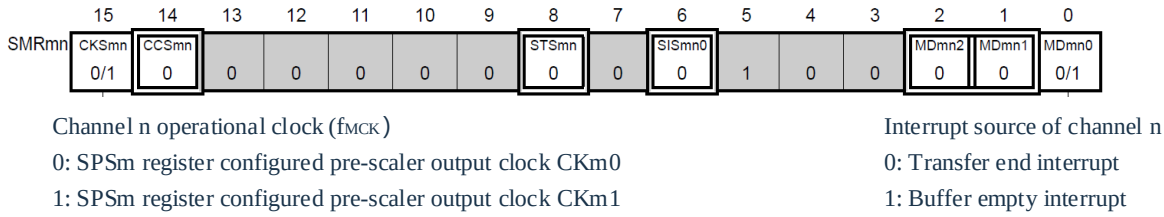
n: channel number (n=0, 1);

p: SSPI number (p=00).

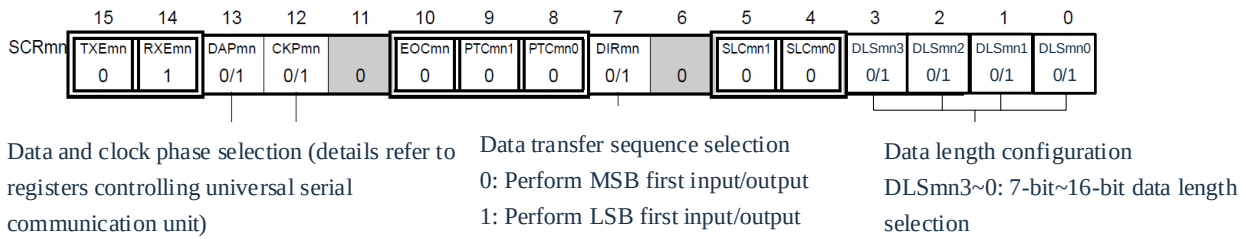
(1) Register setting

Figure 12-12: 3-wire serial I/O(SSPI00)
Example of register setting content when the master receives

(a) Serial mode register mn (SMRmn)

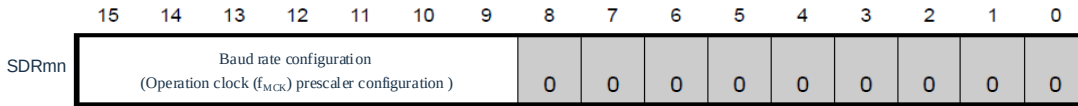


(b) Serial communication operation configuration register mn (SCRmn)

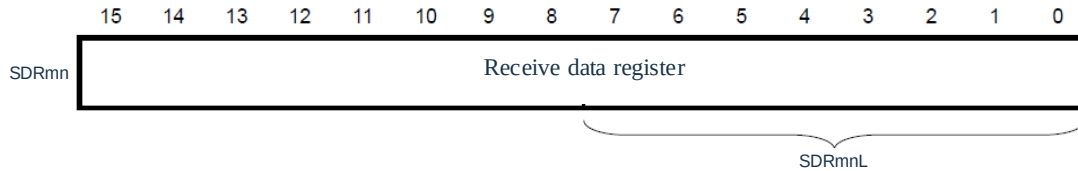


(c) Serial data register mn (SDRmn)

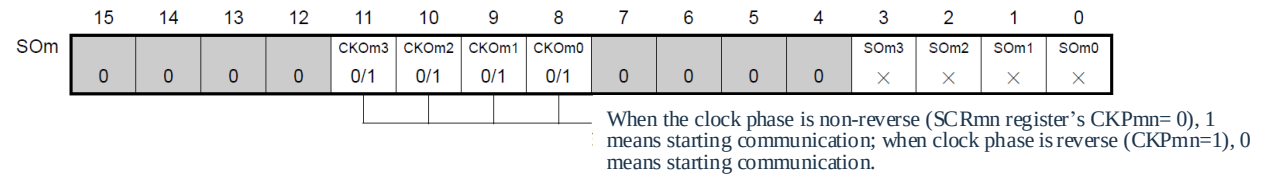
(1) Run stop (SEmn=0)



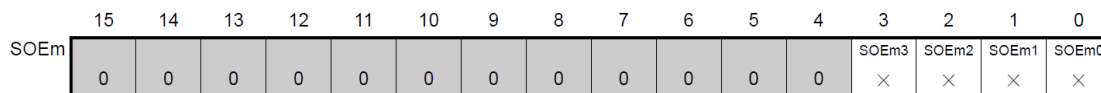
(2) During operation (SEmn=1) (low 8 bits: SDRmnL)



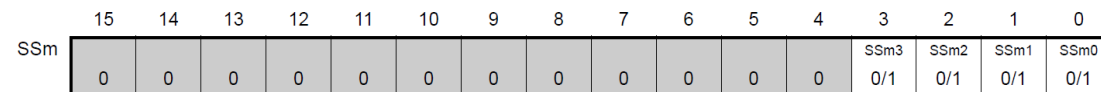
(d) Serial output register m (SOM)..... only configure the target channel bits



(e) Serial output register m (SOEm) Not used in this mode



(f) Serial channel start register m (SSm) only set the bit of the target channel to 1.



Remark: m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00);

□: Fixed in SSPI master receive mode

■: Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set the initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

(2)Procedure

Figure 12-13: Initial setup steps for master reception

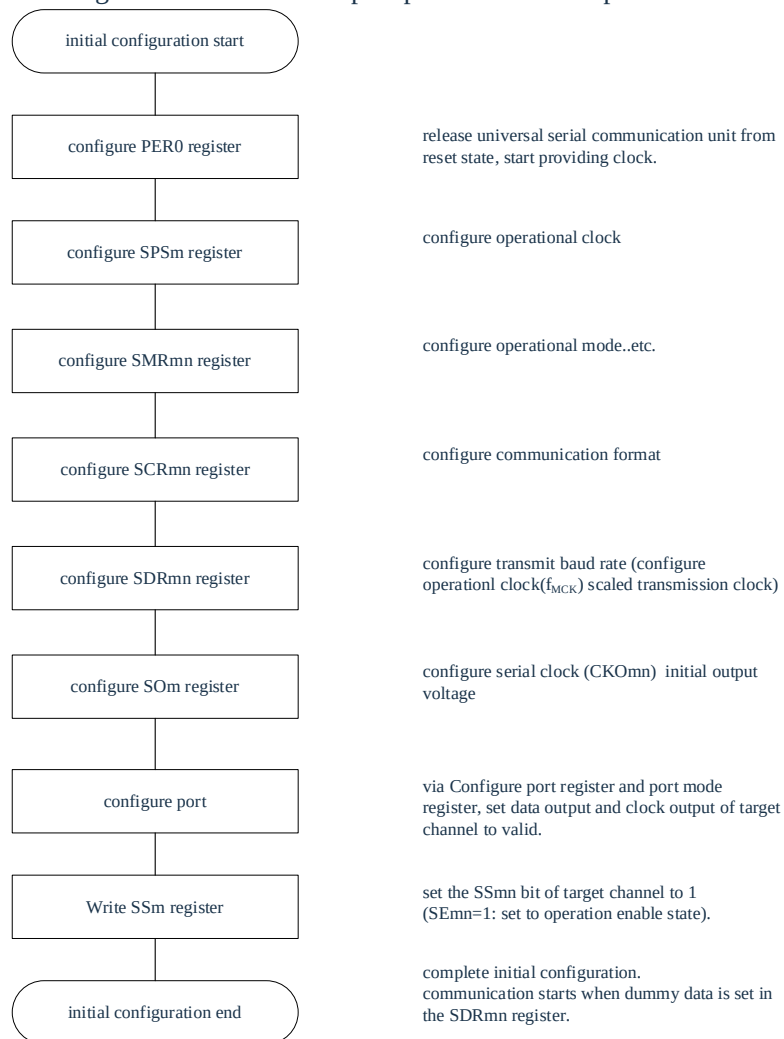


Figure 12-14: Stop steps for master reception

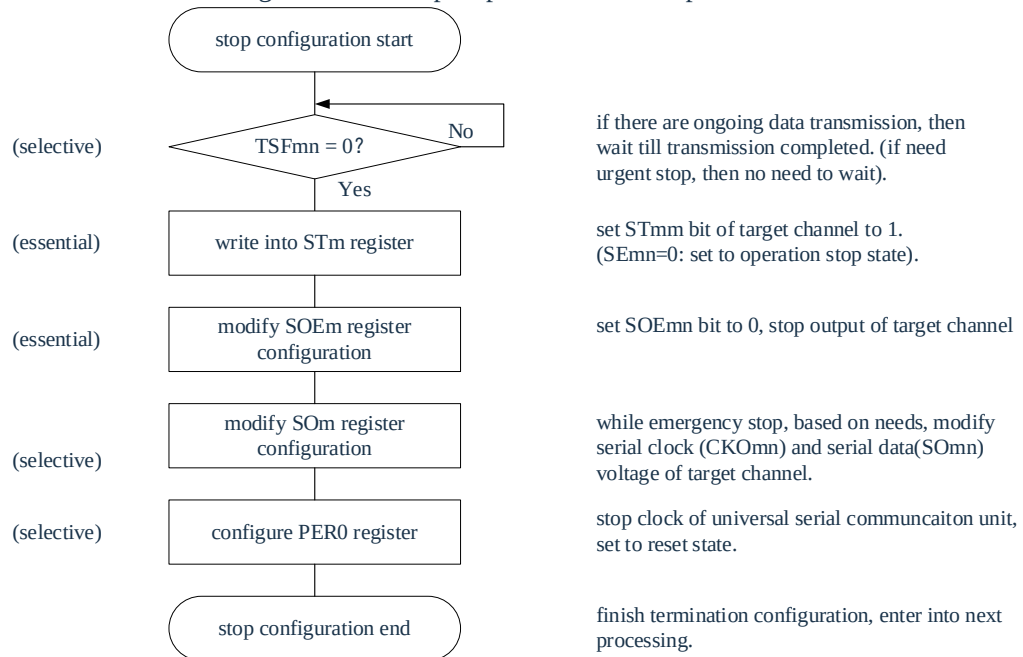


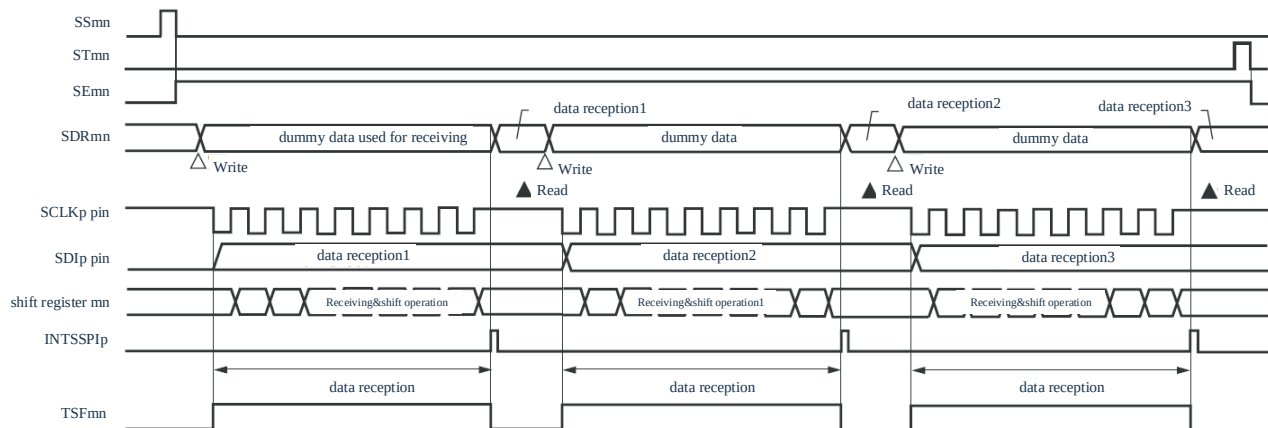
Figure 12-15: Restart steps for master reception



Caution: If you override PER0 in the abort setting to stop the clock, you must wait until the communication target (slave device) stops or the communication is over to make the initial setting instead of starting the setting again.

(3) Processing flow (single receive mode)

Figure 12-16: Timing diagram of master reception (single receive mode) (type 1: DAPmn=0, CKPmn=0)

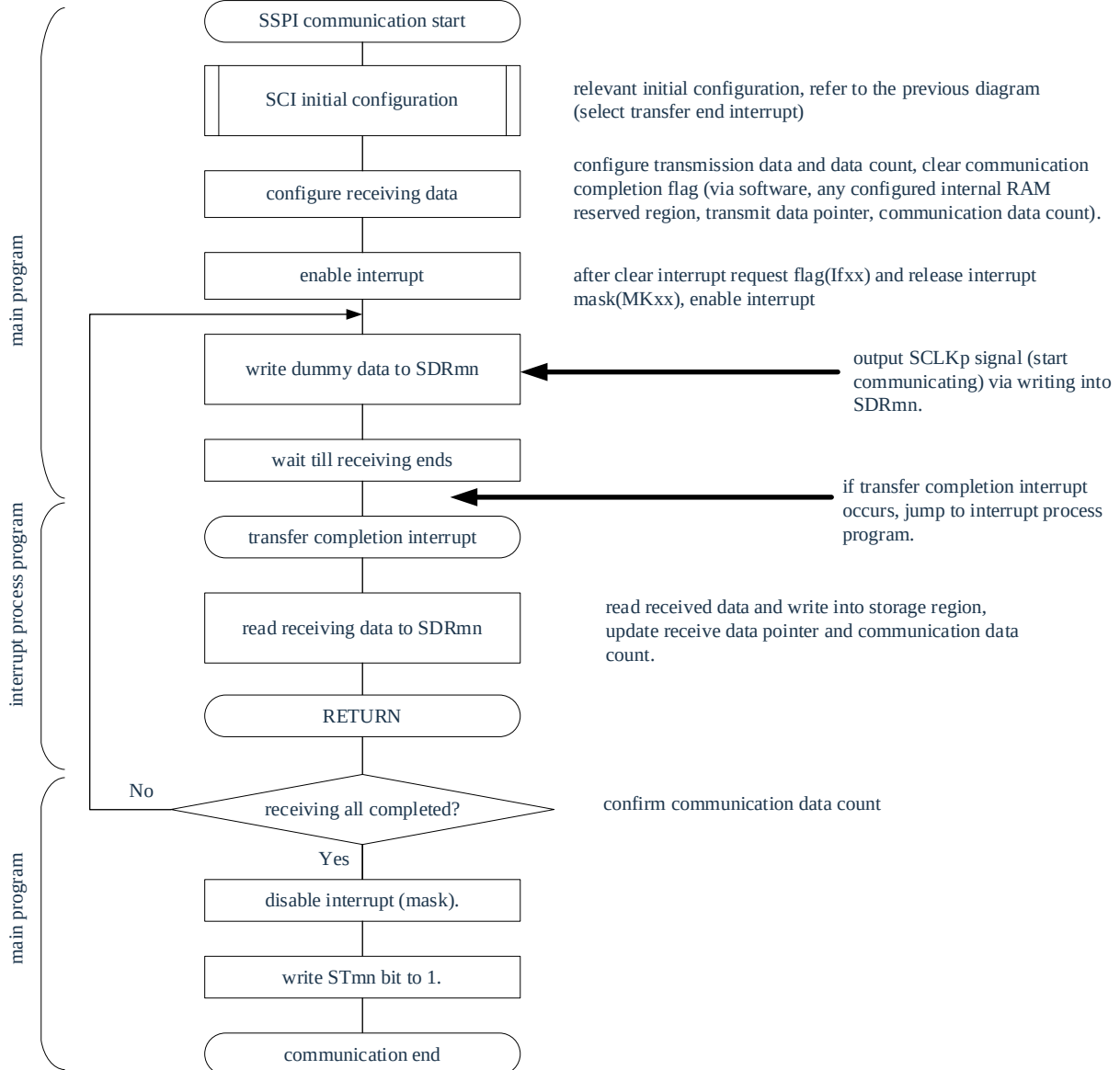


Remark: m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00).

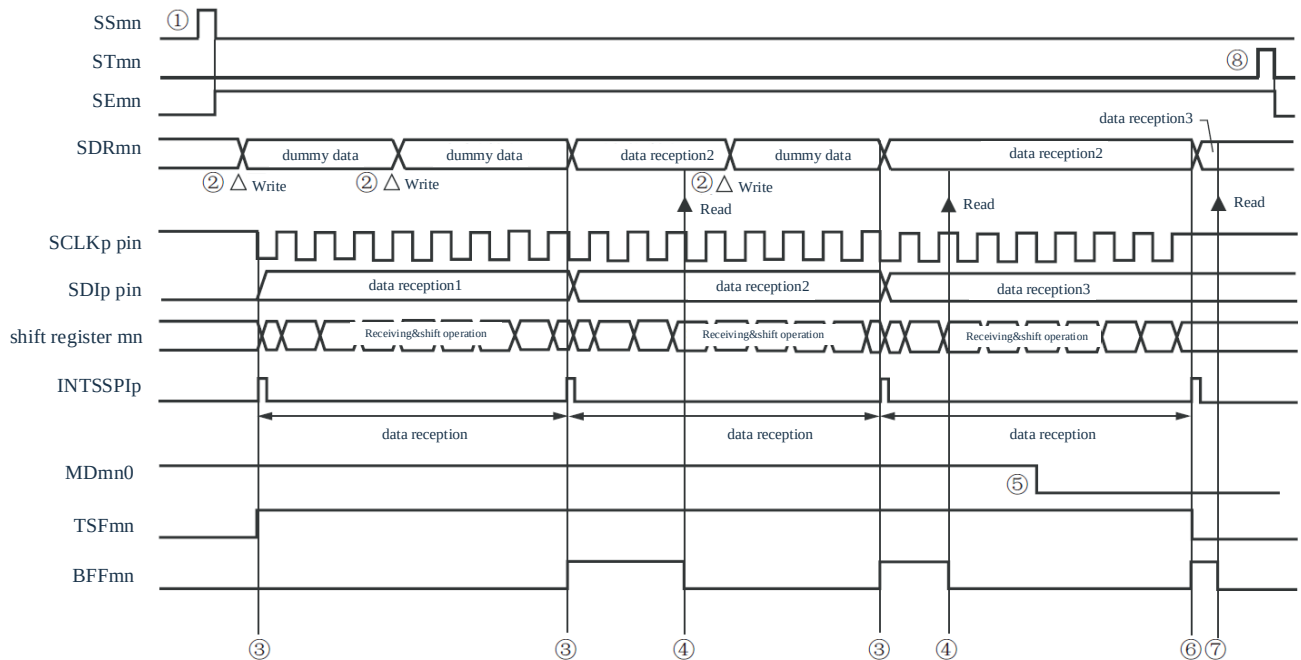
Figure 12-17: Flowchart of master reception (single receive mode)



(4) Processing flow (continuous receive mode)

Figure 12-18: Timing diagram of master reception (continuous receive mode)

(type 1: DAPmn=0, CKPmn=0)



Caution: The MDmn0 bit can be overridden even during operation. However, in order to catch up with the end of transmission interruption of the last received data, it must be overridden before the last bit of reception begins.

Remark:

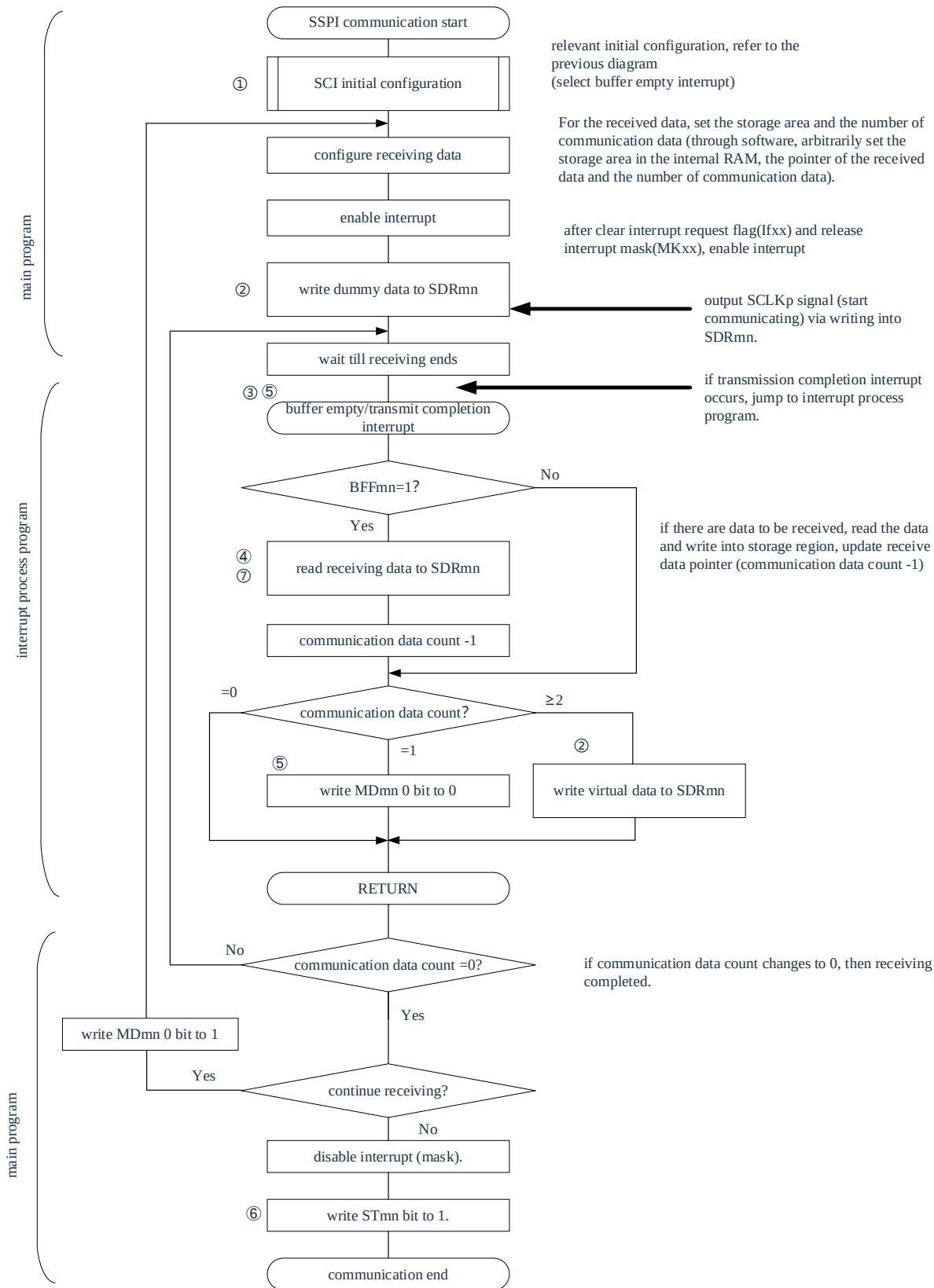
① to ⑧ in the figure correspond to the ① to ⑧ in “Figure 12-19 Flowchart of master reception (continuous receive mode)”.

m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00).

Figure 12-19: Flowchart of master reception (continuous receive mode)



Remark: ① to ⑧ in the diagram correspond to ① to ⑧ in “Figure 12-18 Timing diagram of master reception (continuous receive mode)”.

12.5.3 Master Transmission and Reception

Master transmission/reception is an operation wherein this product outputs a transfer clock and transmits/receives data to/from other device.

Table 12-24: Master transmission/reception

3-wire serial I/O	SSPI00
Target channel	Channel of SCIO
Pin used	SCLKOI00, SDIO0, SDO00
Interrupt	INTSSPI00
	Transfer end interrupt (in single transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Only the overrun error detection flag (OVFmn)
Transfer data length	7 or 8 bits
Transfer rate ^{Note}	Max.6M[Hz] Min. $F_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] F_{CLK} : system clock frequency
Data phase	It can be selected by the DAPmn bit of the SCRmn register. DAPmn=0: Start data output when the serial clock starts running. DAPmn=1: Start data output half a clock before the serial clock starts running.
Clock phase	It can be selected by the CKPmn bit of the SCRmn register. CKPmn=0: Non-reverse CKPmn=1: Reverse
Data direction	MSB or LSB first

Caution: It must be used within the scope of the peripheral functional characteristics that meet this condition and meet the electrical characteristics (refer to the data sheet).

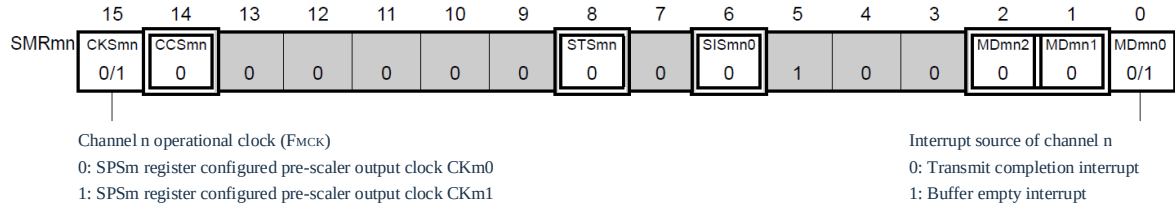
Remark: m: unit number (m=0);
n: channel number (n=0, 1);
p: SSPI number (p=00).

(1) Register setting

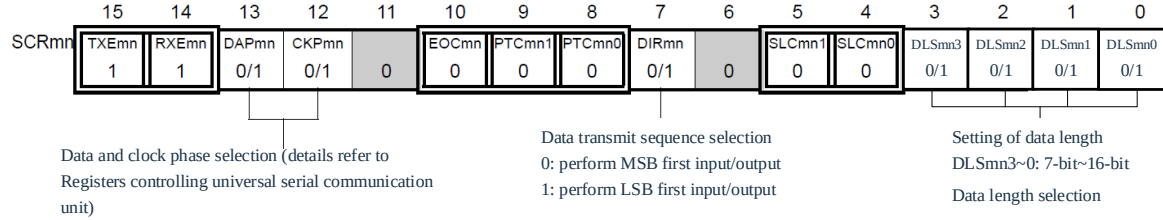
Figure 12-20: 3-wire serial I/O(SSPI00)

Example of register setting content when master transmits and receives

(a) Serial mode register mn (SMRmn)

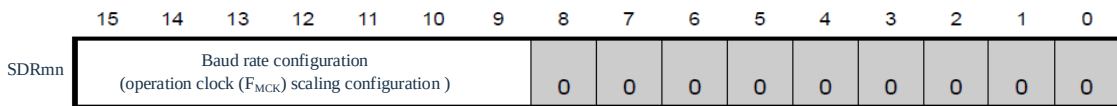


(b) Serial communication operation configuration register mn (SCRmn)

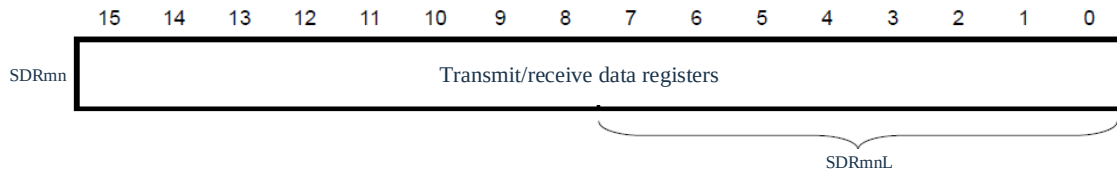


(c) Serial data register mn (SDRmn)

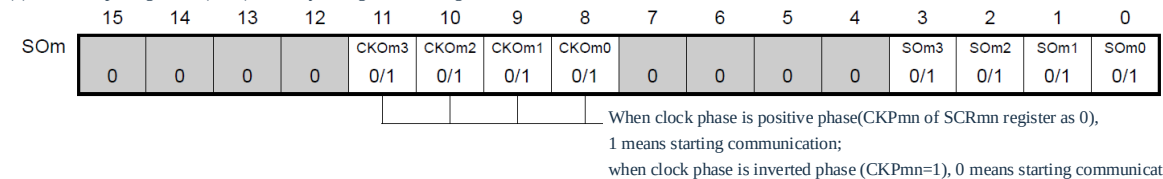
(1) When operation stops (SEmn=0)



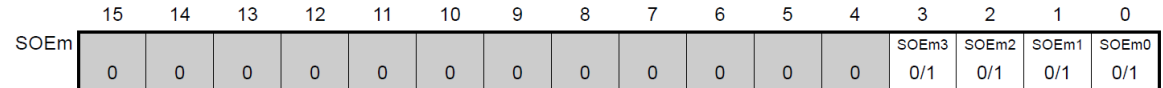
(2) During operation (SEmn=1) (lower 8 bits: SDRmnL)



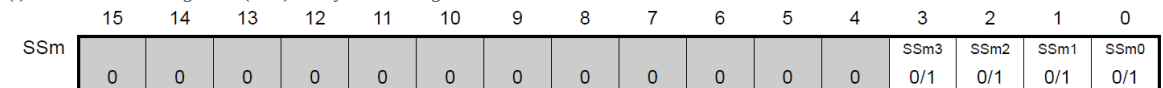
(d) Serial output register m(SOm).....Only configure bit of target channel



(e) Serial output enable register m (SOEm)....Only set bit of target channel to 1.



(f) Serial channel start register m (SSm)....only set bit of target channel to 1.



Remark: m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00);

□ : Fixed in SSPI master receive mode

■ : Cannot be set (initial value);

0/1: Set 0 or 1 according to the user's purpose.

(2)Procedure

Figure 12-21: Initial set-up steps for master transmission and reception



Figure 12-22: Stop steps for master transmission and reception

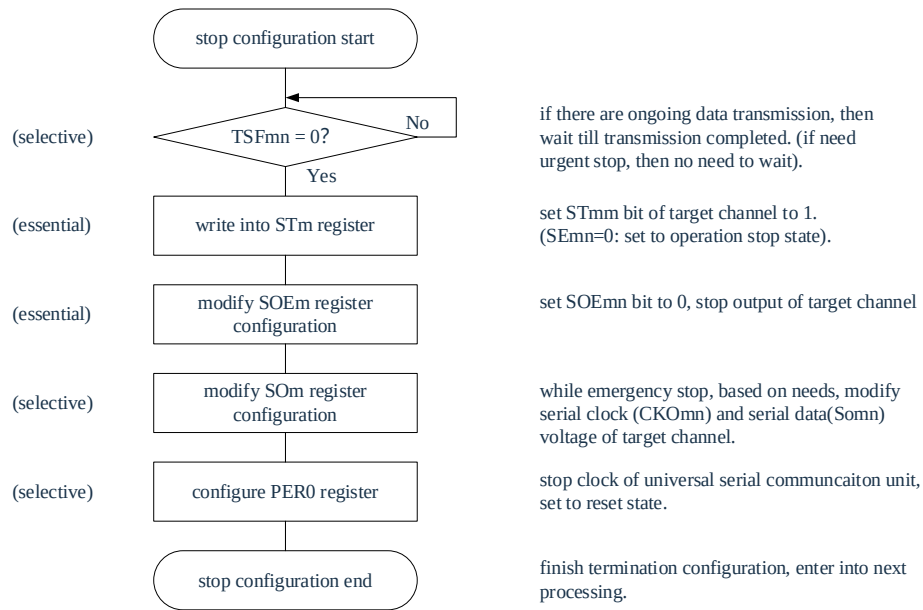
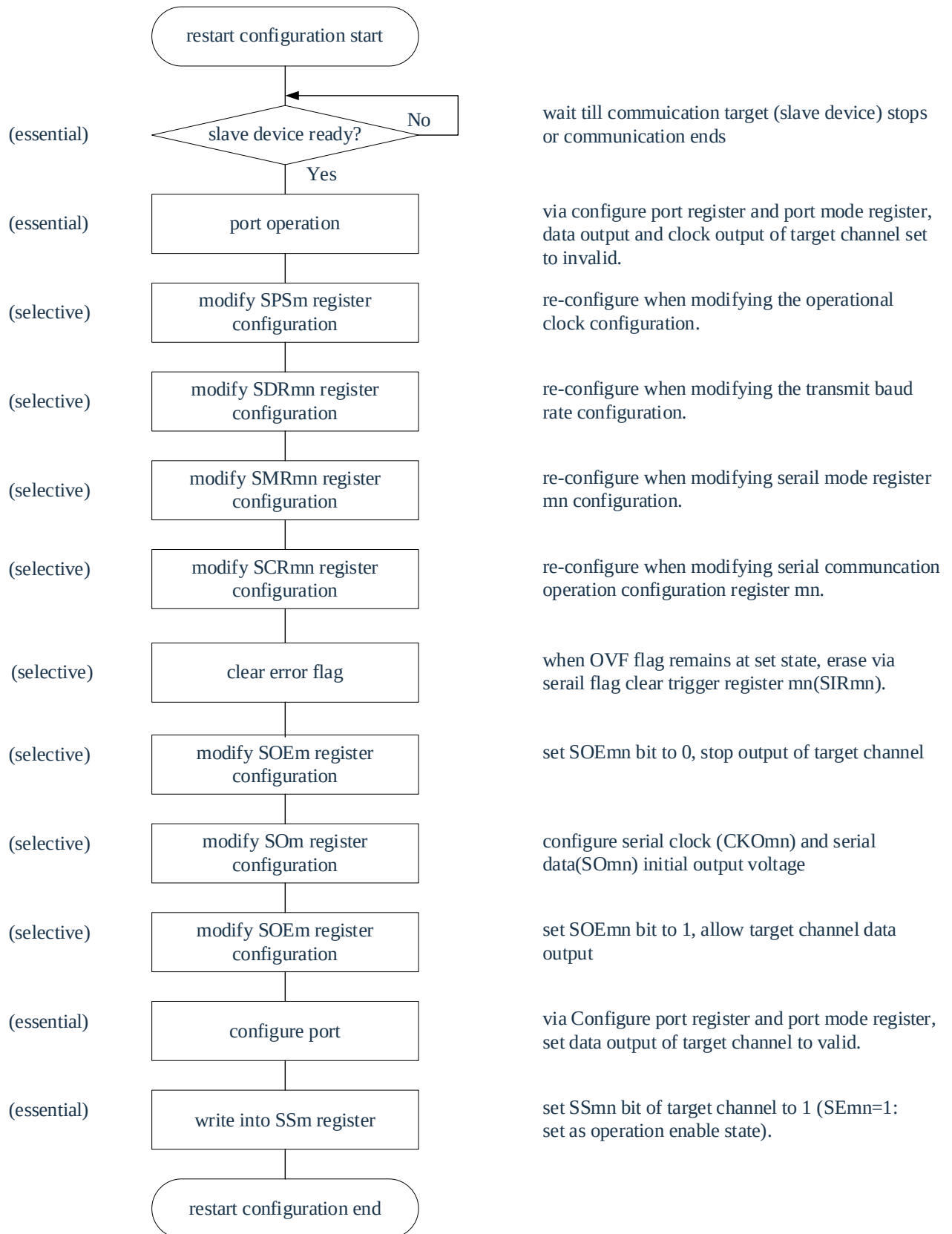
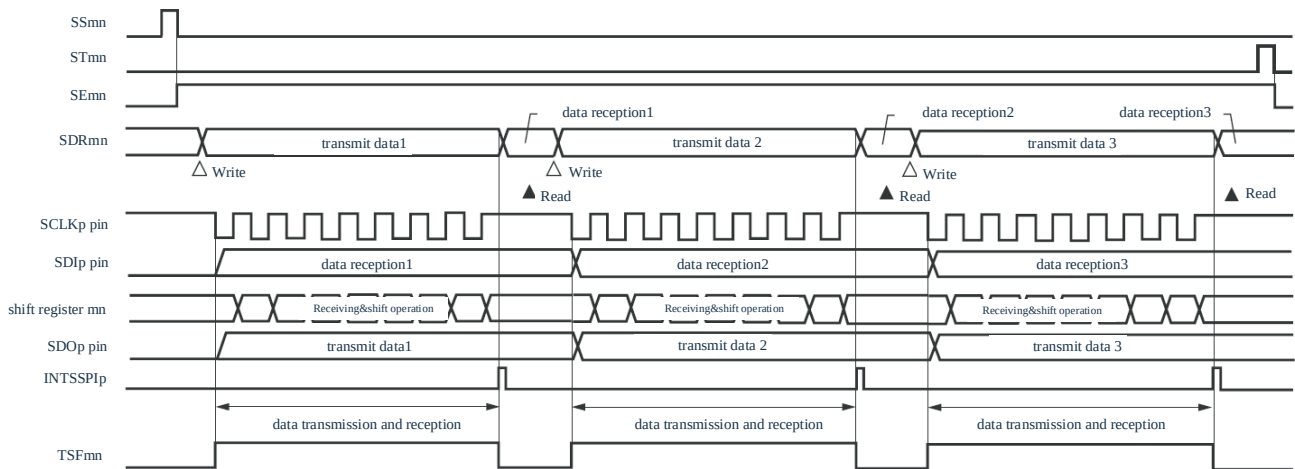


Figure 12-23: Restart steps for master transmission and reception



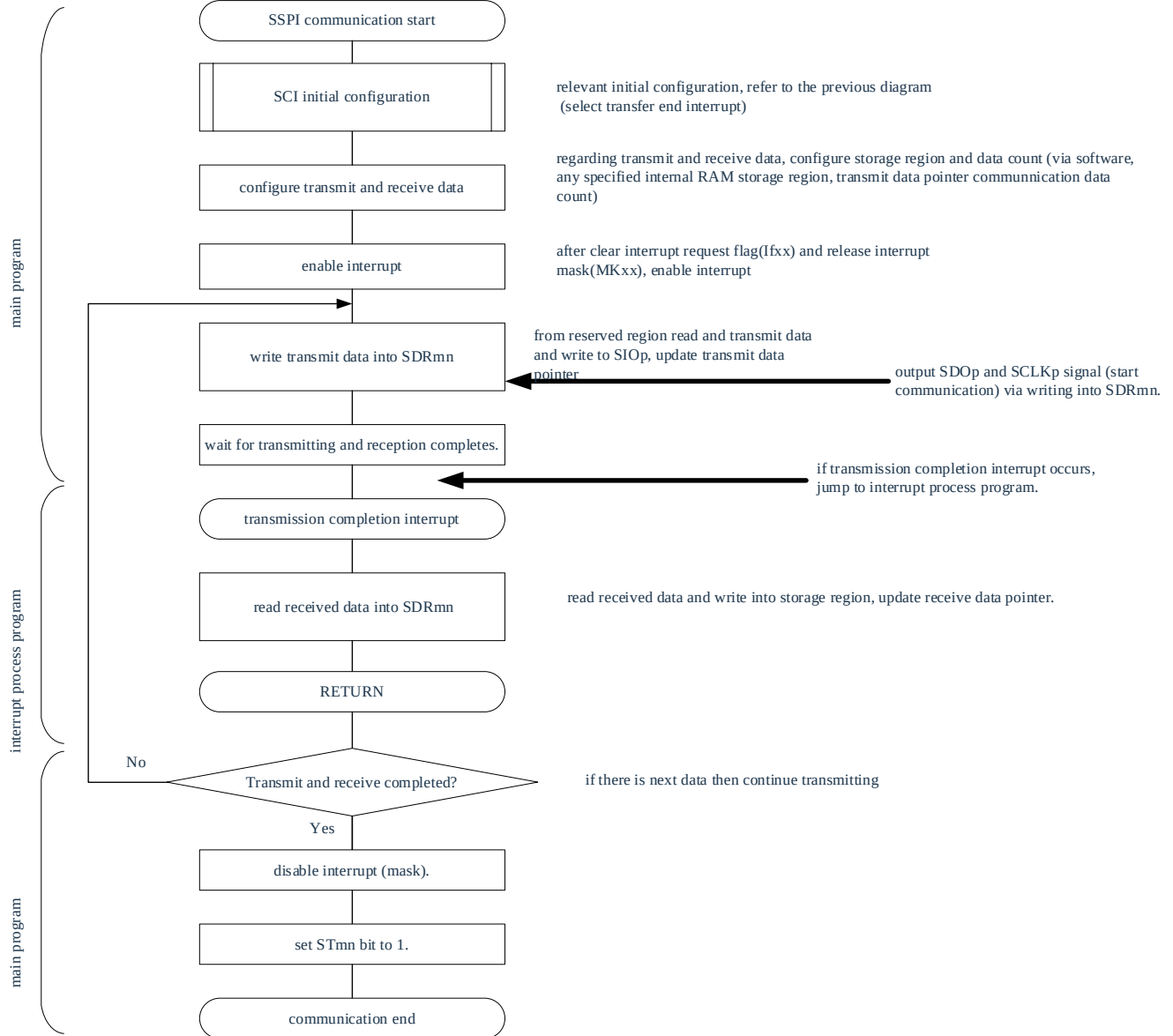
(3) Processing flow (single transmit and receive mode)

Figure 12-24: Timing diagram of master transmission and reception (single transmit and receive mode) (Type 1: DAPmn=0, CKPmn=0)



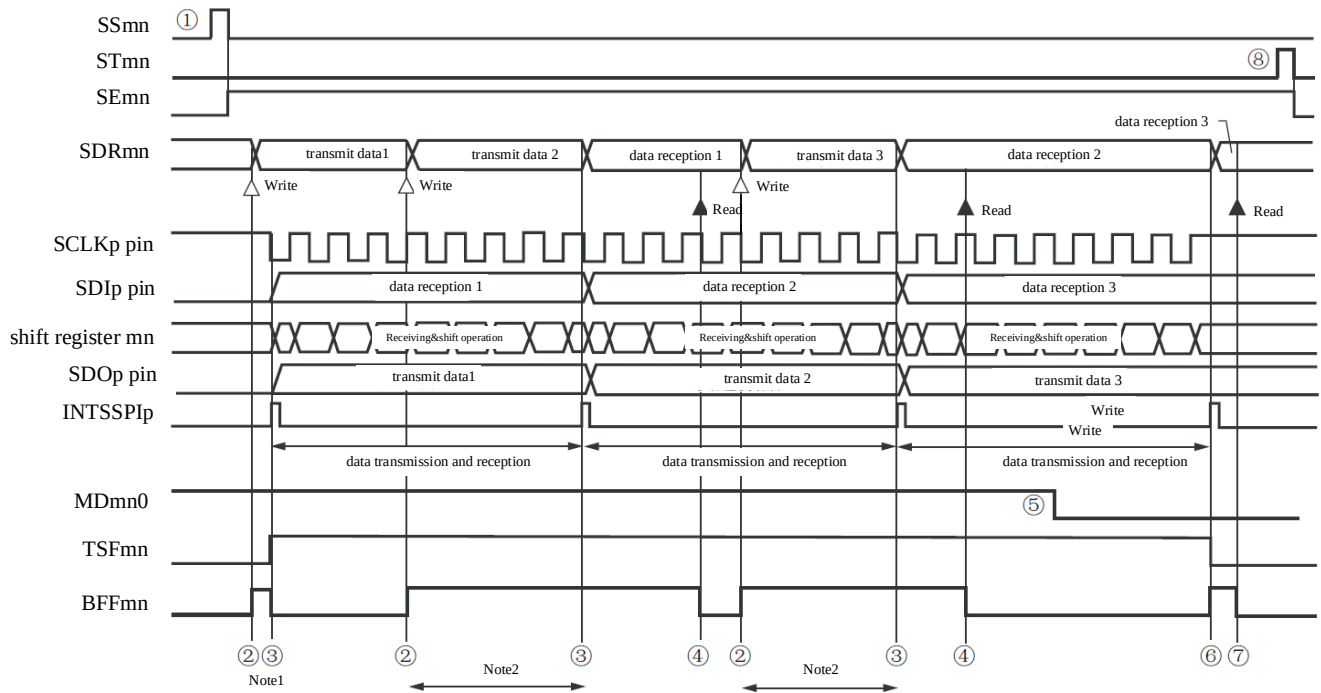
Remark: m: unit number (m=0); n: channel number (n=0, 1); p: SSPI number (p=00).

Figure 12-25: Flowchart of master transmission and reception (single transmit and receive mode)



(4) Processing flow (continuous transmit and receive mode)

Figure 12-26: Timing diagram of master transmission and reception (continuous transmit and receive mode)
(type 1: DAPmn=0, CKPmn=0)



Note 1: If the BFFmn bit of the serial status register mn (SSRmn) is "1" (valid data is saved in the serial data register mn (SDRmn)) to write the send data to the SDRmn register, and rewrite the sent data.

Note 2: If the SDRmn register is read during this period, the transmitted data can be read. At this point, the transfer operation is not affected.

Caution: The MDmn0 bit of the serial mode register mn (SMRmn) can be overridden even during operation. However, in order to catch up with the end of the transmission interruption of the last transmitted data, it must be overwritten before the last bit of transmission begins.

Remark:

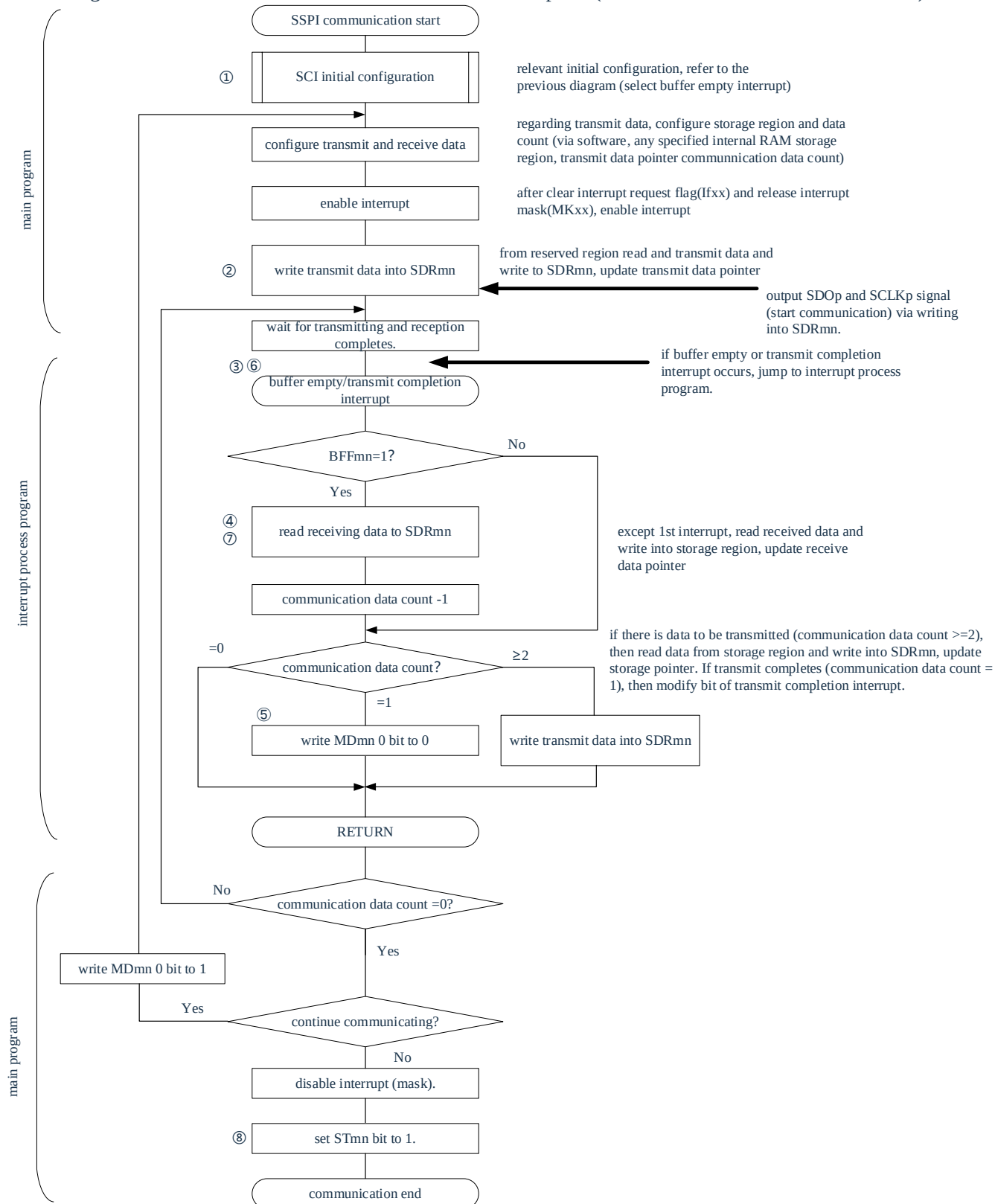
① to ⑧ in the figure correspond to ① to ⑧ in the "Flowchart of Figure 12-27".

m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00).

Figure 12-27: Flowchart of master transmission and reception (continuous transmit and receive mode)



Remark: ① to ⑧ correspond to ① to ⑧ in “Figure 12-26 Timing diagram of master transmission and reception (continuous transmit and receive mode)”.

12.5.4 Slave Transmission

Slave transmission is the operation of the microcontroller to transmit data to other devices in the state of inputting the transfer clock from other devices.

Table 12-25: Slave transmission

3-wire serial I/O	SSPI00
Target channel	Channel of SCIO
Pin used	SCLKOI00, SDO00
Interrupt	INTSSPI00
	Transfer end interrupt (in single transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Only the overrun error detection flag (OVFmn).
Transfer data length	7 or 8 bits
Transfer rate	Max.6M[Hz] ^{Note 1,2}
Data phase	It can be selected by the DAPmn bit of the SCRmn register. DAPmn=0: Start data output when the serial clock starts running. DAPmn=1: Start data output half a clock before the serial clock starts running.
Clock phase	It can be selected by the CKPmn bit of the SCRmn register. CKPmn=0: Non-reverse CKPmn=1: Reverse
Data direction	MSB or LSB first

Note 1: The maximum transfer rate is 6 [Hz] because the external serial clocks input from pin SCLKOI00 is sampled internally and then used.

Note 2: It must be used within the scope of the peripheral functional characteristics that meet this condition and meet the electrical characteristics (refer to the data sheet).

Remark:

F_{MCK} : Target channel operation clock frequency

m: unit number (m=0);

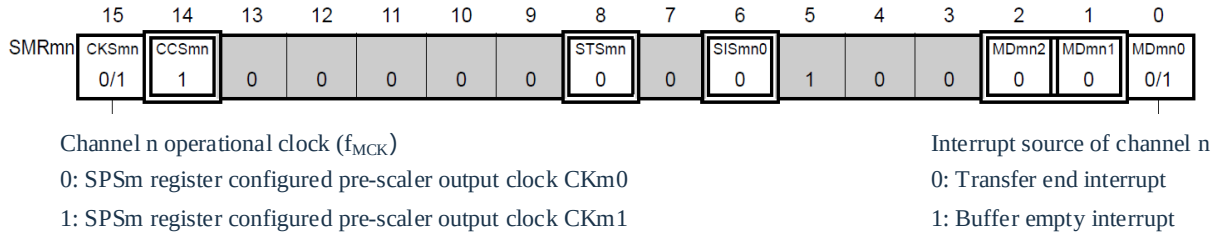
n: channel number (n=0, 1).

(1) Register setting

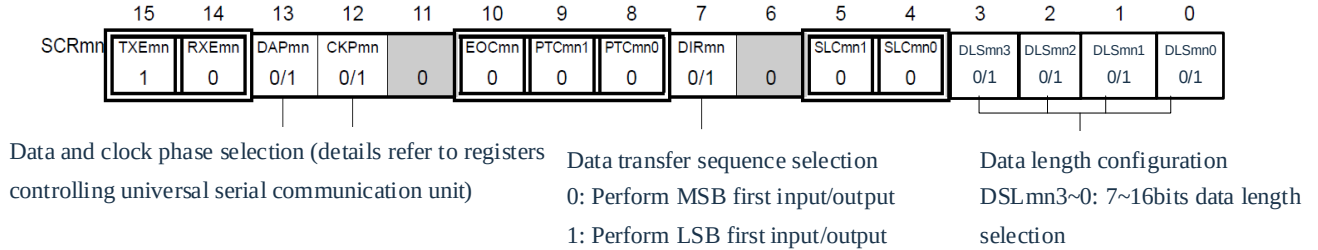
Figure 12-28: 3-wire serial I/O(SSPI00)

Example of register settings for slave transmission

(a) Serial mode register mn (SMRmn)

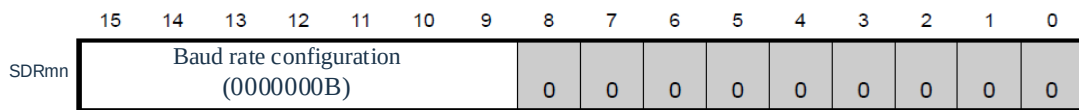


(b) Serial communication operation configuration register mn (SCRmn)

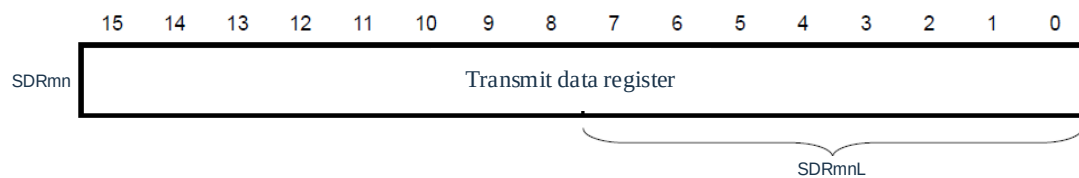


(c) Serial data register mn (SDRmn)

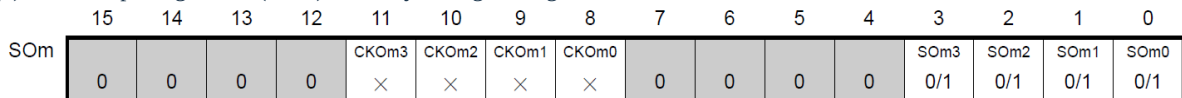
(1) Run stop (SEmn=0)



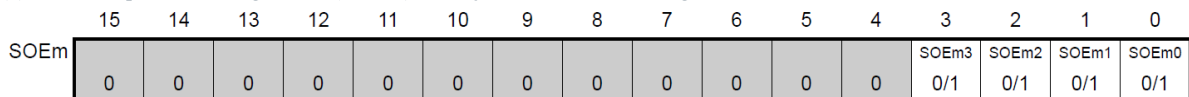
(2) During operation (SEmn=1) (low 8 bits: SDRmnL)



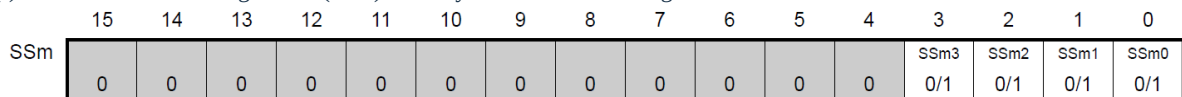
(d) Serial output register m (SOm) Only configure target channel bits



(e) Serial output enable register m (SOEm)....only set the bit of the target channel to 1.



(f) Serial channel start register m (SSm) only set the bit of the target channel to 1.



Remark: m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00);

☐ : Fixed in SSPI master receive mode

☐ : Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set the initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

(2)Procedure

Figure 12-29: Initial setup steps for slave transmission

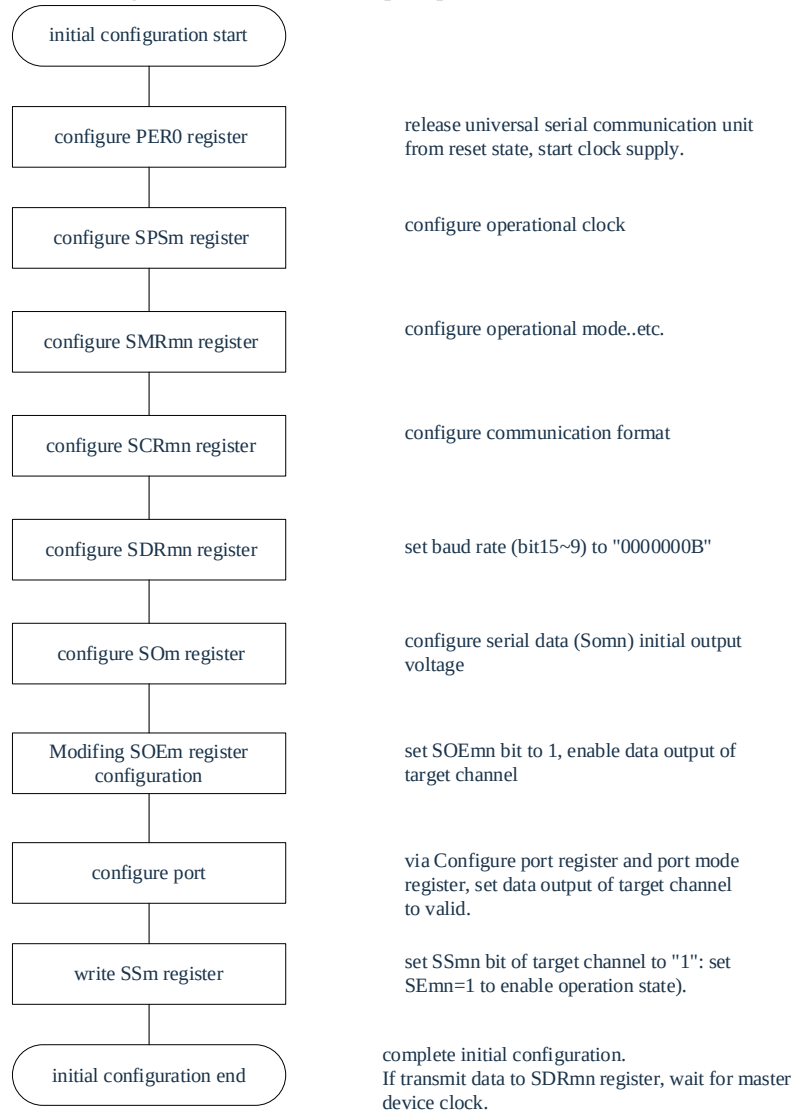


Figure 12-30: Stop steps for slave transmission

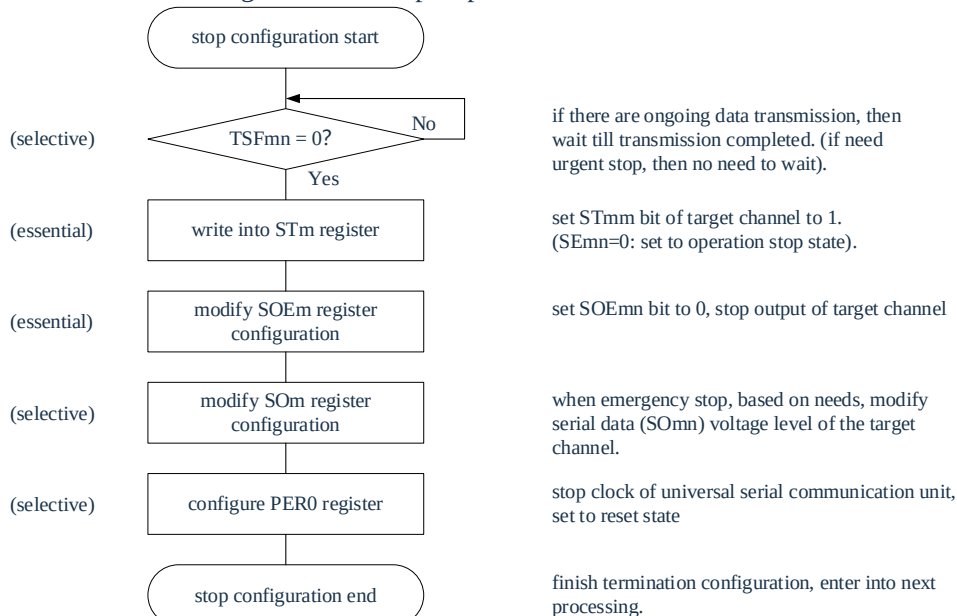
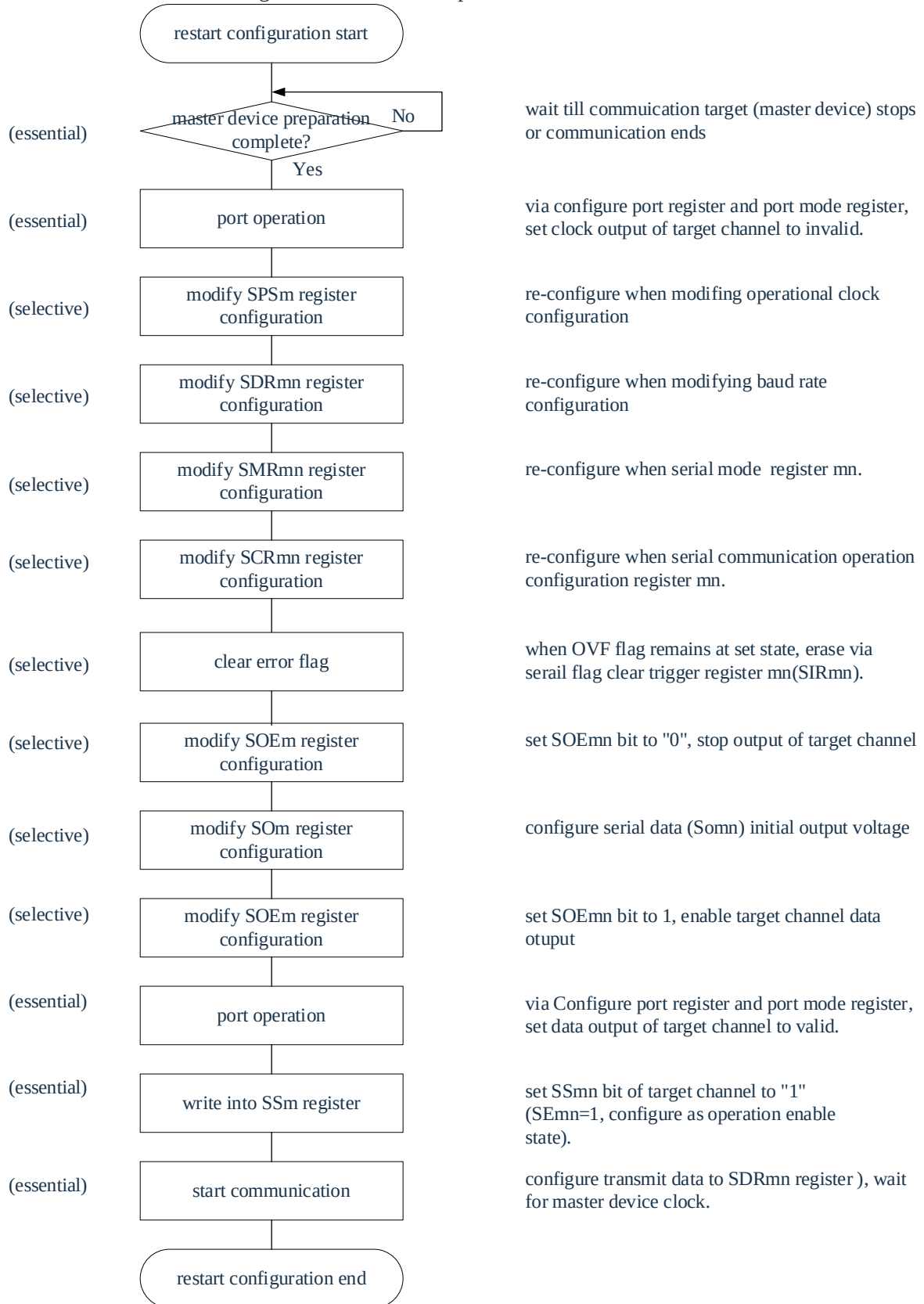


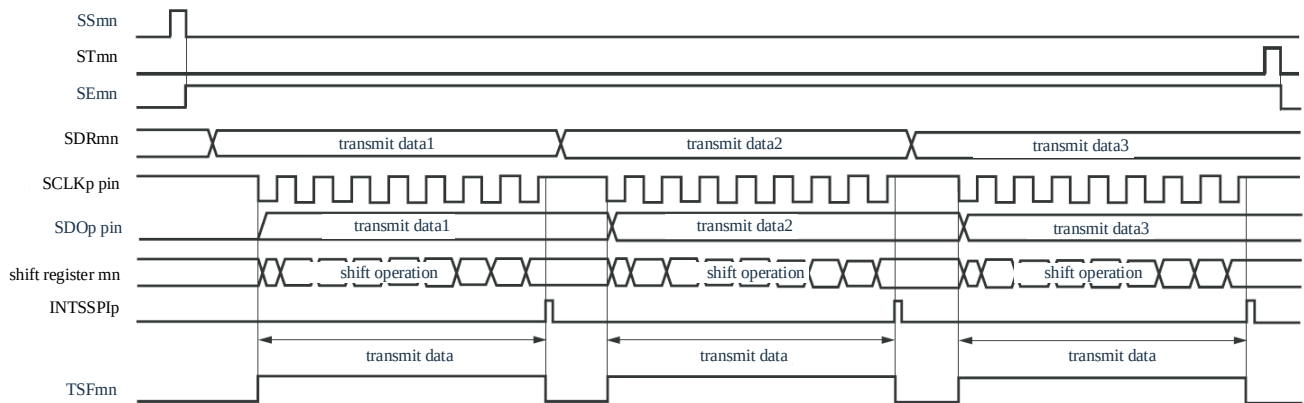
Figure 12-31: Restart steps for slave transmission



Caution: If you override PER0 in the abort setting to stop the clock, you must wait until the communication target (the master device) stops or the communication is over to make the initial setting instead of the restart setting.

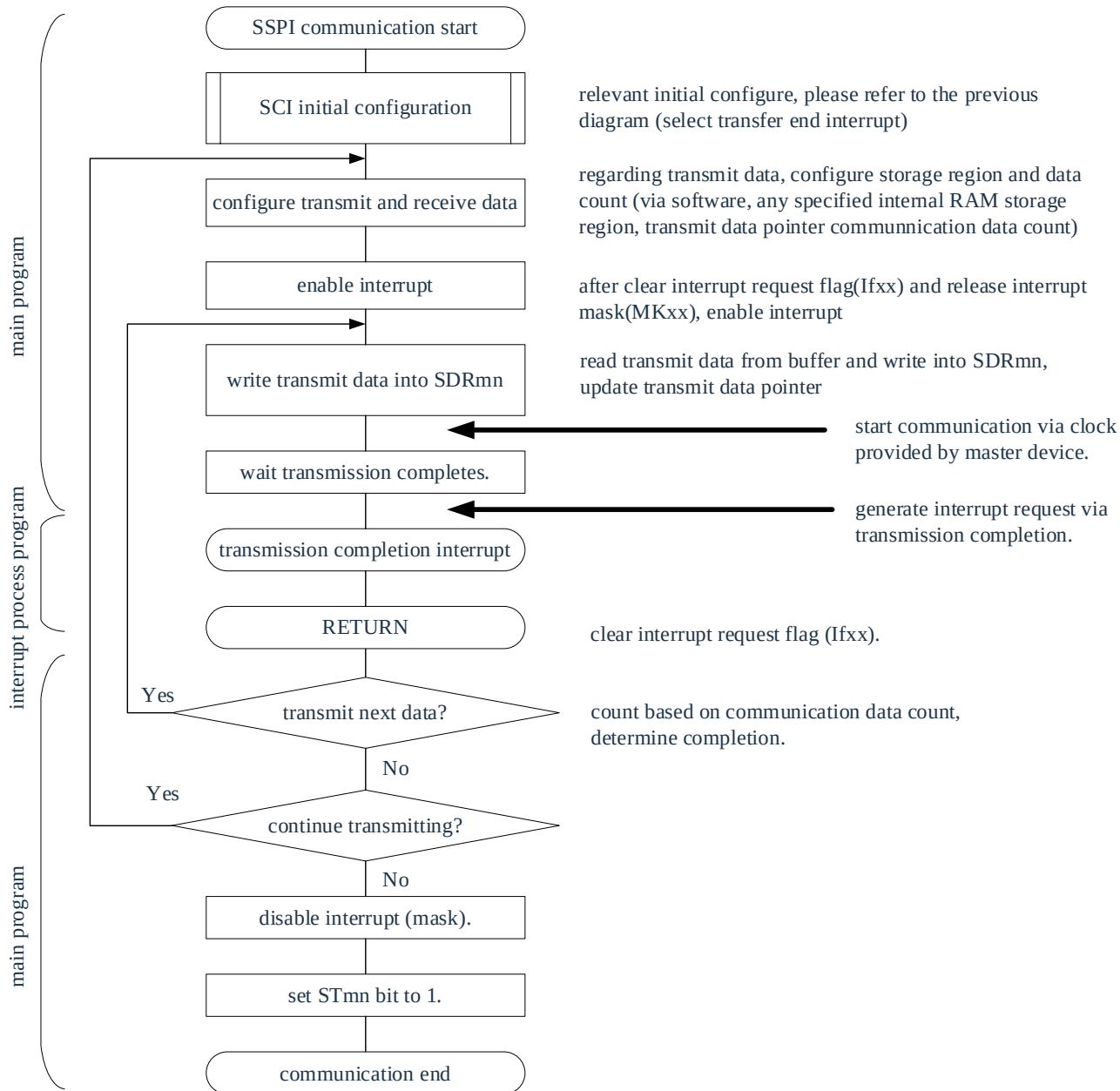
(3) Processing flow (single transmit mode)

Figure 12-32: Timing diagram of slave transmission (single transmit mode) (type 1: DAPmn=0, CKPmn=0)



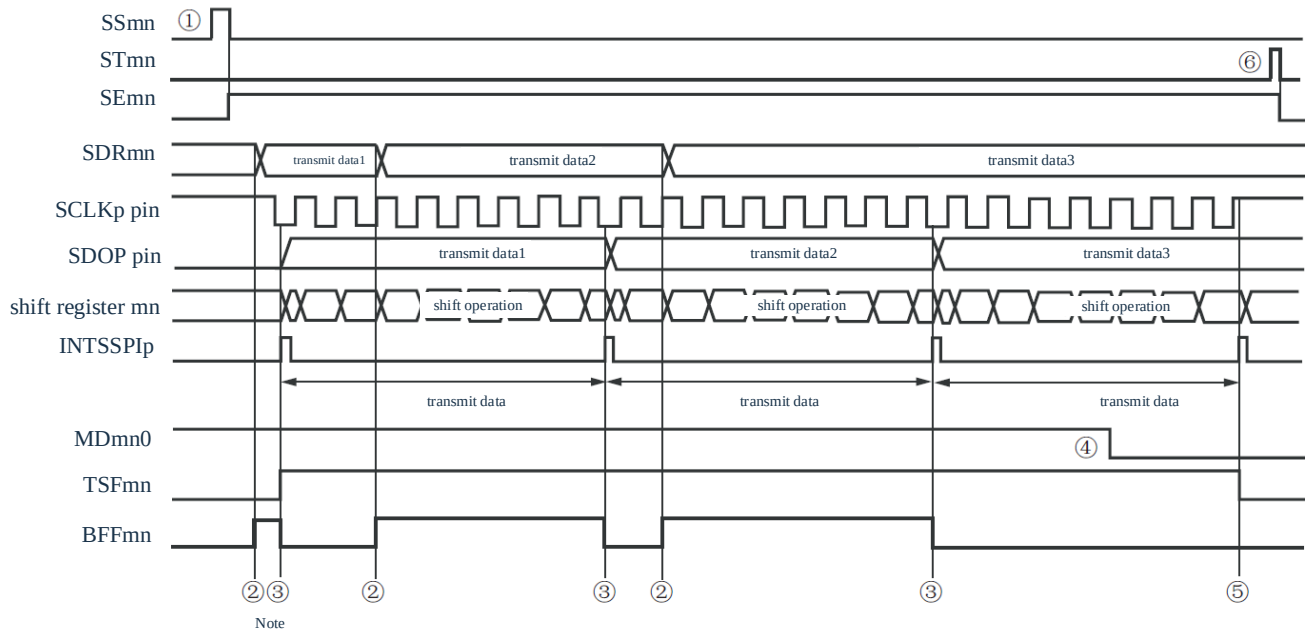
Remark: m: unit number (m=0);
n: channel number (n=0, 1);
p: SSPI number (p=00).

Figure 12-33: Flowchart of slave transmission (single transmit mode)



(4) Processing flow (continuous transmit mode)

Figure 12-34: Timing diagram of slave transmission (continuous transmit mode) (type 1: DAPmn=0, CKPmn=0)



Note: If the BFFmn bit of the serial status register mn (SSRmn) is “1” (when valid data is saved in the serial data register mn (SDRmn)) is given The SDRmn register writes the transmitted data and overrides the transmitted data.

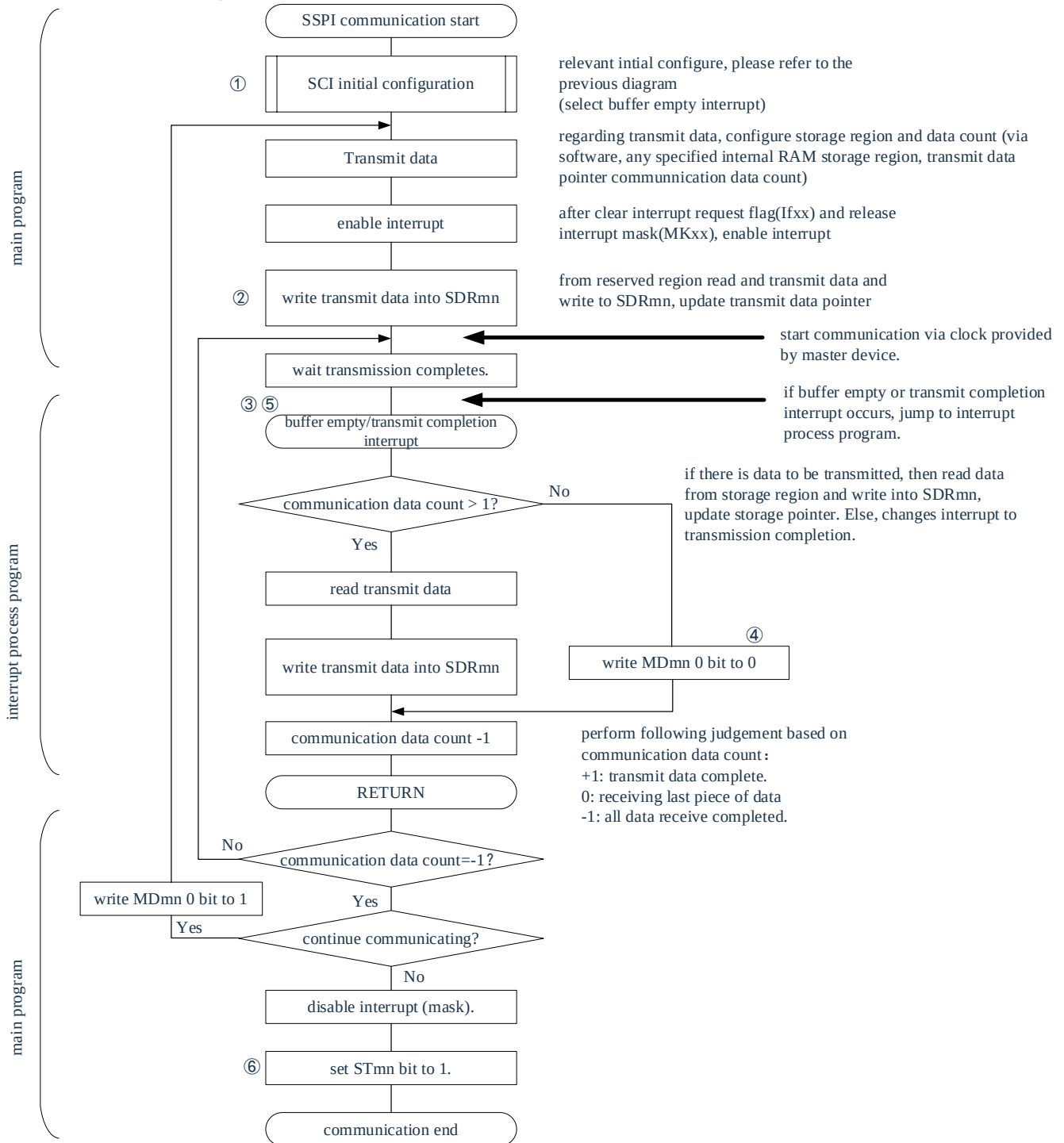
Caution: The MDmn0 bit of the serial mode register mn (SMRmn) can be overridden even during operation. However, it must be overridden before the last bit can be transferred.

Remark: m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00).

Figure 12-35: Flowchart of slave transmission (continuous transmit mode)



Remark: ① to ⑥ in the diagram correspond to ① to ⑥ in “Figure 12-34 Timing diagram of slave transmission (continuous transmit mode)”.

12.5.5 Slave Reception

Slave reception refers to the operation of this product to receive data from other devices in the state of transfer clocks from other devices.

Table 12-26: Slave reception

3-wire serial I/O	SSPI00
Target channel	Channel of SCIO
Pin used	SCLKOI00, SDI00
Interrupt	INTSSPI00 Limited to transfer end interrupts (disable setting buffer empty interrupts).
Error detection flag	Only the overrun error detection flag (OVFmn).
Transfer data length	7 or 8 bits
Transfer rate	Max.6MHz] ^{Note 1,2}
Data phase	It can be selected by the DAPmn bit of the SCRmn register. DAPmn=0: Start data output when the serial clock starts running. DAPmn=1: Start data output half a clock before the serial clock starts running.
Clock phase	It can be selected by the CKPmn bit of the SCRmn register. CKPmn=0: Non-reverse CKPmn=1: Reverse
Data direction	MSB or LSB first

Note 1: The maximum transfer rate is 6 [Hz] because the external serial clocks input from pin SCLKOI00 is sampled internally and then used.

Note 2: It must be used within the scope of the peripheral functional characteristics that meet this condition and meet the electrical characteristics (refer to the data sheet).

FMCK: Target channel operation clock frequency

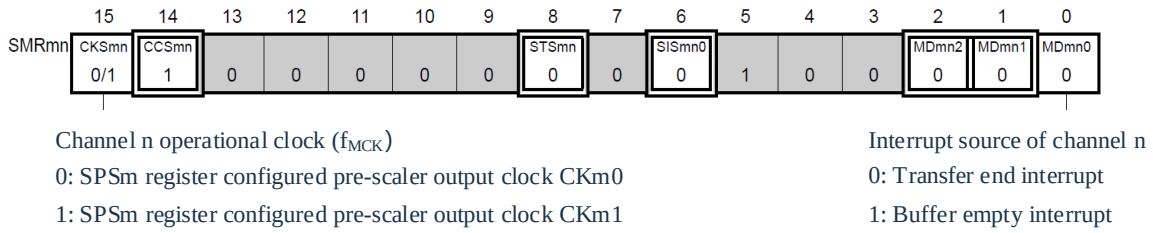
m: unit number (m=0);

n: channel number (n=0, 1).

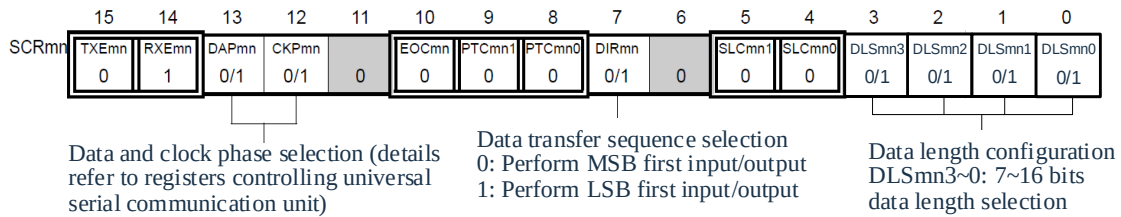
(1) Register setting

Figure 12-36: 3-wire serial I/O(SSPI00)
Example of register settings for slave reception

(a) Serial mode register mn (SMRmn)

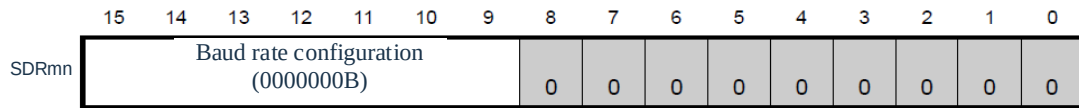


(b) Serial communication operation configuration register mn (SCRmn)

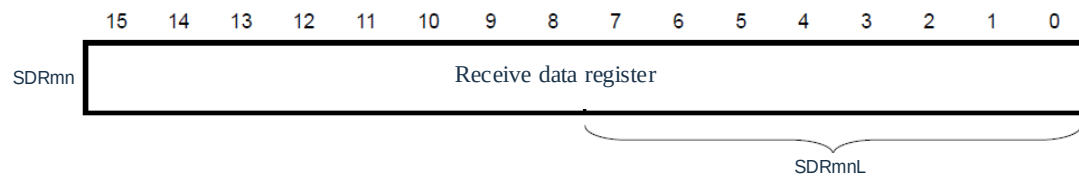


(c) Serial data register mn (SDRmn)

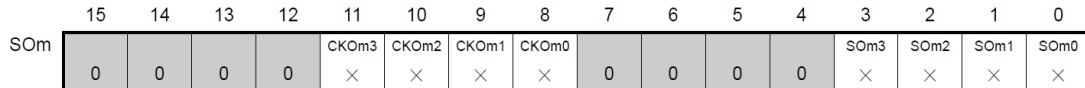
(1) Run stop (SEmn=0)



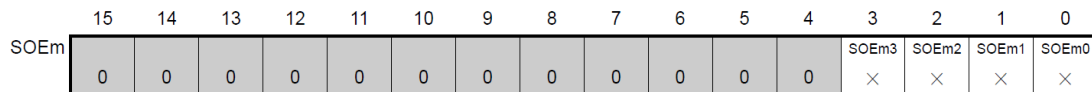
(2) During operation (SEmn=1) (Low 8 bits: SDRmnL)



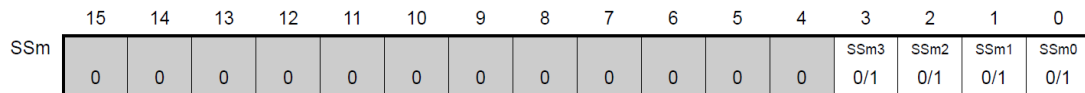
(d) Serial output register m (Som) Not used in this mode.



(e) Serial output enable register m (SOEm).... Not used in this mode.



(f) Serial channel start register m (SSm) Only set the bit of the target channel to 1.



Remark: m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00);

 : Fixed in SSPI master receive mode : Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set the initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

(2) Procedure

Figure 12-37: Initial setup steps for slave reception

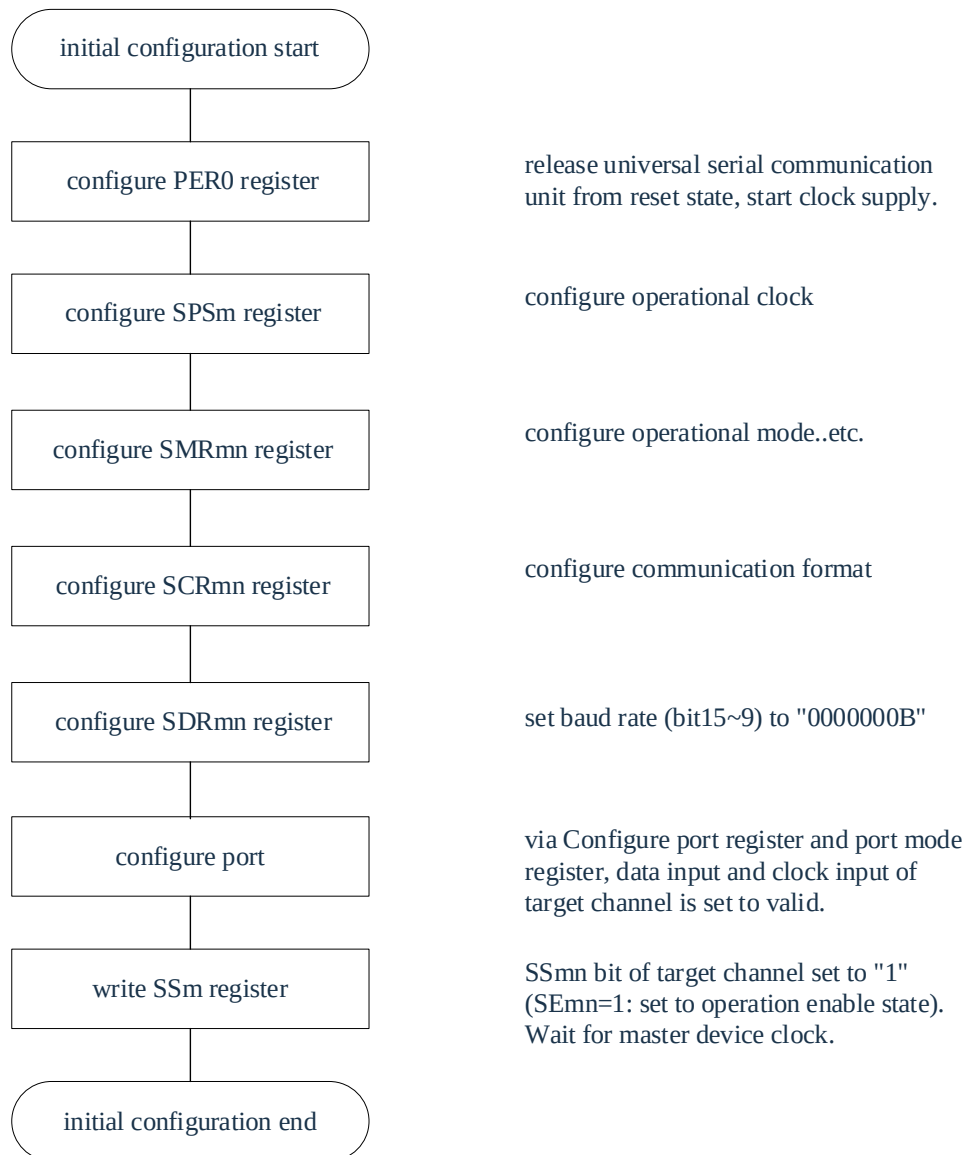


Figure 12-38: Stop steps for slave reception

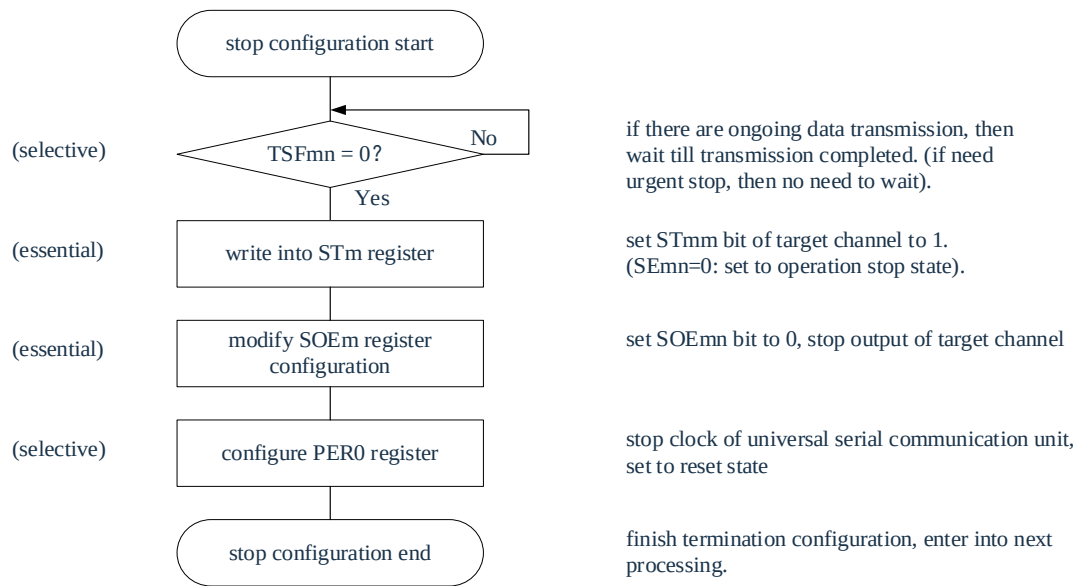
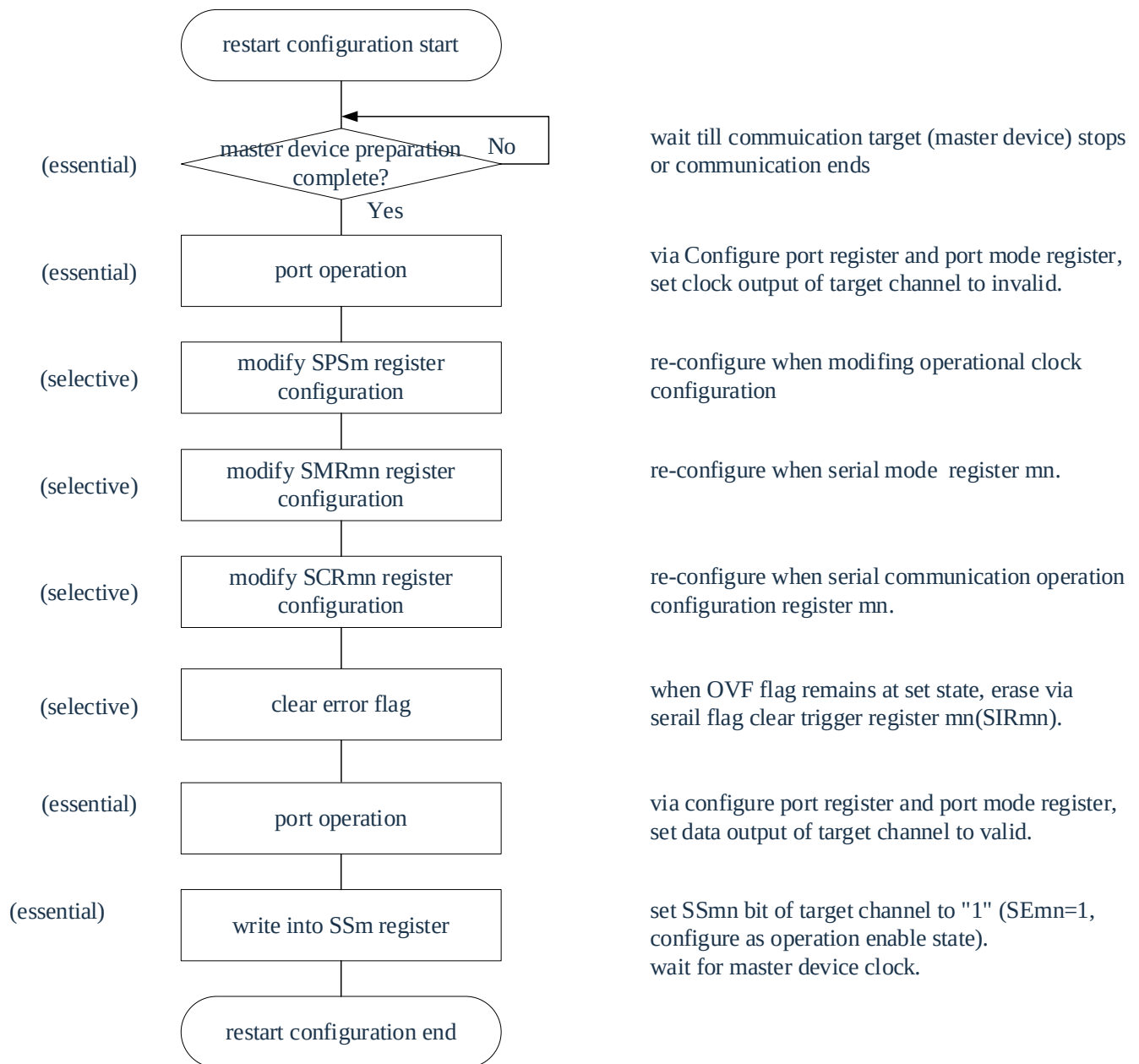


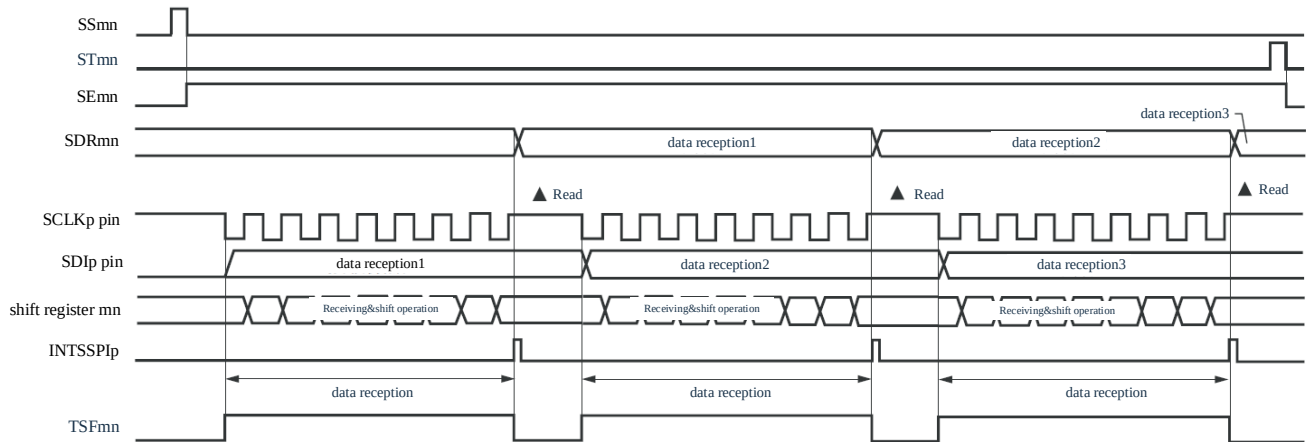
Figure 12-39: Restart steps for slave reception



Caution: If you override PER0 in the abort setting to stop the clock, you must wait until the communication target (the master device) stops or the communication is over to make the initial setting instead of the restart setting.

(3) Processing flow (single receive mode)

Figure 12-40: Timing diagram of slave reception (single receive mode) (type 1: DAPmn=0, CKPmn=0)

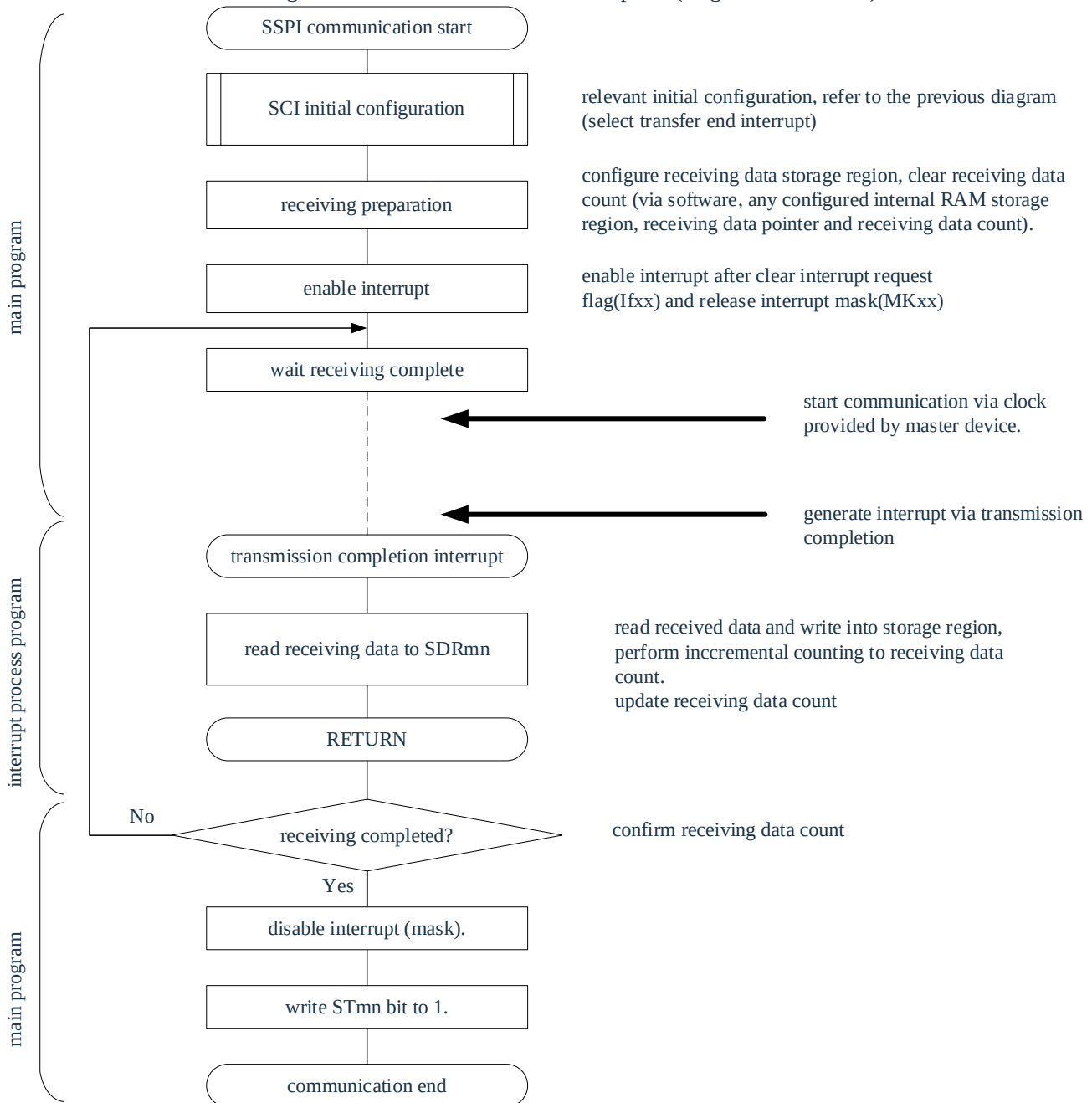


Remark: m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00).

Figure 12-41: Flowchart of slave reception (single receive mode)



12.5.6 Slave Transmission and Reception

Slave transmission and reception refers to the operation where, under the condition of receiving the transfer clock from other devices, the microcontroller of this product performs data transmission and reception with other devices.

Table 12-27: Slave transmission and reception

3-wire serial I/O	SSPI00
Target channel	Channel of SCI0
Pin used	SCLKOI00, SDI00, SDO00
Interrupt	INTSSPI00 Transfer end interrupt (in single transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Only the overrun error detection flag (OVFmn).
Transfer data length	7 or 8 bits
Transfer rate	Max.6M[Hz] ^{Note 1,2}
Data phase	It can be selected by the DAPmn bit of the SCRmn register. DAPmn=0: The data input/output starts when the serial clock starts running. DAPmn=1: Start data input/output half a clock before the serial clock starts running.
Clock phase	It can be selected by the CKPmn bit of the SCRmn register. CKPmn=0: Non-reverse CKPmn=1: Reverse
Data direction	MSB or LSB first

Note 1: The maximum transfer rate 6 [Hz] is used because the external serial clock input from SCLKOI00 pin is sampled internally and then used.

Note 2: It must be used within the scope of the peripheral functional characteristics that meet this condition and meet the electrical characteristics (refer to the data sheet).

Remark:

F_{MCK} : Target channel operation clock frequency

m: unit number (m=0);

n: channel number (n=0, 1).

(1) Register setting

Figure 12-42: 3-wire serial I/O(SSPI00)

Example of register settings for slave transmission and reception

(a) Serial mode register mn (SMRmn)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKSmn	CCSmn						STSmn		SIsmn0				MDmn2	MDmn1	MDmn0
0/1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Channel n operational clock (f_{MCK})

0: SPSm register configured pre-scaler output clock CKm0

1: SPSm register configured pre-scaler output clock CKm1

Interrupt source of channel n

0: Transfer end interrupt

1: Buffer empty interrupt

(b) Serial communication operation configuration register mn (SCRmn)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0	DSLmn3	DSLmn2	DSLmn1	DSLmn0
1	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0/1	0/1	0/1	0/1

Data and clock phase selection (details refer to registers controlling universal serial communication unit)

Data transfer sequence selection
0: Perform MSB first input/output
1: Perform LSB first input/output

Data length configuration

DSLmn3~0: 7-16 bits data length selection

(c) Serial data register mn (SDRmn)

(1) Run stop (SEmn=0)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Baud rate configuration (0000000B)								0	0	0	0	0	0	0	0

(2) During operation (SEmn=1) (low 8 bits: SDRmnL)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Transmit/receive data register															

SDRmnL

(d) Serial output register m(SOm) Only configure target channel bits

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				CKOm3	CKOm2	CKOm1	CKOm0					SOm3	SOm2	SOm1	SOm0
0	0	0	0	×	×	×	×	0	0	0	0	0/1	0/1	0/1	0/1

(e) Serial output enable register mn (SOEm)....only set the bit of the target channel to 1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												SOEm3	SOEm2	SOEm1	SOEm0
0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

(f) Serial channel start register m (SSm) only set the bit of the target channel to 1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												SSm3	SSm2	SSm1	SSm0
0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

Caution: Before the master device starts outputting the clock, the transmit data must be set in the SDRmn register.

Remark: m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00);



: Fixed in SSPI master receive mode



: Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set the initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

(2)Procedure

Figure 12-43: Initial setup steps for slave transmission and reception



Caution: Before the master device starts outputting the clock, the transmit data must be set in the SDRmn register.

Figure 12-44: Stop steps for slave transmission and reception

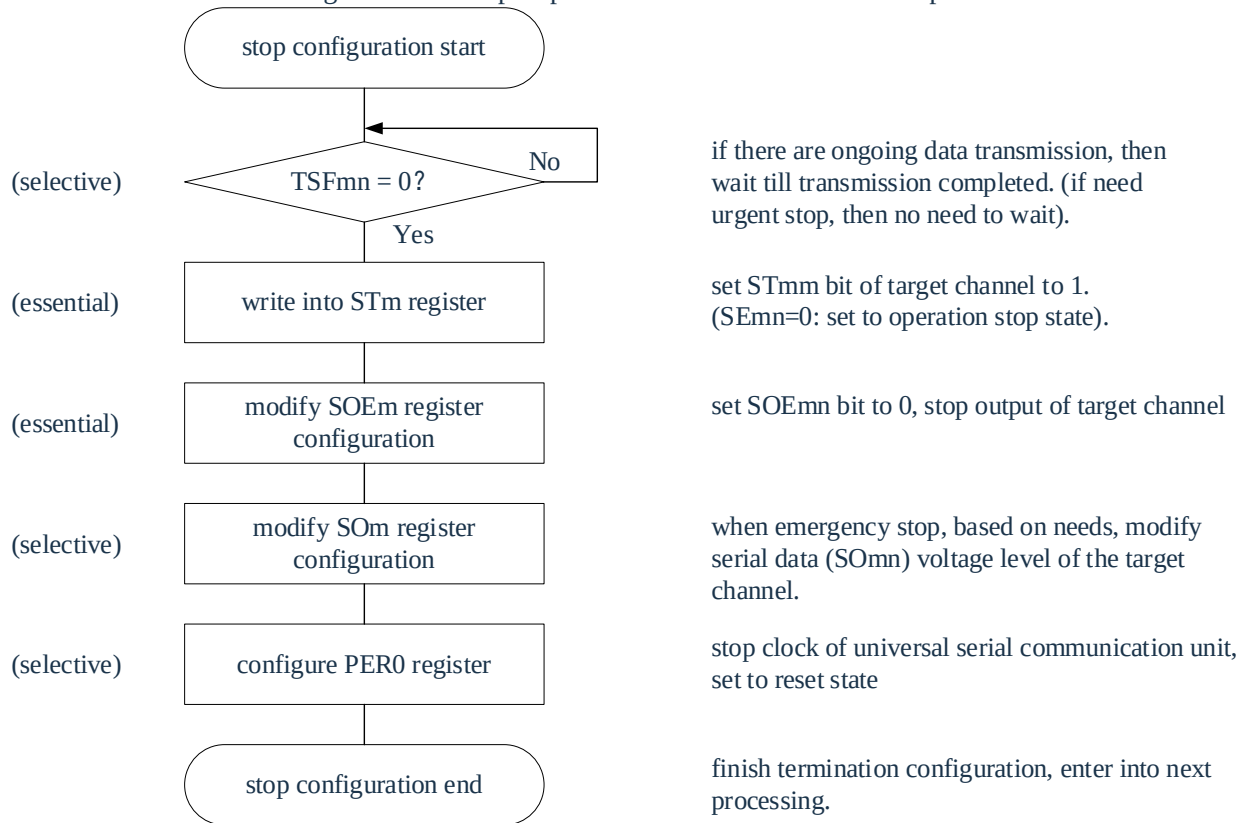
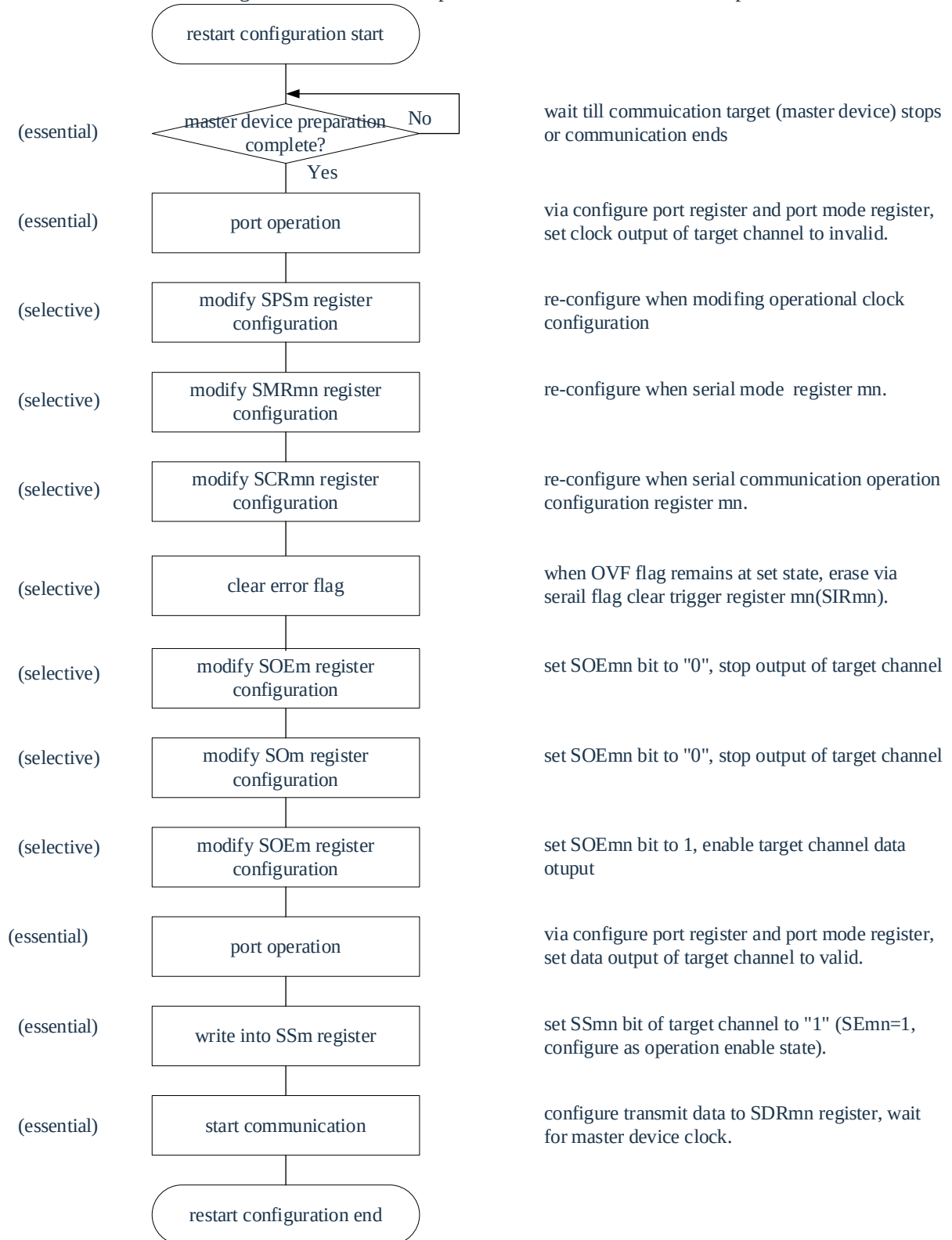


Figure 12-45: Restart steps for slave transmission and reception

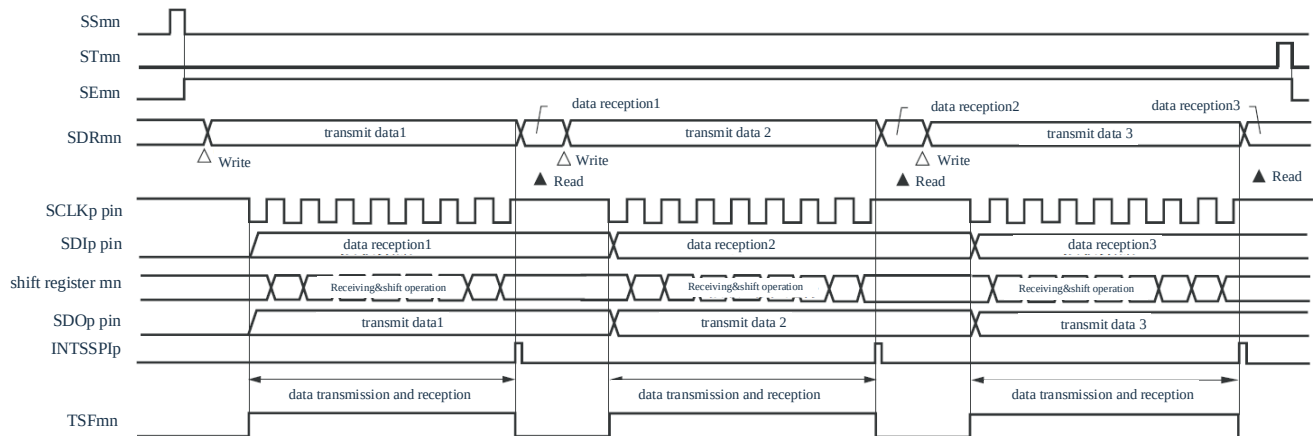


Caution:

Before the master device starts outputting the clock, the transmit data must be set in the SDRmn register. If you override PER0 in the abort setting to stop the clock, you must wait until the communication target (the master device) stops or the communication is over to make the initial setting instead of the restart setting.

(3) Processing flow (single transmit and receive mode)

Figure 12-46: Timing diagram of slave transmission and reception (single transmit and receive mode)
(type 1: DAPmn=0, CKPmn=0)

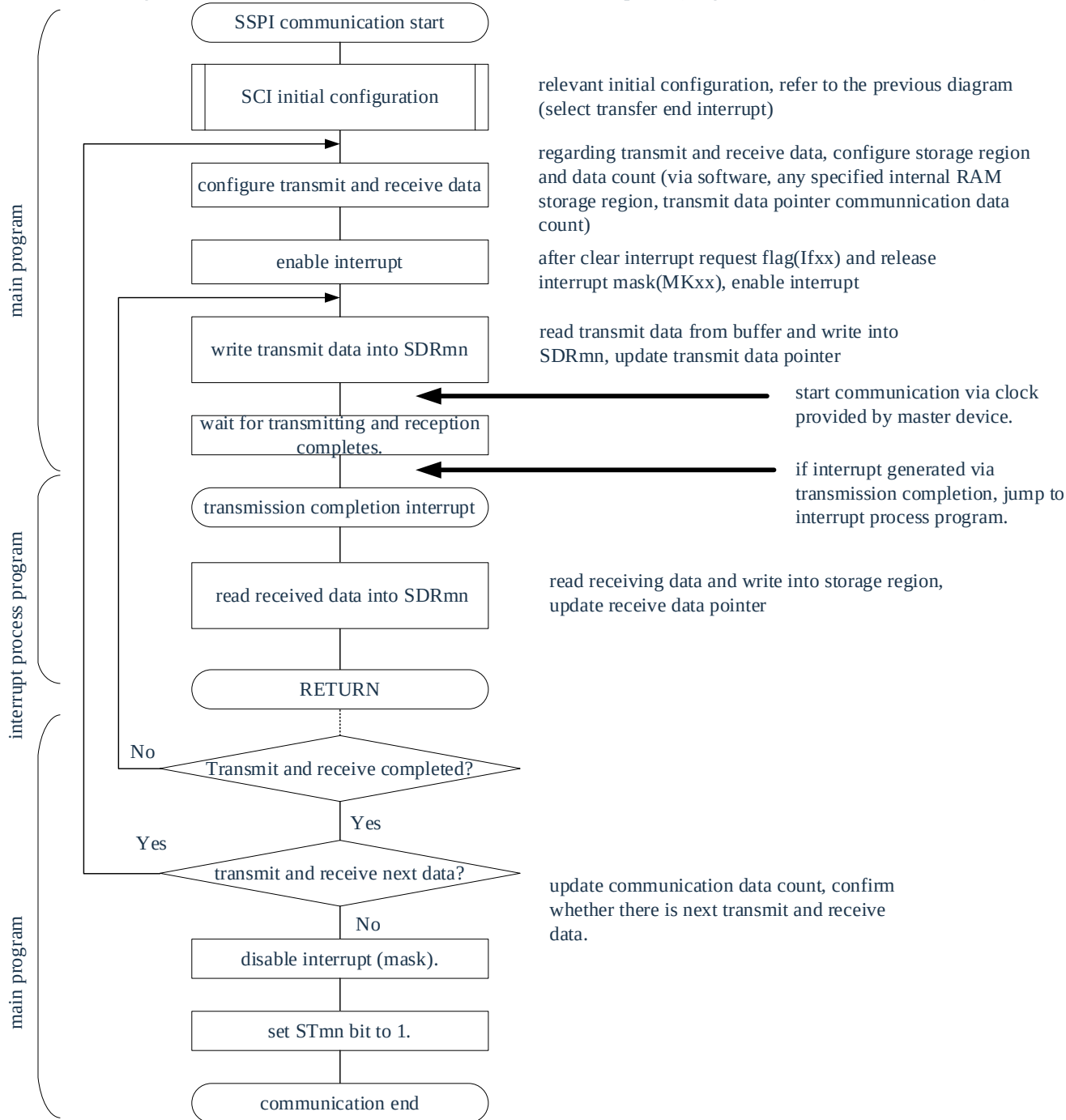


Remark: m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00).

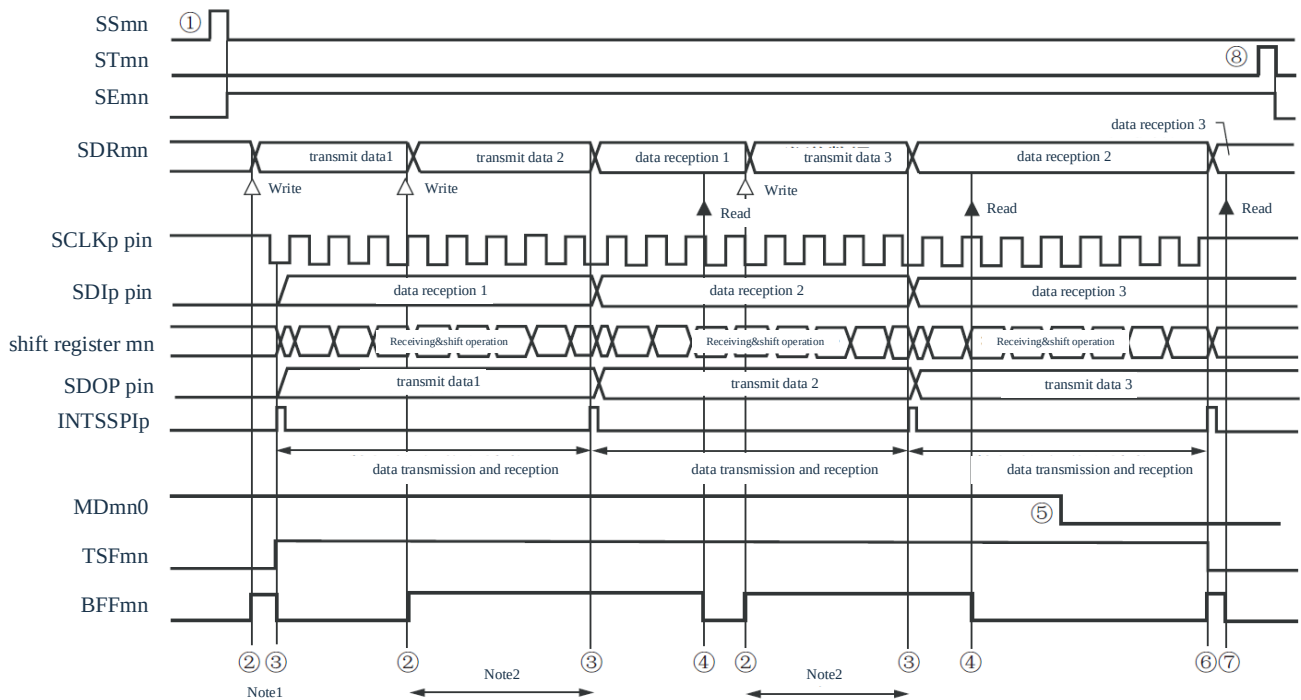
Figure 12-47: Flowchart of slave transmission and reception (single transmit and receive mode)



Caution: Before the master device starts outputting the clock, the transmit data must be set in the SIOp register.

(4) Processing flow (continuous send and receive mode)

Figure 12-48: Timing diagram of slave transmission and reception (continuous transmit and receive mode) (type 1: DAPmn=0, CKPmn=0)



Note 1: If the BFFmn bit of the serial status register mn (SSRmn) is “1” (valid data is saved in the serial data register mn (SDRmn)) to write the send data to the SDRmn memory, and rewrite the transmitted data.

Note 2: If the SDRmn register is read during this period, the transmitted data can be read. At this point, the transfer operation is not affected.

Caution: Even during operation, the MDmn0 bit of the serial mode register mn (SMRmn) can be rewritten. However, to ensure that the transmission end interrupt for the last transmitted data is triggered, the rewriting must be done before the last bit is transmitted.

Remark:

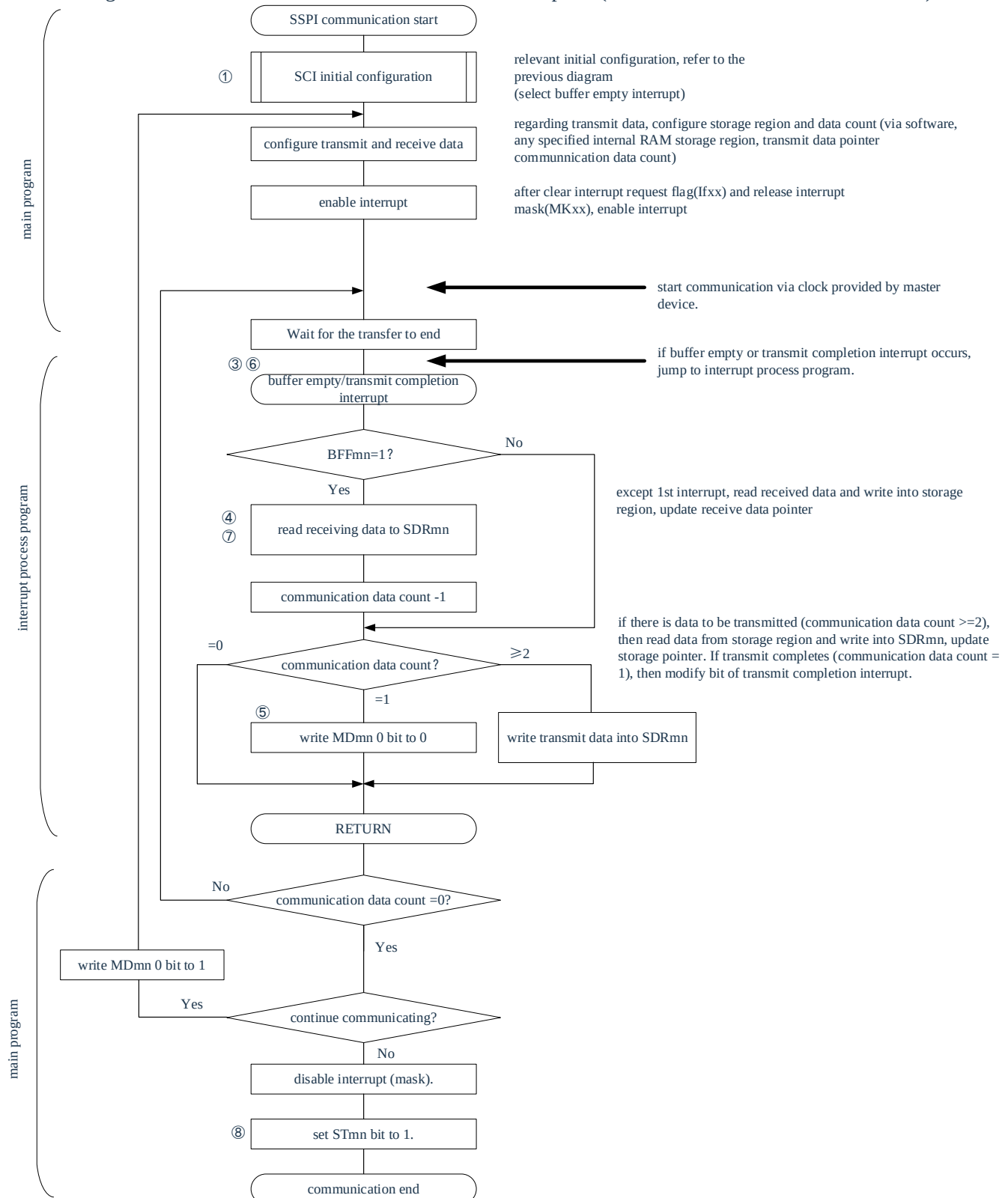
① to ⑧ in the diagram correspond to ① to ⑧ in “Figure 12-49 Flowchart of slave transmission and reception (continuous transmit and receive mode)”.

m: unit number (m=0);

n: channel number (n=0, 1);

p: SSPI number (p=00).

Figure 12-49: Flowchart of slave transmission and reception (continuous transmit and receive mode)



Caution: Before the master device starts outputting the clock, the transmit data must be set in the SDRmn register.

Remark: ① to ⑧ correspond to ① to ⑧ in “Figure 12-48 Timing diagram of slave transmission and reception (continuous transmit and receive mode)”.

12.5.7 Calculation of Transfer Clock Frequency

3-wire serial I/O(SSPI00) communication transfer clock frequency can be calculated using the following calculation equations.

Master device

$$(\text{Transfer clock frequency}) = \{ \text{Operation clock frequency of the target channel}(f_{\text{MCK}}) \} / (\text{SDRmn}[15:9] + 1) / 2 [\text{Hz}]$$

Slave device

$$(\text{Transfer clock frequency}) = \{ \text{Serial clock (SCLK) frequency provided by the master device} \}^{\text{Note}} [\text{Hz}]$$

Remark:

The maximum allowable transfer clock frequency is 6MHz;

Since the value of SDRmn[15:9] is the value of bits 15-9 of the serial data register mn (SDRmn) (0000000B to 1111111B), it ranges from 0 to 127;

The operation clock (F_{MCK}) depends on the serial clock selection register m (SPSm) and bit 15 (CKSmn) of the serial mode register mn (SMRmn).

Table 12-28: Selection of 3-wire serial I/O operation clock

SMRmn register	SPSm register								Operation clock (F _{MCK}) ^{Note}	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		F _{CLK} =24MHz in operation
0	X	X	X	X	0	0	0	0	F _{CLK}	24MHz
	X	X	X	X	0	0	0	1	F _{CLK} /2	12MHz
	X	X	X	X	0	0	1	0	F _{CLK} /2 ²	6MHz
	X	X	X	X	0	0	1	1	F _{CLK} /2 ³	3MHz
	X	X	X	X	0	1	0	0	F _{CLK} /2 ⁴	1.5MHz
	X	X	X	X	0	1	0	1	F _{CLK} /2 ⁵	0.75MHz
	X	X	X	X	0	1	1	0	F _{CLK} /2 ⁶	375KHz
	X	X	X	X	0	1	1	1	F _{CLK} /2 ⁷	187.5KHz
	X	X	X	X	1	0	0	0	F _{CLK} /2 ⁸	93.75KHz
	X	X	X	X	1	0	0	1	F _{CLK} /2 ⁹	46.875KHz
	X	X	X	X	1	0	1	0	F _{CLK} /2 ¹⁰	23.44KHz
	X	X	X	X	1	0	1	1	F _{CLK} /2 ¹¹	11.72KHz
	X	X	X	X	1	1	0	0	F _{CLK} /2 ¹²	5.86KHz
	X	X	X	X	1	1	0	1	F _{CLK} /2 ¹³	2.93KHz
	X	X	X	X	1	1	1	0	F _{CLK} /2 ¹⁴	1.46KHz
	X	X	X	X	1	1	1	1	F _{CLK} /2 ¹⁵	732Hz
1	0	0	0	0	X	X	X	X	F _{CLK}	24MHz
	0	0	0	1	X	X	X	X	F _{CLK} /2	12MHz
	0	0	1	0	X	X	X	X	F _{CLK} /2 ²	6MHz
	0	0	1	1	X	X	X	X	F _{CLK} /2 ³	3MHz
	0	1	0	0	X	X	X	X	F _{CLK} /2 ⁴	1.5MHz
	0	1	0	1	X	X	X	X	F _{CLK} /2 ⁵	0.75MHz
	0	1	1	0	X	X	X	X	F _{CLK} /2 ⁶	375KHz
	0	1	1	1	X	X	X	X	F _{CLK} /2 ⁷	187.5KHz
	1	0	0	0	X	X	X	X	F _{CLK} /2 ⁸	93.75KHz
	1	0	0	1	X	X	X	X	F _{CLK} /2 ⁹	46.875KHz
	1	0	1	0	X	X	X	X	F _{CLK} /2 ¹⁰	23.44KHz
	1	0	1	1	X	X	X	X	F _{CLK} /2 ¹¹	11.72KHz
	1	1	0	0	X	X	X	X	F _{CLK} /2 ¹²	5.86KHz
	1	1	0	1	X	X	X	X	F _{CLK} /2 ¹³	2.93KHz
	1	1	1	0	X	X	X	X	F _{CLK} /2 ¹⁴	1.46KHz
	1	1	1	1	X	X	X	X	F _{CLK} /2 ¹⁵	732Hz

Note: To change the clock selected as F_{CLK} (change the value of the system clock control register (CKC)), the change must be made after stopping the operation of the universal serial communication unit (SCI) (serial channel stop register m (STm) = 000FH).

X: Ignore;

m: unit number (m=0);

n: channel number (n=0, 1).

12.5.8 Procedure for Handling Errors During 3-Wire Serial I/O Communication (SSPI00)

In 3-wire serial I/O (SSPI00), the processing steps when an error occurs during communication are shown in Figure 12-29.

Table 12-29: Steps to handle when an overrun error occurs

Software operation	Hardware status	Remark
Read the serial data register mn (SDRmn) →	The BFFmn bit of the SSRmn register is “0” and the channel n is in receivable status.	This is to prevent overrun errors from ending the next reception during mishandling.
Read the serial status register mn (SSRmn)		The type of error is judged, and the read value is used to clear the error flag.
Set the serial flag clear trigger register mn (SDIRmn) to 1 →	Clear the error flag.	By writing the read value of the SSRmn register directly to the SDIRmn register, errors during read operations can only be cleared.

Remark: m: unit number (m=0);

n: channel number (n=0, 1).

12.6 Operation of Clock-Synchronous Serial Communication with Slave Selection Input Function

Channel 0 of SCIO is the channel that supports clock-synchronous serial communication with the slave select input function.

[Data transmission and reception]

- Data length of 7, 8 or 9 bits
- Phase control for transmitting and receiving data
- MSB/LSB FIRST
- Level settings for transmitting and receiving data

[Clock control]

- Phase control of input/output clocks
- Set the transfer period generated by the prescaler and the in-channel counter.
- Maximum transfer rate ^{Note} for slave communication: Max.6MHz

[Interrupt function]

- Transfer end interrupt, buffer empty interrupt

[Error detection flag]

- Overrun error

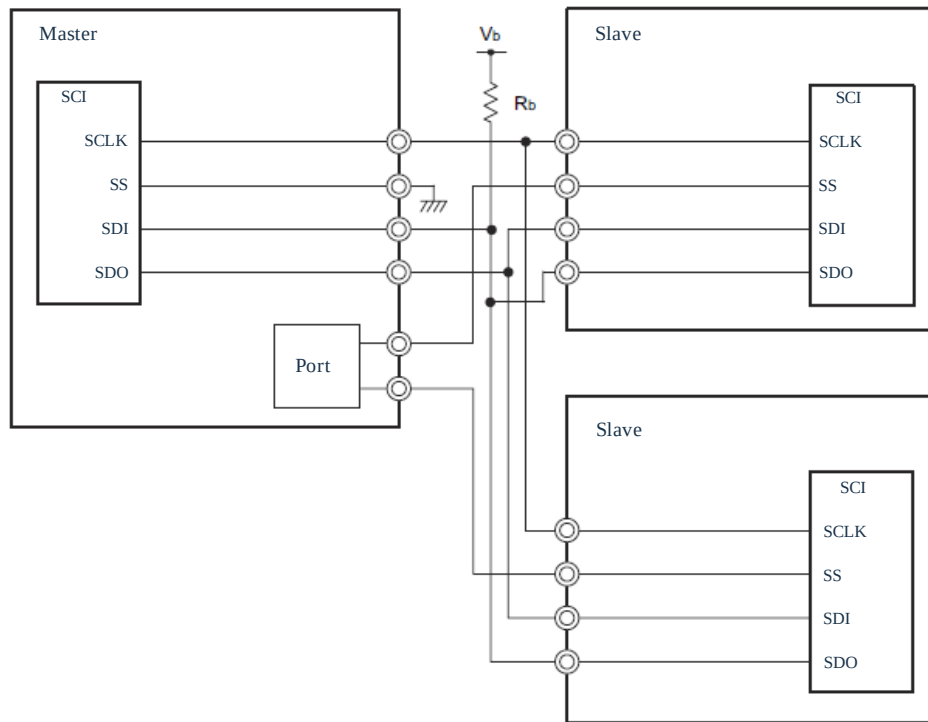
The slave select input function operates in three types of communication:

- Slave transmission (see 12.6.1)
- Slave reception (see 12.6.2)
- Slave transmission and reception (see 12.6.3)

Caution: It must be used within the range specified by the SCLK cycle time (T_{KCY}) characteristics. For detailed information, please refer to the datasheet.

By using the slave select input function, one master device can be connected to multiple slave devices for communication. The master device outputs a slave selection signal to the slave device (1) of the communication target, and each slave device determines whether it is selected as a communication target and controls the output of the SDO pin. When selected as a slave device for communication target, the SDO pin can communicate with the master device to send data; When a slave device is not selected as a communication target, the SDO pin becomes a high-level output, so in the environment where multiple slaves are connected, the SDO pin needs to be set to Nch-O.D and the node pulled up. In addition, even the serial clock of the input master device is not transmitted and received.

Figure 12-50: Structural example of slave select input function

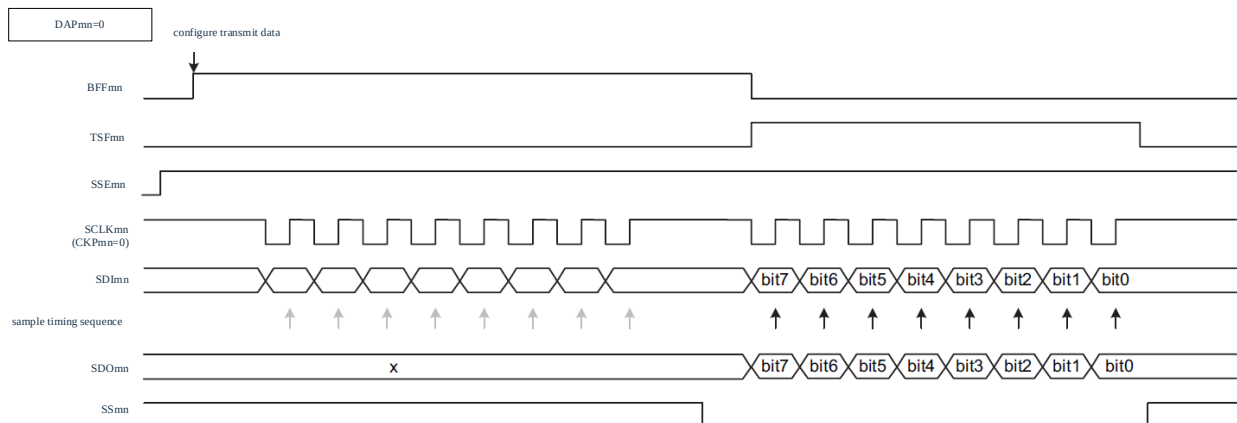


Caution:

The SDO00 pin is selected for N-channel open-drain output mode.

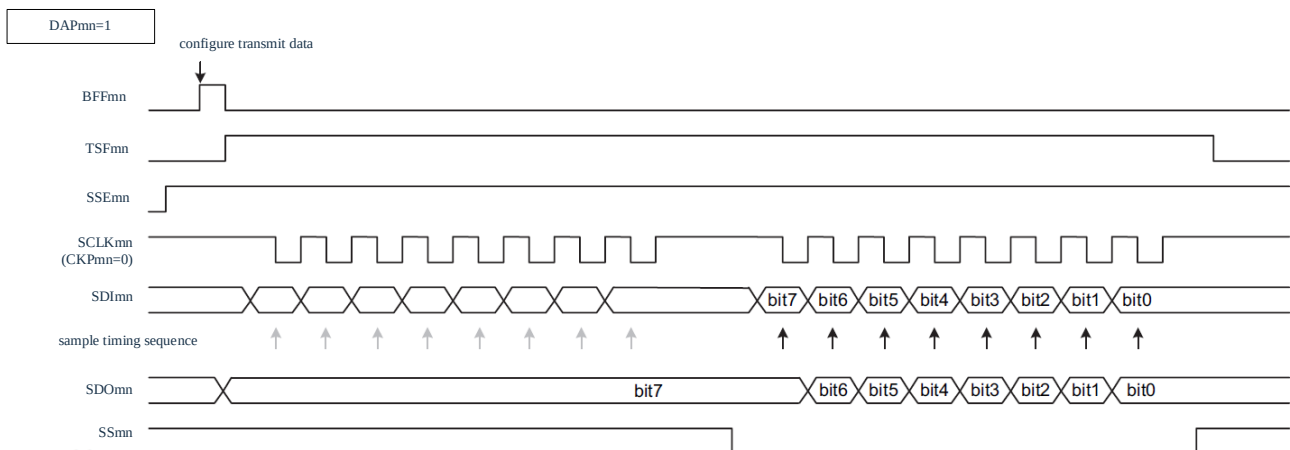
The slave select signal must be output through the port operation.

Figure 12-51: Slave timing diagram of the select input function



During SSmn is high, even on the falling edge of the SCKmn (serial clock), no transmission occurs, and no sampling of received data synchronized with the rising edge is taken.

During SSmn low, the output data is synchronized (shifted) with the falling edge of the serial clock and received synchronously with the rising edge.



When the DAPmn bit is “1”, the initial data (bit7) is supplied to the data output if the transmit data is set when SSmn is high. However, even the rising edge of the SCLK mn (serial clock) is not shifted, and the accepted data synchronized with the falling edge is not sampled. If SSmn goes low, the output data is synchronized (shifted) with the next rising edge and received synchronously with the falling edge.

Remark: m: unit number (m=0);

n: channel number (n=0).

12.6.1 Slave Transmission

Slave transmission refers to the operation where this product transmits data to other devices while receiving the transfer clock from other devices.

Table 12-30: Slave transmission

Slave select input function	SSPI00
Target channel	Channel of SCI0
Pin used	SCLKOI00, SDO00, SS00
Interrupt	INTSSPI00 Transfer end interrupt (in single transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Only the overrun error detection flag (OVFmn).
Transfer data length	7 or 8 bits
Transfer rate	Max.6M[Hz] ^{Note 1,2}
Data phase	It can be selected by the DAPmn bit of the SCRmn register. DAPmn=0: Start data output when the serial clock starts running. DAPmn=1: Start data output half a clock before the serial clock starts running.
Clock phase	It can be selected by the CKPmn bit of the SCRmn register. CKPmn=0: Non-reverse CKPmn=1: Reverse
Data direction	MSB or LSB first
Slave select input function	Operation of the slave selection function can be selected.

Note 1: Since the external serial clock input on the SCLK00 pin is sampled internally, the maximum transmission rate is 6 MHz.

Note 2: It must be used within the scope of peripheral functional characteristics that meet this condition and meet the electrical characteristics (see data sheet).

F_{MCK}: Target channel operation clock frequency

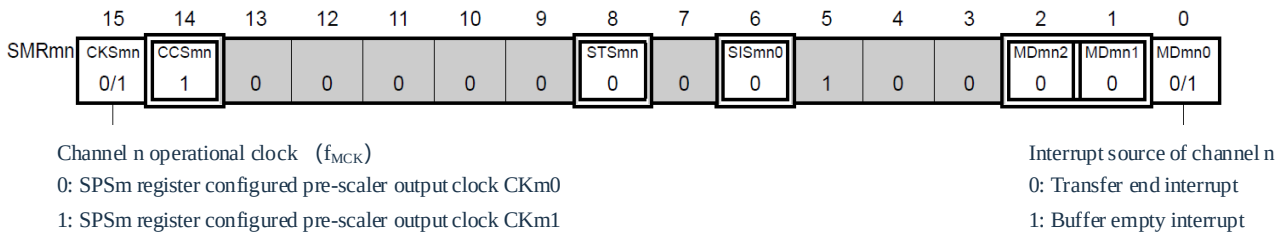
m: unit number (m=0);

n: channel number (n=0).

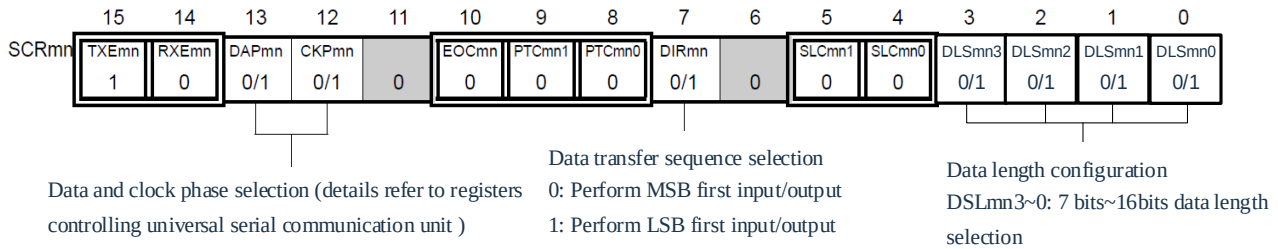
(1) Register setting

Figure 12-52: Example of register settings when slave select input function (SSPI00) slave transmits (1/2)

(a) Serial mode register mn (SMRmn)

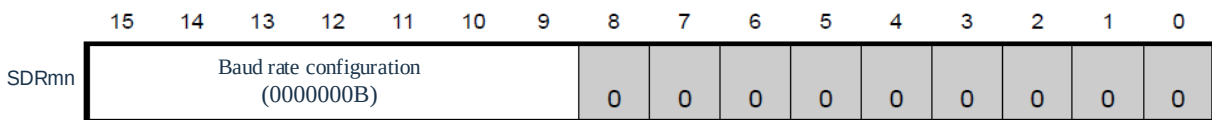


(b) Serial communication operation configuration register mn (SCRmn)

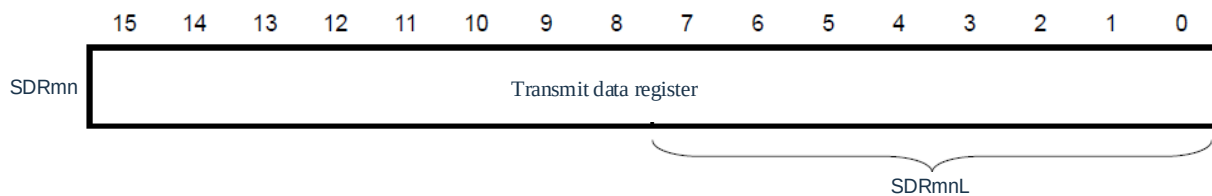


(c) Serial data register mn (SDRmn)

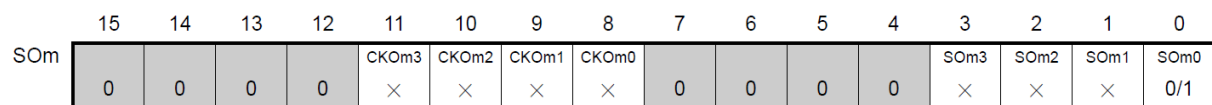
(1) Run stop (SEmn=0)



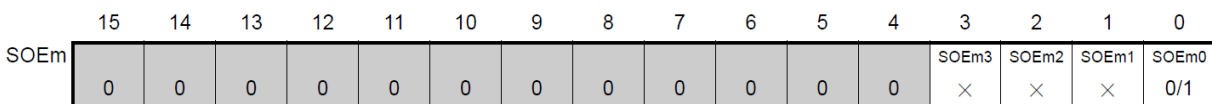
(2) During operation (SEmn=1) (low 8 bits: SDRmnL)



(d) Serial output register m (SOm).... Only configure target channel bits



(e) Serial output enable register m (SOEm).... Only set the bit of the target channel to 1.



Remark: m: unit number (m=0);

n: channel number (n=0);

p: SSPI number (p=00);

 : Fixed in SSPI master receive mode

 : Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set the initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

Figure 12-52: Example of register settings when slave select input function (SSPI00) slave transmits (2/2)

(f) Serial channel start register m (SSm) Only set bit of target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 ×	SSm0 0/1

(g) Input switch control register (ISC).... This is controlled by SS00 pin of SSPI00 slave channel (channel 0 of unit 0).

	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	ISC1 0/1	ISC0 0/1

0: SS00 pin input invalid
1: SS00 pin input valid

Remark: m: unit number (m=0);

n: channel number (n=0);

p: SSPI number (p=00);

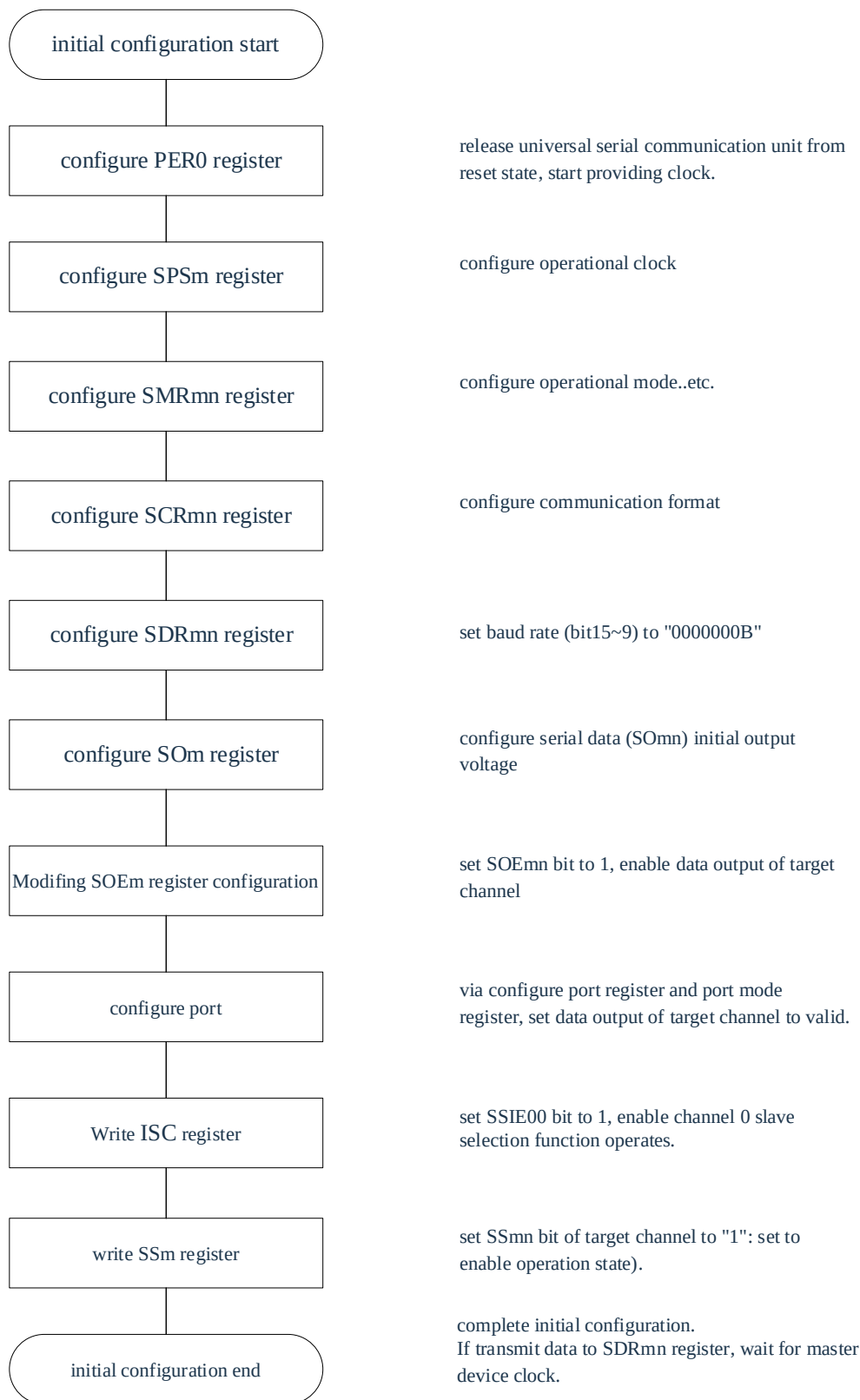
☐: Fixed in SSPI master receive mode ☐: Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set the initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

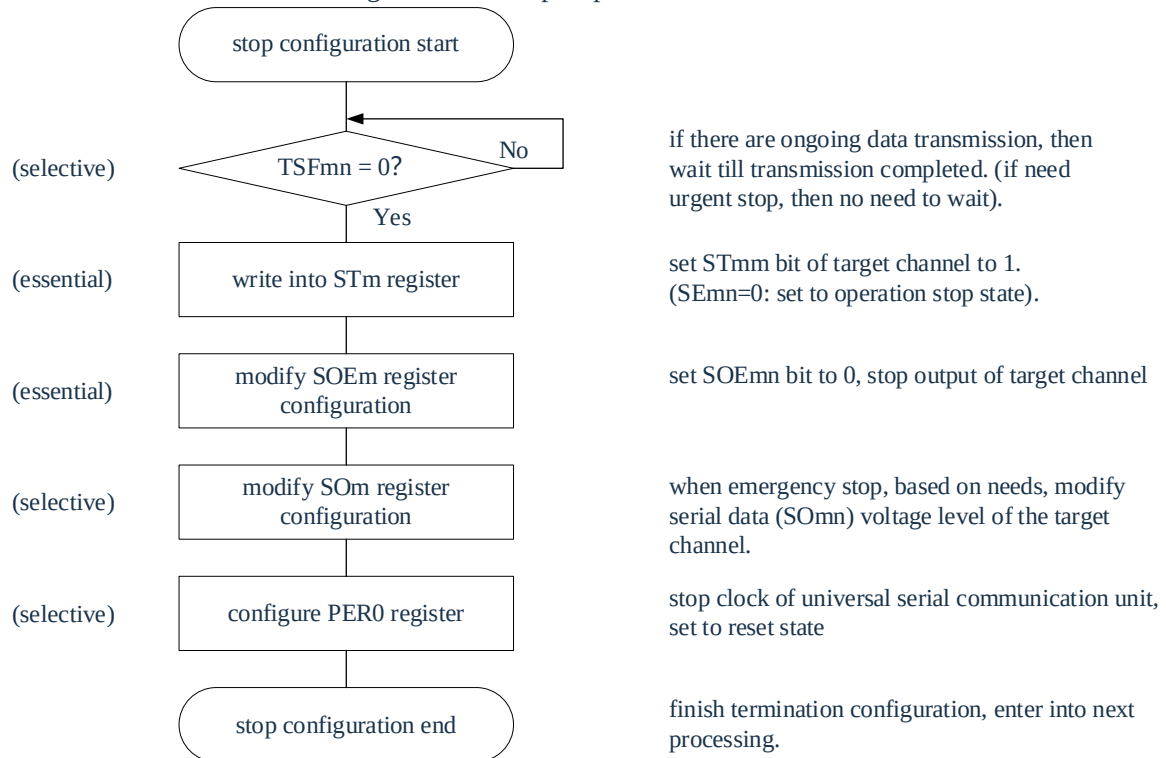
(2) Procedure

Figure 12-53: Initial setup steps for slave transmission



Remark: m: unit number (m=0); n: channel number (n=0); p: SSPI number (p=00).

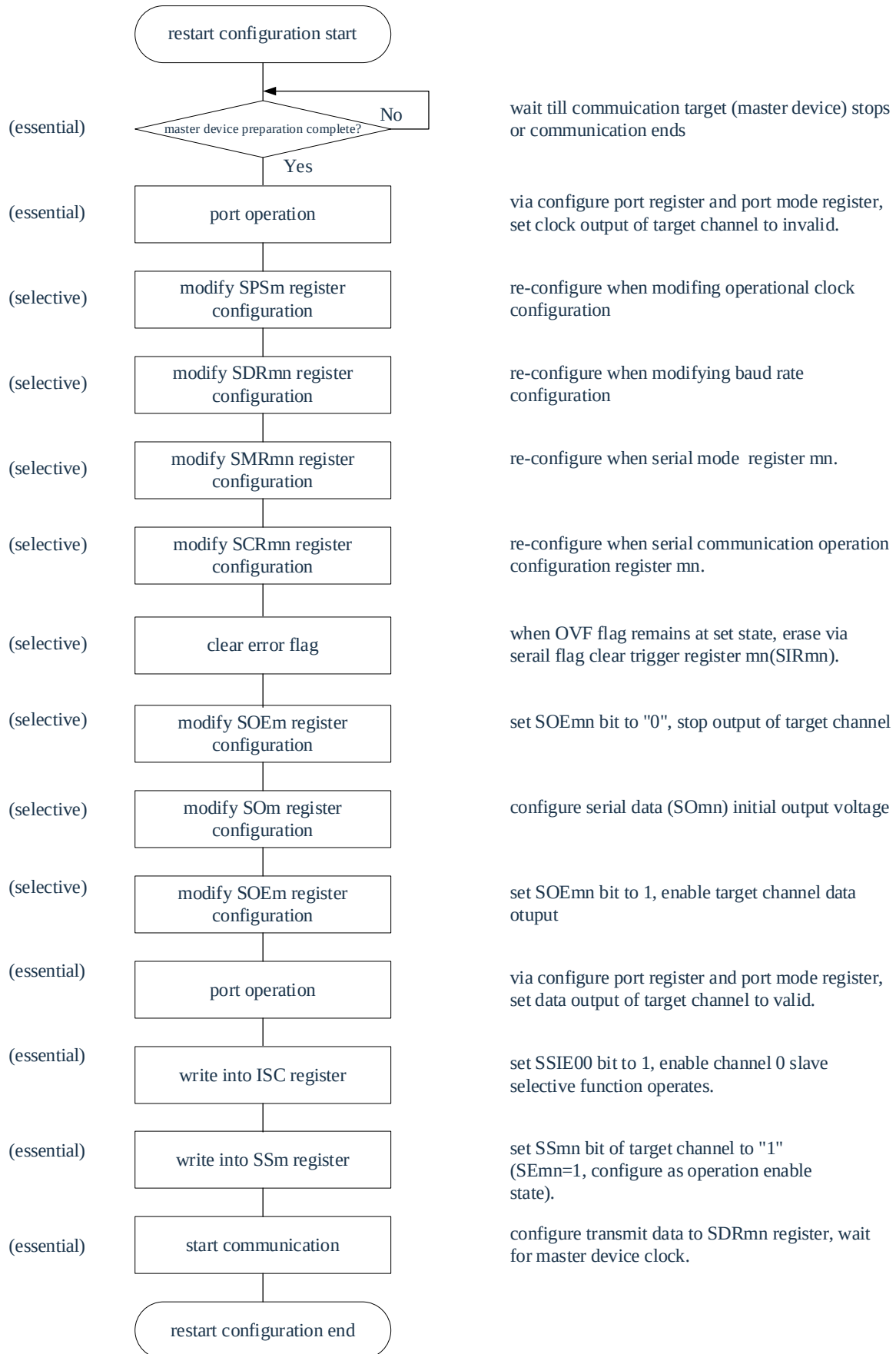
Figure 12-54: Stop steps for slave transmission



Caution: If the clock supply is stopped by rewriting PER0 in the abort setting, the initial setup must be performed after waiting for the communication partner (master device) to stop or the communication to end, rather than performing a restart setup.

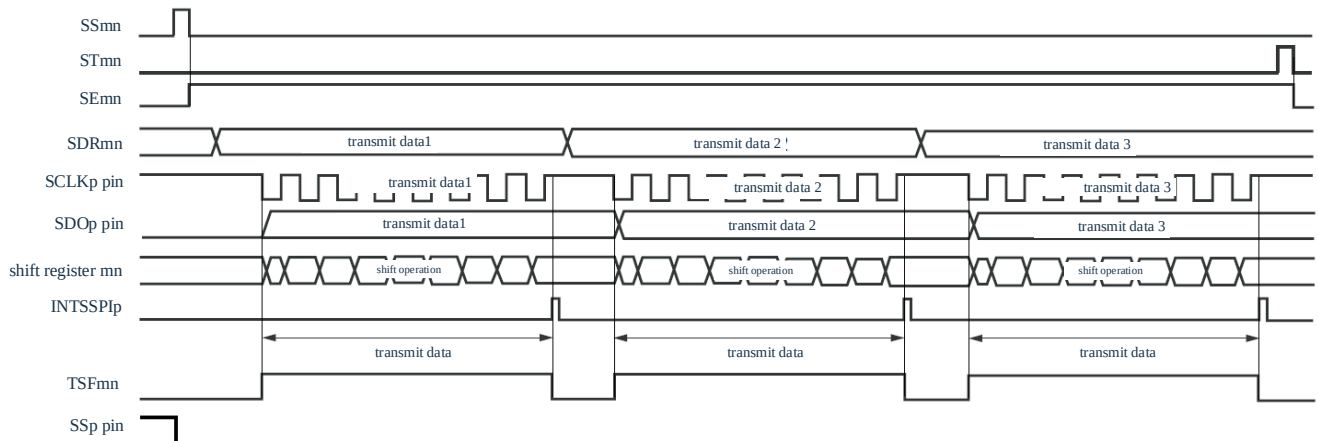
Remark: m: unit number (m=0); n: channel number (n=0); p: SSPI number (p=00);

Figure 12-55: Restart steps for slave transmission



(3) Processing flow (single transmit mode)

Figure 12-56: Timing diagram of slave transmission (single transmit mode) (type 1: DAPmn=0, CKPmn=0)

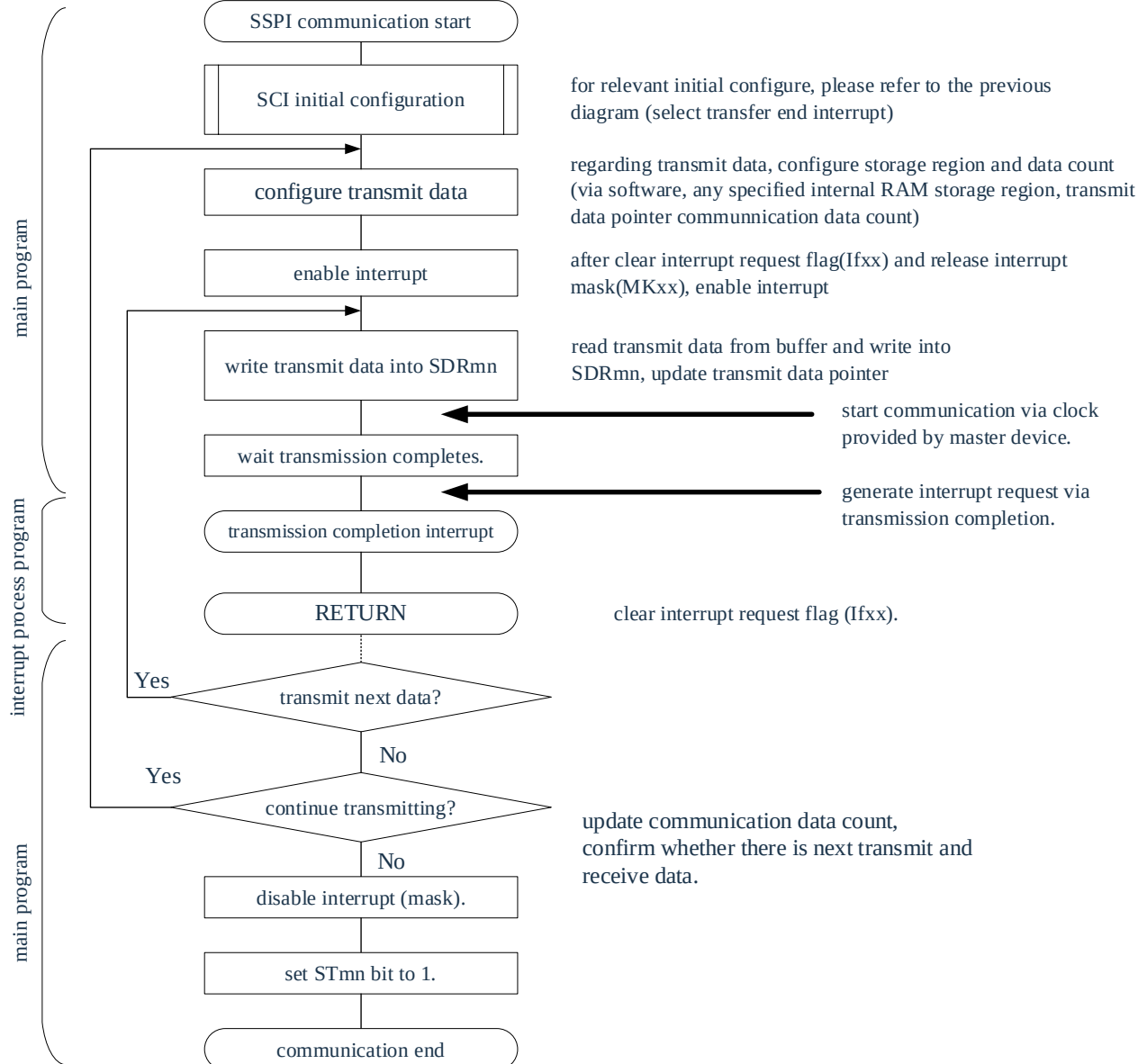


Remark: m: unit number (m=0);

n: channel number (n=0);

p: SSPI number (p=00).

Figure 12-57: Flowchart of slave transmission (single transmit mode)



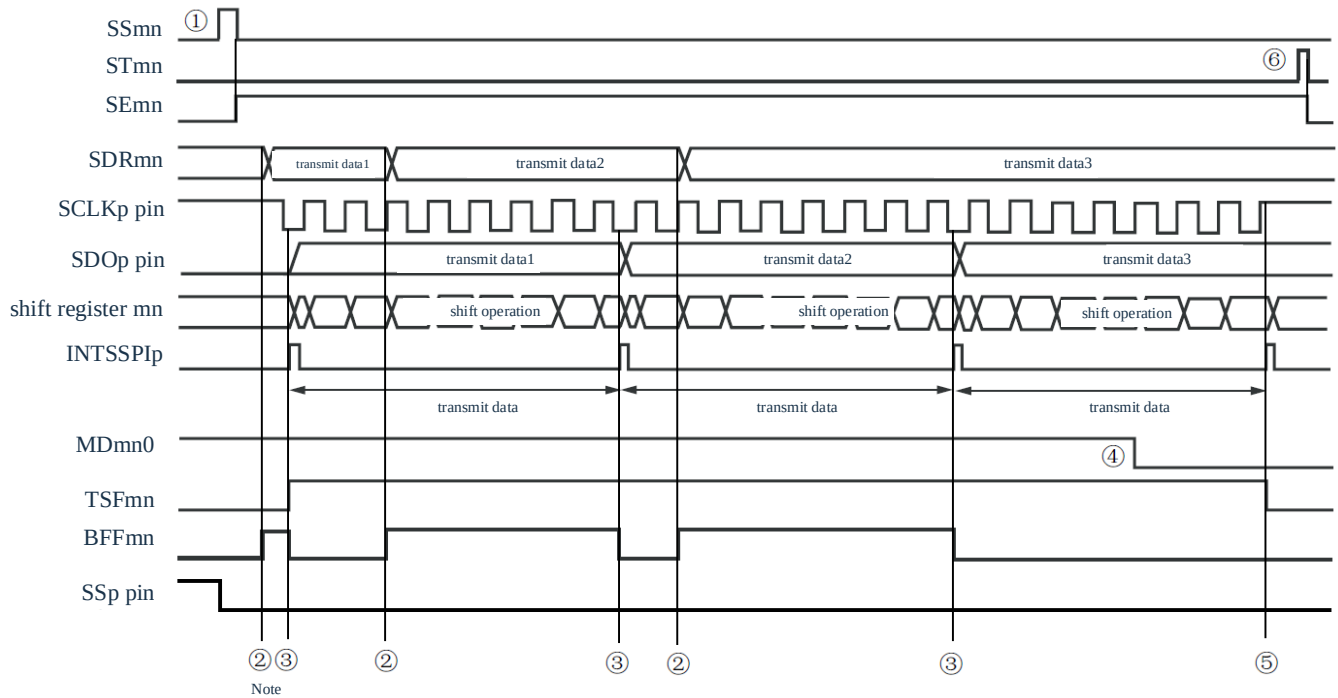
Remark: m: unit number (m=0);

n: channel number (n=0);

p: SSPI number (p=00).

(4) Processing flow (continuous transmit mode)

Figure 12-58: Timing diagram of slave transmit (continuous transmit mode) (type 1: DAPmn=0, CKPmn=0)



Note: If data is written to the SDRmn register while the BFFmn bit of the serial status register mn (SSRmn) is “1” (when valid data is stored in the serial data register mn (SDRmn)), the transmit data will be overwritten.

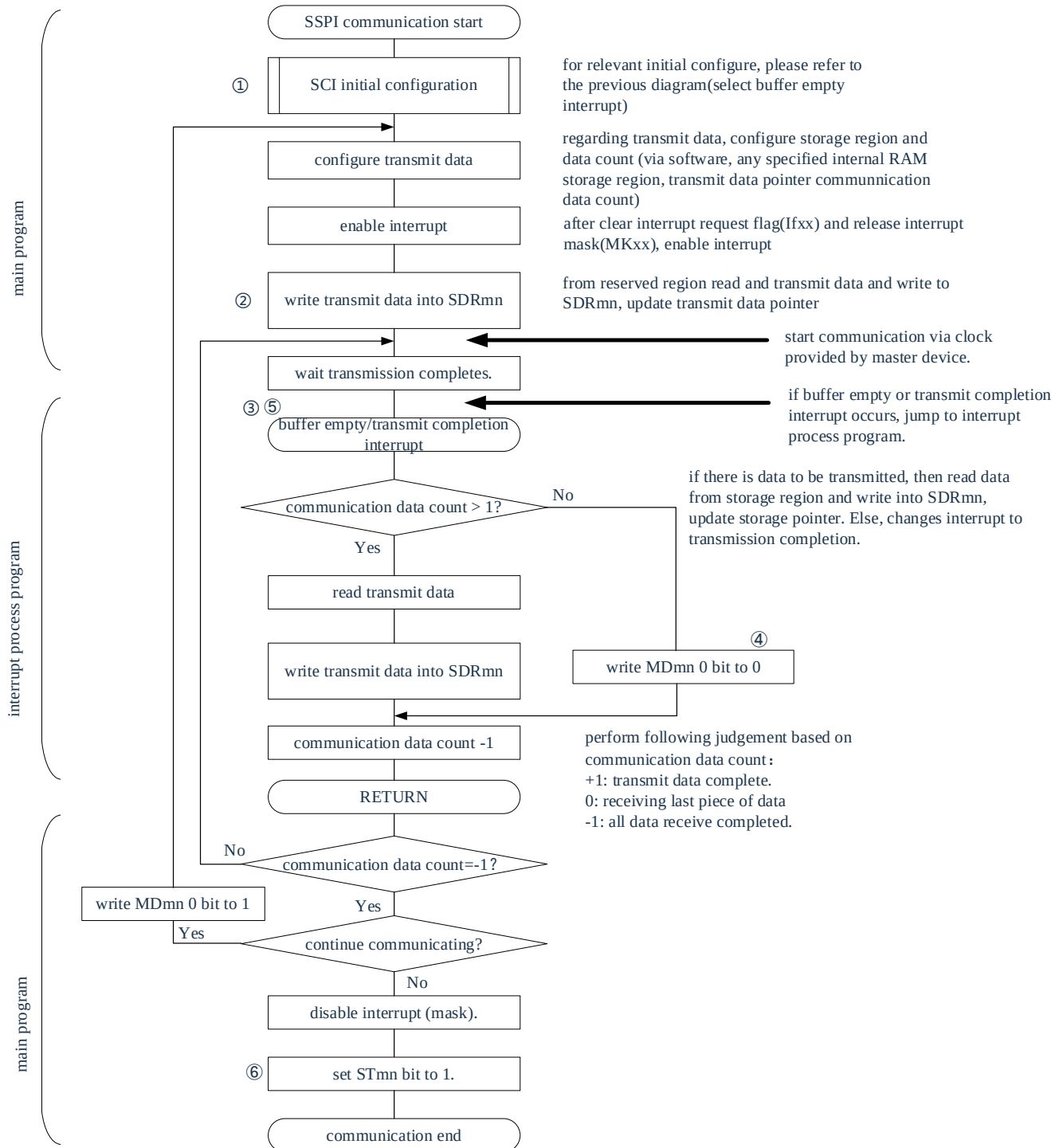
Caution: The MDmn0 bit of the serial mode register mn (SMRmn) can be rewritten even during operation. However, the rewriting must be done before the transmission of the last bit begins.

Remark: m: unit number (m=0);

n: channel number (n=0);

p: SSPI number (p=00).

Figure 12-59: Flowchart of slave transmission (continuous send mode)



Remark: ① to ⑥ correspond to ① to ⑥ in “Figure 12-25 Timing diagram of slave transmit (continuous transmit mode)”.

m: unit number (m=0);

n: channel number (n=0);

p: SSPI number (p=00).

12.6.2 Slave Reception

Slave reception refers to the operation of this product to receive data from other devices in the state of transmitting clocks from other devices.

Table 12-31: Slave reception

Slave select input function	SSPI00
Target channel	Channel of SCIO
Pin used	SCLKOI00, SDI00, SS00
Interrupt	INTSSPI00
	Limited to transfer end interrupts (disable setting buffer empty interrupts).
Error detection flag	Only the overrun error detection flag (OVFmn).
Transfer data length	7 or 8 bits
Transfer rate	Max.6M[Hz] ^{Note 1,2}
Data phase	It can be selected by the DAPmn bit of the SCRmn register.
	DAPmn=0: Start data output when the serial clock starts running. DAPmn=1: Start data output half a clock before the serial clock starts running.
Clock phase	It can be selected by the CKPmn bit of the SCRmn register.
	CKPmn=0: Non-reverse CKPmn=1: Reverse
Data direction	MSB or LSB first
Slave select input function	Select the operation of the slave select input function.

Note 1: Since the external serial clock input on the SCLK00 pin is sampled internally, the maximum transmission rate is 6 MHz.

Note 2: It must be used within the scope of peripheral functional characteristics that meet this condition and meet the electrical characteristics (see data sheet).

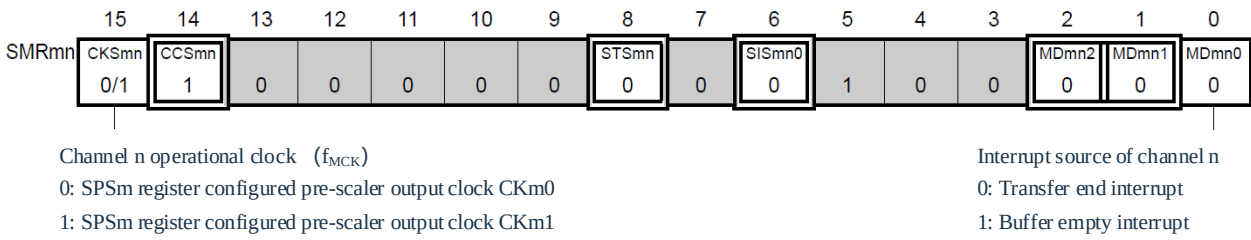
F_{MCK}: Target channel operation clock frequency

m: unit number (m=0) n: channel number (n=0).

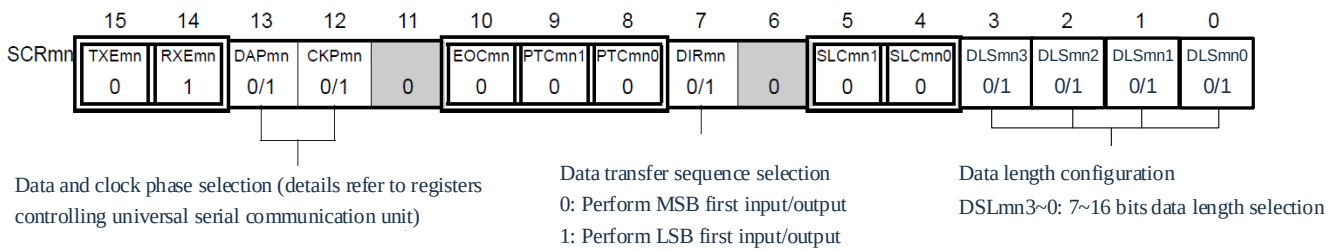
(1) Register setting

Figure 12-60: Example of register setting contents for slave selection input function (SSPI00) slave reception (1/2)

(a) Serial mode register mn (SMRmn)

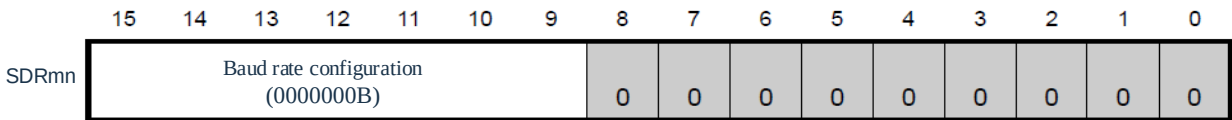


(b) Serial communication operation configuration register mn (SCRmn)

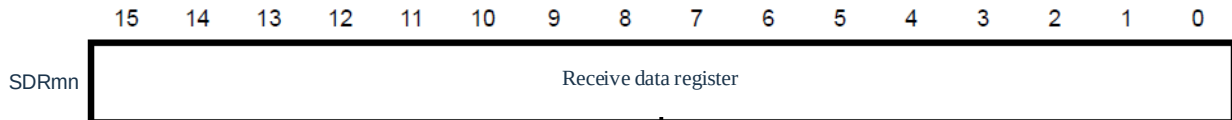


(c) Serial data register mn (SDRmn)

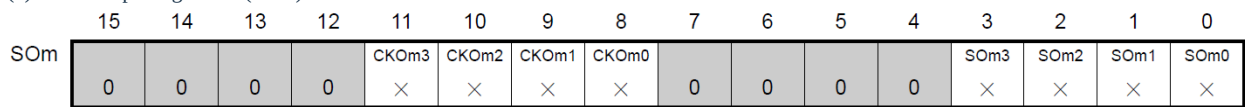
(1) Run stop (SEmn=0)



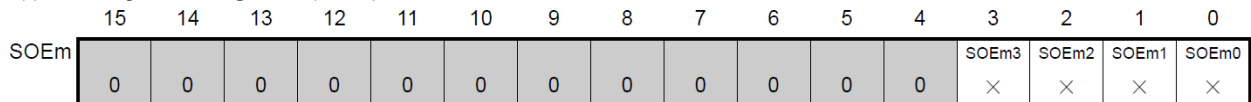
(2) During operation (SEmn=1) (low 8 bits: SDRmnL)



(d) Serial output register m (SOM).... Not used in this mode.



(e) Serial output enable register m (SOEm).... Not used in this mode.



Remark: m: unit number (m=0);

n: channel number (n=0);

p: SSPI number (p=00);

□ : Fixed in SSPI master receive mode

■ : Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set the initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

Figure 12-60: Example of register setting contents for slave selection input function (SSPI00) slave reception (2/2)

(f) Serial channel start register m (SSm) Only set the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 ×	SSm0 0/1

(g) Input switch control register (ISC).... This is controlled by the SS00 pin of the SSPI00 slave channel (channel 0 of unit 0).

	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	ISC1 0/1	ISC0 0/1

0: SS00 pin input invalid
1: SS00 pin input valid

Remark: m: unit number (m=0);

n: channel number (n=0);

p: SSPI number (p=00);

☐ : Fixed in SSPI master receive mode ☐ : Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set the initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

(2)Procedure

Figure 12-61: Initial setup steps for slave reception

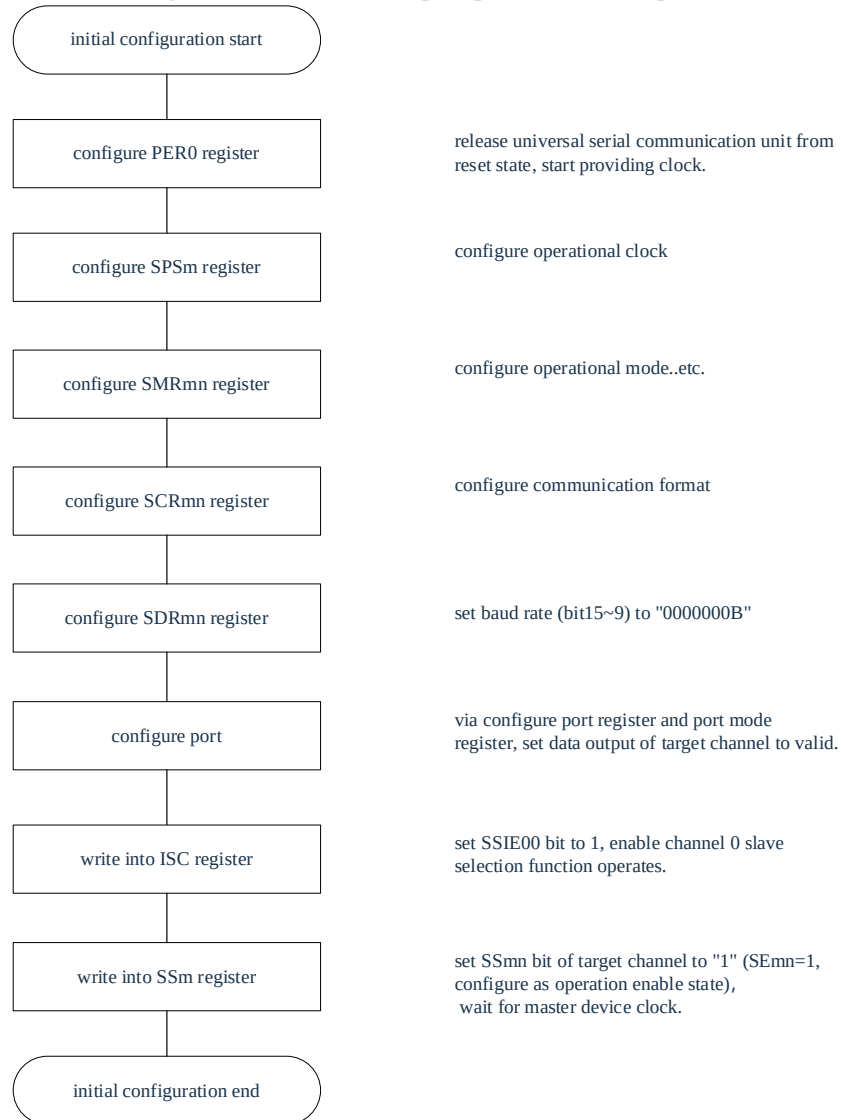
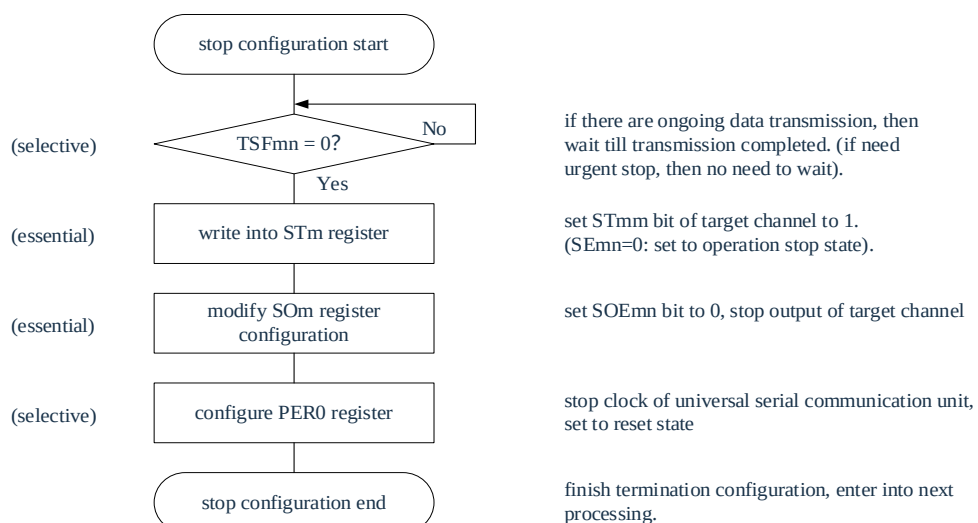
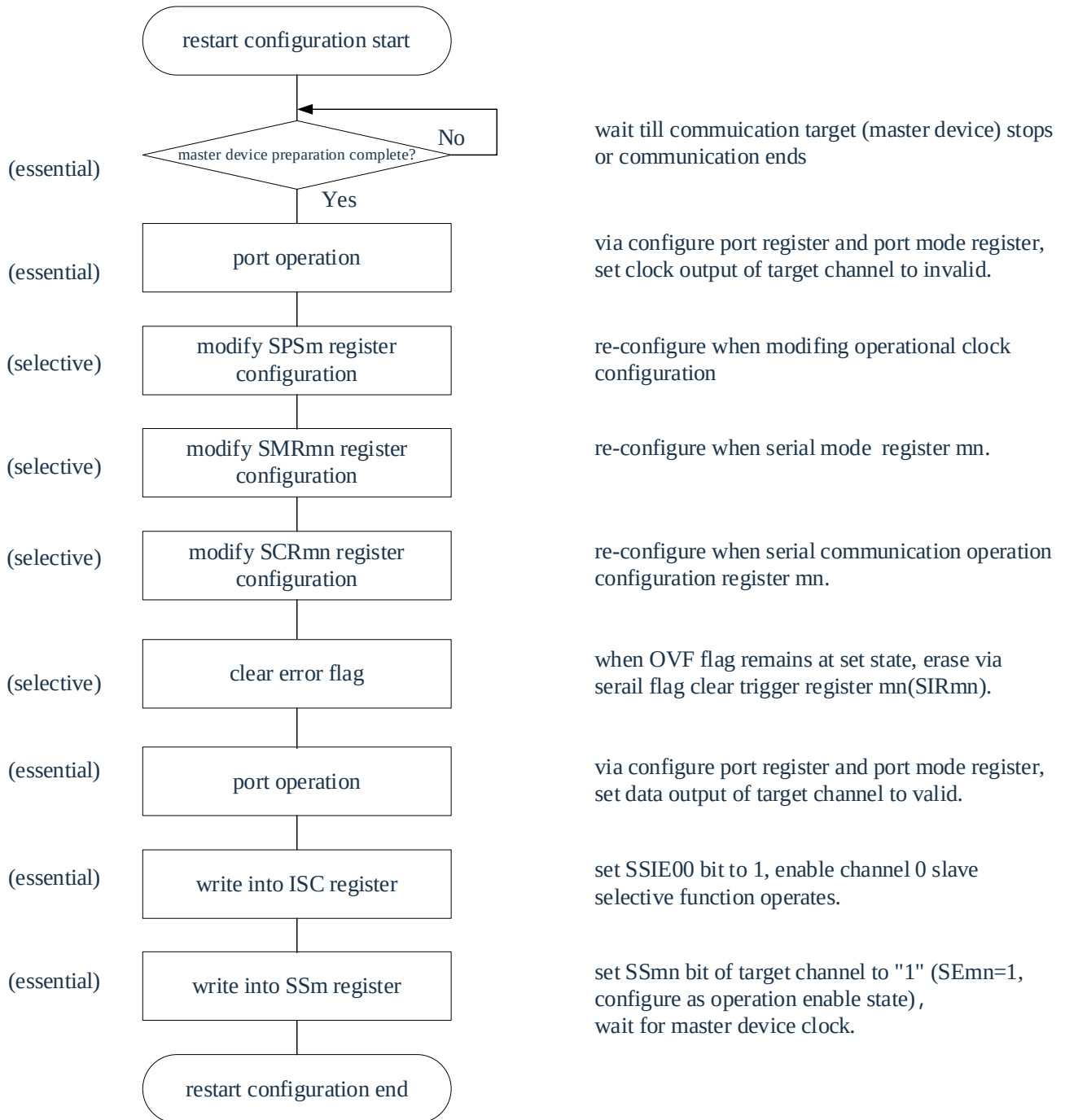


Figure 12-62: Stop steps for slave reception



Remark: m: unit number (m=0); n: channel number (n=0); p: SSPI number (p=00).

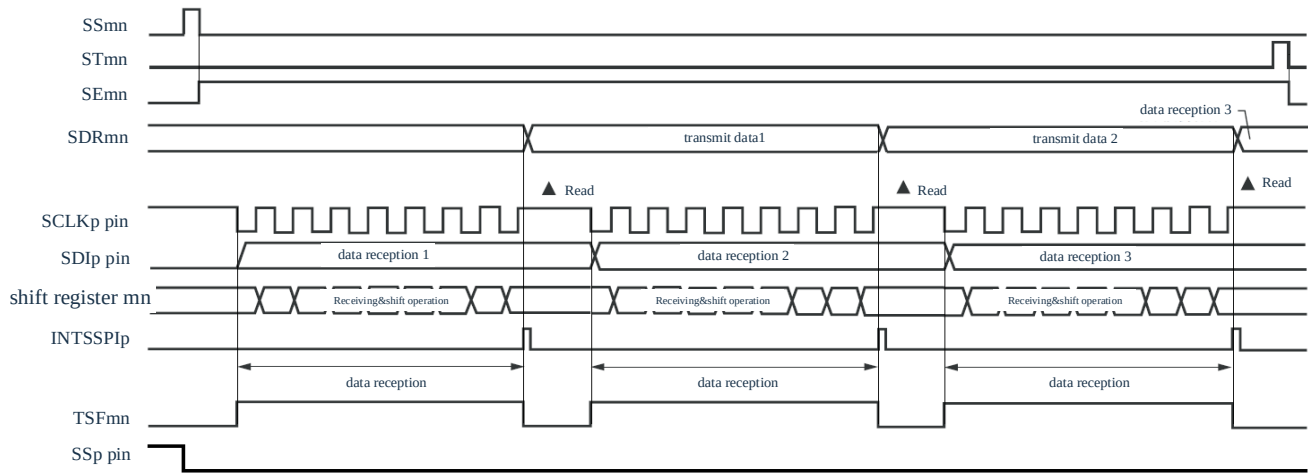
Figure 12-63: Restart steps for slave reception



Remark: m: unit number (m=0); n: channel number (n=0); p: SSPI number (p=00).

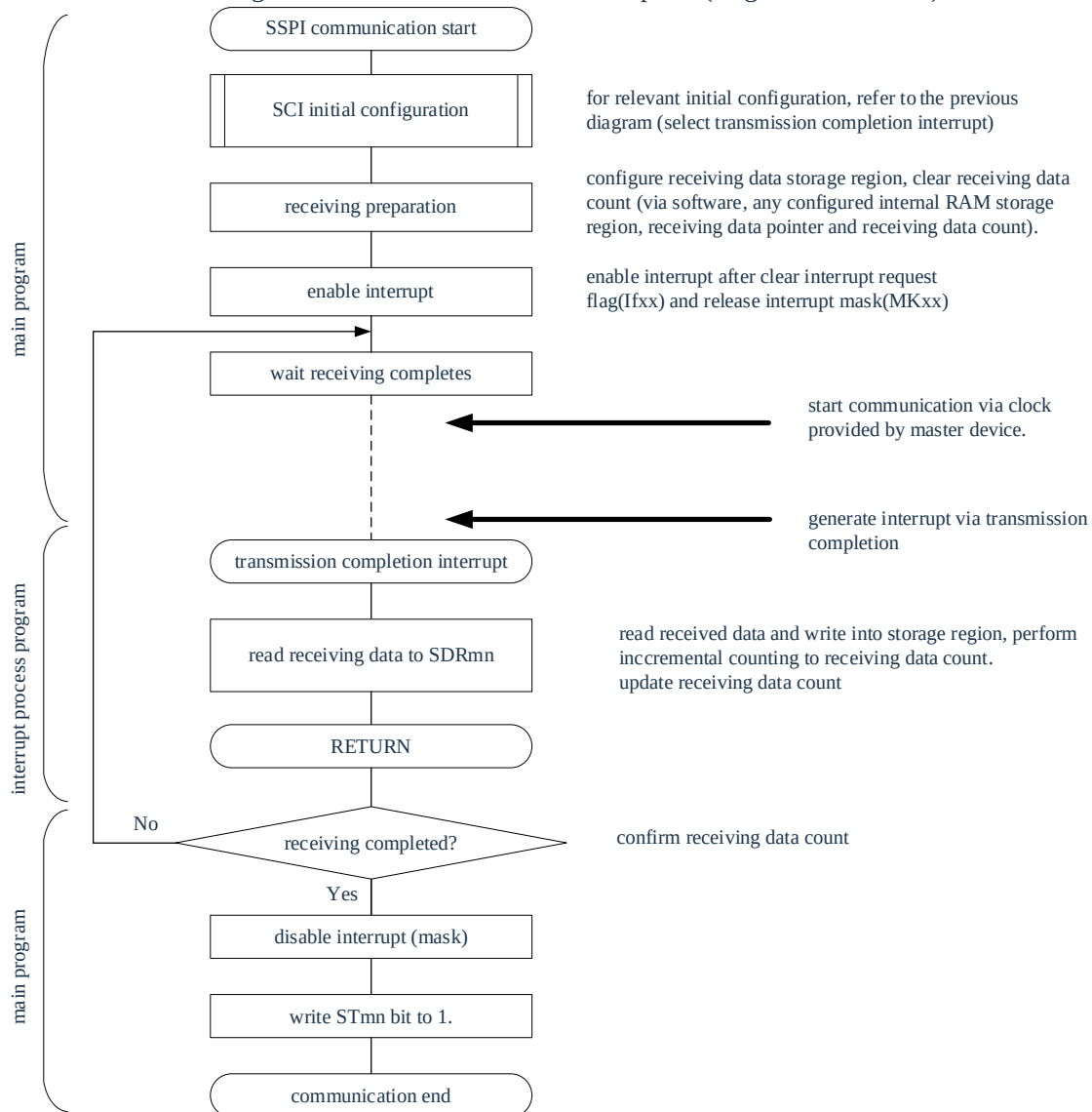
(3) Processing flow (single receive mode)

Figure 12-64: Timing diagram of slave reception (single receive mode) (type 1: DAPmn=0, CKPmn=0)



Remark: m: unit number (m=0); n: channel number (n=0); p: SSPI number (p=00).

Figure 12-65: Flowchart of slave reception (single receive mode)



12.6.3 Slave Transmission and Reception

Slave transmission and reception refer to the operation where this product and other devices perform data transmission and reception while receiving the transfer clock from other devices.

Table 12-32: Slave transmission and reception

Slave select input function	SSPI00
Target channel	Channel of SCIO
Pin used	SCLKOI00, SDI00, SDO00, SS00
Interrupt	INTSSPI00 Transfer end interrupt (in single transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Only the overrun error detection flag (OVFmn).
Transfer data length	7 or 8 bits
Transfer rate	Max.6M[Hz] ^{Note 1,2}
Data phase	It can be selected by the DAPmn bit of the SCRmn register. DAPmn=0: Start data output when the serial clock starts running. DAPmn=1: Start data output half a clock before the serial clock starts running.
Clock phase	It can be selected by the CKPmn bit of the SCRmn register. CKPmn=0: Non-reverse CKPmn=1: Reverse
Data direction	MSB or LSB first
Slave select input function	Select the operation of the slave select input function.

Note 1: Since the external serial clock input on the SCLK00 pin is sampled internally, the maximum transmission rate is 6 MHz.

Note 2: It must be used within the scope of peripheral functional characteristics that meet this condition and meet the electrical characteristics (see data sheet).

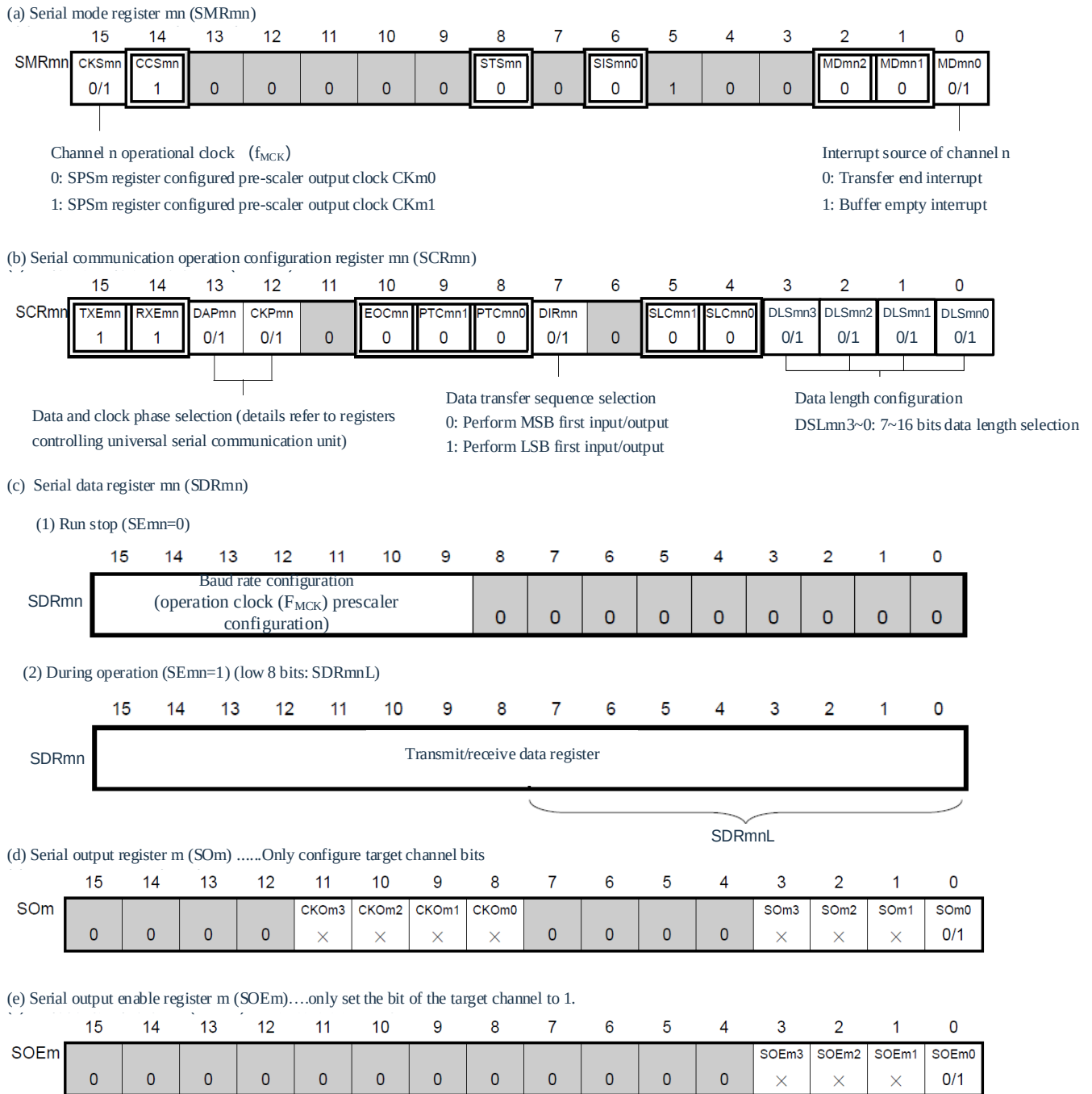
F_{MCK}: Target channel operation clock frequency

m: unit number (m=0);

n: channel number (n=0).

(1) Register setting

Figure 12-66: Example of register setting contents for slave selection input function (SSPI00) slave transmission and reception (1/2)



Caution: Before the master device starts outputting the clock, the transmit data must be set in the SDRmn register.

Remark: m: unit number (m=0);

n: channel number (n=0);

p: SSPI number (p=00);

□: Fixed in SSPI master receive mode

■: Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set the initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

Figure 12-66: Example of register setting contents for slave selection input function (SSPI00) slave transmission and reception (2/2)

(f) Serial channel start register m (SSm) Only set the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 ×	SSm0 0/1

(g) Input switch control register (ISC).... This is controlled by the SS00 pin of the SSPI00 slave channel (channel 0 of unit 0).

	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	ISC1 0/1	ISC0 0/1

0: SS00 pin input invalid
1: SS00 pin input valid

Caution: Before the master device starts outputting the clock, the transmit data must be set in the SDRmn register.

Remark: m: unit number (m=0);

n: channel number (n=0);

p: SSPI number (p=00);

☐: Fixed in SSPI master receive mode

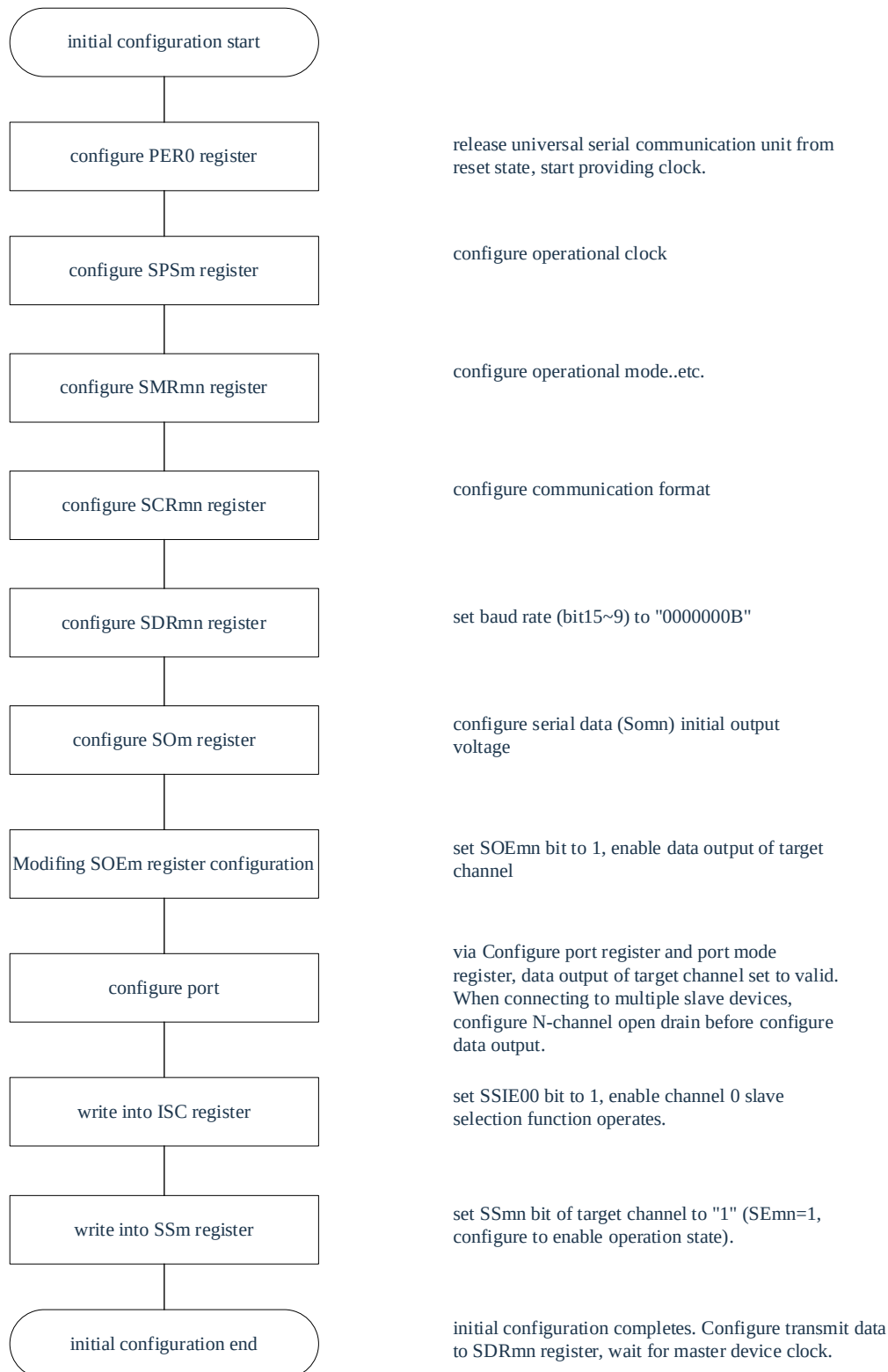
☐: Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set the initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

(2) Procedure

Figure 12-67: Initial setup steps for slave transmission and reception



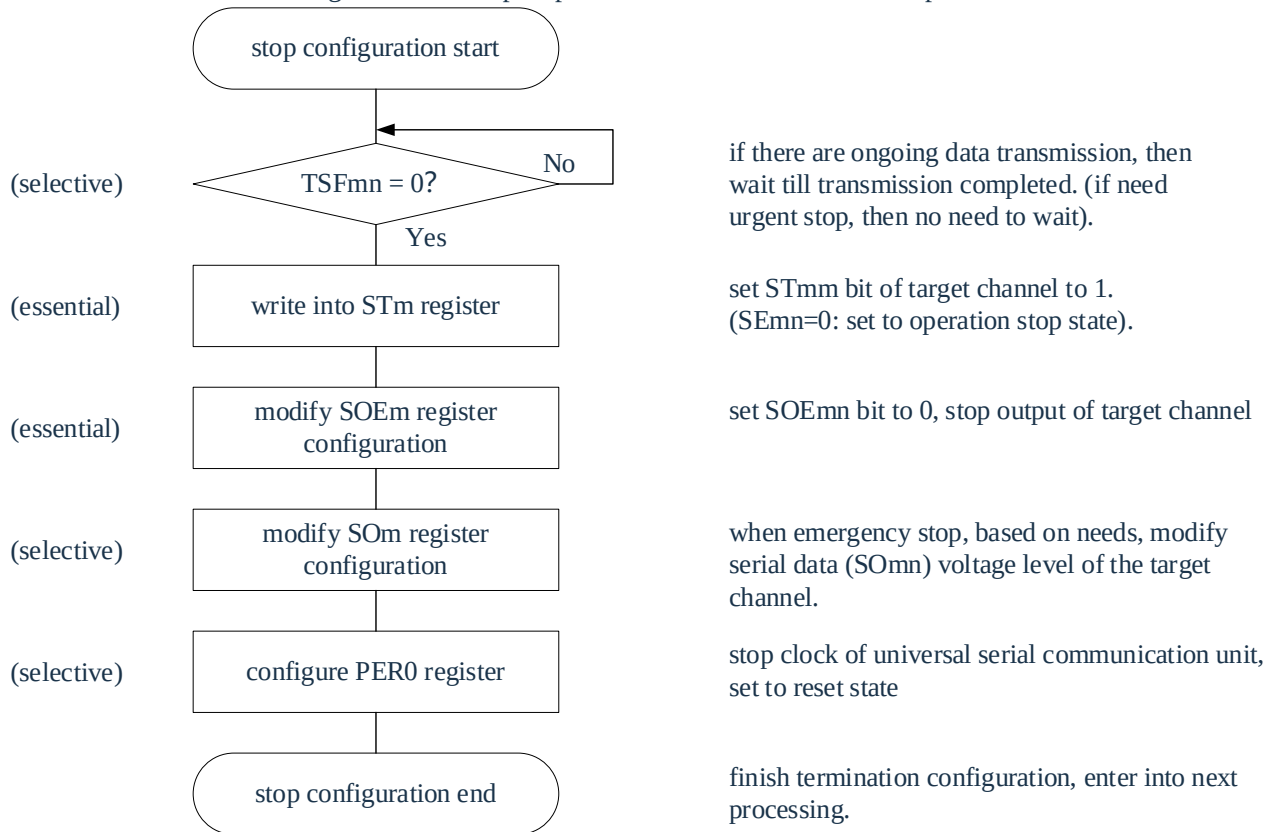
Caution: Before the master device starts outputting the clock, the transmit data must be set in the SDRmn register.

Remark: m: unit number (m=0);

n: channel number (n=0);

p: SSPI number (p=00).

Figure 12-68: Stop steps of slave transmission and reception

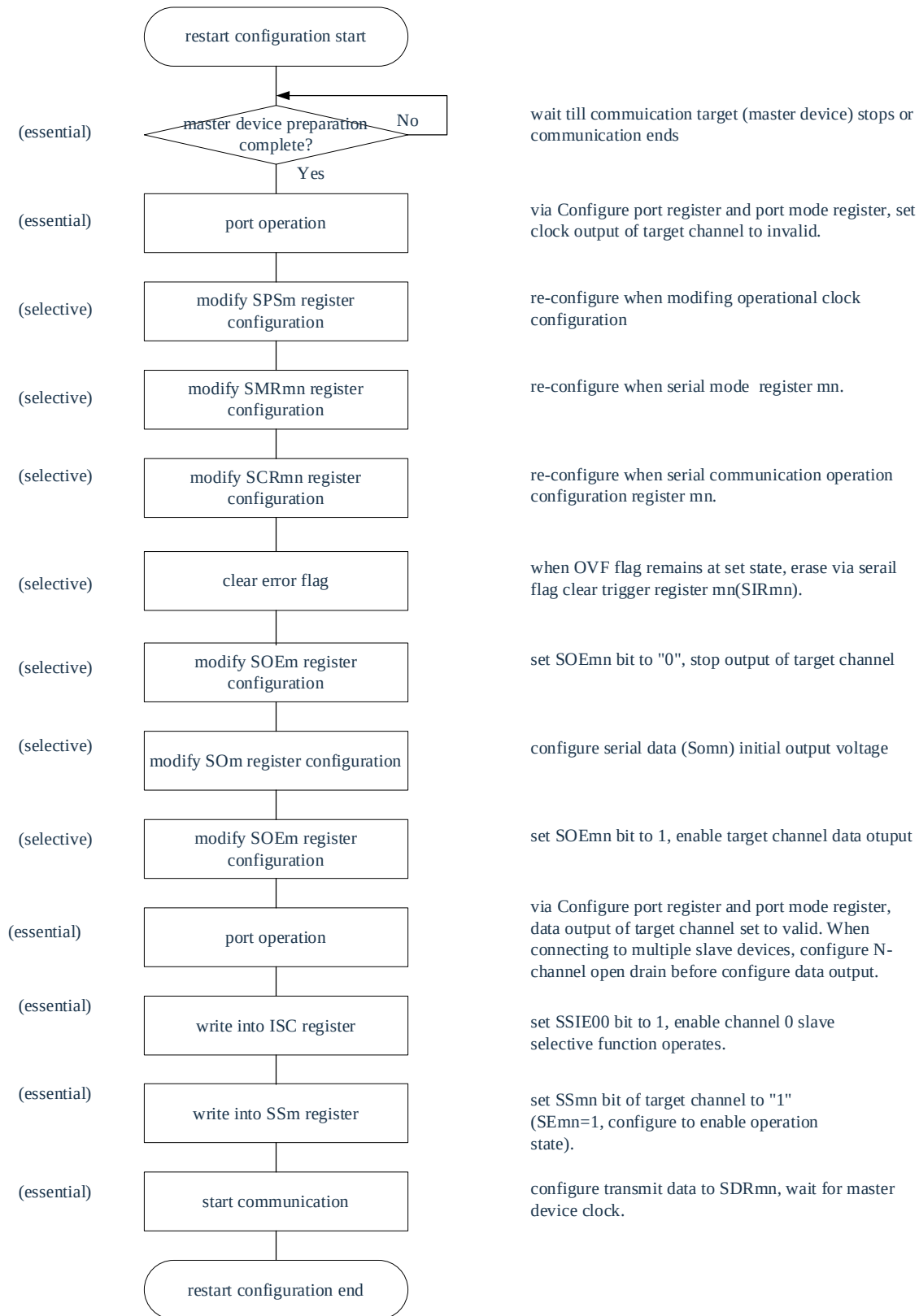


Remark: m: unit number (m=0);

n: channel number (n=0);

p: SSPI number (p=00).

Figure 12-69: Restart steps of slave transmission and reception



Caution:

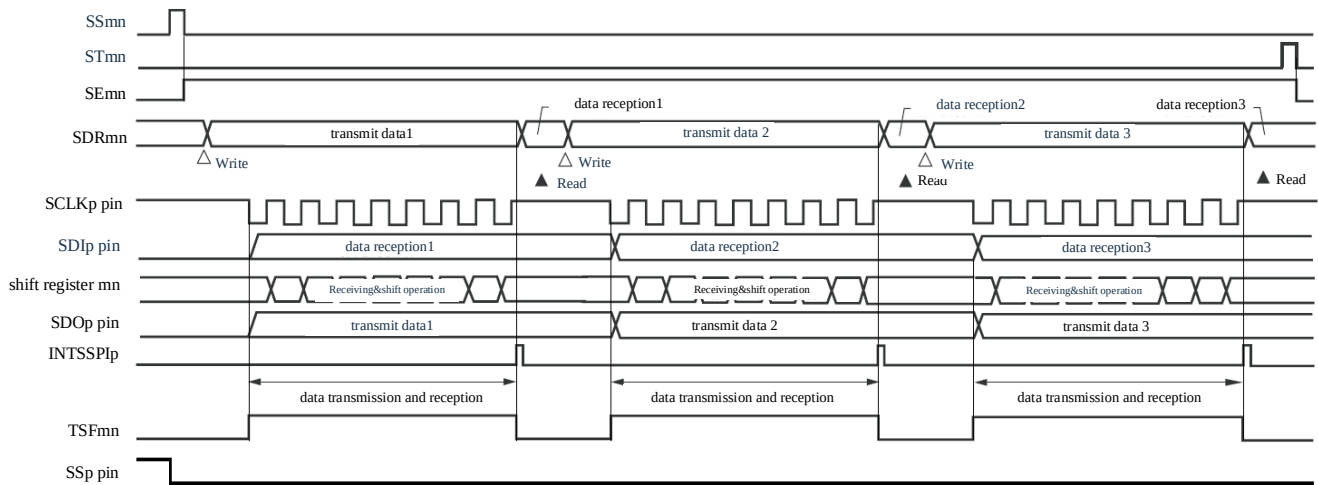
Before the master device starts outputting the clock, the transmit data must be set in the SDRmn register.

If the clock supply is stopped by rewriting PER0 in the abort setting, the initial setup must be performed after waiting for the communication partner (master device) to stop or the communication to end, rather than performing a restart setup.

(3) Processing flow (single transmit and receive mode)

Figure 12-70: Timing diagram of slave transmit and receive (single transmit and receive mode)

(type 1: DAPmn=0, CKPmn=0)

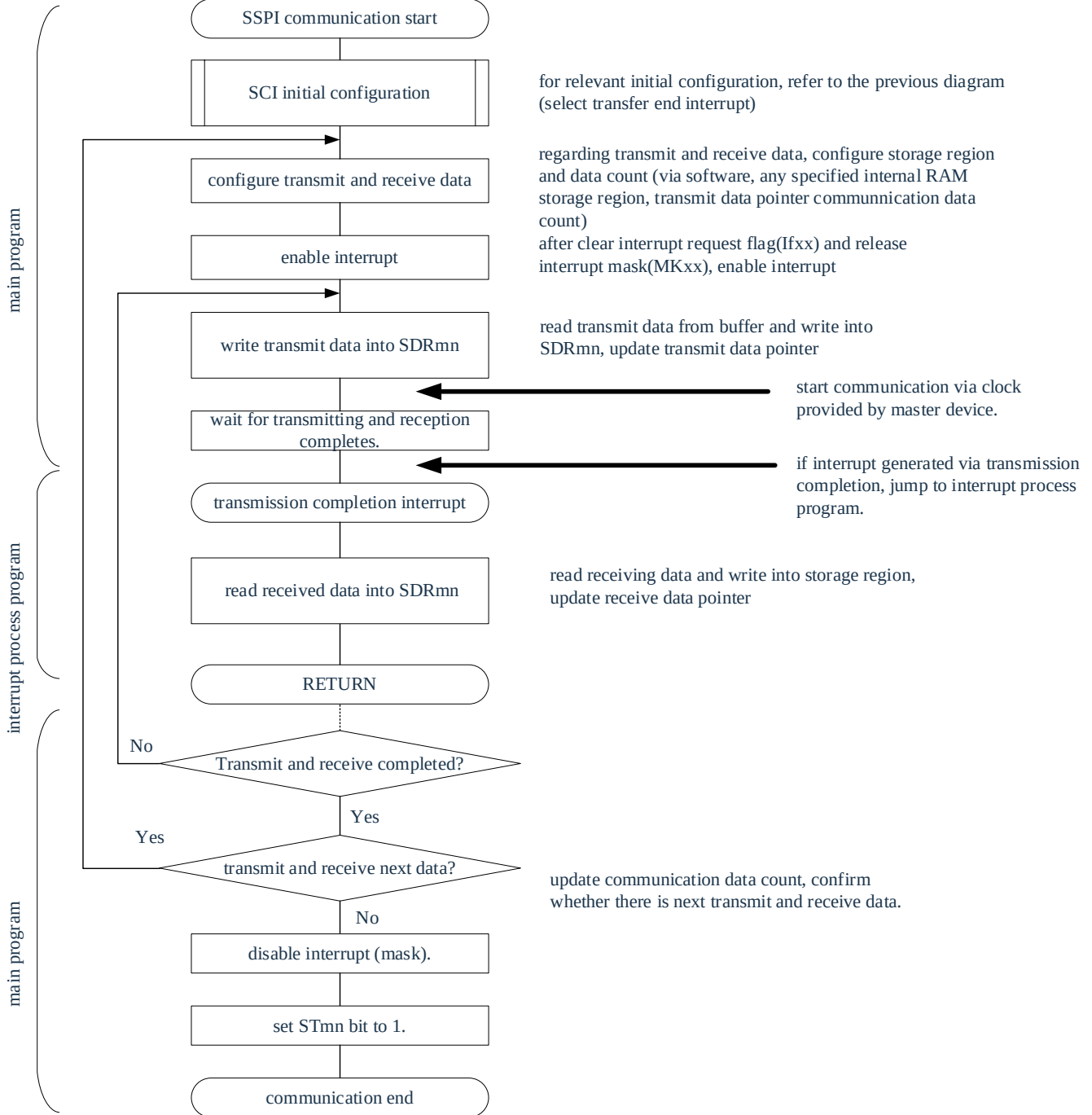


Remark: m: unit number (m=0);

n: channel number (n=0);

p: SSPI number (p=00).

Figure 12-71: Flowchart of slave transmission and reception (single transmit and receive mode)



Caution: Before the master device starts outputting the clock, the transmit data must be set in the SDRmn register.

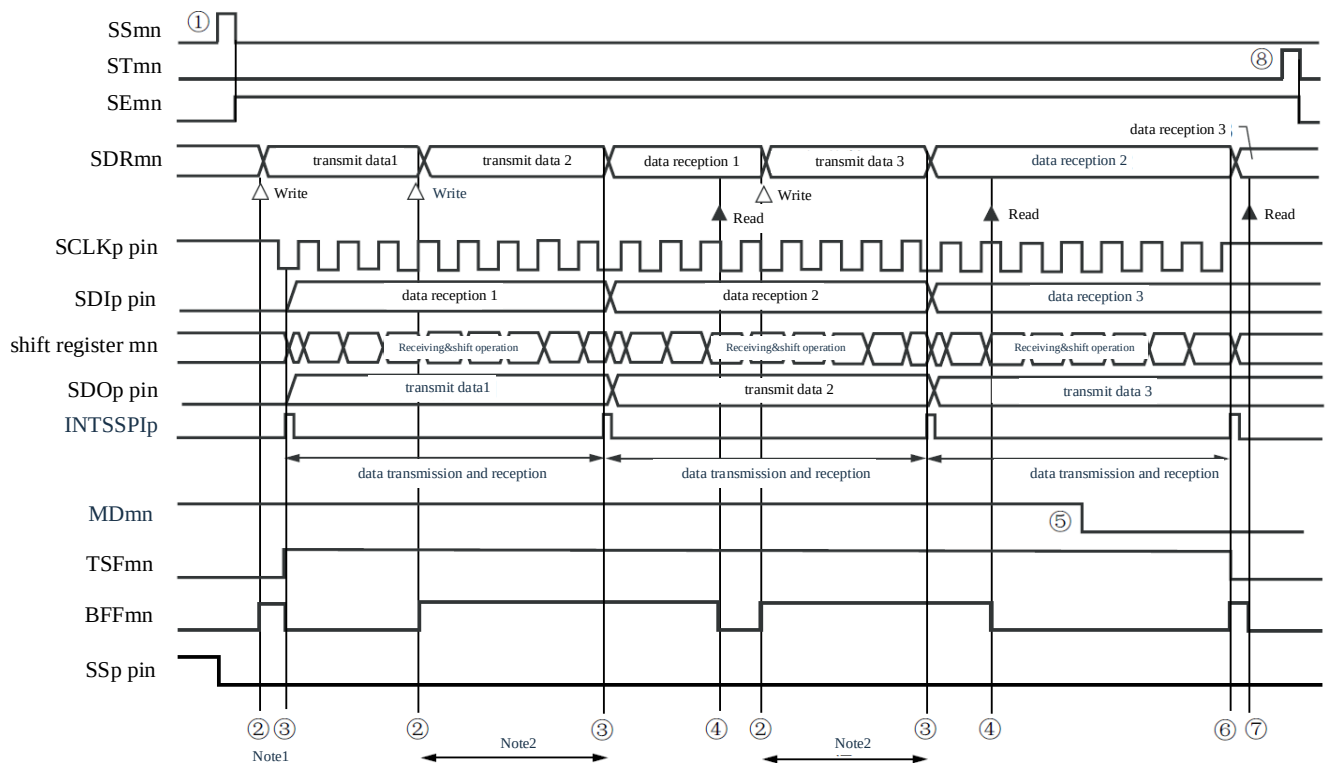
m: unit number (m=0);

n: channel number (n=0);

p: SSPI number (p=00).

(4) Processing flow (continuous transmit and receive mode)

Figure 12-72: Timing diagram of slave transmission and reception (continuous transmit and receive mode)
(type 1: DAPmn=0, CKPmn=0)



Note 1: If data is written to the SDRmn register while the BFFmn bit of the serial status register mn (SSRmn) is “1” (when valid data is stored in the serial data register mn (SDRmn)), the transmit data will be overwritten.

Note 2: If the SDRmn register is read during this time, the transfer data can be read. At this point, it does not affect the transfer operation.

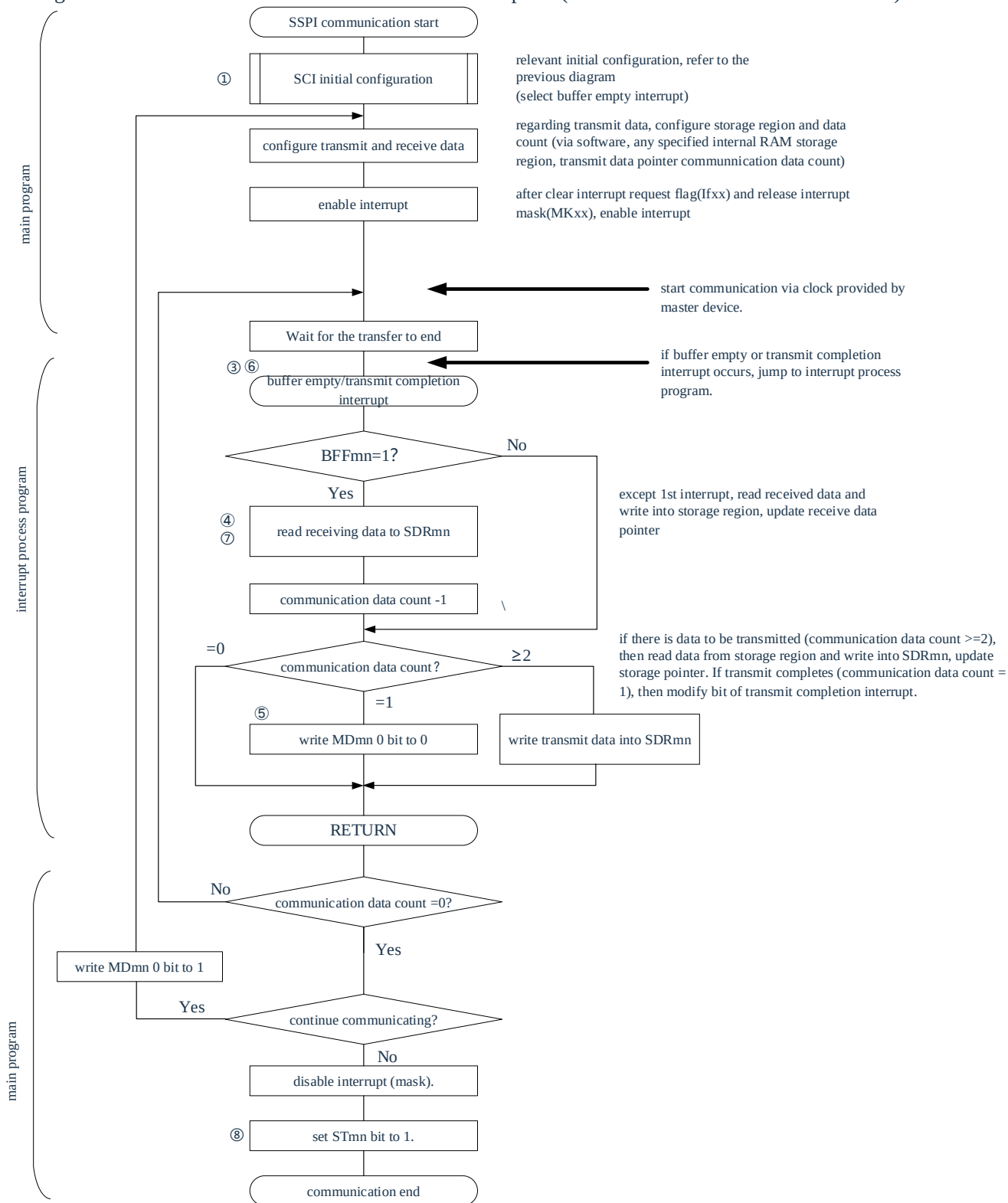
Caution: The MDmn0 bit of the serial mode register mn (SMRmn) can be rewritten even during operation. However, to ensure that the transmission end interrupt for the last transmitted data is triggered, the rewriting must be done before the last bit is transmitted.

Remark:

① to ⑧ in the diagram correspond to ① to ⑧ in “Figure 12-73 Flowchart of slave transmission and reception (continuous transmit and receive mode)”.

m: unit number (m=0) n: channel number (n=0)p: SSPI number (p=00).

Figure 12-73: Flowchart of slave transmission and reception (continuous transmit and receive mode)



Caution: Before the master device starts outputting the clock, the transmit data must be set in the SDRmn register.

Remark:

① to ⑧ correspond to ① to ⑧ in “Figure 12-72 Timing diagram of slave transmission and reception (continuous transmit and receive mode)”.

m: unit number (m=0) n: channel number (n=0)p: SSPI number (p=00).

12.6.4 Calculation of Transfer Clock Frequency

The transfer clock frequency of the slave select input function (SSPI00) communication can be calculated using the following calculation formula.

Slave device:

$$(\text{Transfer clock frequency}) = \{ \text{Serial clock (SCLK) frequency provided by the master device} \}^{\text{Note1}} [\text{Hz}]$$

Note 1: The maximum allowable transfer clock frequency is 6MHz.

m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00).

Table 12-33: Slave select input function operation clock selection

SMR _{mn} register	SPS _m register								Operation clock (F _{MCK}) ^{Note}	
CKS _{mn}	PRS _{m13}	PRS _{m12}	PRS _{m11}	PRS _{m10}	PRS _{m03}	PRS _{m02}	PRS _{m01}	PRS _{m00}		F _{CLK} =24MHz in operation
0	X	X	X	X	0	0	0	0	F _{CLK}	24MHz
	X	X	X	X	0	0	0	1	F _{CLK} /2	12MHz
	X	X	X	X	0	0	1	0	F _{CLK} /2 ²	6MHz
	X	X	X	X	0	0	1	1	F _{CLK} /2 ³	3MHz
	X	X	X	X	0	1	0	0	F _{CLK} /2 ⁴	1.5MHz
	X	X	X	X	0	1	0	1	F _{CLK} /2 ⁵	0.75MHz
	X	X	X	X	0	1	1	0	F _{CLK} /2 ⁶	375KHz
	X	X	X	X	0	1	1	1	F _{CLK} /2 ⁷	187.5KHz
	X	X	X	X	1	0	0	0	F _{CLK} /2 ⁸	93.75KHz
	X	X	X	X	1	0	0	1	F _{CLK} /2 ⁹	46.875KHz
	X	X	X	X	1	0	1	0	F _{CLK} /2 ¹⁰	23.44KHz
	X	X	X	X	1	0	1	1	F _{CLK} /2 ¹¹	11.72KHz
	X	X	X	X	1	1	0	0	F _{CLK} /2 ¹²	5.86KHz
	X	X	X	X	1	1	0	1	F _{CLK} /2 ¹³	2.93KHz
	X	X	X	X	1	1	1	0	F _{CLK} /2 ¹⁴	1.46KHz
	X	X	X	X	1	1	1	1	F _{CLK} /2 ¹⁵	732Hz

Note: To change the clock selected as F_{CLK} (change the value of the system clock control register (CKC)), you must stop the operation of the universal serial communication unit (SCI) (serial channel stop register m(STm)=000FH) after making the change.

X: Ignore

m: unit number (m=0) n: channel number (n=0).

12.6.5 Procedure for Handling Errors in Clock-Synchronous Serial Communication with Slave Select Input Function

The steps for handling errors in the clock-synchronous serial communication process with slave selection input function are shown in Figure 12-34.

Table 12-34: Handling steps when an overrun error occurs

Software operation	Hardware status	Comments
Read the serial data register mn (SDRmn). →	The BFFmn bit of the SSRm n register is "0" and channel n is in receivable status.	This is to prevent overrun errors from ending the next reception during mishandling.
Read the serial status register mn (SSRmn).		The type of error is judged, and the reading value is used to clear the error flag.
Set the serial flag clear trigger register mn to 1 →	Clear the error flag.	By writing the read value of the SSRmn register directly to the SDIRmn register, errors during read operations can only be cleared.

Remark: m: unit number (m=0) n: channel number (n=0).

12.7 Operation of UART0 Communication

This is the functionality of asynchronous communication using two lines: serial data transmission (TxD) and serial data reception (RxD). Using these two communication lines, data is sent and received asynchronously with other communication parties according to a data frame (composed of start bit, data, parity bit, and stop bit) using an internal baud rate. Full-duplex asynchronous UART communication can be achieved by using two dedicated channels: one for transmission (even channels 00/10) and one for reception (odd channels 01/11).

[Data transmission and reception]

- 7 or 8 bits data length
- MSB/LSB FIRST
- Level settings for transmitting and receiving data (choose whether the level is inverted)
- Parity bit appending and parity check functions
- Stop bit appending and stop bit detection functions

[Interrupt function]

- Transfer end interrupt, buffer empty interrupt
- Error interrupt caused by frame error, parity error, and overrun error

[Error detection flag]

- Frame error, parity error, overrun error

UART0 uses channels 0 and 1 of SCIO.

Each channel can be assigned any function, but no other functions can be used except the selected one.

For example, when UART0 is used on channel 0 and channel 1 of unit 0, SSPI00 and IIC00 cannot be used.

UART has the following two communication modes:

- UART Transmission (see Section 12.7.1)
- UART reception (see Section 12.7.2)

Caution: When used as UART, the transmitter (even channels) and receiver (odd channels) can only be used for UART.

12.7.1 UART Transmission

UART transmission is the operation where the microcontroller of this product asynchronously sends data to other devices.

The even channels of the two channels used by UART are for UART transmission.

Table 12-35: UART transmission

UART	UART0
Target channel	Channel of SCIO
Pin used	TxD0
Interrupt	INTST0 Transfer end interrupt (in single transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	None
Transfer data length	7 or 8 bits
Transfer rate	$\text{Max.F}_{\text{MCK}}/6[\text{bps}](\text{SDRmn}[15:9] \geq 3)$, $\text{Min.F}_{\text{CLK}}/(2 \times 2^{11} \times 128)[\text{bps}]^{\text{Note}}$
Data phase	Non-reverse output (default: high). Reverse output (default: low).
Parity bit	Select from the following: • No parity bits. • Append zero parity. • Append even parity. • Append odd parity.
Stop bit	Select from the following: • Append 1 bit. • Append 2 bits.
Data direction	MSB or LSB first

Caution: It must be used within the scope of peripheral functional characteristics that meet this condition and meet the electrical characteristics (see data sheet).

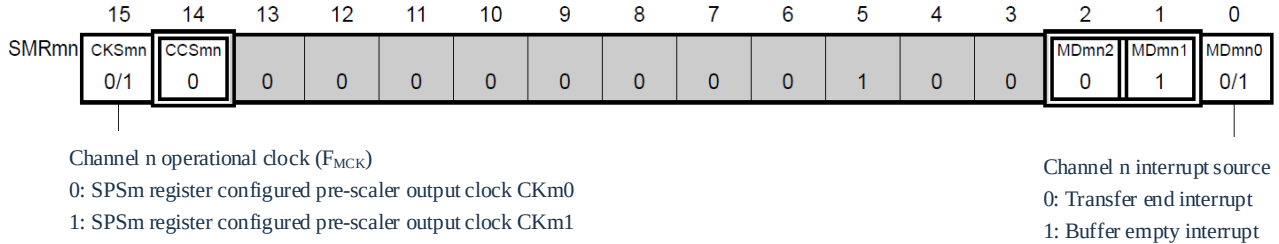
Remark:

1. F_{MCK} : Target channel operation clock frequency
2. F_{CLK} : system clock frequency
3. m: unit number (m=0) n: channel number (n=0).

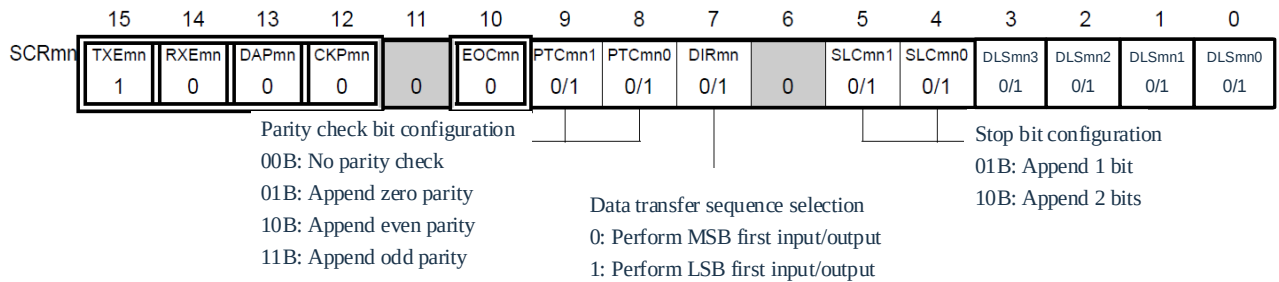
(1) Register setting

Figure 12-74: Example of register settings when UART is sent by UART (UART0) (1/2)

(a) Serial mode register mn (SMRmn)

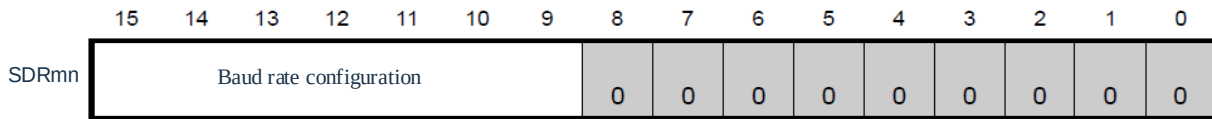


(b) Serial communication operation configuration register mn (SCRmn)

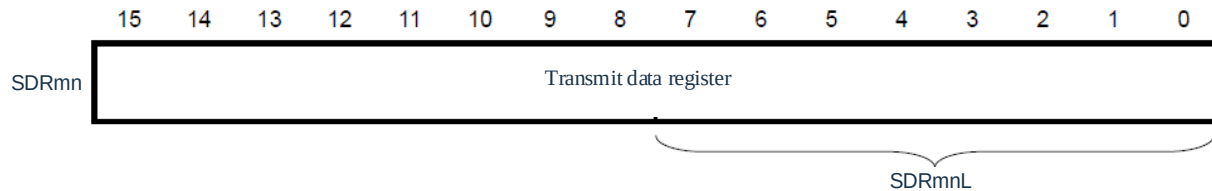


(c) Serial data register mn (SDRmn)

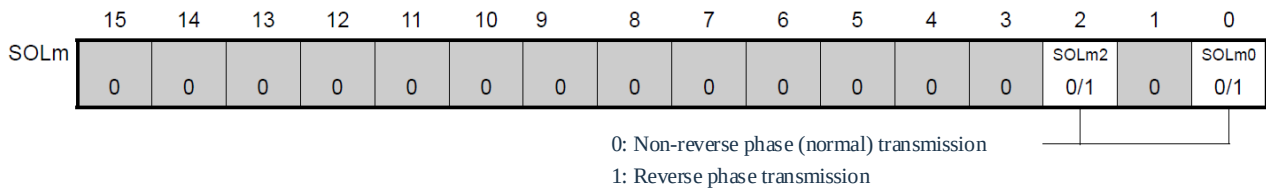
(1) Run stop (SEmn=0)



(2) During operation (low 8 bits: SDRmnL)



(d) Serial output voltage register m (SOLm) Only configure target channel bits.



Remark: m: unit number (m=0) n: channel number (n=0) q: UART number (q=0)

 : Fixed in SSPI master receive mode : Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set to its initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

Figure 12-74: Example of register settings when UART is sent by UART(UART0) (2/2)

(e) Serial output register m (SOM).... Only configure target channel bits

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM	0	0	0	0	0	0	CKOm1	CKOm0	0	0	0	0	0	0	SOM1	SOM0
	0	0	0	0	0	0	×	×	0	0	0	0	0	0	×	0/1 ^{Note}

0: Serial data output value=0
1: Serial data output value=1

(f) Serial output enable register m (SOEm).... Only set the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1

(g) Serial channel start register m (SSm) Only set the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1

Caution: When the SOLmn bit of the corresponding channel is “0”, it must be set to “1”, and when the SOLmn bit of the corresponding channel is “1”, it must be set to “0” before starting transmission. During communication, the value changes depending on the communication data.

Remark: m: unit number (m=0) n: channel number (n=0) q: UART number (q=0)

☐ : Fixed in SSPI master receive mode ☐ : Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set to its initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user’s purpose.

(2)Procedure

Figure 12-75: Initial setup steps for UART transmission

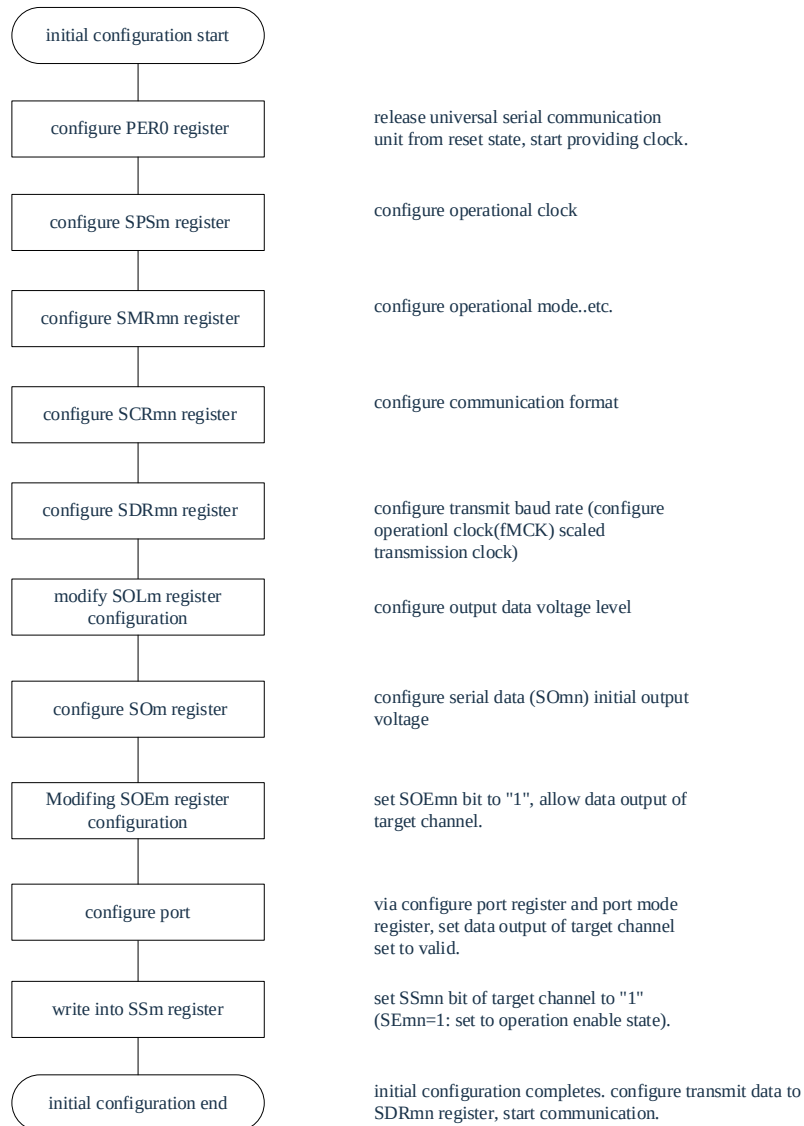


Figure 12-76: Stop steps for UART transmission

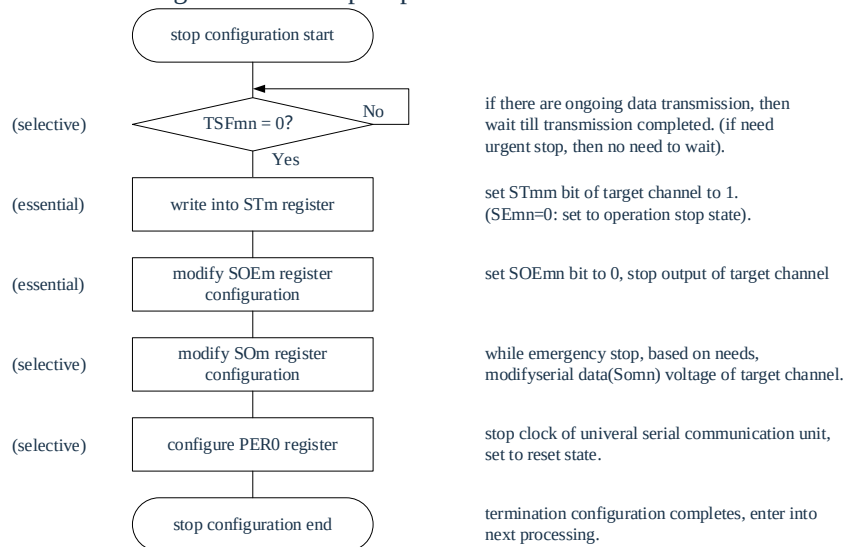
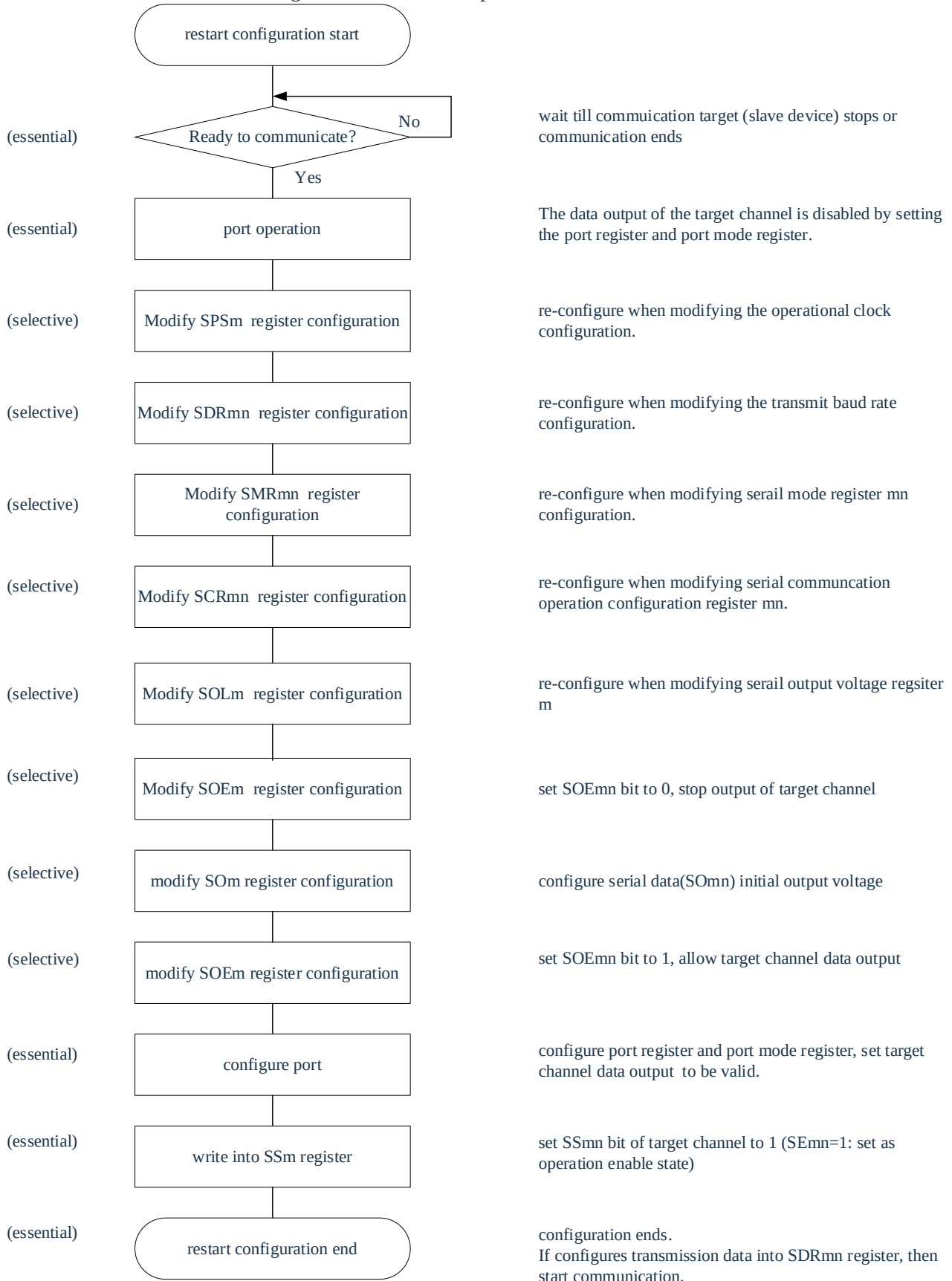


Figure 12-77: Restart steps for UART transmission



Caution: If you override PER0 in the abort setting to stop the clock, you must wait until the communication target stops or the communication ends, instead of starting the setting again.

(3) Processing flow (single transmit mode)

Figure 12-78: Timing diagram of UART transmission (single transmit mode)

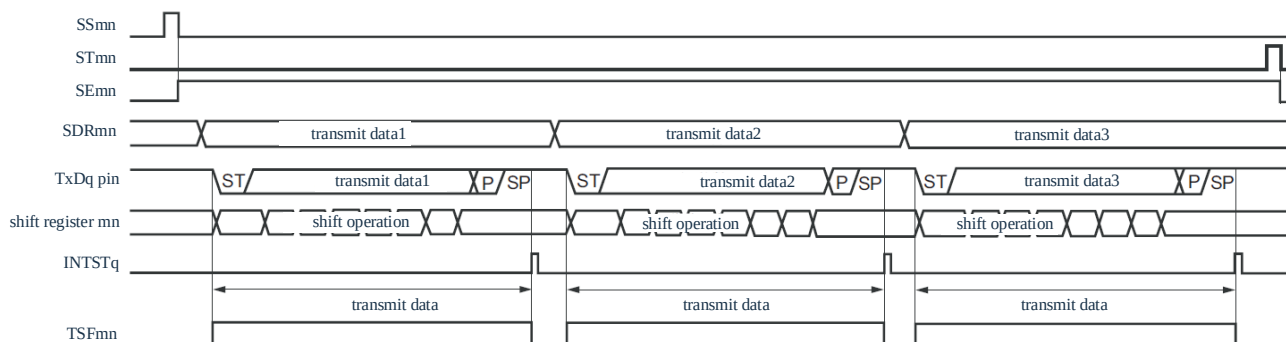
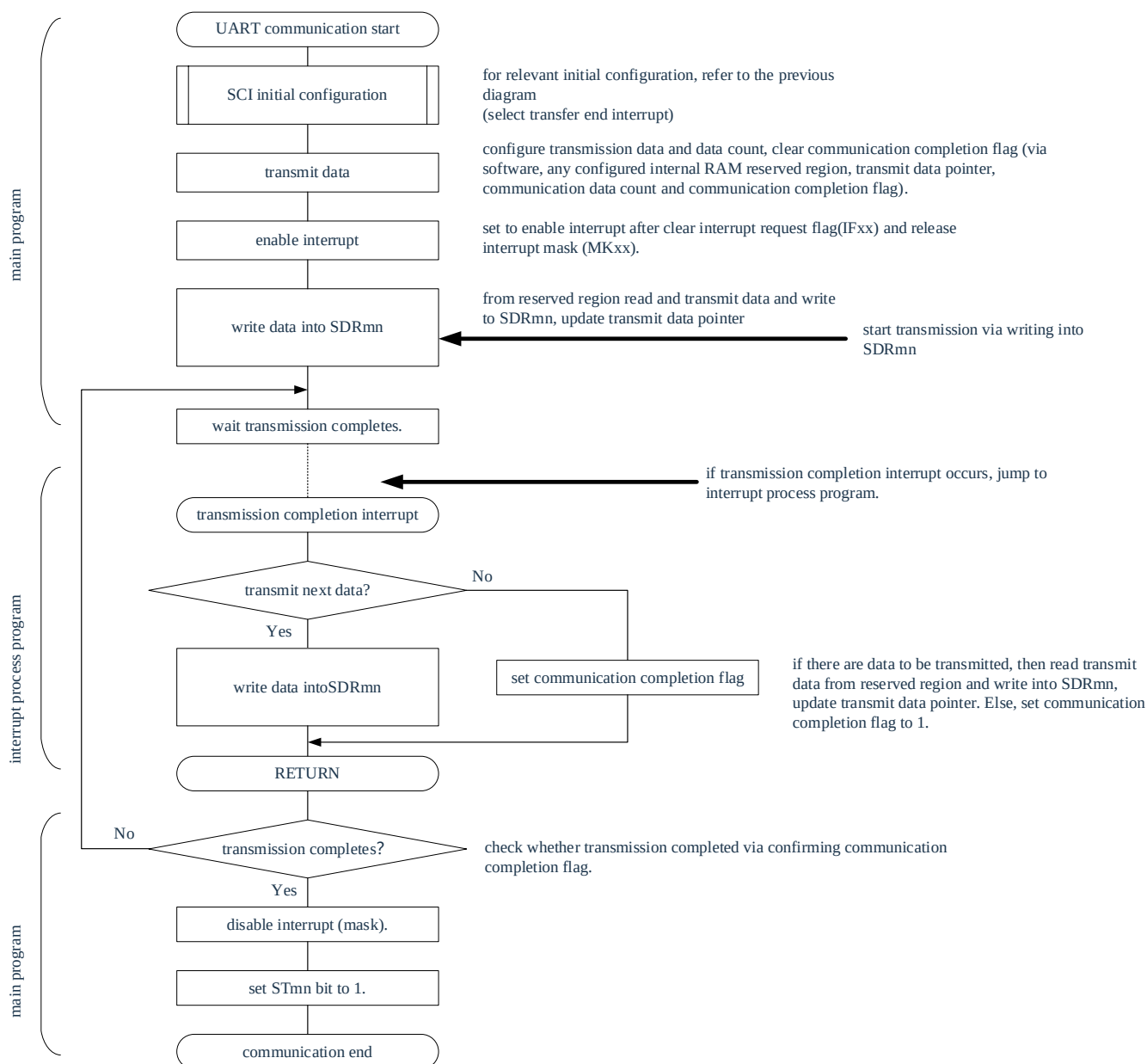


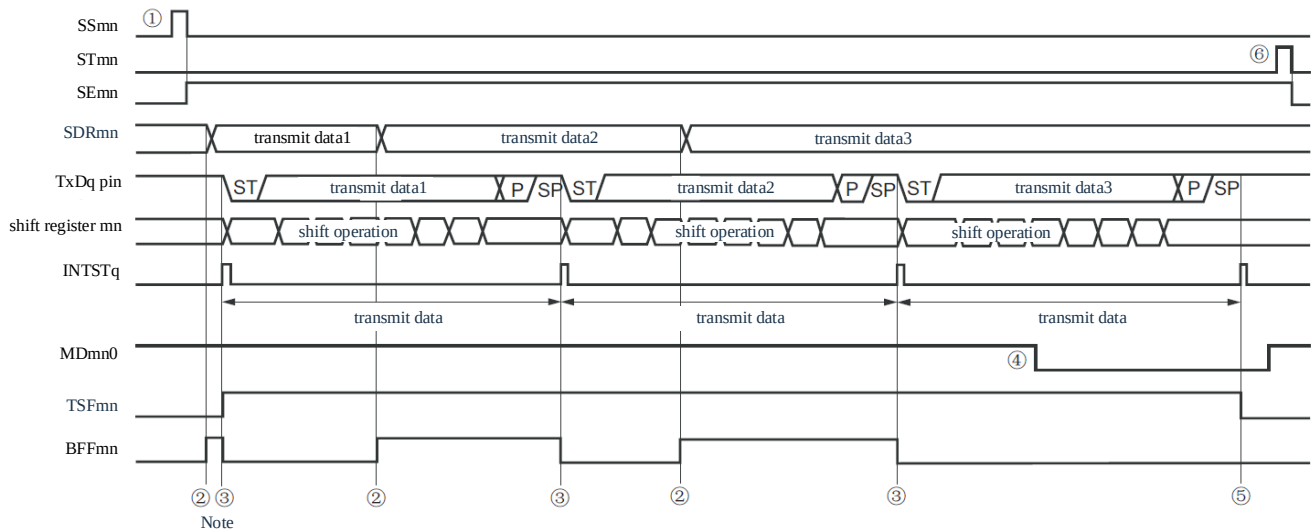
Figure 12-79: Flowchart of UART transmission (single transmit mode)



Remark: m: unit number (m=0) n: channel number (n=0,) q: UART number (q=0).

(4) Processing flow (continuous transmit mode)

Figure 12-80: Timing diagram of UART transmission (continuous transmit mode)



Note: If data is written to the SDRmn register while the BFFmn bit of the serial status register mn (SSRmn) is “1” (when valid data is stored in the serial data register mn (SDRmn)), the transmit data will be overwritten.

Caution: The MDmn0 bit of the serial mode register mn (SMRmn) can be rewritten even during operation. However, to ensure that the transmission end interrupt for the last transmitted data is triggered, the rewriting must be done before the last bit is transmitted.

Remark: m: unit number (m=0) n: channel number (n=0) q: UART number (q=0).

```

graph TD
    Start([UART communication start]) --> Init[SCI initial configuration]
    Init --> Config[configure transmit data]
    Config --> Enable[enable interrupt]
    Enable --> WriteSDRmn1[write data into SDRmn]
    WriteSDRmn1 --> Wait[wait transmission completes.]
    Wait --> Interrupt([buffer empty/transmit completion interrupt])
    Interrupt --> Count{communication data count > 0}
    Count -- Yes --> WriteSDRmn2[write data into SDRmn]
    WriteSDRmn2 --> DecCount[communication data count-1]
    DecCount --> Return([RETURN])
    Count -- No --> MDmn{MDmn=1?}
    MDmn -- Yes --> WriteMDmn[write MDmn 0 bit to 0]
    MDmn -- No --> SetFlag[set communication completion flag]
    WriteMDmn --> Comp{transmission completes?}
    SetFlag --> Comp
    Return --> Comp
    Comp -- No --> SetSTmn1[set STmn bit to 1]
    SetSTmn1 --> End([communication end])
    Comp -- Yes --> Continue{continue communicating?}
    Continue --> Disable[disable interrupt (mask).]
    Disable --> SetSTmn2[set STmn bit to 1.]
    SetSTmn2 --> End
  
```

① SCI initial configuration

configure transmit data

enable interrupt

② write data into SDRmn

wait transmission completes.

③ buffer empty/transmit completion interrupt

communication data count > 0

Yes

② write data into SDRmn

communication data count-1

RETURN

No

MDmn=1?

Yes ④ write MDmn 0 bit to 0

No ⑤ set communication completion flag

transmission completes?

Yes

continue communicating?

disable interrupt (mask).

⑥ set STmn bit to 1.

communication end

No

set STmn bit to 1

check whether transmission completed via confirming communication completion flag.

from reserved region read and transmit data and write to SDRmn, update transmit data pointer

start transmission via writing into SDRmn

if transmission completion interrupt occurs, jump to interrupt process program.

if there is transmit data, then read transmit data from storage region and write into SDRmn, update transmit data pointer and transmit data count. If no data to transmit, then clear MDmn bit while MDmn bit is '1'. Else, complete transmission.

for relevant initial configuration, refer to the previous diagram (select transfer end interrupt)

configure transmission data and data count, clear communication completion flag (via software, any configured internal RAM reserved region, transmit data pointer, communication data count and communication completion flag).

set to enable interrupt after clear interrupt request flag(1Fxx) and release interrupt mask (MKxx).

Remark: ① to ⑥ in the figure correspond to ① to ⑥ in “Figure 12-80 Timing diagram of UART transmission (continuous transmit mode)”.

12.7.2 UART Reception

UART reception refers to the operation where the microcontroller of this product asynchronously receives data from other devices.

The odd channel of the two channels used by UART is dedicated to UART reception. However, the SMR registers for both the odd and even channels need to be configured.

Table 12-36: UART reception

UART	UART0
Target channel	Channel 1 of SCIO
Pin used	RxD0
Interrupt	INTSR0 Limited to transfer end interrupts (disable setting buffer empty interrupts).
Error interrupt	INTSRE0
Error detection flag	• Frame error detection flag (FEFmn). • Parity error detection flag (PEFmn). • Overrun error detection flag (OVFmn).
Transfer data length	7 or 8 bits
Transfer rate	$\text{Max.F}_{\text{MCK}}/6[\text{bps}](\text{SDRmn}[15:9] \geq 2)$, $\text{Min.F}_{\text{CLK}}/(2 \times 2^{15} \times 128)[\text{bps}]$
Data phase	Non-reverse output (default: high). Reverse output (default: low).
Parity bit	Select from the following: • No parity bits (no parity). • Append zero parity (no parity). • Even parity • Odd parity
Stop bit	Append 1 bit.
Data direction	MSB or LSB first

Caution: It must be used within the scope of peripheral functional characteristics that meet this condition and meet the electrical characteristics (see data sheet).

Remark:

F_{MCK} : Target channel operation clock frequency

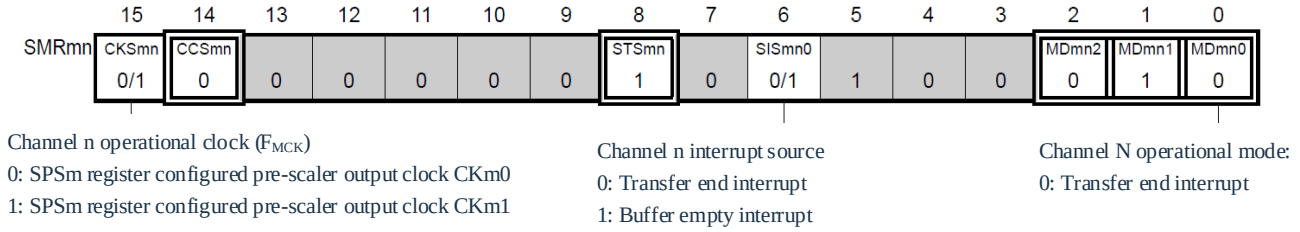
F_{CLK} : system clock frequency

m: unit number (m=0) n: channel number (n=1, 3).

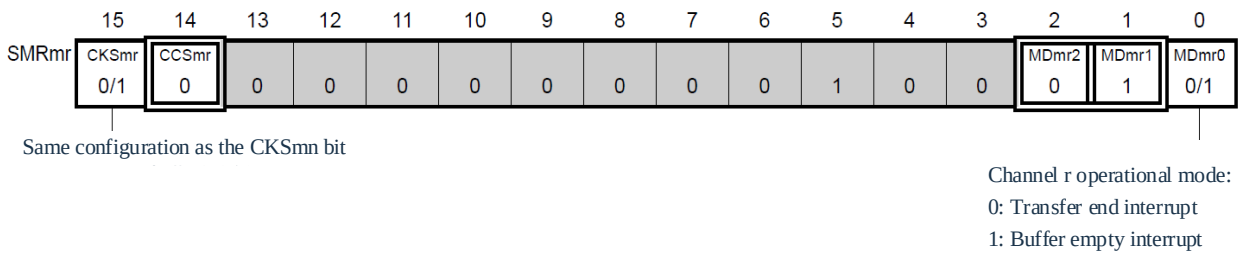
(1) Register setting

Figure 12-81: Example of register settings when UART is received by UART (UART0) (1/2)

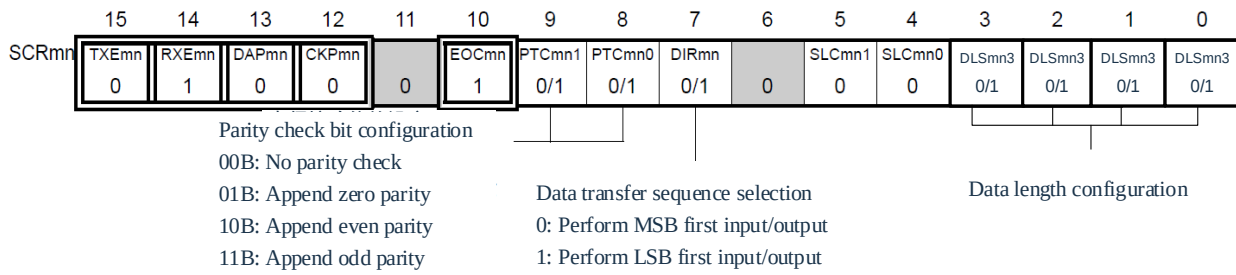
(a) Serial mode register mn (SMRmn)



(b) Serial mode register mr (SMRmr)

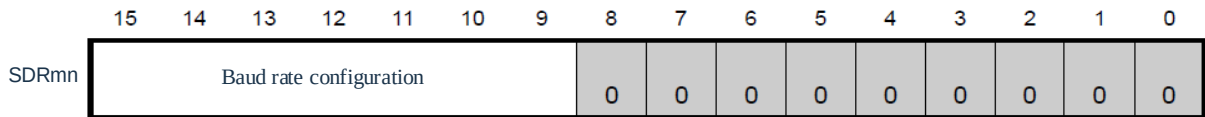


(c) Serial communication operation configuration register mn (SCRmn)

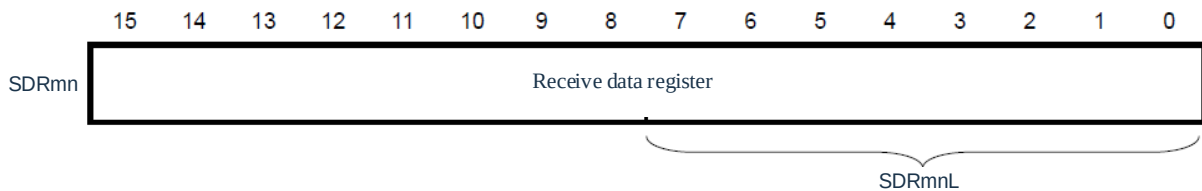


(c) Serial data register mn (SDRmn)

(1) Run stop (SEmn=0)



(2) During operation (SEmn=1) (low 8 bits: SDRmnL)



Caution: When receiving with UART, the SMRmr register of channel r paired with channel n must also be configured.

Remark: m: unit number (m=0); n: channel number (n=1, 3);

r: channel number (r=n-1); q: UART number (q=0~5).

 : Fixed in SSPI master receive mode : Cannot set (set initial value);

×: This is the bit that cannot be used in this mode (set to its initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

Figure 12-82: Example of register settings when UART is received by UART (UART0) (2/2)

(e) Serial output register m (SOM)... Not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM							CKOm1	CKOm0							SOM1	SOM0
	0	0	0	0	0	0	×	×	0	0	0	0	0	0	×	×

(f) Serial output enable register m (SOEm)... Not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm															SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	×	×

(g) Serial channel start register m (SSm) Only set the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm															SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×

Caution: During UART reception, the SMRmr register of the channel r paired with channel n must also be configured.

Remark: m: unit number (m=0); n: channel number (n=1);

r: channel number (r=n-1); q: UART number (q=0).

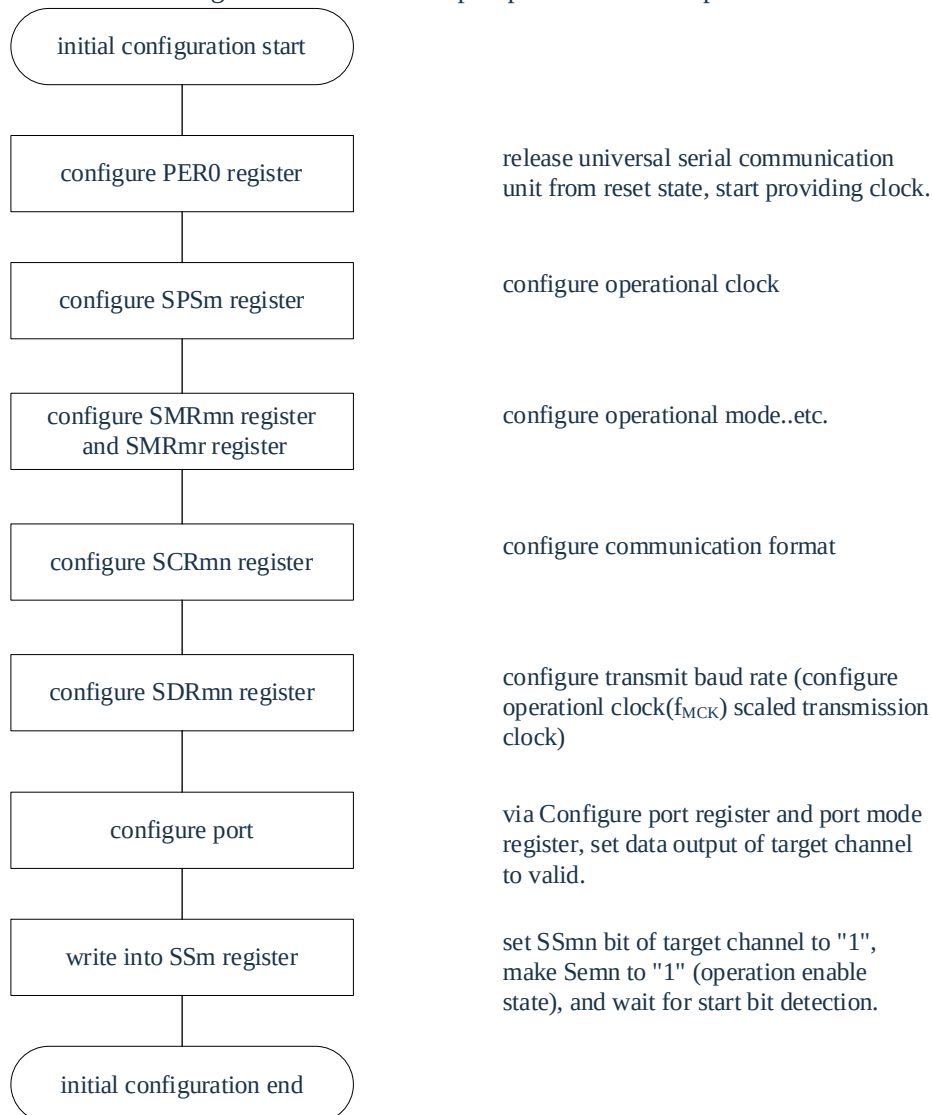
☐ : Fixed in UART receive mode; ☐ : Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set to its initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

(2) Procedure

Figure 12-83: Initial setup steps for UART reception



Caution: At least 4 F_{MCK} clocks must be spaced after setting the RXEmn bit of the SCRmn register to 1 and then set the SSmn bit to "1".

Figure 12-84: Stop steps for UART reception

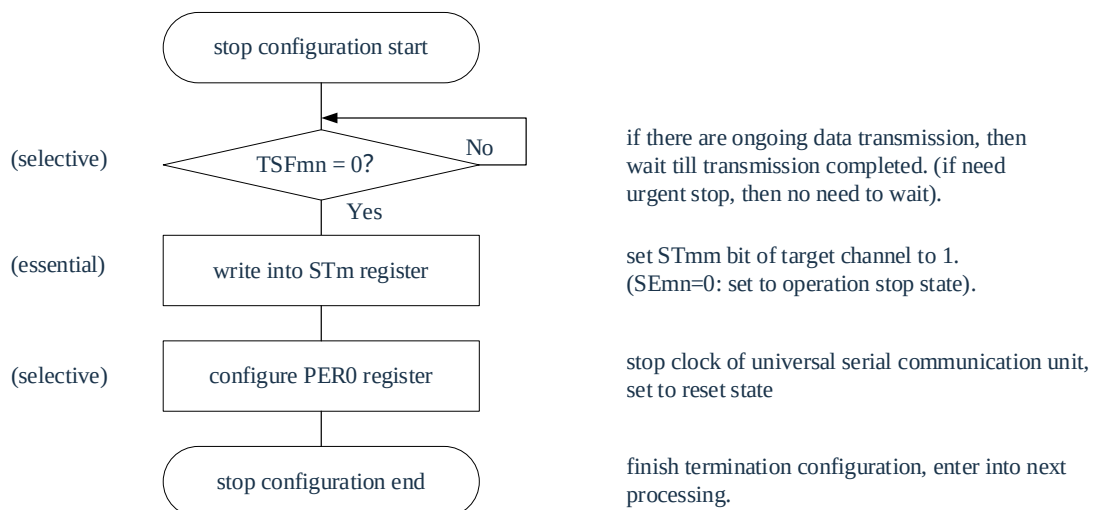
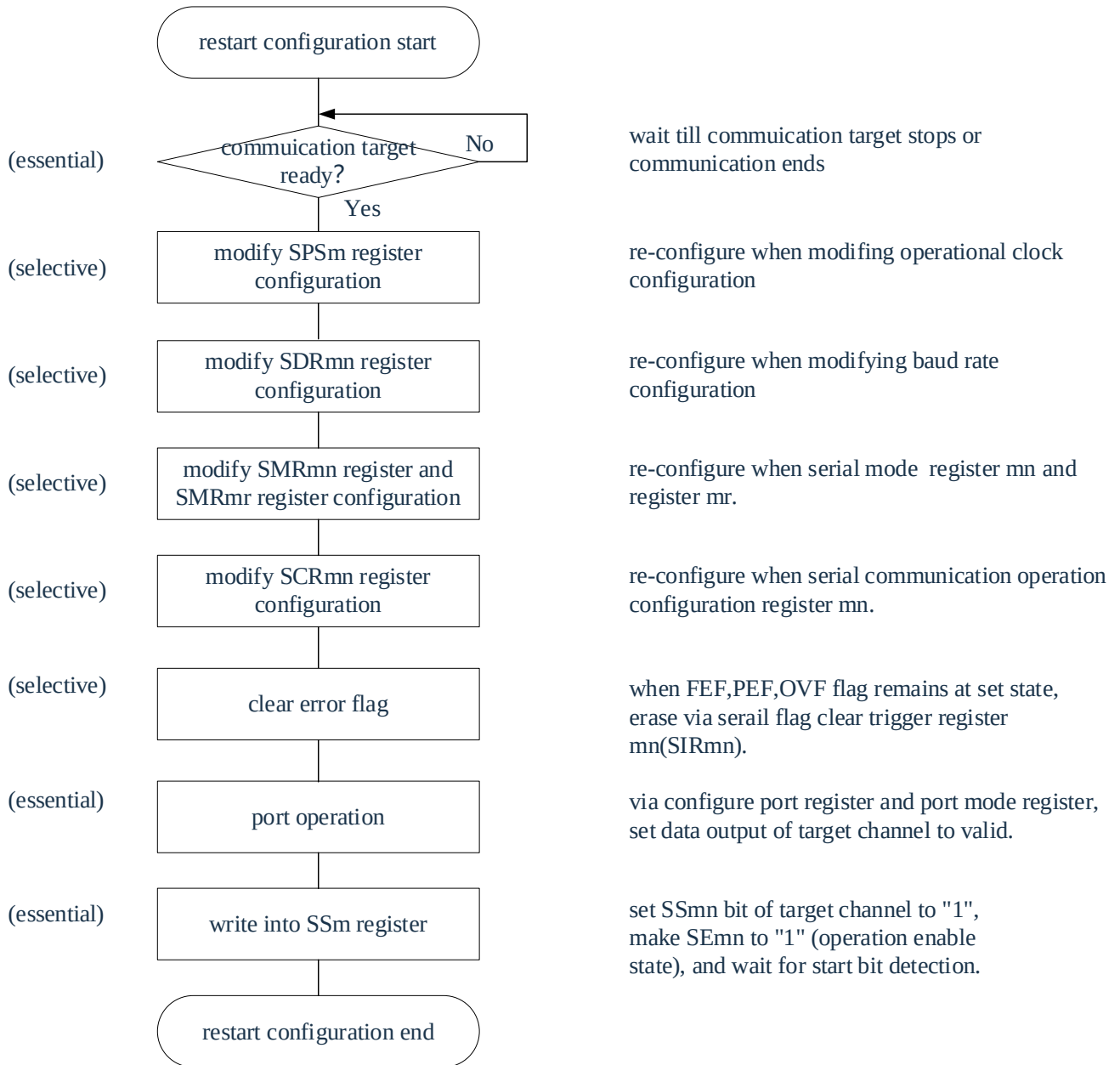


Figure 12-85: Restart steps for UART reception

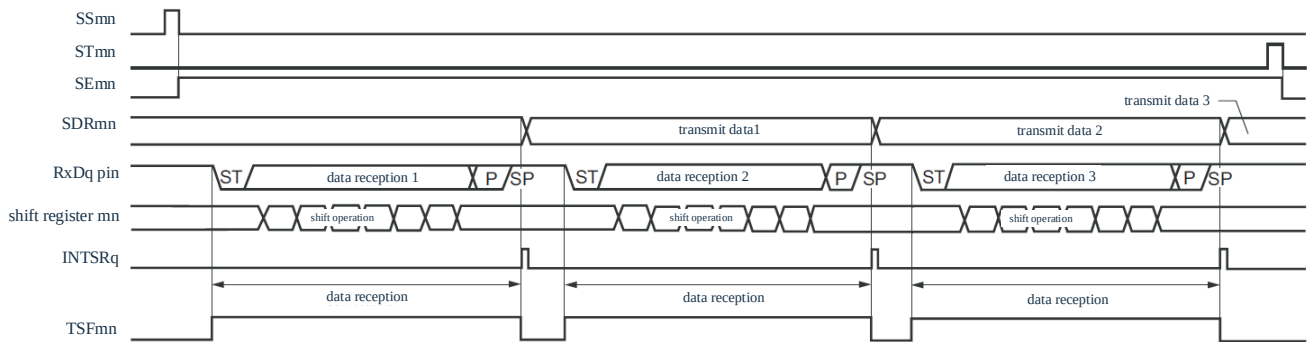

Caution:

At least 4 F_{MCK} clocks must be spaced after setting the RXEmn bit of the SCRmn register to "1" and then set the SSmn bit to "1".

If you override PER0 in the abort setting to stop the clock, you must wait until the communication target stops or the communication ends, instead of starting the setting again.

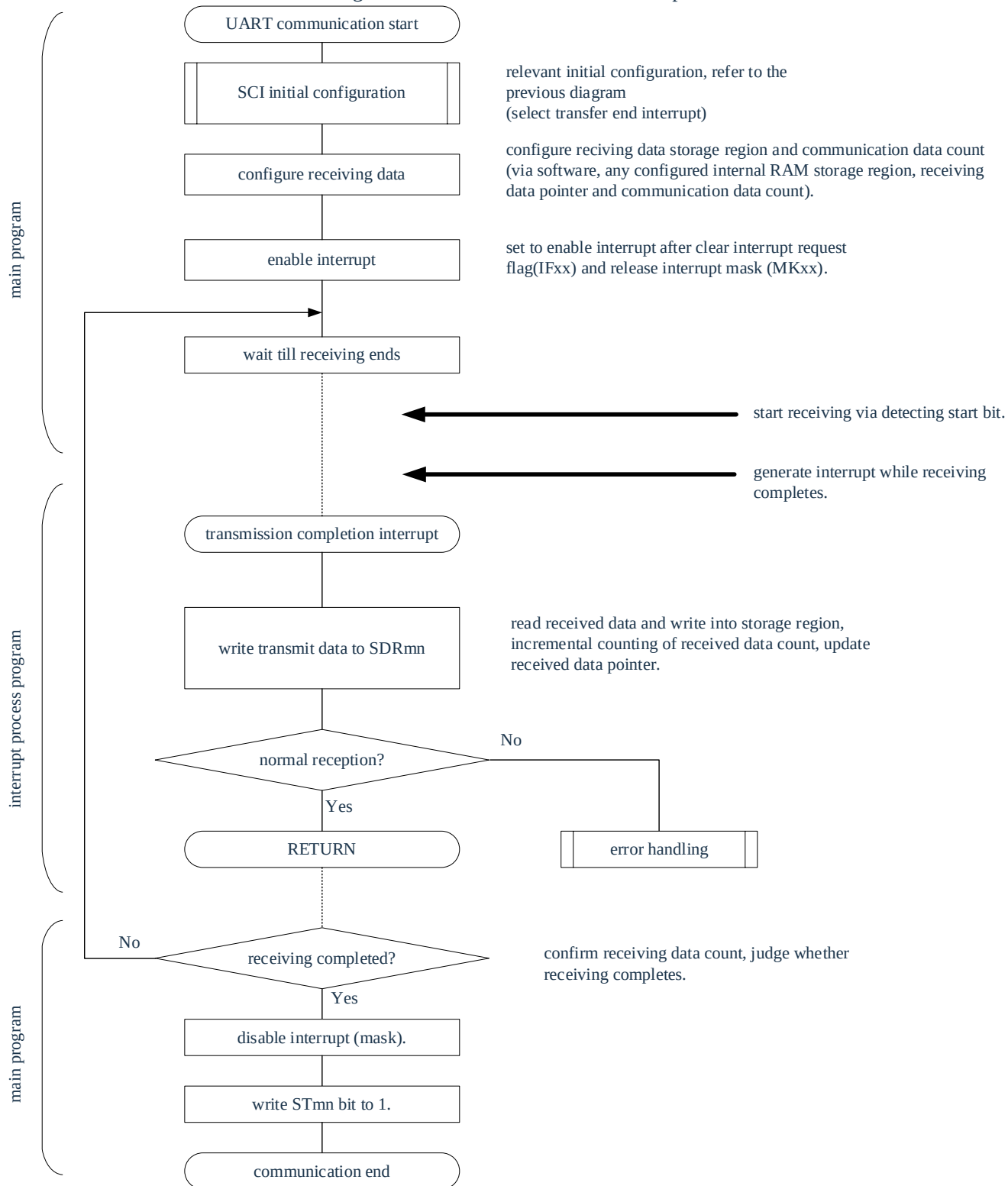
(3) Processing flow

Figure 12-86: Timing diagram of UART reception



Remark: m: unit number (m=0); n: channel number (n=1);
r: channel number (r=n-1); q: UART number (q=0).

Figure 12-87: Flowchart of UART reception



12.7.3 Calculation of Baud Rate

Calculating the baud rate

The baud rate of UART0 communication can be calculated using the following formula:

$$(\text{Baud rate}) = \{\text{Clock frequency of the target channel (f}_{\text{MCK}})\} / (\text{SDRmn}[15:9] + 1) / 2 [\text{bps}]$$

The operating clock (F_{MCK}) depends on the bit 15 (CKSmn bit) of the serial clock selection register m (SPSm) and the serial mode register mn (SMRmn).

Caution: It is prohibited to set the serial data register mn (SDRmn) bits [15:9] to “0000000B” and “0000001B”.

Remark: When using UART, the value of SDRmn[15:9] is the value of bits 15–9 of the SDRmn register (0000010B–1111111B), so it ranges from 2 to 127.

m: unit number (m=0) n: channel number (n=0, 1).

Table 12-37: Selection of UART operating clock

SMRmn register	SPSm register								Operation clock (F _{MCK}) ^{Note}	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		F _{CLK} =24MHz in operation
0	X	X	X	X	0	0	0	0	F _{CLK}	24MHz
	X	X	X	X	0	0	0	1	F _{CLK} /2	12MHz
	X	X	X	X	0	0	1	0	F _{CLK} /2 ²	6MHz
	X	X	X	X	0	0	1	1	F _{CLK} /2 ³	3MHz
	X	X	X	X	0	1	0	0	F _{CLK} /2 ⁴	1.5MHz
	X	X	X	X	0	1	0	1	F _{CLK} /2 ⁵	0.75MHz
	X	X	X	X	0	1	1	0	F _{CLK} /2 ⁶	375KHz
	X	X	X	X	0	1	1	1	F _{CLK} /2 ⁷	187.5KHz
	X	X	X	X	1	0	0	0	F _{CLK} /2 ⁸	93.75KHz
	X	X	X	X	1	0	0	1	F _{CLK} /2 ⁹	46.875KHz
	X	X	X	X	1	0	1	0	F _{CLK} /2 ¹⁰	23.44KHz
	X	X	X	X	1	0	1	1	F _{CLK} /2 ¹¹	11.72KHz
	X	X	X	X	1	1	0	0	F _{CLK} /2 ¹²	5.86KHz
	X	X	X	X	1	1	0	1	F _{CLK} /2 ¹³	2.93KHz
	X	X	X	X	1	1	1	0	F _{CLK} /2 ¹⁴	1.46KHz
	X	X	X	X	1	1	1	1	F _{CLK} /2 ¹⁵	732Hz
1	0	0	0	0	X	X	X	X	F _{CLK}	24MHz
	0	0	0	1	X	X	X	X	F _{CLK} /2	12MHz
	0	0	1	0	X	X	X	X	F _{CLK} /2 ²	6MHz
	0	0	1	1	X	X	X	X	F _{CLK} /2 ³	3MHz
	0	1	0	0	X	X	X	X	F _{CLK} /2 ⁴	1.5MHz
	0	1	0	1	X	X	X	X	F _{CLK} /2 ⁵	0.75MHz
	0	1	1	0	X	X	X	X	F _{CLK} /2 ⁶	375KHz
	0	1	1	1	X	X	X	X	F _{CLK} /2 ⁷	187.5KHz
	1	0	0	0	X	X	X	X	F _{CLK} /2 ⁸	93.75KHz
	1	0	0	1	X	X	X	X	F _{CLK} /2 ⁹	46.875KHz
	1	0	1	0	X	X	X	X	F _{CLK} /2 ¹⁰	23.44KHz
	1	0	1	1	X	X	X	X	F _{CLK} /2 ¹¹	11.72KHz
	1	1	0	0	X	X	X	X	F _{CLK} /2 ¹²	5.86KHz
	1	1	0	1	X	X	X	X	F _{CLK} /2 ¹³	2.93KHz
	1	1	1	0	X	X	X	X	F _{CLK} /2 ¹⁴	1.46KHz
	1	1	1	1	X	X	X	X	F _{CLK} /2 ¹⁵	732Hz

Note: To change the clock selected as F_{CLK} (change the value of the system clock control register (CKC)), you must stop the operation of the universal serial communication unit (SCI) (serial channel stop register m(STm)=000FH) after making the change.

X: Ignore

m: unit number (m=0) n: channel number (n=0, 1).

Baud rate error during transmission

The baud rate error during UART0 communication transmission can be calculated using the following formula. The transmitter's baud rate must be set within the allowable range of the receiver's baud rate.

$$(\text{Baud rate error}) = (\text{calculated value of baud rate}) / (\text{value of target baud rate}) \times 100 - 100[\%]$$

An example of setting the UART baud rate at $F_{CLK}=24\text{MHz}$ is shown below.

UART baud rate (Target baud rate)	$F_{CLK}=24\text{MHz}$			
	Operation clock (F_{MCK})	SDRmn[15:9]	Calculated value of the baud rate	Error with target baud rate
300bps	$F_{CLK}/2^9$	77	300.48bps	+0.16%
600bps	$F_{CLK}/2^8$	77	600.96bps	+0.16%
1200bps	$F_{CLK}/2^7$	77	1201.92bps	+0.16%
2400bps	$F_{CLK}/2^6$	77	2403.85bps	+0.16%
4800bps	$F_{CLK}/2^5$	77	4807.69bps	+0.16%
9600bps	$F_{CLK}/2^4$	77	9615.38bps	+0.16%
19200bps	$F_{CLK}/2^3$	77	19230.8bps	+0.16%
31250bps	$F_{CLK}/2^3$	47	31250.0bps	±0.0%
38400bps	$F_{CLK}/2^2$	77	38461.5bps	+0.16%
76800bps	$F_{CLK}/2$	77	76923.1bps	+0.16%
153600bps	F_{CLK}	77	153846bps	+0.16%
312500bps	F_{CLK}	37	315789bps	±1.05%

Remark: m: unit number (m=0) n: channel number (n=0, 1).

Baud rate tolerance range for reception

The baud rate tolerance range for UART0 communication can be calculated using the following formula, and the baud rate of the transmitter must be set within the baud rate tolerance range of the receiver.

$$\text{(Max. baud rate that can be received)} = \frac{2 \times k \times \text{Nfr}}{2 \times k \times \text{Nfr} - k + 2} \times \text{Brate}$$

$$\text{(Min. baud rate that can be received)} = \frac{2 \times k \times (\text{Nfr} - 1)}{2 \times k \times \text{Nfr} - k - 2} \times \text{Brate}$$

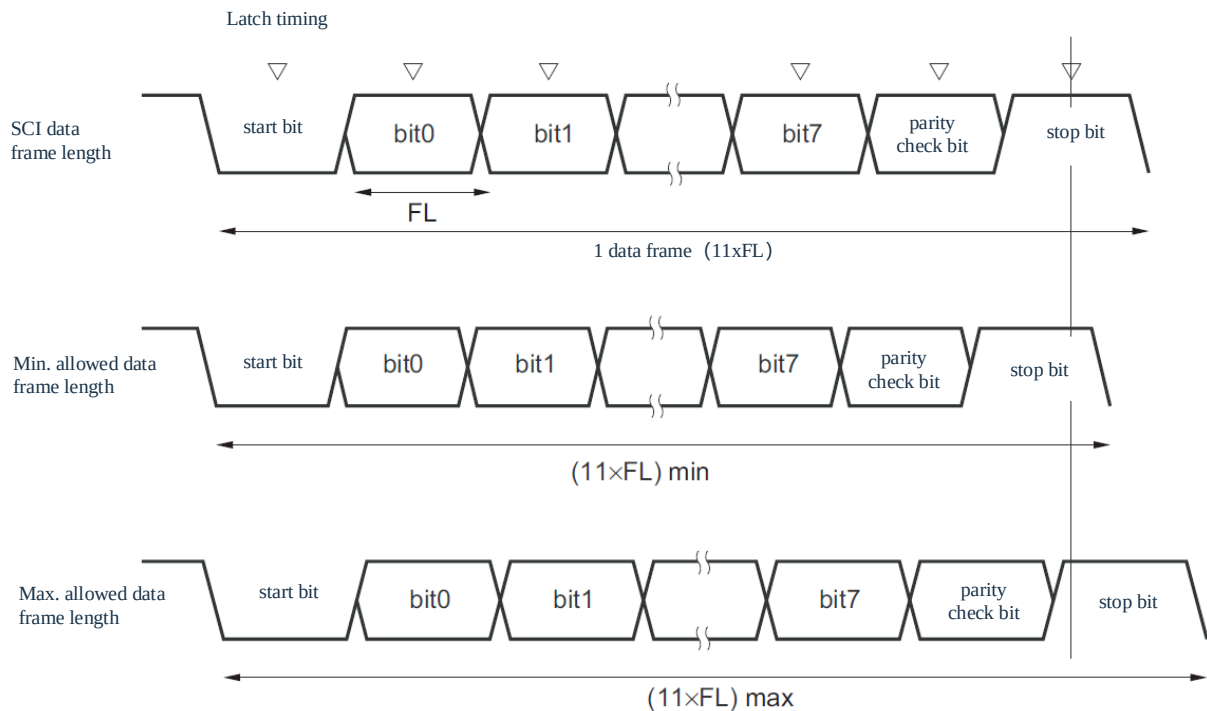
Brate: Calculated baud rate of the receiver (see 12.7.4 Baud rate calculation formula).

K: SDRmn[15:9]+1.

Nfr: 1 frame length of data [bit]

= (start bit) + (data length) + (parity bit) + (stop bit)

Figure 12-88: Baud rate tolerance range for reception (1 data frame length = 11 bits)



As shown in Figure 12-88 after the start bit is detected, the latch timing of the received data depends on the divider ratio set by bit15 to 9 of the serial data register mn (SDRmn). If the last data (stop bit) can catch up with this latch timing, it can be received normally.

Remark: m: unit number (m=0); n: channel number (n=0, 1).

12.7.4 Handling Steps When an Error Occurs During UART0 Communication

The handling steps when an error occurs during UART0 communication are shown in Figure 12-38 and Figure 13-39.

Table 12-38: Processing steps when a parity error or overrun error occurs

Software operation	Hardware status	Remark
Read the serial data register mn (SDRmn).	The BFFmn bit of the SSRmn register is “0” and channel n is in receivable status.	This is to prevent overrun errors from occurring when the next receive ends during error handling.
Read the serial status register mn (SSRmn).		Determine the type of error, and read the value to clear the error marker.
Set the serial flag clear trigger register mn (SDIRmn) to “1”.	Clear the error flag.	By writing the read value of the SSRmn register directly to the SDIRmn register, errors during read operations can only be cleared.

Table 12-39: Processing steps when a frame error occurs

Software operation	Hardware status	Remark
Read the serial data register mn(SDRmn).	The BFFmn bit of the SSRmn register is “0” and channel n is in receivable status.	This is to prevent overrun errors from ending the next reception during mishandling.
Read the serial status register mn(SSRmn).		Determine the error category, and read the value to remove the error marker.
Write the serial flag to clear the trigger register mn (SIRmn).	Clear the error flag.	By writing the read value of the SSRmn register directly to the SDIRmn register, errors during read operations can only be cleared.
Set the STmn bit of the serial channel stop register m (STm) to “1”.	The SEMn bit of the serial channel enable status register m (SEm) is 0 and channel n in run stop state.	
Synchronize processing with the communicating party.		Because the start bit is offset, a frame error can be considered to have occurred. Therefore, it is necessary to re-synchronize with the communicating party and restart the communication.
Set the SSmn bit of the serial channel start register m (SSm) to “1”.	The SEMn bit of the serial channel enable status register m (SEm) is 1 and channel n in operational state.	

Remark: m: unit number (m=0) n: channel number (n=0, 1).

12.8 Simplified I²C (IIC00) Communication Operation

This function uses two lines, the Serial Clock (SCL) and Serial Data (SDA), for clock-synchronized communication with multiple devices. Since this simplified I²C is designed for single communication with devices such as EEPROMs, flash memory, and A/D converters, it is used only as a master device.

For the start condition and stop condition, AC specifications must be followed, and software processing is required while operating the control registers.

[Data transmission and reception]

- Master transmission, master reception (limited to single master control functions).
- ACK output function ^{Note}, ACK detection function
- 8 bits of data length (when sending the address, specify the address with a high 7 bits, and use the lowest bit for R/W control).
- Generate start conditions and stop conditions through the software.

[Interrupt function]

- Transfer end interruption

[Error detection flag]

- ACK error

[Features not supported by Simplified I²C]

- Slave transmission, slave reception
- Multi-master function (arbitration failure detection function).
- Wait for detection function

Channel of SCI0 supports simplified I²C(IIC00) channels.

Simplified I²C(IIC00) operates in the following 4 communication modes:

- Address field transmission (see 12.8.1)
- Data transmission (see 12.8.2)
- Data reception (see 12.8.3)
- Generation of stop conditions (see 12.8.4)

Caution: When receiving the last piece of data, if 0 is written to the SDOEmn bit (in the SDOEm register) to stop the serial communication data output, ACK will not be output. For detailed information, refer to 12.8.3 (2) Processing Flow.

Remark: m: unit number (m=0) n: channel number (n=0, 1).

12.8.1 Address Field Transmission

Address field transmission is the first transmission operation to specifically specify the transmitting target (slave device) that is the first to occur during I²C communication. After generating the start condition, the address (7 bits) and the transmission direction (1 bit) are transmitted as 1 frame.

Table 12-42: Address field transmission

Simplified I ² C	IIC00
Target channel	Channel of SCI0
Pin used	SCLKOI00, SAU_SDA00 ^{Note 1}
Interrupt	INTIIC00
	Limited to transmit complete interrupts (buffer empty interrupts cannot be selected)
Error detection flag	ACK error detection flag (PEFmn)
Transfer data length	8 bits (transmit the highest 7 bits as the address and the lower 1 bit as R/W control).
Transfer rate ^{Note2}	Max.F _{MCK} /4[Hz](SDRmn[15:9]≥1)F _{MCK} : The operation clock frequency of the target channel must meet the following conditions in each mode of I ² C: • Max.1MHz (enhanced fast mode). • Max.400kHz (fast mode). • Max.100kHz (standard mode).
Data level	Non-reverse output (default: high).
Parity bits	No parity bits.
Stop bit	Append 1 bit (for ACK reception).
Data direction	MSB first

Note 1: To communicate via simplified I²C, the N-channel open-drain output mode must be set through the Port Output Mode Register (POMxx) (POMxx=1). For detailed information, refer to Chapter 2: Pin Functions.

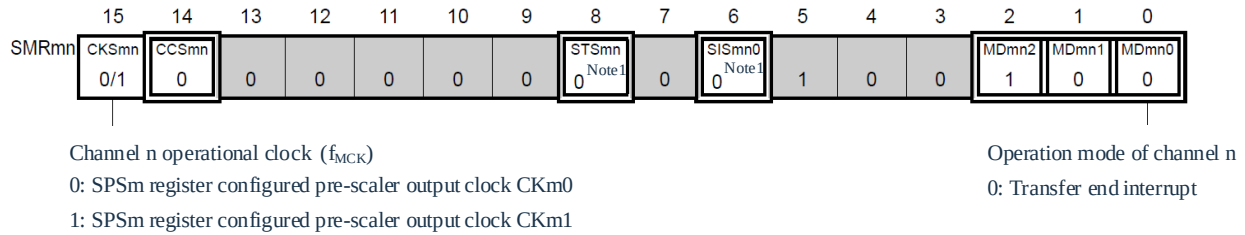
Caution: It must be used within the scope of peripheral functional characteristics that meet this condition and meet the electrical characteristics (see data sheet).

Remark: m: unit number (m=0) n: channel number (n=0, 1)

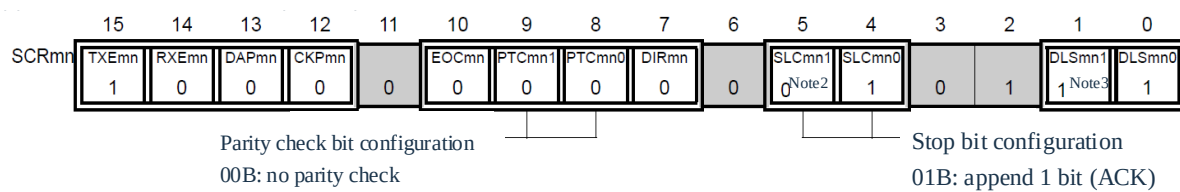
① Register setting

Figure 12-94: Example of register setting contents when transmitting address field of simplified I²C (IIC00)

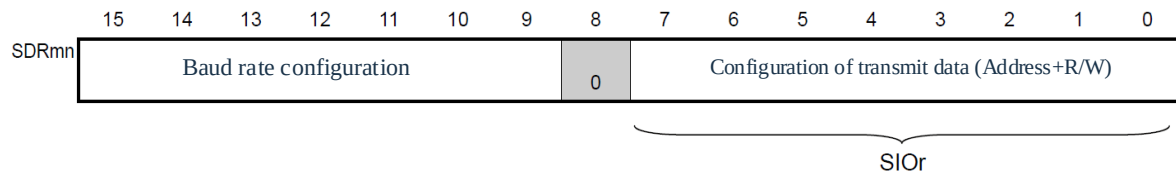
(a) Serial mode register mn (SMRmn)



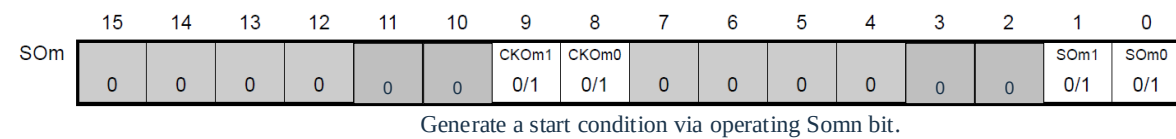
(b) Serial communication operation configuration register mn (SCRmn)



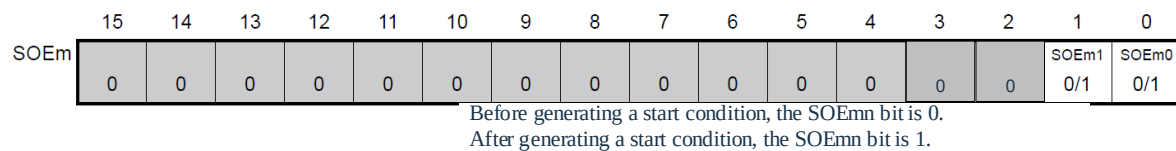
(c) serial data register mn (SDRmn) (low 8 bit: SIO_r)



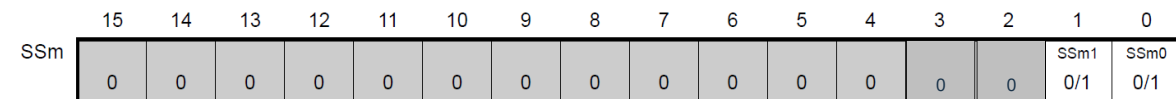
(d) Serial output register m (SOM)



(e) Serial output enable register m (SOEm)



(f) Serial channel start register m (SSm) Only set the bit of the target channel to 1.



Note 1: Limited to SMR01.

Note 2: Limited to SCR00.

Note 3: Limited to SCR00 register and SCR01 register, other fixed as “1”.

m: unit number (m=0, 1); n: channel number (n=0, 1); r: IIC number (r=00)

 : Fixed in IIC mode;

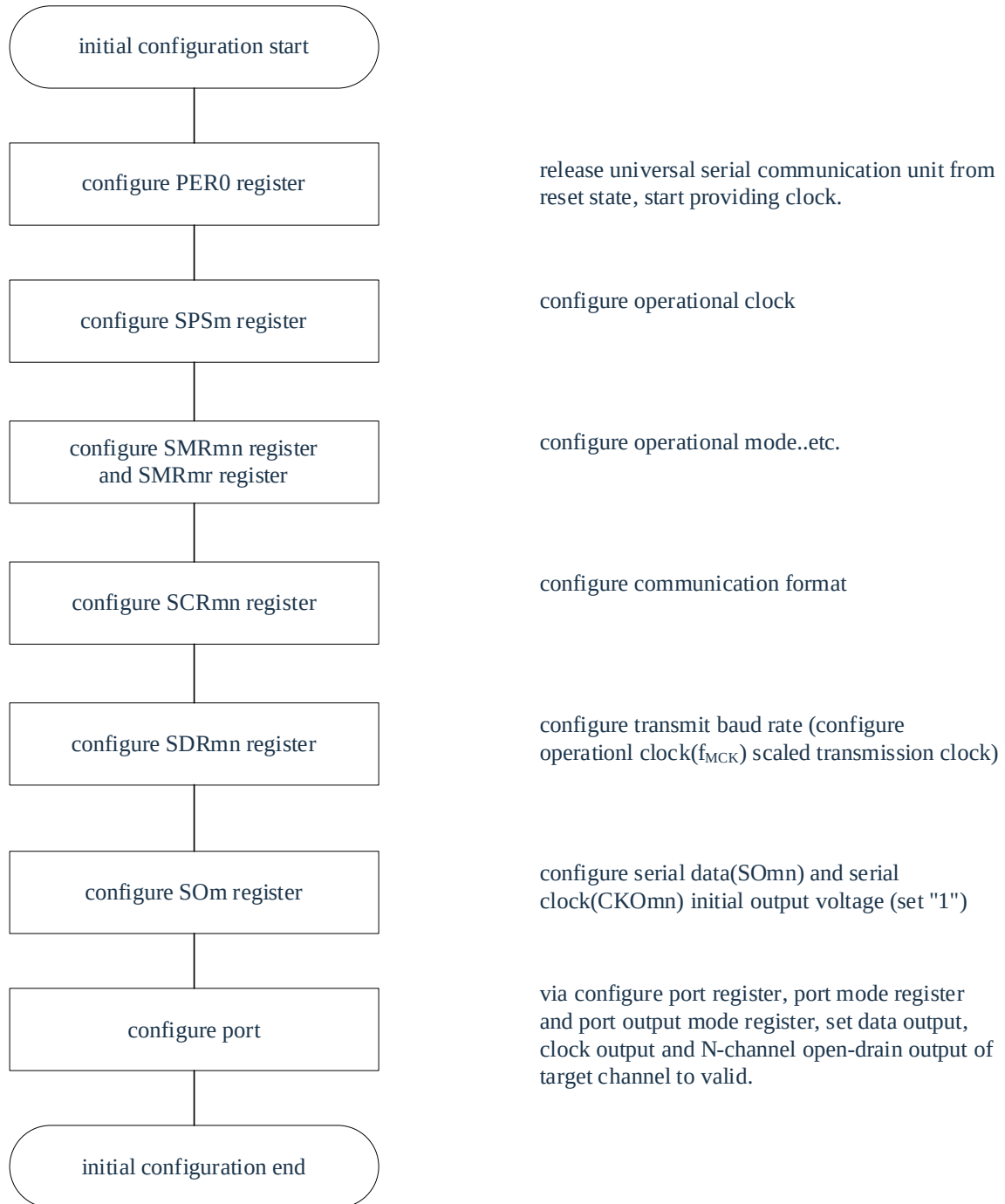
 : Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set to its initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

② Procedure

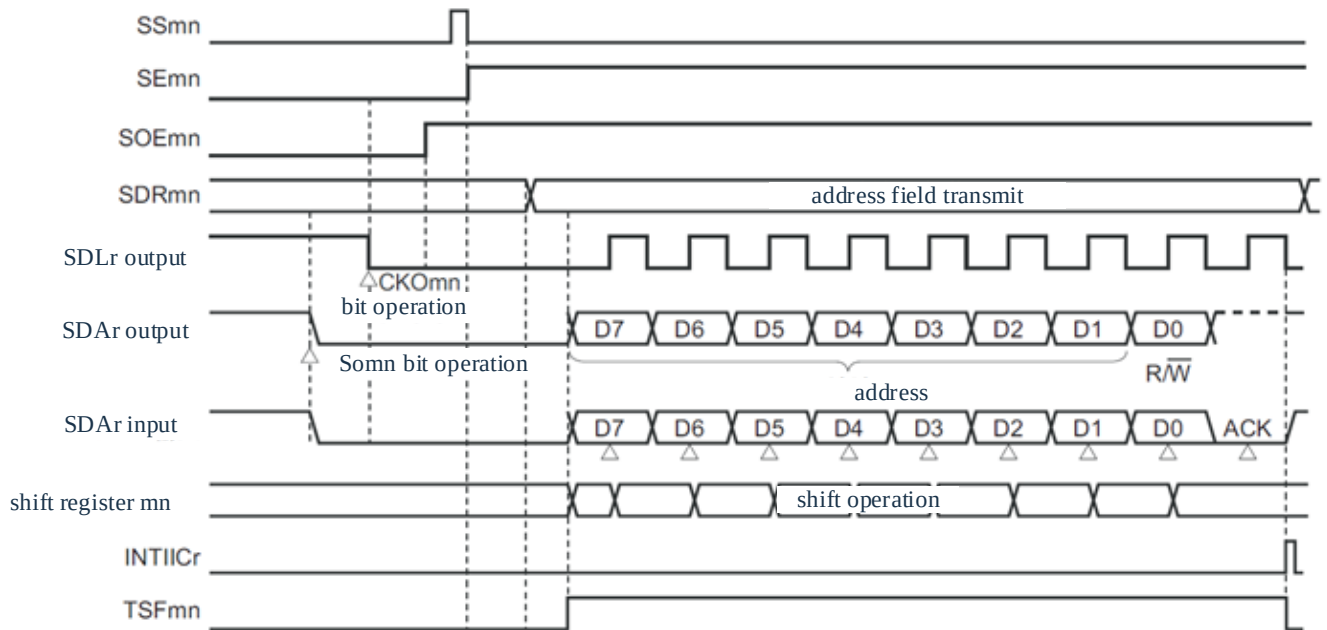
Figure 12-95: Initial setup steps for address field transmission



Remark: At the end of the initial setup, the simplified I²C (IIC00) is disabled for output and is in a stopped state.

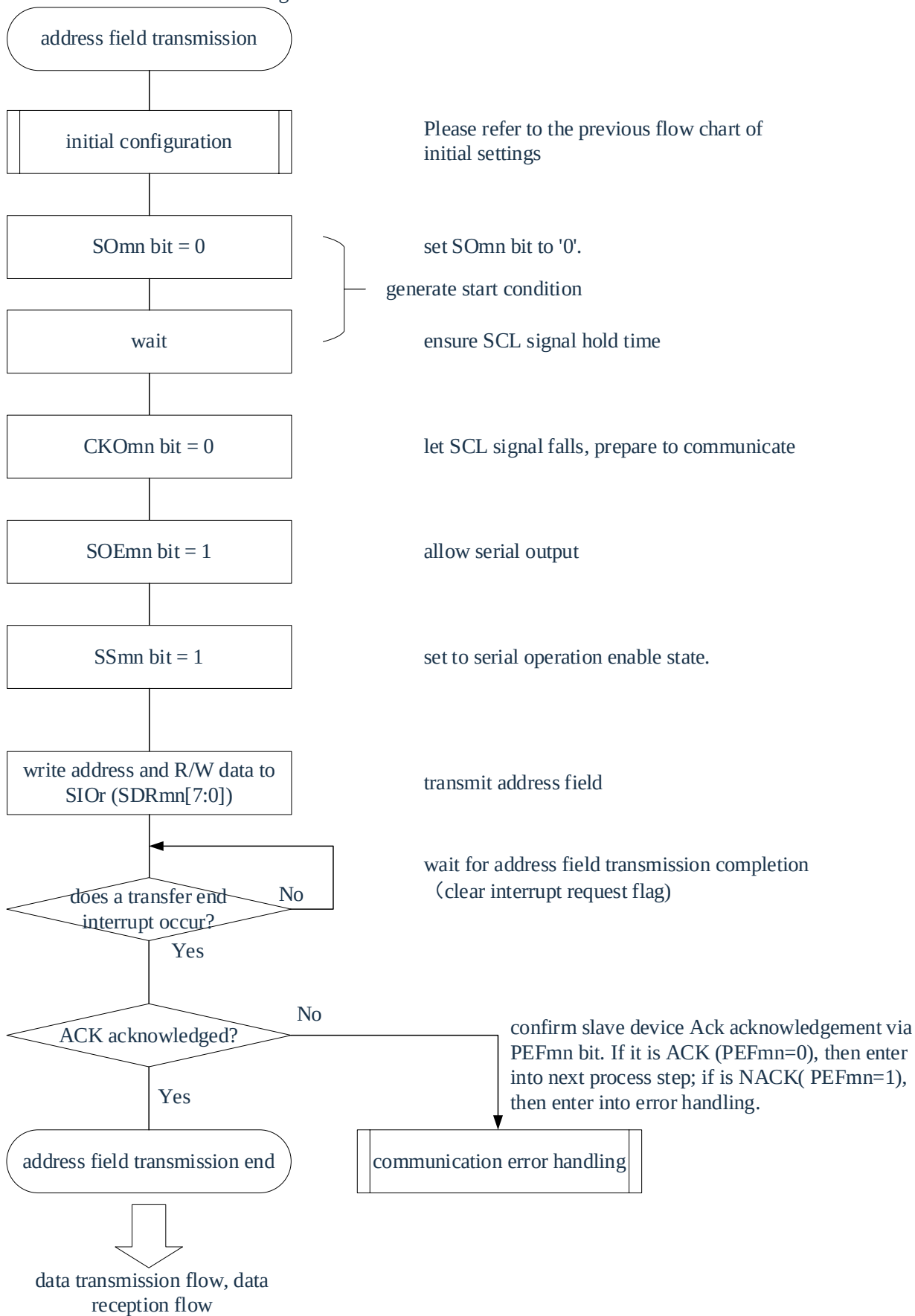
③ Processing flow

Figure 12-96: Timing diagram of address field transmission



Remark: m: unit number (m=0) n: channel number (n=0, 1)r: IIC number (r=00)

Figure 12-97: Flowchart of address field transmission



12.8.2 Data Transmission

Data transmission is the operation of transmitting data to the transmission target (slave device) after the address field is transmitted. A stop condition is generated after all data is sent to the target slave and the bus is released.

Table 12-43: Data transmission

Simplified I ² C	IIC00
Target channel	Channel of SCIO
Pin used	SCLKOI00, SAU_SDA00 ^{Note 1}
Interrupt	INTIIC00
	Limited to transmit end interrupts (buffer empty interrupts cannot be selected).
Error detection flag	ACK error flag (PEFmn)
Transfer data length	8 bits
Transfer rate ^{Note2}	$\text{Max. } F_{\text{MCK}}/4[\text{Hz}](\text{SDRmn}[15:9] \geq 1)$ F_{MCK} : operation clock frequency of the target channel. However, the following conditions must be met in each mode of I ² C. • Max.1MHz (enhanced fast mode). • Max.400kHz (fast mode). • Max.100kHz (standard mode).
Data level	Non-reverse output (default: high).
Parity bit	No parity bits.
Stop bit	Append 1 bit (for ACK reception).
Data direction	MSB first

Note 1: To communicate via simplified I²C, the N-channel open-drain output mode must be set through the Port Output Mode Register (POMxx) (POMxx=1). For detailed information, refer to "2.2 Port Alternate Functions" and "2.3.2 Output/Input Alternate Function Register Mapping.

Note 2: It must be used within the scope of peripheral functional characteristics that meet this condition and meet the electrical characteristics (see data sheet).

Remark: m: unit number (m=0) n: channel number (n=0, 1)

① Register setting

Figure 12-98: Example of register setting contents for simplified I²C data transmission

(a) Serial mode register mn (SMRmn).....do not operate this register while data is transmitting or receiving.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0 ^{Note1}	0	0 ^{Note1}	1	0	0	1	0	0

(b) Serial communication operation configuration register mn (SCRmn).....do not operate bits other than TXEmn and RXEmn of this register while data is transmitting or receiving.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0	0	0	0	0	0	0	0	0 ^{Note2}	1	0	1	1 ^{Note3}	1

(c) Serial data register mn (SDRmn) (low 8 bits: SIO_r) only lower 8 bits valid while data is transmitting or receiving.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate configuration ^{Note4}							0	Configuration of transmit data							
									SIO _r							

(d) Serial output register m (Som) do not operate this register while data is transmitting or receiving.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Som							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	0/1 ^{Note5}	0/1 ^{Note5}	0	0	0	0	0	0	0/1 ^{Note5}	0/1 ^{Note5}

(e) Serial output enable register m (SOEm) do not operate this register while data is transmitting or receiving.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm															SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

(f) Serial channel start register m (SSm) do not operate this register while data is transmitting or receiving.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm															SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

Note 1: Limited to SMR01 register.

Note 2: Limited to SCR00 register.

Note 3: Limited to SCR00 register and SCR01 register, other fixed as “1”.

Note 4: Because it is already set when sending the address segment, it does not need to be set.

Note 5: During the operation of communication, the value changes due to the communication data.

Remark: m: unit number (m=0) n: channel number (n=0, 1)r: IIC number (r=00)



: Fixed in IIC mode;



: Cannot be set (initial value);

×: This is the bit that cannot be used in this mode (set to its initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

② Processing flow

Figure 12-99: Timing diagram of data transmission

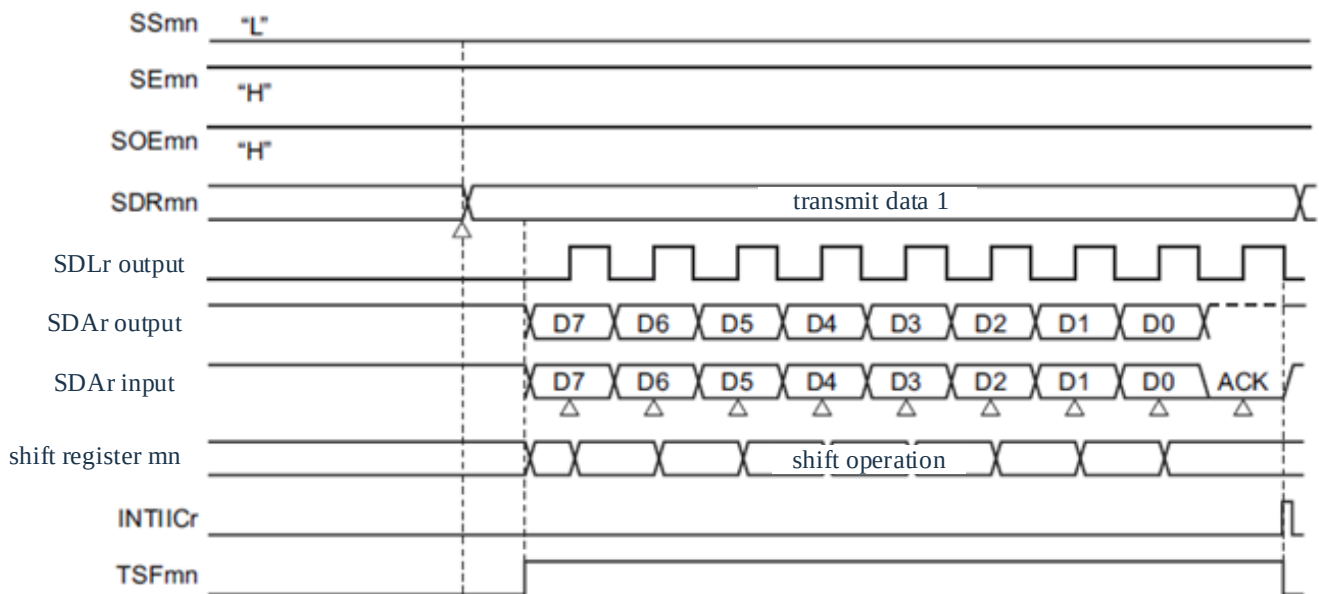
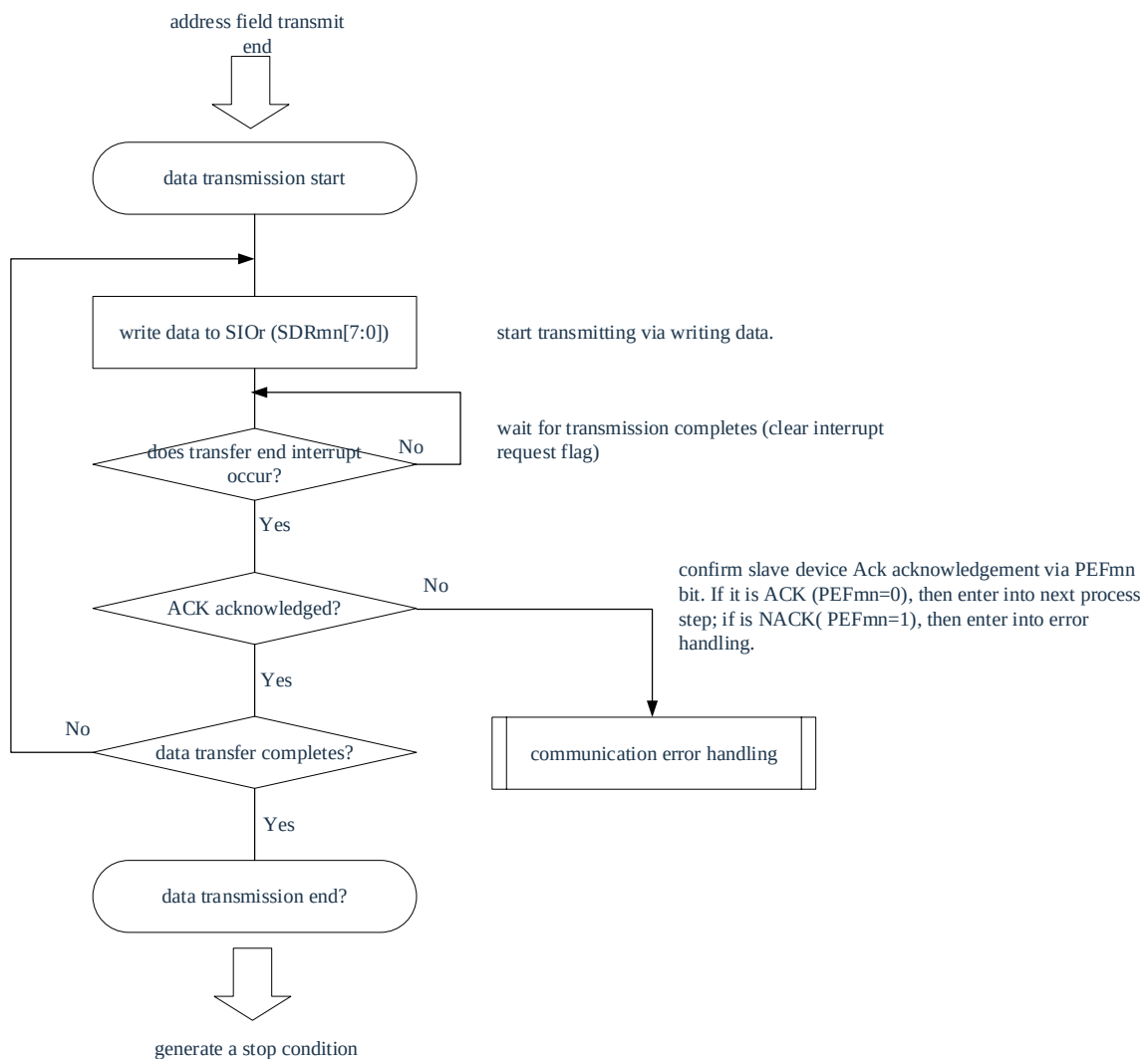


Figure 12-100: Flow chart of data transmission



12.8.3 Data Reception

Data reception is an operation of receiving data from the transmission target (slave) after sending the address field. A stop condition is generated and the bus is released after all data is received from the target slave.

Table 12-44: Data reception

Simplified I ² C	IIC00
Target channel	Channel of SCIO
Pin used	SCLKOI00, SDIO0 ^{Note 1}
Interrupt	INTIIC00 Limited to transmit complete interrupts (buffer empty interrupts cannot be selected).
Error detection flag	Only the overrun error detection flag (OVFmn).
Transfer data length	8 bits
Transfer rate ^{Note2}	Max. $f_{MCK}/4[Hz]$ ($SDRmn[15:9] \geq 1$) f_{MCK} : Operation clock frequency of the target channel. However, the following conditions must be met in each mode of I ² C. • Max.1MHz (enhanced fast mode). • Max.400kHz (fast mode). • Max.100kHz (standard mode).
Data level	Non-reverse output (default: high).
Parity bit	No parity bits.
Stop bit	Append 1 bit (for ACK transmission).
Data direction	MSB first

Note 1: To communicate via simplified I²C, the N-channel open-drain output mode must be set through the Port Output Mode Register (POMxx) (POMxx=1). For detailed information, refer to 2.2 Port Alternate Functions and 2.3.2 Output/Input Alternate Function Register Mapping.

Note 2: It must be used within the scope of peripheral functional characteristics that meet this condition and meet the electrical characteristics (see data sheet).

Remark: m: unit number (m=0) n: channel number (n=0, 1)

① Register setting

Figure 12-101: Example of register setting contents for simplified I²C data reception (IIC00)

(a) Serial mode register mn (SMRmn).....do not operate this register while data is transmitting or receiving.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn					STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0 ^{Note1}	0	0 ^{Note1}	1	0	0	1	0	0

(b) Serial communication operation configuration register mn (SCRmn).....do not operate bits other than TXEmn and RXEmn of this register while data is transmitting or receiving.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0		DLSmn1	DLSmn0
	0	1	0	0	0	0	0	0	0	0	0 ^{Note2}	1	0	1 ^{Note3}	1

(c) Serial data register mn (SDRmn) (low 8 bits: SIO_r)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate configuration ^{Note4}							0	Dummy transmit data configuration (FFH)						

SIO_r

(d) Serial output register m (SOM) do not operate this register while data is transmitting or receiving.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM							CKOm1	CKOm0						SOm1	SOm0
	0	0	0	0	0	0	0/1 ^{Note5}	0/1 ^{Note5}	0	0	0	0	0	0/1 ^{Note5}	0/1 ^{Note5}

(e) Serial output enable register m (SOEm) do not operate this register while data is transmitting or receiving.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm														SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

(f) Serial channel start register m (SSm) do not operate this register while data is transmitting or receiving.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm														SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

Note 1: Limited to SMR01 register.

Note 2: Limited to SCR00 register.

Note 3: Limited to SCR00 register and SCR01 register, other fixed as “1”.

Note 4: Because it is already set when sending the address segment, it does not need to be set.

Note 5: During the operation of communication, the value changes due to the communication data.

Remark: m: unit number (m=0) n: channel number (n=0, 1)r: IIC number (r=00)



: Fixed in IIC mode;



: Cannot be set (initial value);

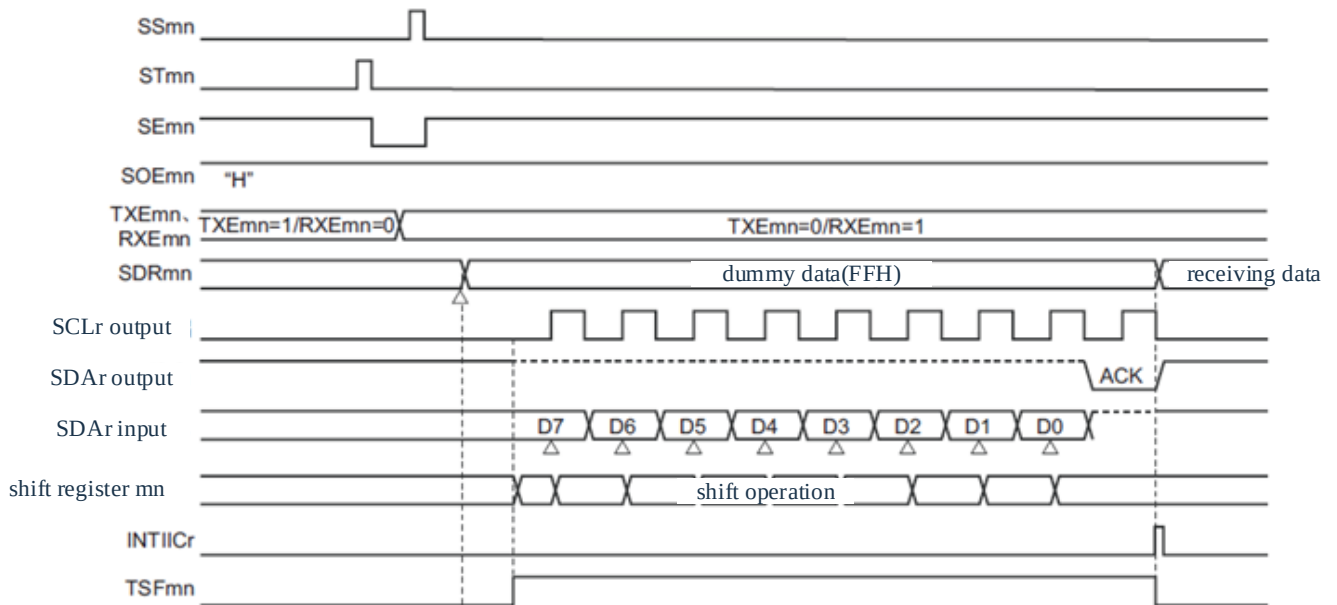
×: This is the bit that cannot be used in this mode (set to its initial value if it is not used in other modes either).

0/1: Set 0 or 1 according to the user's purpose.

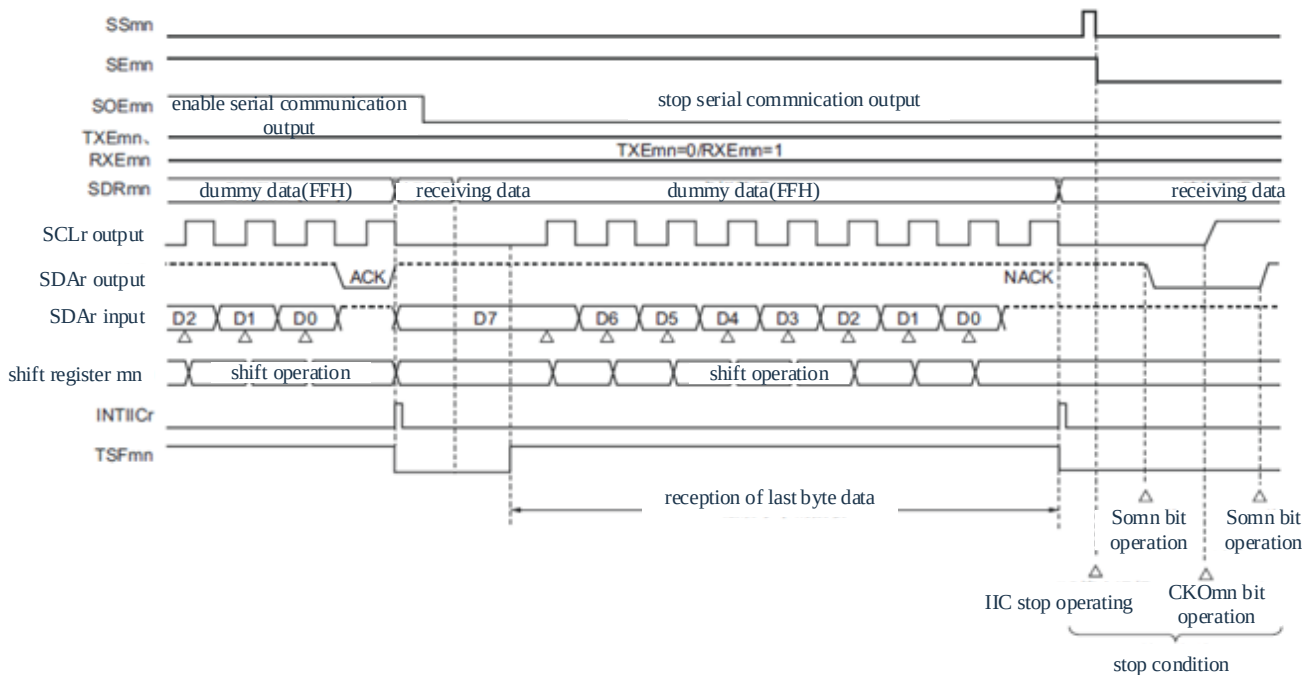
② Processing flow

Figure 12-102: Timing diagram of data reception

- Start of receiving data

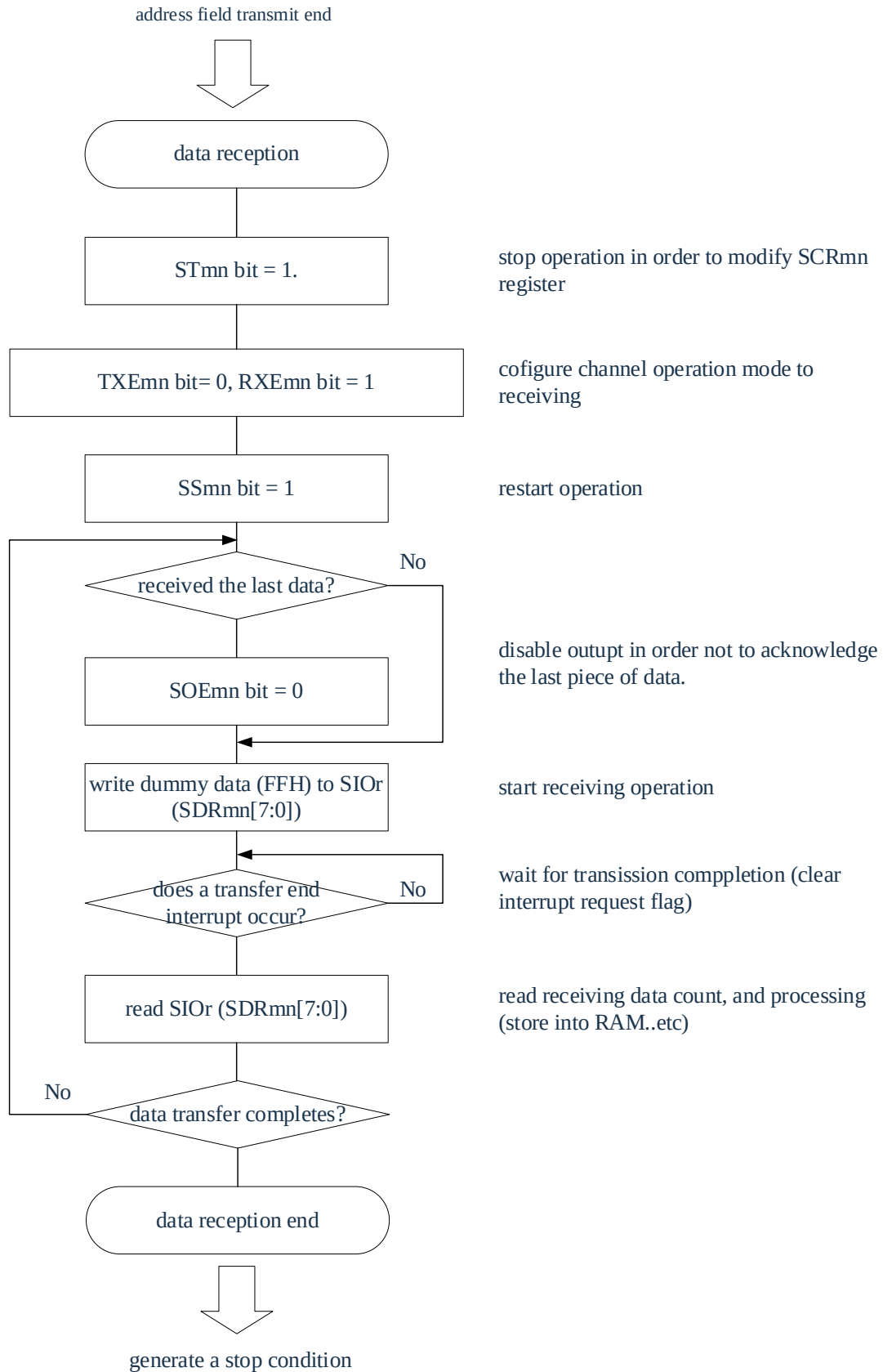


- Receive the last data



Remark: m: unit number (m=0) n: channel number (n=0, 1): IIC number (r=00)

Figure 12-103: Flowchart of data reception



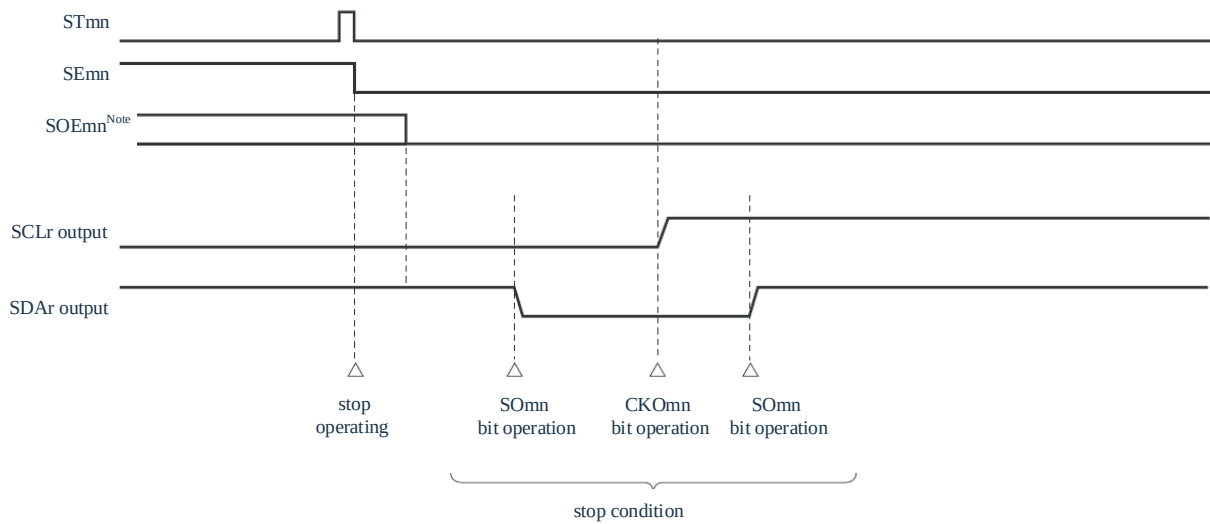
Remark: No ACK(NACK) is output when the last data is received. Thereafter, the communication is terminated by stopping the STmn bit of the serial channel stop register m (STm) to 1.

12.8.4 Generation of Stop Condition

After all data has been transmitted and received with the target slave, a stop condition is generated and the bus is released.

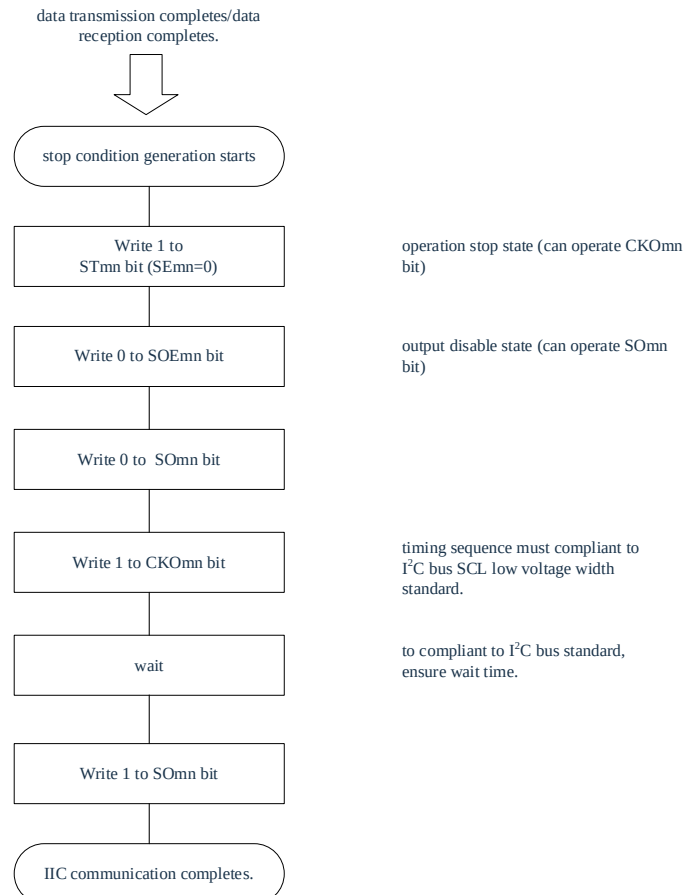
Processing flow

Figure 12-104: Timing diagram for generating a stop condition



Note: At the time of receiving, set the SOEmn bit of the serial output enable register m (SOEm) to “0”.

Figure 12-105: Flow chart for generating a stop condition



12.8.5 Calculation of Transfer Rate

The transfer rate for simplified I²C (IIC00) communication can be calculated using the following equation.

$$(\text{Transfer rate}) = \{ \text{Operation clock (F}_{\text{MCK}}) \text{ frequency of the target channel} \} / (\text{SDRmn}[15:9] + 1) / 2$$

The operating clock (FMCK) depends on the bit 15 (CKSmn bit) of the serial clock selection register m (SPSm) and the serial mode register mn (SMRmn).

Caution:

It is prohibited to set SDRmn[15:9] to “0000000B”. The set value of SDRmn[15:9] must be greater than or equal to “0000001B”. The duty cycle of the SCL signal output in simplified I²C is 50%. In the I²C bus specification, the low-level width of the SCL signal is greater than the high-level width. Therefore, if set to fast mode of 400kbps or enhanced fast mode of 1Mbps, the low-level width of the SCL signal output will be smaller than the specified value for the I²C bus. The SDRmn[15:9] must be set to a value that meets the I²C bus specification.

Since the value of SDRmn[15:9] corresponds to bits 15–9 of the serial data register (SDRmn) (0000001B–1111111B), it ranges from 1 to 127.

Remark: m: unit number (m=0) n: channel number (n=0, 1)

Table 12-45: Selection of simplified I²C operation clock

SMRmn register	SPSm register								Operation clock (F _{MCK}) ^{Note1}	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		F _{CLK} =24MHz in operation
0	X	X	X	X	0	0	0	0	F _{CLK}	24MHz
	X	X	X	X	0	0	0	1	F _{CLK} /2	12MHz
	X	X	X	X	0	0	1	0	F _{CLK} /2 ²	6MHz
	X	X	X	X	0	0	1	1	F _{CLK} /2 ³	3MHz
	X	X	X	X	0	1	0	0	F _{CLK} /2 ⁴	1.5MHz
	X	X	X	X	0	1	0	1	F _{CLK} /2 ⁵	0.75MHz
	X	X	X	X	0	1	1	0	F _{CLK} /2 ⁶	375KHz
	X	X	X	X	0	1	1	1	F _{CLK} /2 ⁷	187.5KHz
	X	X	X	X	1	0	0	0	F _{CLK} /2 ⁸	93.75KHz
	X	X	X	X	1	0	0	1	F _{CLK} /2 ⁹	46.875KHz
	X	X	X	X	1	0	1	0	F _{CLK} /2 ¹⁰	23.44KHz
	X	X	X	X	1	0	1	1	F _{CLK} /2 ¹¹	11.72KHz
1	0	0	0	0	X	X	X	X	F _{CLK}	24MHz
	0	0	0	1	X	X	X	X	F _{CLK} /2	12MHz
	0	0	1	0	X	X	X	X	F _{CLK} /2 ²	6MHz
	0	0	1	1	X	X	X	X	F _{CLK} /2 ³	3MHz
	0	1	0	0	X	X	X	X	F _{CLK} /2 ⁴	1.5MHz
	0	1	0	1	X	X	X	X	F _{CLK} /2 ⁵	0.75MHz
	0	1	1	0	X	X	X	X	F _{CLK} /2 ⁶	375KHz
	0	1	1	1	X	X	X	X	F _{CLK} /2 ⁷	187.5KHz
	1	0	0	0	X	X	X	X	F _{CLK} /2 ⁸	93.75KHz
	1	0	0	1	X	X	X	X	F _{CLK} /2 ⁹	46.875KHz
	1	0	1	0	X	X	X	X	F _{CLK} /2 ¹⁰	23.44KHz
	1	0	1	1	X	X	X	X	F _{CLK} /2 ¹¹	11.72KHz
Other than the above									Disable settings.	

Note 1: To change the clock selected as F_{CLK} (modify the value of the System Clock Control Register (CKC)), the change must be made after stopping the operation of the General Serial Communication Unit (SCI) (Serial Channel Stop Register m (STm) = 000FH).

Remark: X: Ignore

m: unit number (m=0); n: channel number (n=0, 1).

An example of setting the I²C transfer rate at $F_{MCK}=F_{CLK}=24\text{MHz}$ is shown below.

Table 12-46: I²C transfer rate configuration

I ² C transfer mode (expected transfer rate)	$F_{CLK}=24\text{MHz}$			
	Operation clock (F_{MCK})	SDRmn[15:9]	Calculated transfer rate	Error with expected transfer rate
100KHz	$F_{CLK}/2$	61	100KHz	3.2%
400KHz	F_{CLK}	30	380KHz	3.2% ^{Note}
1MHz	F_{CLK}	12	0.92MHz	7.7% ^{Note}

Note: The error cannot be set to 0% because the SCL signal has a 50% duty cycle.

12.8.5.1 Processing Steps When an Error Occurs During Simplified I²C (IIC00) Communication

The processing steps when an error occurs during a simplified I²C (IIC00) communication are shown in Figure 12-47 and Figure 12-48.

Table 12-47: Steps for handling when an overrun error occurs

Software operation	Hardware status	Remark
Read serial data register mn (SDRmn)	The BFFmn bit of the SSRmn register is 0 and channel n is receivable.	This is to prevent overrun errors from ending the next reception during mishandling.
Read serial status register mn (SSRmn)		The type of error is judged, and the reading value is used to clear the error flag.
Write 1 to the serial flag clear trigger register mn (SDIRmn).	Clear the error flag.	By writing the read value of the SSRmn register directly to the SDIRmn register, errors during read operations can only be cleared.

Table 12-48: Processing steps when an ACK error occurs during simplified I²C mode

Software operation	Hardware status	Remark
Read the serial status register mn (SSRmn)		Determine the error category, and the read value is used to clear the error flag.
Write the serial flag to clear the trigger register mn (SDIRmn).	Clear the error flag.	By writing the read value of the SSRmn register directly to the SDIRmn register, errors during read operations can only be cleared.
Set the STmn bit of the serial channel stop register m (STm) to 1.	The SEMn bit of the Serial Channel Enable Status Register m (SEm) is 0 and channel n is in run stop status.	Because ACK is not returned, the slave device is not ready for receiving. Thus, a stop condition is generated and the bus is released, and communication is started again from the start bar, or a restart can also be generated and start again from the address to send.
Generate a stop condition.		
Generate a start condition		
Set the SSmn bit of the serial channel start register m (SSm) to 1.	The SEMn bit of the Serial Channel Enable Status Register m (SEm) is 1 and the channel n is operational.	

Remark: m: unit number (m=0) n: channel number (n=0, 1) r: IIC number (r=00)

Chapter 13 Serial Peripheral Interface (SPI)

13.1 Introduction

The Serial Peripheral Interface (SPI) allows for half-duplex/full-duplex synchronous serial communication with external devices. This interface can be configured in master mode, in which case it provides a communication clock (SCK) for external slave devices. It can also operate in multi-master mode.

It is used for various purposes, including single-wire synchronous transmission based on a two-wire setup, where one wire serves as a bidirectional data line, or for reliable communication using CRC.

13.2 SPI

13.2.1 SPI Features

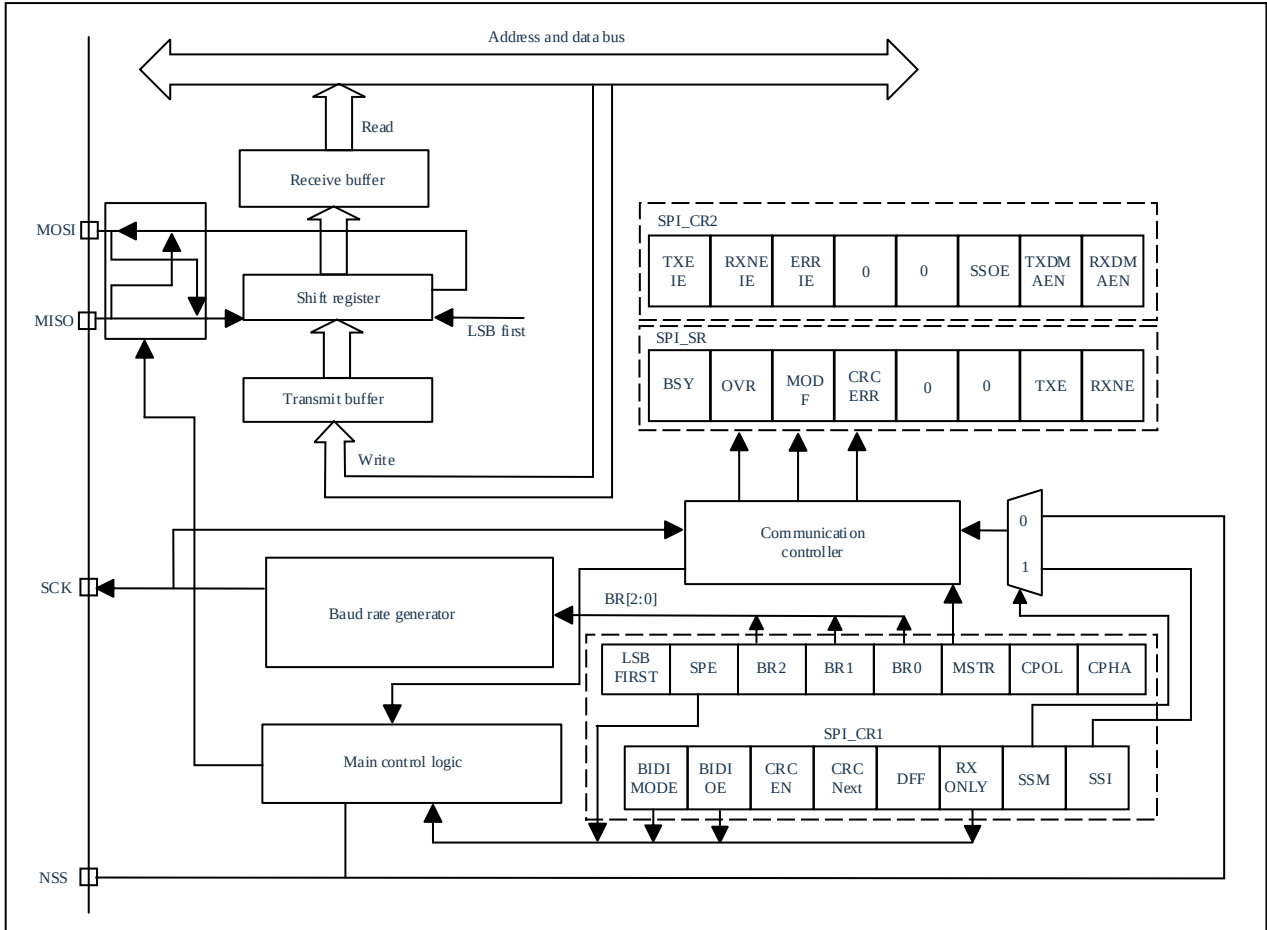
- Full-duplex synchronous transfers on three lines
- Simplex synchronous transfers on two lines (one with bidirectional data line)
- 8 to 16-bit data size selection
- Master/slave mode operation
- Multimaster mode capability
- 8 master mode baud rate prescalers up to $f_{PCLK}/2$
- Slave mode frequency up to 6MHz.
- Faster communication can be achieved for both master and slave modes.
- Both master and slave modes can manage NSS via hardware or software: Dynamic switching of master/slave operation.
- Programmable clock polarity and phase.
- Programmable data order, with MSB or LSB first shifting.
- Dedicated transmit and receive flags that can trigger interrupts.
- SPI bus busy status flag
- SPI TI mode
- Hardware CRC functionality to ensure reliable communication:
 - CRC value can be transmitted as last byte in Tx mode
 - Automatic CRC error checking for last received byte
- Master mode fault, overrun, and CRC error flags that can trigger interrupts.
- 1-byte transmit and receive buffers with DMA functionality: transmit and receive requests.

13.3 SPI Functional Description

13.3.1 General Description

The SPI is shown in the following block diagram.

Figure 13-1 SPI block diagram

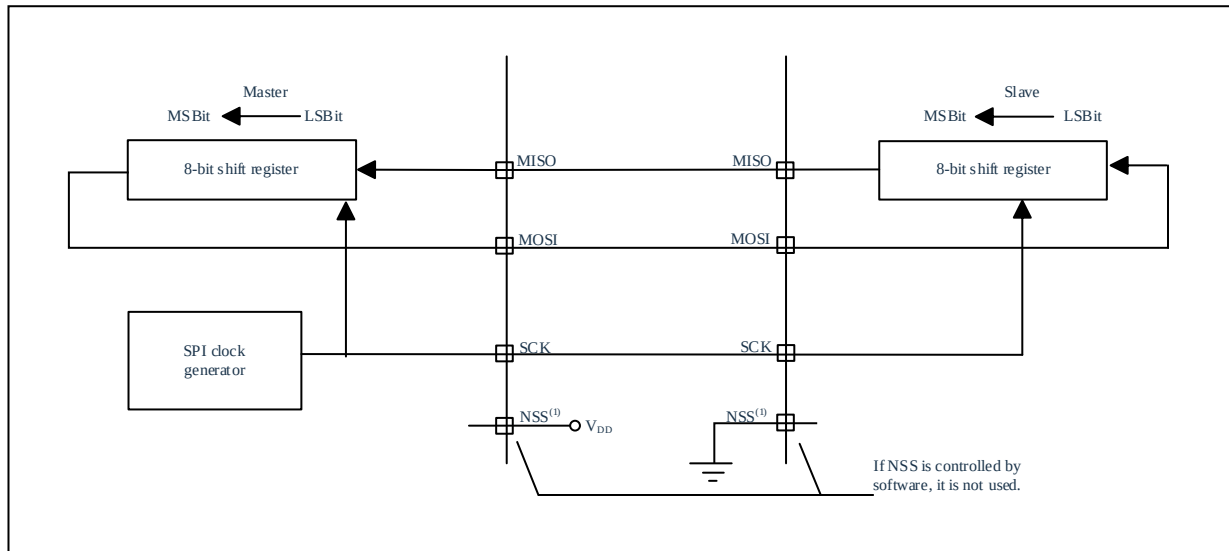


Four I/O pins are dedicated to SPI communication with external devices.

- **MISO:** Master In / Slave Out data. In the general case, this pin is used to transmit data in slave mode and receive data in master mode.
- **MOSI:** Master Out / Slave In data. In the general case, this pin is used to transmit data in master mode and receive data in slave mode.
- **SCK:** Serial Clock output pin for SPI masters and input pin for SPI slaves.
- **NSS:** Slave device select. This is an optional pin used to select the slave device. This pin acts as chip select and allows the SPI master device to communicate with the slave device individually, thereby avoiding contention on the data line. The NSS input of the slave device can be driven by a standard I/O port on the master device. When enabled (SSOE bit), the NSS pin can also function as an output and can be driven low when the SPI is in master mode configuration. In this way, as long as the device is configured in NSS hardware management mode, all other devices connected to the NSS pin of the master device will have their NSS pins pulled low and thus function as slave devices. When configured in master mode, and NSS is configured as input (MSTR=1 and SSOE=0), if NSS is pulled low, the SPI will enter a master mode fault state: the MSTR bit will be automatically cleared, and the device will be configured as a slave mode.

Figure 13-2 shows a basic example of the interconnection between a single master device and a single slave device.

Figure 13-2 Single master/ single slave application



Here, the NSS pin is configured as input.

The MOSI pin and MISO pin are connected together. In this way, data is transmitted serially between the master and slave devices (with the most significant bit first). Communication is always initiated by the master device. When the master device sends data to the slave device through the MOSI pin, the slave device simultaneously responds through the MISO pin. This is a full-duplex communication process, where both data output and data input are synchronized by the same clock.

Slave Device Select (NSS) Pin Management

The slave device select information can be managed by hardware or software through the SSM bit in the SPI_CR1 register.

- Software Management of NSS (SSM = 1)

The NSS information is driven internally by the value of the SSI bit in the SPI_CR1 register. The external NSS pin is free for use by other applications.

- Hardware Management of NSS (SSM = 0)

There are two modes of hardware-managed NSS depending on the NSS output configuration (SSOE bit in the SPI_CR1 register).

- NSS Output Enabled (SSM = 0, SSOE = 1): This configuration is only used when the device is in master mode. When the master device starts communication, the NSS signal is driven low and stays low until SPI is disabled.
- NSS Output Disabled (SSM = 0, SSOE = 0):

For devices operating in master mode, this configuration allows multi-master functionality. For devices configured as slave mode, the NSS pin functions as a traditional NSS input: when NSS is low, the slave device is selected, and when NSS is high, the slave device is deselected.

Clock Phase and Clock Polarity

Through the CPOL and CPHA bits in the SPI_CR1 register, four possible timing relationships can be selected via software. The CPOL (Clock Polarity) bit controls the clock level state when no data is being transmitted. This bit affects both the master and slave devices. If CPOL is reset, the SCK pin is low in the idle state. If CPOL is set to 1, the SCK pin is high in the idle state.

If the CPHA (Clock Phase) bit is set to 1, the second edge of the SCK pin (falling edge if CPOL is reset, rising edge if CPOL is set to 1) samples the MSBit. In other words, data is latched on the second clock edge. If CPHA is reset, the first edge of the SCK pin (falling edge if CPOL is set to 1, rising edge if CPOL is reset) samples the MSBit. In other words, data is latched on the first clock edge.

The combination of the CPOL (Clock Polarity) and CPHA (Clock Phase) bits is used to select the clock edge for data capture.

Figure 13-3 shows the SPI transmission for all four combinations of CPHA and CPOL bits. This diagram can be interpreted as the timing diagram for either the master or slave device, where the SCK, MISO, and MOSI pins are directly connected between the master and slave devices.

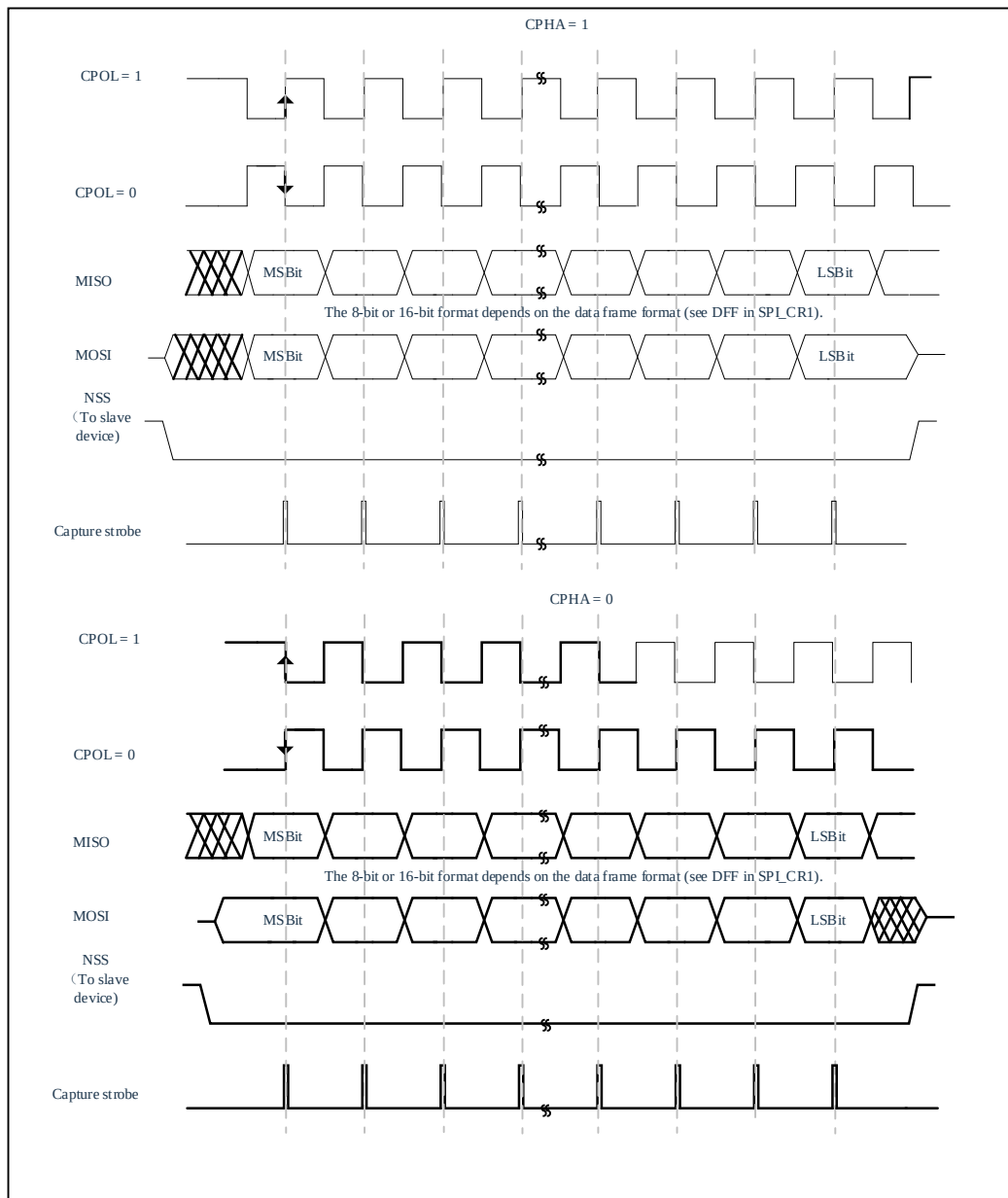
Caution: SPI must be disabled by resetting the SPE bit before switching the CPOL/CPHA bits.

The master and slave devices must be programmed with the same timing mode.

The idle state of the SCK must correspond to the selected polarity in the SPI_CR1 register (if CPOL=1, then SCK is pulled high; if CPOL=0, then SCK is pulled low).

The data frame format (8 or 16 bits) is selected through the DFF bit in the SPI_CR1 register, which determines the data length for transmission/reception.

Figure 13-3 Data clock timing diagram



1. The timing shown is for when the LSBFIRST bit in the SPI_CR1 register is reset.

Data Frame Format

Whether the MSB or LSB is transmitted first depends on the value of the LSBFIRST bit in the SPI_CR1 register. Each data frame is either 8 bits or 16 bits long, depending on the DFF bit in the SPI_CR1 register. The selected data frame format applies to both transmission and reception.

13.3.2 Configuring SPI as a Slave Device

In slave mode configuration, the serial clock from the master device is received on the SCK pin. The value set in the BR[2:0] bits of the SPI_CR1 register does not affect the data transfer rate.

Caution: It is recommended to enable the SPI slave device before the master device sends the clock. Otherwise, data transmission may be abnormal. The data register of the slave device must be ready before the first edge of the master clock arrives or before ongoing communication ends. The clock polarity of the communication must be set to the idle clock level before enabling the slave and master devices.

To configure the SPI mode to slave mode, follow the steps below:

Steps:

1. Set the DFF bit to define the 8-bit or 16-bit data frame format.
2. Choose the CPOL and CPHA bits to define the relationship between data transfer and the serial clock (one of four possible relationships) (refer to Figure 13-3). To ensure correct data transfer, the CPOL and CPHA bits must be configured in the same way in both the slave and master devices. If the TI mode is selected through the FRF bit in the SPI_CR2 register, this step is not needed.
3. The frame format (MSB first or LSB first depending on the LSBFIRST bit in the SPI_CR1 register) must be the same as that of the master device. This step is not necessary if TI mode is selected.
4. In hardware mode (refer to NSS pin management), the NSS pin must remain low throughout the entire byte transfer sequence. In NSS software mode, set the SSM bit in the SPI_CR1 register to 1 and clear the SSI bit. If TI mode is selected, this step is not required.
5. Set the FRF bit in the SPI_CR2 register to 1 to select the TI mode protocol for serial communication.
6. Clear the MSTR bit and set the SPE bit to 1 (both bits are in the SPI_CR1 register).

In this configuration, the MOSI pin is the data input, and the MISO pin is the data output.

13.3.2.1 Transmission Sequence

Data bytes are loaded in parallel into the transmission buffer during the write cycle.

When the slave device receives the clock signal and the most significant bit of the data on its MOSI pin, the transmission sequence begins. The remaining bits (7 bits for an 8-bit data frame format, 15 bits for a 16-bit data frame format) are loaded into the shift register. The TXE flag in the SPI_SR register is set to 1 when data is transferred from the transmission buffer to the shift register, and an interrupt will be generated if the TXEIE bit in the SPI_CR2 register is set to 1.

13.3.2.2 Reception Sequence

For the receiver, when the data transfer is complete:

- The data in the shift register is transferred to the reception buffer, and the RXNE flag (SPI_SR register) is set to 1.
- If the RXNEIE bit in the SPI_CR2 register is set to 1, an interrupt is generated.

After the last sampling clock edge, RXNE is set to 1, and the received data byte in the shift register is copied to the reception buffer. When the SPI_DR register is read, this buffer value is returned by the SPI peripheral.

Reading the SPI_DR register clears the RXNE bit.

13.3.2.3 SPI TI Protocol in Slave Mode

In slave mode, the SPI interface is compatible with the TI protocol. The FRF bit in the SPI_CR2 register can be used to configure the SPI for serial communication in slave mode to be compatible with this protocol.

Both the clock polarity and phase are forced to follow the TI protocol and are independent of the settings in SPI_CR1. NSS management is also specific to the TI protocol, which means the user does not need to configure NSS management by setting the SPI_CR1 and SPI_CR2 registers (e.g., SSM, SSI, SSOE).

In slave mode (refer to Figure 13-4: TI Mode – Slave Mode, Single Transfer and Figure 13-5: TI Mode – Slave Mode, Continuous Transfer), the SPI baud rate prescaler is used to control the moment when the MISO pin switches to high-impedance state. Any baud rate can be used, providing flexibility in determining this moment. However, the baud rate is usually set to match the external master clock baud rate. The time when the MISO signal enters the high-impedance state (t_{release}) depends on the internal synchronization circuits of the chip and the baud rate value set by the BR[2:0] in the SPI_CR1 register. The specific formula is as follows:

$$\frac{t_{\text{baud\_rate}}}{2} + 4 \times t_{\text{pclk}} < t_{\text{release}} < \frac{t_{\text{baud\_rate}}}{2} + 6 \times t_{\text{pclk}}$$

Caution: This feature does not apply to Motorola SPI communication (when the FRF bit is 0).

To use the error interrupt (ERRIE = 1) for detecting TI frame errors in slave transmitter mode, the SPI must be configured for a bidirectional, single-wire mode by setting the BIDIMODE and BIDIOE bits in the SPI_CR1 register to 1. When BIDIMODE is set to 0, the OVR flag will be set to 1 because the data register is never read, leading to a continuous overrun error interrupt. When BIDIMODE is set to 1, no data is received, and the OVR flag will not be set to 1.

Figure 13-4 TI Mode – Slave Mode, Single Transfer

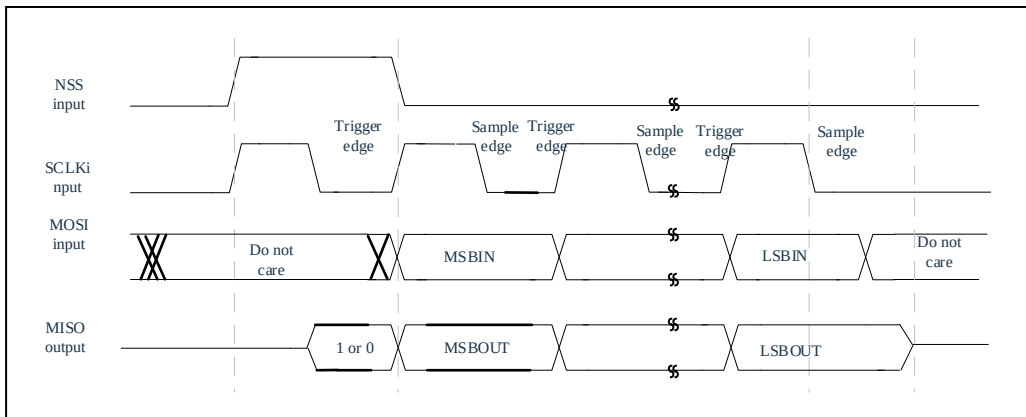
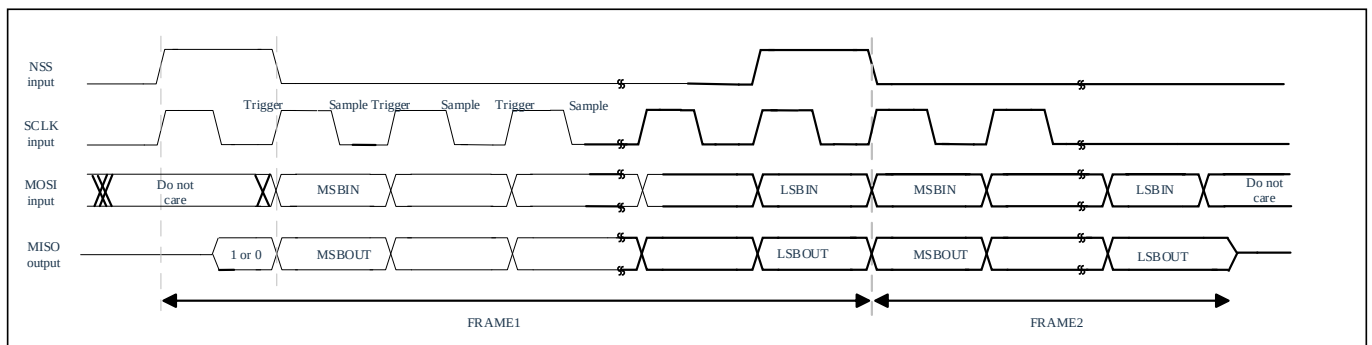


Figure 13-5 TI Mode – Slave Mode, Continuous Transfer



13.3.3 Configuring SPI as a Master Device

In master mode configuration, the serial clock is output on the SCK pin.

Steps:

1. Set the BR[2:0] bits to define the serial clock baud rate (refer to the SPI_CR1 register).
2. Choose the CPOL and CPHA bits to define the relationship between data transfer and the serial clock (one of four possible relationships) (refer to Figure 13-3). If TI mode is selected, this step is not needed.
3. Set the DFF bit to define the 8-bit or 16-bit data frame format.
4. Configure the LSBFIRST bit in the SPI_CR1 register to define the frame format. If TI mode is selected, this step is not needed.
5. If the NSS pin is configured as input, in NSS hardware mode, the NSS pin must be connected to a high-level signal during the entire byte transmission sequence; in NSS software mode, set the SSM and SSI bits in the SPI_CR1 register to 1. If the NSS pin is configured as output, set the SSOE bit to 1. If TI mode is selected, this step is not needed.
6. Set the FRF bit in the SPI_CR2 register to 1 to select the TI protocol for serial communication.
7. Set the MSTR and SPE bits to 1 (these bits will remain set to 1 only when the NSS pin is connected to a high-level signal).

In this configuration, the MOSI pin is the data output, and the MISO pin is the data input.

13.3.3.1 Transmission Sequence

The transmission sequence begins when a byte is written to the transmit buffer.

During the first bit transfer, the data byte (from the internal bus) is loaded in parallel into the shift register, then shifted out serially to the MOSI pin. Whether the MSB or LSB is transmitted first depends on the LSBFIRST bit in the SPI_CR1 register. The TXE flag is set to 1 when data is transferred from the transmit buffer to the shift register, and an interrupt will be generated if the TXEIE bit in the SPI_CR2 register is set to 1.

13.3.3.2 Reception Sequence

For the receiver, when the data transfer is complete:

- The data in the shift register is transferred to the receive buffer, and the RXNE flag is set to 1.
- If the RXNEIE bit in the SPI_CR2 register is set to 1, an interrupt will be generated.

When the final sampling clock edge occurs, the RXNE flag is set to 1, and the received data byte in the shift register is copied to the receive buffer. When the SPI_DR register is read, this buffer value is returned by the SPI peripheral.

Reading the SPI_DR register clears the RXNE bit.

If the next data to be transmitted is placed in the transmit buffer after transmission begins, continuous transmission flow can be maintained. Note that the transmit buffer can only be written when the TXE flag is 1.

Caution: If the slave device requires pulling the chip select signal low between each byte transfer, the master device's NSS pin must be configured as GPIO or use another GPIO to control the slave's chip select through software.

13.3.3.3 SPI TI Protocol in Master Mode

In master mode, the SPI interface is compatible with the TI protocol. The FRF bit in the SPI_CR2 register can be used to configure the master SPI serial communication to be compatible with this protocol.

Both the clock polarity and phase are forced to follow the TI protocol, regardless of the settings in SPI_CR1. NSS management is also specific to the TI protocol, which means users do not need to configure NSS management by setting the SPI_CR1 and SPI_CR2 registers (e.g., SSM, SSI, SSOE).

Below figures show the SPI master mode communication waveforms when TI mode is selected in master mode.

Figure 13-6 TI Mode – Master Mode, Single Transfer

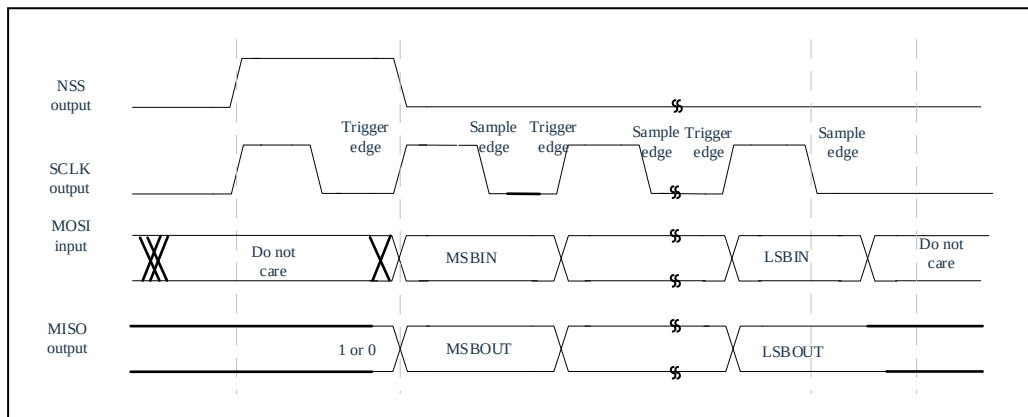
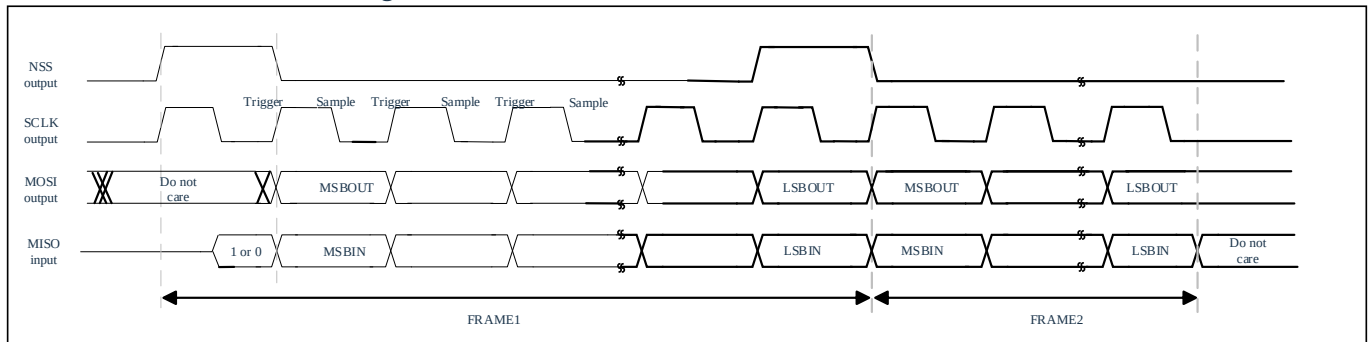


Figure 13-7 TI Mode – Master Mode, Continuous Transfer



13.3.4 Configuring SPI for Half-Duplex Communication

SPI can operate in half-duplex mode in the following two configurations:

- One clock and one bidirectional data line
- One clock and one data line (either receive or transmit only)

One clock and one bidirectional data line (BIDIMODE=1)

This mode can be enabled by setting the BIDIMODE bit in the SPI_CR1 register. In this mode, SCK is used for the clock, and MOSI (in master mode) or MISO (in slave mode) is used for data communication. The direction of data transmission (input/output) is selected using the BIDIOE bit in the SPI_CR1 register. When this bit is set to 1, the data line is output; otherwise, it is input.

One clock and one unidirectional data line (BIDIMODE=0)

In this mode, the application can use SPI in either transmit-only or receive-only mode:

- Transmit-only mode is similar to full-duplex mode (BIDIMODE=0, RXONLY=0): data is transmitted on the transmit pin (MOSI in master mode or MISO in slave mode), and the receive pin (MISO in master mode or MOSI in slave mode) can be used as a general-purpose I/O. In this case, the application only needs to ignore the receive buffer (even if the data register is read, it will not contain valid received data).
- Receive-only mode: In this mode, the application can set the RXONLY bit in the SPI_CR2 register to 1 to disable SPI output functionality. In this case, the transmit I/O pin (MOSI in master mode or MISO in slave mode) can be used for other purposes.

To start communication in receive-only mode, configure and enable SPI:

- In master mode, communication will begin immediately and will stop when the SPE bit is cleared and the current receive operation is finished. The BSY flag does not need to be read in this mode, as it is always set to 1 during SPI communication.
- In slave mode, as long as the NSS pin is pulled low (or in NSS software mode when the SSI bit is cleared) and there is continuous SCK input from the master device, SPI will continue to receive.

13.3.5 Data Transmission and Reception Process

During the reception and transmission process, data is first stored in the internal reception buffer upon receiving, and during transmission, data is first stored in the internal transmission buffer before being sent.

Reading from the SPI_DR register will return the value from the reception buffer, while writing to the SPI_DR register will store the data into the transmission buffer.

In Master Mode, Starting the Communication Sequence:

- In Full-Duplex Mode (BIDIMODE=0 and RXONLY=0)
 - When data is written to the SPI_DR register (transmission buffer), the communication sequence starts.
 - During the first bit transmission, the data is parallel loaded from the transmission buffer into the 8-bit shift register, then serially shifted out to the MOSI pin.
 - Simultaneously, the data received on the MISO pin is serially shifted into the 8-bit shift register, then parallel loaded into the SPI_DR register (reception buffer).
- In Unidirectional Receive-Only Mode (BIDIMODE=0 and RXONLY=1)
 - The communication sequence starts immediately as long as SPE = 1.
 - The receiver is activated, and data received on the MISO pin is serially shifted into the 8-bit shift register, then parallel loaded into the SPI_DR register (reception buffer).
- In Bidirectional Mode, Transmission (BIDIMODE=1 and BIDIOE=1):
 - When data is written to the SPI_DR register (transmission buffer), the communication sequence starts.
 - During the first bit transmission, the data is parallel loaded from the transmission buffer into the 8-bit shift register, then serially shifted out to the MOSI pin.
 - No data is received.
- In Bidirectional Mode, Reception (BIDIMODE=1 and BIDIOE=0):
 - The communication sequence starts immediately as long as SPE=1 and BIDIOE=0.
 - The data received on the MOSI pin is serially shifted into the 8-bit shift register, then parallel loaded into the SPI_DR register (reception buffer).
 - The transmitter is not activated, so no data is serially shifted out of the MOSI pin.

In Slave Mode, Starting the Communication Sequence:

- In Full-Duplex Mode (BIDIMODE=0 and RXONLY=0):
 - When the slave device receives the first bit of clock and data on its MOSI pin, the communication sequence starts. The remaining 7 bits are loaded into the shift register.
 - Simultaneously, during the first bit transmission, the data is parallel loaded from the transmission buffer into the 8-bit shift register, then serially shifted out to the MISO pin. Before the SPI master device starts transmission, the software must have written the data to be sent from the slave device into the transmission buffer.
- In Unidirectional Receive-Only Mode (BIDIMODE=0 and RXONLY=1):
 - When the slave device receives the first bit of clock and data on its MOSI pin, the communication sequence starts. The remaining 7 bits are loaded into the shift register.
 - The transmitter is not activated, so no data is serially shifted out of the MISO pin.
- In Bidirectional Mode, Transmission (BIDIMODE=1 and BIDIOE=1):

- When the slave device receives the clock signal and sends the first bit from the transmission buffer on the MISO pin, the communication sequence starts.
- During the first bit transmission, the data is parallel loaded from the transmission buffer into the 8-bit shift register, then serially shifted out to the MISO pin. Before the SPI master device starts transmission, the software must have written the data to be sent from the slave device into the transmission buffer.
- No data is received.
- In Bidirectional Mode, Reception (BIDIMODE=1 and BIDIOE=0):
 - When the slave device receives the first bit of clock and data on its MISO pin, the communication sequence starts.
 - The data received on the MISO pin is serially shifted into the 8-bit shift register, then parallel loaded into the SPI_DR register (reception buffer).
 - The transmitter is not activated, so no data is serially shifted out of the MISO pin.

13.3.5.1 Data Transmission and Reception Processing

When data is transferred from the transmission buffer to the shift register, the TXE flag (Transmission Buffer Empty) is set to 1. This flag indicates that the internal transmission buffer is ready to load the next data. If the TXEIE bit in the SPI_CR2 register is set to 1, an interrupt can be triggered. Writing to the SPI_DR register clears the TXE flag.

Caution: The software must ensure that the TXE flag is set to 1 before attempting to write to the transmission buffer. Otherwise, the data previously written to the transmission buffer will be overwritten. When data is transferred from the shift register to the reception buffer, the RXNE flag (Reception Buffer Not Empty) is set to 1 on the last sampling clock edge. It indicates that the data is ready to be read from the SPI_DR register. If the RXNEIE bit in the SPI_CR2 register is set to 1, an interrupt can be triggered. Reading from the SPI_DR register clears the RXNE flag. In some configurations, the BSY flag can be used to wait for the transfer to complete during the last data transfer.

13.3.5.2 Full-Duplex Transmission and Reception Process in Master or Slave Mode (BIDIMODE=0 and RXONLY=0)

Software must follow the steps below to transmit and receive data (refer to Figures 13-8 and 13-9):

1. Enable SPI by setting the SPE bit to 1.
2. Write the first data item to the SPI_DR register (this operation clears the TXE flag).
3. Wait until TXE=1, then write the second data item. Afterward, wait until RXNE=1, and read the SPI_DR to get the first received data item (this operation clears the RXNE flag). Repeat this process for each data item to transmit/receive until the (n-1)-th received data.
4. Wait until RXNE=1, then read the last received data.
5. Wait until TXE=1, then wait until BSY=0, and disable SPI.

Additionally, this process can also be implemented using the interrupt service routines corresponding to the RXNE or TXE flags.

Figure 13-8 TXE/RXNE/BSY behavior in Master/Full-Duplex Mode (BIDIMODE=0 and RXONLY=0) during continuous transmission

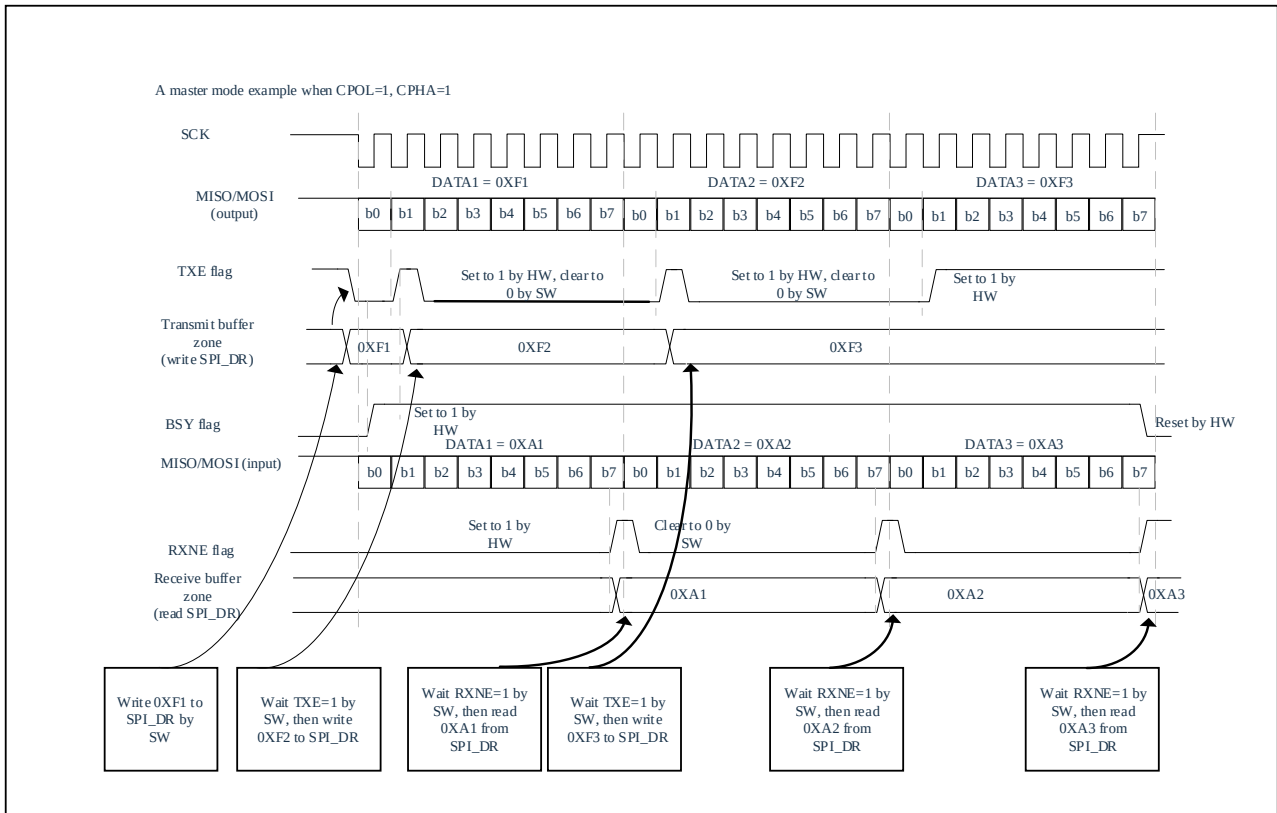
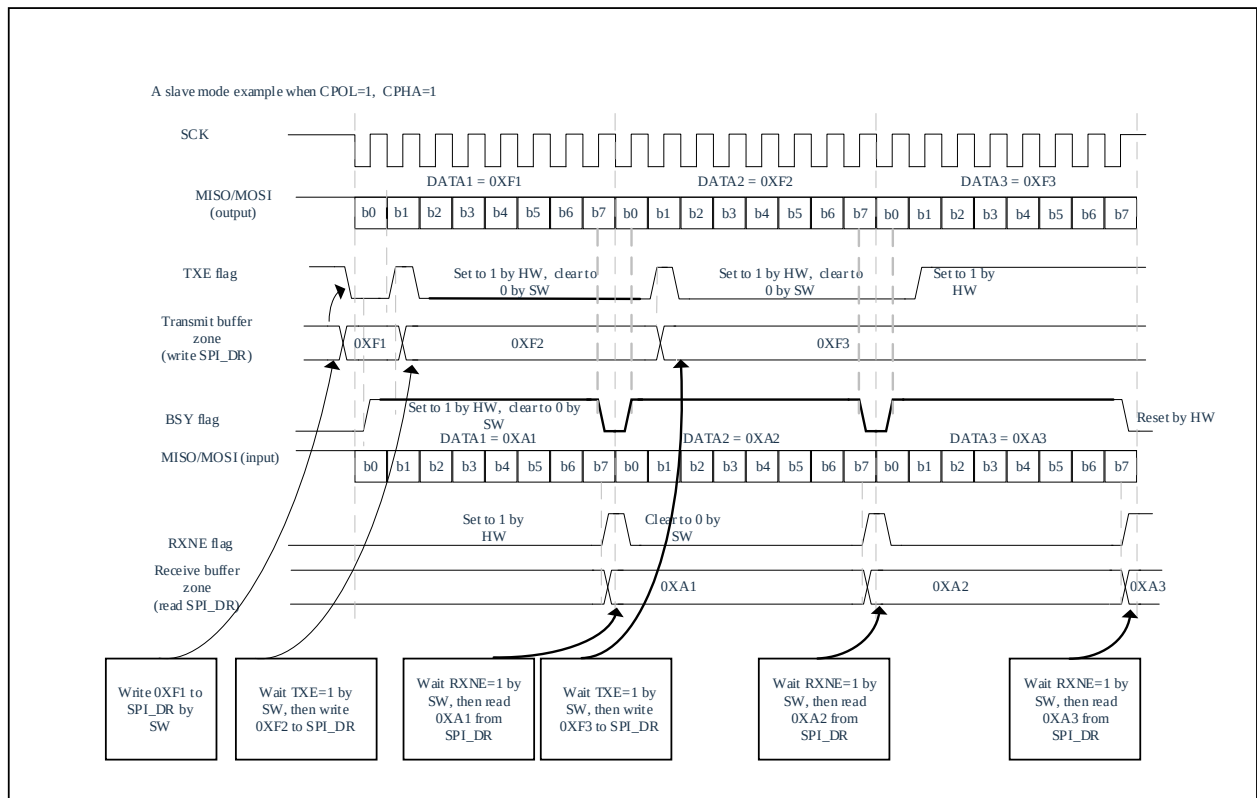


Figure 13-9 TXE/RXNE/BSY behavior in Slave/Full-Duplex Mode (BIDIMODE=0 and RXONLY=0) during continuous transmission



13.3.5.3 Data Transmission Process in Transmit-Only Mode (BIDIMODE=0, RXONLY=0)

In this mode, the process can be simplified as described below, and the BSY flag can be used to wait for the transmission to complete (refer to Figures 13-10 and 13-11).

1. Enable SPI by setting the SPE bit to 1.
2. Write the first data item to the SPI_DR register (this operation clears the TXE flag).
3. Wait until TXE=1, then write the next data item to be sent. Repeat this step for each data item to be sent.
4. After writing the last data item to the SPI_DR register, wait until TXE=1, then wait until BSY=0, indicating that the last data has been transmitted.

Additionally, this process can also be implemented using the interrupt service routine corresponding to the TXE flag.

Caution: During non-continuous communication, there is a delay of 2 APB clock cycles between writing to SPI_DR and the BSY flag being set to 1. Therefore, in transmit-only mode, after writing the last data, you must wait for TXE to be set to 1, and then wait for the BSY flag to clear. In transmit-only mode, after sending two data items, the OVR flag in the SPI_SR register will be set to 1, since the received data is never read.

Figure 13-10 TXE/BSY behavior in Master Transmit-Only Mode (BIDIMODE=0 and RXONLY=0) during continuous transmission.

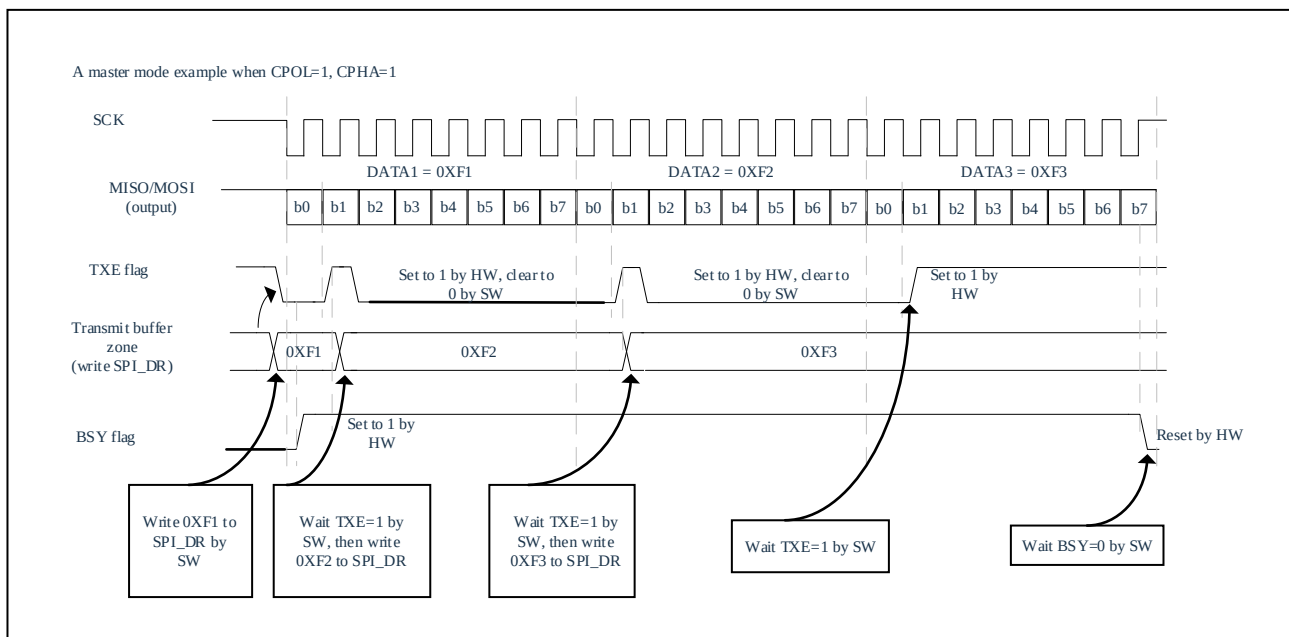
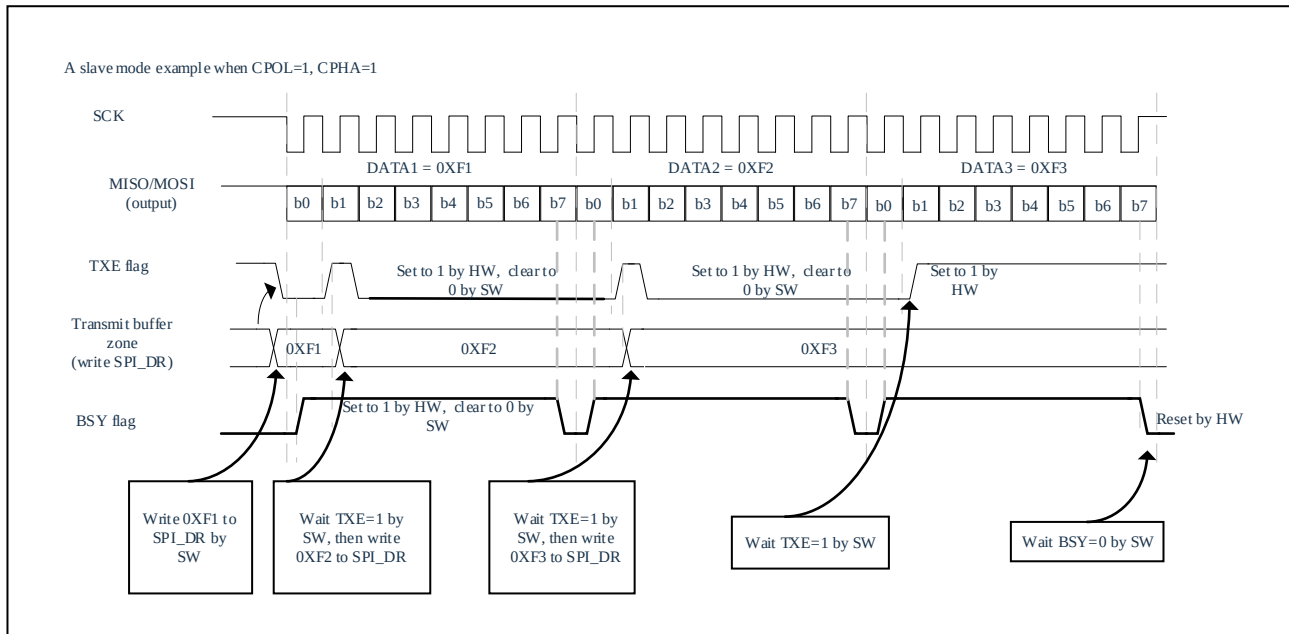


Figure 13-11 TXE/BSY behavior in Slave Transmit-Only Mode (BIDIMODE=0 and RXONLY=0) during continuous transmission



13.3.5.4 Transmission Process in Single-Wire Bi-Directional Mode (BIDIMODE=1 and BIDIOE=1)

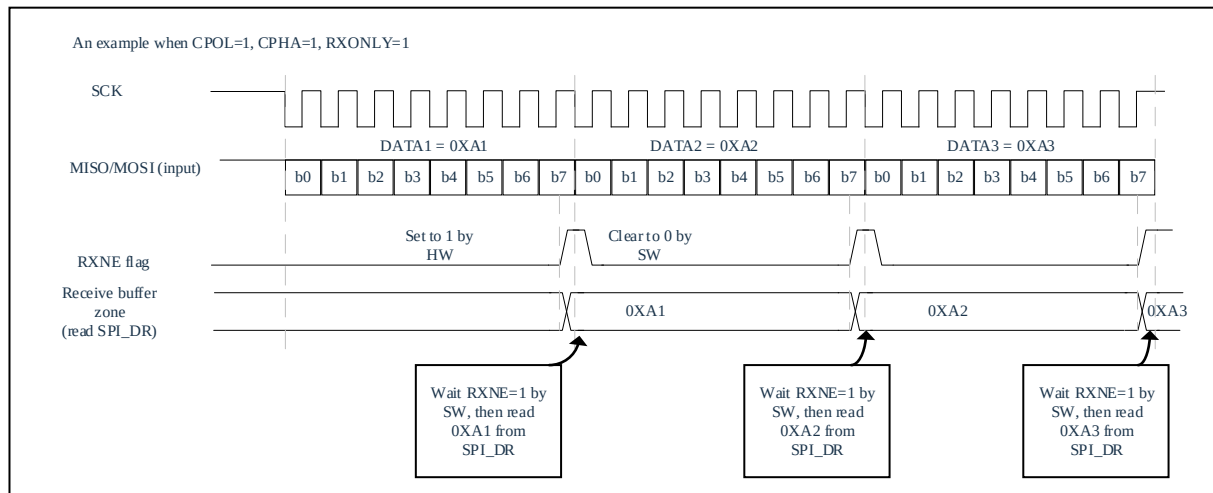
In this mode, the process is similar to the transmit-only mode, except that both the BIDIMODE and BIDIOE bits in the SPI_CR2 register must be set to 1 before enabling SPI.

13.3.5.5 Receive-Only Process in Single-Wire Mode (BIDIMODE=0 and RXONLY=1)

In this mode, the process can be simplified as described below (refer to Figure 13-12):

1. Set the RXONLY bit in the SPI_CR2 register to 1.
2. Enable SPI by setting the SPE bit to 1:
 - a) In master mode, this immediately activates the SCK clock generation and serially receives data until SPI is disabled (SPE=0).
 - b) In slave mode, data is received when the SPI master drives the NSS of the slave to low and outputs the SCK clock.
3. Wait until RXNE=1, then read the SPI_DR register to retrieve the received data (this operation clears the RXNE flag). Repeat this process for each data item to be received. Additionally, the process can be implemented using the interrupt service routine corresponding to the RXNE flag.

Figure 13-12 RXNE behavior in Receive-Only Mode (BIDIMODE=0 and RXONLY=1) during continuous transmission



13.3.5.6 Reception Process in Single-Wire Bi-Directional Mode (BIDIMODE=1 and BIDIOE=0)

In this mode, the process is similar to the receive-only mode, except that before enabling SPI, the BIDIMODE bit in the SPI_CR2 register must be set to 1, and the BIDIOE bit must be cleared (set to 0).

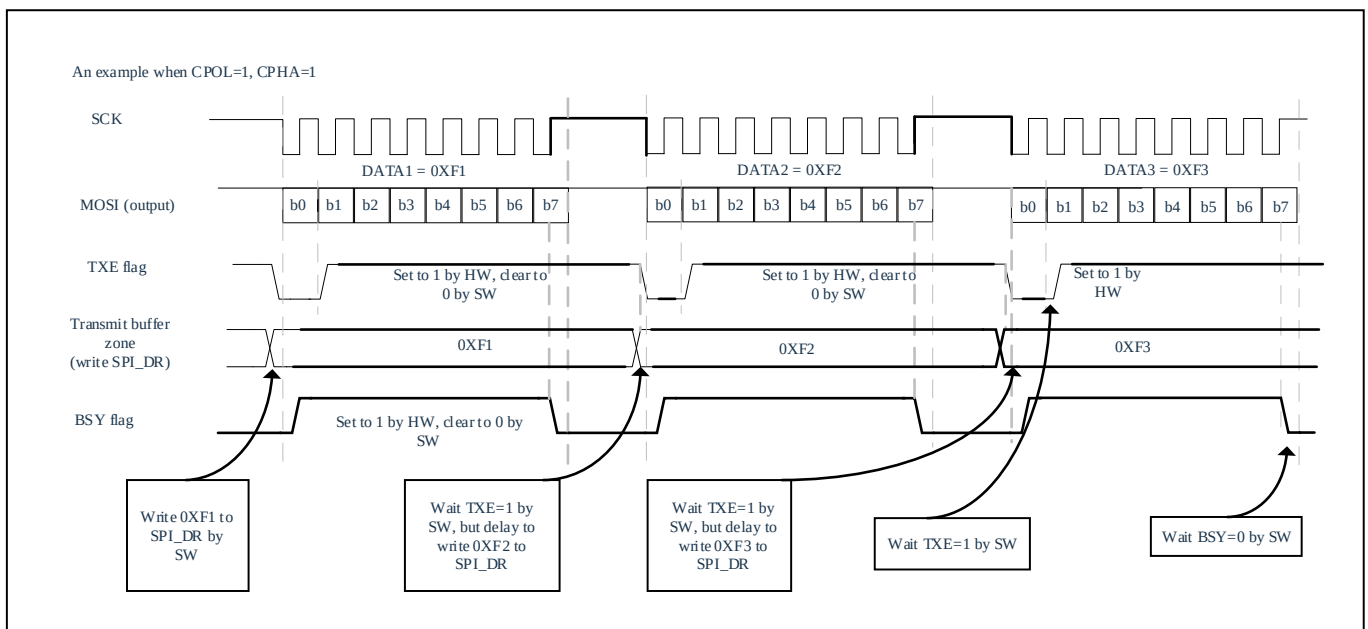
13.3.5.7 Continuous and Discontinuous Transmission

When transmitting data in master mode, if the software is fast enough to respond to the TXE interrupt triggered by the transfer of data from the data register to the shift register before the completion of the current data transfer, and can immediately write the next data to SPI_DR, this communication is called continuous communication. In this case, the generation of the SPI clock between data items will not be interrupted, and the BSY flag will not be cleared between data transfers.

In contrast, if the software is not fast enough, communication may be interrupted. In this case, the BSY flag will be cleared between data transfers (refer to Figure 13-13). In master mode with RXONLY=1 (receive-only mode), communication is always continuous, and the BSY flag always reads as 1.

In slave mode, the continuity of communication is determined by the SPI master device. In any case (even if communication is continuous), the BSY flag will briefly become low between data transfers for one SPI clock cycle (refer to Figure 13-11).

Figure 13-13 TXE/BSY behavior during transmission (BIDIMODE=0 and RXONLY=1) in the case of discontinuous transmission



13.3.6 CRC Calculation

To ensure communication reliability, the SPI module implements hardware CRC functionality. CRC calculation is performed separately for transmitted and received data. A programmable polynomial is used to calculate the CRC for each bit. The bits are sampled on the clock edges defined by the CPHA and CPOL bits in the SPI_CR1 register.

Caution: The SPI module provides two CRC calculation standards depending on the data frame format chosen for transmission and/or reception: 8-bit data uses CR8, and 16-bit data uses CRC16 (CRC16/XMODEM). CRC calculation is enabled by setting the CRCEN bit in the SPI_CR1 register. This operation resets the CRC registers (SPI_RXCRCR and SPI_TXCRCR). In full-duplex or transmit-only mode, if the transfer is managed by software (CPU mode), a write operation to the CRCNEXT bit must be performed immediately after the last data is written to SPI_DR. After the last data transfer, the value of SPI_TXCRCR will be transmitted.

In receive-only mode, if the transfer is managed by software (CPU mode), a write operation to the CRCNEXT bit must be performed after the second-to-last data is received. After the last data is received, the CRC will be obtained and a CRC check will be performed.

If data corruption occurs during transfer, the CRCERR flag in the SPI_SR register will be set to 1 when both data and CRC transfer ends.

If there is data in the transmission buffer, the CRC value will only be sent after the data byte has been transmitted. During CRC transmission, the CRC calculator is disabled, and the register value remains unchanged.

The steps to use CRC for SPI communication are as follows:

1. Program the CPOL, CPHA, LSBFirst, BR, SSM, SSI, and MSTR values.
2. Program the polynomial in the SPI_CRCPR register.
3. Enable CRC calculation by setting the CRCEN bit in the SPI_CR1 register. This operation also clears the SPI_RXCRCR and SPI_TXCRCR registers.
4. Enable SPI by setting the SPE bit in the SPI_CR1 register.
5. Start and maintain communication until only one byte or half-word remains to be transmitted or received:
 - In full-duplex or transmit-only mode, if the transfer is managed by software, set the CRCNEXT bit in the SPI_CR1 register after writing the last byte or half-word to the transmission buffer, indicating that the CRC will be sent after the last byte.
 - In receive-only mode, after receiving the second-to-last data, set the CRCNEXT bit immediately so that SPI is ready to enter the CRC stage after receiving the last data. During CRC transfer, CRC calculation will freeze.
6. After the last byte or half-word is transmitted, SPI enters the CRC transmission and verification stage. In full-duplex mode or receive-only mode, the received CRC is compared with the SPI_RXCRCR value. If the two values do not match, the CRCERR flag in the SPI_SR register will be set to 1, and an interrupt will occur if the ERRIE bit is set in the SPI_CR2 register.

Caution: When the SPI is in slave mode, be aware that CRC calculation can only be enabled when the clock is stable (i.e., the clock is at the idle level). Otherwise, CRC calculation errors may occur because the CRC calculator starts working as long as the CRCEN bit is set to 1, regardless of the SPE bit value, whenever there is a clock input.

When the SPI communication clock frequency is high, special care should be taken when sending the CRC. Since the CPU should remain as idle as possible during the CRC transmission stage, it is advised not to call functions during

this stage to avoid errors in receiving the last data and CRC. In fact, the CRCNEXT bit must be written before transmitting/receiving the last data.

For high SPI communication clock frequencies, it is recommended to use DMA mode to avoid performance degradation due to CPU access affecting SPI bandwidth.

If the device is configured as a slave and uses NSS hardware mode, the NSS pin must be kept low between the data and CRC phases.

When SPI is configured as a slave and CRC functionality is enabled, CRC calculation will still take place even when the NSS pin is high. This may occur in multi-slave mode, where the master device alternates addressing the slave devices.

During the switch of the chip select for the slave device, the CRC value should be cleared simultaneously at both the master and slave ends to re-synchronize the CRC calculation. To clear the CRC, follow these steps:

1. Disable SPI (SPE = 0)
2. Clear the CRCEN bit
3. Set the CRCEN bit to 1
4. Enable SPI (SPE = 1)

13.3.7 Status Flags

Applications can monitor the status of the SPI bus through three status flags.

13.3.7.1 Transmit Buffer Empty (TXE)

When this flag is set to 1, it indicates that the transmit buffer is empty, and the next data can be loaded into the buffer. The TXE flag is cleared when a write operation to the SPI_DR register is performed.

13.3.7.2 Receive Buffer Not Empty (RXNE)

When this flag is set to 1, it indicates that there is valid received data in the receive buffer. The flag is cleared when reading from the SPI_DR register.

BUSY Flag

The BSY flag is set and cleared by hardware (writing to this flag has no effect). The BSY flag is used to indicate the state of the SPI communication.

When BSY is set to 1, it means SPI is busy with communication.

If software needs to stop the SPI and enter the stop mode (or turn off the peripheral clock), it can use the BSY flag to check if the transfer has ended to avoid corrupting the last data transfer. To do this, the following steps must be strictly followed.

The BSY flag can also be used to avoid write conflicts in multi-master systems.

The BSY flag is set to 1 when the transfer begins.

Hardware clears the BSY flag in the following situations:

- When the transfer is complete (except in continuous communication in master mode)
- When the SPI is turned off

When communication is not continuous, the BSY flag remains low between communications.

When communication is continuous:

- In master mode, the BSY flag stays high during the entire transfer.
- In slave mode, the BSY flag becomes low for one SPI clock cycle between transfers.

Caution: Do not use the BSY flag to handle each data transmission or reception. It is better to use the TXE and RXNE flags instead.

13.3.8 Procedure for Disabling the SPI

When the transfer ends, the application can stop communication by disabling the SPI peripheral. This is done by clearing the SPE bit.

For some configurations, disabling the SPI during an ongoing transfer and entering stop mode may cause the current transfer to be corrupted, and/or the BSY flag may become unreliable.

To avoid these issues, it is recommended to follow the steps below when disabling the SPI:

In Master mode or Full-Duplex Slave mode (BIDIMODE=0, RXONLY=0):

1. Wait until RXNE=1 to receive the last data.
2. Wait until TXE=1.
3. Then wait until BSY=0.
4. Disable SPI (SPE=0), and finally enter stop mode (or disable the peripheral clock).

In Master mode or Simplex Transmit-only Slave mode (BIDIMODE=0, RXONLY=0) or Bidirectional Transmit mode (BIDIMODE=1, BIDIOE=1):

After writing the last data to the SPI_DR register:

1. Wait until TXE=1.
2. Then wait until BSY=0.
3. Disable SPI (SPE=0), and finally enter stop mode (or disable the peripheral clock).

In Simplex Receive-only Master mode (MSTR=1, BIDIMODE=0, RXONLY=1) or Bidirectional Receive mode (MSTR=1, BIDIMODE=1, BIDIOE=0):

This case must be handled specially to avoid unnecessary SPI data transfer. The following sequence applies only to SPI Motorola configuration (FRF bit = 0):

1. Wait until the RXNE flag corresponding to the second last data (the (n-1)th data) is set.
2. Then wait one SPI clock cycle (using a software loop) before disabling SPI (SPE=0).
3. Finally, wait until the last RXNE=1, then enter stop mode (or disable the peripheral clock).

When SPI is configured in TI mode (FRF bit = 1): To avoid unwanted pulses on NSS when disabling SPI, follow these steps:

1. Wait until the RXNE flag for the second last data (the (n-1)th data) is set.
2. Disable SPI (SPE=0) using a software loop within the following window frame:
 - After at least one SPI clock cycle,
 - But before the LSB data starts transmitting.

Caution: In bidirectional receive master mode (MSTR=1, BDM=1, BDOE=0), the BSY flag remains low during transfer.

In Receive-only Slave mode (MSTR=0, BIDIMODE=0, RXONLY=1) or Bidirectional Receive mode (MSTR=0, BIDIMODE=1, BDOE=0):

1. SPI can be disabled anytime (write SPE=0). SPI will actually shut down only after the current transfer completes.
2. Afterwards, if entering stop mode, you must first wait until BSY=0 before entering stop mode (or disabling the peripheral clock).

13.3.9 Using DMA (Direct Memory Access) for SPI Communication

To operate at maximum speed, the SPI needs a continuous supply of data to be transmitted and timely reading of data from the receive buffer to avoid overrun. To accelerate the transfer, SPI provides DMA functionality to implement a simple request/response protocol. When the enable bit in the SPI_CR2 register is set, DMA access is requested. The transmit buffer and receive buffer each generate their own DMA request (see Figures 13-14 and 13-15):

- During transmission, each time TXE is set to 1, a DMA request is generated. DMA then writes to the SPI_DR register (this operation clears the TXE flag).
- During reception, each time RXNE is set to 1, a DMA request is generated. DMA then reads from the SPI_DR register (this operation clears the RXNE flag).

When SPI is used for transmission only, you can enable only the SPI Tx DMA channel. In this case, the OVE flag is set to 1, because the received data was not read.

When SPI is used for reception only, you can enable only the SPI Rx DMA channel. In transmission mode, after DMA has completed the transfer of all the data to be sent (when the TCIF flag in the DMA_ISR register is set to 1), you can monitor the BSY flag to ensure SPI communication has completed. This step must be performed before disabling SPI or entering stop mode to avoid corrupting the final transmission of data. The software must first wait until TXE=1, and then wait until BSY=0.

Caution: During non-continuous communication, there is a delay of two APB clock cycles between writing to the SPI_DR register and BSY being set to 1. Therefore, after writing the last data, you must first wait for TXE=1 and then wait for BSY=0.

Figure 13-14 Using DMA for transmission

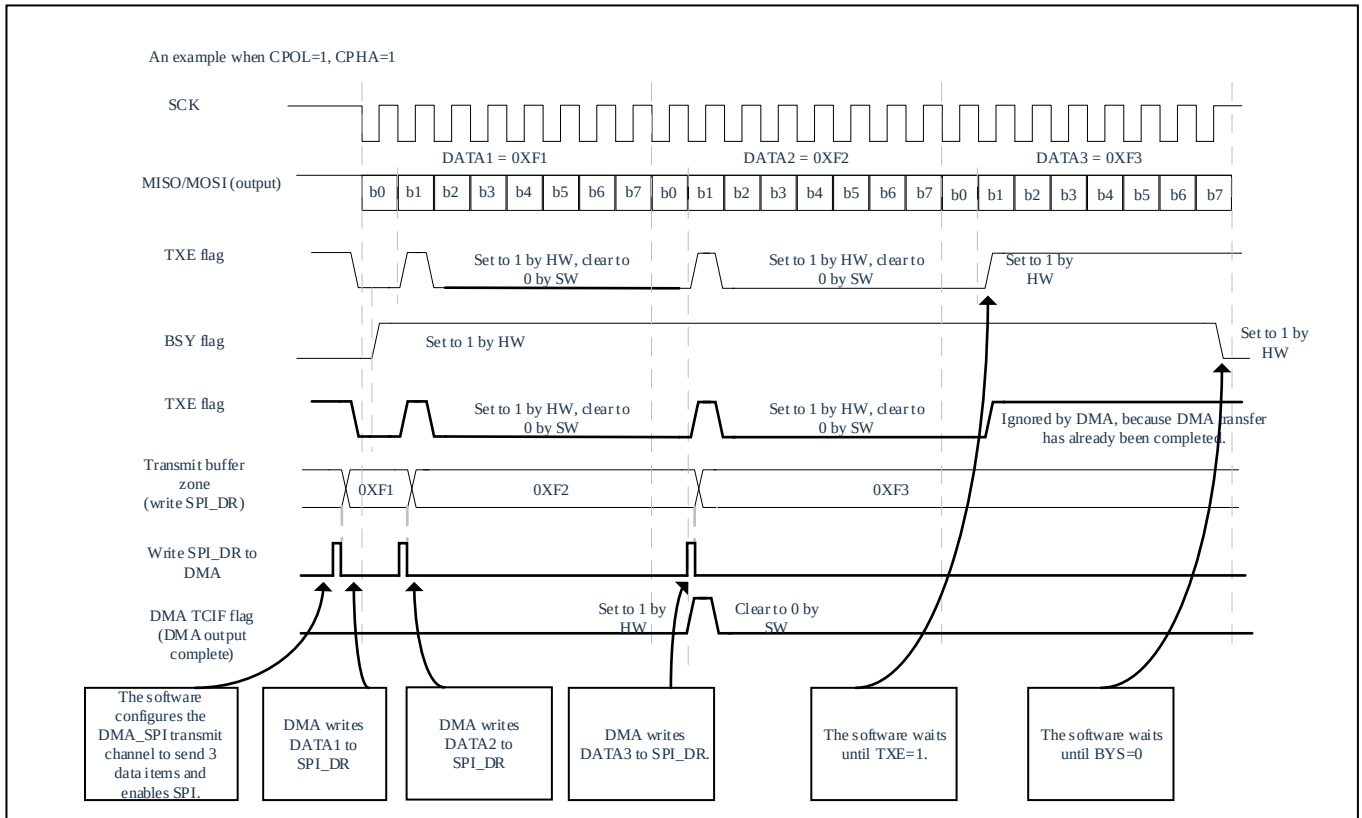
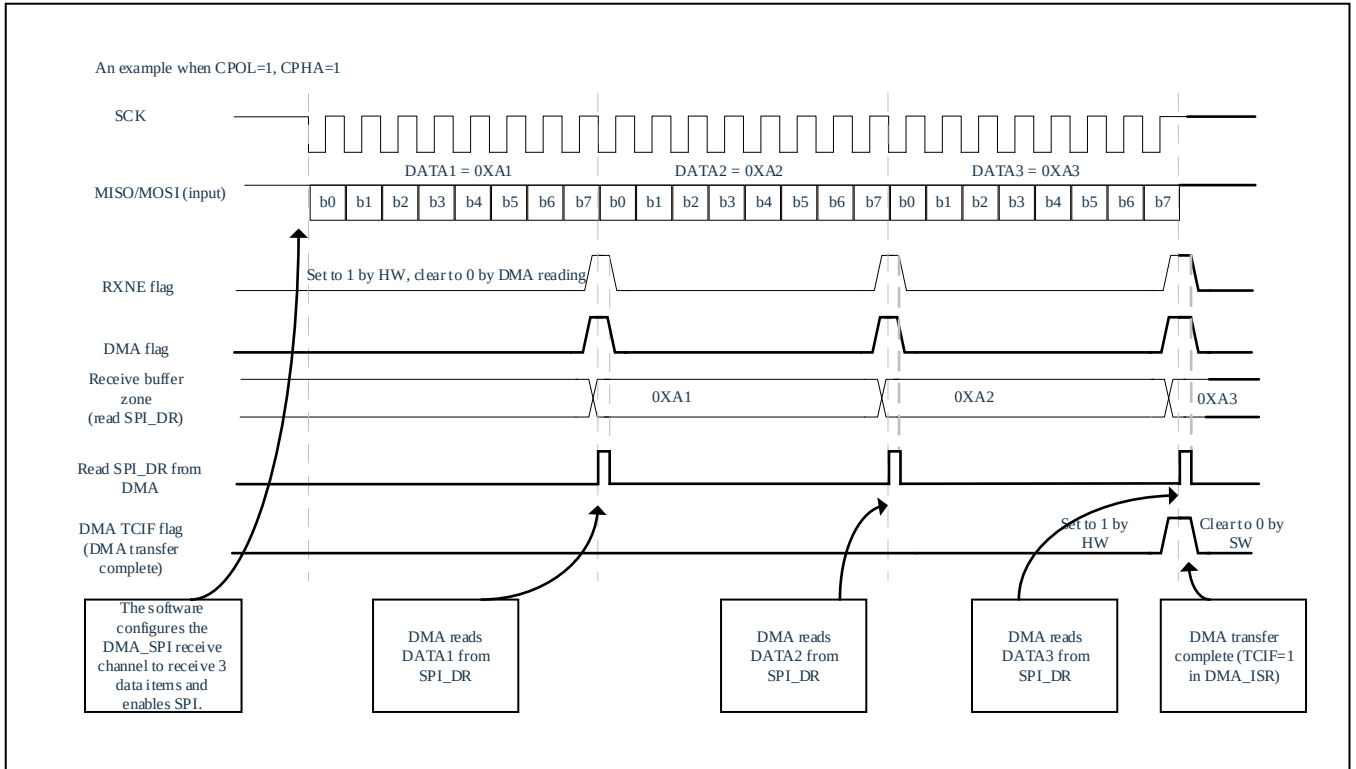


Figure 13-15 Using DMA for reception



13.3.9.1 DMA Function and CRC

When SPI communication with CRC and DMA mode is enabled, the CRC will be automatically transmitted and received at the end of the communication, without needing to use the CRCNEXT bit. After receiving the CRC, the CRC must be read from the SPI_DR register to clear the RXNE flag.

If a transfer error occurs, the CRCERR flag in the SPI_SR register will be set to 1 at the end of the data and CRC transfer.

13.3.10 Error Flags

Master Mode Fault (MODF)

A master mode fault occurs when the NSS pin of the master device is pulled low (in NSS hardware mode) or when the SSI bit is 0 (in NSS software mode). This automatically sets the MODF bit to 1. A master mode fault affects the SPI peripheral in the following ways:

- If ERRIE is set to 1, the MODF bit is set to 1, and an SPI interrupt is generated.
- The SPE bit is cleared, which disables all outputs of the device and shuts down the SPI interface.
- The MSTR bit is cleared, forcing the device to enter slave mode.

The MODF bit can be cleared using the following software sequence:

1. When MODF is set to 1, perform a read or write access to the SPI_SR register.
2. Then, write to the SPI_CR1 register.

To avoid conflicts in multi-MCU systems, the NSS pin must be pulled high during the MODF bit clear sequence. After this sequence, the SPE and MSTR bits can be restored to their original states.

As a safety measure, the hardware does not allow SPE and MSTR to be set to 1 when MODF is set to 1.

In slave devices, the MODF bit cannot be set. However, in multi-master configurations, the device can be in slave mode when MODF is set to 1. In such cases, the MODF bit indicates a possible multi-master conflict in the system. Interrupt handlers can fully recover from this state by performing a reset or returning to the default state.

Overrun error

An overrun occurs when the master device has finished sending a data byte, but the slave device has not yet cleared the RXNE flag for the previously received data. When an overrun error occurs:

- The OVR bit is set to 1, and an interrupt is generated if ERRIE is set to 1.

In this case, the receiver buffer will not be updated with new data from the master. Reading from the SPI_DR register will return the byte that caused the overrun. All subsequent bytes sent by the master device will be lost.

The OVR flag can be cleared by sequentially reading the SPI_DR and SPI_SR registers.

CRC Error

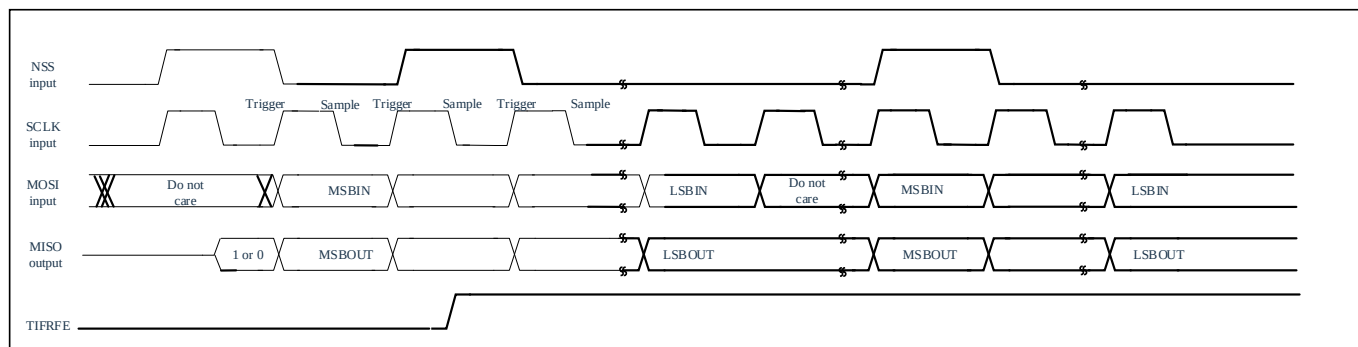
When the CRCEN bit in the SPI_CR1 register is set to 1, this flag is used to verify the validity of the received data. If the value received in the shift register does not match the value in the SPI_RXCRCR register, the CRCERR flag in the SPI_SR register will be set to 1.

TI Mode Frame Format Error

If SPI is operating in slave mode and is configured to follow the TI mode protocol, a TI mode frame format error will be detected if an NSS pulse occurs during ongoing communication. When this error occurs, the FRE flag in the SPI_SR register will be set to 1. The SPI will not shut down upon error detection, but it will ignore the NSS pulse and wait for the next NSS pulse before starting a new transfer. Since error detection may result in the loss of two data bytes, data may be corrupted.

Reading the SPI_SR register will clear the FRE flag. If ERRIE is set to 1, an interrupt will be generated when a frame format error is detected. In this case, due to the lack of data continuity, SPI should be disabled, and after re-enabling SPI, the master device should initiate communication again.

Figure 13-16 TI mode frame format error detection



13.3.11 SPI Interrupts

Table 13-1: SPI interrupt requests

Interrupt event	Event flag	Enable control bit
Transmit buffer empty	TXE	TXEIE
Receive buffer non-empty	RXNE	PXNEIE
Master mode fault	MODF	ERRIE
Overrun error	OVR	
CRC error	CRCERR	
TI frame format error	FRE	ERRIR

13.4 SPI Register Mapping

(SPI base address = 0x4007_3000)

RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
SPI_CR1	0x000	R/W	SPI Control Register 1	0x0
SPI_CR2	0x004	R/W	SPI Control Register 2	0x0
SPI_SR	0x008	R/W	SPI Status Register	0x02
SPI_DR	0x00C	R/W	SPI Data Register	0x0
SPI_CRCPR	0x010	R/W	SPI CRC Polynomial Register	0x07
SPI_RXCRCR	0x014	RO	SPI RX CRC Register	0x0
SPI_TXCRCR	0x018	RO	SPI TX CRC Register	0x0

13.5 SPI Register Description

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

13.5.1 SPI Control Register 1 (SPI_CR1)

Bit	Symbol	Description	Reset value
15	BIDIMODE	Bidirectional data mode enable: 0: 2-line unidirectional data mode selected 1: 1-line bidirectional data mode selected	0
14	BIDIOE	Output enable in bidirectional mode: 0: Output disabled (receive-only mode) 1: Output enabled (transmit-only mode) This bit combined with the BIDIMODE bit selects the direction of transfer in bidirectional mode. Caution: In master mode, the MOSI pin is used and in slave mode, the MISO pin is used.	0
13	CRCEN	Hardware CRC calculation enable: 0: CRC calculation disabled 1: CRC calculation enabled Caution: This bit should be written only when SPI is disabled (SPE = '0') for correct operation.	0
12	CRCNEXT	CRC transfer next: 0: Data phase (no CRC phase) 1: The next transfer is CRC (CRC phase) Caution: When SPI is configured in full-duplex or transmit-only mode, a write operation to CRCNEXT must be performed as soon as the last data is written to the SPI_DR register. When SPI is configured in receive-only mode, CRCNEXT must be set to 1 after receiving the second-to-last data. When the transfer is managed by DMA, this bit should remain cleared.	0
11	DFB	Data frame format: 0: Select 8-bit data frame format for transmission/reception 1: Select 16-bit data frame format for transmission/reception Caution: To ensure proper operation, this bit should only be written when SPI is disabled (SPE = 0).	0
10	RXONLY	Receive only 0: Full-duplex (transmit and receive) 1: Disable output (receive-only mode) This bit, in conjunction with the BIDIMODE bit, is used to select the transfer direction in the two-wire unidirectional mode. This bit also applies to multi-slave mode systems, where specific slave devices are not accessed and the outputs of accessed slave devices are not affected.	0
9	SSM	Software slave management: 0: Disable software slave management 1: Enable software slave management When SSM is set to 1, the NSS pin input is replaced by the value of the SSI bit. Not applicable in SPI TI mode.	0
8	SSI	Internal slave select: This bit is only valid when SSM is set to 1. The value of this bit will affect the NSS pin, ignoring the I/O of the NSS pin. Not applicable in SPI TI mode.	0
7	LSBFIRST	Frame format: 0: Transmit MSB first 1: Transmit LSB first Caution: Do not change this bit while communication is ongoing. Not applicable in SPI TI mode.	0
6	SPE	SPI enable: 0: Disable peripheral 1: Enable peripheral	0

		Caution: When disabling SPI, follow the steps outlined in Section 13.3.8 for disabling SPI.	
5:3	BR	Baud rate control: 000: fPCLK/2 100: fPCLK/32 001: fPCLK/4 101: fPCLK/64 010: fPCLK/8 110: fPCLK/128 011: fPCLK/16 111: fPCLK/256 Caution: Do not change these bits while communication is ongoing.	0x0
2	MSTR	Master selection: 0: Slave configuration 1: Master configuration Caution: Do not change this bit while communication is ongoing.	0
1	CPOL	Clock polarity: 0: SCK stays low when idle 1: SCK stays high when idle Caution: Do not change this bit while communication is ongoing.	0
0	CPHA	Clock phase: 0: Sample data on the first clock edge 1: Sample data on the second clock edge Caution: Do not change this bit while communication is ongoing.	0

13.5.2 SPI Control Register 2 (SPI_CR2)

Bit	Symbol	Description	Reset value
15:8	-	Reserved	0x0
7	TXEIE	Tx buffer empty interrupt enable: 0: TXE interrupt masked 1: TXE interrupt not masked. Used to generate an interrupt request when the TXE flag is set.	0
6	RXNEIE	RX buffer not empty interrupt enable: 0: RXNE interrupt masked 1: RXNE interrupt not masked. Used to generate an interrupt request when the RXNE flag is set.	0
5	ERRIE	Error interrupt enable: 0: Error interrupt is masked 1: Error interrupt is enabled This bit controls the generation of an interrupt when an error condition occurs (CRCERR, OVR, MODF in SPI mode, UDR, OVR and FRE).	0
4	FRF	Frame format: 0: SPI Motorola mode 1: SPI TI mode	0
3	-	Reserved	0
2	SSOE	SS output enable: 0: In master mode, disable SS output, can work in multi-master mode configuration. 1: In master mode, enable SS output, cannot work in multi-master mode.	0
1	TXDMAEN	Tx buffer DMA enable: 0: Disable transmit buffer DMA. 1: Enable transmit buffer DMA. When this bit is set to 1, a DMA request is generated every time the TXE flag is set to 1.	0
0	RXDMAEN	Rx buffer DMA enable 0: Disable receive buffer DMA. 1: Enable receive buffer DMA. When this bit is set to 1, a DMA request is generated every time the RXNE flag is set to 1.	0

13.5.3 SPI Status Register (SPI_SR)

Bit	Symbol	Description	Reset value
15: 9	-	Reserved	0x0
8	FRE	Frame format error: 0: No frame format error 1: A frame format error occurred This flag is set by hardware and reset when SPIx_SR is read by software.	0
7	BSY	Busy flag: 0: SPI not busy 1: SPI is busy in communication or Tx buffer is not empty This flag is set and cleared by hardware. Caution: The BSY flag must be used with caution.	0
6	OVR	Overrun flag: 0: No overrun occurred 1: Overrun occurred This flag is set by hardware and reset by a software sequence.	0
5	MODF	Mode fault: 0: No mode fault occurred 1: Mode fault occurred This flag is set by hardware and reset by a software sequence.	0
4	CRCERR	CRC error flag: 0: CRC value received matches the SPIx_RXCRCR value 1: CRC value received does not match the SPIx_RXCRCR value This flag is set by hardware and cleared by software writing 0.	0
3:2	-	Reserved	0x0
1	TXE	Transmit buffer empty: 0: Tx buffer not empty 1: Tx buffer empty	1
0	RXNE	Receive buffer not empty: 0: Rx buffer empty 1: Rx buffer not empty	0

13.5.4 SPI Data Register (SPI_DR)

Bit	Symbol	Description	Reset value
15:0	DR	Data register Data received or to be transmitted	0x0

The data register is divided into two buffers: one for writing (transmit buffer) and one for reading (receive buffer). When a write operation is performed on the data register, the data is written to the transmit buffer. When reading from the data register, the value from the receive buffer is returned.

For SPI mode: The transmitted or received data can be 8 bits or 16 bits, depending on the data frame format selection bit (DFF in the SPI_CR1 register). This selection must be made before enabling SPI to ensure correct operation. For an 8-bit data frame, the buffer is 8 bits wide, and only the LSB (SPI_DR[7:0]) of the register is used for transmission/reception. In receive mode, the MSB (SPI_DR[15:8]) of the register is forced to 0. For a 16-bit data frame, the buffer is 16 bits wide, and the entire register (SPI_DR[15:0]) is used for transmission/reception.

13.5.5 SPI CRC Polynomial Register (SPI_CRCPR)

Bit	Symbol	Description	Reset value
15:0	CRCPOLY	CRC polynomial register This register contains the polynomial for the CRC calculation.	0x0007

The CRC polynomial (0x0007) is the reset value of this register. Another polynomial can be configured as required.

13.5.6 SPI RX CRC Register (SPI_RXCRCR)

Bit	Symbol	Description	Reset value
15:0	RXCRC	Rx CRC register	0x0

When CRC calculation is enabled, the RxCRC[15:0] bits contain the computed CRC value of the subsequently received bytes. This register is reset when the CRCEN bit in SPI_CR1 register is written to 1. The CRC is calculated serially using the polynomial programmed in the SPI_CRCPR register.

Only the 8 LSB bits are considered when the CRC frame format is set to be 8-bit length (CRCL bit in the SPI_CR1 is cleared). CRC calculation is done based on any CRC8 standard.

The entire 16-bits of this register are considered when a 16-bit CRC frame format is selected (CRCL bit in the SPI_CR1 register is set). CRC calculation is done based on any CRC16 standard.

Caution: A read to this register when the BSY Flag is set could return an incorrect value.

13.5.7 SPI TX CRC Register (SPI_TXCRCR)

Bit	Symbol	Description	Reset value
15:0	TXCRC	Tx CRC register	0x0

When CRC calculation is enabled, the TxCRC[7:0] bits contain the computed CRC value of the subsequently transmitted bytes. This register is reset when the CRCEN bit of SPI_CR1 is written to 1. The CRC is calculated serially using the polynomial programmed in the SPI_CRCPR register.

Only the 8 LSB bits are considered when the CRC frame format is set to be 8-bit length (CRCL bit in the SPI_CR1 is cleared). CRC calculation is done based on any CRC8 standard.

The entire 16-bits of this register are considered when a 16-bit CRC frame format is selected (CRCL bit in the SPI_CR1 register is set). CRC calculation is done based on any CRC16 standard.

Caution: A read to this register when the BSY flag is set could return an incorrect value.

Chapter 14 I²C Interface

14.1 Introduction to I²C

The I²C (Inter-Integrated Circuit) bus interface is used as an interface between a microcontroller and the I²C serial bus. It provides multi-master mode functionality, allowing control over I²C-specific sequences, protocols, arbitration, and timing. It supports both standard and fast modes and is also compatible with SMBus 2.0. It can be used for a variety of applications, including CRC generation and verification, SMBus (System Management Bus), and PMBus (Power Management Bus). Depending on the device, DMA functionality can be used to reduce the CPU workload.

14.2 I²C Main Features

- Parallel bus/I²C protocol converter
- Multi-master mode functionality: The same interface can be used in both master and slave modes
- I²C master mode features:
 - Clock generation
 - Start and stop bit generation
- I²C Slave Mode Features:
 - Programmable I²C address detection
 - Dual addressing mode, capable of responding to two slave addresses
 - Stop bit detection
- 7-bit/10-bit addressing and generation and detection of broadcast calls
- Supports different communication speeds:
 - Standard speed (up to 100 KHz)
 - Fast speed (up to 400 KHz)
- Status flags:
 - Transmit/receive mode flag
 - Byte transfer complete flag
 - I²C busy flag
- Error flags:
 - Arbitration lost in master mode
 - Acknowledge failure after address/data transfer
 - Detection of invalid start/stop bits
 - Overflow/underrun after clock stretching is disabled
- Two interrupt vectors:
 - One interrupt triggered by successful address/data byte transfer events
 - One interrupt triggered by error states
- Optional clock stretching
- 1-byte buffer with DMA functionality
- Configurable PEC (Packet Error Check) generation or verification:
 - In Tx mode, the PEC value can be sent as the last byte

- PEC error checking for the last received byte
- SMBus 2.0 Compatibility:
 - 25 ms clock low timeout delay
 - 10 ms master device cumulative clock low extension time
 - 25 ms slave device cumulative clock low extension time
 - Hardware PEC generation/verification with ACK control
 - Supports Address Resolution Protocol (ARP)
- PMBus Compatibility

Caution: Some of the features listed above may not be available on certain products. Users should refer to the product datasheet to determine the specific features supported by the I²C interface implementation.

14.3 I²C Functional Description

In addition to receiving and transmitting data, this interface can also convert between serial and parallel formats, and vice versa. Interrupts can be enabled or disabled by software. The interface connects to the I²C bus via the data (SDA) and clock (SCL) pins. It can be connected to either a standard (up to 100 KHz) or a fast (up to 400 KHz) I²C bus.

14.3.1 Mode Selection

The interface can operate in one of the following four modes:

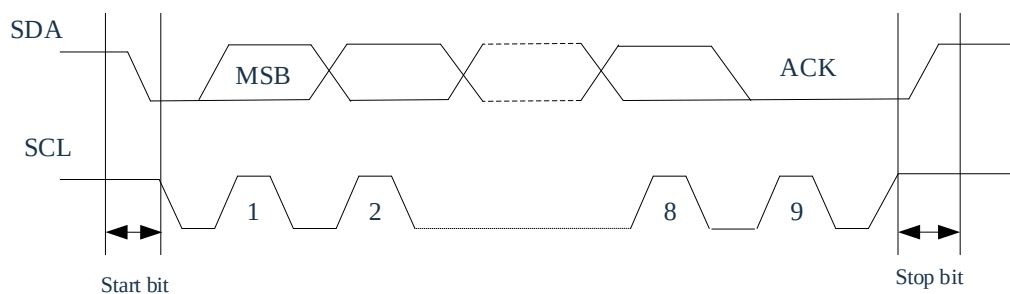
- Slave transmitter
- Slave receiver
- Master transmitter
- Master receiver

By default, it operates in slave mode. After generating the start bit, the interface automatically switches from slave mode to master mode, and switches back to slave mode upon arbitration loss or stop bit generation, enabling multi-master functionality.

Communication Flow

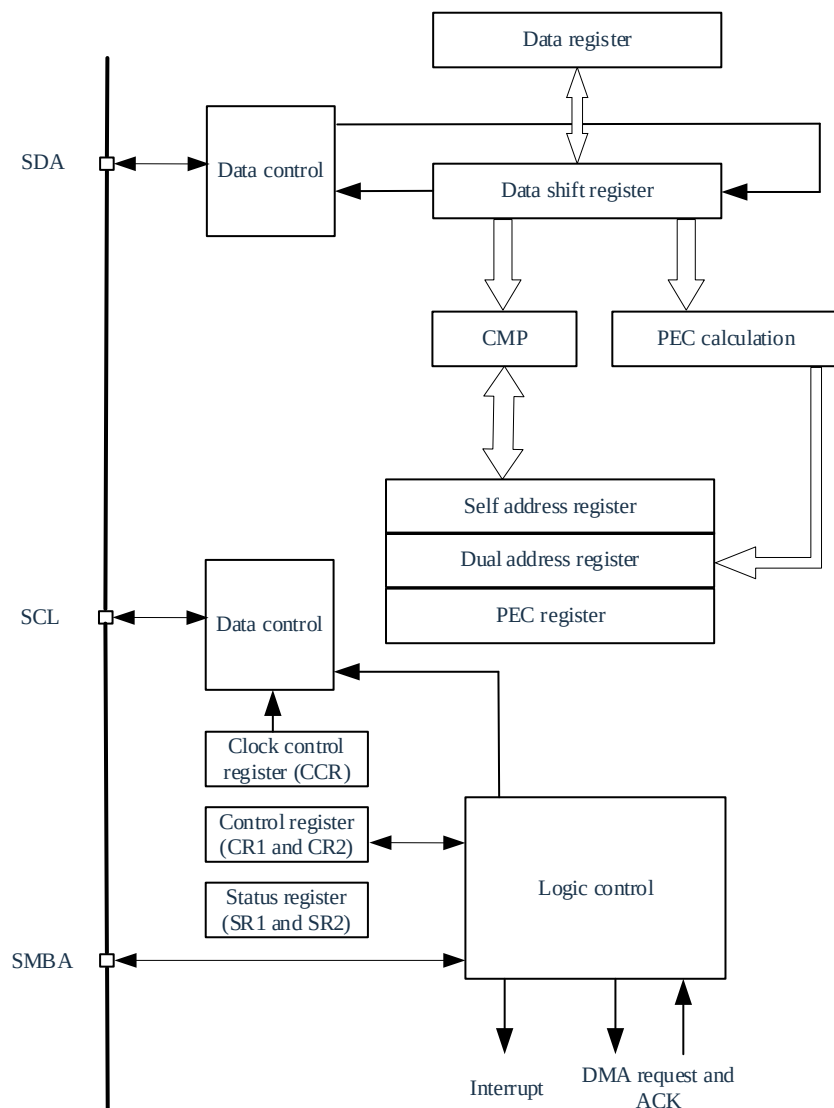
In master mode, the I²C interface initiates data transmission and generates the clock signal. Serial data transfer always begins with the start bit and ends with the stop bit. Both the start and stop bits are generated by software in master mode. In slave mode, the interface can recognize its own address (7-bit or 10-bit) as well as broadcast call addresses. The detection of the broadcast call address can be enabled or disabled by software. Data and addresses are transmitted as 8-bit bytes, with the MSB transmitted first. After the start bit, the address byte follows (a 7-bit address occupies one byte; a 10-bit address occupies two bytes). The address is always transferred in master mode. After 8 clock cycles of byte transfer, the 9th clock pulse occurs, during which the receiver must send an acknowledge bit to the transmitter. Refer to Figure 14-1.

Figure 14-1 IIC bus protocol



The acknowledge bit can be enabled or disabled by software. The I²C interface address (7-bit/10-bit dual-address mode and/or broadcast call address) can be selected by software. The I²C interface block diagram is shown in Figure 14-2.

Figure 14-2 IIC block diagram



14.3.2 I²C Slave Mode

By default, the I²C interface operates in slave mode. To switch from the default slave mode to master mode, a start bit must be generated. For proper timing, the external clock input for the peripheral must be programmed in the I2C_CR2 register. The minimum frequency for the external clock input is:

- 2 MHz for standard mode
- 4 MHz for fast mode

After detecting the start bit, the address from the SDA line is immediately received and passed to the shift register. It is then compared with the interface address (OAR1) and OAR2 (if ENDUAL=1), or a broadcast address (if ENGC=1).

Caution: In 10-bit addressing mode, the comparison object also includes the header sequence (11110xx0), where xx represents the two most significant bits of the address.

If the header or address doesn't match, the interface will ignore it and wait for the next start bit.

If the header matches (only for 10-bit mode), and the ACK bit is 1, the interface will generate an acknowledgment pulse and wait for the 8-bit slave address.

If the address matches, the interface will:

- Transmit an acknowledgment pulse (if the ACK bit is 1)
- The ADDR bit will be set to 1, and an interrupt will be generated when ITEVFEN is 1.
- If ENDUAL=1, the software must read the DUALF bit to check which slave addresses acknowledged the request.

In 10-bit mode, after the address sequence is received, the slave mode will always remain in receive mode. Upon receiving a repeated start bit with a matching address and the least significant bit set to 1 (11110xx1), it will switch to transmit mode.

The TRA bit indicates whether the slave device is in receive or transmit mode.

Slave Transmitter

After receiving the address and clearing the ADDR bit, the slave device will send the byte from the DR register to the SDA line via the internal shift register.

The slave will extend the SCL low time until the ADDR bit is cleared and the DR register is filled with data to send (see Figure 14-3 Transfer Sequence EV1 EV3).

When the acknowledgment pulse is received:

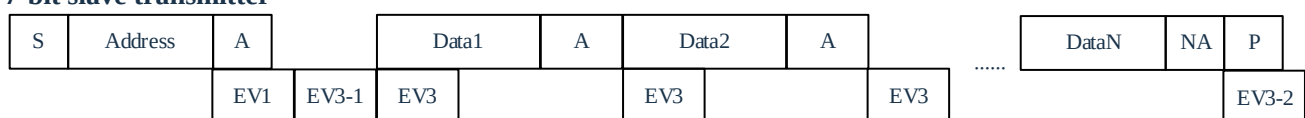
- The TxE bit is set to 1 by hardware and an interrupt will be generated when both ITEVFEN and ITBUFEN are set to 1. If the TxE bit is set to 1 before the next data transfer finishes, but some data has not been written to the I2C_DR register, the BTF bit will be set to 1, and the interface will continue to extend the SCL low time until the BTF is cleared by software via a read operation on I2C_SR1 and a write operation to I2C_DR.

1. EV1 and EV3_1 events may extend the SCL low time until the corresponding software sequence finishes.

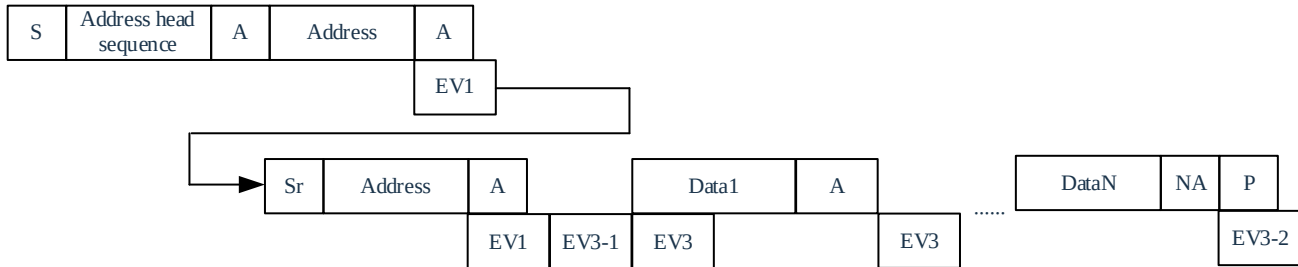
2. If the software sequence is not completed before the next byte transfer ends, the EV3 event will extend the SCL low time.

Figure 14-3 Slave transmitter transfer sequence

7-bit slave transmitter



10-bit slave transmitter



Note: S = Start bit, Sr = Repeated start bit, P = Stop bit, A = Acknowledge, NA = No Acknowledge

EVx = Event (interrupt occurs if ITEVFEN=1)

EV1: ADDR=1, cleared by reading SR1 then SR2.

EV3-1: TxE =1, shift register is empty, data register is empty, write Data1 to DR.

EV3: TxE =1, shift register is not empty, data register is empty, cleared by writing to DR.

EV3-2: AF=1: cleared AF by writing "0" to the AF bit in the SR1 register.

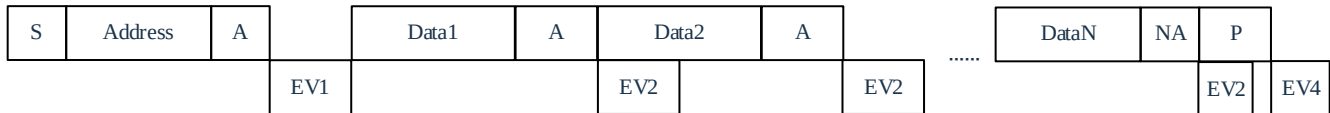
Slave Receiver

After receiving the address and clearing the ADDR bit, the slave device will receive bytes from the SDA line via the internal shift register and store them in the DR register. After each byte is received, the interface will:

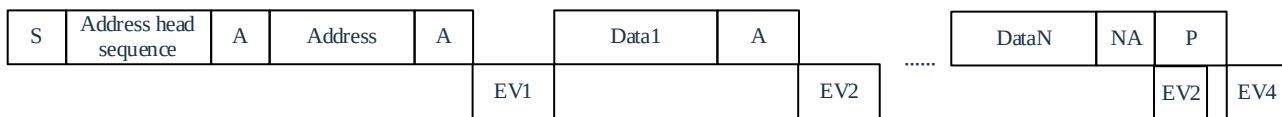
- Transmit an acknowledgment pulse (if the ACK bit is 1)
- The RxNE bit will be set to 1 by hardware and an interrupt will be generated when both ITEVFEN and ITBUFEN are set to 1. If the RxNE bit is set to 1 before the next data reception ends, but the DR register has not been read, the BTF bit will be set to 1, and the interface will continue to extend the SCL low time until the BTF is cleared by reading the I2C_DR register (see Figure 14-4).

Figure 14-4 Slave receiver transfer sequence

7-bit slave receiver



10-bit slave receiver



Note: S = Start bit, Sr = Repeated start bit, P = Stop bit, A = Acknowledge, NA = No Acknowledge

Evx = Event (interrupt occurs if ITEVFEN=1)

EV1: ADDR=1, cleared by reading SR1 then SR2.

EV2: RxNE =1, cleared by reading DR register.

EV4: STOPF=1, cleared by reading SR1 then writing to CR1.

1. EV1 event may extend the SCL low time until the corresponding software sequence finishes.
2. If the software sequence is not completed before the next byte reception ends, the EV2 event will extend the SCL low time.

3. After checking SR1, users should perform a full clearing sequence for each set flag.

For the ADDR and STOPF flags, the following sequence should be executed in the I²C interrupt service routine:

Read SR1

```
if (ADDR == 1) {read SR1; read SR2}  
if (STOPF == 1) {read SR1; write CR1}
```

This ensures that both the ADDR and STOPF flags are cleared if both are set.

Disabling Slave Communication

After the final data byte is transmitted, the master device will generate a stop bit. The interface will detect this condition and:

Set the STOPF bit to 1 and generate an interrupt when ITEVFEN is 1.

- Clear the STOPF bit by reading the SR1 register and writing to the CR1 register (see Figure 14-4: Slave Receiver Transfer Sequence EV4).

14.3.3 I²C Master Mode

In master mode, the I²C interface starts data transfer and generates clock signals. Serial data transfer always begins with a start condition and ends with a stop condition. Once a start bit is generated on the bus via the START bit, master mode is selected.

In master mode, the following sequence should be performed:

- Program the peripheral input clock in the I2C_CR2 register to generate the correct timing.
- Configure the clock control register.
- Configure the rise time register.
- Program the I2C_CR1 register to enable the peripheral.
- Set the START bit in the I2C_CR1 register to generate a start condition.

The minimum input clock frequency for peripherals is:

- 2 MHz in standard mode
- 4 MHz in fast mode

Start Condition

After setting the START bit to 1, the interface generates a start condition and switches to master mode (M/SL bit set to 1) once the BUSY bit is cleared.

Caution: After the START bit is set to 1 in master mode, the interface will generate a repeated start condition after the current byte transfer ends.

After the start condition is issued:

- The SB bit will be set to 1 by hardware, and an interrupt will be generated when the ITEVFEN bit is set to 1.

Next, the master will wait for software to read SR1 and then write the slave address to the DR register (see transfer sequences in Figures 14-5 and 14-6, EV5).

Slave Address Transfer

Next, the slave address will be sent via the internal shift register to the SDA line.

- In 10-bit addressing mode, sending the header sequence generates the following events:
 - The ADD10 bit will be set to 1 by hardware, and an interrupt will be generated when ITEVFEN is set to 1.
The master will then wait for software to read SR1 and write the second address byte to the DR register (see transfer sequences in Figures 14-5 and 14-6).
 - The ADDR bit will be set to 1 by hardware and an interrupt will be generated when ITEVFEN is set to 1.

The master will then wait to read the SR1 register, followed by reading the SR2 register (see transfer sequences in Figures 14-5 and 14-6).

- In 7-bit addressing mode, an address byte is sent.

After the address byte is transmitted:

- The ADDR bit will be set to 1 by hardware, and an interrupt will be generated when ITEVFEN is set to 1.
- The master will then wait to read SR1, followed by reading SR2 (see transfer sequences in Figures 14-5 and 14-6).

The master will decide whether to enter the transmit mode or receive mode based on the LSB of the slave address byte.

- In 7-bit addressing mode:
 - To enter transmit mode, the master will send the slave address and reset the LSB.

- To enter receive mode, the master will send the slave address and set the LSB to 1.
- In 10-bit addressing mode:
 - To enter transmit mode, the master will first send the header sequence (11110xx0), then send the slave address (where xx represents the two highest bits of the address).
 - To enter receive mode, the master will first send the header sequence (11110xx0), then send the slave address. Afterward, a repeated start condition will be generated, and the header sequence (11110xx1) will be sent again (where xx represents the two highest bits of the address).

The TRA bit indicates whether the master is in receive or transmit mode.

Master Transmitter

After sending the address and clearing the ADDR bit, the master will send the byte from the DR register to the SDA line via the internal shift register.

The master will continue to wait until the first data byte is written to the I2C_DR (see transfer sequence EV8_1 in Figure 14-5).

Upon receiving the acknowledge pulse, the TxE bit will be set to 1 by hardware, and an interrupt will be generated when both ITEVFEN and ITBUFEN bits are set to 1.

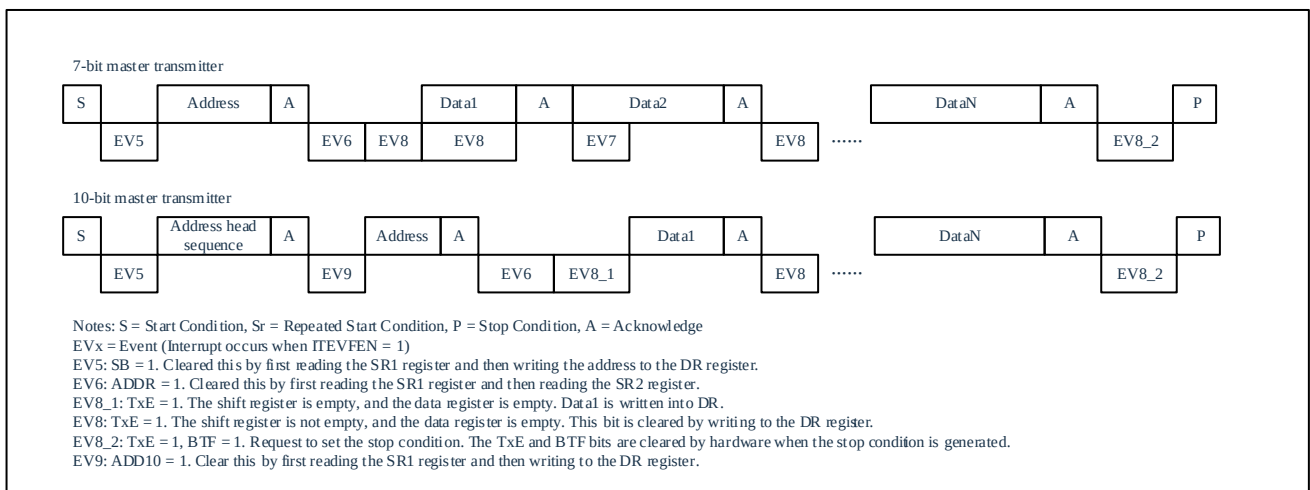
If the TxE bit is set to 1 before the previous data transfer finishes but the data byte has not been written to the DR register, the BTF bit will be set to 1, and the interface will extend the SCL low period, waiting for the I2C_DR register to be written to clear the BTF bit.

End Communication

When the last byte is written to the DR register, software will set the STOP bit to 1 to generate a stop condition (see transfer sequence EV8_2 in Figure 14-5). The interface will automatically return to slave mode (M/SL bit cleared).

Caution: When either the TxE or BTF bit is set to 1, the stop condition should be programmed during the EV8_2 event.

Figure 14-5 Master transmitter transfer sequence diagram



1. Events EV5, EV6, EV9, EV8_1, and EV8_2 can extend the SCL low level time until the corresponding software sequence is completed.

2. If the software sequence does not complete before the next byte transfer finishes, the EV8 event will extend the SCL low level time.

Master Receiver

After completing the address transfer and clearing the ADDR bit, the I²C interface enters master receiver mode. In this mode, the interface receives bytes from the SDA line through the internal shift register and saves them in the DR register. After each byte transfer, the interface will sequentially:

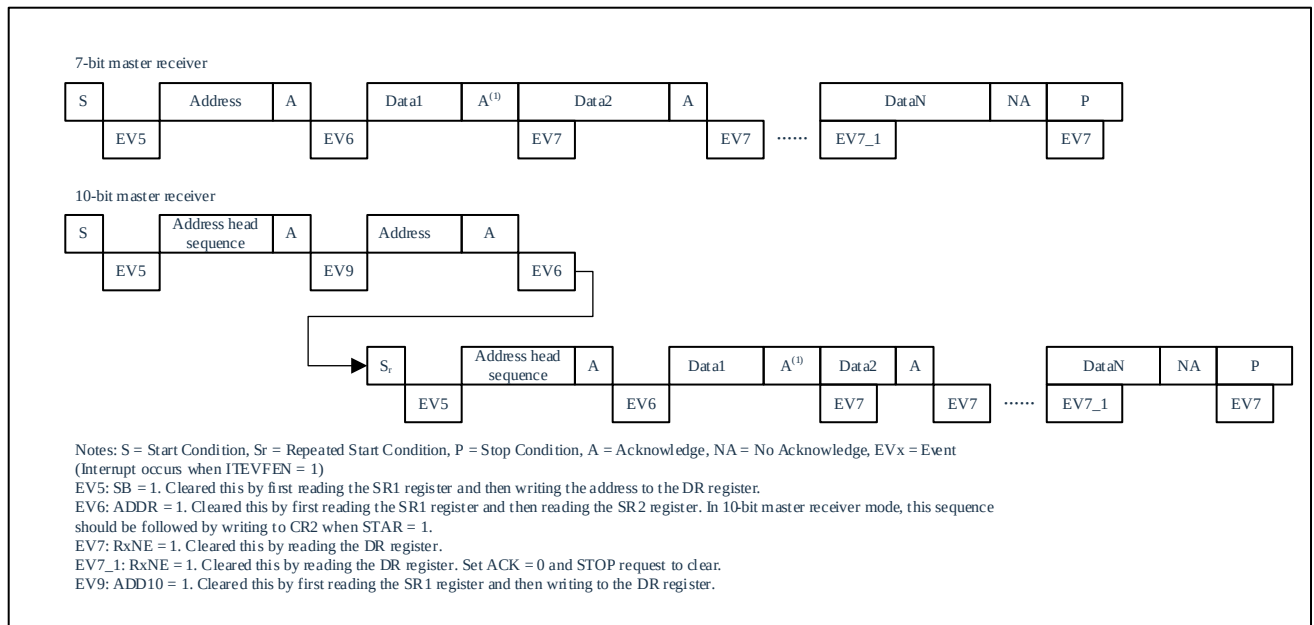
1. Generate an acknowledgment pulse (if the ACK bit is 1).
2. Set the RxNE bit to 1, and generate an interrupt when both the ITEVFEN and ITBUFEN bits are set to 1 (refer to Figure 14-6, transfer sequence EV7). If the RxNE bit is set to 1 before the previous data reception is completed but the data in the DR register has not been read, the BTF bit will be set to 1 by hardware. The interface will continue to extend the SCL low level time, waiting for the I2C_DR register to be written to clear the BTF bit.

Ending Communication

The master device sends a NACK after receiving the last byte from the slave device. After receiving this NACK, the slave device releases control of the SCL and SDA lines. The master device can then send a stop condition or a repeated start condition.

1. To generate a non-acknowledgment pulse after receiving the last data byte, the ACK bit must be cleared immediately after reading the second-to-last data byte (after the second-to-last RxNE event).
2. To generate a stop or repeated start condition, the software must set the STOP/START bit to 1 after reading the second-to-last data byte (after the second-to-last RxNE event).
3. In the case of receiving only a single byte, acknowledgment will be disabled during EV6 (before the ADDR flag is cleared), and a stop condition will be generated after EV6. After generating the stop condition, the interface will automatically return to slave mode (M/SL bit cleared).

Figure 14-6 Master receiver transfer sequence diagram



1. If receiving a single byte, it is NA (No Acknowledge).
2. Events EV5, EV6, and EV9 can extend the SCL low level time until the corresponding software sequence is completed.
3. If the software sequence does not complete before the next byte reception finishes, the EV7 event will extend the SCL low level time.
4. The EV7_1 software sequence must finish before the ACK pulse of the current byte transfer.

If the EV7_1 software sequence does not finish before the ACK pulse of the current byte transfer, it is recommended to follow these steps.

To ensure this, the following steps must be followed:

- Reset the ACK bit promptly before finishing the previous data reception.
- Set the STOP bit after the last data reception and do not receive any supplementary data.

For 2-byte reception:

- Wait for ADDR = 1 (SCL remains low until the ADDR flag is cleared).
- Reset ACK, set POS.
- Clear the ADDR flag.
- Wait for BTF = 1 (data 1 is in DR, data 2 is in the shift register, SCL remains low until data 1 is read).
- Set STOP.
- Read data 1 and data 2.

For N > 2 bytes reception, starting from the N-2 data reception:

- Wait for BTF = 1 (data N-2 is in DR, data N-1 is in the shift register, SCL remains low until data N-2 is read).
- Reset ACK.
- Read data N-2.
- Wait for BTF = 1 (data N-1 is in DR, data N is in the shift register, SCL remains low until data N-1 is read).
- Set STOP.
- Read data N-1 and data N.

14.3.4 Error Conditions

The following error conditions may lead to communication failure:

Bus Error (BERR)

This error occurs when the I²C interface detects an external stop condition or start condition during the transfer of an address or data. In this case:

- The BERR bit is set to 1 and an interrupt is generated if the ITERREN bit is set to 1.
- In slave mode: The hardware discards the data and releases the data line:
 - In the case of an erroneous start condition, the slave mode treats it as a repeated start condition and waits for an address or stop condition.
 - In the case of an erroneous stop condition, the slave mode operates according to the stop condition, and the hardware releases the data line.
- In master mode: The data line is not released, and it does not affect the current transfer state. Software decides whether to abort the current transfer.

Acknowledge Failure (AF)

This error occurs when the interface detects a no-acknowledge pulse. In this case:

- The AF bit is set to 1 and an interrupt is generated if the ITERREN bit is set to 1.
- The transmitter receiving the NACK must reset the communication:
 - In slave mode: The hardware releases the data line.
 - In master mode: Software must generate a stop condition or repeated start condition.

Arbitration Lost (ARLO)

This error occurs when the I²C interface detects a lost arbitration. In this case:

- The ARLO bit is set to 1 by hardware (and an interrupt is generated if the ITERREN bit is set to 1).
- The I²C interface automatically returns to slave mode (M/SL bit cleared). When arbitration is lost, the interface cannot acknowledge to its own slave address in the current transfer. However, it can acknowledge after the master device that wins the arbitration issues a repeated start condition.
- The hardware releases the data line.

Overflow/Underflow Error (OVR)

Overflow Error: Occurs in slave mode when clock stretching is disabled and the I²C interface is receiving data. The interface has received a byte (RxNE = 1), but the data in the DR register has not been read before the next byte is received. In this case:

- The last received byte is lost.
- When an overrun error occurs, software should clear the RxNE bit, and the transmitter should resend the last byte.

Underflow Error: Occurs in slave mode when clock stretching is disabled, and the I²C interface is sending data. The next clock signal arrives before the slave device has written the next byte of data into the DR register (TxE = 1). In this case:

- The byte in the DR register will be retransmitted.
- Users should ensure that the data received from the receiver during an underrun error is discarded and ensure the next byte is written to the DR register within the clock low time specified by the I²C bus standard.

For the first byte to be transmitted, it must be written to the DR register after clearing the ADDR flag and before the first rising edge of SCL. Otherwise, the receiver must discard the first data byte.

14.3.5 Programmable Noise Filter

In fast mode, the I²C standard requires that noise with spike widths on the SDA and SCL lines below 50 ns be suppressed. An analog noise filter is implemented in the SDA and SCL I/Os. This filter is enabled by default, and it can be disabled by setting the ANOFF bit in the I²C_FLTR register to 1.

The DNF[3:0] bits can be configured with a non-zero value to enable the digital noise filter. This can suppress noise on the SDA and SCL inputs with pulse widths less than $DNF[3:0] * TPCLK1$.

When the digital noise filter is enabled, the SDA hold time can increase by $(DNF[3:0] + 1) * TPCLK$.

To meet the requirements for the maximum hold time in the I²C bus specification version 2.1 (Thd:dat), the DNF bits must be programmed according to the restrictions shown in Table 14-1, assuming the analog filter is disabled.

Caution: The DNF[3:0] bits must be configured when I²C is disabled (PE = 0). If the analog filter is also enabled, the digital filter must be added to the analog filter.

Table 14-1 Maximum values of DNF[3:0] to meet Thd:dat(max)

PCLK frequency	DNF Max. value	
	Standard mode	Fast mode
$2 \leq F_{PCLK} \leq 5$	2	0
$5 < F_{PCLK} \leq 10$	12	0
$10 < F_{PCLK} \leq 20$	15	1
$20 < F_{PCLK} \leq 30$	15	7
$30 < F_{PCLK} \leq 40$	15	13
$40 < F_{PCLK} \leq 48$	15	15

Caution: The above table provides corresponding constraints based on the lowest frequency in each frequency range. If the system can tolerate the maximum hold time violation, a larger DNF value can be used.

14.3.6 SDA/SCL Line Control

- If clock stretching is enabled:
 - Transmit Mode: If TxE = 1 and BTF = 1: The interface will keep the clock line low before transmitting data to wait for the microcontroller to write the byte into the data register (both the buffer and shift registers are empty).
 - Receive Mode: If RxNE = 1 and BTF = 1: The interface will keep the clock line low after receiving data to wait for the microcontroller to read the byte from the data register (both the buffer and shift registers are full).
- If clock stretching is disabled in slave mode:
 - RxNE = 1, and if the data in the DR register has not been read before the next data is received, an overrun error will occur, causing the last received byte to be lost.
 - TxE = 1, and if the data has not been written into the DR register before the arrival of the next byte, an underrun error will occur, causing the same byte to be transmitted again.
 - Write conflicts are not managed.

14.3.7 SMBus

Preface

The System Management Bus (SMBus) is a two-wire interface that allows devices to communicate with each other or with other parts of the system. It is based on the operational principles of I²C. SMBus provides a control bus for system and power management-related tasks. The system can use SMBus for message passing with devices without switching individual control lines.

The SMBus specification involves three types of devices: Slave devices, which receive or respond to commands. Master devices, which issue commands, generate clocks, and terminate transfers. Host, a dedicated master device that provides the main interface to connect to the system CPU. The host must have both master and slave capabilities and must support the SMBus host notification protocol. There can only be one host in the system.

Similarities Between SMBus and I²C

- Two-wire bus protocol (1 clock line, 1 data line) + optional SMBus alarm line.
- Master-slave communication, with the master device providing the clock.
- Multi-master functionality.
- The SMBus data format is similar to the I²C 7-bit address format (see Figure 14-1).

Differences Between SMBus and I2C

The following table highlights the differences between SMBus and I2C:

Table 14-2 SMBus and I²C

SMBus	I ² C
Maximum speed 100KHz	Maximum speed 400KHz
Minimum clock speed 10KHz	No minimum clock speed
35m clock low timeout	No timeout
Fixed logic levels	Logic levels depend on V _{DD}
Different address types (Reserved, Dynamic, etc.)	7-bit, 10-bit, and broadcast slave mode address types
Different bus protocols (Fast Commands, Process Calls, etc.)	No bus protocol

Applications of SMBus

Through the System Management Bus, devices can provide manufacturer information, report their model/part number to the system, store the status of paused events, report different types of errors, accept control parameters, and return their status. SMBus can provide a control bus for system and power management-related tasks.

Device Identification

Any device acting as a slave on the SMBus has a unique address, referred to as the slave address. For a list of reserved slave addresses, refer to the SMBus specification version 2.0 (<http://smbus.org/specs/>).

Bus Protocol

The SMBus specification supports up to 9 types of bus protocols. For detailed information about these protocols and SMBus address types, refer to the SMBus specification version 2.0 (<http://smbus.org/specs/>). These protocols should be implemented through user software.

Address Resolution Protocol (ARP)

The SMBus address conflict issue among slave devices can be resolved by dynamically assigning a new unique address to each slave device. ARP has the following characteristics:

- Address assignment uses the standard SMBus physical layer arbitration mechanism.
- Once powered on, the assigned address remains unchanged; addresses may also be retained when the device powers off.
- After address assignment, no additional SMBus data packet overhead occurs (i.e., subsequent accesses to the assigned slave address incur the same overhead as accessing devices with fixed addresses).
- Any SMBus master device can scan the bus.

Unique Device Identifier (UDID)

To provide a mechanism for isolating devices during address assignment, each device must have a unique device identifier (UDID).

For detailed information about the 128-bit UDID and ARP, please refer to the SMBus Specification Version 2.0 (<http://smbus.org/specs/>).

SMBus Alert Mode

SMBus Alert is an optional interrupt line signal mainly used by devices that want to extend their control capabilities at the expense of an additional pin. The SMBA line and signal are similar to the SCL and SDA signals. SMBA is used together with the SMBus broadcast address. Messages invoked by SMBus in this mode are 2 bytes long.

When the ALERT bit in the I2C_CR1 register is set to 1, a slave-only device can signal the host via SMBA that it wants to communicate. The host processes this interrupt by sending the Alert Response Address (ARA, with a value of 0001100X) to synchronize access to all SMBA devices. Only devices pulling SMBA low will acknowledge the alert response address. This status can be detected by the SMBALERT status flag in the I2C_SR1 register. The host performs a modified receive byte operation. The 7-bit device address from the slave sender is placed in the 7 most significant bits of the byte; the 8th bit can be 0 or 1.

If more than one device pulls SMBA low, the device with the highest priority (lowest bit address) wins arbitration during the slave address transmission. After acknowledging the slave address, the device must release SMBA. If after message transfer the host still detects SMBA low, it will read the ARA again. Hosts that do not implement the SMBA signal periodically poll the ARA.

For more details on SMBus Alert Mode, please refer to the SMBus Specification Version 2.0 (<http://smbus.org/specs/>).

Timeout Errors

There are some timing specification differences between SMBus and I2C.

SMBus defines a clock low timeout, TIMEOUT, as 35 ms. In addition, SMBus specifies TLOW: SEXT as the cumulative clock low extension time for slave devices, and TLOW: MEXT for master devices. For detailed information on these timeouts, refer to SMBus Specification Version 2.0 (<http://smbus.org/specs/>).

Timeout or Tlow error status flags in the I2C_SR1 register indicate these conditions.

How to Use the Interface in SMBus Mode

To switch from I2C mode to SMBus mode, perform the following steps:

- Set the SMBus bit in the I2C_CR1 register to 1.
- Configure the SMBTYPE and ENARP bits in the I2C_CR1 register as per application requirements.

If configuring a device as a master, follow the start bit generation procedure described in the I²C Master mode section; otherwise, follow the procedure in the I²C Slave mode section.

This application requires software control of various SMBus protocols:

- ENARP=1 and SMBTYPE=0: Use SMB device default address.
- ENARP=1 and SMBTYPE=1: Use SMB host header field.
- SMBALERT=1: Use SMB Alert Response Address.

14.3.8 DMA Requests

DMA requests (after being enabled) are used solely for data transfer. A DMA request is generated when the transmit data register is empty and the receive data register is full. Before performing I²C data transfer, DMA must be initialized and enabled. The DMAEN bit in the I2C_CR2 register must be set to 1 before the ADDR event. In both master and slave modes, if clock stretching is enabled, the DMAEN bit can also be set to 1 during the ADDR event before clearing the ADDR flag. A DMA request must be issued before the current byte transfer ends. When the data amount reaches the value programmed in the corresponding DMA channel, the DMA controller will send an End Of Transmission (EOT) signal to the I²C interface and generate a transfer complete interrupt (if enabled).

- **Master Transmitter:** In the EOT interrupt routine, disable the DMA request, then set the stop bit after waiting for the BTF event.
- **Master Receiver:**
 - When the number of bytes to be received is equal to or greater than 2, the DMA controller will send a hardware EOT_1 signal after receiving the penultimate byte (the N-1-th byte). If the LAST bit in the I2C_CR2 register is set to 1, I²C will automatically send a NACK after the next byte following EOT_1. The user can generate the stop bit in the DMA transfer complete interrupt (if enabled).
 - When only a single byte is to be received: A NACK must be programmed during the EV6 event before clearing the ADDR flag, i.e., when ADDR = 1, program ACK = 0. Then, the user can program the stop bit either after clearing the ADDR flag or in the DMA transfer complete interrupt routine.

Using DMA for Transmission

Setting the DMAEN bit in the I2C_CR2 register to 1 enables the DMA mode for transmission. When TXE is 1, data will be loaded into the I2C_DR register by DMA from the pre-configured memory area. To map a DMA channel for I2C transmission, follow these steps (where x represents the channel number):

1. Set the I2C_DR register address in the DMDARj register. After each TXE event, data will be moved from memory to this address.
2. Set the memory address in the DMSARj register. After each TXE event, data will be loaded from this memory to the I2C_DR register.
3. Configure the total number of bytes to be transmitted in the DMACTj register. This value will decrement after each TXE event.
4. Use the DMA_CHj_SEL register to select the channel trigger source. The priority decreases from channel 0 to channel 3.
5. Set the EN bit in the DMAEN0 register to 1 to activate the channel. When the transmitted data reaches the value programmed in the DMA controller, DMA will generate an interrupt on the DMA channel interrupt vector (if enabled).

Caution: If using DMA for transfer, do not enable the ITBUFEN bit in the I2C_CR2 register.

Using DMA for Reception

Setting the DMAEN bit in the I2C_CR2 register to 1 enables the DMA mode for reception. When receiving data bytes, data will be loaded from the I2C_DR register to the memory area configured by the DMA peripheral (refer to DMA specifications). To map a DMA channel for I²C reception, follow these steps (where x represents the channel number):

1. Set the I2C_DR register address in the DMDARj register. After each RXNE event, data will be moved from this address to memory.
2. Set the memory address in the DMSARj register. After each RXNE event, data will be loaded from the I2C_DR register to this memory area.
3. Configure the total number of bytes to be received in the DMACTj register. This value will decrement after each RXNE event.
4. Use the DMA_CHj_SEL register to select the channel trigger source. The priority decreases from channel 0 to channel 3.
5. Set the EN bit in the DMAEN0 register to 1 to activate the channel. When the received data reaches the value programmed in the DMA controller, DMA will generate an interrupt on the DMA channel interrupt vector (if enabled).

Caution: If using DMA for reception, do not enable the ITBUFEN bit in the I2C_CR2 register.

14.3.9 Packet Error Checking (PEC)

The purpose of the PEC (Packet Error Checking) calculator is to improve the reliability of communication. PEC performs a serial CRC-8 calculation on each bit using the polynomial ($C(x) = x^8 + x^2 + x + 1$).

- To enable PEC, set the ENPEC bit in the I2C_CR1 register to 1. PEC calculates the CRC-8 for all message bytes, including the address and R/W bit.
 - During transmission: After the TxE event corresponding to the last byte, set the PEC transfer bit in the I2C_CR1 register to 1. PEC will be transferred after the last byte is transferred.
 - During reception: After the RxNE event corresponding to the last byte, set the PEC bit in the I2C_CR1 register to 1. If the next received byte is not equal to the internally calculated PEC, a NACK will be sent. In master receiver mode, a NACK will be sent after PEC, regardless of the check result. In slave mode, PEC must be set before CRC reception ACK. In master mode, the PEC bit must be set before resetting ACK.
- The I2C_SR1 register provides a PECERR error flag and interrupt.
- When both DMA and PEC are enabled:
 - During transmission: When the I²C interface receives the EOT signal from the DMA controller, PEC will automatically be sent after the last byte.
 - During reception: When the I²C interface receives the EOT_1 signal from the DMA controller, the next byte will be treated as PEC and checked. A DMA request will be generated after PEC reception.
- To allow intermediate PEC transfer, use the LAST control bit in the I2C_CR2 register to determine whether this is truly the last DMA transfer. If it is indeed the last DMA request for the master receiver, a NACK will be automatically sent after receiving the last byte.
- PEC calculation may fail due to arbitration loss.

14.4 I²C Interrupts

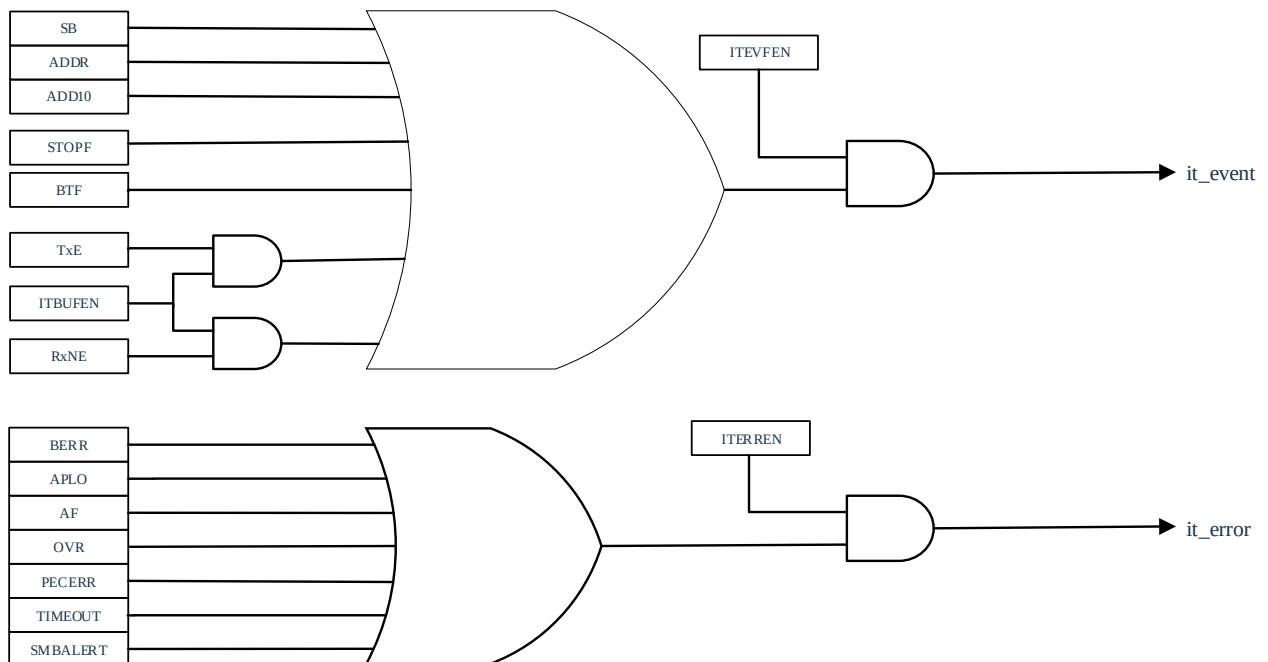
The following table lists the I²C interrupt requests:

Table 14-3 Maximum values of DNF[3:0] that meet Thd:dat(max)

Interrupt event	Event flag	Enable control bit
Transmit start bit (master mode)	SB	ITEVFEN
Transmitted address (master mode) or matched address (slave mode)	ADDR	
10-bit address header transmitted (master mode)	ADD10	
Stop bit received (slave mode)	STOPF	
Data byte transfer complete	BTF	
Receive buffer not empty	RxNE	ITEVFEN and ITBUFEN
Transmit buffer empty	TxE	
Bus error	BERR	ITERREN
Arbitration lost (master mode)	ARLO	
Acknowledge failure	AF	
Overrun/underrun	OVR	
PEC error	PECERR	
Timeout/Tlow error	TIMEOUT	
SMBus alert	SMBALERT	

Caution: SB, ADDR, ADD10, STOPF, BTF, RxNE, and TxE are mapped to the same interrupt channel via a logical OR. BERR, ARLO, AF, OVR, PECERR, TIMEOUT, and SMBALERT are also mapped to the same interrupt channel via a logical OR.

Figure 14-7 I²C interrupt mapping diagram



14.5 I²C Debug Mode

When the microcontroller enters debug mode, SMBus will either continue normal operation or stop working based on 1.4 selection.

14.6 Register Mapping

(I2C base address = 0x4006_5400) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Address	R/W	Description	Reset value
I2C_CR1	0x00	R/W	I ² C Control Register 1	0x00
I2C_CR2	0x04	R/W	I ² CControl Register 2	0x00
I2C_OAR1	0x08	R/W	I ² C Own Address Register 1	0x00
I2C_OAR2	0x0C	R/W	I ² C Own Address Register 2	0x00
I2C_DR	0x10	R/W	I ² C Data Register	0x00
I2C_SR1	0x14	R/W	I ² C Status Register 1	0x00
I2C_SR2	0x18	R/W	I ² C Status Register 2	0x00
I2C_CCR	0x1C	R/W	I ² C Clock Control Register	0x00
I2C_TRISE	0x20	R/W	I ² C TRISE Register	0x02
I2C_FLTR	0x24	R/W	I ² C FLTR Register	0x00

14.7 Register Description

14.7.1 I²C Control Register 1 (I2C_CR1)

Bit	Symbol	Description	Reset value
15	SWRST	Software reset When set to 1, I²C is in reset state. Before resetting this bit, ensure that the I2C lines are released and the bus is idle. 0: I²C peripheral is not in reset state. 1: I²C peripheral is in reset state. Caution: This bit can be used to reinitialize the device after an error or lock state. For example, if the BUSY bit is set to 1 but cannot be reset due to bus interference, the SWRST bit can be used to exit this state.	0
14	--	Reserved	--
13	ALERT	SMBus alert This bit is set to 1 and cleared by software, and can be cleared by hardware when PE=0. 0: Release the SMBA pin to a high level. The alert response address header is followed by NACK. 1: Drive the SMBA pin to a low level. The alert response address header is followed by ACK.	0
12	PEC	Packet error checking This bit is set to 1 and cleared by software, and can be cleared by hardware when PEC transfer is complete, or when PE=0 or when a start or stop bit is detected. 0: Do not transfer PEC. 1: PEC transfer (in Tx or Rx mode) Caution: PEC calculation may fail due to arbitration loss.	0
11	POS	Acknowledge/PEC Position (for received data) This bit is set to 1 and cleared by software, and can be cleared by hardware when PE=0. 0: The ACK bit controls the (N)ACK for the byte currently being received in the shift register. The PEC bit indicates that the current byte in the shift register is a PEC. 1: The ACK bit controls the (N)ACK for the next byte to be received in the shift register. The PEC bit indicates that the next byte in the shift register is a PEC. Caution: The POS bit can only be used when the master device receives 2 bytes. It must be configured before data reception begins.	0
10	ACK	Acknowledge enable This bit is set to 1 and cleared by software, and can be cleared by hardware when PE=0. 0: No acknowledgment is returned. 1: Return acknowledgment after receiving a byte (matching address or data).	0
9	STOP	Generate stop bit This bit is set to 1 and cleared by software, and can be cleared by hardware when a stop bit is detected, or set to 1 by hardware when a timeout error is detected. In master mode: 0: Do not generate a stop bit. 1: Generate a stop bit after transferring the current byte or sending the current start bit. In slave mode: 0: Do not generate a stop bit. 1: Release the SCL and SDA lines after completing the current byte transfer.	0
8	START	Generate start bit This bit is set to 1 and cleared by software, and can be cleared by hardware after the start bit is transmitted, or when PE=0 in slave mode.	0

		In master mode: 0: Do not generate a start bit. 1: Generate a repeated start bit. In slave mode: 0: Do not generate a start bit. 1: Generate a start bit when the bus is idle.	
7	NOSTRETCH	Clock stretching disable (slave mode) In Slave Mode, when the ADDR or BTF flag is set to 1, this bit is used to disable clock stretching until it is reset by software. 0: Enable clock stretching. 1: Disable clock stretching.	0
6	ENGCG	General call enable 0: Disable general call. Do not acknowledge address 00h. 1: Enable general call. Acknowledge address 00h.	0
5	ENPEC	PEC enable 0: Disable PEC calculation. 1: Enable PEC calculation.	0
4	ENARP	ARP enable 0: Disable ARP 1: Enable ARP SMBTYPE=0: SMBus device default address. SMBTYPE=1: SMBus master address.	0
3	SMBTYPE	SMBus type 0: SMBus device 1: SMBus host	0
2	--	Reserved	--
1	SMBUS	SMBus mode 0: I2C mode 1: SMBus mode	0
0	PE	Peripheral enable 0: Disable peripheral 1: Enable peripheral Caution: If this bit is reset during communication, the peripheral will be disabled after the current communication ends and will enter the IDLE state. Since PE=0 when communication ends, all bits will reset. In Master Mode, this bit cannot be reset before the communication ends.	0

Caution: After setting the STOP, START, or PEC to 1, no write operations should be performed on I2C_CR1 by software until the hardware clears the bit. Otherwise, another STOP, START, or PEC set request may be issued.

14.7.2 I²C Control Register 2 (I2C_CR2)

Bit	Symbol	Description	Reset value
15:13	--	Reserved	--
12	LAST	DMA last transfer 0: Next DMA EOT is not the last transfer 1: Next DMA EOT is the last transfer Caution: This bit is used in master receive mode and can generate a NACK for the last received data.	0
11	DMAEN	DMA requests enable 0: Disable DMA request 1: Enable DMA request when TxE = 1 or RxNE = 1	0
10	ITBUFEN	Buffer interrupt enable 0: No interrupt generated when TxE = 1 or RxNE = 1 1: Generate an event interrupt when TxE = 1 or RxNE = 1 (regardless of DMAEN state)	0
9	ITEVTEN	Event interrupt enable 0: Disable event interrupt 1: Enable event interrupt when the following conditions are met: —SB = 1(Master mode) —ADDR = 1(Master/Slave mode) —ADD10= 1(Master mode) —STOPF = 1(Slave mode) —BTF = 1, with no TxE or RxNE event —ITBUFEN = 1 and TxE event is set to 1 —ITBUFEN = 1 and RxNE event is set to 1	0
8	ITERREN	Error interrupt enable 0: Disable error interrupt 1: Enable error interrupt when the following conditions are met: —BERR = 1 —ARLO = 1 —AF = 1 —OVR = 1 —PECERR = 1 —TIMEOUT = 1 —SMBALERT = 1	0
7:6	--	Reserved	--
5:0	FREQ[5:0]	Peripheral clock frequency The peripheral clock frequency must be configured using the APB clock frequency (I2C peripheral connected to APB). The minimum allowed frequency is 3 MHz, and the maximum frequency is limited by the maximum APB frequency (48 MHz). 0b000011: 3 MHz 0b000110: 6 MHz 0b001100: 12 MHz 0b010111: 24 MHz 0b101111: 48 MHz Other: Disabled	0

14.7.3 I²C Own Address Register 1 (I2C_OAR1)

Bit	Symbol	Description	Reset value
15	ADDMODE	Addressing mode (Slave mode) 0: 7-bit slave address (cannot respond to 10-bit address) 1: 10-bit slave address (cannot respond to 7-bit address)	0
14:10	--	Reserved	--
9:8	ADD[9:8]	Interface address 7-bit addressing mode: N/A 10-bit addressing mode: Address bits 9:8	0
7:1	ADD[7:1]	Interface address Address bits 7:1	0
0	ADD0	Interface address 7-bit addressing mode: N/A 10-bit addressing mode: Address bit 0	0

14.7.4 I²C Own Address Register 2 (I2C_OAR2)

Bit	Symbol	Description	Reset value
15:8	--	Reserved	--
7:1	ADD2[7:1]	Interface address In dual addressing mode, address bits 7:1	0
0	ENDUAL	Dual addressing mode enable 0: In 7-bit addressing mode, responds only to OAR1 address 1: In 7-bit addressing mode, responds to both OAR1 and OAR2 addresses	0

14.7.5 I²C Data Register (I2C_DR)

Bit	Symbol	Description	Reset value
15:8	--	Reserved	--
7:0	DR[7:0]	8-bit data register The byte received or the byte to be sent to the bus. Transmit mode: The transmission of bytes automatically begins when the first byte is written into the DR register. If the next byte to be transmitted is placed in DR immediately after starting the transfer (TxE = 1), continuous transmission flow can be maintained. Receive mode: The received byte is copied to the DR register (RxNE = 1). Continuous transfer flow can be maintained if the DR is read before receiving the next data byte (RxNE = 1). Caution: In slave mode, the address is not copied to the DR register. Caution: The hardware does not manage write conflicts (writing to DR is possible even when TxE = 0). Caution: If an ARLO event occurs during the ACK pulse, the received byte will not be copied to the DR register, and thus, the byte cannot be read.	0

14.7.6 I²C Status Register 1 (I2C_SR1)

Bit	Symbol	Description	Reset value
15	SMBALERT	SMBus alert In SMBus host mode: 0: No SMBALERT 1: An SMBALERT event occurred on the pin In SMBus slave mode: 0: No SMBALERT response address header 1: Received SMBALERT response address header indicating a low SMBALERT level —Cleared by software writing 0, or cleared by hardware when PE=0 in slave mode.	0
14	TIMEOUT	Timeout or Tlow error 0: No timeout error 1: SCL low level duration exceeds 25 ms (timeout) or the master device accumulates clock low time exceeding 10 ms (Tlow:mext), or slave device accumulates clock low time exceeding 25 ms (Tlow:sext) — When set to 1 in slave mode: Slave device resets communication and hardware releases the data line. — When set to 1 in master mode: Hardware sends a stop bit. — Cleared by software writing 0, or cleared by hardware when PE=0 in slave mode. Caution: This function is available only in SMBus mode.	0
13	--	Reserved	--
12	PECERR	PEC Error in reception 0: No PEC error: Receiver returns ACK after receiving PEC (if ACK=1) 1: PEC error: Receiver returns NACK after receiving PEC (regardless of ACK value) — Cleared by software writing 0, or cleared by hardware when PE=0. — Caution: If an incorrect CRC is received and the PEC control bit is not set to 1 before receiving the end CRC, the PECERR bit will not be set in slave mode. However, the received CRC can be checked by reading the PEC value.	0
11	OVR	Overrun/Underrun 0: No overrun/underrun 1: Overrun or underrun — In slave mode, set to 1 by hardware when NOSTRETCH=1 and: — A new byte is received during reception (including ACK pulse) but not yet read from DR register. The newly received byte will be lost. — A new byte will be transmitted but data is not yet written to DR register. The same byte is transmitted twice. — Cleared by software writing 0, or cleared by hardware when PE=0. Caution: If the write operation to DR is close to the time of SCL rising edge, the transmitted data is uncertain, and data hold-time errors can occur.	0
10	AF	Acknowledge failure 0: No acknowledge failure 1: Acknowledge failure — Set to 1 by hardware when no ACK is returned. — Cleared by software writing 0, or cleared by hardware	0

		when PE=0 in slave mode.	
9	ARLO	Arbitration lost (Master Mode) 0: No arbitration lost detected 1: Arbitration lost detected — Set to 1 by hardware when the interface loses the bus arbitration to another master device during bus competition. — Cleared by software writing 0, or cleared by hardware when PE=0. — After the ARLO event, the interface automatically switches back to slave mode (M/SL=0). Caution: In SMBus, data arbitration in slave mode only occurs during the data phase or during the transmit confirmation period (not for address confirmation).	0
8	BERR	Bus error 0: No erroneous start or stop bit 1: Erroneous start or stop bit detected — When SCL is high, if the interface detects an invalid position during byte transmission where SDA rises or falls, set to 1 by hardware. — Cleared by software writing 0, or cleared by hardware when PE=0 in slave mode.	0
7	TxE	Data register empty (Transmitter) 0: Data register not empty 1: Data register empty — Set to 1 when DR is empty during transmission. TxE will not be set to 1 during the address phase. — Cleared by writing data to DR register, or cleared by hardware during start, stop bit, or when PE=0 in slave mode. If a NACK is received or the next byte to be transmitted is PEC (PEC=1), TxE will not be set to 1. Caution: Writing the first byte to transmit or writing data when BTF is set to 1 will not clear TxE, as the data register is still empty in both cases.	0
6	RxNE	Data register not empty (Receiver) 0: Data register empty 1: Data register not empty — Set to 1 when the data register is not empty in receive mode. RxNE will not be set to 1 during the address phase. — Cleared by reading or writing to DR register, or cleared by hardware when PE=0. RxNE will not be set to 1 after an ARLO event. Caution: Reading data when BTF is set to 1 will not clear RxNE, as the data register is still full at that time.	0
5	--	Reserved	--
4	STOPF	Stop detection (Slave mode) 0: No stop bit detected 1: Stop bit detected — Set to 1 by hardware when the slave device detects a stop bit after the ACK pulse (if ACK=1). — Cleared by reading SR1 and writing to CR1 register, or cleared by hardware when PE=0. Caution: The STOPF bit will not be set after receiving NACK. It is recommended to perform a full clearing sequence after STOPF is set (first read SR1, then write to CR1).	0
3	ADD10	Transmit 10-bit header (master mode) 0: No ADD10 event 1: Master device has sent the first address byte (header) — Set to 1 by hardware when the master device sends the first byte	0

		in 10-bit address mode. — Cleared by software after reading SR1 register and writing the second address byte to DR register, or cleared by hardware when PE=0. Caution: The ADD10 bit will not be set after receiving NACK.	
2	BTF	Byte transfer finished 0: Data byte transfer not completed 1: Data byte transfer completed — Set to 1 by hardware when NOSTRETCH=0 and: — A new byte is received during reception (including ACK pulse) but not yet read from DR register (RxNE=1). — A new byte is about to be transmitted but not yet written to DR register (TxNE=1). — Cleared by software reading or writing DR register, or cleared by hardware after start or stop bit, or cleared by hardware when PE=0 in slave mode. Caution: The BTF bit will not be set after receiving NACK. If the next byte to be transmitted is PEC (when TRA=1 in the I2C_SR2 register and PEC=1 in the I2C_CR1 register), the BTF bit will not be set to 1.	0
1	ADDR	Address transmitted (Master Mode)/Address match (Slave Mode) Cleared by software after reading SR1 register and then reading SR2 register, or cleared by hardware when PE=0 in slave mode. Address match (Slave Mode): 0: Address does not match or no address received 1: Received address matches — Set to 1 by hardware when the received slave address matches the OAR register content, broadcast address, or default SMBus device address, or when SMBus host or SMBus alert is recognized. (Enabling depends on configuration). Caution: In slave mode, it is recommended to perform a full clearing sequence after ADDR is set to 1 (first read SR1, then write SR2). Address transmitted (Master Mode) 0: Address transmission not finished 1: Address transmission finished — Set to 1 when ACK is received after the second address byte in 10-bit addressing mode. — Set to 1 when ACK is received after the address byte in 7-bit addressing mode. Caution: The ADDR bit will not be set after receiving NACK.	0
		0: Address transmission not finished 1: Address transmission finished — Set to 1 when ACK is received after the second address byte in 10-bit addressing mode. — Set to 1 when ACK is received after the address byte in 7-bit addressing mode. Caution: The ADDR bit will not be set after receiving NACK.	0
0	SB	Start bit (Master Mode) 0: No start bit 1: Start bit has been sent — Set to 1 when the start condition is generated. — Cleared by writing to DR register after reading SR1 register, or cleared by hardware when PE=0.	0

14.7.7 I²C Status Register 2 (I2C_SR2)

Bit	Symbol	Description	Reset value
15:8	PEC[7:0]	Packet error checking register When ENPEC=1, this register contains the internal PEC.	0
7	DUALF	Dual flag (Slave Mode) 0: Received address matches OAR1 1: Received address matches OAR2 — Cleared by hardware upon stop bit, repeated start, or when PE=0.	0
6	SMBHOST	SMBus host header (Slave Mode) 0: No SMBus host address 1: SMBTYPE=1 and ENARP=1 when SMBus host address is received —Cleared by hardware upon stop bit, repeated start, or when PE=0.	0
5	SMBDEFAULT	SMBus device default address (Slave Mode) 0: No SMBus device default address 1: ENARP=1 when SMBus device default address is received —Cleared by hardware upon stop bit, repeated start, or when PE=0.	0
4	GENCALL	General call address (Slave Mode) 0: No general call 1: ENGC=1 when general call address is received —Cleared by hardware upon stop bit, repeated start, or when PE=0.	0
3	--	Reserved	--
2	TRA	Transmitter/receiver 0: Receiver 1: Transmitter This bit is set at the end of the entire address phase according to the R/W bit status of the address byte. Also cleared by hardware when STOP bit is detected (STOPF=1), repeated start, bus arbitration lost (ARLO=1), or when PE=0 in slave mode.	0
1	BUSY	Bus busy 0: No communication on bus 1: Communication ongoing on bus — Set by hardware when SDA or SCL is detected low — Cleared by hardware when stop bit is detected This bit indicates whether communication is ongoing on the bus. This information updates even if the interface is disabled (PE=0).	0
0	MSL	Master/slave 0: Slave mode 1: Master mode — Set by hardware when the interface enters master mode (SB=1) — Cleared by hardware upon stop bit detection, arbitration lost (ARLO=1), or when PE=0 in slave mode.	0

Caution: Reading I2C_SR1 followed by reading I2C_SR2 will clear the ADDR flag, even if the ADDR flag is set again after reading I2C_SR1. Therefore, I2C_SR2 must only be read after the ADDR bit in I2C_SR1 is set or the STOPF bit is cleared.

14.7.8 I²C Clock Control Register (I2C_CCR)

Caution: F_{PCLK} must be at least 2 MHz to achieve the standard mode I²C frequency. It must be at least 4 MHz to achieve the fast mode I²C frequency. It must be a multiple of 10 MHz to achieve the maximum 400 KHz I²C fast mode clock. The CCR register must only be configured when I²C is disabled (PE = 0).

Bit	Symbol	Description	Reset value
15	F/S	I ² C master mode selection 0: Standard mode I ² C 1: Fast mode I ² C	0
14	DUTY	Fast mode duty cycle 0: Fast mode $T_{low}/T_{high} = 2$ 1: Fast mode $T_{low}/T_{high} = 16/9$ (refer to CCR)	0
13:12	--	Reserved	--
11:0	CCR[11:0]	Clock control register in Fast/Standard mode (Master Mode) Controls the SCL clock in master mode. Standard mode or SMBus mode: $T_{high} = CCR * T_{PCLK}$ $T_{low} = CCR * T_{PCLK}$ Fast mode: If DUTY = 0: $T_{high} = CCR * T_{PCLK}$ $T_{low} = 2 * CCR * T_{PCLK}$ If DUTY = 1: (to reach 400 KHz) $T_{high} = 9 * CCR * T_{PCLK}$ $T_{low} = 16 * CCR * T_{PCLK}$ Example: To generate a 100 KHz SCL frequency in standard mode: If the APB maximum frequency is 8 MHz, $T_{PCLK} = 125$ ns, then CCR must be programmed to 0x28 ($0x28 \Leftrightarrow 40d \times 125$ ns = 5000 ns). Caution: · The minimum allowed value is 0x04, except for fast mode (duty cycle 9/16), where the minimum value is 0x01. · These times are not filtered. · The CCR register must only be configured when I²C is disabled (PE = 0).	0

14.7.9 I²C TRISE Register (I2C_TRISE)

Bit	Symbol	Description	Reset value
15:6	--	Reserved	--
5:0	TRISE[5:0]	Maximum rise time in Fast/Standard Mode (Master Mode) These bits must be programmed to the maximum SCL rise time specified in the I2C bus specification plus 1. For example: The maximum allowed SCL rise time in standard mode is 1000 ns. If the APB maximum frequency is 8 MHz and TPCLK = 125 ns, then the TRISE[5:0] bits must be programmed to 0x09. (1000 ns / 125 ns = 8 + 1) The filter value can also be added to the TRISE[5:0]. If the result is not an integer, then the TRISE[5:0] must be programmed to the integer part to meet the t_{HIGH} parameter requirement. Caution: TRISE[5:0] must only be configured when I2C is disabled (PE = 0).	2

14.7.10 I²C FLTR Register (I2C_FLTR)

Bit	Symbol	Description	Reset value
15:5	--	Reserved	--
4	ANOFF	Analog noise filter OFF 0: Enable analog noise filter 1: Disable analog noise filter Caution: ANOFF must only be configured when I2C is disabled (PE = 0).	0
3:0	DNF[3:0]	Digital noise filter These bits are used to configure the digital noise filter on the SDA and SCL input lines. The digital filter can suppress spikes with a pulse width less than DNF[3:0] * T_{PCLK}. 0000: Disable digital noise filter 0001: Enable digital noise filter, filtering capability up to 1* T_{PCLK} ... 1111: Enable digital noise filter, filtering capability up to 15* T_{PCLK} Caution: DNF[3:0] must only be configured when I2C is disabled (PE = 0). If the analog filter is also enabled, the digital filter must be added to the analog filter.	0

Chapter 15 Universal Synchronous/Asynchronous Receiver Transmitter (USART)

15.1 USART Introduction

The universal synchronous asynchronous receiver transmitter (USART) offers a flexible means of Full-duplex data exchange with external equipment requiring an industry standard NRZ asynchronous serial data format. The USART offers a very wide range of baud rates using a programmable baud rate generator.

It supports synchronous unidirectional communication and half-duplex single-wire communication. It also supports LIN (Local Interconnect Network), IrDA (Infrared Data Association) SIR ENDEC specification, and modem operations (CTS/RTS). Moreover, it also supports multiprocessor communication.

High speed data communication is possible by using the DMA (direct memory access) for multibuffer configuration.

15.2 USART Main Features

- Full-duplex asynchronous communications
- NRZ standard format (mark/space)
- Fractional baud rate generator system
 - General-purpose programmable baud rate transmitter/receiver.
- Programmable data word length (8 or 9 bits)
- Configurable stop bits (1 or 2 stop bits)
- LIN master mode synchronous stop symbol transmission and LIN slave mode stop symbol detection
 - In LIN hardware configuration, it can generate a 13-bit stop symbol and detect 10/11-bit stop symbols.
- Transmitter clock output for synchronous transmission
- IrDA SIR encoder/decoder
 - In normal mode, supports 3/16-bit duration.
- Single-wire half-duplex communication
- DMA (Direct Memory Access) for configurable multi-buffer communication
 - Using DMA to receive/send bytes in reserved SRAM buffers.
- Separate enable bits for transmitter and receiver
- Transfer detection flags:
 - Receiver buffer full
 - Transmitter buffer empty
 - Transfer complete
- Parity control:
 - Transmit parity bit
 - Check parity of received data bytes
- Four error detection flags:
 - Overrun error
 - Noise detection
 - Framing error

- Parity error
- Ten interrupt sources with flag bits:
 - CTS change
 - LIN stop symbol detection
 - Transmit data register empty
 - Transmission complete
 - Receive data register full
 - Line idle detected
 - Overrun error
 - Framing error
 - Noise error
 - Parity error
- Multiprocessor communication - Enters mute mode if the address doesn't match.
- Wake-up from Mute Mode (via line idle detection or address mark detection).
- Two Receiver Wake-up Modes: Address bit (MSB, 9th bit), Line idle.

15.3 USART Functional Description

The interface connects to other devices via three pins externally (see Figure 15-1). Any USART bidirectional communication requires at least two pins: the receive data input pin (RX) and the transmit data output pin (TX):

RX: The receive data input pin is the serial data input pin. Oversampling techniques can distinguish between valid input data and noise, which is used to recover the data.

TX: The transmit data output pin. If the transmitter is disabled, the output pin mode is determined by its I/O port configuration. If the transmitter is enabled but there is no data to transmit, the TX pin is at a high level. In single-wire mode, this I/O is used for both transmitting and receiving data (at USART level, followed by receiving data on SW_RX).

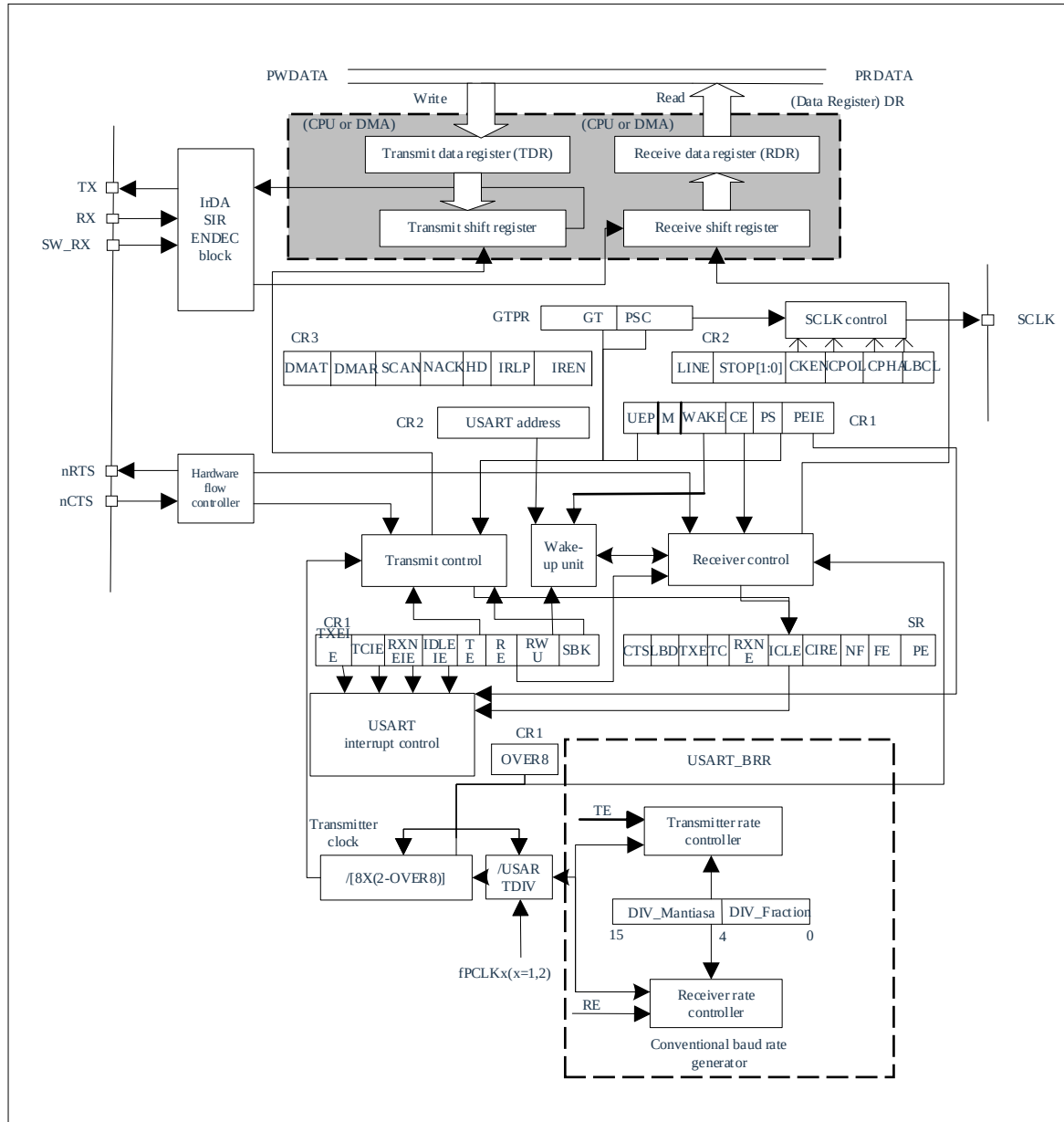
In normal USART mode, serial data is transmitted and received in frames via these pins:

- Keep the line idle before transmitting or receiving.
- Start bit.
- Data (word length of 8 or 9 bits), with the least significant bit first.
- One or two stop bits to indicate the end of frame transfer.
- The interface uses a fractional baud rate generator - with a 12-bit fraction and 4-bit decimal.
- Status Register (USART_SR).
- Data Register (USART_DR).
- Baud Rate Register (USART_BRR) - with a 12-bit fraction and 4-bit decimal.

In synchronous mode connection, the following pins are required:

- **SCLK:** Transmitter clock output. This pin is used to output the transmitter data clock to synchronize data transmission in SPI master mode (no clock pulses on the start and stop bits, and a clock pulse can be sent to the last data bit via software). Parallel data can be received synchronously on RX. This can be used to control peripherals with shift registers (such as LCD drivers). Clock phase and polarity can be programmed via software.
- In hardware flow control mode, the following pins are required:
- **nCTS:** “Clear to transmit” is used to block data transfer when the current transfer ends (high level).
- **nRTS:** “Request to transmit” is used to indicate that USART is ready to receive data (low level).

Figure 15-1 USART block diagram



15.3.1 USART Character Description

The word length of 8 or 9 bits can be selected by programming the M bit in the USART_CR1 register (see Figure 15-2). The TX pin is in a low level during the start bit period and in a high level during the stop bit period.

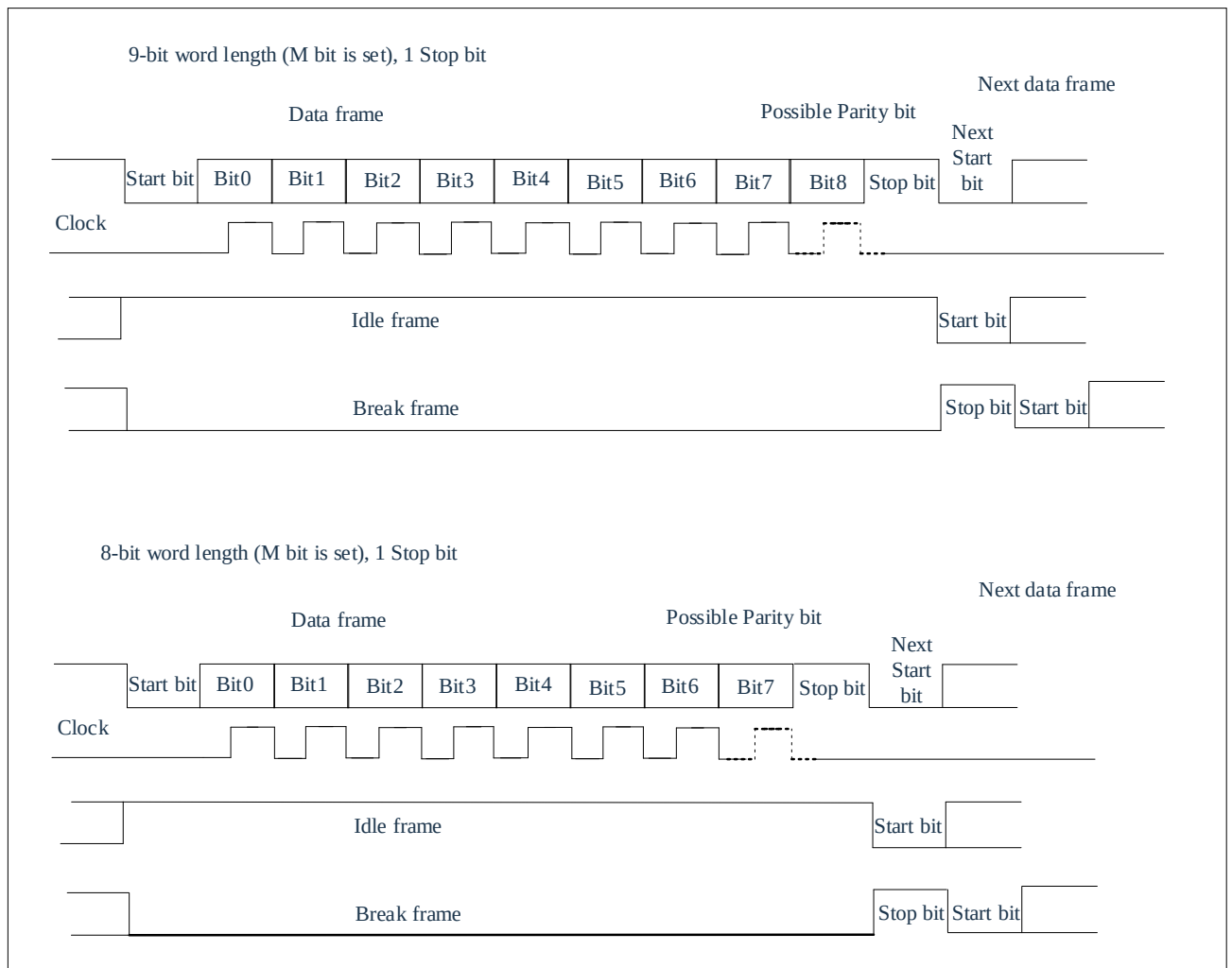
An idle character is understood as an average voltage level of “1” throughout the entire frame cycle (including the “1” level during the stop bit). The next data frame’s start bit follows this character.

The stop character is understood as receiving an average voltage level of “0” over one frame cycle. The transmitter inserts one or two stop bits (logic “1”) at the end of a frame to confirm the start bit.

Both sending and receiving are driven by a general-purpose baud rate generator. When the enable bits for the transmitter and receiver are set to 1, the corresponding transmit and receive clocks are generated.

Detailed information about each block is given below.

Figure 15-2 Word length programming



15.3.2 Transmitter

The transmitter can send data words of either 8 or 9 bits depending on the M bit status. The Transmit Enable bit (TE) must be set in order to activate the transmitter function. The data in the transmit shift register is output on the TX pin and the corresponding clock pulses are output on the SLCK pin.

15.3.2.1 Character Transmission

During an USART transmission, data shifts out least significant bit first (default configuration) on the TX pin. In this mode, the USART_TDR register consists of a buffer (TDR) between the internal bus and the transmit shift register (see Figure 15-1).

Every character is preceded by a start bit which is a logic level low for one bit period. The character is terminated by a configurable number of stop bits.

The following stop bits are supported by USART: 1 and 2 stop bits.

Note 1: The TE bit should not be reset during transmission of data. Resetting the TE bit during the transmission will corrupt the data on the TX pin as the baud rate counters will get frozen. The current data being transmitted is lost.

Note 2: An idle frame is sent after the TE bit is enabled.

15.3.2.2 Configurable Stop Bits

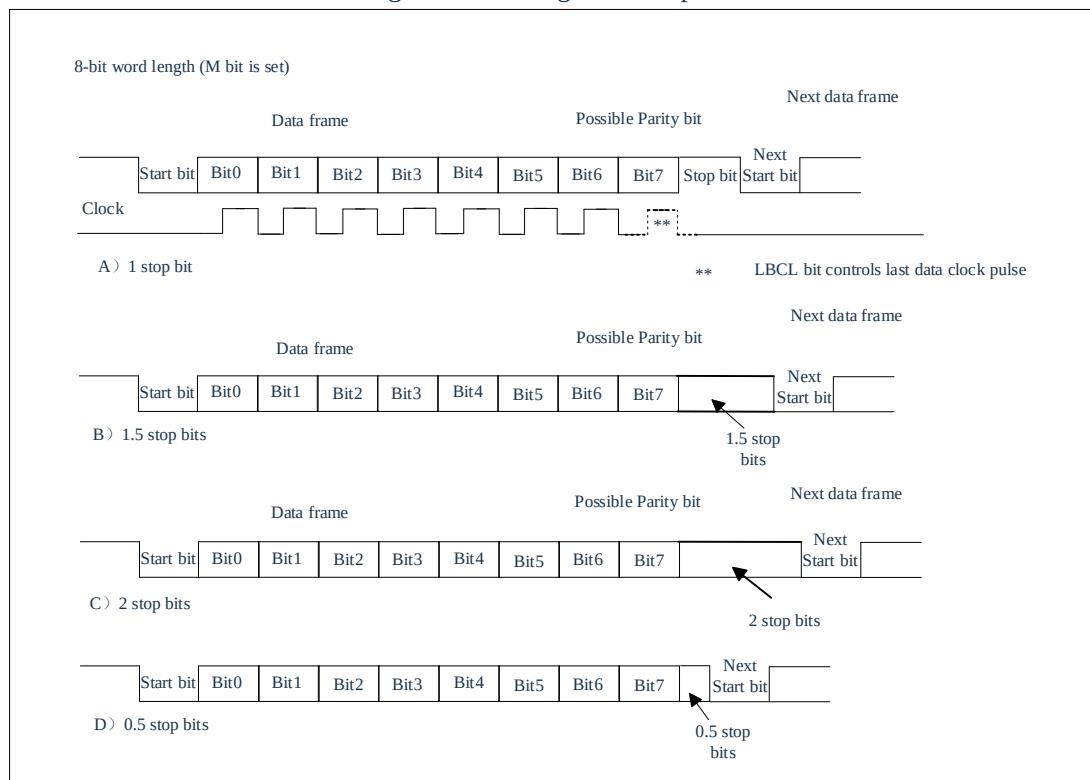
The number of stop bits to be transmitted with every character can be programmed in Control register 2, bit 13.

- 1 stop bit: This is the default value of number of stop bits.
- 2 stop bits: This is supported by normal USART, Single-wire and Modem modes.

An idle frame transmission will include the stop bits.

A break transmission will be 10 low bits (when M= 0) or 11 low bits (when M= 1) followed by configured stop bits. It is not possible to transmit long breaks (break of length greater than 10/11 low bits).

Figure 15-3 Configurable stop bits



Procedure:

1. Enable USART by writing 1 to the UE bit in the USART_CR1 register.
2. Program the M bit in USART_CR1 to define the word length.
3. Program the number of stop bits in the USART_CR2 register.
4. If using multi-buffer communication, enable DMA in USART_CR3 by setting the DMAT bit. Configure the DMA registers as explained in the multi-buffer communication section.
5. Use the USART_BRR register to select the desired baud rate.
6. Set the TE bit in USART_CR1 to 1 to transmit an idle frame on the first transmission.
7. Write the data to be transmitted into the USART_DR register (this clears the TXE bit). Repeat this step for each data byte to be sent in single-buffer mode.
8. After writing the last data byte to the USART_DR register, wait until TC=1 to indicate that the last frame has been transmitted. This step is required before disabling USART or entering suspend mode to avoid corrupting the last transmission.

15.3.2.3 Single Byte Communication

Clearing the TXE bit is always performed by a write to the transmit data register.

The TXE bit is set by hardware and it indicates:

- The data has been moved from the TDR register to the shift register and the data transmission has started.
- The TDR register is empty.
- The next data can be written in the USART_DR register without overwriting the previous data.

When the TXEIE bit is set to 1, it generates an interrupt.

During transmission, the data in the TDR register is written into the USART_DR register, and it will be copied into the shift register once the current transmission finishes.

When the transmission has not yet started, writing the data directly into the USART_DR register places it into the shift register. Once data transmission starts, the TXE bit is immediately set to 1.

If the frame has been sent (after the stop bit) and TXE is set to 1, the TC bit will be set to high. If the TCIE bit in the USART_CR1 register is set to 1, an interrupt will be generated.

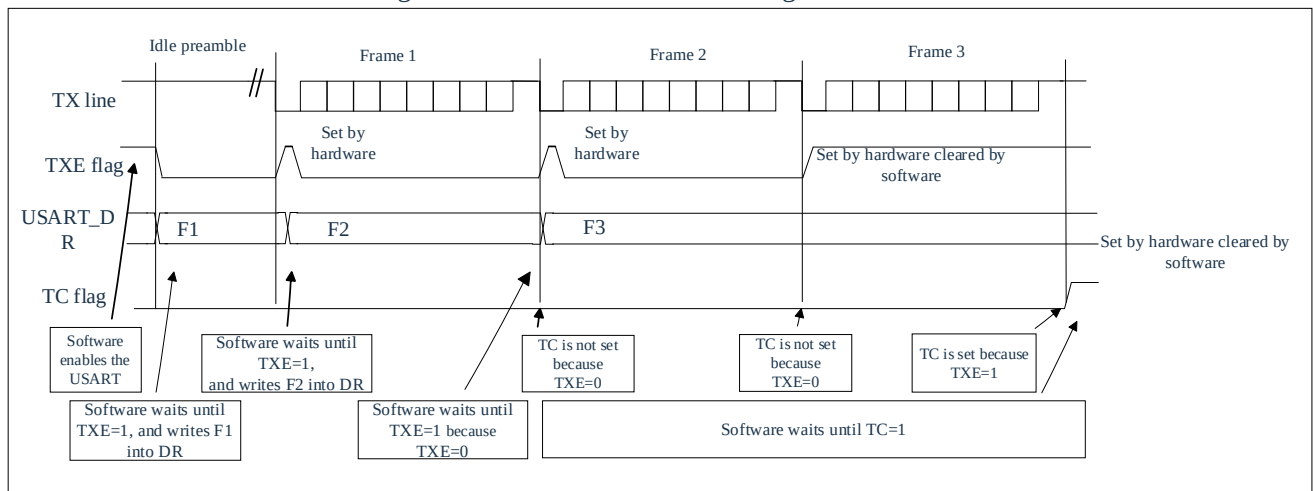
After writing the last data to the USART_DR register, it is necessary to wait until TC=1, and only then can the USART be disabled or the microcontroller be put into low-power mode (see Figure 15-4: TC/TXE Behavior during Transmission).

The TC bit can be cleared via the following software sequence:

1. Read data from the USART_SR register.
2. Write data into the USART_DR register.

Caution: The TC bit can also be cleared by writing a “0” to it, but it is recommended to only use this sequence in multi-buffer communication.

Figure 15-4 TC/TXE behavior during transmission



15.3.2.4 Break Characters

When the SBK bit is set to 1, a break character will be transmitted. The length of the break frame depends on the M bit (see Figure 15-2).

If SBK is set to 1, a break character will be sent on the TX line after the current character transmission is complete. Once the break character is transmitted (during the stop bit of the break character transmission), this bit is automatically reset by the hardware. The USART inserts a logic 1 bit at the end of the previous break frame to ensure proper identification of the next frame's start bit.

Caution: If the software resets the SBK bit before the interrupt transmission begins, the break character will not be transmitted. For two consecutive interrupts, the SBK bit should be set to 1 after the stop bit of the previous break character has been transmitted.

15.3.2.5 Idle Characters

Setting the TE bit drives the USART to send an idle frame before the first data frame.

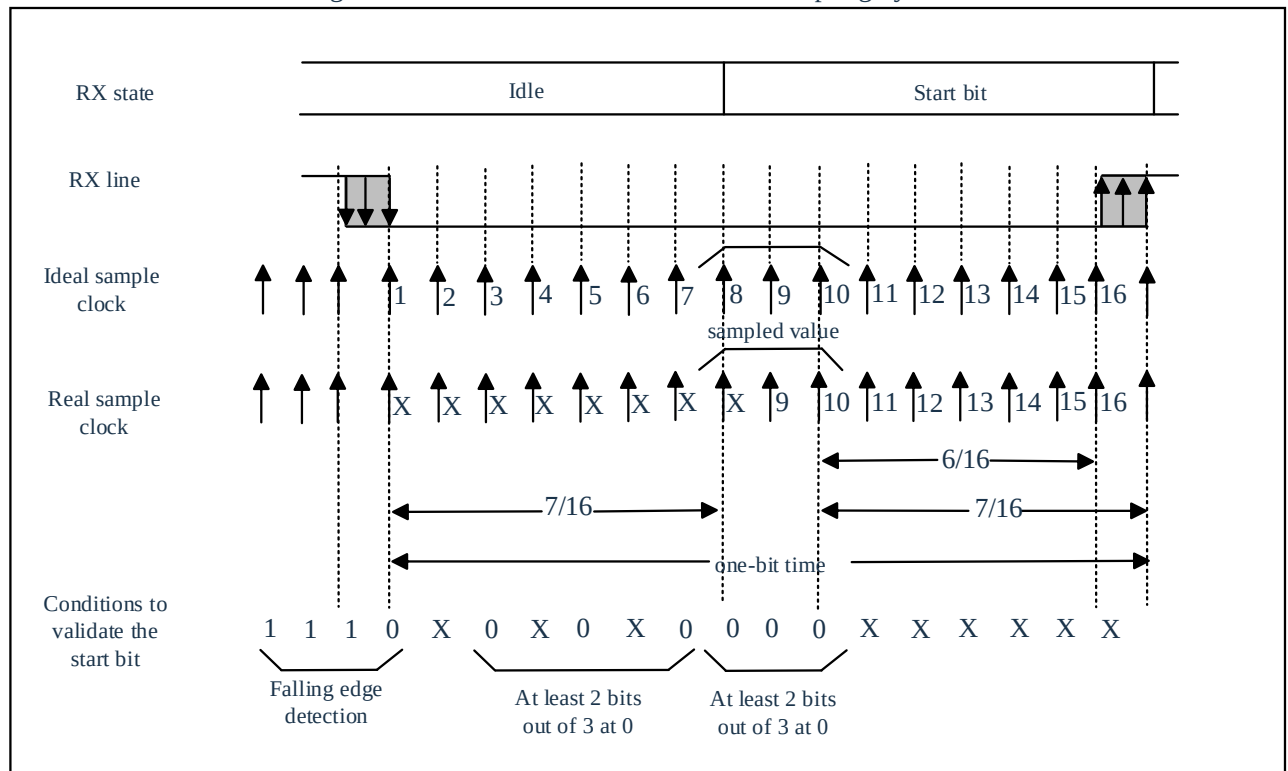
15.3.3 Receiver

The USART can receive data words of either 8 or 9 bits depending on the M bit in the USART_CR1 register.

15.3.3.1 Start Bit Detection

In the USART, the start bit is detected when a specific sequence of samples is recognized. This sequence is: 1 1 1 0 X 0 X 0 X 0 0 0.

Figure 15-5 Start bit detection when oversampling by 16



Caution: If the sequence is not complete, the start bit detection aborts and the receiver returns to the idle state (no flag is set), where it waits for a falling edge.

The start bit is confirmed (RXNE flag set, interrupt generated if RXNEIE=1) if the 3 sampled bits are at 0 (first sampling on the 3rd, 5th and 7th bits finds the 3 bits at 0 and second sampling on the 8th, 9th and 10th bits also finds the 3 bits at 0).

The start bit is validated (RXNE flag set, interrupt generated if RXNEIE=1) but the NF noise flag is set if,

a) for both samplings, 2 out of the 3 sampled bits are at 0 (sampling on the 3rd, 5th and 7th bits and sampling on the 8th, 9th and 10th bits) or

b) for one of the samplings (sampling on the 3rd, 5th and 7th bits or sampling on the 8th, 9th and 10th bits), 2 out of the 3 bits are found at 0.

If neither conditions a. or b. are met, the start detection aborts and the receiver returns to the idle state (no flag is set).

15.3.3.2 Character Reception

During an USART reception, data shifts in least significant bit first (default configuration) through the RX pin. In this mode, the USART_DR register consists of a buffer (RDR) between the internal bus and the receive shift register.

Procedure:

1. Enable USART by writing 1 to the UE bit in the USART_CR1 register.
2. Configure the M bit in USART_CR1 to define the word length.
3. Configure the number of stop bits in USART_CR2.
4. If multi-buffer communication will be used, enable DMA in USART_CR3 (DMAR). Configure the DMA registers as explained in the multi-buffer communication section.
5. Select the desired baud rate using the USART_BRR register.
6. Set the RE bit in USART_CR1 to 1. This operation enables the receiver to begin searching for the start bit.

When a character is received:

- RXNE is set to 1. This indicates that the content of the shift register has been transferred to the RDR, meaning data has been received and can be read (along with its corresponding error flags).
- If RXNEIE is set to 1, an interrupt will be generated.
- If a framing error, noise error, or overrun error is detected during reception, the error flags may be set to 1.
- In multi-buffer mode, RXNE is set to 1 after receiving each byte, and the data register is cleared by a DMA read operation.
- In single-buffer mode, the RXNE bit is cleared by a software read operation of the USART_DR register. The RXNE flag can also be cleared by writing 0 to this bit. The RXNE bit must be cleared before finishing the reception of the next character to avoid an overrun error.

Caution: The RE bit should not be reset during data reception. If the RE bit is disabled during reception, the current byte reception will be aborted.

15.3.3.3 Break Character

When a break character is received, USART will process it as a framing error.

15.3.3.4 Idle Character

When an idle frame is detected, the handling procedure is the same as when data is received; if the IDLEIE bit is set to 1, an interrupt will be generated.

15.3.3.5 Overrun Error

An overrun error occurs when a character is received when RXNE has not been reset. Data can not be transferred from the shift register to the RDR register until the RXNE bit is cleared.

The RXNE flag is set after every byte received. An overrun error occurs if RXNE flag is set when the next data is received or the previous DMA request has not been serviced. When an overrun error occurs:

- The ORE bit is set.
- The RDR content will not be lost. The previous data is available when a read to USART_DR is performed.
- The shift register are overwritten. After that point, any data received during overrun is lost.
- An interrupt is generated if either the RXNEIE bit is set or EIE and DMAR bit are set.
- The ORE bit is cleared by reading the USART_SR register and then the USART_DR register.

Caution: The ORE bit, when set, indicates that at least 1 datum has been lost. There are two possibilities:

- if RXNE=1, then the last valid data is stored in the receive register RDR and can be read,
- if RXNE=0, then it means that the last valid data has already been read and thus there is nothing to be read in the RDR. This case can occur when the last valid data is read in the RDR at the same time as the new (and lost) data is received.

15.3.3.6 Selecting the Proper Oversampling Method

The receiver uses different user-configurable oversampling techniques (except in synchronous mode) to extract valid data from noise. The sampling method can be selected by programming the OVER8 bit in the USART_CR1 register, and the sampling clock can be 16 times or 8 times the baud rate clock (refer to Figures 15-6 and 15-7).

Caution: This product does not support 8x oversampling.

According to the application:

- Select 8x oversampling (OVER8 = 1) for higher speed (up to $f_{\text{PCLK}}/8$). In this case, the receiver's tolerance to clock deviations will be reduced (see Section 15.3.5: Tolerance of the USART Receiver to Clock Deviation).
- Select 16x oversampling (OVER8 = 0) to increase the receiver's tolerance to clock deviations. In this case, the maximum speed is limited to $f_{\text{PCLK}}/16$.

The method used to evaluate logic levels can be selected by programming the ONEBIT bit in the USART_CR3 register. There are two options:

- Triple sampling at the center of the received bit for majority voting. In this case, if the results of the 3 samples for majority voting are not equal, the NF flag will be set to 1.
- Single sampling at the center of the received bit.

According to the application:

- When working in a noisy environment, select triple sampling majority voting (ONEBIT = 0) to reject data when noise is detected (see Table 15-1), as this indicates interference during the sampling process.
- When the line is noise-free, select single sampling (ONEBIT = 1) to increase the receiver's tolerance to clock deviations (see Tolerance of the USART Receiver to Clock Deviation). In this case, the NF flag will never be set to 1.

When noise is detected in the frame:

- The NF flag is set to 1 at the rising edge of the RXNE flag.
- Invalid data is transferred from the shift register to the USART_DR register.

- No interrupt is generated in single-byte communication. However, an interrupt will occur when the RXNE flag is raised. In multi-buffer communication, an interrupt will be triggered when the EIE bit in the USART_CR3 register is set to 1.

The NF flag can be cleared by reading the USART_SR and USART_DR registers sequentially.

Caution: 8x oversampling is not supported in IrDA and LIN modes. In these modes, the OVER8 bit is forced to 0 by hardware.

Figure 15-6: Data sampling when oversampling by 16

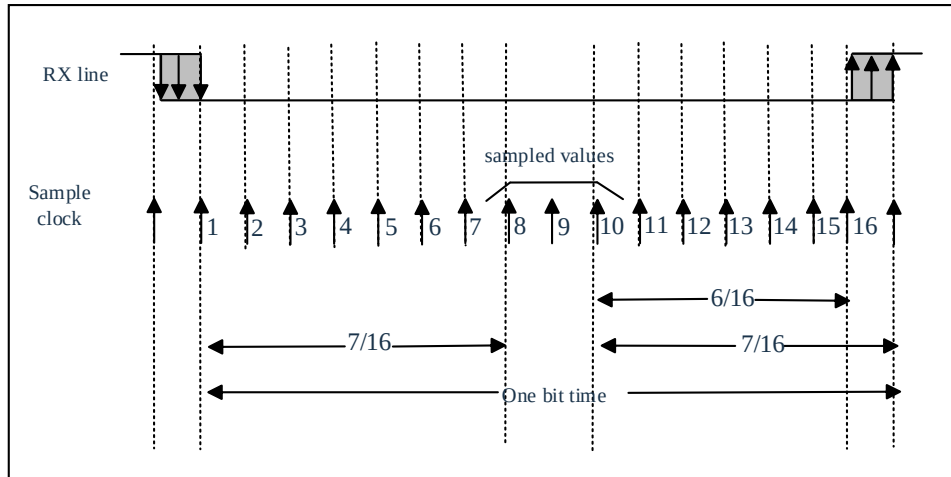


Figure 15-7: Data sampling when oversampling by 8

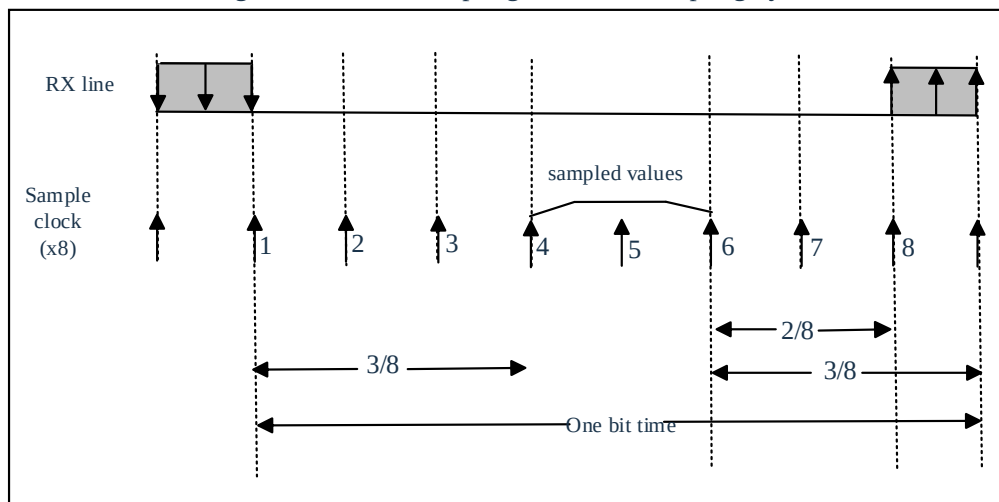


Table 15-1 Noise detection from sampled data

Sampled value	NE status	Received bit value
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

15.3.3.7 Framing Error

A framing error is detected when the stop bit is not recognized on reception at the expected time, following either a de-synchronization or excessive noise.

When the framing error is detected:

- The FE bit is set by hardware
- The invalid data is transferred from the Shift register to the USART_RDR register.
- No interrupt is generated in case of single byte communication. However this bit rises at the same time as the RXNE bit which itself generates an interrupt. In case of multibuffer communication an interrupt is issued if the EIE bit is set in the USART_CR3 register.

The FE bit can be cleared by reading the USART_SR and USART_DR registers sequentially.

15.3.3.8 Configurable Stop Bits During Reception

The number of stop bits to be received can be configured using the control bits in Control Register 2. The number of stop bits can be either 1 or 2 (in normal mode).

1. 1 stop bit: The 1st stop bit is sampled during the 8th, 9th, and 10th sampling.
2. 2 stop bits: The first stop bit is sampled during the 8th, 9th, and 10th sampling. If a framing error is detected during the first stop bit, the FE flag will be set to 1. If a framing error occurs, the second stop bit will not be detected. The RXNE flag will be set to 1 at the end of the first stop bit.

15.3.4 Generation of Decimal Baud Rate

When programming the mantissa and fractional values of USARTDIV, both the receiver and transmitter (Rx and Tx) baud rates are set to the same value.

Formula 1: Baud rate for standard USART (including SPI mode)

$$\text{Tx/Rx baud rate} = f_{\text{CK}} / 8 \times (2 - \text{OVER8}) \times \text{USARTDIV}$$

Formula 2: Baud rate for LIN and IrDA modes

$$\text{Tx/Rx baud rate} = f_{\text{CK}} / 16 \times \text{USARTDIV}$$

USARTDIV is an unsigned fixed-point number stored in the USART_BRR register.

- When OVER8 = 0, the fractional part is encoded as 4 bits and programmed through the DIV_fraction[3:0] bits in the USART_BRR register.
- When OVER8 = 1, the fractional part is encoded as 3 bits and programmed through the DIV_fraction[2:0] bits in the USART_BRR register, with the DIV_fraction[3] bit remaining cleared.

Caution: After writing to the USART_BRR, the baud rate counter is updated to the new value in the baud rate register. Therefore, the baud rate register value should not change during communication.

How to obtain USARTDIV from the USART_BRR register when OVER8 = 0:

Example 1:

If DIV_Mantissa = 0d27 and DIV_Fraction = 0d12 (USART_BRR = 0x1BC),
then the mantissa (USARTDIV) = 0d27, fractional (USARTDIV) = 12/16 = 0d0.75,
therefore, USARTDIV = 0d27.75.

Example 2:

To set USARTDIV = 0d25.62,
this leads to: DIV_Fraction = 16 * times 0d0.62 = 0d9.92,
the closest integer is 0d10 = 0xA,
DIV_Mantissa = Mantissa (0d25.620) = 0d25 = 0x19,
therefore, USART_BRR = 0x19A, thus USARTDIV = 0d25.625.

Example 3:

To set USARTDIV = 0d50.99,
this leads to: DIV_Fraction = 16 * times 0d0.99 = 0d15.84,
the closest integer is 0d16 = 0x10 $\Rightarrow$ DIV_fraction[3:0] overruns $\Rightarrow$ mantissa must carry over,
DIV_Mantissa = Mantissa (0d50.990 + carry) = 0d51 = 0x33,
therefore, USART_BRR = 0x330, thus USARTDIV = 0d51.000.

How to obtain USARTDIV from the USART_BRR register when OVER8 = 1:

Example 1:

If DIV_Mantissa = 0x27 and DIV_Fraction[2:0] = 0d6 (USART_BRR = 0x1B6),
then the mantissa (USARTDIV) = 0d27,
fractional (USARTDIV) = 6/8 = 0d0.75,
therefore, USARTDIV = 0d27.75.

Example 2:

To set USARTDIV = 0d25.62,
this leads to:
DIV_Fraction = 8*times 0d0.62 = 0d4.96,
the closest integer is 0d5 = 0x5,
DIV_Mantissa = Mantissa (0d25.620) = 0d25 = 0x19,
therefore, USART_BRR = 0x195, thus USARTDIV = 0d25.625.

Example 3:

To set USARTDIV = 0d50.99,
this leads to:
DIV_Fraction = 8*times 0d0.99 = 0d7.92,
the closest integer is 0d8 = 0x8 $\Rightarrow$ DIV_fraction[2:0] overruns $\Rightarrow$ mantissa must carry over,
DIV_Mantissa = Mantissa (0d50.990 + carry) = 0d51 = 0x33,
therefore, USART_BRR = 0x0330, thus USARTDIV = 0d51.000.

Table 15-2: Error calculation for programmed baud rates at $f_{\text{PLCK}} = 12 \text{ MHz}$ in both cases of oversampling by 16 ⁽¹⁾

Baud rate		$f_{\text{PLCK}}=12\text{MHz}$		
S.No	Desired	Actual	Programming value in the baud rate register	% Error
1	1.2KBps	1.2KBps	625	0
2	2.4KBps	2.4KBps	312.5	0
3	9.6KBps	9.6KBps	78.125	0
4	19.2KBps	19.2KBps	39.0625	0
5	38.4KBps	38.336KBps	19.5625	0.16
6	57.6KBps	57.692KBps	13	0.16
7	115.2KBps	115.385KBps	6.5	0.16
8	230.4KBps	230.769KBps	3.25	0.16
9	460.8KBps	461.538KBps	1.625	0.16
10	921.6KBps	NA	NA	NA
11	2MBps	NA	NA	NA
12	3MBps	NA	NA	NA

Note 1: The lower the CPU clock the lower the accuracy for a particular baud rate. The upper limit of the achievable baud rate can be fixed with these data.

Table 15-3: Error calculation for programmed baud rates at $f_{PLCK} = 24 \text{ MHz}$ in both cases of oversampling by 16 (1)

Baud rate		$f_{PLCK}=24\text{MHz}$		
S.No	Desired	Actual	Programming value in the baud rate register	% Error
1	1.2KBps	1.2	1250	0
2	2.4KBps	2.4	625	0
3	9.6KBps	9.6	156.25	0
4	19.2KBps	19.2	78.125	0
5	38.4KBps	38.4	39.0625	0
6	57.6KBps	57.554	26.0625	0.08
7	115.2KBps	115.385	13	0.16
8	230.4KBps	230.769	6.5	0.16
9	460.8KBps	461.538	3.25	0.16
10	921.6KBps	923.077	1.625	0.16
11	2MBps	NA	NA	NA
12	3MBps	NA	NA	NA

Note 1: The lower the CPU clock the lower the accuracy for a particular baud rate. The upper limit of the achievable baud rate can be fixed with these data.

15.3.5 Tolerance of the USART Receiver to Clock Deviation

The asynchronous receiver of the USART works correctly only if the total clock system deviation is less than the tolerance of the USART receiver.

The causes which contribute to the total deviation are:

- DTRA: Deviation due to the transmitter error (which also includes the deviation of the transmitter's local oscillator)
- DQUANT: Error due to the baud rate quantization of the receiver
- DREC: Deviation of the receiver's local oscillator
- DTCL: Deviation due to the transmission line (generally due to the transceivers which can introduce an asymmetry between the low-to-high transition timing and the high-to-low transition timing)

$DTRA + DQUANT + DREC + DTCL < \text{USART receiver's tolerance}$

For normal data reception, the tolerance of the USART receiver is equal to the maximum allowable deviation, which depends on the following options:

- The 8-bit or 9-bit character length defined by the M bit in the USART_CR1 register.
- The 8x or 16x oversampling defined by the OVER8 bit in the USART_CR1 register.
- Whether a fractional baud rate is used.
- Whether 1-bit or 3-bit sampling is used for data, depending on the value of the ONEBIT bit in the USART_CR3 register.

Table 15-4: Tolerance of the USART receiver when DIV_Faction is 0

M bit	OVER8 bit=0		OVER8 bit =1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
0	3.75%	4.375%	2.50%	3.75%
1	3.41%	3.97%	2.27%	3.41%

Table 15-5: Tolerance of the USART receiver when DIV_Faction is not 0

M bit	OVER8 bit =0		OVER8 bit =1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
0	3.33%	3.88%	2%	3%
1	3.03%	3.53%	1.82%	2.73%

15.3.6 Multiprocessor Communication

Multiprocessor communication can be performed with USART (multiple USARTs connected in a network). For example, one USART can be the master USART, with its TX output connected to the RX input of the other USARTs. The other USARTs are slave USARTs, whose TX outputs are logically connected together using an AND operation and connected to the RX input of the master USART.

In a multiprocessor configuration, ideally, only the expected message receiver actively receives the complete message, thus reducing redundant USART service overhead caused by all the unaddressed receivers.

Devices that are not addressed can be put into mute mode using the mute functionality. In mute mode:

- The receiver status bit must not be set to 1.
- Any receive interrupts are disabled.
- The RWU bit in the USART_CR1 register is set to 1. RWU can be automatically controlled by hardware or written by software under specific conditions.

Depending on the setting of the WAKE bit in the USART_CR1 register, the USART can enter or exit mute mode using the following two methods:

- If the WAKE bit is reset, idle line detection is used.
- If the WAKE bit is set to 1, address mark detection is used.

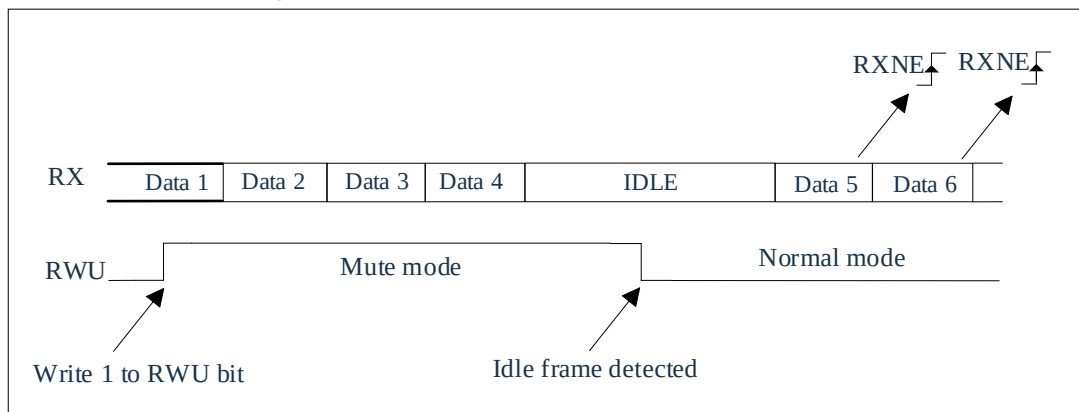
15.3.6.1 Idle Bus Detection (WAKE=0)

When writing 1 to the RWU bit, the USART enters mute mode.

When an idle frame is detected, the USART is awakened. At this point, the RWU bit is cleared by hardware, but the IDLE bit in the USART_SR register is not set to 1. The RWU bit can also be cleared by software writing 0 to it.

Figure 15-8 provides an example of mute mode behavior when using idle line detection.

Figure 15-8 Mute mode with idle line detection



15.3.6.2 Address Mark Detection (WAKE=1)

In this mode, if the MSB of a byte is 1, the byte is recognized as an address; otherwise, it is recognized as data. In the address byte, the target receiver's address is located in the 4 least significant bits. The receiver compares these 4 bits with its own address, which is set in the ADD bit of the USART_CR2 register.

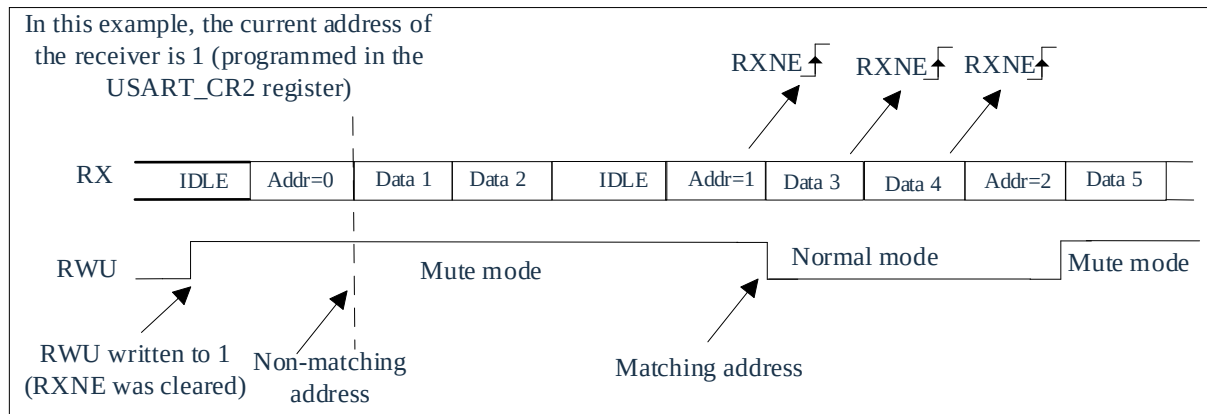
When an address character that does not match its programmed address is received, the USART enters mute mode. At this point, the RWU bit is set to 1 by hardware. Since the USART is already in mute mode, the RXNE flag will not be set to 1 for this address byte, and no interrupt or DMA request will be generated.

When an address character that matches its programmed address is received, it exits mute mode. The RWU bit is cleared, and normal reception of subsequent bytes can begin. Since the RWU bit is cleared, the RXNE flag will be set to 1 for the address character.

If the receiver's buffer does not contain any data (RXNE = 0 in the USART_SR register), the RWU bit can be written with 0 or 1. Otherwise, the write attempt will be ignored.

Figure 15-9 provides an example of mute mode behavior when using address mark detection.

Figure 15-9: Mute mode with address mark detection



15.3.7 Parity Control

Enabling the parity control (to generate a parity bit during transmission and check the parity during reception) can be done by setting the PCE bit in the USART_CR1 register to 1. The possible USART frame formats are listed in Table 15-9, based on the frame length defined by the M bit.

Table 15-9 Frame formats

M bit	PCE bit	USART frame ⁽¹⁾
0	0	SB 8-bit data STB
0	1	SB 7-bit data PB STB
1	0	SB 9-bit data STB
1	1	SB 8-bit data PB STB

Even parity

The parity bit is calculated to obtain an even number of “1s” inside the frame of the 7 or 8 LSB bits (depending on M bit value) and the parity bit.

As an example, if data=00110101, and 4 bits are set, then the parity bit is 0 if even parity is selected (PS bit in USART_CR1 = 0).

Odd parity

The parity bit is calculated to obtain an odd number of “1s” inside the frame made of the 7 or 8 LSB bits (depending on M bit value) and the parity bit.

As an example, if data=00110101 and 4 bits set, then the parity bit is 1 if odd parity is selected (PS bit in USART_CR1 = 1).

Parity check during reception

If the parity check fails, the PE flag is set in the USART_ISR register and an interrupt is generated if PEIE is set in the USART_CR1 register. The PE flag is cleared by software sequence (by reading the status register and then performing a read or write access to the USART_DR data register).

Caution: If the device is awakened by address marking, the MSB of the data is used for address recognition, not the parity bit. Also, the receiver will not perform a parity check on address data (the PE flag is not set to 1 if a parity error occurs).

Parity generation during transmission

If the PCE bit is set in USART_CR1, then the MSB bit of the data written in the data register is transmitted but is changed by the parity bit (even number of “1s” if even parity is selected (PS=0) or an odd number of “1s” if odd parity is selected (PS=1)).

Caution: The software managing the transmission process can activate a software sequence to clear the PE flag (by reading the status register and then performing a read or write access to the data register). In half-duplex mode (depending on software), this could result in the PE flag being unintentionally cleared.

15.3.8 LIN (Local Interconnect Network) Mode

The LIN mode can be selected by setting the LINEN bit in the USART_CR2 register to 1. In LIN mode, the following bits must be cleared:

- CLKEN bit in the USART_CR2 register
- STOP, SCEN, HDSEL, and IREN bits in the USART_CR3 register.

LIN Transmission

Compared to normal USART transmission, when transmitting in the LIN master device, the steps described in section 16.3.2 must be followed, with the following differences:

- The M bit is cleared to configure an 8-bit word length.
- The LINEN bit is set to 1 to enter LIN mode. At this point, setting the SBK bit to 1 sends 13 “0” bits as a break character. Then, a “1” bit will be sent for the next wake-up detection.

LIN Reception

The break detection circuit is implemented on the USART interface. This detection is entirely independent of the normal USART receiver. The break can be detected during idle state or during any frame.

Once the receiver (RE=1 in the USART_CR1 register) is enabled, the circuit starts monitoring the start signal from the RX input. The method of detecting the start bit is the same as the method for searching for the break character or data. After detecting the start bit, the circuit will sample the subsequent bits in the same way as data sampling (at the 8th, 9th, and 10th sampling points). If 10 (when LBDL=0 in the USART_CR2 register) or 11 (when LBDL=1 in the USART_CR2 register) consecutive “0” bits are detected, followed by a delimiter, the LBD flag in the USART_SR register will be set to 1. If LBDIE=1, an interrupt will be generated. Before confirming the break, the delimiter is checked because it indicates that the RX line has returned to a high level.

If a “1” is sampled before the 10th or 11th sample, the break detection circuit will cancel the current detection and re-search for the start bit.

If LIN mode is disabled (LINEN=0), the receiver will work as a normal USART and will no longer perform break detection.

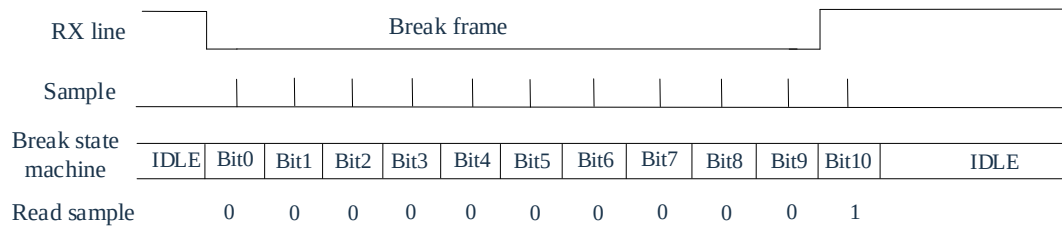
If LIN mode is enabled (LINEN=1), the receiver will stop as soon as a frame error occurs (for example, if a stop bit is detected at “0”, which can happen in any break frame) until the break detection circuit receives a “1” (when the break word is incomplete) or a delimiter (when the break is detected).

Figure 15-10 shows the behavior of the break detection state machine and break flag in LIN mode with an 11-bit break length (LBDL=1).

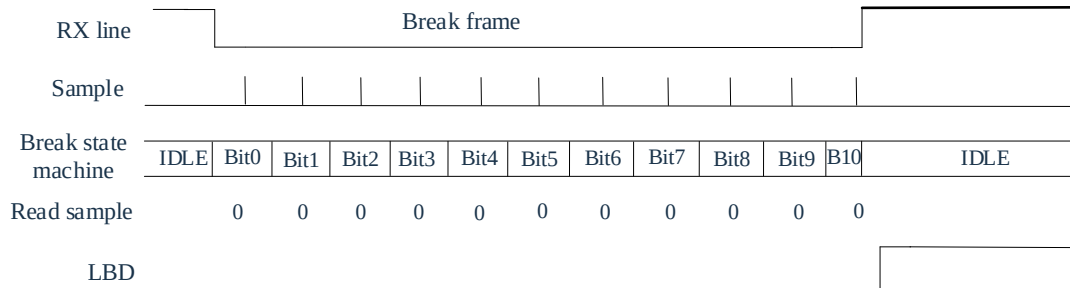
Figure 15-11 lists examples of break frames in LIN mode, including break detection and frame error detection.

Figure 15-10 Break detection in LIN mode (11-bit break length — LBDL = 1)

Example 1: Break signal not long enough => Discard break, LBD not set to 1



Example 2: Break signal exactly long enough => Break detected, LBD set to 1



Example 3: Break signal long enough => Break detected, LBD set to 1

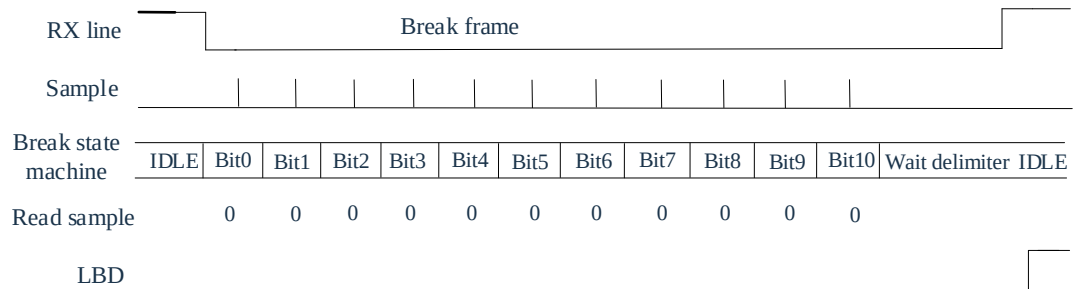
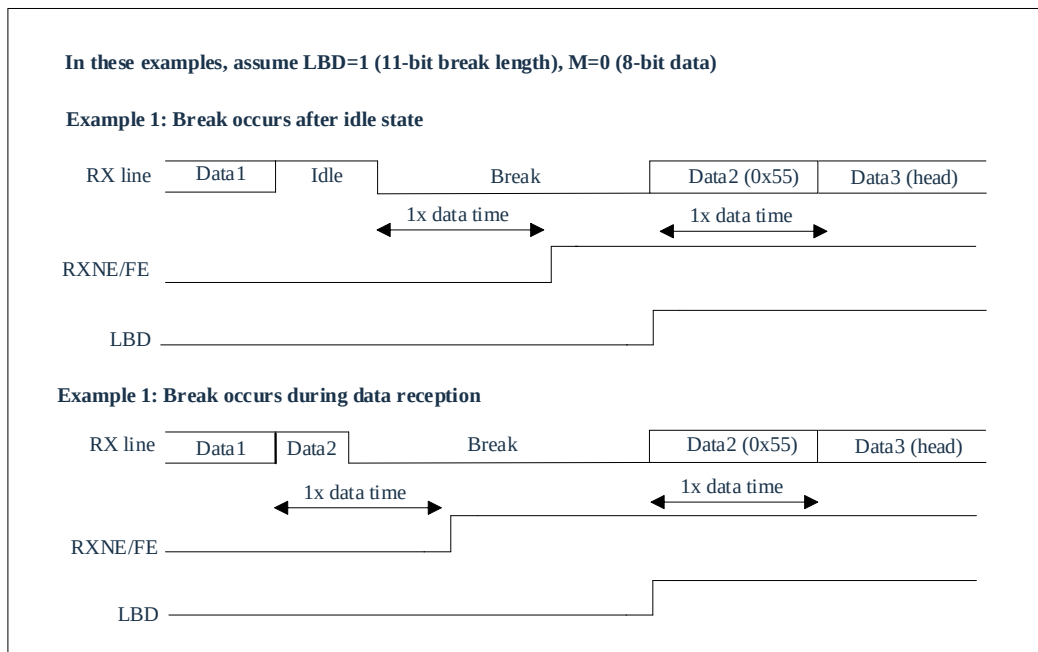


Figure 15-11 Break detection and frame error detection in LIN mode



15.3.9 USART Synchronous Mode

Synchronous mode is selected by setting the CLKEN bit in the USART_CR2 register to 1. In synchronous mode, the following bits must be cleared:

- LINEN bit in the USART_CR2 register.
- SCEN, HDSEL, and IREN bits in the USART_CR3 register.

In synchronous mode, the user can control bidirectional synchronous serial communication in master mode via USART. The SCLK pin is the output for the USART transmitter clock. No clock pulses are sent to the SCLK pin during the start or stop bit period. A clock pulse will (or may not) be generated during the last valid data bit (address mark), depending on the state of the LBCL bit in the USART_CR2 register. By using the CPOL bit in the USART_CR2 register, the user can select the clock polarity, and by using the CPHA bit in the same register, the user can select the external clock phase (refer to Figures 15-12, 15-13, and 15-14).

During idle state, header mode, and break transmission, the external SCLK clock is inactive.

The operation of the USART transmitter in synchronous mode is identical to its operation in asynchronous mode. However, since SCLK is synchronized with TX (according to CPOL and CPHA), the data on TX is synchronous in this mode.

In this mode, the operation of the USART receiver differs from its operation in asynchronous mode. If RE=1, data is sampled on SCLK (on the rising or falling edge, depending on CPOL and CPHA) without any oversampling. It is essential that the setup and hold times (depending on the baud rate: 1/16 bit time) meet the required specifications.

Caution: The SCLK pin can be combined with the TX pin. Therefore, clock pulses are only provided when the transmitter is enabled (TE=1) and data is being transmitted (i.e., data has been written to the USART_DR data register). This means that synchronous data cannot be received unless data is being transmitted.

When both the transmitter and receiver (TE=RE=0) are disabled, the LBCL, CPOL, and CPHA bits must be selected to ensure proper clock pulse operation. These bits should not be changed once the transmitter or receiver is enabled.

It is recommended to set TE and RE to 1 simultaneously to minimize the receiver's setup and hold times.

The USART only supports master mode: it cannot receive or send data that is related to the input clock (i.e., SCLK is always output).

Figure 15-12 USART synchronous transmit example

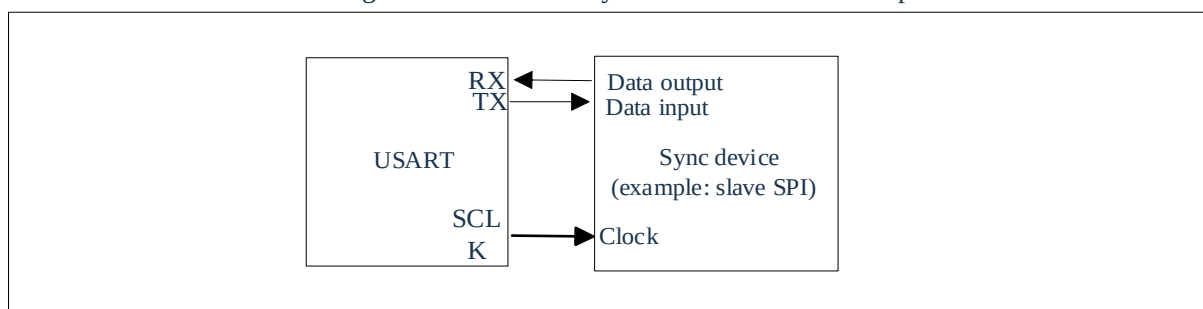


Figure 15-13: USART data clock timing diagram (M=0)

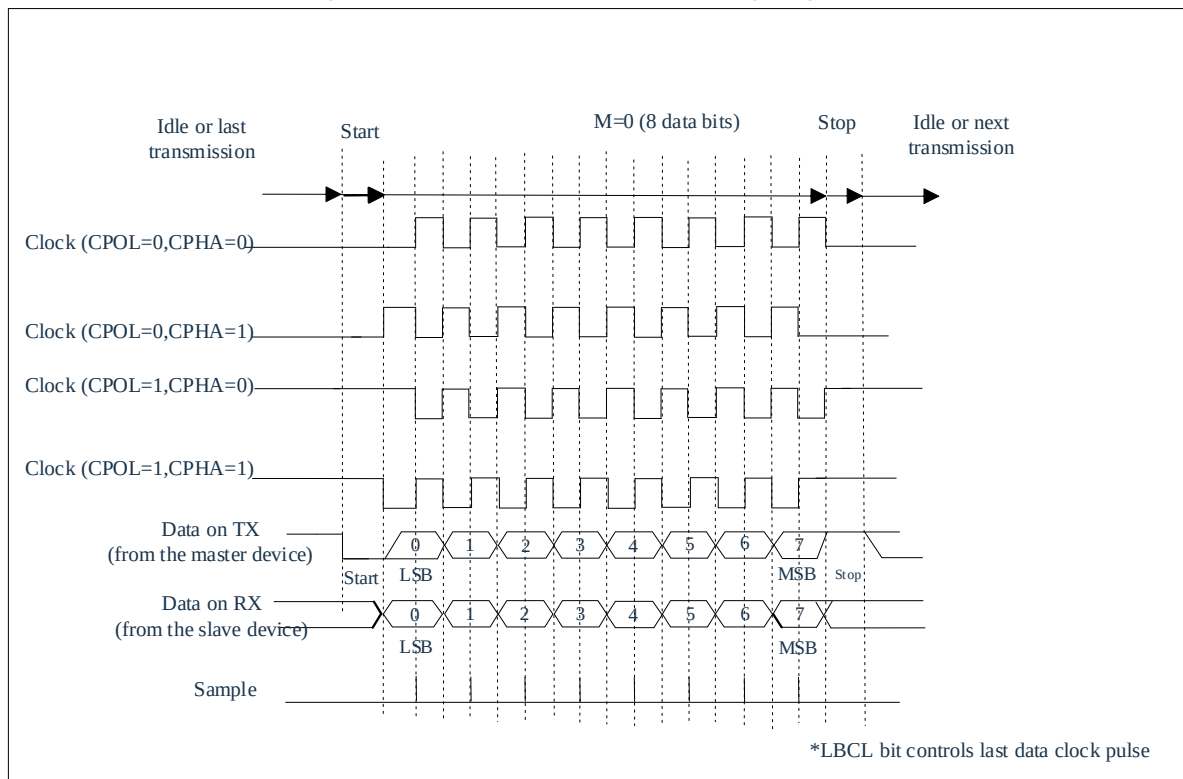


Figure 15-14: USART data clock timing diagram (M=1)

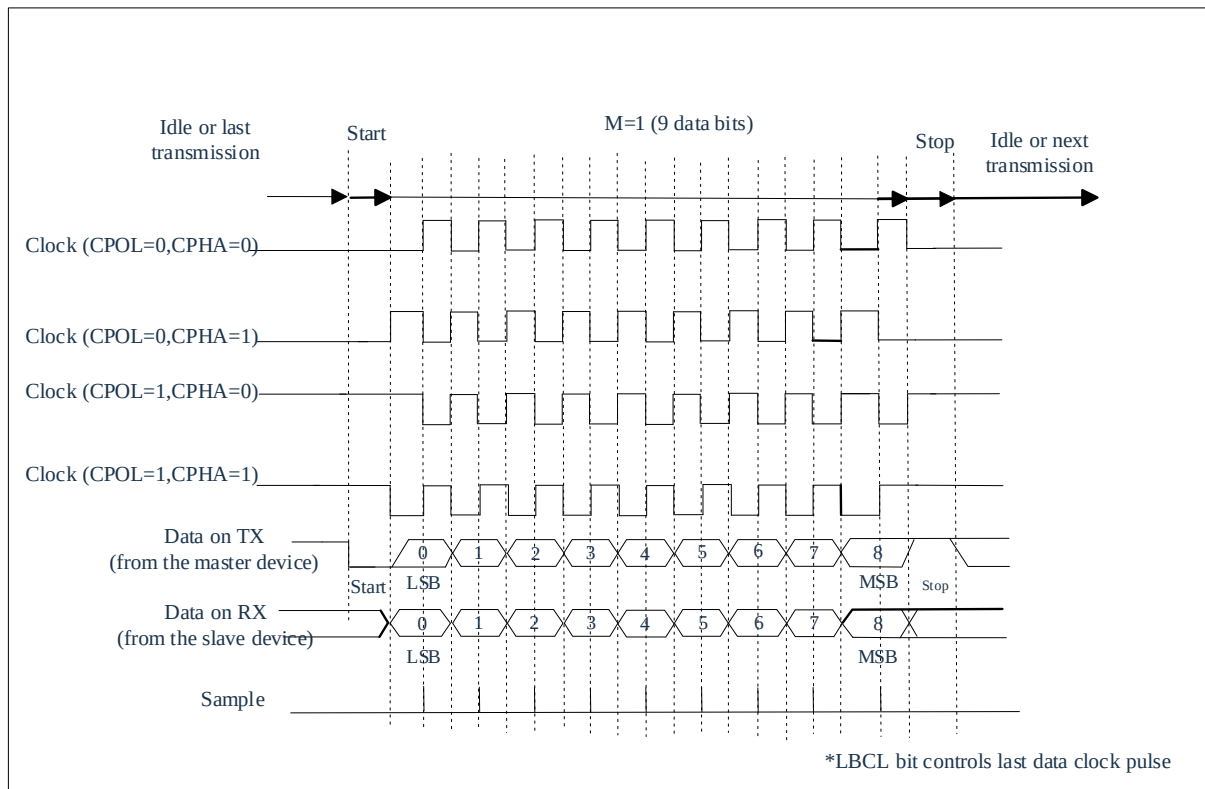
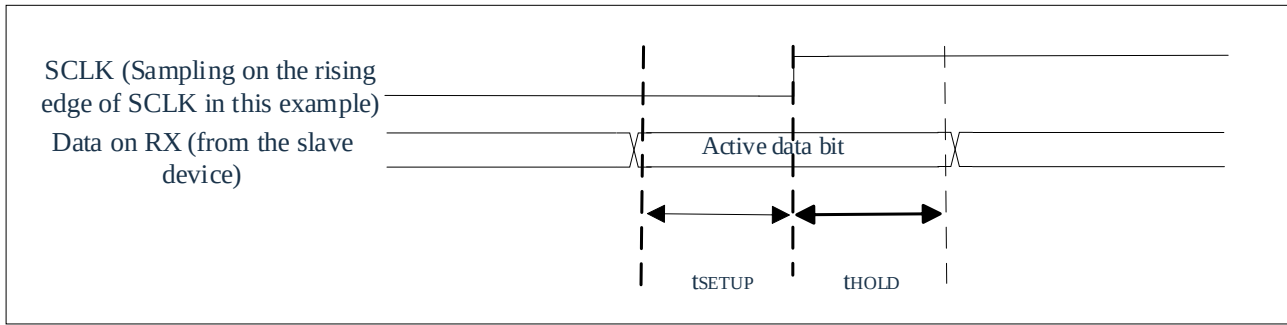


Figure 15-15: RX data setup/hold time



15.3.10 Single-Wire Half-Duplex Communication

Single-wire Half-duplex mode is selected by setting the HDSEL bit in the USART_CR3 register. In this mode, the following bits must be kept cleared:

- CLKEN bit in the USART_CR2 register,
- SCEN and IREN bits in the USART_CR3 register.

The USART can be configured to follow a Single-wire Half-duplex protocol where the TX and RX lines are internally connected. The selection between half- and Full-duplex communication is made with a control bit HALF DUPLEX SEL (bit HDSEL in USART_CR3).

As soon as HDSEL is written to 1:

- The TX and RX lines are internally connected
- The RX pin is no longer used
- The TX pin is always released when no data is transmitted. Thus, it acts as a standard I/O in idle or in reception. It means that the I/O must be configured so that TX is configured as alternate function open-drain with an external pull-up. Apart from this, the communication protocol is similar to normal USART mode. Any conflicts on the line must be managed by software (by the use of a centralized arbiter, for instance). In particular, the transmission is never blocked by hardware and continues as soon as data is written in the data register while the TE bit is set.

15.3.11 IrDA SIR ENDEC Module

By setting the IREN bit in the USART_CR3 register to 1, IrDA mode is selected. In IrDA mode, the following bits must be cleared:

- LINEN, STOP, and CLKEN bits in the USART_CR2 register;
- SCEN and HDSEL bits in the USART_CR3 register.

IrDA SIR physical layer specifies the use of Inverted Return to Zero (RZI) modulation scheme, which represents logical 0 with infrared light pulses (see Figure 15-18).

The SIR transmitter encoder modulates the non-return-to-zero (NRZ) bitstream sent by USART. The output pulse stream is sent to the external output driver and infrared LED. The maximum bit rate supported by the SIR ENDEC in USART is 115.2 Kbps. In normal mode, the pulse width is specified as 3/16 of one bit period. The SIR receiver decoder demodulates the zero-return bitstream received from the infrared detector and outputs the received NRZ serial bitstream to USART. In the idle state, the decoder input is typically high (mark state).

The polarity of the transmitter encoder output is opposite to that of the decoder input. When the decoder input is low, a start bit is detected.

- IrDA is a half-duplex communication protocol. If the transmitter is busy (for example, USART is sending data to the IrDA encoder), the IrDA decoder will ignore all data on the IrDA receive line. If the receiver is busy (for example, USART is receiving decoded data from USART), IrDA will not encode data sent from USART to IrDA TX. When receiving data, transmission should be avoided simultaneously, as this may corrupt the data to be transmitted.
- A “0” is transmitted as a high-level pulse, while “1” is transmitted as “0.” In normal mode, the pulse width is specified as 3/16 of the selected bit period (see Figure 15-19).
- The SIR decoder is used to convert IrDA-compatible received signals to USART bitstream.
- The SIR receiver logic treats high-level states as logic “1” and low-level pulses as logic “0.”
- The polarity of the transmitter encoder output is opposite to that of the decoder input. The SIR output is low during idle times.
- IrDA specification requires the pulse tolerance to be greater than 1.41 us. Acceptable pulse width can be set through register configuration. The interference detection logic at the receiver end will filter out pulses shorter than 2 PSC cycles (PSC is the pre-scaler value programmed in the IrDA low-power baud rate register USART_GTPR). Pulses shorter than 1 PSC cycle will be rejected, while pulses greater than 1 but less than 2 cycles may be accepted or rejected. Pulses greater than 2 cycles will be accepted as valid pulses. When PSC=0, the IrDA encoder/decoder does not work.
- The receiver can communicate with a low-power transmitter.
- In IrDA mode, the STOP bit in the USART_CR2 register must be configured to “1 stop bit”.

15.3.11.1 IrDA Low Power Mode

Transmitter:

In low power mode, the pulse width is no longer held at 3/16 of the bit period. Instead, the pulse width becomes three times the low-power baud rate, with a minimum of 1.42 MHz. Typically, this value is 1.8432 MHz ($1.42 \text{ MHz} < \text{PSC} < 2.12 \text{ MHz}$). The programmable pre-scaler in low power mode divides the system clock to achieve this value.

Receiver:

Receiving in low power mode is similar to receiving in normal mode. To perform interference detection, USART should discard pulses shorter than 1/PSC duration. A pulse is valid as a low level only if its duration is greater than 2 IrDA low power baud clock cycles (the PSC value in the USART_GTPR register).

Caution: Pulses shorter than two but greater than one PSC cycle may be accepted or rejected.

The receiver's setup time should be managed by software. The IrDA physical layer specification mandates at least a 10 ms delay between transmission and reception (IrDA is a half-duplex protocol).

Figure 15-18 IrDA SIR ENDEC—block diagram

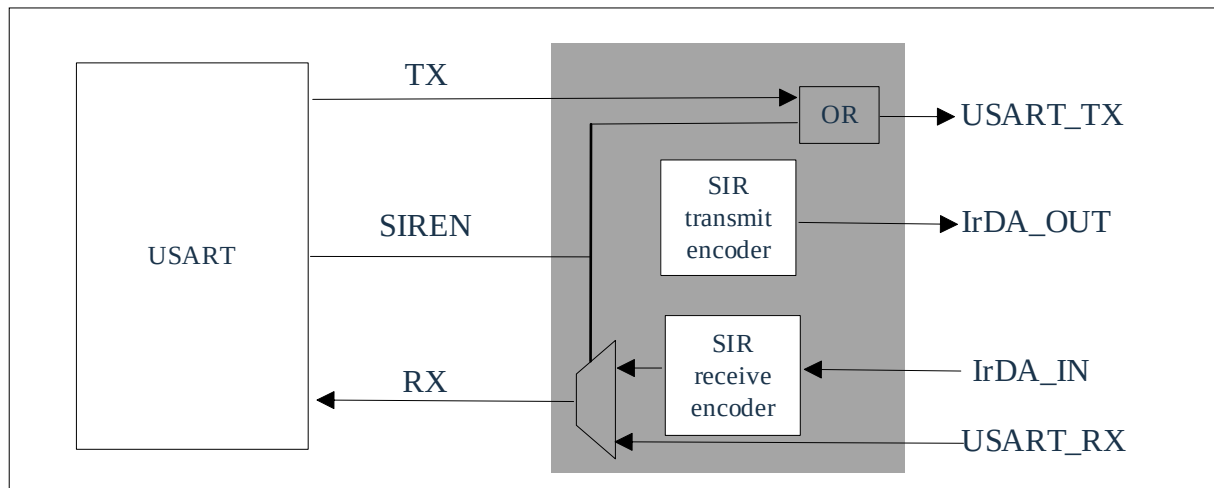
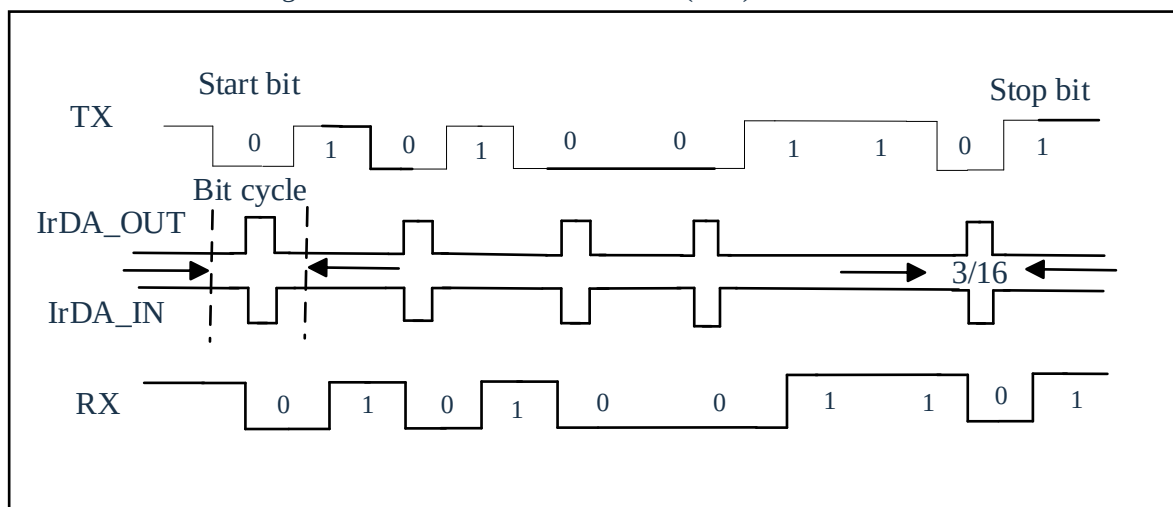


Figure 15-19 IrDA Data Modulation (3/16)—normal mode



15.3.12 2 Using DMA for Continuous Communication

USART can use DMA for continuous communication. The DMA requests for the receive buffer and the transmit buffer are independent.

Caution: For information on the availability of the DMA controller, refer to the product specification. If DMA is not available for the product, use USART as explained in Section 16.3.2 or 16.3.3. The continuous communication can be achieved by clearing the TXE/RXNE flags in the USART_SR register.

15.3.12.1 Transmission Using DMA

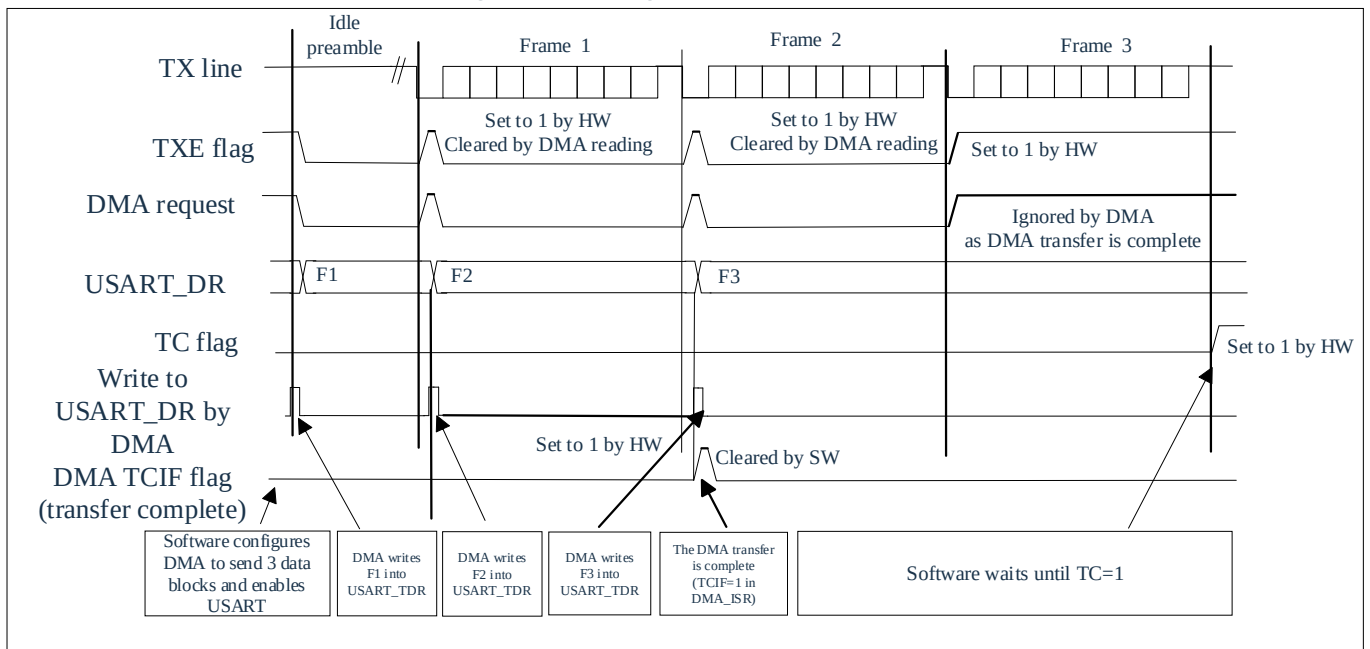
Setting the DMAT bit in the USART_CR3 register to 1 enables DMA mode for transmission. When TXE is set to 1, data can be loaded from SRAM (through DMA configuration, see the DMA section) into the USART_DR register. To map a DMA channel for USART transmission, follow these steps (x represents the channel number):

1. Write the address of the USART_DR register into the DMA control register to configure it as the destination address for the transfer. After each TXE event, data is moved from memory to this address.
2. Write the memory address into the DMA control register to configure it as the source address for the transfer. After each TXE event, data is loaded from this memory area into the USART_DR register.
3. Configure the total number of bytes to be transferred in the DMA control register.
4. Configure the channel priority in the DMA register.
5. Depending on the application requirements, generate a DMA interrupt after half or all of the transfer is complete.
6. Write 0 to the TC bit in the SR register to clear it.
7. Activate the channel in the DMA register.

When the data transfer amount set in the DMA controller is reached, an interrupt is generated on the DMA channel's interrupt vector.

In transmit mode, after DMA performs the write operation for all the data to be transmitted (TCIF flag in the DMA_ISR register is set to 1), the TC flag can be monitored to ensure that USART communication is complete. This step must be performed before disabling USART or entering stop mode to avoid corruption of the last transmission. Software must wait until TC=1. The TC flag must remain cleared throughout the entire data transmission and will be set to 1 by hardware after the last frame transmission ends.

Figure 15-20 sing DMA for transmission



15.3.12.2 Reception Using DMA

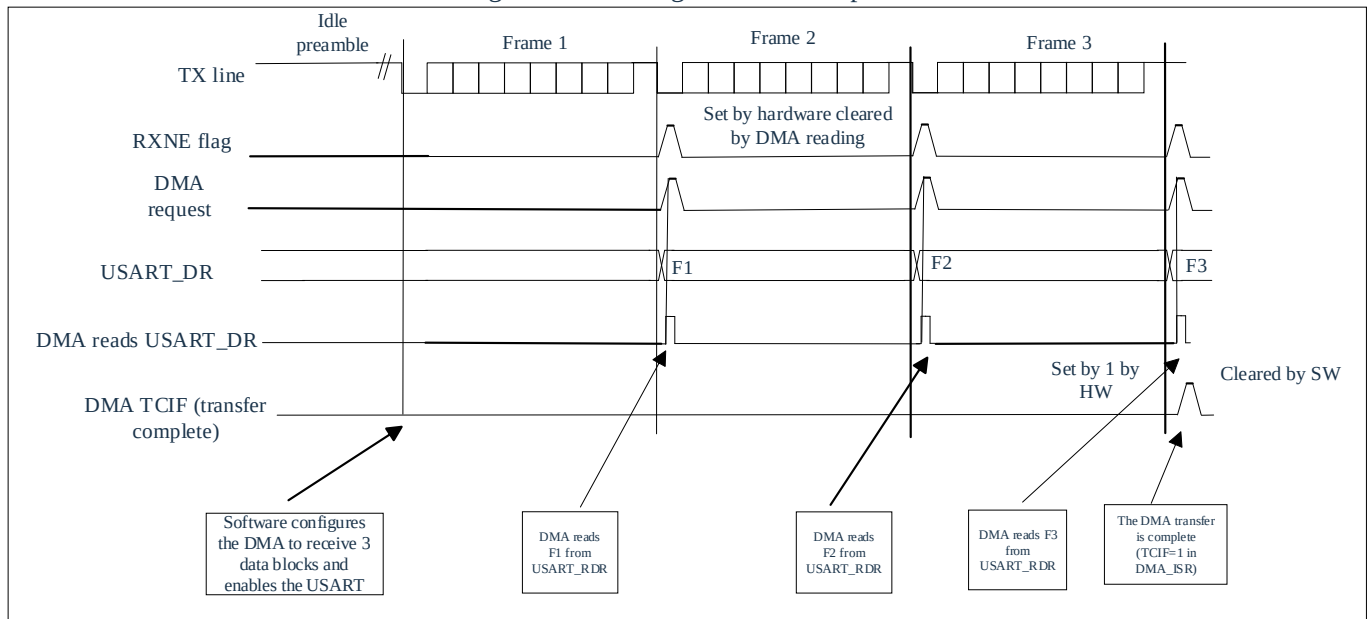
DMA mode can be enabled for reception by setting the DMAR bit in USART_CR3 register. Data is loaded from the USART_RDR register to a SRAM area configured using the DMA peripheral (refer to DMA Specification) whenever a data byte is received. To map a DMA channel for USART reception, use the following procedure:

1. Write the USART_RDR register address in the DMA control register to configure it as the source of the transfer. The data is moved from this address to the memory after each RXNE event.
2. Write the memory address in the DMA control register to configure it as the destination of the transfer. The data is loaded from USART_RDR to this memory area after each RXNE event.
3. Configure the total number of bytes to be transferred to the DMA control register.
4. Configure the channel priority in the DMA control register
5. Configure interrupt generation after half/ full transfer as required by the application.
6. Activate the channel in the DMA control register.

When the data transfer amount set in the DMA controller is reached, the DMA controller generates an interrupt on the interrupt vector of the DMA channel. In the interrupt service routine, the DMAR bit in the USART_CR3 register should be cleared by software.

Caution: If DMA is used for reception, do not enable the RXNEIE bit.

Figure 15-21 Using DMA for reception



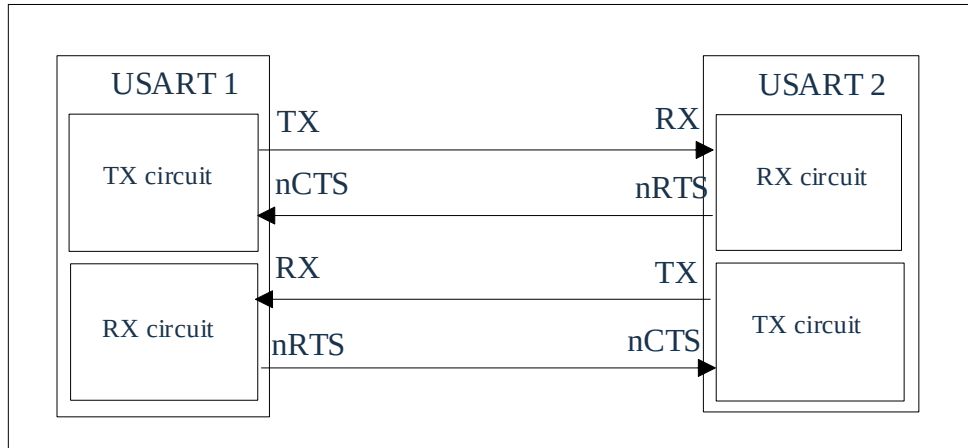
15.3.12.3 Error Flagging and Interrupt Generation in Multibuffer Communication

In multibuffer communication if any error occurs during the transaction the error flag is asserted after the current byte. An interrupt is generated if the interrupt enable flag is set. For framing error, overrun error and noise flag which are asserted with RXNE in single byte reception, there is a separate error flag interrupt enable bit (EIE bit in the USART_CR3 register), which, if set, enables an interrupt after the current byte if any of these errors occur.

15.3.13 Hardware Flow Control

Using the nCTS input and nRTS output can control the serial data flow between two devices. Figure 15-22 shows how to connect two devices in this mode.

Figure 15-22: Hardware flow control between 2 USARTs

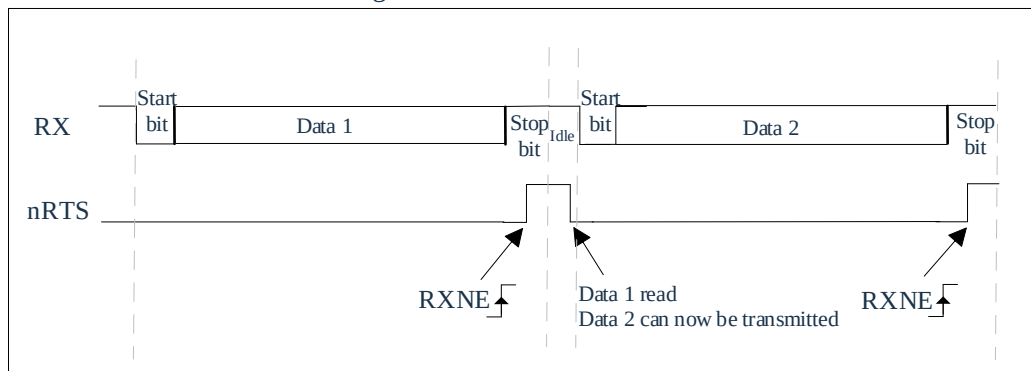


RTS and CTS flow control can be enabled independently by writing the RTSE and CTSE bits respectively to 1 (in the USART_CR3 register).

15.3.13.1 RTS Flow Control

If the RTS flow control is enabled (RTSE=1), then nRTS is deasserted (tied low) as long as the USART receiver is ready to receive a new data. When the receive register is full, nRTS is asserted, indicating that the transmission is expected to stop at the end of the current frame. Figure 15-23 shows an example of communication with RTS flow control enabled.

Figure 15-23 RTS flow control

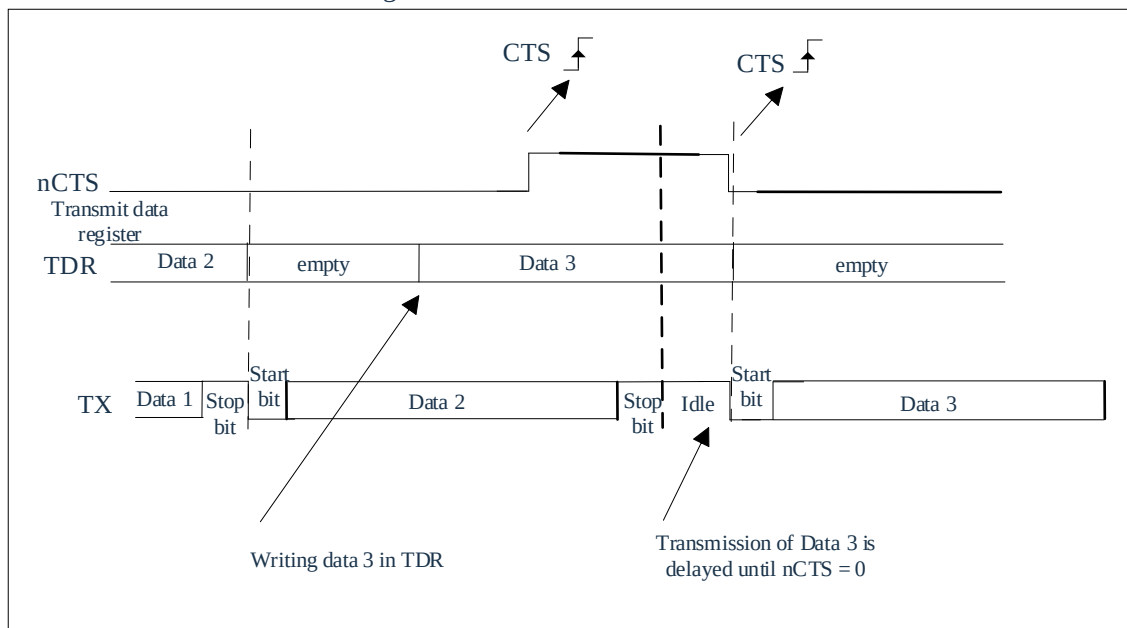


15.3.13.2 CTS Flow Control

If the CTS flow control is enabled (CTSE=1), then the transmitter checks the nCTS input before transmitting the next frame. If nCTS is deasserted (tied low), then the next data is transmitted (assuming that data is to be transmitted, in other words, if TXE=0), else the transmission does not occur. when nCTS is asserted during a transmission, the current transmission is completed before the transmitter stops.

When CTSE=1, the CTSIF status bit is automatically set by hardware as soon as the nCTS input toggles. It indicates when the receiver becomes ready or not ready for communication. An interrupt is generated if the CTSIE bit in the USART_CR3 register is set. The figure below shows an example of communication with CTS flow control enabled.

Figure 15-24 CTS flow control



Caution: Special behavior of the stop frame: After enabling CTS flow, the transmitter will not check the nCTS input state when sending the stop signal.

15.4 USART Interrupts

Table 15-10 USART interrupt requests

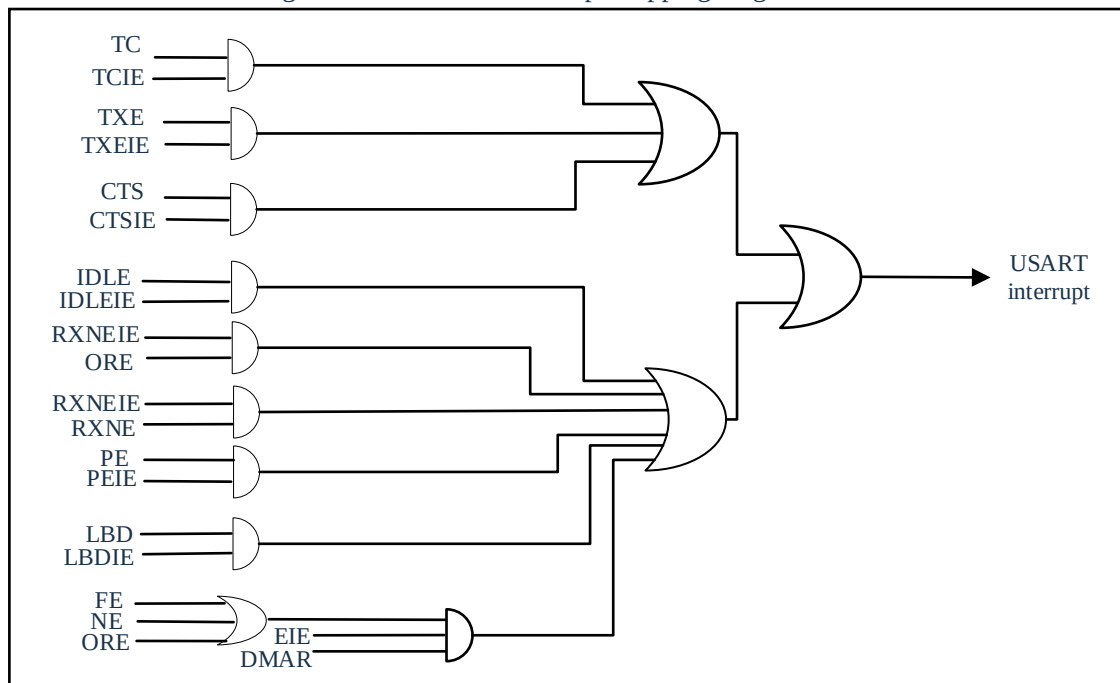
Interrupt event	Event flag	Enable control bit
Transmit data register empty	TXE	TXEIE
CTS flag	CTS	CTSIE
Transmission complete	TC	TCIE
Receive data register not empty (data ready to be read)	RXNE	RXNEIE
Overrun error detected	ORE	
Idle line detected	IDLE	IDLEIE
Parity error	PE	PEIE
Line break down	LBD	LBDIE
Noise Flag, Overrun error and Framing Error in multibuffer communication	NF or ORE or FE	EIE

The USART interrupt events are connected to the same interrupt vector (see Figure 15-25).

- During transmission: Transmission Complete, Clear to Send, Transmit data Register empty interrupt.
- During reception: Idle Line detection, Overrun error, Receive data register not empty, Parity error, LIN line break down detection, Noise Flag (multi-buffer communication only), and Framing Error (multi-buffer communication only).

These events generate an interrupt if the corresponding Enable Control Bit is set.

Figure 15-25 USART interrupt mapping diagram



15.5 USART Mode Configuration

Table 15-11 USART mode configuration⁽¹⁾

USART mode	USART
Asynchronous mode	X
Hardware flow control	X
Multibuffer communication (DMA)	X
Multiprocessor communication	X
Synchronous	X
-	NA
Half-duplex (single-wire mode)	X
IrDA	X
LIN	X

Note 1: X=available; NA=not applicable.

15.6 Register Mapping

(USART base address = 0x4007_1000) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Address	R/W	Description	Reset value
USART_SR	0x00	R/W	USART Status Register	0xC0
USART_DR	0x04	R/W	USART Data Register	0x00
USART_BRR	0x08	R/W	USART Baud Rate Register	0x00
USART_CR1	0x0C	R/W	USART Control Register 1	0x00
USART_CR2	0x10	R/W	USART Control Register 2	0x00
USART_CR3	0x14	R/W	USART Control Register 3	0x00
USART_GTPR	0x18	R/W	USART Guard Time and Prescaler Register	0x00

15.6.1 USART Status Register (USART_SR)

Bit	Symbol	Description	Reset value
31:10	--	Reserved	--
9	CTS	CTS flag: If the CTSE bit is set to 1, this bit is set to 1 by hardware when the nCTS input changes. It can be cleared by software (by writing 0 to this bit). If CTSIE in the USART_CR3 register is set to 1, an interrupt will be generated. 0: No change on the nCTS status line 1: Change occurred on the nCTS status line	0
8	LBD	LIN break detection flag: This bit is set to 1 by hardware when a LIN break is detected. It can be cleared by software (by writing 0 to this bit). If LBDIE in the USART_CR2 register is set to 1, an interrupt will be generated. 0: No LIN break detected 1: LIN break detected Caution: If LBDIE=1, an interrupt will be generated when LBD=1.	0
7	TXE	Transmit data register empty: This bit is set to 1 by hardware when the content of the TDR register has been transmitted to the shift register. If TXEIE in the USART_CR1 register is set to 1, an interrupt will be generated. The bit is cleared by writing to the USART_DR register. 0: Data not yet transferred to the shift register 1: Data transferred to the shift register Caution: This bit is used during single-buffer transmission.	1
6	TC	Transmission complete: This bit is set to 1 by hardware when the transmission of a frame containing data is completed and TXE is set to 1. If TCIE in the USART_CR1 register is set to 1, an interrupt will be generated. This bit is cleared by a software sequence (read the USART_SR register, then write to the USART_DR register). The TC bit can also be cleared by writing 0 to it. It is recommended to use this clearing sequence only in multibuffer communication. 0: Transmission not complete 1: Transmission complete	1
5	RXNE	Read data register not empty: This bit is set to 1 by hardware when the content of the RDR shift register has been transferred to the USART_DR register. If RXNEIE in the USART_CR1 register is set to 1, an interrupt will be generated. The bit is cleared by reading the USART_DR register. The RXNE flag can also be cleared by writing 0 to this bit. It is recommended to use this clearing sequence only in multibuffer communication. 0: Data not received 1: Received data ready to be read	0
4	IDLE	IDLE line detected: This bit is set to 1 by hardware when an idle line is detected. If IDLEIE in the USART_CR1 register is set to 1, an interrupt will be generated. This bit is cleared by a software sequence (read the USART_SR register, then read the USART_DR register). 0: No idle line detected 1: Idle line detected Caution: The IDLE bit will only be set to 1 again when the RXNE bit is set (i.e., when a new idle line occurs).	0
3	ORE	Overrun error: When RXNE=1, this bit is set to 1 by hardware when the data currently being received in the shift register is ready to be transferred to the RDR register. If RXNEIE in the USART_CR1 register is set to 1, an interrupt will be generated. This bit is cleared by a software sequence (read the USART_SR register, then read the USART_DR	0

		register). 0: No overrun error 1: Overrun error detected Caution: When this bit is set to 1, the contents of the RDR register are not lost, but the shift register will be overwritten. If the EIE bit is set to 1, an interrupt will be generated for the ORE flag during multibuffer communication.	
2	NF	Noise detected flag: This bit is set to 1 by hardware when noise is detected on the received frame. The bit is cleared by a software sequence (read the USART_SR register, then read the USART_DR register). 0: No noise detected 1: Noise detected Caution: If the EIE bit is set to 1, this bit will not generate an interrupt during multibuffer communication, as the time this bit appears coincides with the time the RXNE bit, triggered by the NF flag, generates an interrupt. Caution: When the line is noise-free, the NF flag can be disabled by setting the ONEBIT bit to 1 to increase USART tolerance for deviations.	0
			0
1	FE	Framing error: This bit is set to 1 by hardware when desynchronization, excessive noise, or an interrupt character is detected. The bit is cleared by a software sequence (read the USART_SR register, then read the USART_DR register). 0: No framing error detected 1: Framing error or interrupt character detected Caution: This bit does not generate an interrupt, as its appearance coincides with the appearance of the RXNE bit, which generates an interrupt. If the current word being transmitted causes both a framing error and an overrun error, the word is transmitted, and only the ORE bit is set to 1. If the EIE bit is set to 1, an interrupt will be generated for the FE flag during multibuffer communication.	0
			0
0	PE	Parity error: This bit is set to 1 by hardware when a parity error occurs in receiver mode. The bit is cleared by a software sequence (read the status register, then perform a read or write access to the USART_DR data register). Software must wait for the RXNE flag to be set before clearing the PE bit. If PEIE in the USART_CR1 register is set to 1, an interrupt will be generated. 0: No parity error 1: Parity error	0

15.6.2 USART Data Register (USART_DR)

Bit	Symbol	Description	Reset value
31:9	--	Reserved	--
8:0	DR[8:0]	Data values: It contains either the data characters received or the data characters transmitted, depending on whether the operation is a “read” or “write” operation. Since the data register consists of two registers, one for transmission (TDR) and one for reception (RDR), it has dual functionality (read and write). The TDR register provides a parallel interface between the internal bus and the output shift register. The RDR register provides a parallel interface between the input shift register and the internal bus. When parity bits are enabled (the PCE bit in the USART_CR1 register is set to 1), during transmission, the value of the MSB (bit 7 or bit 8, depending on the data length) is replaced by the parity bit, so the value has no effect. When receiving with parity bits enabled, the value read from the MSB bit is the received parity bit.	0x0

15.6.3 USART Baud Rate Register (USART_BRR)

Caution: If the TE or RE bit is disabled, the baud rate counter will stop counting.

Bit	Symbol	Description	Reset value
31:16	--	Reserved	--
15:4	DIV_Mantissa[11:0]	Fraction of USARTDIV: These 12 bits are used to define the fractional part of the USART divisor (USARTDIV).	0x0
3:0	DIV_Fraction[3:0]	Decimal of USARTDIV: These 4 bits are used to define the decimal part of the USART divisor (USARTDIV). When OVER8 = 1, the DIV_Fraction3 bit is ignored, and it must be kept cleared.	0x0

15.6.4 USART Control Register 1 (USART_CR1)

Bit	Symbol	Description	Reset value
31:16	--	Reserved	--
15	OVER8	Oversampling mode: 0: 16x oversampling 1: 8x oversampling (not allowed)	0
14	--	Reserved	--
13	UE	USART enable: When this bit is cleared, the USART prescaler and output will stop, and the current byte transmission will end to reduce power consumption. This bit is set and cleared by software. 0: Disable USART prescaler and output 1: Enable USART	0
12	M	Word length: This bit determines the word length. This bit is set and cleared by software. 0: 1 start bit, 8 data bits, n stop bits 1: 1 start bit, 9 data bits, n stop bits Caution: The M bit must not be changed during data transfer (both transmission and reception).	0
11	WAKE	Wakeup method: This bit determines the USART wakeup method. This bit is set and cleared by software. 0: Idle line 1: Address mark	0
10	PCE	Parity control enable: This bit selects hardware parity control (generation and detection). When parity control is enabled, the calculated parity bit is inserted into the MSB position (if M=1, the 9th bit; if M=0, the 8th bit), and the received data is checked for parity. This bit is set and cleared by software. Once this bit is set to 1, PCE is active after the current byte (during both reception and transmission). 0: Disable parity control 1: Enable parity control	0
9	PS	Parity selection: This bit is used to select odd or even parity when parity generation/detection (PCE bit set to 1) is enabled. This bit is set and cleared by software and selects odd or even parity for the current byte. 0: Even parity 1: Odd parity	0
8	PEIE	PE interrupt enable: (This bit is set and cleared by software.) 0: Disable interrupt 1: Generate USART interrupt when PE=1 in USART_SR register	0
7	TXEIE	TXE interrupt enable: (This bit is set and cleared by software) 0: Disable interrupt 1: Generate USART interrupt when TXE=1 in USART_SR register	0
6	TCIE	Transmission complete interrupt enable: This bit is set and cleared by software. 0: Disable interrupt 1: Generate USART interrupt when TC=1 in USART_SR register	0
5	RXNEIE	RXNE interrupt enable: This bit is set and cleared by software. 0: Disable interrupt	0

		1: Generate USART interrupt when ORE=1 or RXNE=1 in USART_SR register	
4	IDLEIE	IDLE interrupt enable: This bit is set and cleared by software. 0: Disable interrupt 1: Generate USART interrupt when IDLE=1 in USART_SR register	0
3	TE	Transmitter enable: This bit enables the transmitter. This bit is set and cleared by software. 0: Disable transmitter 1: Enable transmitter Caution: 1: During transmission, a “0” pulse on the TE bit (followed immediately by a “1”) will transmit a header (idle line) after the current word. 2: When TE is set to 1, there is a 1-bit time delay before transmission begins.	0
2	RE	Receiver enable: This bit enables the receiver. This bit is set and cleared by software. 0: Disable receiver 1: Enable receiver and begin searching for start bit	0
1	RWU	Receiver wakeup: This bit determines whether the USART is in mute mode. This bit is set and cleared by software and can be cleared by hardware when a wakeup sequence is detected. 0: Receiver is in active mode 1: Receiver is in mute mode Caution: 1: Before selecting mute mode (by setting RWU bit to 1), the USART must first receive a data word. Otherwise, it will not operate correctly in mute mode when wakeup is detected by the idle line. 2: In address mark detection wakeup configuration (WAKE bit = 1), RWU bit cannot be modified by software when RXNE bit is set to 1.	0
			0
0	SBK	Send break: This bit is used to send a break character. This bit can be set and cleared by software. It should be set by software and reset by hardware during the break stop bit period. 0: Do not send break character 1: Send break character	0

15.6.5 USART Control Register 2 (USART_CR2)

Bit	Symbol	Description	Reset value
31:15	--	Reserved	--
14	LINEN	LIN mode enable: This bit is set and cleared by software. 0: Disable LIN mode 1: Enable LIN mode LIN mode allows the use of the SBK bit in the USART_CR1 register to send LIN synchronization break (13 low bits) and detect the LIN synchronization break.	0
13:12	STOP	STOP bit: These bits are used for programming the stop bits. 00: 1 stop bit 01: Disable setting 10: 2 stop bits 11: Disable setting	0
11	CLKEN	Clock enable: This bit allows the user to enable the SCLK pin. 0: Disable SCLK pin 1: Enable SCLK pin	0
10	CPOL	Clock polarity: This bit allows the user to select the polarity of the clock output on the SCLK pin in synchronous mode. It works in conjunction with the CPHA bit to achieve the desired clock/data relationship. 0: SCLK pin is low when idle 1: SCLK pin is high when idle	0
9	CPHA	This bit allows the user to select the phase of the clock output on the SCLK pin in synchronous mode. It works in conjunction with the CPOL bit to achieve the desired clock/data relationship (refer to Figures 258 to 259). 0: Capture data on the first clock transition edge 1: Capture data on the second clock transition edge	0
8	LBCL	Last bit clock pulse This bit allows the user to choose whether the clock pulse associated with the last transmitted data bit (MSB) must be output on the SCLK pin in synchronous mode. 0: No clock pulse for the last data bit on the SCLK pin 1: Clock pulse for the last data bit on the SCLK pin Caution: 1: The last bit is the 8th or 9th data bit sent, depending on the M bit selected for 8-bit or 9-bit format in the USART_CR1 register. 2: This bit is not applicable for UART4 and UART5.	0
7	--	Reserved	--
6	LBDIE	LIN break detection interrupt enable: Break interrupt mask (used with break delimiter for break detection) 0: Disable interrupt 1: Generate interrupt when LBD = 1 in the USART_SR register	0
5	LBDL	LIN break detection length: This bit is used to select either 11-bit break detection or 10-bit break detection. 0: 10-bit break detection 1: 11-bit break detection	0

4	--	Reserved	--
3:0	ADD[3:0]	USART node address: This bit field is used to specify the USART node address. It will be used in multiprocessor communication in mute mode to wake up via address mark detection.	0x0

Caution: When enabling the transmitter, do not write to these 3 bits (CPOL, CPHA, LBCL).

15.6.6 USART Control Register 3 (USART_CR3)

Bit	Symbol	Description	Reset value
31:12	--	Reserved	--
11	ONEBIT	One sample bit method enable: This bit allows the user to choose the sampling method. When the one-sample bit method is selected, the noise flag (NF) will be disabled. 0: Three sampling bit methods 1: One sample bit method (not allowed for use)	0
10	CTSIE	CTS interrupt enable: 0: Disable interrupt 1: Generate an interrupt when CTS = 1 in the USART_SR register.	0
9	CTSE	CTS enable: 0: Disable CTS hardware flow control 1: Enable CTS mode; data is only transmitted when the nCTS input is valid (connected to 0). If nCTS input becomes invalid during data transmission, the transmission will complete before stopping. If nCTS is valid and data has been written to the data register, transmission will be delayed until nCTS is valid.	0
8	RTSE	RTS enable: 0: Disable RTS hardware flow control 1: Enable RTS interrupt; data will be requested only when there is space in the receive buffer. Data transmission will stop after sending the current character. nRTS output will be valid (connected to 0) when data can be received.	0
7	DMAT	DMA enable transmitter: This bit is set/reset by software. 0: Disable DMA mode for transmission 1: Enable DMA mode for transmission	0
6	DMAR	DMA enable receiver: This bit is set/reset by software. 0: Disable DMA mode for reception 1: Enable DMA mode for reception	0
5:4		Reserved	0
3	HDSEL	Half-duplex selection: Select single-wire half-duplex mode. 0: Half-duplex mode not selected 1: Half-duplex mode selected	0

2	IRLP	IrDA low-power: This bit selects between normal mode and low-power IrDA mode. 0: Normal mode 1: Low-power mode	0
1	IREN	IrDA mode enable: 0: Disable IrDA 1: Enable IrDA	0
0	EIE	Error interrupt enable: For multi-buffer communication (when DMAR = 1 in USART_CR3 register), if a frame error, overrun error, or noise flag (FE = 1 or ORE = 1 or NF = 1 in USART_SR register) occurs, the error interrupt enable bit must be used to enable the interrupt generation. 0: Disable interrupt 1: Generate interrupt when DMAR = 1 in USART_CR3 register and FE = 1, ORE = 1, or NF = 1 in USART_SR register.	0

15.6.7 USART Guard Time and Prescaler Register (USART_GTPR)

Bit	Symbol	Description	Reset value
31:8	--	Reserved	--
7:0	PSC[7:0]	Prescaler values: —In IrDA low-power mode: PSC[7:0] = IrDA low-power baud rate Used for programming the prescaler to divide the system clock to obtain a low-power frequency: Divide the source clock using the value given in the register (8 valid bits): 00000000: Reserved - Do not program this value Source clock divided by 1 00000010: Source clock divided by 2 ... —In normal IrDA mode: PSC must be set to 00000001.	0x0

Chapter 16 Enhanced DMA

16.1 Functions of DMA

DMA is a feature that allows data transfer between memory units without using the CPU. DMA transfers are initiated through peripheral interrupts. When DMA and the CPU access the same unit in FLASH, SRAM, or peripheral modules simultaneously, DMA has higher priority for bus access. However, if DMA and the CPU access different units in FLASH, SRAM, or peripheral modules, both can execute in parallel without interference.

The specifications for DMA are shown in Table 16-1.

Table 16-1 Specifications for DMA (1/2)

Item		Specification
Boot source		Up to 3 boot sources
Assignable control data		3 groups
The address space that can be transferred	Address space	Full address range space
	Source	Full address range space is optional
	Target	Full address range space is optional
Maximum number of transfers	Normal mode	65535 times
	Repeat mode	65535 times
The maximum transfer block size	Normal mode (8-bit transfer)	65535 bytes
	Normal mode (16-bit transfer)	131070 bytes
	Normal mode (32-bit transfer)	262140 bytes
	Repeat mode	65535 bytes
Transfer unit		8-bit/16-bit/32-bit
Transfer mode	Normal mode	Ends after a transfer of the DMACTj register from 1 to 0.
	Repeat mode	After the transfer of the DMCTj register from 1 to 0 is completed, the address of the repeat area is initialized, and the DMRLDj is changed. The value of the register continues after reloading into the DMCTj register.
Address control	Normal mode	Fixed or incremented
	Repeat mode	Fixed or incremented the address of the non-repeating area.
Priority of the start-up sources		Refer to Table 16-4.

Table 16-1 Specifications for DMA (2/2)

Item		Specification
Interrupt requests	Normal mode	When a data transfer with a DMCTj register changed from 1 to 0, an interrupt of the start-up source is requested to the CPU and interrupt processing is performed.
	Repeat pattern	When the RPTINT bit of the DMACRj register is 1 (enable an interrupt to be generated) and the data transfer of the DMACTj register changes from 1 to 0, an interrupt from the startup source is requested to the CPU and interrupt processing is performed.
Transfer start		If the DMAENi register's DMAENi0~DMAENi1, DMAENi3 bits are set to "1" (allowing initiation), data transfer will begin each time a DMA trigger source occurs.
Transfer stop	Normal mode	- Set DMAENi0~DMAENi1, DMAENi3 to "0"(disable startup). - At the end of the data transfer when the DMACTj register changes from 1 to 0.
	Repeat mode	- Set DMAENi0~DMAENi1, DMAENi3 to "0"(disable startup). - At the end of data transfer when the RPTINT bit is 1(interrupt generation is allowed) and the DMACTj register changes from 1 to 0.

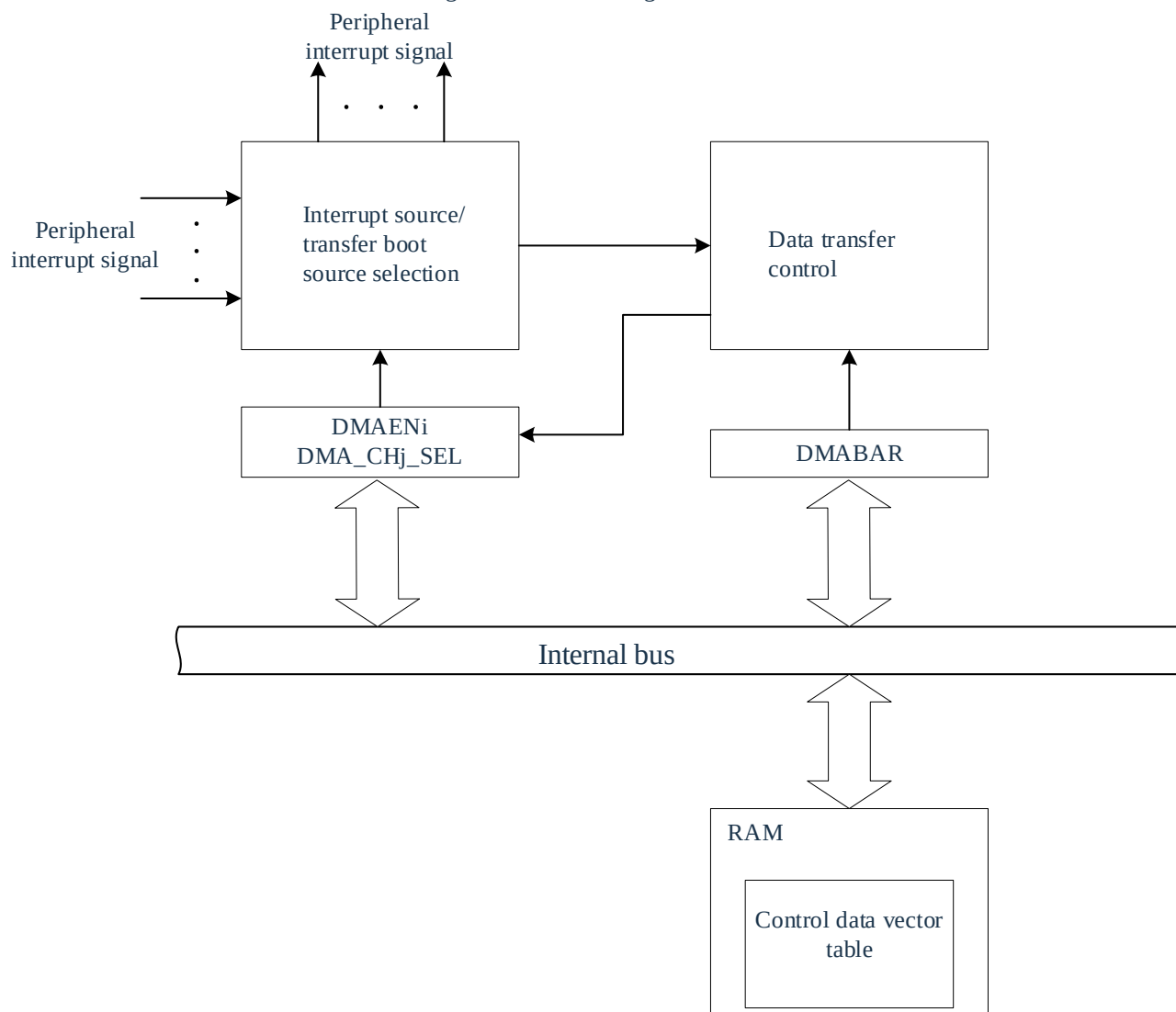
Note 1: In deep sleep mode, the flash memory cannot be used as a DMA transfer source because it is stopped.

Note 2: i=0, j=0~1, 3.

16.2 Structure of DMA

The block diagram for DMA is shown in Figure 16-1.

Figure 16-1 Block diagram of DMA



16.3 Register Mapping

(Base address = 0x40020810) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
PER10	0x000	R/W	Peripheral Enable Register 10	0x00

(Base address = 0x40005000) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
DMAEN0	0x000	R/W	DMA Start Enable Register 0	0x00
DMABAR	0x008	R/W	DMA Base Address Register	0x00
DMASET0	0x018	WO	DMAEN0 Set Register	0x00
DMACLR0	0x020	WO	DMAEN0 Reset Register	0x00

(Base address = 0x40020D00) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
DMA_CH0_SEL	0x000	R/W	DMA Channel 0 Trigger Source Selection Register	0x3F
DMA_CH1_SEL	0x001	R/W	DMA Channel 1 Trigger Source Selection Register	0x3F
DMA_CH3_SEL	0x003	R/W	DMA Channel 3 Trigger Source Selection Register	0x3F

The DMA control data is shown in Table 16-2.

The DMA control data is allocated in the DMA control data section of the RAM. The DMA control data section and the 80-byte region containing the DMA vector table (which stores the starting address of the control data) are set through the DMABAR register.

Table 16-2 DMA control data

Register name	Symbol
DMA Control Register j	DMACRj
DMA Block Size Register j	DMBLSj
DMA Transfer Count Register j	DMACTj
DMA Transfer Count Reload Register j	DMRLDj
DMA Source Address Register j	DMSARj
DMA Target Address Register j	DMDARj

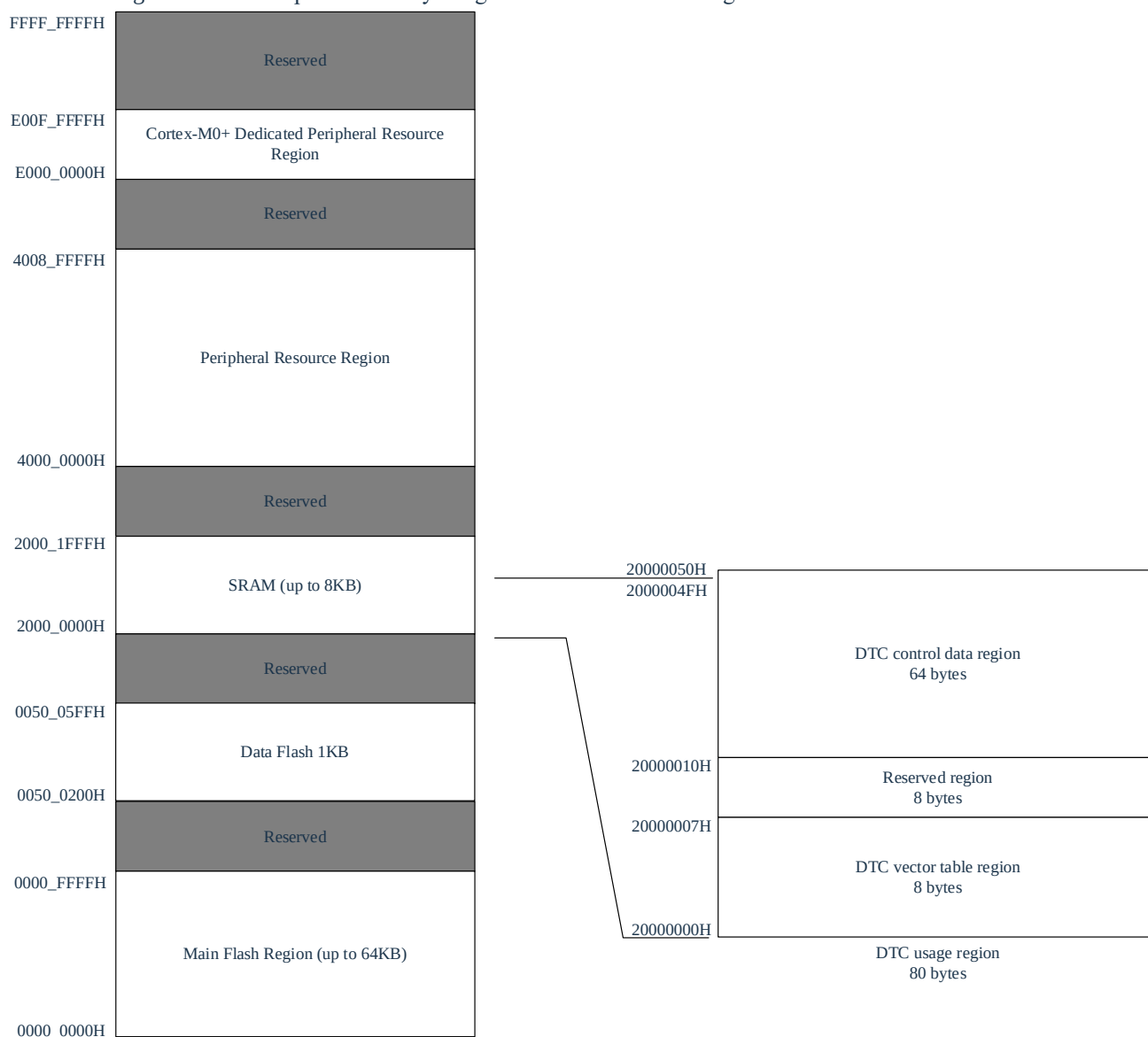
16.4 Registers for Controlling DMA

16.4.1 Allocation of DMA Control Data Region and DMA Vector Table Region

The 80-byte area of the DMA control data and vector table is allocated to the RAM region via the DMABAR register.

An example of the memory image when the DMABAR register is set to “20000000H” is shown in Figure 16-2. The unused space in the 80 bytes of the MA control data region not used by DMA can be used as RAM.

Figure 16-2: Example of memory image when the DMABAR register is set to “20000000H”



16.4.2 Controlling Data Allocation

From the starting address, control data is allocated in the order of DMACRj, DMBLSj, DMACTj, DMRLDj, DMSARj, DMDARj (j=0~1,3) registers.

The starting address is set by the DMABAR register, with the low 8 bits being set by each trigger source's respective vector table.

Each channel's trigger source is set by the DMA_CHj_SEL (j=0~1,3) register.

Control data allocation is shown in Figure 16-3.

Note 1: The DMACRj, DMBLSj, DMACTj, DMRLDj, DMSARj, DMDARj registers' data must be changed when the corresponding DMAENi(i=0) DMAENi0~DMAENi2, DMAENi3 bits are set to 0 (disable startup).

Note 2: DMACRj, DMBLSj, DMACTj, DMRLDj, DMSARj, and DMDARj cannot be accessed through DMA transfer.

Figure 16-3 Controlling data allocation (DMABAR=20000000H)

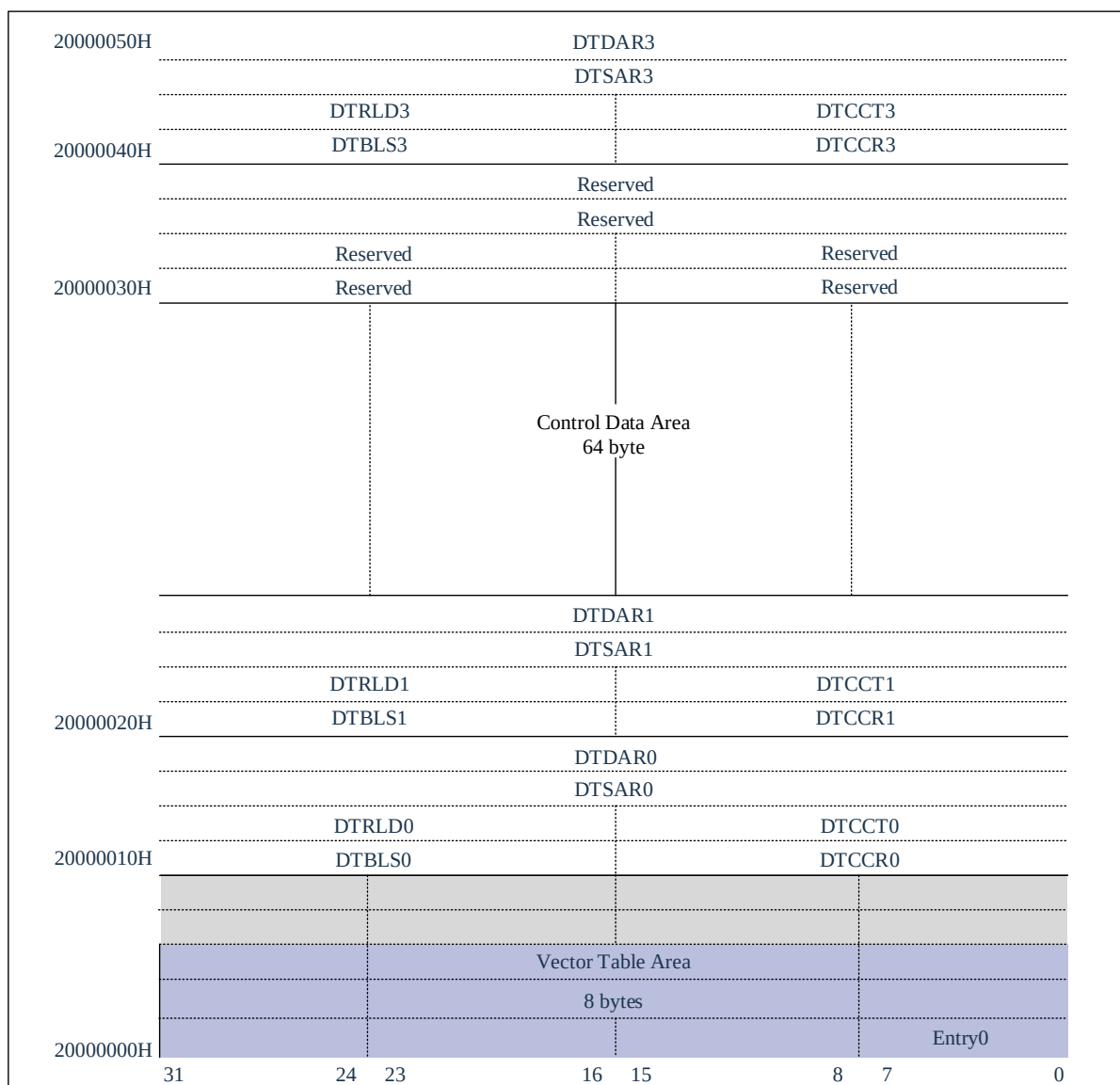


Table 16-3 Start addresses of control data

j	Address
3	baseaddr+40H
-	-
1	baseaddr+20H
0	baseaddr+10H

Note: baseaddr: Setting values of the DMABAR register.

16.4.3 Vector Table

Once DMA is started, the control data is determined by reading the data from the vector table allocated for each trigger source. The control data is read from the DMA control data section.

The DMA trigger sources and vector addresses are shown in Table 16-4. Each trigger source's vector table is 1 byte, storing the data “00H” ~ “02H”, “03H”, and selects one set of control data from the three sets of control data. The high 24 bits of the vector address are set by the DMABAR register, while the low 8 bits are allocated to the corresponding trigger source as “00H” ~ “02H”, “03H”.

Note: The start address of the DMA control data area set in the vector table must be changed when the DMAENi0 to DMAENi7 bits of the corresponding DMAENi (i=0~4) registers are 0 (startup disabled).

Figure 16-4: Starting address and vector table for control data

A case where the DMABAR register is set to “20000000H”

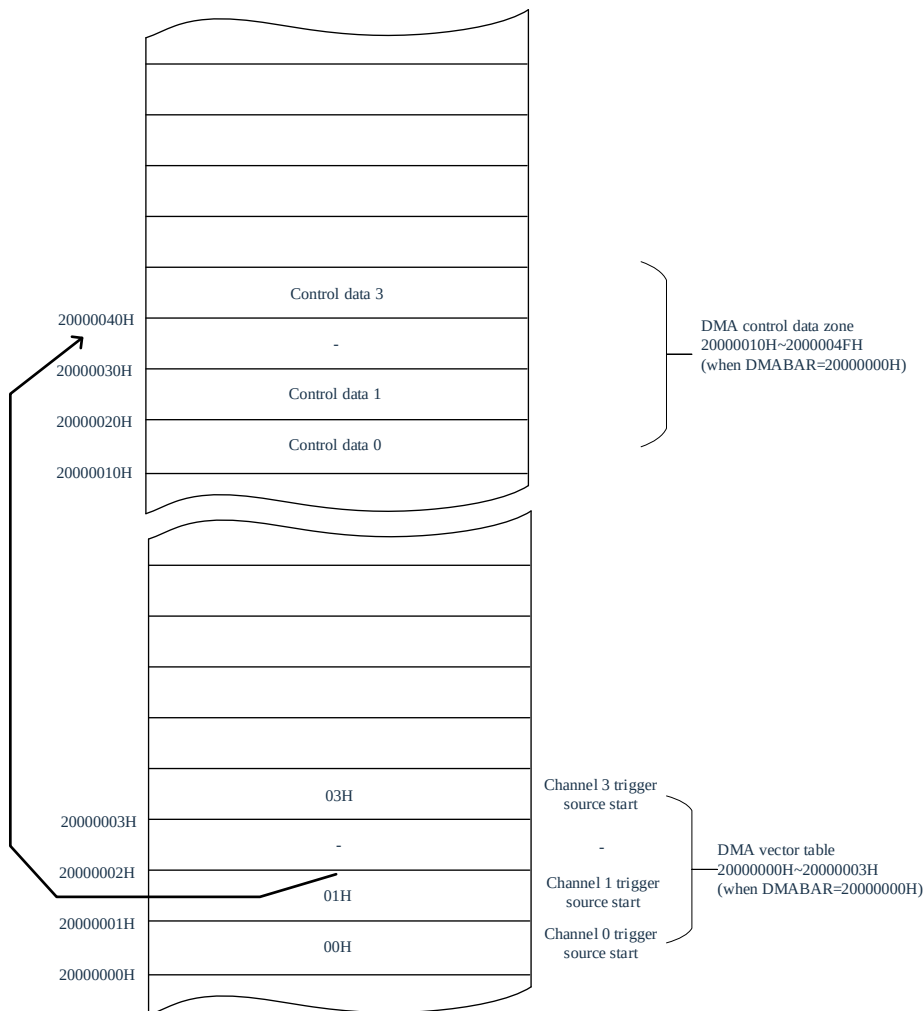


Table 16-4 DMA startup sources and vector addresses

DMA boot source (interrupt request generation source)	Source number	Vector address	Priority
Channel 0 trigger source start	0	DMABAR register set address +00H	High ↑ ↓ Low
Channel 1 trigger source start	1	DMABAR register set address +01H	
-	-	-	
Channel 3 trigger source start	3	DMABAR register set address +03H	

16.4.4 Peripheral Enable Register 10(PER10)

The PER10 register is a register that sets a clock that is allowed or prohibited to supply to each peripheral hardware. Reduce power consumption and noise by stopping clock supply to unused hardware. To use DMA, the bit3 (DMAEN) must be set to 1.

The PER10 register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register changes to “00H”.

Peripheral enable register10 (PER10)

Bit	Symbol	Description	Reset value
7:4	-	Reserved	0
3	DMAEN	Control of DMA input clock supply 0: Stop to supply the input clock, DMA cannot run. 1: Supply the input clock, DMA can run.	0
2	CRCEN	Control of CRC input clock supply 0: Stop to supply the input clock, CRC cannot run. 1: Supply the input clock, CRC can run.	0
1:0	-	Reserved	0

16.4.5 DMA Channel Trigger Source Selection Register j (DMA_CHj_SEL) (j=0~1, 3)

Bit	Symbol	Description	Reset value
7:6	-	Reserved	0x0
5:0	TRGSEL	DMA channel j selects trigger sources (j=0~3): 0x00: ADC 0x01: SPI1_TX 0x02: SPI1_RX 0x03: SAU module UART transmission, SPI/IIC transmit/receive end 0x04: SAU module UART receive end 0x05: USART_TX 0x06: USART_RX 0x07: Reserved 0x08: Reserved 0x09: I2C_TX 0x0A: I2C_RX 0x0B: TIM1_CH1 0x0C: TIM1_CH2 0x0D: TIM1_CH3 0x0E: TIM1_CH4 0x0F: TIM1_COM 0x10: TIM1_UP 0x11: TIM1_TRIG 0x12: TIM3_CH1 0x13: TIM3_CH3 0x14: TIM3_CH4 0x15: TIM3_TRG 0x16: TIM3_UP 0x17: Reserved 0x18: TIM16_CH1 0x19: TIM16_UP 0x1A: TIM17_CH1 0x1B: TIM17_UP 0x1C: TIM3_CH2 Others: Disable setting	0x3F

16.4.6 DMA Control Register j (DMACRj) (j=0~1, 3)

The DMACRj register controls the mode of operation of the DMA.

Address: refer to 16.4.2 After reset: undefined R/W

Bit	Symbol	Description	Reset value
15:8	-	Reserved	Undefined
7:6	SZ	Transfer data length selection 00: 8 bits 01: 16 bits 10: 32 bits 11: Disable setting	
5	RPTINT	Enable/disable repeat mode interrupt ^{Note 1} 0: Interrupts are disabled 1: Interrupts are enabled	
4	CHNE	Chain transfer enabled/disabled ^{Note 2} 0: Chain transfer is disabled. 1: Chain transfer is enabled.	
3	DAMOD	Control of the transfer target address ^{Note 3} 0: Fixed 1: Incremental	
2	SAMOD	Control of transfer source address ^{Note 4} 0: Fixed 1: Incremental	
1	RPTSEL	Selection of repeat region ^{Note 5} 0: The transfer target is a repeat region. 1: The transfer source is a repeat region.	
0	MODE	Selection of transfer modes 0: Normal mode 1: Repeat mode	

Note 1: In normal mode (MODE bit is 0), the setting of the RPTINT bit is invalid.

Note 2: The CHNE bit in the DMACR7 register must be set to 0 (disable chained transfer).

Note 3: In repeat mode (MODE bit is 1) and when RPTSEL bit is 0 (transfer target is the repeat region), the setting of the DAMOD bit is invalid.

Note 4: In repeat mode (MODE bit is 1) and when RPTSEL bit is 1 (transfer source is the repeat region), the setting of the SAMOD bit is invalid.

Note 5: In normal mode (MODE bit is 0), the setting of the RPTSEL bit is invalid.

Note 6: DMACRj register cannot be accessed through DMA transfer.

16.4.7 DMA Block Size Register j (DMBLSj) (j=0~1, 3)

This register sets the block size for 1-time startup transfer data.

Address: refer to 16.4.2 After reset: undefined R/W

Bit	Symbol	Description	Reset value
15:0	DMBLSj	Set 1-time start transfer data block size	Undefined

DMBLSj	Transfer block size		
	8-bit transfer	16-bit transfer	32-bit transfer
00H	Settings are disabled	Settings are disabled	Settings are disabled
01H	1 byte	2 bytes	4 bytes
02H	2 bytes	4 bytes	8 bytes
03H	3 bytes	6 bytes	12 bytes
⋮	⋮	⋮	⋮
FDH	253 bytes	506 bytes	1012 bytes
FEH	254 bytes	508 bytes	1016 bytes
FFH	255 bytes	510 bytes	1020 bytes
⋮	⋮	⋮	⋮
FFFFH	65535 bytes	131070 bytes	262140 bytes

Note 1: Access to the DMBLSj register cannot be performed via DMA transfer.

16.4.8 DMA Transfer Count Register j (DMACTj) (j=0~1, 3)

This register sets the number of DMA data transfers. It is decremented by 1 whenever 1 DMA transfer is initiated.

Address: refer to 16.4.2 After reset: undefined R/W

Bit	Symbol	Description	Reset value
15:0	DMRLDj	Set the transfer count register initial value in repeat mode	Undefined

DMACTj	Transfer count
00H	Settings are disabled
01H	1 time
02H	2 times
03H	3 times
•	•
•	•
•	•
FDH	253 times
FEH	254 times
FFH	255 times
•	•
•	•
•	•
FFFFH	65535 times

Note 1: Access to the DMACTj register cannot be performed via DMA transfer.

16.4.9 DMA Transfer Count Reload Register j (DMRLDj) (j=0~1, 3)

This register sets the initial value of the transfer number register in repeat mode. In Repeat mode, since the value of this register is reloaded into the DMACT register, the set value must be the same as the initial value of the DMACT register.

Address: refer to 16.4.2 After reset: undefined R/W

Bit	Symbol	Description	Reset value
15:0	DMRLDj	Set the transfer count register initial value in repeat mode	Undefined

Note: Access to the DMRLDj register cannot be performed via DMA transfer.

16.4.10 DMA Source Address Register j (DMSARj) (j=0~1, 3)

This register specifies the transfer source address when the data is transferred.

When the SZ bit of the DMACRj register is 01 (16-bit transfer), the lowest bit is ignored and processed as an even address.

When the SZ bit of the DMACRj register is 10 (32-bit transfer), the low 2 bits are ignored and processed as word addresses.

Address: refer to 16.4.2 After reset: undefined R/W

Bit	Symbol	Description	Reset value
31:0	DMSARj	Set the transfer source address during data transfer	Undefined

Note: Access to the DMSARj register cannot be performed via DMA transfer.

16.4.11 DMA Target Address Register j (DMDARj) (j=0~1, 3)

This register specifies the target address of the transfer when the data is transferred.

When the SZ bit of the DMACRj register is 01 (16-bit transfer), the lowest bit is ignored and processed as an even address.

When the SZ bit of the DMACRj register is 10 (32-bit transfer), the low 2 bits are ignored and processed as word addresses.

Address: refer to 16.4.2 After reset: undefined R/W

Bit	Symbol	Description	Reset value
31:0	DMDARj	Set the target address during data transfer	Undefined

Note: Access to the DMDARj register cannot be made via DMA transfer.

16.4.12 DMA Start Enable Register i(DMAEN0)

This is an 8-bit register that controls whether or not starting DMA through each interrupt source. The correspondence between the channels and the DMAEN0 bits is shown in Table 16-5.

The DMAEN register can be set by an 8-bit memory manipulation instruction.

Note 1: The DMAENi0 to DMAENi3 bits must be changed at a location that does not generate a startup source corresponding to this bit.

Note 2: Access to the DMAENi register cannot be performed via DMA transfer.

Note 3: The assigned function varies from product to product and must be set to 0 if no function is assigned.

Bit	Symbol	Description	Reset value
31:4	--	Reserved, set to 0.	0x0
3:0	DMAEN0n (n=0~3)	DMAin start enable ^{Note 1} (n=0~3) 0: Start disabled 1: Start enabled	0x0

Note 1: The DMAENin bit changes to 0 (start disabled) according to the condition of the transfer end interrupt. n=0,1,2,3.

Note 2: DMAEN02=0.

Table 16-5 Correspondence of channels and DMAEN00~DMAEN03 bits

Register	DMAENi3 bit	DMAENi2 bit	DMAENi1 bit	DMAENi0 bit
DMAEN0	Channel 3 start enable	-	Channel 1 start enable	Channel 0 start enable

Note 1: Bits that are not assigned a function must be set to 0.

Note 2: i=0

16.4.13 DMA Base Address Register (DMABAR)

This is a 32-bit register that sets the vector address that holds the starting address of the DMA control data area and the address of the DMA control data region.

Note 1: The DMABAR register must be changed with all DMA startup sources set to disable startup.

Note 2: The DMABAR register can be rewritten only 1 time.

Note 3: Access to the DMABAR register cannot be performed via DMA transfer.

Note 4: Refer to Table 16-4-1 for the DMA control data area and DMA vector table area assignment.

Note 5: To set this register, keep it 256 Byte aligned, that is, the lower 8 bits are set to zero. The DMA hardware ignores the lower 8 bits.

Note 6: The register can only be accessed by WORD, BYTE and HALWORD access are ignored.

Bit	Symbol	Description	Reset value
31:8	DMABARj	Set the target address high 24 bits for data transfer	0x0
7:0	DMABARj	Set the target address low 8 bits for data transfer. Due to 256-byte alignment, the low 8 bits are set to 0.	0x0

Note: j=0.

16.4.14 DMAEN0 Set Register (DMASET0)

This is the set register for the DMA Enable Register DMAEN0. Setting the corresponding bit to 1 will set the corresponding bit of DMAEN0 to 1.

Bit	Symbol	Description	Reset value
31:4	--	Reserved, set to 0.	0x0
3:0	DMASET0n (n=0~3)	DMAEN0n set register (n=0~3) 0: No set action 1: Set DMAEN0 bit n to 1	0x0

Note: DMASET02=0.

16.4.15 DMAEN0 Reset Register (DMACLR0)

This is the reset register for the DMA Enable Register DMAEN0. Setting the corresponding bit to 1 will reset the corresponding bit of DMAEN0 to 0.

Bit	Symbol	Description	Reset value
31:4	--	Reserved, set to 0.	0x0
3:0	DMACLR0n (n=0~3)	DMAEN0n reset register (n=0~3) 0: No set action 1: Reset DMAEN0 bit n to 0	0x0

Note: DMACLR02=0.

16.4.16 Operation of DMA

Once the DMA is started, the control data is read from the DMA control data area, data transfer is performed based on the control data, and the control data transferred is written back. It is possible to store 3 sets of control data to a DMA control data area and to perform transfer of 3 sets of data. The transfer mode has a normal mode and a repeat mode, and the transfer size has 8-bit transfer, 16-bit transfer and 32-bit transfer. When the CHNE bit of the DMACRj (j=0~1, 3) register is 1 (allowing chain transfer), the continuous data transfer (chain transfer) is read by a start source.

The transfer source address and the transfer target address are specified through the 32-bit DMSARj register and the 32-bit DMDARj register, respectively. After the data is transferred, the values of the DMSARj register and the DMDARj register are incremented or fixed according to the control data.

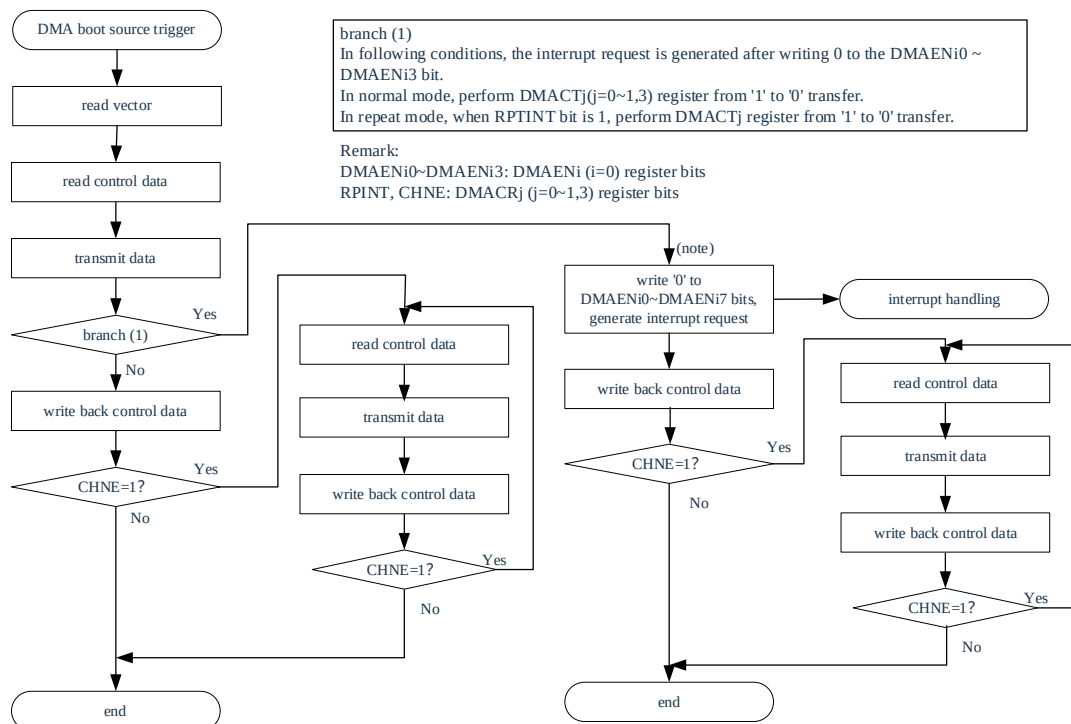
16.4.17 Boot Source

DMA can be started through an interrupt signal from a peripheral function, and the DMAENi (i=0) register is used to select the interrupt signal for starting DMA. When the data transfer (during chained transfer, where the initial transfer is continuously conducted) is set to one of the following two conditions, the corresponding DMAENi register's DMAENi0~DMAENi3 bits are set to 0 (disable start) during DMA operation:

- In normal mode: When the DMACTj (j=0~1, 3) register is set to 0 for transfer.
- In repeat mode: When the RPTINT bit of the DMACRj register is set to 1 (interrupt allowed) and the DMACTj register is set to 0 for transfer.

The internal operation flow of DMA is shown in Figure 16-5.

Figure 16-5 Internal operation flowchart of DMA



Note: In a data transfer initiated by a setting enable chain transfer (CHNE=1), the DMAENi0~DMAENi3 bits are not written to 0 and no interrupt request is generated.

16.4.18 Normal Mode

At 8-bit transfer, the transfer data of 1 boot is 1~65535 bytes. In 16-bit transfer, the transfer data of 1 boot is 2~131070 bytes; At 32-bit transfer, the transfer data of 1 boot is 4 to 262140 bytes. The number of transfers is 1 to 65535. If a data transfer in which the DMACTj (j=0~39) register becomes 0 is performed, an interrupt request corresponding to the boot source is generated to the interrupt controller during DMA operation, and the DMAENi0~DMAENi3 bits of the corresponding DMAENi (i=0) register are set to 0 (boot disabled).

The register functions and data transfers for normal mode are shown in Table 16-6 and Figure 16-6, respectively.

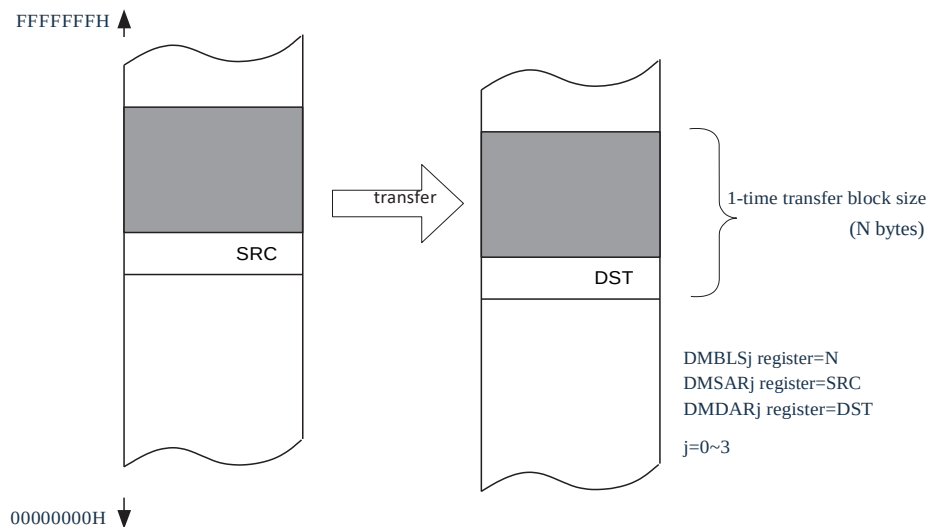
Note: j=0

Table 16-6 Register functions in normal mode

Register name	Symbol	Function
DMA block size register j	DMBLSj	1 boot data block size to be transferred
DMA transfer count register j	DMACTj	Number of data transfers
DMA transfer count reload register j	DMRLDj	Not used ^{Note}
DMA source address register j	DMSARj	Data transfer source address
DMA target address register j	DMDARj	Data transfer target address

Note: j=0

Figure 16-6 Normal mode data transfer



Settings for the DMACR register				Control of the source address	Control of the target address	The source address after transmission	The target address after transmission
DAMOD	SAMOD	RPTSEL	MODE				
0	0	X	0	Fixed	Fixed	SRC	DST
0	1	X	0	Incremental	Fixed	SRC+N	DST
1	0	X	0	Fixed	Incremental	SRC	DST+N
1	1	X	0	Incremental	Incremental	SRC+N	DST+N

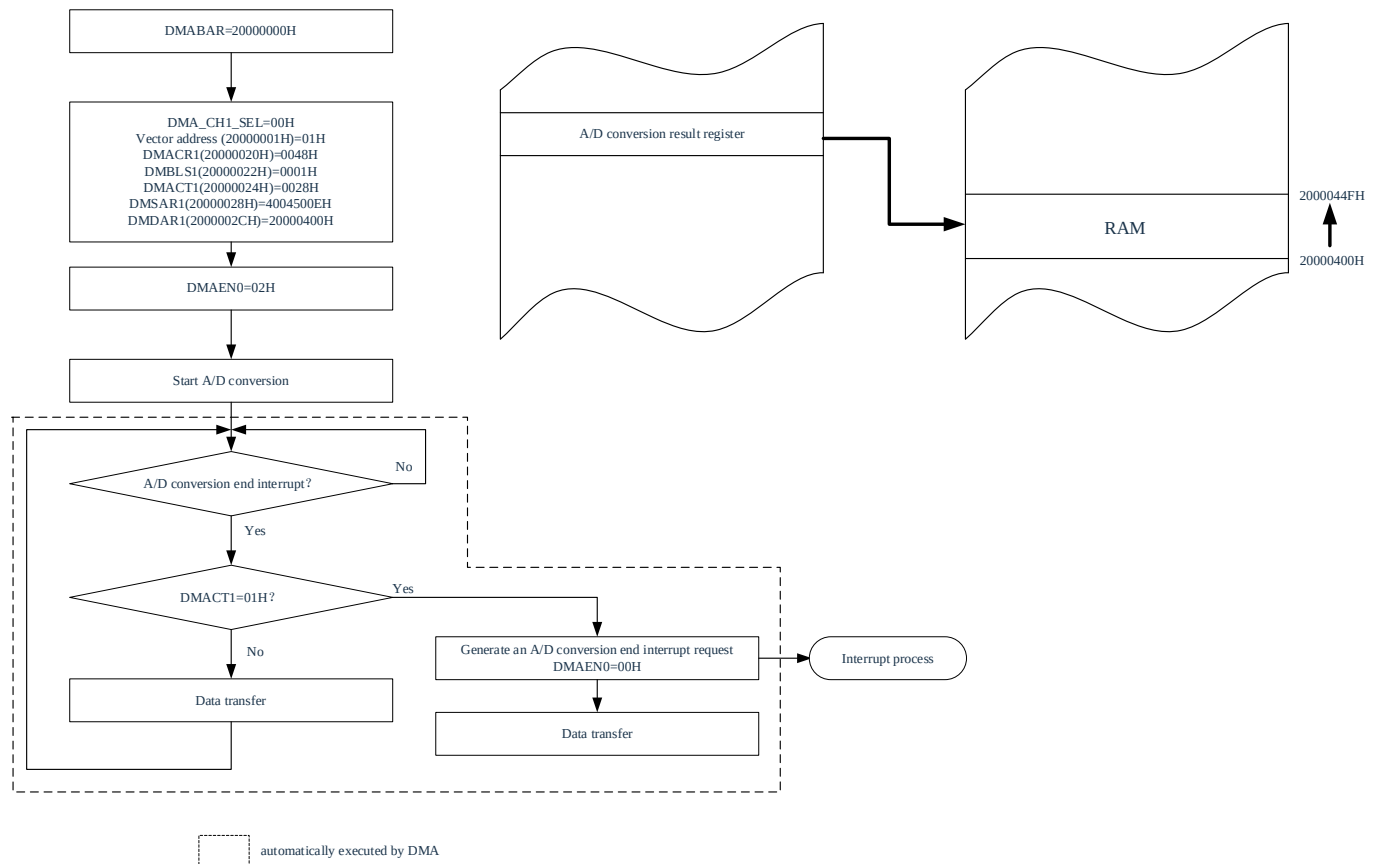
X: "0" or "1"

(1) Example 1 of normal mode usage: Channel 1 continuous A/D conversion results

The DMA is started by an A/D conversion end interrupt, and the value of the A/D conversion result register is transferred to the RAM.

- The vector address is assigned at 20000001H and the control data is assigned at 20000020H~20000002FH.
- Transfer 2 bytes of data from the A/D conversion result register (4004500EH) 40 times to 80 bytes of RAM 20000400H~2000044FH.

Figure 16-7 Example 1 of normal mode usage: Continuous A/D conversion results



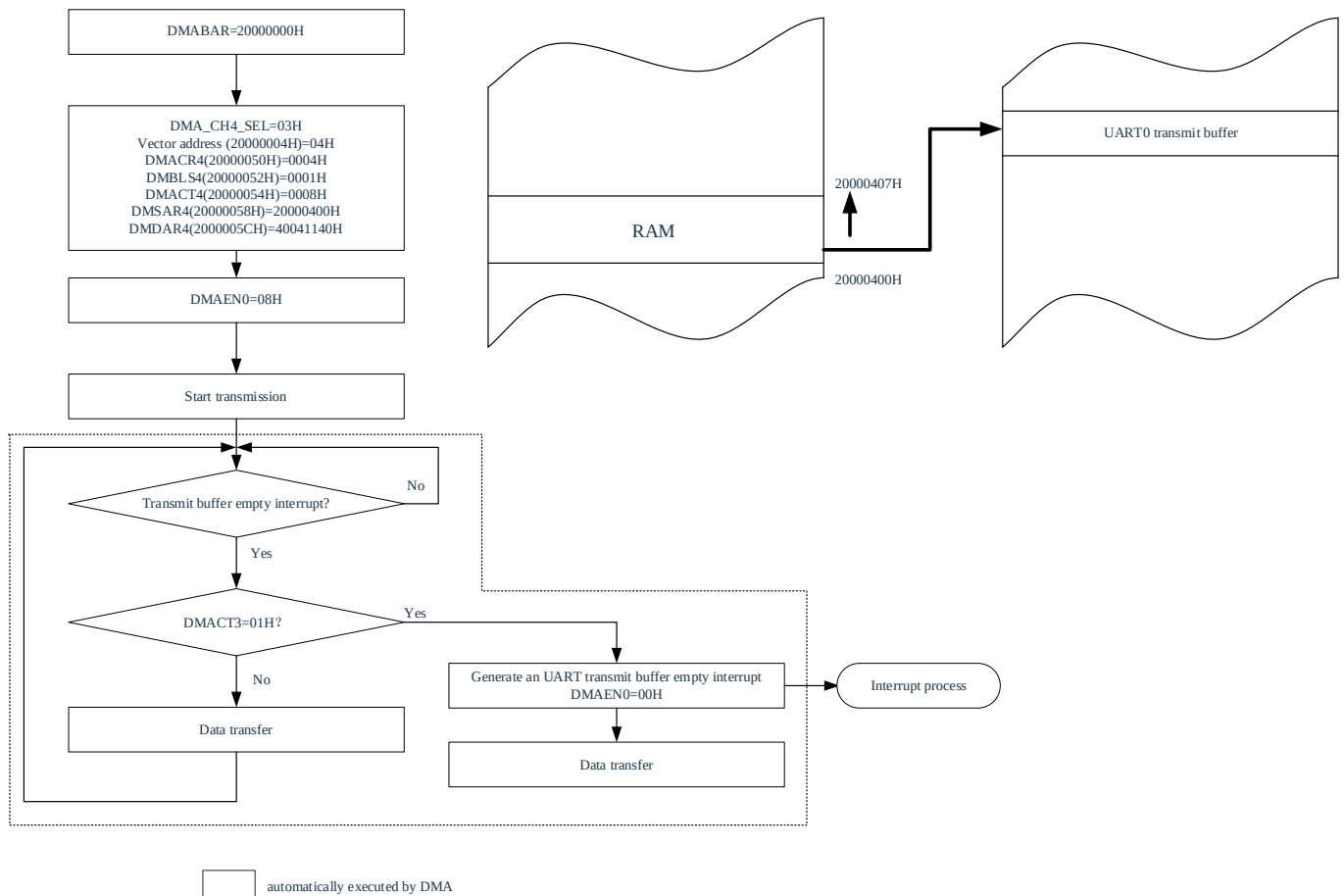
Since it is in normal mode, the value of the DMRLD1 register is not used.

(2) Example 2 of normal mode usage: Channel 3 UART0 continuous transmission

DMA is started by the UART0 transmit buffer empty interrupt, and the value from RAM is transferred to the UART0 transmit buffer.

- The vector address is allocated at 20000004H, and the control data is allocated from 20000050H to 2000005FH.
- 8 bytes from RAM (address range 20000400H~20000407H) are transferred to UART0 transmit buffer SDR00 (address 40041140H).

Figure 16-8 Example 2 of normal mode usage: UART0 continuous transmission



The first UART0 transmission must be initiated by software. DMA is started upon completion of the transmission, and then subsequent transmissions (from the second one onwards) are automatically triggered.

16.4.19 Repeat Mode

The transfer data for 1 boot is 1 to 65535 bytes. The transfer source or the transfer target is specified as a repeat area, and the number of transfers from 1 to 65535. Once the specified number of transfers is completed, the DMACTj (j=0 to 1, 3) register and the address specified as the repeat area are initialized, and the transfer is repeated. When the RPTINT bit in the DMACRj register is 1 (interrupt allowed) and a data transfer is performed with the DMACTj register turned to 0, an interrupt request corresponding to the start source is generated to the interrupt controller during DMA operation. And the DMAENi0 to DMAENi3 bits of the corresponding DMAENi (i=0) register are set to 0 (boot disabled). When the RPTINT bit of the DMACRj register is 0 (interrupt generation is prohibited), an interrupt request is not generated even if a data transfer in which the DMACTj register is changed to 0 is performed, and the DMAENi0 to DMAENi3 bits remain unchanged at 0.

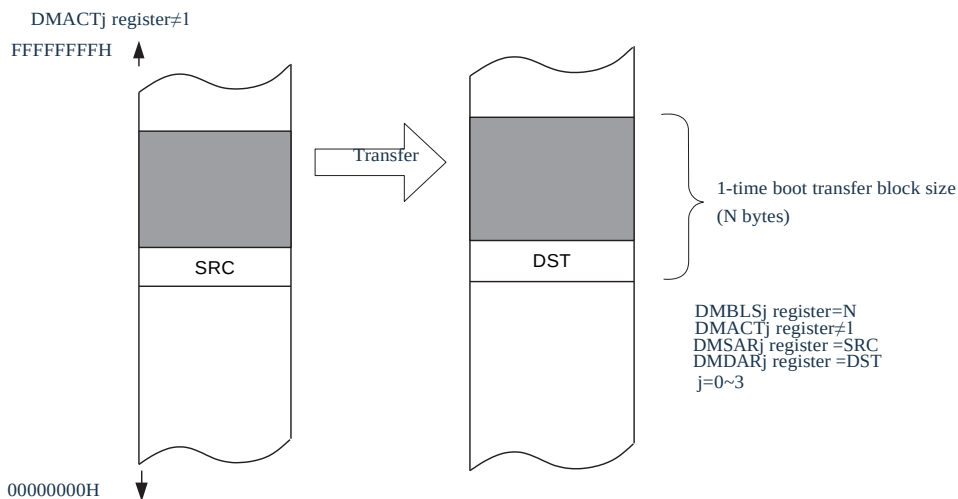
The register functions and data transfers for Repeat mode are shown in Table 16-7 and Figure 16-9, respectively.

Table 16-7 : Register functions for repeat mode

Register name	Symbol	Function
DMA block size register j	DMBLSj	The size of the data block to be transferred by 1 boot
DMA transmit count register j	DMACTj	The number of times the data was transferred
DMA transfer count reload register j	DMRLDj	Reload the value of this register to the DMACT register. (Initialize the number of data transfers)
DMA source address register j	DMSARj	The address of the source from which the data is transferred
DMA target address register j	DMDARj	Target address for data transfer

Note: j=0~1, 3

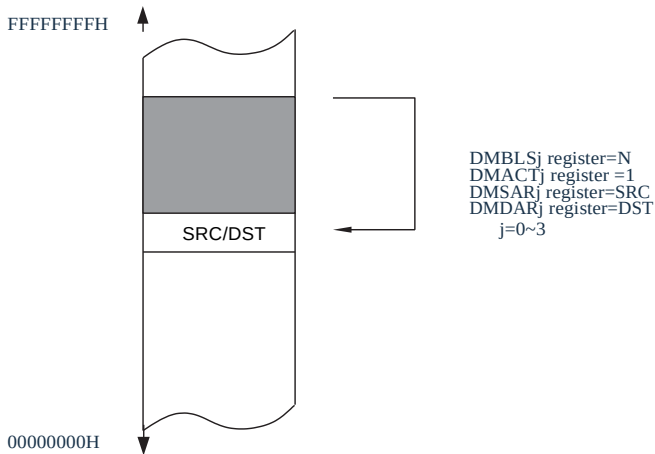
Figure 16-9 Repeat mode data transfer



DMACR register setting				Control of the source address	Control of the target address	The source address after transfer	The target address after transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	repeat area	fixed	SRC+N	DST
1	X	1	1	repeat area	incremental	SRC+N	DST+N
X	0	0	1	fixed	repeat area	SRC	DST+N
X	1	0	1	incremental	repeat area	SRC+N	DST+N

Remark: X: “0” or “1”

DMACTj register=1



DMACR register setting				Control of the source address	Control of the target address	The source address after transfer	The target address after transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	repeat area	fixed	SRC	DST
1	X	1	1	repeat area	incremental	SRC	DST+N
X	0	0	1	fixed	repeat area	SRC	DST
X	1	0	1	incremental	repeat area	SRC+N	DST

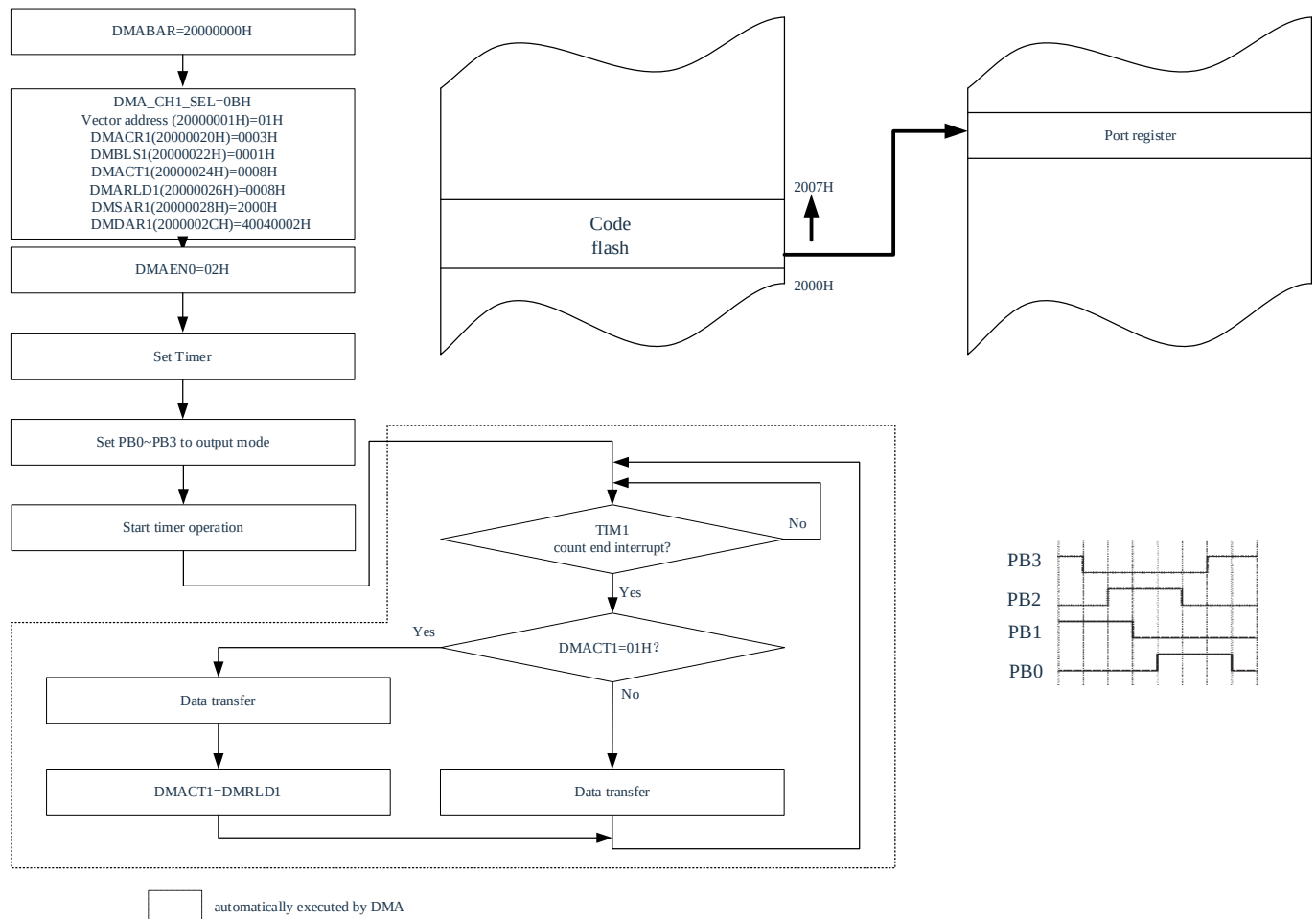
Remark: X: “0” or “1”

Note: In repeat mode, the data length of the repeat area must be set within 65535 bytes.

1) Example 1 of repeat mode usage: Use the stepper motor of the port to control the pulse output by channel 1
When the TIM1 Channel 1 count ends, DMA is started, and the motor control pulse mode stored in the code flash is transferred to the general-purpose port PB (address 0x40040002H).

- The vector address is assigned in 20000001H, and the control data is assigned at 20000020H~2000002FH.
- Transfer 8 bytes from 02000H~02007H of the code flash to port register 1(40040002H).
- Disable repeat mode interrupts.

Figure 16-10: Example 1 of repeat mode usage: use the stepper motor of the port to control the pulse output



To stop the output, the bit 1 of the DMAEN0 must be cleared after stopping the timer.

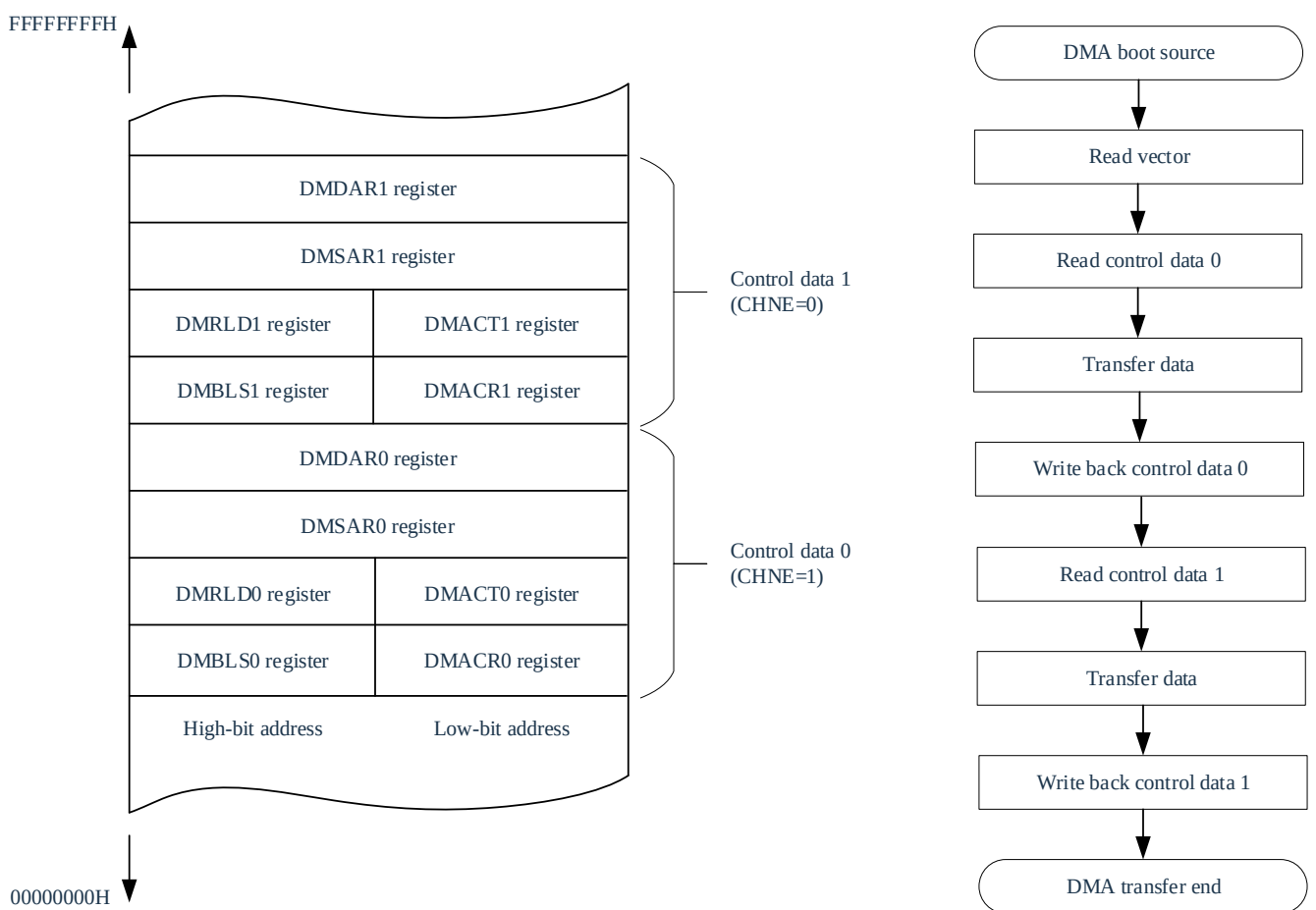
16.4.20 Chain Transfer

When the CHNE bit of the DMACRj (j=0~1, 3) register is 1 (chain transfer is enabled), multiple data transfers can be performed consecutively through 1 boot source.

Once the DMA is initiated, the control data assigned to the DMA control data area is read by selecting the control data by the data read from the vector address corresponding to the boot source. If the CHNE bit of the control data read is 1 (chain transfer enabled), the next allocated control data is read at the end of the transfer and the transfer continues. This operation is repeated until the transfer ends with control data where the CHNE bit is 0 (chained transfer disabled).

The flow chart for chain transfer is shown in Figure 16-11.

Figure 16-11: Flow chart for chain transfer



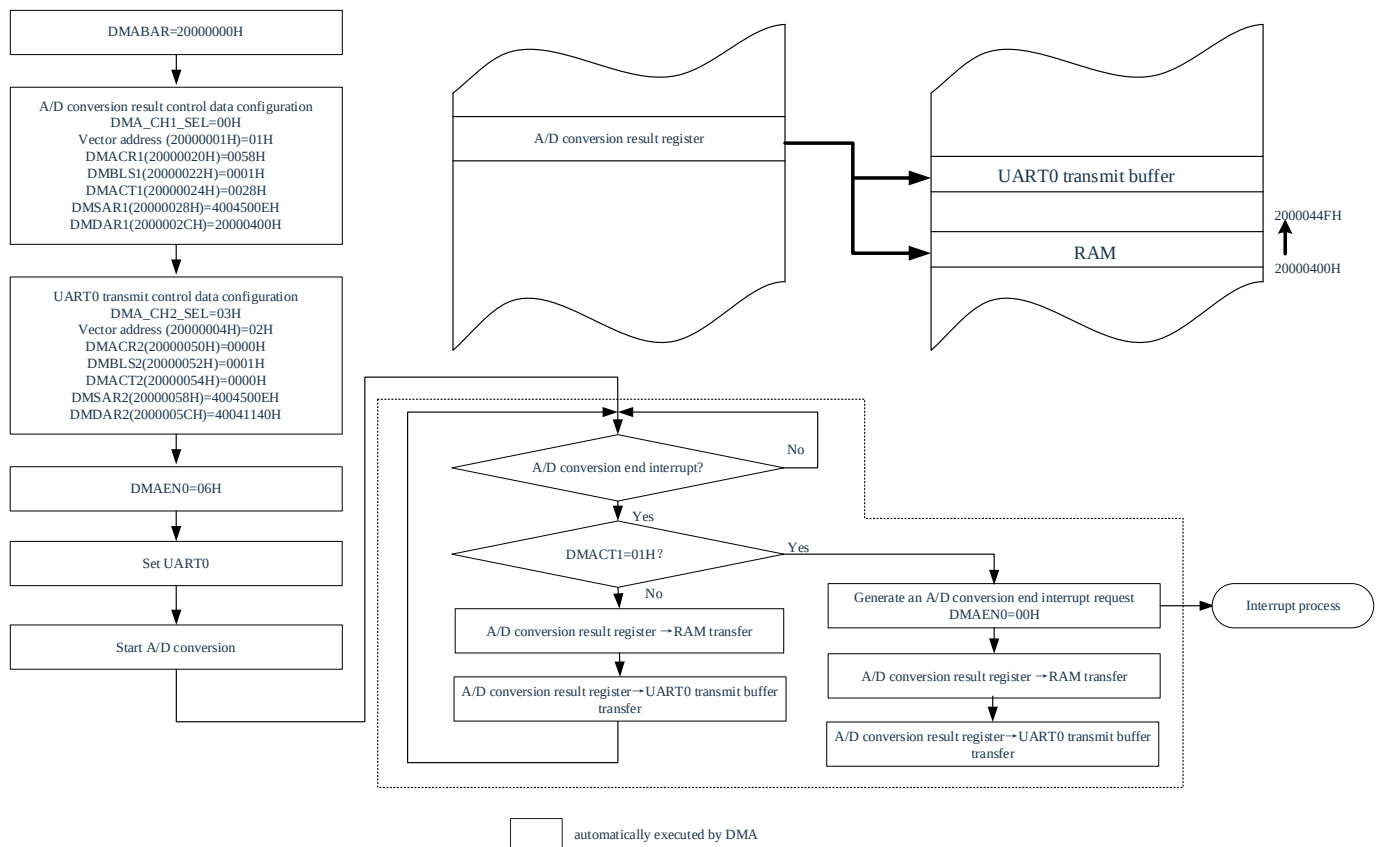
Note 1: During the second and subsequent data transfers in a chained transfer, the DMAENi (i=0) register's DMAENi0~DMAENi3 bits do not change to 0 (disabling DMA start), and no interrupt request is generated.

1. Example of chained transfer: continuously fetching A/D conversion results for UART0 transmission

Channel 1 uses the A/D conversion end interrupt to start DMA, transferring the A/D conversion results to RAM for UART transmission.

- The vector addresses are 20000001H and 20000002H.
- The control data for the A/D conversion results are allocated at 20000020H~2000002FH.
- The control data for UART0 transmission are allocated at 20000030H~2000003FH.
- The 2-byte data from the A/D conversion result register ADCR (4004500EH) is transferred to RAM at 20000400H~2000044FH, and the lower 1-byte of the A/D conversion result register (4004500EH) is transferred to the UART0 transmit buffer SDR00 (40041140H).

Figure 16-12 Example of chained transfer: Continuously fetching A/D conversion result for UART0 transmission



16.5 Cautions When Using DMA

16.5.1 DMA Control Data and Vector Table Setup

- The DMA base address register (DMABAR) must be changed with all DMA boot sources set to a disabled state.
- The DMA base address register (DMABAR) can only be rewritten once.
- The DMACRj, DMBLSj, DMACTj, DMRLDj, DMSARj, and DMDARj registers must be modified when the corresponding DMAENi (i=0) register's DMAENi0~DMAENi3 bits are set to 0 (DMA start disabled).
- The starting address of the DMA control data area in the vector table must be modified when the corresponding DMAENi (i=0) register's DMAENi0~DMAENi3 bits are set to 0 (DMA start disabled).

16.5.2 DMA Control Data Region and DMA Vector Table Region Allocation

(1) The allocation of DMA control data and vector table regions varies depending on the product and usage conditions.

(2) The stack region, DMA control data region, and DMA vector table region cannot overlap.

16.5.3 DMA Execution Clock Counts

The execution status and the required clock counts when DMA is started are shown in Table 16-8.

Table 16-8: Execution and number of clocks required when DMA is started

Read vector	Control data		Read data	Write data
	Read	Write back		
1	4	Note 1	Note 2	Note 2

Note 1: For the number of clocks required to write back control data, please refer to Table 16-9.

Note 2: For the number of clocks required to read and write data, please refer to Table 16-10.

Table 16-9: Number of clocks required to write back control data

Settings for the DMACR register				Address setting		Write back of control register				Clock count
DAMOD	SAMOD	RPTSEL	MODE	Source	Target	DMACTj register	DMRLDj register	DMSARj register	DMDARj register	
0	0	X	0	fixed	fixed	write-back	write-back	do not write back	do not write back	1
0	1	X	0	incremental	fixed	write-back	write-back	write-back	do not write back	2
1	0	X	0	fixed	incremental	write-back	write-back	do not write back	write-back	2
1	1	X	0	incremental	incremental	write-back	write-back	write-back	write-back	3
0	X	1	1	repeat region	fixed	write-back	write-back	write-back	do not write back	2
1	X	1	1		incremental	write-back	write-back	write-back	write-back	3
X	0	0	1	fixed	repeat region	write-back	write-back	do not write back	write-back	2
X	1	0	1	incremental		write-back	write-back	write-back	write-back	3

Note: j=0~7, X: “0” or “1”

Table 16-10: Number of clocks required to read and write data

Execution status	RAM	Code flash	Data flash	Special function register (SFR)	Extended special function register (2nd SFR)	
					No wait	Wait
Read data	1	2	4	1	1	1+wait cycles
Write data	1	—	—	1	1	1+wait cycles

16.5.4 DMA Response Time

The DMA response time is shown in Table 16-11. DMA response time refers to the time from detecting the DMA start source to the beginning of DMA transfer, excluding the DMA execution clock counts.

Table 16-11 DMA response time

	Minimum time	Maximum time
Response time	3 clocks	23 clocks

However, the DMA response time may be delayed under the following conditions. The number of delayed clocks depends on the conditions:

- When executing instructions from internal RAM, the maximum response time is 20 clocks.

Note: 1 clock: $1/F_{CLK}$ (F_{CLK} : CPU/peripheral hardware clock)

16.5.5 DMA Boot Sources

- The same boot source cannot be input during the period from the input of the DMA start source to the end of the DMA transfer.
- The DMA start enable bit corresponding to the boot source should not be operated at the position where the DMA start source is generated.
- If there is a DMA boot source contention, the priority is determined based on which source the CPU accepts for DMA transfer. For information about the priority of start sources, please refer to 16.4.3 Vector Table.

16.5.6 Operation in Standby Mode

State	DMA operation
Sleep Mode	DMA can operate.
Deep Sleep Mode	DMA start sources can be accepted and DMA transfer can be performed. ^{Note 1}

Note: In deep sleep mode, DMA transfer can be performed once a DMA boot source is detected, and after the transfer is completed, it will return to deep sleep mode. However, since code flash and data flash are stopped in deep sleep mode, flash memory cannot be set as a transfer source.

Chapter 17 A/D Converter

17.1 Functions of A/D Converter

The A/D converter is a converter that converts analog inputs into digital values, capable of controlling A/D conversion for up to 34 analog channels (30 input pins and 4 internal channels).

The A/D converter has the following features:

- 12-bit resolution A/D conversion

The analog input for conversion can be selected from the following channels: ANI0~ANI20, ANI24~ANI32, VDD, GND, BG, and TS. The A/D conversion is performed in a repeated manner with 12-bit resolution. After each A/D conversion, an interrupt request (INTAD) is generated (depending on the selected mode).

Various A/D conversion modes can be set by configuring the mode combinations.

Trigger mode	Software trigger	Start the conversion with software operations.
	Hardware trigger no-wait mode	Start the conversion by detecting hardware triggers.
	Hardware trigger wait mode	In the conversion standby state where the power is cut off, the power is turned on by detecting a hardware trigger, and the conversion starts automatically after a waiting time for the A/D power supply to stabilize.
Channel selection mode	Select mode	Select 1 channel of analog input for A/D conversion.
	Scan mode	The A/D converter performs sequential A/D conversions on four analog input channels. You can select a continuous range of four channels from ANI0~ANI20 or ANI24~ANI32 as the analog inputs.
Conversion mode	Single conversion mode	Perform 1 A/D conversion for the selected channel.
	Continuous conversion mode	Perform continuous A/D conversion for the selected channel until stopped by the software.

Figure 17-1: Block diagram of A/D converter

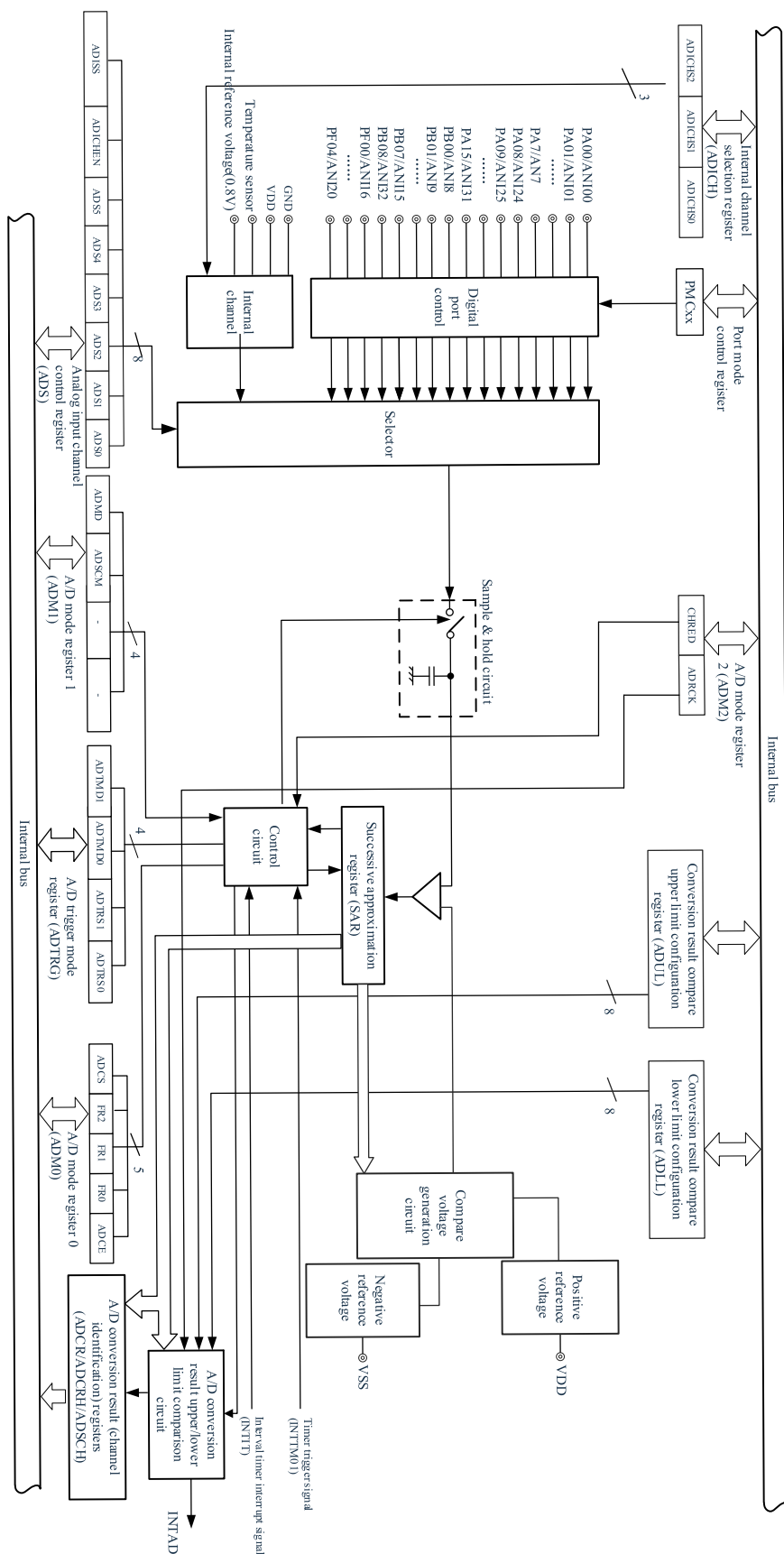
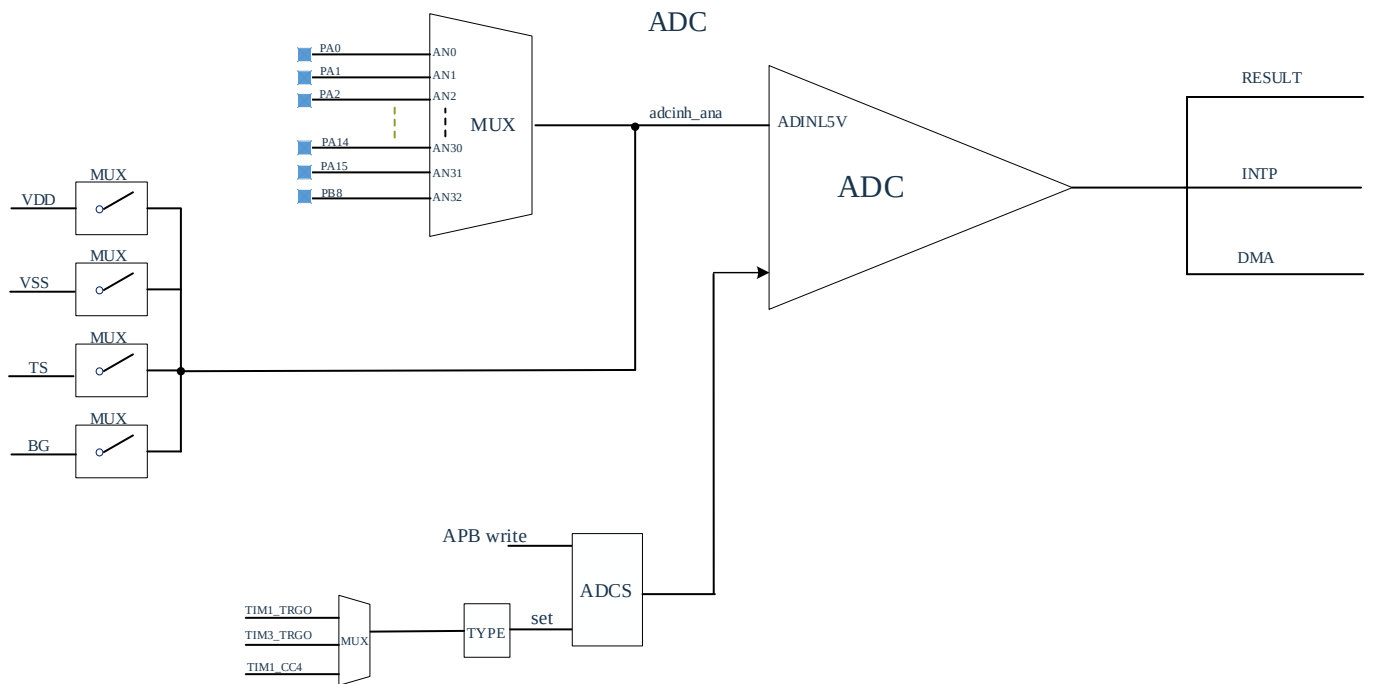


Figure 17-2 Block diagram of ADC



17.2 Register Mapping

ADC base address = 0x40045000 RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Register description	Reset value
ADM0	0x00	R/W	A/D Converter Mode Register 0	18H
ADM1	0x02	R/W	A/D Converter Mode Register 1	00H
ADM2	0x04	R/W	A/D Converter Mode Register 2	00H
ADTRG	0x06	R/W	A/D Converter Trigger Mode Register	00H
ADS	0x08	R/W	Analog Input Channel Specification Register	00H
ADICHS	0x09	R/W	Internal Channel Selection Register	00H
ADLL	0x0A	R/W	Conversion Result Comparison Lower Limit Configuration Register	00H
ADUL	0x0B	R/W	Conversion Result Comparison Upper Limit Configuration Register	FFH
ADSCH	0x0D	RO	Scan Mode A/D Conversion Channel Identifier Register	00H
ADCR	0x0E	RO	12-Bit A/D Conversion Result Register	0000H
ADCRH	0x0F	RO	8-Bit A/D Conversion Result Register	00H
ADSMPWAIT	0x15	R/W	A/D Converter Sampling Time Extension Control Register	00H

17.3 Register Description

17.3.1 A/D Converter Mode Register 0 (ADM0)

This register is used to set the A/D conversion clock, and to control the start or stop of the conversion. The ADM0 register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register is set to “00H”.

Bit	Symbol	Description	Reset value
7	ADCS	A/D conversion operation control 0: Write: Stop conversion operation Read: Stop conversion operation/standby state 1: Write: Enable conversion operation Read: In software-triggered mode, conversion is running; in hardware-triggered waiting mode, A/D power is in a waiting stable state + conversion is running	0
6	-	Reserved, set to 0	-
5:3	FR<2:0>	ADC clock division selection bit 000: F _{CLK} /16 001: F _{CLK} /12 010: F _{CLK} /8 011: F _{CLK} /6 100: F _{CLK} /3 101: F _{CLK} /2 110: F _{CLK} /1 Other: Reserved, forbidden selection.	0x3
2:1	-	Reserved, set to 0.	-
0	ADCE	A/D voltage comparator enable bit ^{Note 2} 0: Disable A/D voltage comparator 1: Enable A/D voltage comparator	0

Note 1: For detailed information about the A/D conversion of the FR2~FR0 bits, please refer to Table 17-3 A/D Conversion Time Selection.

Note 2: It takes 1 μs from the start of operation for the operation to stabilize. While in the software trigger mode or hardware trigger no-wait mode, when the ADCS bit is set to 1 after 1 μs or more has elapsed from the time ADCE bit is set to 1, the conversion result is valid. Otherwise, ignore data of the conversion. In hardware-triggered wait mode, a wait time of 1us is guaranteed by design.

Caution:

1. The FR2~FR0 bits must be changed when ADCS = 0 (conversion stopped state).
2. Setting ADCS = 1, ADCE = 0 is prohibited.
3. Setting ADCS = 0, ADCE = 0 to ADCS = 1, ADCE = 1 through 8-bit manipulation instructions is prohibited. The settings must be performed according to the steps in 17.6 A/D Converter Setup Flowchart.

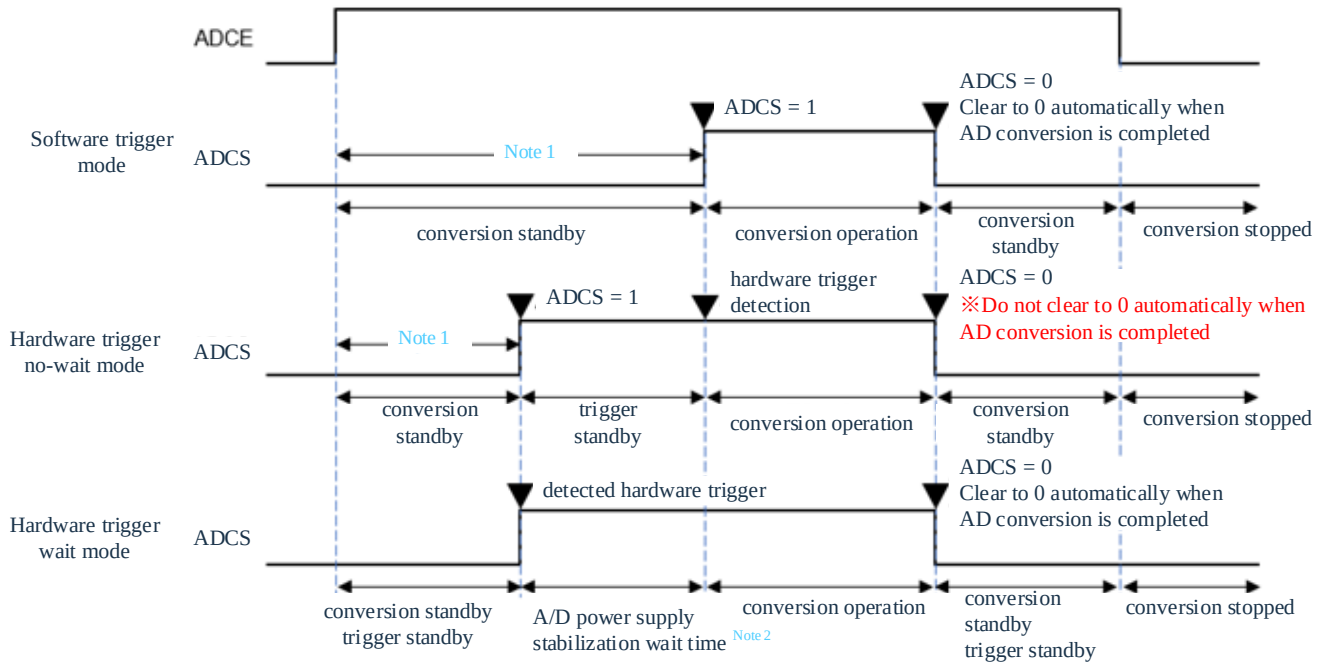
Table 17-1 ADCS bit and ADCE bit settings

ADCS	ADCE	A/D conversion operation
0	0	Conversion stop state
0	1	Conversion standby state
1	0	Settings are disabled
1	1	Conversion running state

Table 17-2: ADCS bit set and clear conditions

A/D conversion mode			Set condition	Clear condition
Software trigger	Select mode	Continuous conversion mode	When writing “1” to the ADCS bit	When writing “0” to the ADCS bit
		Single conversion mode		•When writing “0” to the ADCS bit •Automatically cleared at the end of A/D conversion.
	Scan mode	Continuous conversion mode		When writing “0” to the ADCS bit
		Single conversion mode		•When writing “0” to the ADCS bit •When the conversion of the four selected channels is complete, it will automatically reset to 0.
Hardware triggered no-wait mode	Select mode	Continuous conversion mode	When writing “1” to the ADCS bit	When writing “0” to the ADCS bit
		Single conversion mode		•When writing “0” to the ADCS bit
	Scan mode	Continuous conversion mode		When writing “0” to the ADCS bit
		Single conversion mode		•When writing “0” to the ADCS bit
Hardware trigger wait mode	Select mode	Continuous conversion mode	When the input hardware triggers	When writing “0” to the ADCS bit
		Single conversion mode		•When writing “0” to the ADCS bit •Automatically cleared at the end of A/D conversion.
	Scan mode	Continuous conversion mode		When writing “0” to the ADCS bit
		Single conversion mode		•When writing “0” to the ADCS bit •When the conversion of the four selected channels is complete, it will automatically reset to 0.

Figure 17-3 Action state diagram with A/D modes



Note 1: While in the software trigger mode or hardware trigger no-wait mode, the time from the rising of the ADCE bit to the rising of the ADCS bit must be 1 μ s or longer to stabilize the internal circuit.

Note 2: In hardware triggered wait mode, the A/D power supply stabilization time of 1 μ s is guaranteed by design.

Caution:

1. To use the hardware trigger wait mode, it is prohibited to set the ADCS bit to “1” (automatically switches to “1” when a hardware trigger signal is detected). However, ADCS can be set to “0” in order to set the A/D conversion standby state.
2. The ADCE bit must be rewritten when the ADCS bit is “0” (stop conversion/conversion standby state).
3. To end the A/D conversion, the hardware trigger interval must be set to at least the following times:

Hardware triggered no-wait mode: $2 F_{CLK}$ clocks + A/D conversion time

Hardware triggered wait mode: $2 F_{CLK}$ clocks + A/D power supply stabilization wait time + A/D conversion time

Remark: F_{CLK} : CPU/peripheral hardware clock frequency

Table 17-3 A/D conversion time selection (1/2)

1. Without A/D power stabilization wait time (software trigger mode/hardware trigger no wait mode)

A/D converter mode register 0 (ADM0)			A/D sampling time extension register (ADSMPWAIT)	Conversion clock ADCLK frequency (F_{AD})	12-bit resolution conversion time	
FR2	FR1	FR0	ADSMPWAIT		Converted clocks	Conversion time
0	0	0	0	$F_{clk}/16$	16 ADCLKs (sample clocks: 4 ADCLKs)	$16/F_{AD}$
0	0	1		$F_{clk}/12$		
0	1	0		$F_{clk}/8$		
0	1	1		$F_{clk}/6$		
1	0	0		$F_{clk}/3$		
1	0	1		$F_{clk}/2$		
1	1	0		$F_{clk}/1$		
0	0	0	1	$F_{clk}/16$	20 ADCLKs (sample clocks: 8 ADCLKs)	$20/F_{AD}$
0	0	1		$F_{clk}/12$		
0	1	0		$F_{clk}/8$		
0	1	1		$F_{clk}/6$		
1	0	0		$F_{clk}/3$		
1	0	1		$F_{clk}/2$		
1	1	0		$F_{clk}/1$		

Caution: 1. To override FR2~FR0 bits to different data, it must be done in the conversion stop state (ADCS=0).

Remark: F_{clk} : CPU/peripheral hardware clock frequency

Table 17-3: A/D conversion time selection (2/2)

2. With A/D power stabilization wait time (hardware triggered wait mode ^{Note 1)}

A/D converter mode register 0 (ADM0)			A/D sampling time extension register (ADSMPWAIT)	Conversion clock ADCLK frequency (F_{AD})	A/D power stabilization wait time	Converted clocks	A/D power stabilization wait time + conversion time
FR2	FR1	FR0	ADSMPWAIT				
0	0	0	0	Fclk/16	1 μ s	16 ADCLKs (sample clocks:4 ADCLKs)	1 μ s+16/ F_{AD}
0	0	1		Fclk/12			
0	1	0		Fclk/8			
0	1	1		Fclk/6			
1	0	0		Fclk/3			
1	0	1		Fclk/2			
1	1	0		Fclk/1			
0	0	0	1	Fclk/16	1 μ s	20 ADCLKs (sample clocks: 8 ADCLKs)	1 μ s+20/ F_{AD}
0	0	1		Fclk/12			
0	1	0		Fclk/8			
0	1	1		Fclk/6			
1	0	0		Fclk/3			
1	0	1		Fclk/2			
1	1	0		Fclk/1			

Note 1: In continuous conversion mode, the A/D power stabilization wait time only occurs after the first hardware trigger is detected (refer to Table 17-3).

Cautions:

1. To rewrite the FR2~FR0 bits with different data, it must be done in the conversion stop state (ADCS = 0).
2. In hardware trigger wait mode, the conversion time includes the A/D power stabilization wait time after detecting the hardware trigger.

Remark: F_{CLK} : CPU/peripheral hardware clock frequency.

17.3.2 A/D Converter Mode Register 1 (ADM1)

This is a register that sets the A/D conversion mode.

The ADM1 register is set by an 8-bit memory manipulation instruction.

After the reset signal is generated, the value of this register changes to “00H”.

Bit	Symbol	Description	Reset value
7	ADMD	A/D conversion channel selection mode 0: Select mode 1: Scan mode	0
6:4	-	Reserved, set to 0.	-
3	ADSCM	A/D conversion mode selection 0: Continuous conversion mode 1: Single conversion mode	0
2:0	-	Reserved	-

Caution:

1. To rewrite the ADM1 Register, it must be done in the conversion stop state (ADCS = 0, ADCE = 0).
2. To properly end an A/D conversion, the hardware trigger interval must be set to at least the following time:

In hardware-triggered no-wait mode: $2 F_{CLK}$ cycles + A/D conversion time

In hardware-triggered wait mode: $2 F_{CLK}$ cycles + A/D power stabilization wait time + A/D conversion time

Remark 1: F_{CLK} : CPU/peripheral hardware clock frequency.

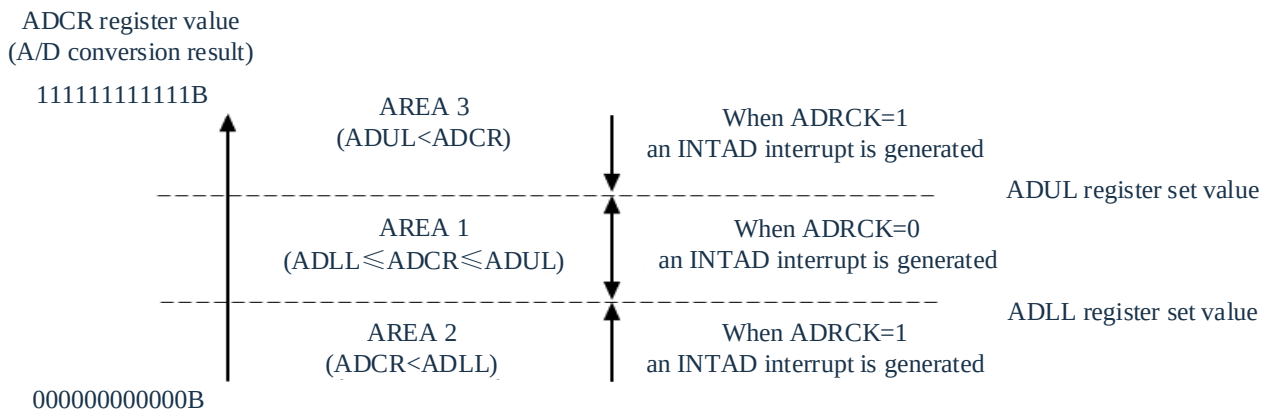
17.3.3 A/D Converter Mode Register 2 (ADM2)

The ADM2 register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “00H”.

Bit	Symbol	Description	Reset value
7:4	-	Reserved, set to 0.	-
3	ADRCK	Checking of upper and lower limit values of conversion results The interrupt signals (INTAD) for AREA1~AREA3 are generated within the ranges shown in Figure 17-4. 0: When ADLL register $\leq$ ADCR register $\leq$ ADUL register (AREA1), an interrupt signal (INTAD) is generated. 1: < ADCR register (AREA3), an interrupt signal (INTAD) is generated.	0
2	-	Reserved, set to 0.	-
1	CHRED	A/D converter scan mode channel identification output enable 0: In scan mode, the conversion channel identification number is not output. 1: In scan mode, the conversion channel identification number is output.	0
0	-	Reserved, set to 0.	0

Figure 17-4 ADRCK bit interrupt signal generation range



Caution: 1. To override the ADM2 register, it must be done in the conversion stop state (ADCS=0).

Remark: When INTAD does not occur, the A/D conversion result is not saved to the ADCR register and ADCRH register.

17.3.4 A/D Converter Trigger Mode Register (ADTRG)

This is a register that sets the A/D conversion trigger mode and hardware trigger signal.

The ADTRG register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “00H”.

Bit	Symbol	Description	Reset value
7:6	ADTMD<1:0>	A/D conversion trigger mode selection 00: Software trigger mode 01: Software trigger mode 10: Hardware trigger no-wait mode 11: Hardware trigger wait mode	0x0
5:2	-	Reserved, set to 0.	-
1:0	ADTRS<1:0>	Selection of hardware trigger signal 00: TIM1_TRGO 01: TIM1_CC4 10: Reserved 11: TIM3_TRGO	0x0

Caution:

1. To rewrite the ADTRG register, it must be in the conversion stop state (ADCS=0, ADCE=0).
2. To end the A/D conversion properly, the hardware trigger interval must be set to at least the following times:

Hardware trigger no-wait mode: $2 F_{CLK}$ clocks + A/D conversion time

Hardware trigger wait mode: $2 F_{CLK}$ clocks + A/D power supply stabilization wait time + A/D conversion time

Remark 1: F_{CLK} : CPU/peripheral hardware clock frequency

17.3.5 Analog Input Channel Specification Register (ADS)

This is a register that specifies the analog voltage input channel to be A/D converted.

The ADS register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “00H”.

Bit	Symbol	Description	Reset value
7	ADISS	See the table below	0
6	ADICHEN	See the table below	0
5:0	ADS<5:0>	See the table below	0x0

Select mode (ADM1.ADMD=0)

ADS register set value								CH selection
ADISS	ADICHEN	ADS5	ADS4	ADS3	ADS2	ADS1	ADS0	
0	0	0	0	0	0	0	0	ANI0
0	0	0	0	0	0	0	1	ANI1
0	0	0	0	0	0	1	0	ANI2
0	0	0	0	0	0	1	1	ANI3
0	0	0	0	0	1	0	0	ANI4
0	0	0	0	0	1	0	1	ANI5
0	0	0	0	0	1	1	0	ANI6
0	0	0	0	0	1	1	1	ANI7
0	0							
0	0	0	1	1	1	1	0	ANI30
0	0	0	1	1	1	1	1	ANI31
0	0	1	0	0	0	0	0	ANI32
0	0	1	1	1	1	1	1	Channels all OFF
0	1	0	0	0	0	0	0	Internal channels enabled
Others								Settings are disabled

Scan mode (ADM1.ADMD=1)

ADISS	ADS5	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel			
							Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	0	0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
0	0	0	0	1	0	1	ANI5	ANI6	ANI7	ANI8
0	0	0	0	1	1	0	ANI6	ANI7	ANI8	ANI9
0	0	0	0	1	1	1	ANI7	ANI8	ANI9	ANI10
0	0	0	1	0	0	0	ANI8	ANI9	ANI10	ANI11
0	0	0	1	0	0	1	ANI9	ANI10	ANI11	ANI12
0	0	0	1	0	1	0	ANI10	ANI11	ANI12	ANI13
0	0	0	1	0	1	1	ANI11	ANI12	ANI13	ANI14
0	0	0	1	1	0	0	ANI12	ANI13	ANI14	ANI15
0	0	0	1	1	0	1	ANI13	ANI14	ANI15	ANI16
0	0	0	1	1	1	0	ANI14	ANI15	ANI16	ANI17
0	0	0	1	1	1	1	ANI15	ANI16	ANI17	ANI18
0	0	1	0	0	0	0	ANI16	ANI17	ANI18	ANI19
0	0	1	0	0	0	1	ANI17	ANI18	ANI19	ANI20
0	0	1	1	0	0	0	ANI24	ANI25	ANI26	ANI27
0	0	1	1	0	0	1	ANI25	ANI26	ANI27	ANI28
0	0	1	1	0	1	0	ANI26	ANI27	ANI28	ANI29
0	0	1	1	0	1	1	ANI27	ANI28	ANI29	ANI30
0	0	1	1	1	0	0	ANI28	ANI29	ANI30	ANI31
0	0	1	1	1	0	1	ANI29	ANI30	ANI31	ANI32
Others							Settings are disabled			

Caution:

1. Bits 4, 5, and 6 must be set to 0.
2. For ports configured as analog inputs via the PMCx register, they can be designated as analog inputs for A/D conversion using the ADS register.
3. For pins configured as digital input/output via the PMCxx register, they cannot be set as analog inputs via the ADS register.
4. To rewrite the ADISS bit, it must be done in the conversion stop state (ADCS = 0, ADCE = 0).
5. To enter deep sleep mode or to enter sleep mode while the CPU is running at a sub-system clock, the ADISS bit must not be set to 1.

17.3.6 A/D Internal Channel Selection Register (ADICHS)

This is the register for setting the internal A/D channels.

The ADICH register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “00H”.

Bit	Symbol	Description	Reset value
7:3	-	Reserved, set to 0.	-
2:0	ADICHS<2:0>	Internal channel selection bit (The bit is only valid when the internal channels are enabled through the ADS register with ADS.ADICHEN = 1) 000: VDD 001: GND 010: TS(Temperature Sensor Signal) 011: BG(0.8V) Other: Reserved, forbidden selection	0x0

17.3.7 12-Bit A/D Conversion Channel Identifier Register (ADSCHS)

This is an 8-bit register that stores the A/D conversion channel identifier. This register is read-only. Each time the A/D conversion finishes, it outputs the channel corresponding to the current A/D conversion result.

In selection mode, the read value of this register is fixed at 0. In scan mode, it can be configured by ADM2.CHRDE = 1 to indicate the channel number of the conversion result.

The ADCHO register is set by a 9-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “00H”.

Bit	Symbol	Description	Reset value
7:6	-	Reserved	
5:0	ADSCH<5:0>	Conversion channel identifier In selection mode (ADM1.ADMD = 0), the read value of ADCH[5:0] is always 6'b0. In scan mode (ADM1.ADMD = 1), and when ADM2.CHEDE = 1, the ADSCH[5:0] result corresponds to the selected channel. 000000: ANI0 000001: ANI1 000010: ANI2 011111: ANI31 100000: ANI32	0x0

17.3.8 12-Bit A/D Conversion Result Register (ADCR)

This is a 16-bit register that stores the A/D conversion result. This register is read-only. Each time the A/D conversion ends, the conversion result is loaded from the Successive Approximation Register (SAR) ^{Note}.

The ADCRH register is read by a 16-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “00H”.

Note: If the value of the A/D conversion result is not within the range of the set value of the A/D conversion result comparison function (set by the ADRCK bit and the ADUL/ADLL register (see Figure 17-4)), the A/D conversion result is not saved.

Bit	Symbol	Description	Reset value
15:12	-	Reserved	-
11:0	ADCR<11:0>	A/D conversion result	0

Cautions:

1. If only an 8-bit resolution A/D conversion result is needed, the high 8 bits of the conversion result can be read through the ADCRH register.
2. When performing a 16-bit access to the ADCR register, the high 12 bits of the conversion result can be read starting from bit 11.

17.3.9 8-Bit A/D Conversion Result Register (ADCRH)

This is an 8-bit register that holds the result of the A/D conversion, and holds the high 8 bits of the 12-bit resolution

Note.

The ADCRH register is read by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “00H”.

Bit	Symbol	Description	Reset value
7:0	ADCRH<7:0>	A/D conversion result	0x0

Note: If the value of the A/D conversion result is not within the range of the set value of the A/D conversion result comparison function (set by the ADRCK bit and the ADUL/ADLL register (see Figure 17-4)), the A/D conversion result is not saved.

Caution: The conversion result must be read after the conversion is completed and before configuring the ADM0, ADS registers. Otherwise, the correct conversion result may not be read.

17.3.10 A/D Converter Sampling Time Extension Control Register (ADSMPWAIT)

This register is used to extend the A/D sampling time.

The ADSMPWAIT register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “00H”.

Bit	Symbol	Description	Reset value
7:1	-	Reserved, set to 0.	-
0	ADSMPWAIT	A/D sampling time selection bit 0: The A/D sampling time is 4 ADCLKs 1: The A/D sampling time is 8 ADCLKs	0

17.3.11 Conversion Result Comparison Upper Limit Configuration Register (ADUL)

This register is used to specify the setting for checking the upper limit of the A/D conversion results.

The A/D conversion results and ADUL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified by the ADRCK bit of A/D converter mode register 2 (ADM2) (see Figure 17-4). The ADUL register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register becomes “FFH”.

Cautions:

1. Only the high 8 bits of the 12-bit A/D conversion result register (ADCR) are compared with the ADUL register and the ADLL register.
2. Rewrite the value of the ADUL register and ADLL register while conversion is stopped (ADCS = 0).
3. Rewrite the value of the ADUL register and ADLL register while ADUL > ADLL.

Bit	Symbol	Description	Reset value
7:0	ADUL<7:0>	Conversion result compare upper limit value	0xff

17.3.12 Conversion Result Comparison Lower Limit Configuration Register (ADLL)

This register is used to specify the setting for checking the lower limit of the A/D conversion results.

The A/D conversion results and ADLL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified by the ADRCK bit of A/D converter mode register 2 (ADM2) (see Figure 14-7). The ADLL register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register becomes “00H”.

Bit	Symbol	Description	Reset value
7:0	ADLL<7:0>	Conversion result compare lower limit value	0x0

Caution:

1. Only the high 8 bits of the 12-bit A/D conversion result register (ADCR) are compared with the ADUL register and the ADLL register.
2. Rewrite the value of the ADUL register and ADLL register while conversion is stopped (ADCS = 0).
3. Rewrite the value of the ADUL register and ADLL register while $ADUL > ADLL$.

17.4 Input Voltage and Conversion Results

The analog input voltage of the analog input pins (ANI0~ANI20, ANI24~ANI32) and the theoretical A/D conversion result register (12-bit A/D Conversion Result Register (ADCR)) are related by the following expressions.

$$ADCR = \text{INT} (V_{AIN} / AV_{REF} \times 4096 + 0.5) \text{ or } (ADCR - 0.5) \times AV_{REF} / 4096 \leq V_{AIN} < (ADCR + 0.5) \times AV_{REF} / 4096$$

INT(): A function that returns the integer portion of a numeric value in parentheses

V_{AIN} : Analog input voltage

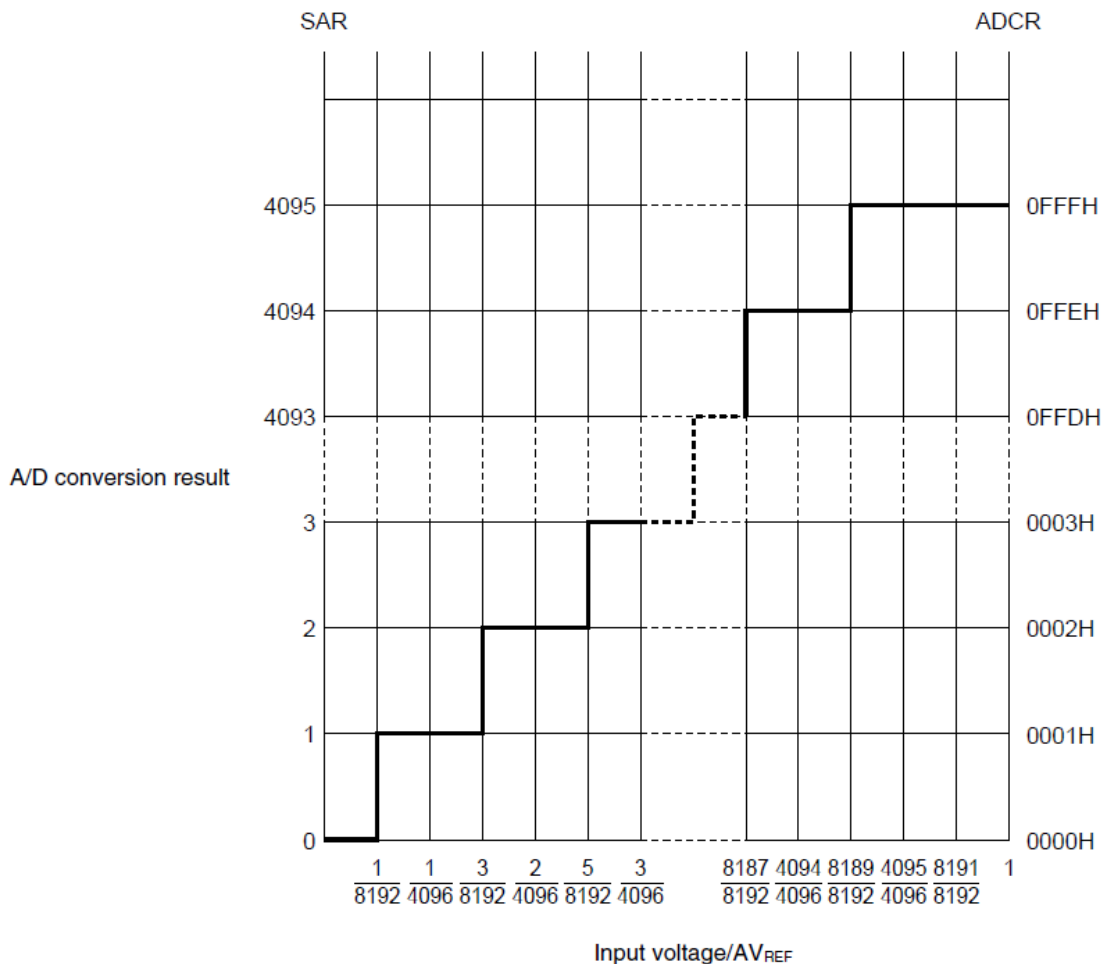
AV_{REF} : AV_{REF} pin voltage

ADCR: The value of the A/D conversion result register (ADCR).

SAR: Successive approximation register

The analog input voltage and the A/D conversion result are shown in the following figure.

Figure 17-5: Analog input voltage vs. A/D conversion result



Remark: AV_{REF} is the positive (+) reference voltage of the A/D converter.

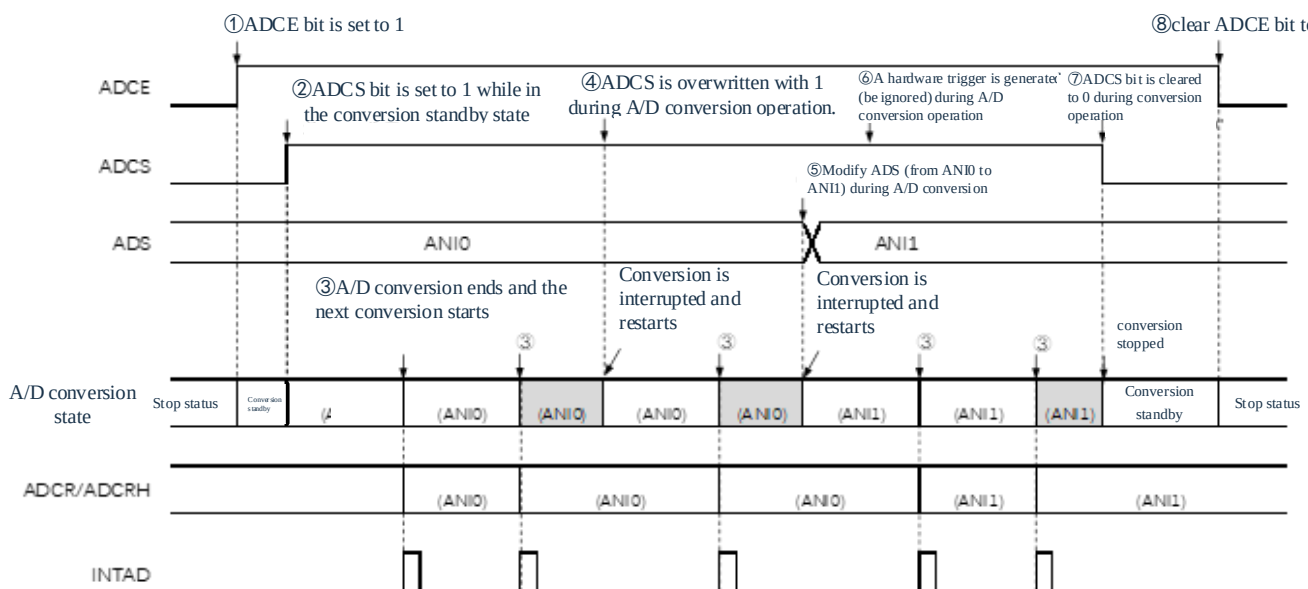
17.5 Operation Mode of A/D Converter

The A/D converter conversion operations are described below. For the setting of each mode, please refer to “17.6 A/D converter setup flowchart”.

17.5.1 Software Trigger Mode (Select Mode, Sequential Conversion Mode)

- ① In the stop state, enter A/D conversion standby state by setting the ADCE bit of the A/D converter mode register 0 (ADM0) to “1”.
- ② After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).
- ③ When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.
- ④ When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts.
- ⑤ When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register.
- ⑥ Even if a hardware trigger is input during conversion operation, A/D conversion does not start.
- ⑦ When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.
- ⑧ When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

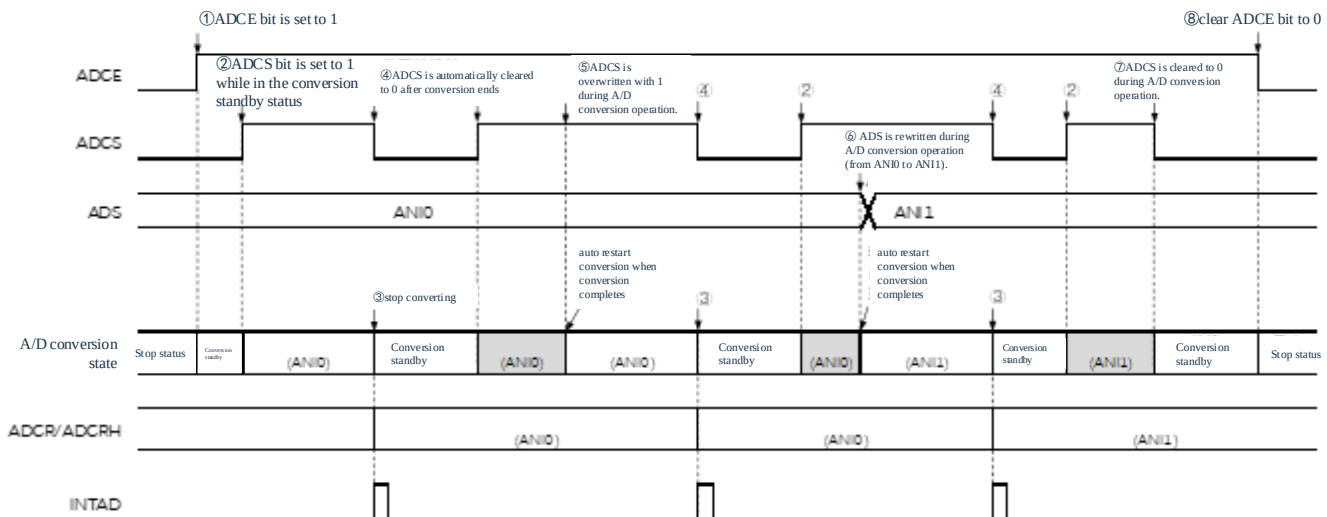
Figure 17-6 Example of software trigger mode (select mode, continuous conversion mode) operation timing



17.5.2 Software Trigger Mode (Select Mode, Single Conversion Mode)

- ① In the stop state, enter A/D conversion standby state by setting the ADCE bit of the A/D converter mode register 0 (ADM0) to “1”.
- ② After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).
- ③ When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.
- ④ After A/D conversion ends, the ADCS bit is automatically cleared to “0”, and the system enters the A/D conversion standby status.
- ⑤ When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts.
- ⑥ When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register.
- ⑦ When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.
- ⑧ When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

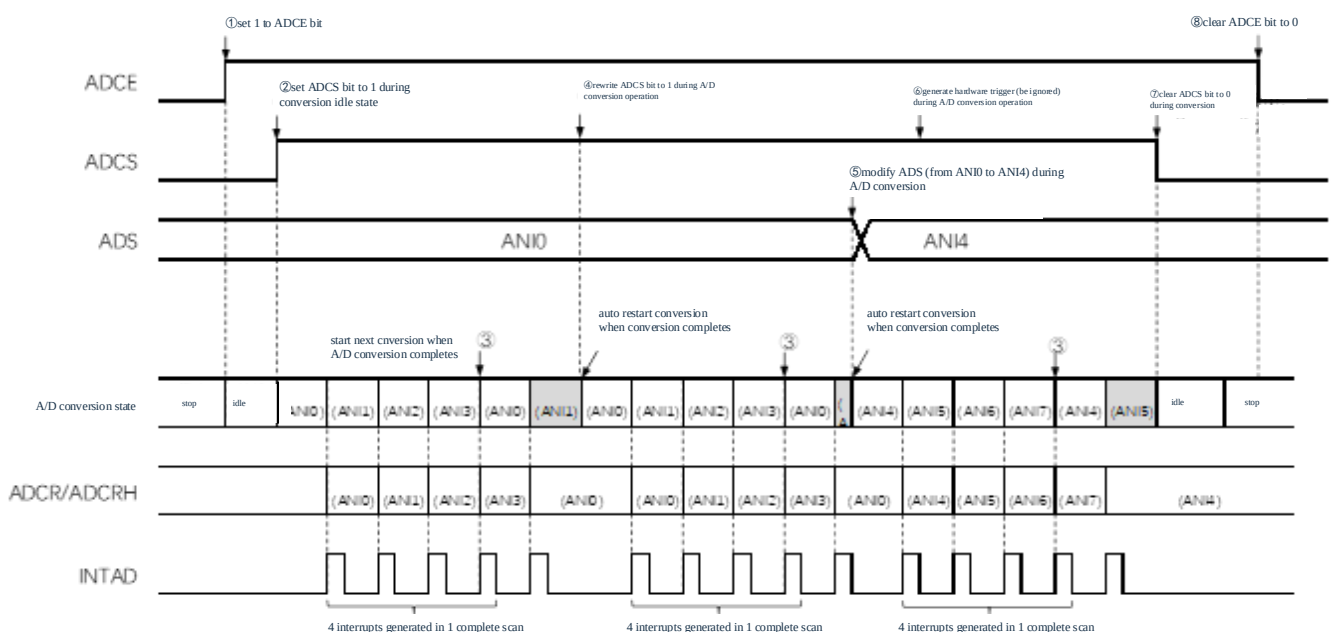
Figure 17-7 Example of software trigger mode (select mode, single conversion mode) operation timing



17.5.3 Software Trigger Mode (Scan Mode, Sequential Conversion Mode)

- ① In the stop state, enter A/D conversion standby state by setting the ADCE bit of the A/D converter mode register 0 (ADM0) to “1”.
- ② After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.
- ③ A/D conversion is sequentially performed on the four analog input channels. When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts (until all four channels are finished).
- ④ When ADCS is overwritten with “1” during conversion operation, the current A/D conversion is interrupted, and conversion restarts.
- ⑤ When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register.
- ⑥ Even if a hardware trigger is input during conversion operation, A/D conversion does not start.
- ⑦ When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.
- ⑧ When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

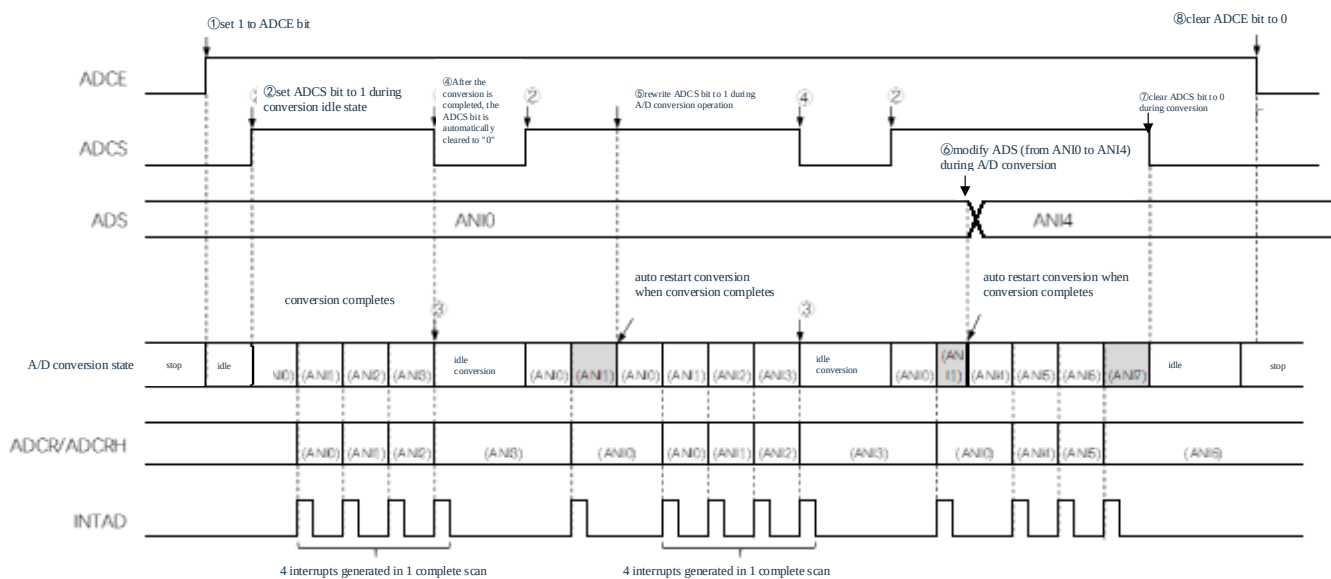
Figure 17-8: Example of software trigger mode (scan mode, sequential conversion mode) operation timing



17.5.4 Software Trigger Mode (Scan Mode, Single Conversion Mode)

- ① In the stop state, enter A/D conversion standby state by setting the ADCE bit of the A/D converter mode register 0 (ADM0) to “1”.
- ② After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.
- ③ A/D conversion is sequentially performed on the four analog input channels. When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.
- ④ After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.
- ⑤ When ADCS is overwritten with “1” during conversion operation, the current A/D conversion is interrupted, and conversion restarts.
- ⑥ When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register.
- ⑦ When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.
- ⑧ When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

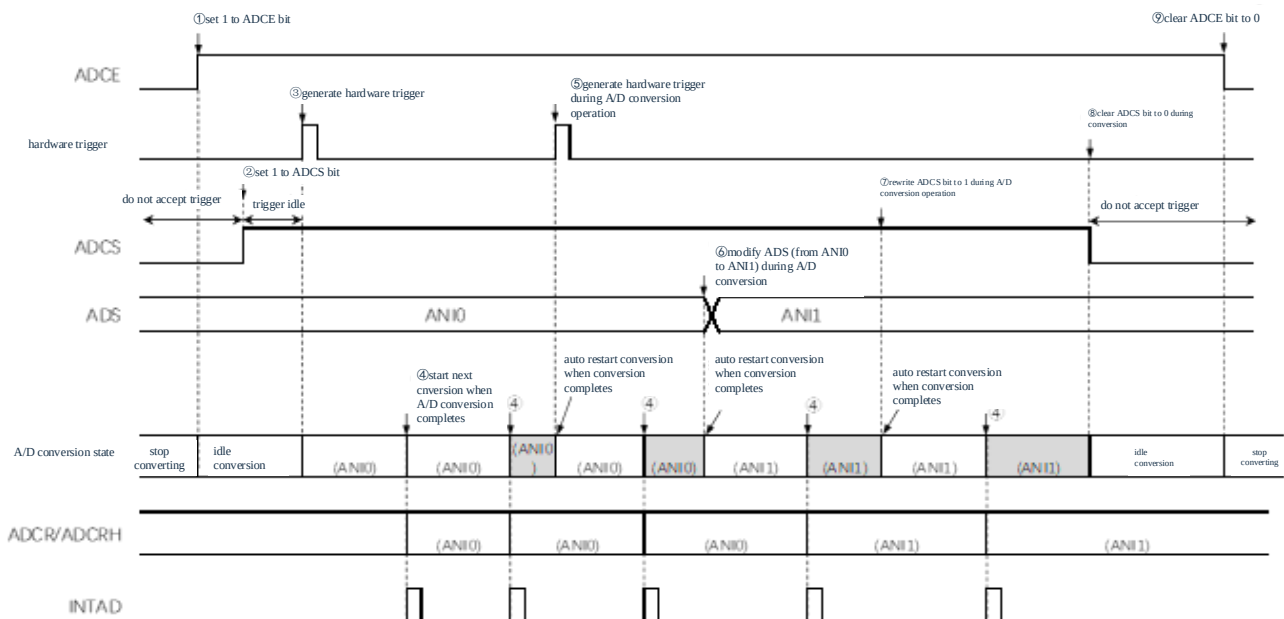
Figure 17-9: Example of software trigger mode (scan mode, single conversion mode) operation timing



17.5.5 Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode)

- ① In the stop state, enter A/D conversion standby state by setting the ADCE bit of the A/D converter mode register 0 (ADM0) to 1.
 - ② After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.
 - ③ If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).
 - ④ When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.
 - ⑤ If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts.
 - ⑥ When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register.
 - ⑦ When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts.
 - ⑧ When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.
 - ⑨ When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status.
- When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

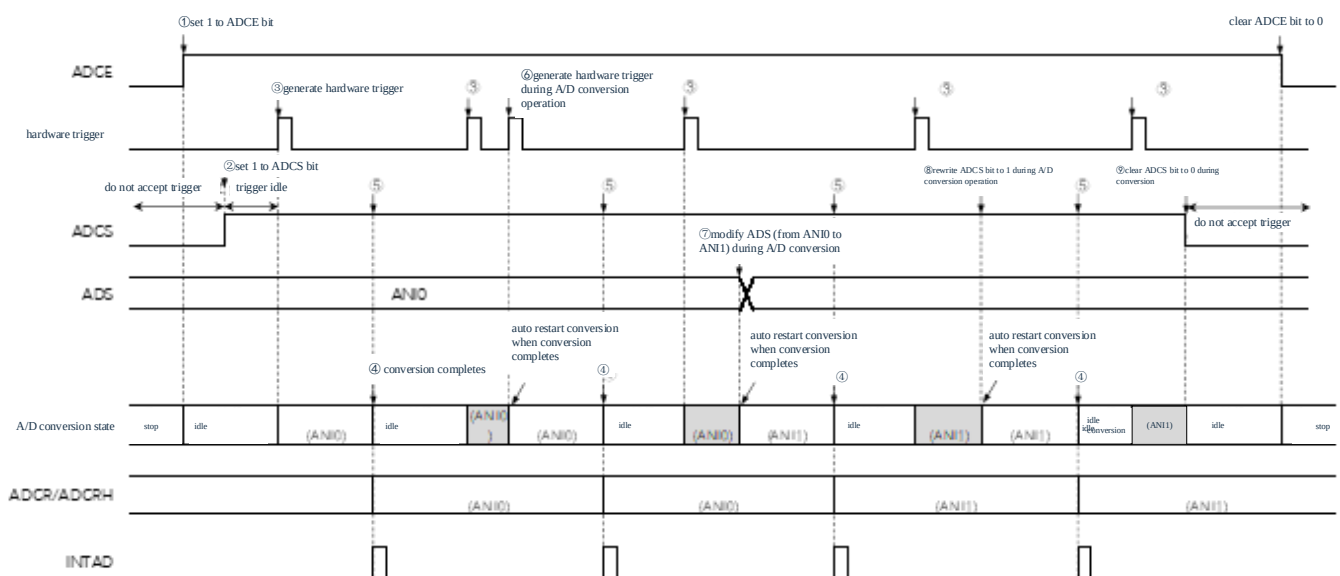
Figure 17-10: Example of hardware trigger no-wait mode (select mode, sequential conversion mode) operation timing



17.5.6 Hardware Trigger No-Wait Mode (Select Mode, Single Conversion Mode)

- ① In the stop state, enter A/D conversion standby state by setting the ADCE bit of the A/D converter mode register 0 (ADM0) to 1.
- ② After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.
- ③ If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).
- ④ When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.
- ⑤ After A/D conversion ends, the ADCS bit remains set to “1”, and the system enters the A/D conversion standby status.
- ⑥ If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts.
- ⑦ When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register.
- ⑧ When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts.
- ⑨ When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.
- ⑩ When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

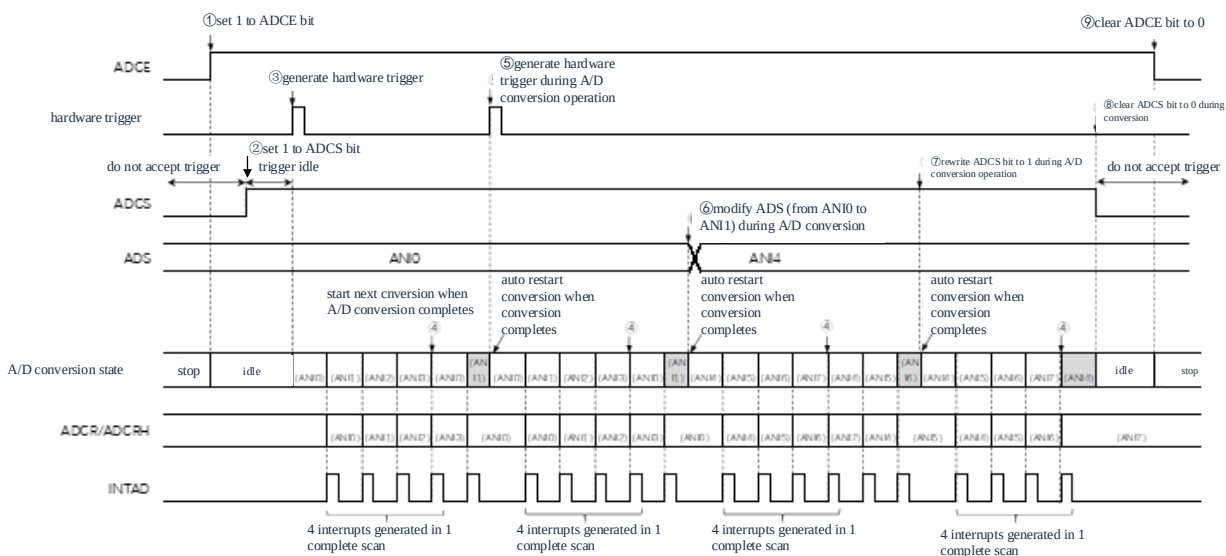
Figure 17-11 Example of hardware trigger no-wait mode (select mode, single conversion mode) operation timing



17.5.7 Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode)

- ① In the stop state, enter A/D conversion standby state by setting the ADCE bit of the A/D converter mode register 0 (ADM0) to 1.
- ② After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.
- ③ If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.
- ④ A/D conversion is sequentially performed on the four analog input channels. When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.
- ⑤ If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel.
- ⑥ When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the channel respecified by the ADS register.
- ⑦ When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel.
- ⑧ When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.
- ⑨ When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

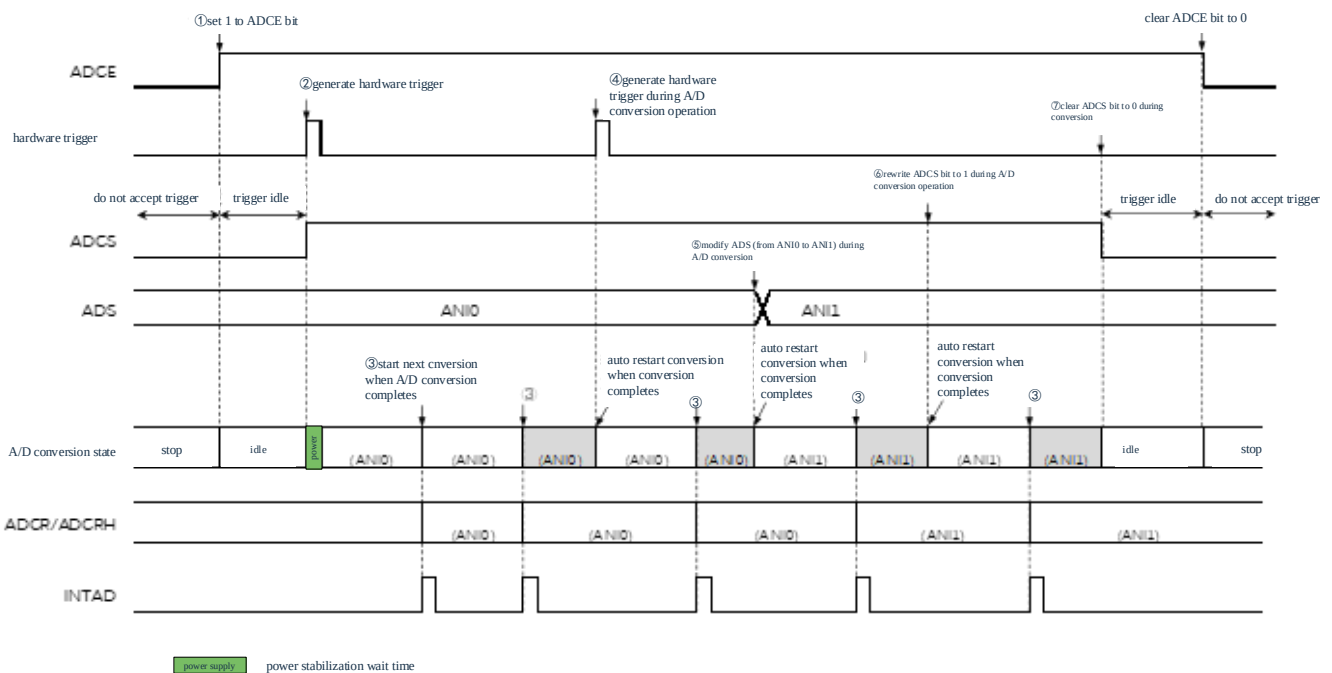
Figure 17-12 Example of hardware trigger no-wait mode (scan mode, sequential conversion mode) operation timing



17.5.9 Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode)

- ① In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.
- ② If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.
- ③ When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts. (At this time, no hardware trigger is necessary.)
- ④ If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts.
- ⑤ When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register.
- ⑥ When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts.
- ⑦ When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

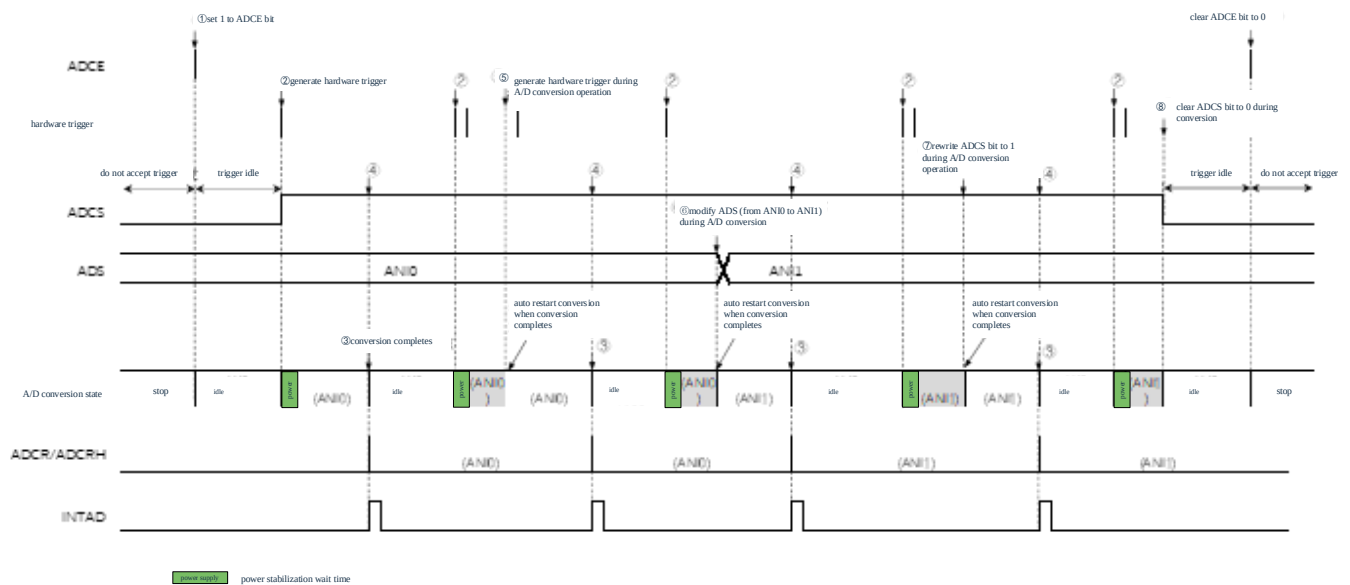
Figure 17-14 Example of hardware trigger wait mode (select mode, sequential conversion mode) operation timing



17.5.10 Hardware Trigger Wait Mode (Select Mode, Single Conversion Mode)

- ① In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.
- ② If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.
- ③ When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.
- ④ After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.
- ⑤ If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts.
- ⑥ When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register.
- ⑦ When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts.
- ⑧ When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

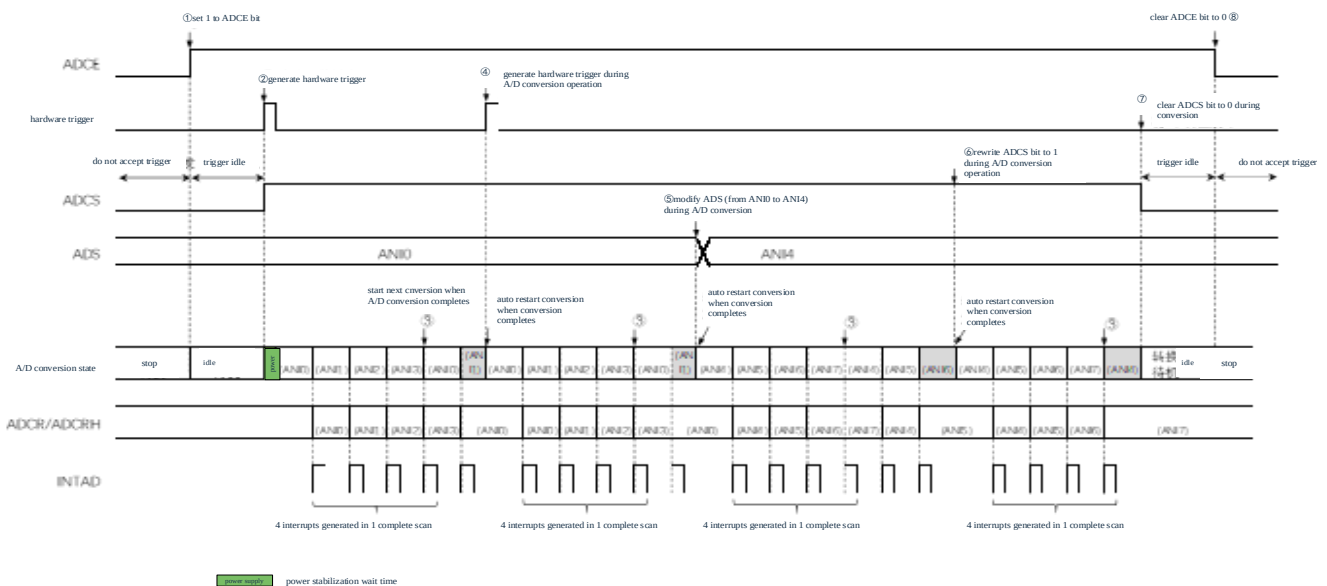
Figure 17-15: Example of hardware trigger wait mode (select mode, single conversion mode) operation timing



17.5.11 Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode)

- ① In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.
- ② If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.
- ③ A/D conversion is sequentially performed on the four analog input channels. When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.
- ④ If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel.
- ⑤ When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the channel respecified by the ADS register.
- ⑥ When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel.
- ⑦ When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

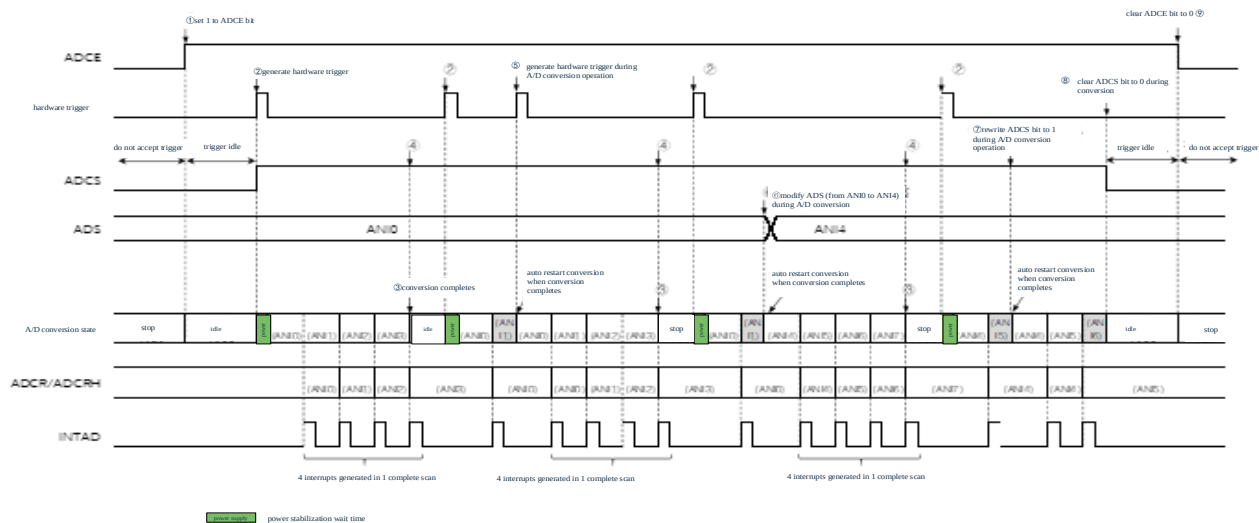
Figure 17-16: Example of hardware trigger wait mode (scan mode, sequential conversion mode) operation timing



17.5.12 Hardware Trigger Wait Mode (Scan Mode, Single Conversion Mode)

- ① In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.
- ② If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.
- ③ A/D conversion is sequentially performed on the four analog input channels. When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.
- ④ After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.
- ⑤ If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and scan conversion restarts at the first channel.
- ⑥ When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and scan conversion is performed on the channel respecified by the ADS register.
- ⑦ When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and scan conversion restarts at the first channel.
- ⑧ When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 17-17: Example of hardware trigger wait mode (scan mode, single conversion mode) operation timing

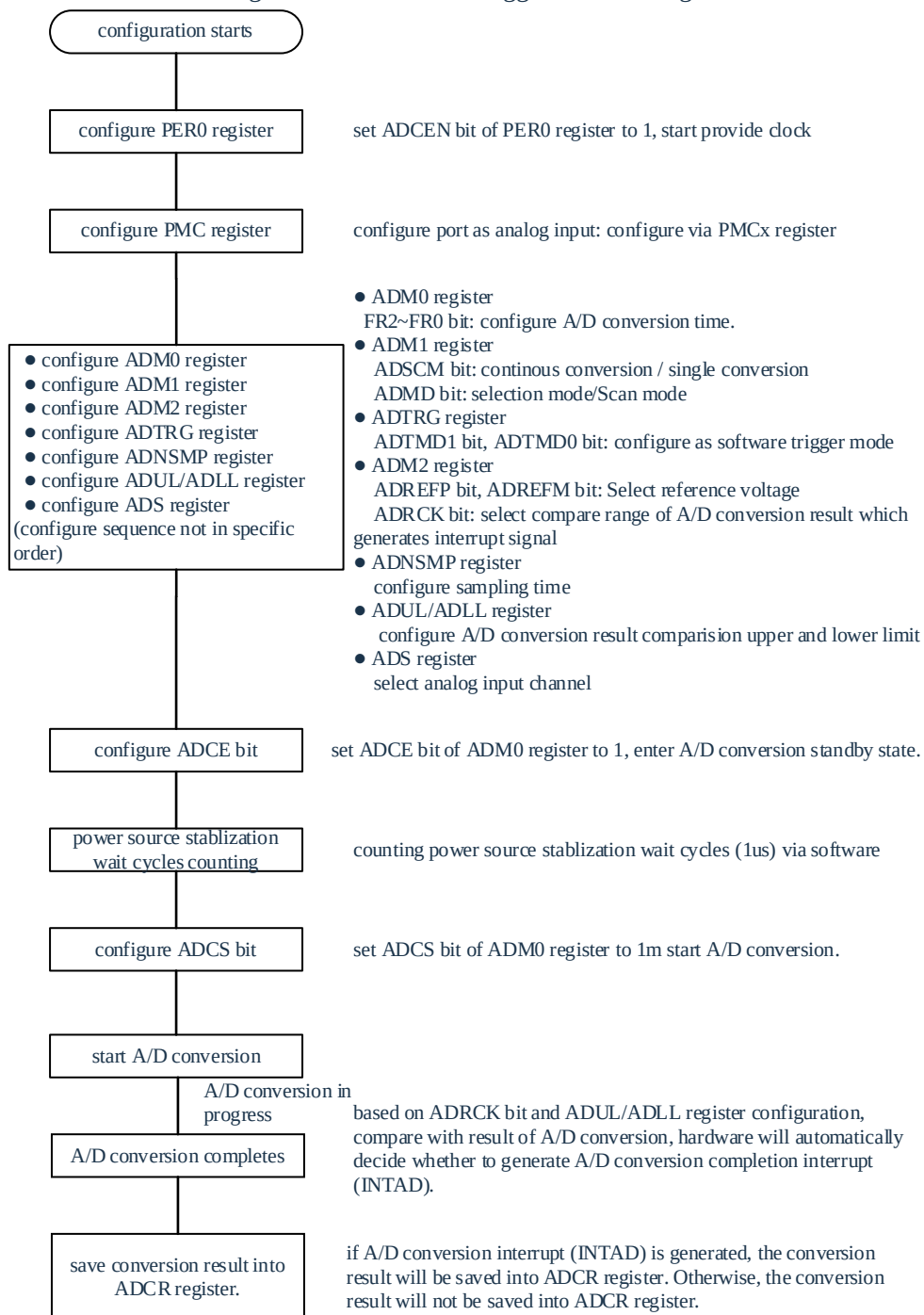


17.6 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

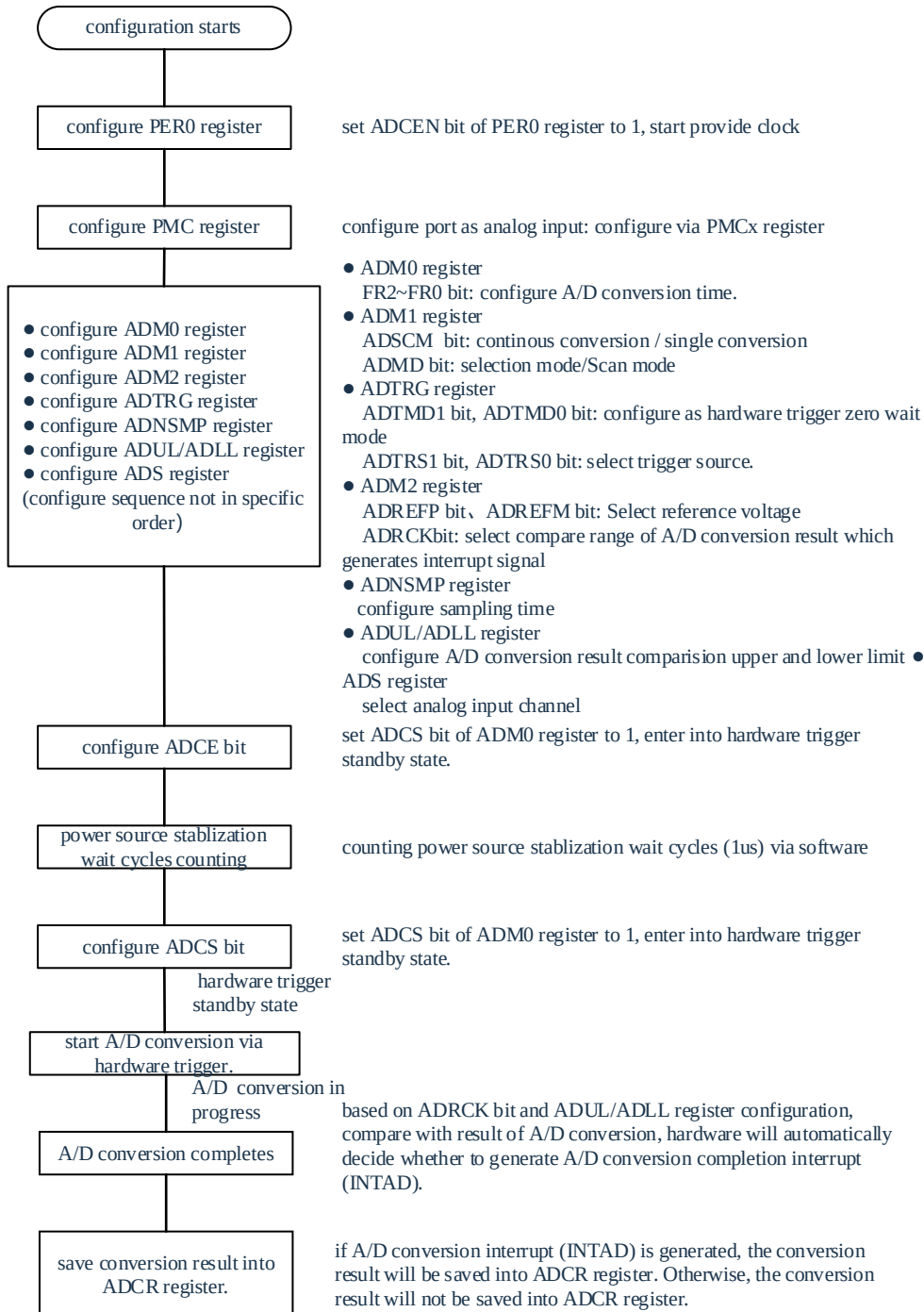
17.6.1 Software Trigger Mode Settings

Figure 17-18: Software trigger mode settings



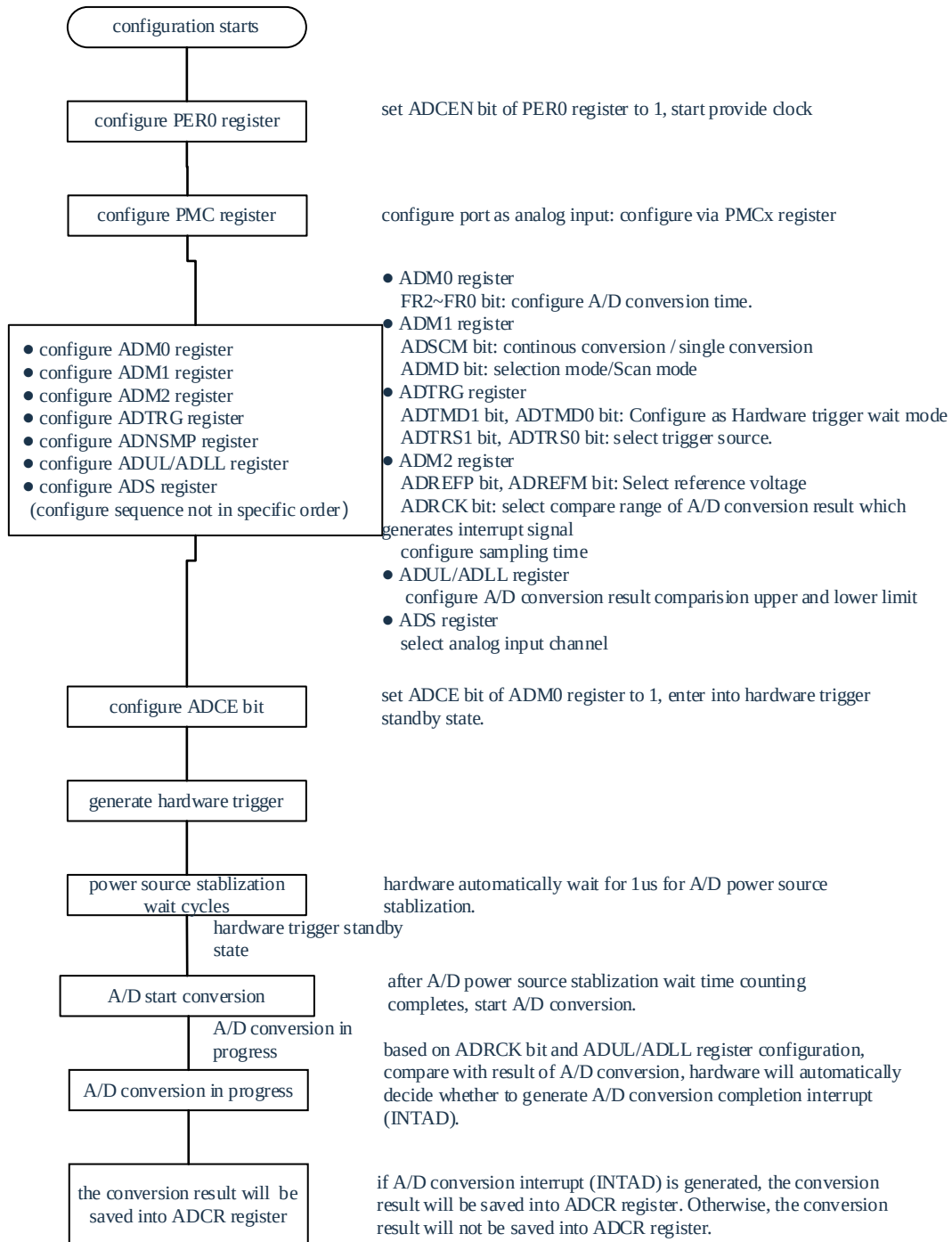
17.6.2 Hardware Trigger No-Wait Mode Settings

Figure 17-19: Hardware trigger no-wait mode settings



17.6.3 Hardware Trigger Wait Mode Settings

Figure 17-20: Hardware trigger wait mode settings



Chapter 18 Analog Comparators (ACMP0/1)

18.1 Overview

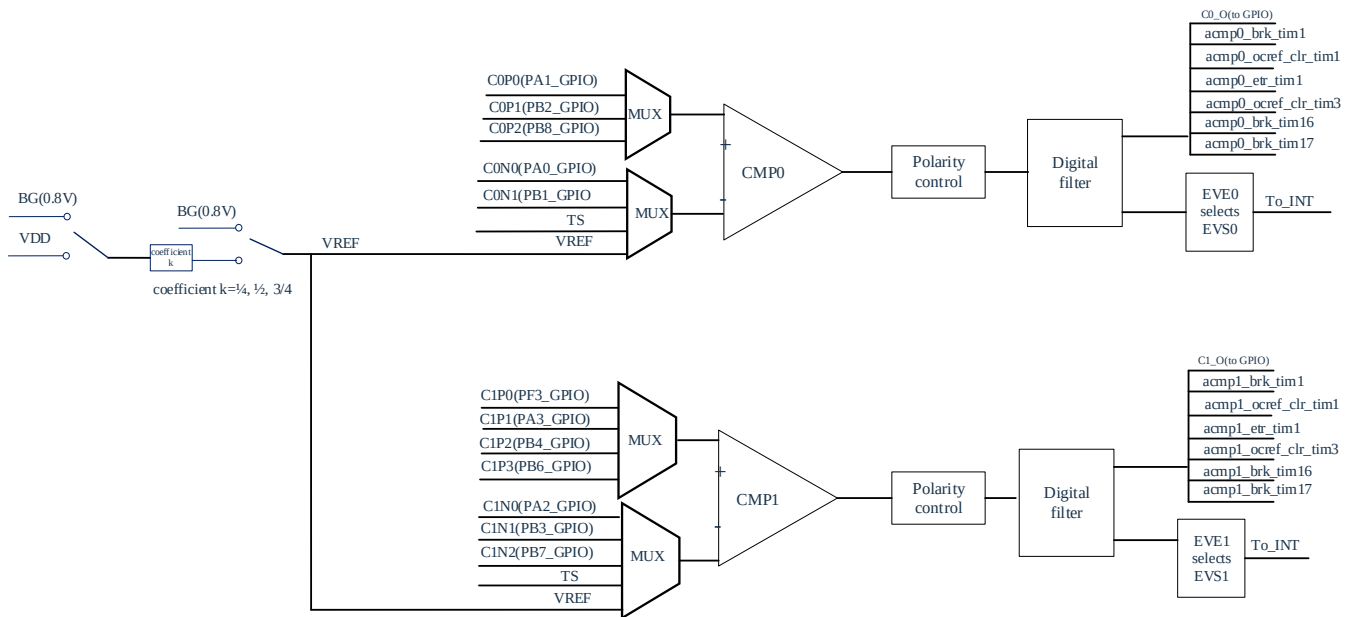
The chip contains two internal analog comparators, which can be configured for various application scenarios. When the voltage at the positive terminal is higher than that at the negative terminal, the comparator outputs logic 1, and vice versa, outputting logic 0. The output polarity can also be reversed through a selectable output polarity bit. Each comparator can generate an interrupt when the output changes.

18.2 Features

- Analog input voltage range: (0 ~ VDD) V.
- Hysteresis voltage selection: Supports 0mV/20mV.
- Multiple selectable input pins for both the positive and negative terminals of each comparator.
- Output filtering time: Can be selected from 0 to 512* T_{sys} .
- Comparator output can trigger Timer 1.
- Interrupt generation: Output changes can trigger interrupts.

18.3 Block Diagram

Figure18-1: Comparator block diagram



Note: If the filter is enabled, the following scenarios will occur:

Filter input signal width < filter set time: No signal output.

Filter input signal width = filter set time: A pulse signal with one clock cycle output.

Filter input signal width > filter set time: Normal filtered output.

18.4 Register Mapping

(ACMP base address = 0x40068200)

RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
C0CON0 _(P1B)	0x000	R/W	Analog Comparator 0 Control Register 0	0x00
C0CON1 _(P1B)	0x004	R/W	Analog Comparator 0 Control Register 1	0x00
C1CON0 _(P1B)	0x008	R/W	Analog Comparator 1 Control Register 0	0x00
C1CON1 _(P1B)	0x00C	R/W	Analog Comparator 1 Control Register 1	0x00
CEVCON _(P1B)	0x010	R/W	Analog Comparator Event Control Register	0x00
CVRCON _(P1B)	0x014	R/W	Analog Comparator Voltage Reference Control Register	0x00
ACMPLOCK	0x018	R/W	Analog Comparator Write Enable Register	0x00

Note:

The registers labeled (P1B) are protected registers.

(P1B): When ACMPLOCK == 55H, the labeled registers are writable; when it is any other value, writing is prohibited.

18.5 Register Description

18.5.1 Analog Comparator 0 Control Register 0 (C0CON0)

Bit	Symbol	Description	Reset value
31:16	-	Reserved	0x0
15	C0_EN	Analog comparator 0 enable bit 0: Disable 1: Enable	0
14	C0_OEN	Analog comparator 0 enable bit 0: Disable 1: Enable	0
13:9	-	Reserved	0x0
8	C0_OUT	Analog comparator 0 result bit (read-only)	0
7:6	-	Reserved	0
5:4	C0_P_S	Analog comparator 0 positive channel selection 00: COP0(PA1) 01: COP1(PB2) 10: COP2(PB8) 11: Disable	0x0
3	--	Reserved	0
2:0	C0_N_S	Analog comparator 0 negative channel selection 000: CON0(PA0) 001: CON1(PB1) 010: Reserved 011: TS 100: VREF Other: Disable	0x0

18.5.2 Analog Comparator 0 Control Register 1 (C0CON1)

Bit	Symbol	Description	Reset value
31:14	-	Reserved	0x0
13:12	C0_HYSPN_S	Analog comparator 0 hysteresis mode selection 00: No hysteresis 01: Positive hysteresis 10: Negative hysteresis 11: Both positive and negative hysteresis	0x0
11	-	Reserved	0
10	C0_HYSV_S	Analog comparator 0 hysteresis selection 0: No hysteresis 1: 20mV	0
9	C0_POS	Analog comparator 0 output polarity selection bit 0: Normal output 1: Inverted output	0
8	C0_FE	Analog comparator 0 output filtering enable bit 0: Disable 1: Enable	0
7:4	-	Reserved	0x0
3:0	C0_FS	Analog comparator 0 output filter time selection bit 0000: (0~1)*Tpclk 0001: (1~2)*Tpclk 0010: (2~3)*Tpclk 0011: (4~5)*Tpclk 0100: (8~9)*Tpclk 0101: (16~17)*Tpclk 0110: (32~33)*Tpclk 0111: (64~65)*Tpclk 1000: (128~129)*Tpclk 1001: (256~257)*Tpclk 1010: (512~513)*Tpclk Other: (0~1)*Tpclk	0x0

18.5.3 Analog Comparator 1 Control Register 0 (C1CON0)

Bit	Symbol	Description	Reset value
31:16	-	Reserved	-
15	C1_EN	Analog comparator 1 enable bit 0: Disable 1: Enable	0
14	C1_OEN	Analog comparator 1 output enable bit 0: Disable 1: Enable	0
13:9	-	Reserved	-
8	C1_OUT	Analog comparator 1 result bit (read-only)	0
7:6	-	Reserved	0x0
5:4	C1_P_S	Analog comparator 1 positive channel selection 00: C1P0(PF3) 01: C1P1(PA3) 10: C1P2(PB4) 11: C1P3(PB6)	0x0
3	--	Reserved	
2:0	C1_N_S	Analog comparator 1 negative channel selection 000: C1N0(PA2) 001: C1N1(PB3) 010: C1N2(PB7) 011: TS 100: VREF Other: Disable	0x0

18.5.4 Analog Comparator 1 Control Register 1 (C1CON1)

Bit	Symbol	Description	Reset value
31:14	-	Reserved	-
13:12	C1_HYSPN_S	Analog comparator 1 hysteresis mode selection 00: No hysteresis 01: Positive hysteresis 10: Negative hysteresis 11: Both positive and negative hysteresis	0x0
11	-	Reserved	-
10	C1_HYSV_S	Analog comparator 1 hysteresis selection 0: No hysteresis 1: 20mV	0
9	C1_POS	Analog comparator 1 output polarity selection bit 0: Normal output 1: Inverted output	0
8	C1_FE	Analog comparator 1 output filtering enable bit 0: Disable 1: Enable	0
7:4	-	Reserved	-
3:0	C1_FS	Analog comparator 1 output filter time selection bit 0000: (0~1)*Tpclk 0001: (1~2)*Tpclk 0010: (2~3)*Tpclk 0011: (4~5)*Tpclk 0100: (8~9)*Tpclk 0101: (16~17)*Tpclk 0110: (32~33)*Tpclk 0111: (64~65)*Tpclk 1000: (128~129)*Tpclk 1001: (256~257)*Tpclk 1010: (512~513)*Tpclk Other: (0~1)*Tpclk	0x0

18.5.5 Analog Comparator Event Control Register (CEVCON)

Bit	Symbol	Description	Reset value
31:18	-	Reserved	0
17	ACMP1_OCREFCLR INTEN_TIM3	Analog comparator 1 output as TIM3 OCREF_CLR_INT enable control bit: 0: Disable 1: Enable	0
16	ACMP0_OCREFCLR INTEN_TIM3	Analog comparator 0 output as TIM3 OCREF_CLR_INT enable control bit: 0: Disable 1: Enable	0
15	ACMP1_OCREFCLR INTEN_TIM1	Analog comparator 1 output as TIM1 OCREF_CLR_INT enable control bit: 0: Disable 1: Enable	0
14	ACMP0_OCREFCLR INTEN_TIM1	Analog comparator 0 output as TIM1 OCREF_CLR_INT enable control bit: 0: Disable 1: Enable	0
13:12	ETR_SRC_TIM1	TIM1 ETR input source selection: 00: ETR from GPIO 01: ETR from COMP0 10: ETR from COMP1 11: Disable	0
11	ACMP1_BRKEN_TIM17	Analog comparator 1 output as TIM17 break enable control bit: 0: Disable 1: Enable	0
10	ACMP0_BRKEN_TIM17	Analog comparator 0 output as TIM17 break enable control bit: 0: Disable 1: Enable	0
9	ACMP1_BRKEN_TIM16	Analog comparator 1 output as TIM16 break enable control bit: 0: Disable 1: Enable	0
8	ACMP0_BRKEN_TIM16	Analog comparator 0 output as TIM16 break enable control bit: 0: Disable 1: Enable	0
7	ACMP1_BRKEN_TIM1	Analog comparator 1 output as TIM1 break enable control bit: 0: Disable 1: Enable	0
6	ACMP0_BRKEN_TIM1	Analog comparator 0 output as TIM1 break enable control bit: 0: Disable 1: Enable	0
5:4	-	Reserved	0x0
3:2	EVS1	Analog comparator 1 event generation condition selection	0x0

		bit: 00: Comparator 1 output transition from 0 -> 1 01: Comparator 1 output transition from 1 -> 0 10: Comparator 1 output transition from 0 -> 1 or 1 -> 0 11: Reserved	
1:0	EVS0	Analog comparator 0 event generation condition selection bit: 00: Comparator 0 output transition from 0 -> 1 01: Comparator 0 output transition from 1 -> 0 10: Comparator 0 output transition from 0 -> 1 or 1 -> 0 11: Reserved	0x0

18.5.6 Analog Comparator Reference Voltage Control Register (CVRCON)

Bit	Symbol	Description	Reset value
31:6	-	Reserved	0x0
5	CNDIVS	ACMP_VREF reference source selection bit 1: Select BG(0.8V) for voltage division 0: Select VDD for voltage division	0
4	CNSVR	Comparator negative internal voltage VREF selection bit 1: Select CMP_VREF 0: Select BG(0.8V)	0
3:2	-	Reserved	0x0
1:0	CVS	Analog comparator reference voltage prescaler coefficient k selection bit 00: 1/4 01: 2/4 10: 3/4 11: Reserved	0x0

18.5.7 Analog Comparator Write Enable Register (ACMPLOCK)

Bit	Symbol	Description	Reset value
31:8	-	Reserved	0x0
7:0	ACMPLOCK[7:0]	ACMP register write enable control; 0x55: Register write enabled. Other: Register write disabled.	0x0

Chapter 19 Universal CRC

19.1 CRC Overview

The CRC (Cyclic Redundancy Check) calculation unit uses a fixed polynomial generator to produce the CRC code from a 32-bit data word.

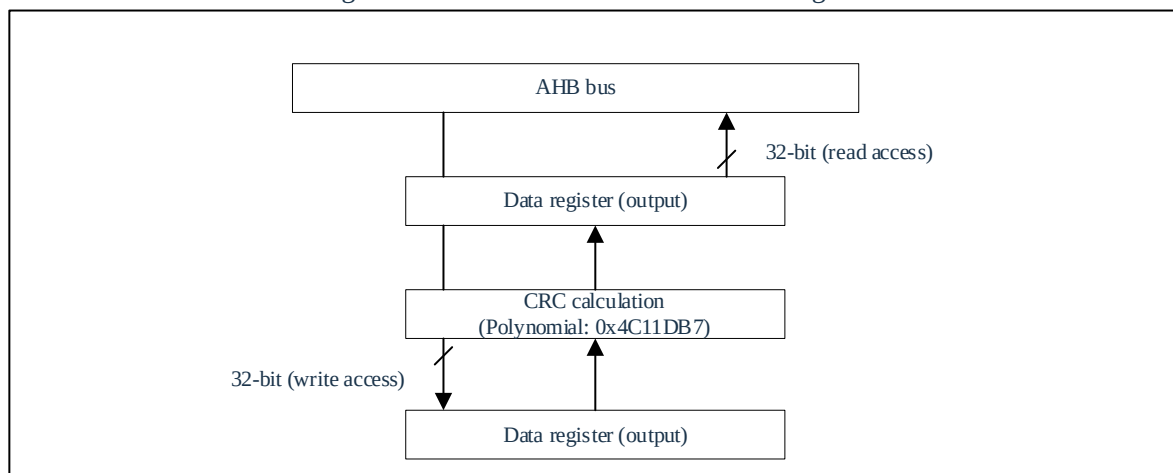
In many applications, CRC-based technology is commonly used to verify the integrity of data transfer or storage. According to the EN/IEC 60335-1 standard, these technologies provide a method for verifying Flash integrity. The CRC calculation unit helps calculate the software signature during runtime and compares it with the reference signature generated and stored in a specified storage unit at the time of linking.

19.2 CRC Features

- Uses CRC-32 (Ethernet) polynomial: 0x4C11DB7
 - $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- Single 32-bit input/output data register
- CRC calculation is completed within 4 AHB clock cycles (HCLK)
- 8-bit general-purpose register (for temporary storage)

The block diagram is shown in Figure 19-1.

Figure 19-1: CRC calculation unit block diagram



19.3 Function Description

The CRC calculation unit consists primarily of a single 32-bit data register, which:

- Acts as an input register to input new data into the CRC calculator (when writing data to the register).
- Can store the previous CRC calculation result (when reading from the register).

Each write operation to the data register performs a CRC calculation using the newly input value and the previously generated CRC value in the data register (the CRC calculation is performed on the entire 32-bit data word, not byte-by-byte).

During the CRC calculation, write operations are blocked, allowing for back-to-back write accesses or continuous write-read accesses.

The CRC calculator can be reset to 0xFFFF FFFF using the RESET control bit in the CRC_CR register. This operation does not affect the contents of the CRC_IDR register.

19.4 Register Mapping

(CRC base address = 0x4000_3000)

RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
CRC_DR	0x000	R/W	CRC Data Register	0xFFFFFFFF
CRC_IDR	0x004	R/W	CRC Independent Data Register	0x0
CRC_CR	0x008	R/W	CRC Control Register	0x0

19.5 Register Description

19.5.1 CRC Data Register (CRC_DR)

Bit	Symbol	Description	Reset value
31:0	DR	Data register bit. Used as an input register when writing new data to the CRC calculator. The previous CRC calculation result can be read from the register when reading.	0xFFFFFFFF

19.5.2 CRC Independent Data Register (CRC_IDR)

Bit	Symbol	Description	Reset value
31:8	--	Reserved, must be kept at the reset value.	0x0
7:0	IDR[7:0]	General 8-bit data register bits. Can be used as a temporary storage unit for one byte. This register is not affected by the CRC reset generated by the RESET bit in the CRC_CR register.	0x0

19.5.3 CRC Control Register (CR)

Bit	Symbol	Description	Reset value
31:1	--	Reserved, must be kept at the reset value.	0x0
0	RESET	This bit is set by software to reset the CRC calculation unit. It can only be set by software and is automatically cleared by hardware.	0x0

Chapter 20 Standby Function

20.1 Standby Function

The standby function further reduces the system's operating current, and it has two modes:

1. Sleep Mode

Sleep mode is a mode where the CPU's running clock is stopped. Before setting the sleep mode, if the high-speed on-chip oscillator or the low-speed on-chip oscillator is oscillating, the clocks will continue to oscillate. Although this mode does not reduce the operating current to the extent of deep sleep mode, it is an effective mode when you want to quickly resume processing through an interrupt request or need frequent intermittent operation.

2. Deep Sleep Mode

Deep sleep mode is a mode where the high-speed on-chip oscillator's oscillation is stopped and the entire system is halted. This can significantly reduce the CPU's operating current.

Since deep sleep mode can be released through an interrupt request, intermittent operation is also possible. However, since there is a wait time to ensure oscillator stability when exiting deep sleep mode, if immediate processing is required through an interrupt request, sleep mode must be selected.

In either mode, registers, flags, and data memory retain their state before entering the standby mode, and the output latches and output buffers of the input/output ports also maintain their state.

Note 1: When transitioning to deep sleep mode, the WFI instruction must be executed after stopping the peripheral hardware running on the system clock.

Note 2: To reduce the operating current of the A/D converter, the mode register 0 (ADM0) bit 7 (ADCS) must be cleared to 0 (ADCS conversion completed, hardware cleared to 0) before stopping the A/D conversion operation and executing the WFI instruction.

20.2 Sleep Mode

20.2.1 Sleep Mode Configuration

When the SLEEPDEEP bit of the SCR register is 0, executing the WFI instruction enters sleep mode. In sleep mode, the CPU stops functioning, but the values of internal registers are still maintained, and the peripheral modules remain in the state they were in before entering sleep mode. The state of peripheral modules, oscillators, and other components in sleep mode is shown in Table 20-1.

Regardless of whether the CPU clock before setting sleep mode is the high-speed on-chip oscillator clock or the low-speed on-chip oscillator clock, sleep mode can be set.

Caution: When the interrupt request flag is 1 (indicating an interrupt request signal), the interrupt request signal is used to exit sleep mode. Therefore, even if the WFI instruction is executed in this case, the system will not transition to sleep mode.

Table 20-1: Operation status in sleep mode

Item		Sleep mode
System clock	F _{IH}	Continues running
	F _{IL}	The operating state is set through the OSMC register and SUBCKSEL register, maintaining the configured state.
CPU		Stop running
Code Flash		Stop running
SRAM		Stop running
Port (latch)		Remain in the state before sleep mode
15-bit interval timer		Can run
TIM1		Can run
TIM3		Can run
TIM14		Can run
TIM16/17		Can run
Windowed watch dog timer		Can run
Independent watchdog timer		Can run
SPI communication unit		Can run
I ² C serial interface controller		Can run
USART		Can run
Data transfer controller (DMA)		Can run
A/D converter		Can run
Analog comparators (ACMP0/1)		Can run
Power-on reset function		Can run
Voltage detection function		Can run
External interrupts		Can run
CRC	High-speed CRC	Can run
	Universal CRC	Stop running
SFR guard		Stop running

Note 1: Stop running: Automatically stops running when shifting to sleep mode.

Note 2: F_{IH}: High-speed on-chip oscillator clock

Note 3: F_{IL}: Low-speed on-chip oscillator clock

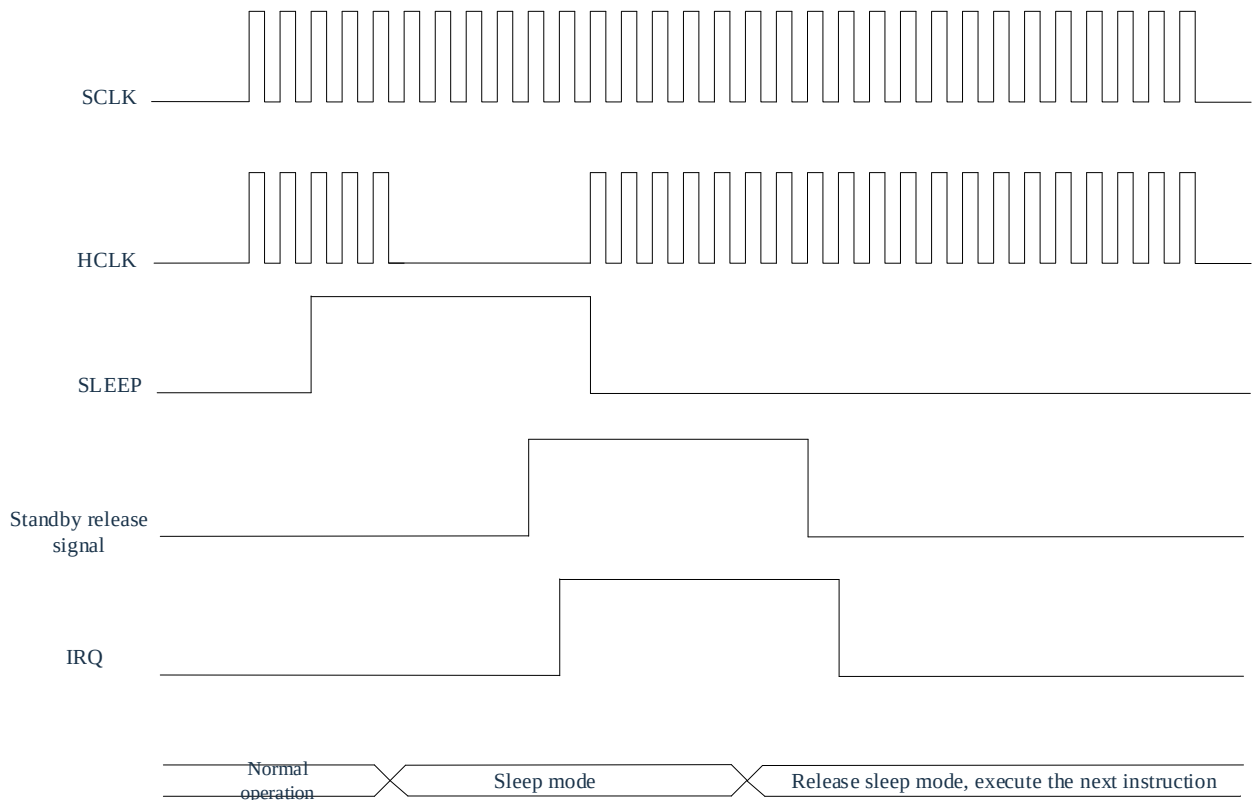
20.2.2 Release of Sleep Mode

Sleep mode can be released by the following two methods:

1. Released by interrupts

When an interrupt request is generated and the system is in a state where interrupts are enabled, sleep mode will be exited, and the CPU will begin processing the interrupt service routine.

Figure 20-1 Release sleep mode by interrupt requests



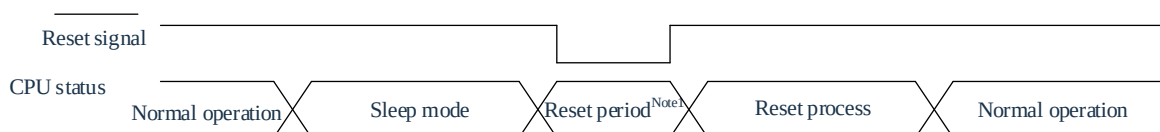
Note 1: From the generation of the standby release signal to the release of sleep mode, it takes 16 clocks to start executing the interrupt service program.

Note 2: The standby release signal cannot clear itself; you must write a register to clear it. It is usually cleared by writing a register in the interrupt service routine.

2. Released by resetting

When a reset signal is generated, the CPU is in the reset state and the sleep mode is released. As with a normal reset, the program is executed after transferring to the reset vector address.

Figure 20-2: Release sleep mode by resetting



Note 1: For the reset processing, please refer to “Chapter 22 Reset Function”. For reset processing for power-on reset (POR) circuits and voltage detection (LVD) circuits, refer to “Chapter 23 Power-on Reset Circuit”.

20.3 Deep Sleep Mode

20.3.1 Deep Sleep Mode Configuration

When the SLEEPDEEP bit of the SCR register is 1, the WFI instruction is executed and deep sleep mode is entered. In this mode, the CPU, most of the peripheral modules, and the oscillator stop running. However, the values of the CPU internal registers, the RAM data, the peripheral modules, the state of the I/O are maintained. The operating status of the peripheral module and the oscillator in deep sleep mode is shown in Table 20-2.

Caution: When the interrupt request flag is 1 (generating an interrupt request signal), the interrupt request signal is used to release deep sleep mode. Therefore, if the WFI instruction is executed in this case, it is released as soon as it enters deep sleep mode. Returns to operation mode after executing the WFI instruction and after a deep sleep mode release time has elapsed.

Table 20-2: Operation status in deep sleep mode

Item		Deep sleep mode
System clock	F _{IH}	Stop running
	F _{IL}	Set the operating state through the OSMC register and the SUBCKSEL register to maintain the configured state.
CPU		Stop running
Code Flash		Stop running
SRAM		Stop running
Port (latch)		Remain in the state before sleep mode
15-bit interval timer		Can run
TIM1		Disable running
TIM3		Disable running
TIM14		Disable running
TIM16/17		Disable running
Windowed watch dog timer		Disable running
Independent watchdog timer		Can run
SPI communication unit		Disable running
I ² C serial interface controller		Disable running
USART		Disable running
Data transfer controller (DMA)		Disable running
A/D converter		Disable running
Analog comparators (ACMP0/1)		Can run
Power-on reset function		Can run
Voltage detection function		Can run
External interrupts		Can run
CRC	High-speed CRC	Stop running
	Universal CRC	Stop running
SFR guard		Stop running

Note 1: Stop running: Automatically stops running when shifting to deep sleep mode.

Disable running: Stops running before shifting to sleep mode.

Note 2: F_{IH}: High-speed on-chip oscillator clock

Note 3: F_{IL}: Low-speed on-chip oscillator clock

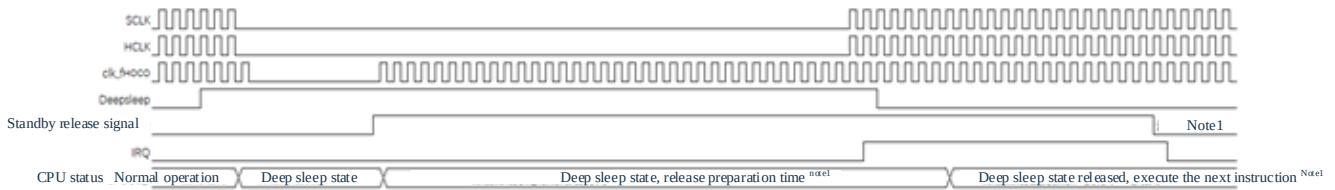
20.3.2 Release of Deep Sleep Mode

Deep sleep mode can be released in 2 ways.

1. Released by unmasked interrupt requests

If an LVD detection, INTP0-15, the 15-bit interval timer, or ACMP interrupt request occurs, the deep sleep mode will be released. After the oscillator stabilization time, if the system is in the state where interrupts are allowed, the interrupt vector will be processed. If interrupts are disabled, the next instruction will be executed.

Figure 20-3: Release deep sleep mode by interrupt requests



Note 1: Standby release signal: For details of the standby release signal, please refer to interrupt related contents.

Note 2: Deep sleep mode release preparation time: When the CPU clock is set to the high-speed on-chip oscillator before entering deep sleep mode: at least 100 μ s.

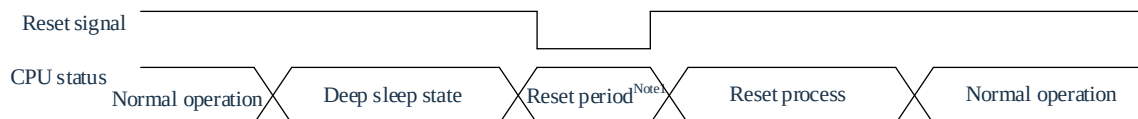
Note 3: Waiting: From when the CPU.IRQ becomes valid to the start of the interrupt service routine, 16 clock cycles are required.

Remark: The oscillation accuracy stability of the high-speed on-chip oscillator varies depending on temperature conditions and during deep sleep mode.

2. Released by generating a reset signal

The deep sleep mode is released by generating a reset signal. Then, as with a normal reset, the program is executed after transferring to the reset vector address.

Figure 20-4: Release deep sleep mode by resetting



Note 1: For reset processing, see “Chapter 22 Reset Function”. For reset processing for power-on reset (POR) circuits and voltage detection (LVD) circuits, see “Chapter 23 Power-on Reset Circuit”.

Chapter 21 Nested Vector Interrupt Controller (NVIC)

The Cortex®-M0+ CPU provides a Nested Vector Interrupt Controller (NVIC) for interrupt processing.

21.1 Features

- Supports nested vector interrupts.
- Automatically saves and restores processor state.
- Dynamically changes priorities.
- Simplifies and ensures predictable interrupt timing.

The NVIC processes all supported exceptions based on priority. All exceptions are handled in Handler Mode. NVIC supports 32 discrete interrupts (IRQ[31:0]), with each interrupt supporting 4 levels of interrupt priority. All interrupts and most system exceptions can be configured with different priorities. When an interrupt occurs, the NVIC compares the priority of the new interrupt with the current interrupt. If the new interrupt has a higher priority, it is processed immediately.

After accepting an interrupt, the start address of the interrupt service routine (ISR) can be obtained from the memory vector table. The software does not need to determine which interrupt is responded to or assign the corresponding ISR start address. After obtaining the start address, the NVIC automatically saves the processor state registers (PC, PSR, LR, R0~R3, R12) to the stack. After the ISR ends, the NVIC restores the relevant register values from the stack and resumes normal operation. This results in minimal and predictable time for handling interrupt requests.

The NVIC supports Tail-Chaining, which allows efficient handling of back-to-back interrupts without saving and restoring the current state, thus reducing the latency when switching from the current ISR to the pending ISR. The NVIC also supports Late Arrival, which improves the efficiency of concurrent interrupts. When a higher-priority interrupt request occurs before the current ISR starts execution (during the state saving and start address fetching phase), the NVIC immediately handles the higher-priority interrupt, improving real-time performance.

For more detailed information, refer to the ARM® Cortex®-M0+ Technical Reference Manual and the ARM® v6-M Architecture Reference Manual.

21.2 Exception Modes and System Interrupt Mapping

The table below lists the exception modes supported by this series of products. Like all interrupts, software can configure 4 priority levels for some of these exceptions. The highest user-configurable priority is 0, and the lowest priority is 3. The default priority for all user-configurable interrupts is 0.

Exception name	Exception number	Priority
Reset	1	-3
NMI	2	-2
Hard Fault	3	-1
Reserved	4~10	Reserved
SVCall	11	Configurable
Reserved	12~13	Reserved
PendSV	14	Configurable
SysTick	15	Configurable
Interrupt (IRQ0~IRQ31)	16~47	Configurable

Note: Priority 0 in the system corresponds to the 4th priority, ranked after the Reset, NMI, and Hard Fault system exceptions.

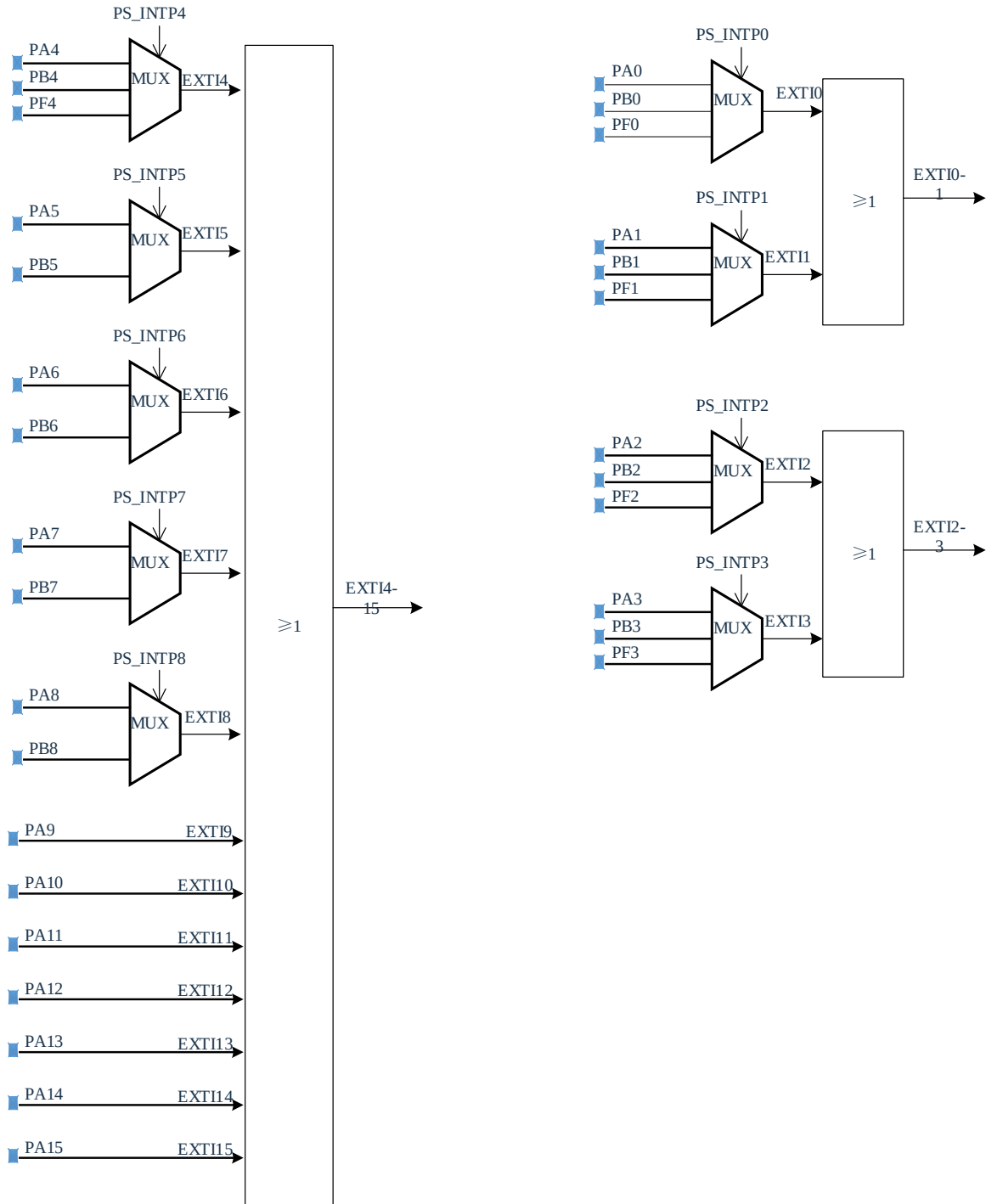
21.3 Vector Table

Table 21-1 Interrupt source overview

No.	Interrupt No.	Vector address	Exception type	Description
1-15	-	0x00-0x3c	System exceptions	
16	0	0x40	WWDG	Windowed watchdog timer interrupt
17	1	0x44	PVD	Power voltage detection interrupt
18	2	0x48	Reserved	-
19	3	0x4c	FLASH	FLASH programming completion/operation error
20	4	0x50	Reserved	-
21	5	0x54	EXT0-1	External interrupt 1_0 interrupt
22	6	0x58	EXT2-3	External interrupt 3_2 interrupt
23	7	0x5c	EXT4-15	External interrupt 15_4 interrupt
24	8	0x60	Reserved	-
25	9	0x64	DMA_0	DMA channel 0 interrupt
26	10	0x68	DMA_1	DMA channel 1 interrupt
27	11	0x6c	DMA_3	DMA channel 3 interrupt
28	12	0x70	ADC	AD conversion complete interrupt
29	13	0x74	TIM1_BRK_UP_TRIG_COM	TIM1 break, update, trigger, and communication interrupt
30	14	0x78	TIM1_CC	TIM1 capture/compare interrupt
31	15	0x7c	Reserved	-
32	16	0x80	TIM3	TIM3 global interrupt
33	17	0x84	LPTIM	Low-power timer interrupt
34	18	0x88	Reserved	-
35	19	0x8c	TIM14	TIM14 global interrupt
36	20	0x90	Reserved	-
37	21	0x94	TIM16	TIM16 global interrupt
38	22	0x98	TIM17	TIM17 global interrupt
39	23	0x9c	I2C	I2C global interrupt
40	24	0xa0	Reserved	-
41	25	0xa4	SPI1	SPI1 global interrupt
42	26	0xa8	SAU	SAU global interrupt
43	27	0xac	USART	USART global interrupt
44	28	0xb0	Reserved	-
45	29	0xb4	Reserved	-
46	30	0xb8	ACMP0	ACMP0 global interrupt
47	31	0xbc	ACMP1	ACMP1 global interrupt

21.4 External Interrupt Structure

Figure 21-1 Basic structure of interrupt function



21.5 Register Mapping

(The following register base address = 0x4004_5B30)

RO: Read-only; WO: Write-only; R/W: Read/Write.

Register	Offset value	R/W	Description	Reset value
EGP0	0x000	R/W	External Interrupt Rising Edge Enable Register	0x0
EGN0	0x001	R/W	External Interrupt Falling Edge Enable Register	0x0
EGP1	0x002	R/W	External Interrupt Rising Edge Enable Register	0x0
EGN1	0x003	R/W	External Interrupt Falling Edge Enable Register	0x0
FILCON	0x00A	R/W	External Interrupt Filter Control Register	0x0
E_IMR0	0x010	R/W	External Interrupt Mask Register 0	0x0
E_PR0	0x011	R/W	External Interrupt Pending Register 0	0x0
E_IMR1	0x012	R/W	External Interrupt Mask Register 1	0x0
E_PR1	0x013	R/W	External Interrupt Pending Register 1	0x0

21.6 Register Description

21.6.1 External Interrupt Rising Edge Enable Register (EGP0)

The EGP0 and EGN0 registers are used to set the valid edge of INTP0~INTP7. The EGP0 and EGN0 registers are set by 8-bit memory manipulation instructions.

After a reset signal is generated, the values of these registers become “00H”.

Bit	Symbol	Description	Reset value
7	EGP7	INTP7 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0
6	EGP6	INTP6 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0
5	EGP5	INTP5 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0
4	EGP4	INTP4 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0
3	EGP3	INTP3 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0
2	EGP2	INTP2 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0
1	EGP1	INTP1 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0
0	EGP0	INTP0 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0

21.6.2 External Interrupt Rising Edge Enable Register (EGP1)

The EGP1 and EGN1 registers are used to set the valid edge of INTP8~INTP15. The EGP1 and EGN1 registers are set by 8-bit memory manipulation instructions.

After a reset signal is generated, the values of these registers become “00H”.

Bit	Symbol	Description	Reset value
7	EGP15	INTP15 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0
6	EGP14	INTP14 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0
5	EGP13	INTP13 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0
4	EGP12	INTP12 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0
3	EGP11	INTP11 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0
2	EGP10	INTP10 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0
1	EGP9	INTP9 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0
0	EGP8	INTP8 external interrupt rising edge enable bit 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge	0

Note: 1. PCLK refers to the APB bus clock.

21.6.3 External Interrupt Falling Edge Enable Register (EGN0)

Bit	Symbol	Description	Reset value
7	EGN7	INTP7 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0
6	EGN6	INTP6 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0
5	EGN5	INTP5 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0
4	EGN4	INTP4 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0
3	EGN3	INTP3 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0
2	EGN2	INTP2 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0
1	EGN1	INTP1 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0
0	EGN0	INTP0 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0

21.6.4 External Interrupt Falling Edge Enable Register (EGN1)

Bit	Symbol	Description	Reset value
7	EGN15	INTP15 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0
6	EGN14	INTP14 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0
5	EGN13	INTP13 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0
4	EGN12	INTP12 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0
3	EGN11	INTP11 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0
2	EGN10	INTP10 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0
1	EGN9	INTP9 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0
0	EGN8	INTP8 external interrupt falling edge enable bit 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge	0

EGPn	EGNn	INTPn pin active edge selection (n=0~15)
0	0	Disable edge detection
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

The corresponding ports for the EGPn and EGNn bits are shown in Table 21-3.

Table 21-3 Interrupt request signals corresponding to EGPn and EGNn bits

Detection enable bit		Interrupt request signal
EGP0	EGN0	INTP0
EGP1	EGN1	INTP1
EGP2	EGN2	INTP2
EGP3	EGN3	INTP3
EGP4	EGN4	INTP4
EGP5	EGN5	INTP5
EGP6	EGN6	INTP6
EGP7	EGN7	INTP7
EGP8	EGN8	INTP8
EGP9	EGN9	INTP9
EGP10	EGN10	INTP10
EGP11	EGN11	INTP11
EGP12	EGN12	INTP12
EGP13	EGN13	INTP13
EGP14	EGN14	INTP14
EGP15	EGN15	INTP15

Note 1: If the input port used for the external interrupt function is switched to output mode, it may detect a valid edge and generate an INTPn interrupt. When switching to output mode, the port mode register (PMxx) must be set to 0 after disabling edge detection (EGPn, EGNn = 0, 0).

Note 2: For edge-detecting ports, please refer to 2.1 Port Functions.

21.6.5 External Interrupt Mask Register 0 (E_IMR0)

Bit	Symbol	Description	Reset value
7	IMR7	INTP7 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0
6	IMR6	INTP6 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0
5	IMR5	INTP5 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0
4	IMR4	INTP4 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0
3	IMR3	INTP3 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0
2	IMR2	INTP2 interrupt mask control: 0: Interrupt masked 1: Interrupt not masked	0
1	IMR1	INTP1 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0
0	IMR0	INTP0 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0

21.6.6 External Interrupt Mask Register 1 (E_IMR1)

Bit	Symbol	Description	Reset value
7	IMR15	INTP15 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0
6	IMR14	INTP14 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0
5	IMR13	INTP13 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0
4	IMR12	INTP12 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0
3	IMR11	INTP11 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0
2	IMR10	INTP10 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0
1	IMR9	INTP9 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0
0	IMR8	INTP8 interrupt mask control 0: Interrupt masked 1: Interrupt not masked	0

21.6.7 External Interrupt Pending Register 0 (E_PR0)

Bit	Symbol	Description	Reset value
7	PR7	INTP7 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0
6	PR6	INTP6 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0
5	PR5	INTP5 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0
4	PR4	INTP4 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0
3	PR3	INTP3 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0
2	PR2	INTP2 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0
1	PR1	INTP1 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0
0	PR0	INTP0 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0

21.6.8 External Interrupt Pending Register 1 (E_PR1)

Bit	Symbol	Description	Reset value
7	PR15	INTP15 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0
6	PR14	INTP14 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0
5	PR13	INTP13 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0
4	PR12	INTP12 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0
3	PR11	INTP11 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0
2	PR10	INTP10 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0
1	PR9	INTP9 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0
0	PR8	INTP8 interrupt pending flag. When a rising or falling edge interrupt is generated by hardware, this flag is set. Software writes 1 to clear the flag. 0: No interrupt request generated 1: A rising/falling edge interrupt request generated	0

21.6.9 External Interrupt Filter Control Register (FILCON)

INTP00~INTP15 share a common digital filter.

Bit	Symbol	Description	Reset value
7:4	---	Reserved, set to 0.	0x0
3:2	FILCNT	Number of filter clocks 0x0: 1 filter clock 0x1: 2 filter clocks 0x2: 3 filter clocks 0x3: 4 filter clocks	0x0
1:0	FILCLK	Filter clock frequency setting filclk 0x0: FILCLK = PCLK 0x1: FILCLK = PCLK/4 0x2: FILCLK = PCLK/16 0x3: FILCLK = PCLK/64	0x0

Chapter 22 Reset Function

There are 8 methods to generate a reset signal:

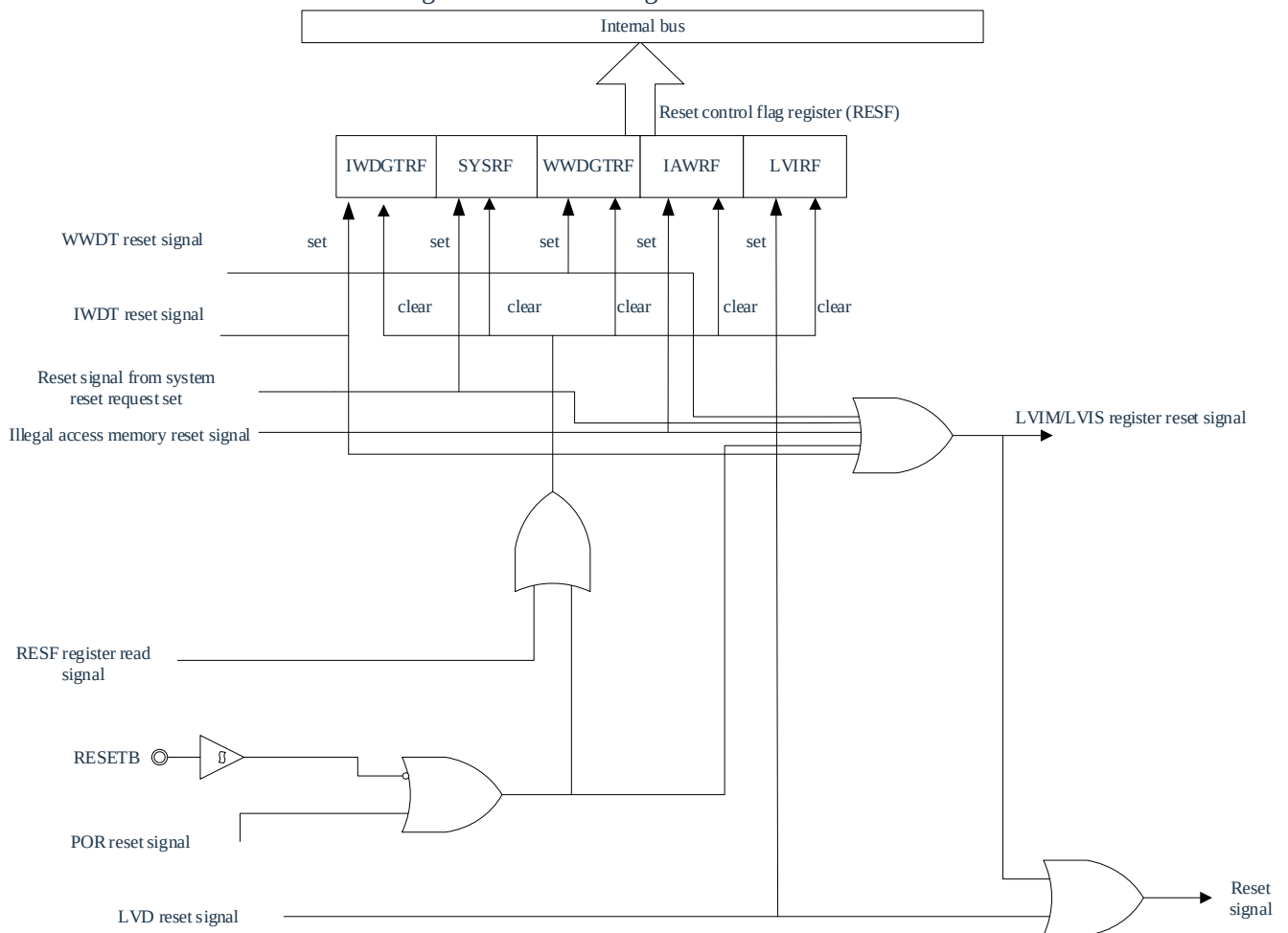
- 1) An external reset is generated by the RESETB pin input.
- 2) An internal reset is generated by the independent watchdog timer's program fault detection (IWDG).
- 3) An internal reset is generated by the window watchdog timer's program fault detection (WWDG).
- 4) An internal reset is generated by the power-on reset (POR) circuit through comparison of the supply voltage and detection voltage.
- 5) An internal reset is generated by the voltage detection circuit (LVD) through comparison of the supply voltage and detection voltage.
- 6) An internal reset is generated when the system reset request register bit (AIRCR.SYSRESETREQ) is set to 1.
- 7) An internal reset is generated by accessing illegal memory.
- 8) A TRIM positive and negative reset.

Internal and external resets are the same. After generating a reset signal, the program starts execution from the user-defined program starting address.

When a low level is input to the RESETB pin, or when the independent watchdog timer detects that the program is out of control, or when the window watchdog timer detects that the program is out of control, or when the POR circuit and LVD circuit detect a voltage issue, or when the system reset request bit is set, or when illegal memory access is detected, a reset is generated, and all hardware is in the state shown in Table 22-1.

Note 1: When performing an external reset, the RESETB pin must be input with a low level for at least 10 μ s. If an external reset is performed when the power voltage rises, the RESETB pin must be given a low level first, then the power should be turned on. After that, the low level must be maintained for at least 10 μ s within the operating voltage range specified in the AC characteristics section of the user manual, and then a high level should be input.

Internal bus



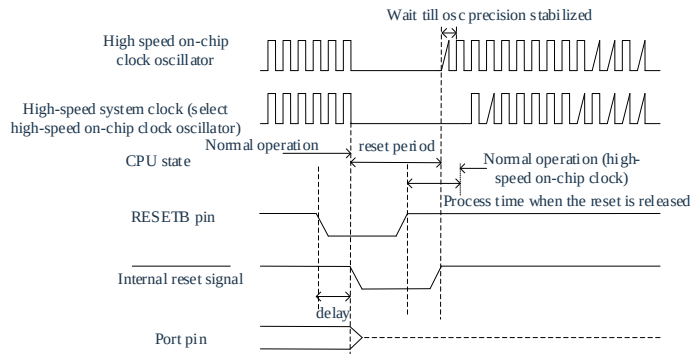
Note 2: LVIM: Voltage detection register.

Note 3: LVIS: Voltage detection level register.

22.1 Reset Timing

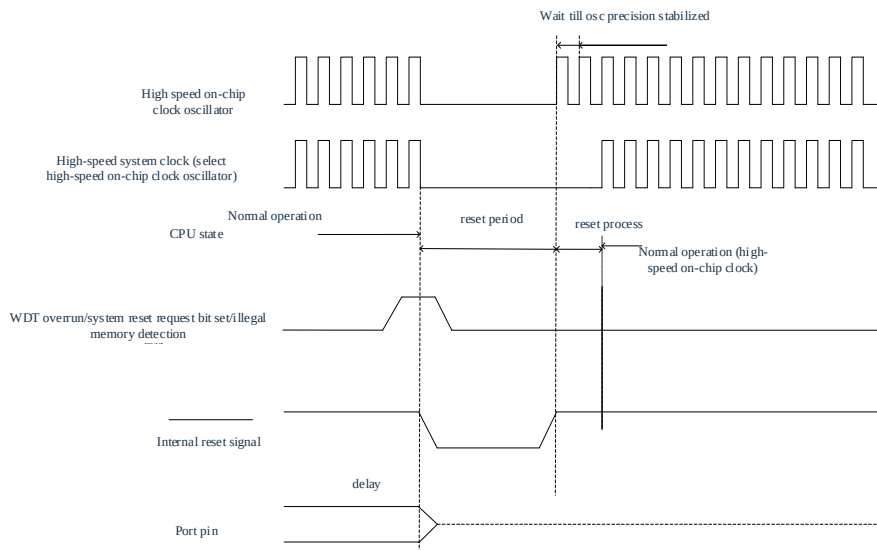
When a low level is input to the RESETB pin, a reset is generated. Then, if a high level is input to the RESETB pin, the reset state is released, and after the reset process is complete, the program starts executing with the high-speed on-chip oscillator clock.

Figure 22-2: Reset timing for RESETB input



For resets triggered by IWDG/WWDG requests, setting of the system reset request bit, TRIM positive and negative code reset, or detection of illegal memory access, the reset state is automatically released. After the reset process is complete, the program starts executing with the high-speed on-chip oscillator clock.

Figure 22-3: Reset timing triggered by IWDG/WWDG requests, setting of the system reset request bit, TRIM positive and negative code reset, or detection of illegal memory access.



Note 1: PF2, PA13, and PA14 become the following states:

- During external reset, PF2 is determined by the external input signal, PA13 is high level, and PA14 is low level.
- During POR reset, PA13 is high level, and PA14 is low level.

Note 2: The watchdog timer is also reset during an internal reset.

For resets triggered by the voltage detection of the POR circuit and LVD circuit, if after the reset, $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD}$ is satisfied, the reset state is cleared. After the reset process, the program starts executing with the high-speed on-chip oscillator clock. For details, refer to Chapter 23: Power-on Reset Circuit and Chapter 24: Voltage Detection Circuit.

Remark: V_{POR} : POR power voltage rise detection voltage

V_{LVD} : LVD detection voltage

Table 22-1: Operation status during reset

Item		Reset period						
		External reset RESINB	IWDT	POR	WWDT reset	LVD voltage detection reset	Write reset register reset	Access illegal memory reset
System clock	f_{IH}	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
	f_{IL}	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
CPU		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
Code Flash		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
SRAM		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
15-bit interval timer		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
TIM1		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
TIM3		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
TIM14		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
TIM16/17		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
WWDT		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
IWDT		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
SPI communication unit		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
I ² C serial interface controller		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
USART		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
DMA		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
A/D converter		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
Analog comparator (ACMP0/1)		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
POR		Can perform detection runs	Can perform detection runs	Can perform detection runs	Can perform detection runs	Can perform detection runs	Can perform detection runs	Can perform detection runs
LVD		Cannot run	Cannot run	Cannot run	Cannot run	Can run	Cannot run	Cannot run
External reset		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
CRC	High-speed CRC	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
	Universal CRC	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
SFR guard		Stop running	Stop running	Stop running	Stop running	Stop running	Stop running	Stop running
Power-on state after reset (option byte configuration)		Reconfigure	Reconfigure	Reconfigure	Reconfigure	Reconfigure	Reconfigure	Reconfigure

Note 1: After power-on, PA13 and PF2 are high (PF2 pin is in external input state during external reset), PA14 is low after power-on, and all other pins are in a high-impedance state before, during, and after reset.

Note 2: f_{IH} : High-speed on-chip oscillator clock; f_{IL} : Low-speed on-chip oscillator clock

22.2 Registers for Confirming the Reset Source

22.2.1 Register Mapping

(Reset control base address = 0x4002_0440) RO: Read-only; WO: Write-only; R/W: Read/Write.

Register	Offset value	R/W	Description	Reset value
RESF	0x000	RO	Reset control flag register	-

22.2.2 Reset Control Flag Register (RESF)

This microcontroller has multiple internal reset sources. The Reset Control Flag Register (RESF) stores the reset source that triggered the reset request. The RESF register can be read through an 8-bit memory manipulation instruction.

By inputting RESETB, performing a power-on reset (POR), or reading the RESF register, the SYSRF, WWDGRF, IWDGRF, OSDCRF, IAWRF, and LVIRF flags are cleared. To determine the reset source, the value of the RESF register must be saved to any RAM and then checked using the value stored in RAM.

Bit	Symbol	Description	Reset value
7	SYSRF	System reset request bit triggered internal reset request 0: No internal reset request generated, or the RESF register has been cleared. 1: An internal reset request is generated	-
6	-	Reserved	-
5	IWDGRF	Independent watchdog timer (IWDG) triggered internal reset request 0: No internal reset request generated, or the RESF register has been cleared. 1: An internal reset request is generated	-
4	WWDGRF	Windowed watchdog timer (WWDG) triggered internal reset request 0: No internal reset request generated, or the RESF register has been cleared. 1: An internal reset request is generated	-
3:2	-	Reserved	-
1	IAWRF	Illegal Memory Access triggered internal reset request 0: No internal reset request generated, or the RESF register has been cleared. 1: An internal reset request is generated	-
0	LVIRF	Low Voltage Detection (LVD) triggered internal reset request 0: No internal reset request generated, or the RESF register has been cleared. 1: An internal reset request is generated	-

Note: The status of the RESF register varies depending on the reset source. Please refer to Table 22-2.

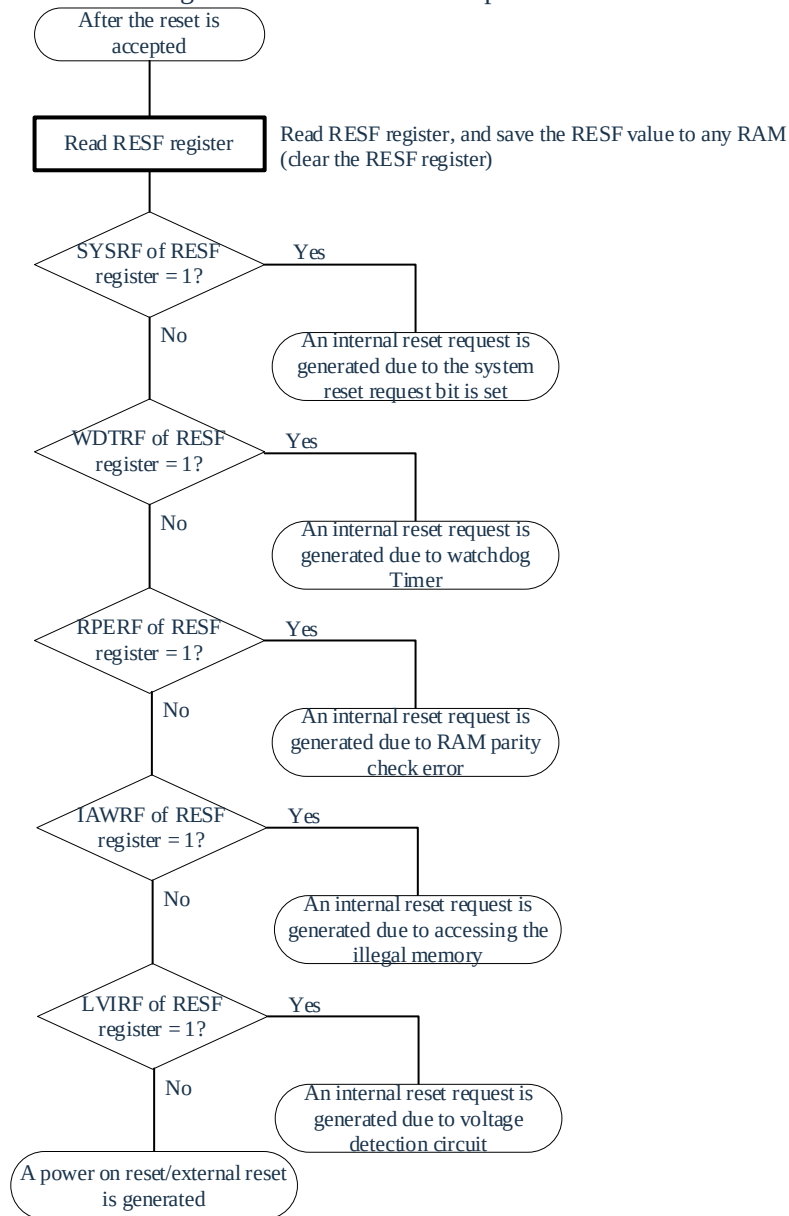
The status of the RESF register during a reset request is shown in the table below.

Table 22-2: RESF register status during a reset request

Flag \ Reset source	RESETB input	Reset generated by POR	Reset generated by system reset request bit set	Reset generated by IWDG	Reset caused by accessing illegal memory	Reset generated by LVD	Reset generated by WWDG	
SYSRF	0	0	1	Held	Held	Held	Held	
IWDGRF			Held	1	Held			
IAWRF					1	1		
LVIRF					Held			
WWDGRF					Held	1		

The confirmation steps for the reset source are shown in Figure 22-4.

Figure 22-4 Confirmation steps for reset sources



Chapter 23 Power-On Reset Circuit

23.1 Functions of Power-On Reset Circuit

The Power-On Reset (POR) circuit has the following functions:

Generates an internal reset signal when power is applied.

- If the power supply voltage (V_{DD}) exceeds the detection voltage (V_{POR}), the reset is released. However, the reset state must be maintained through voltage detection circuitry or an external reset until the power supply voltage reaches the operating voltage range specified in the datasheet's AC characteristics.
- Compare the power supply voltage (V_{DD}) with the detection voltage (V_{PDR}). When $V_{DD} < V_{PDR}$, an internal reset signal is generated. However, when the power supply voltage drops, the reset state must be set through deep sleep mode transition, voltage detection circuitry, or external reset before the power supply voltage falls below the operating voltage range specified in the datasheet's AC characteristics. Upon restarting, it must be confirmed whether the power supply voltage has recovered to the operating voltage range.

Note 1: When the power-on reset circuit generates an internal reset signal, the reset control flag register (RESF) is cleared to "00H".

Note 2: The chip has multiple hardware components that generate internal reset signals. When an internal reset signal is generated by the Window Watchdog Timer (WWDG), Independent Watchdog Timer (IWDG), Voltage Detection (LVD) circuit, System Reset Request bit, or illegal memory access, the flags indicating the reset source are allocated in the RESF register. When an internal reset signal is generated by WWDG, IWDG, LVD, the setting of the system reset request bit, or illegal memory access, the RESF register is not cleared to "00H", but the corresponding flags are set to "1". For detailed information on the RESF register, please refer to Chapter 22: Reset Function.

Note 3: V_{POR} : POR supply voltage rise detection voltage

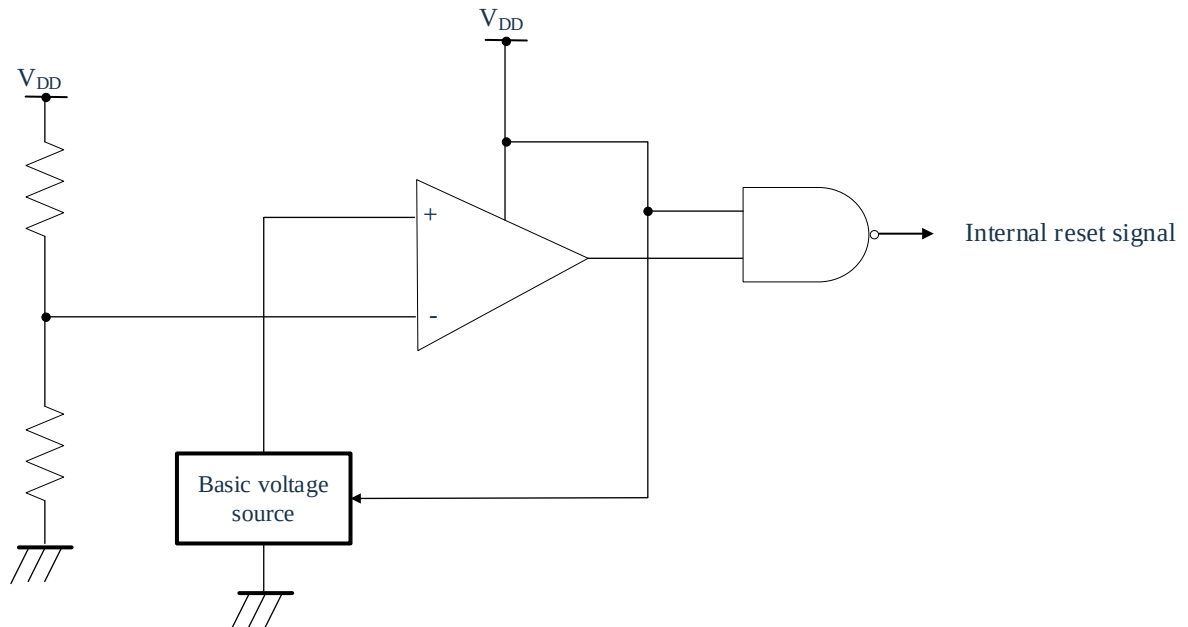
Note 4: V_{PDR} : POR supply voltage fall detection voltage

Note 5: For V_{POR} and V_{PDR} , please refer to the POR circuit characteristics in the data sheet for details.

23.2 Structure of Power-On Reset Circuit

A block diagram of the power-on reset circuit is shown in Figure 23-1.

Figure 23-1 Block diagram of power-on reset circuit

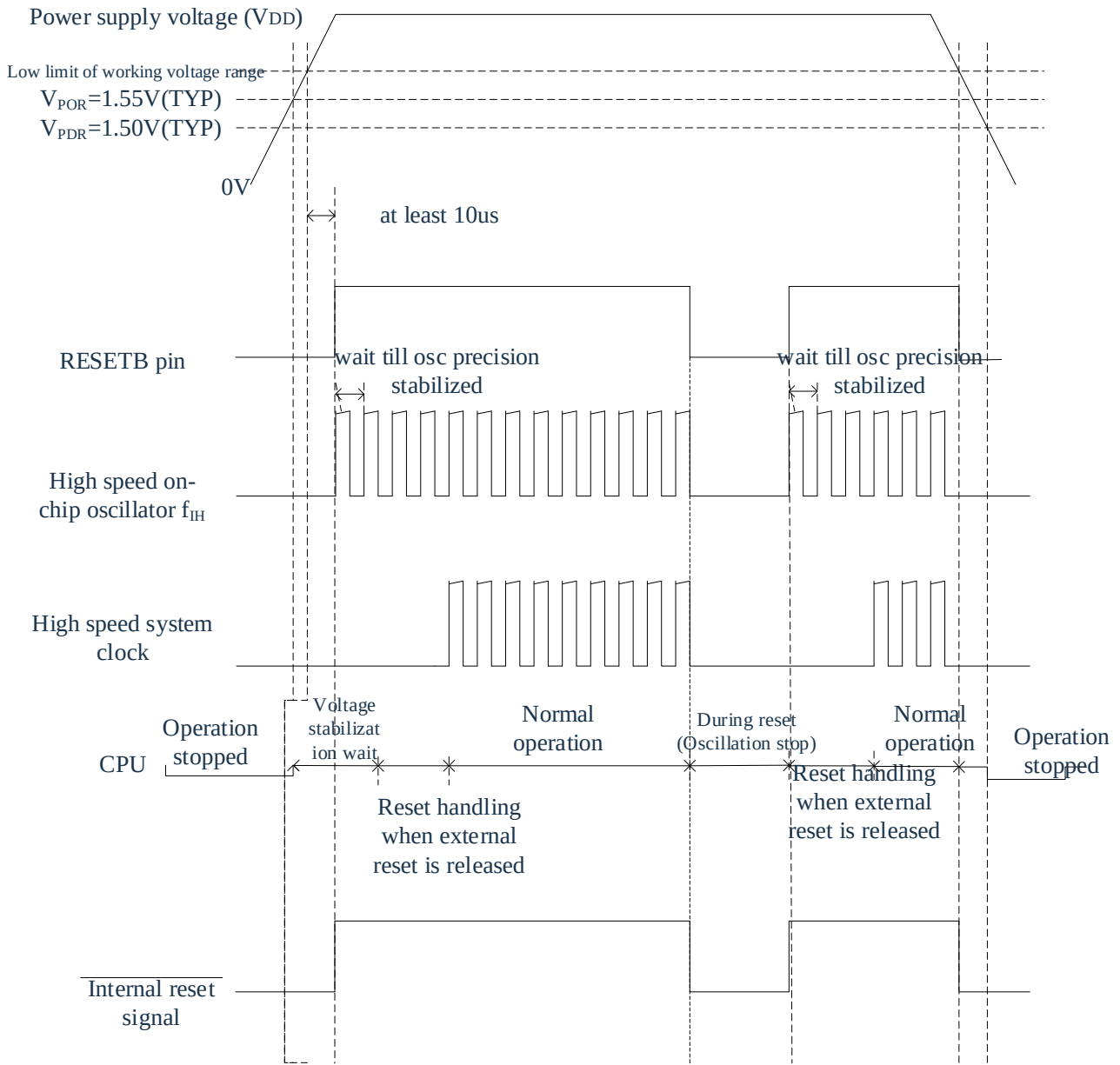


23.3 Operation of Power-On Reset Circuit

The generation timing of the internal reset signal of the power-on reset circuit and the voltage detection circuit is as follows.

Figure 23-2 Timing of internal reset signal generation for power-on reset circuit and voltage detection circuit (1/3)

1. Using an external reset input from the RESETB pin



Note 1: The internal reset processing time includes the oscillation accuracy stable waiting time of the high-speed on-chip oscillator clock.

Note 2: When the power supply voltage rises, the reset state must be maintained through an external reset until the power supply voltage reaches the operating voltage range specified in the datasheet's AC characteristics. When the power supply voltage drops, the reset state must be set through deep sleep mode transition, voltage detection circuitry, or an external reset before the power supply voltage falls below the operating voltage range. Upon restarting, it must be confirmed whether the power supply voltage has recovered to the operating voltage range.

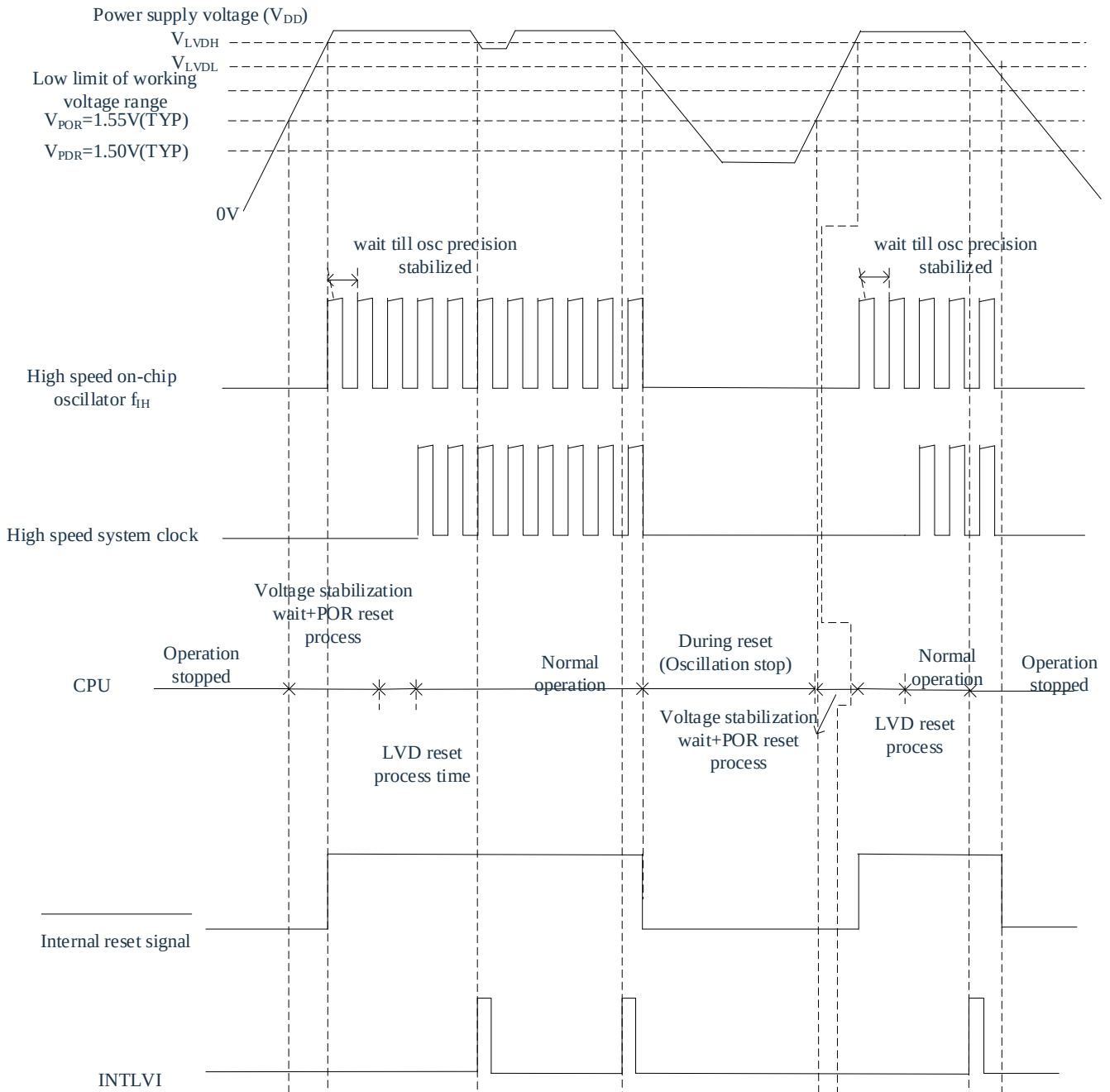
Note 3: V_{POR} : POR power supply voltage rising detection voltage

Note 4: V_{PDR} : POR power supply voltage falling detection voltage

Note 5: When LVD is OFF, an external reset using the RESETB pin must be used. For detailed information, please refer to Chapter 24: Voltage Detection Circuit.

Figure 23-2: Timing of internal reset signal generation for power-on reset circuit and voltage detection circuit (2/3)

2. LVD in interrupt & reset mode (000C1H: LVIMDS1, LVIMDS0=1, 0)



Note 1: The internal reset processing time includes the oscillation stability wait time for the high-speed on-chip oscillator clock.

Note 2: After generating the interrupt request signal (INTLVI), the LVILV bit and LVIMD bit of the Voltage Detection Level Register (LVIS) are automatically set to “1”. Therefore, it must be considered that the power supply voltage may recover to the high voltage detection voltage (V_{LVDH}) or higher while not being lower than the low voltage detection voltage (V_{LVDL}). After generating INTLVI, settings must be made according to Figure 24-5 and Figure 24-6.

Note 3: The time until normal operation begins, in addition to the “Voltage Stabilization Wait + POR Reset Processing” after reaching V_{POR} (1.55V (TYP.)), also requires the “LVD Reset Processing” after reaching the LVD detection level (V_{LVDH}).

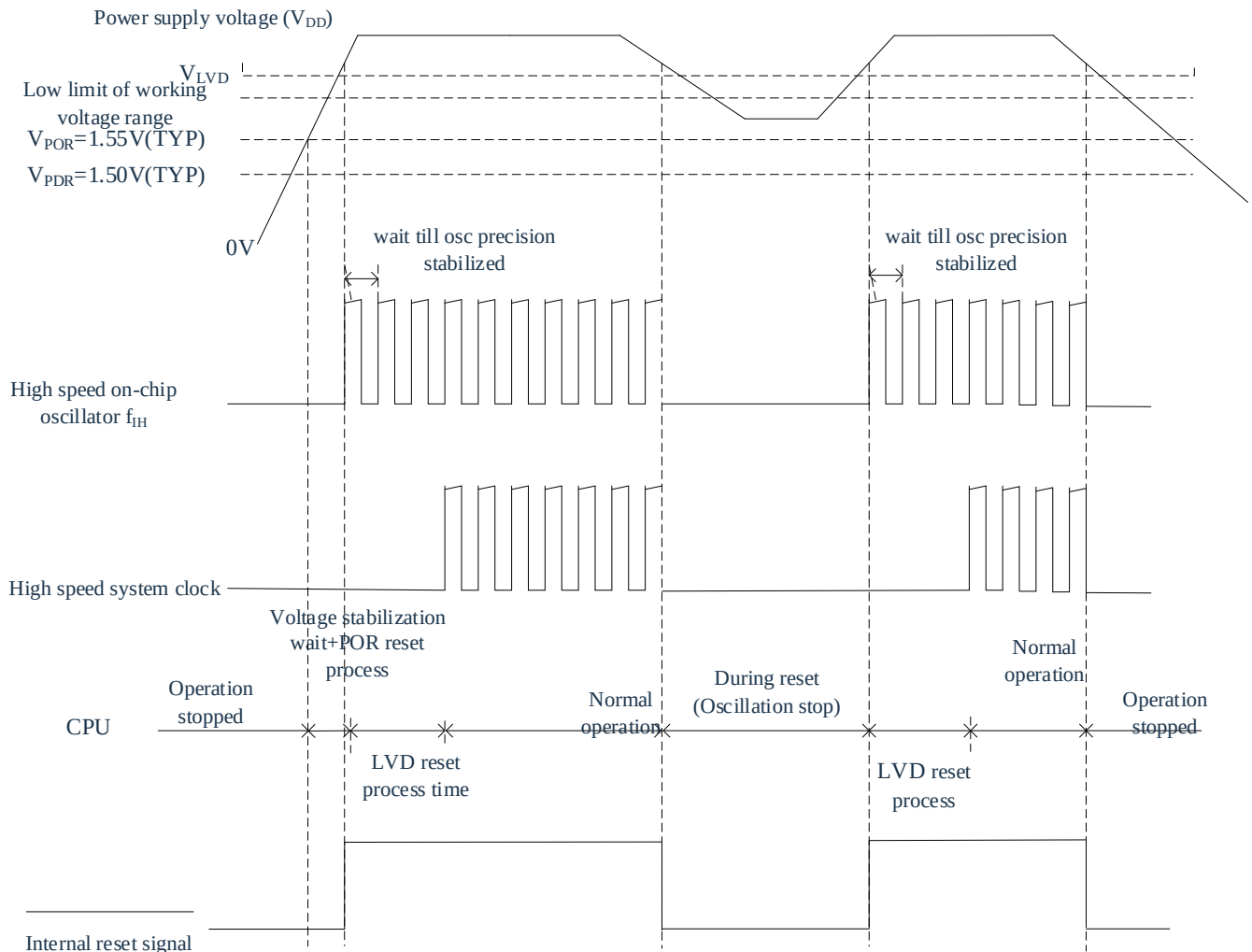
Note 4: V_{LVDH} , V_{LVDL} : LVD detection voltage

Note 5: V_{POR} : POR supply voltage rise detection voltage

Note 6: V_{PDR} : POR supply voltage drop detection voltage

Figure 23-2 Timing of internal reset signal generation for power-on reset circuit and voltage detection circuit (3/3)

3. LVD reset mode (000C1H: LVIMDS1, LVIMDS0=1, 1)



Note 1: The internal reset processing time includes the oscillation accuracy stabilization wait time of the high-speed on-chip oscillator clock.

Note 2: The time until normal operation begins requires not only the “Voltage Stabilization Wait + POR Reset Processing” after reaching V_{POR} (1.55V (TYP.)), but also the “LVD Reset Processing” after reaching the LVD detection level (V_{LVD}).

Note 3: When the power supply voltage drops, if the power is restored only after the internal reset by the voltage detection circuit (LVD), the “LVD Reset Processing” is required after reaching the LVD detection level (V_{LVD}).

Note 4: V_{LVDH} , V_{LVLDL} : LVD detection voltage

Note 5: V_{POR} : POR supply voltage rise detection voltage

Note 6: V_{PDR} : POR supply voltage drop detection voltage

Note 7: When the LVD interrupt mode is selected (option bytes 000C1H with LVIMD1, LVIMD0 = 0, 1), the time from power-on to the start of normal operation is the same as the LVD reset processing time shown in “Figure 23-2 (3/3) LVD Reset Mode.”

Chapter 24 Voltage Detection Circuit

24.1 Functions of Voltage Detection Circuit

The voltage detection circuit sets the operation mode and detection voltages (V_{LVDH} , V_{LVDL} , V_{LVD}) by the option byte (000C1H). The voltage detection (LVD) circuit has the following functions.

- Compare the supply voltage (V_{DD}) with the detection voltages (V_{LVDH} , V_{LVDL} , V_{LVD}) to generate an internal reset or internal interrupt signal.
- The detection voltage of the power supply voltage (V_{LVDH} , V_{LVDL}) can be selected from 12 detection levels using the option byte (refer to Chapter 26 Option Byte).
- Can also run in deep sleep mode
- When the supply voltage rises, the reset state must be maintained by the voltage detection circuit or external reset before the supply voltage reaches the operating voltage range shown in the AC Characteristics of the datasheet; when the supply voltage falls, it must be set to the reset state by the transfer of the deep sleep mode, the voltage detection circuit, or external reset before the supply voltage falls below the operating voltage range. The operating voltage range depends on the setting of the user option byte.

1. Interrupt & reset mode (LVIMDS1, LVIMDS0=1, 0 for option bytes)

Two detection voltages (V_{LVDH} , V_{LVDL}) are selected by the option byte 000C1H, a high voltage detection level (V_{LVDH}) for resetting release or interruption generation, and a low voltage detection level (V_{LVDL}) for reset generation.

2. Reset mode (LVIMDS1, LVIMDS0=1, 1 for option bytes)

Use the 1 detection voltage (V_{LVD}) selected by option byte 000C1H for generating or releasing the reset.

3. Interrupt mode (LVIMDS1, LVIMDS0=0, 1 for option bytes)

Use the 1 detection voltage (V_{LVD}) selected by option byte 000C1H to generate an interrupt or to release the reset. The following interrupt signals and internal reset signals are generated in each mode.

Mode	Interrupt & Reset mode	Reset mode	Interrupt mode
Configuration	LVIMDS1, LVIMDS0=1, 0	LVIMDS1, LVIMDS0=1, 1	LVIMDS1, LVIMDS0=0, 1
Operation process	Generates an interrupt request signal when the operating voltage drops and when $V_{DD} < V_{LVDH}$ is detected; Generates an internal reset when $V_{DD} < V_{LVDL}$ is detected; When $V_{DD} \geq V_{LVDH}$ is detected, the internal reset is released.	When $V_{DD} \geq V_{LVD}$ is detected, the internal reset is released; When $V_{DD} < V_{LVD}$ is detected, an internal reset is generated.	After a reset occurs, the internal reset state of the LVD is maintained until $V_{DD} \geq V_{LVD}$. When $V_{DD} \geq V_{LVD}$ is detected, the internal reset of the LVD is released. After the internal reset of the LVD is released, an interrupt request signal (INTLVI) is generated if $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ is detected.

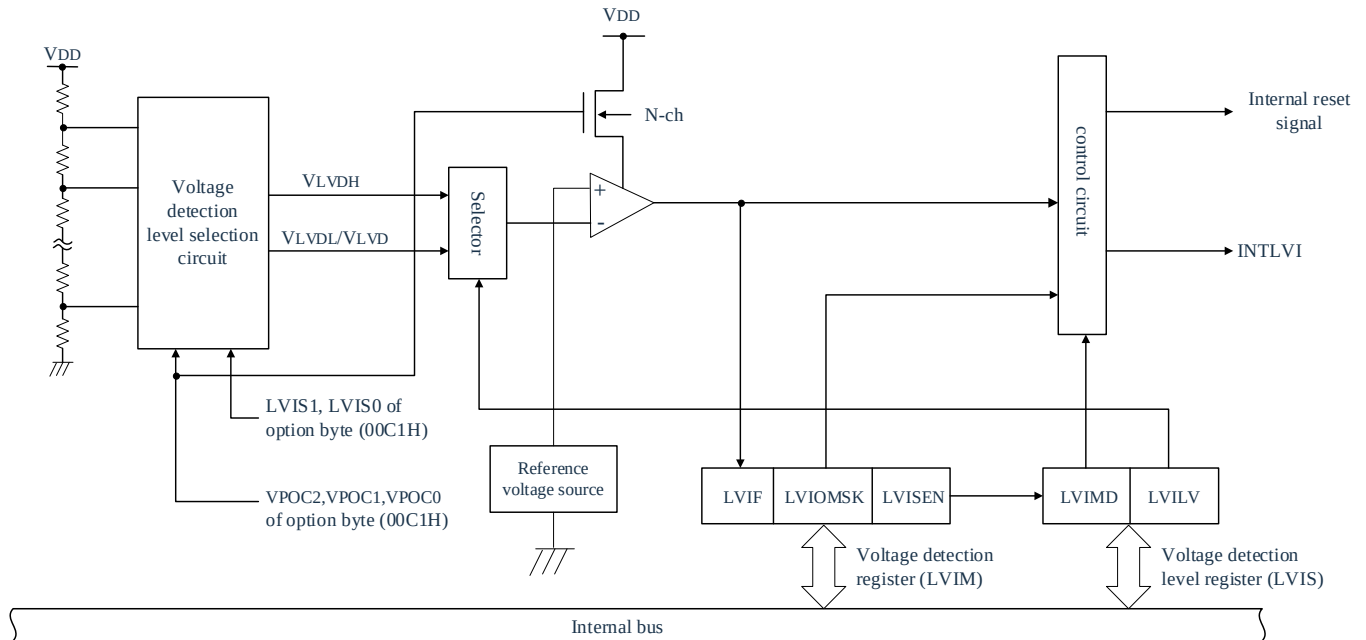
When the voltage detection circuit is operating, it can confirm whether the power supply voltage is greater than or equal to the detection voltage or less than the detection voltage by reading the voltage detection flag (LVIF: bit0 of the voltage detection register (LVIM)).

If a reset occurs, set bit0 (LVIRF) of the reset control flag register (RESF) to 1. For details of the RESF register, refer to Chapter 22 Reset Function.

24.2 Structure of Voltage Detection Circuit

The block diagram of the voltage detection circuit is shown in Figure 24-1.

Figure 24-1: Block diagram of voltage detection circuit



24.3 Register Mapping

(Base address = 0x4002_0441) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
LVIM	0x000	R/W	Voltage detection register	0x0
LVIS	0x001	R/W	Voltage detection level register	--

24.4 Registers for Controlling Voltage Detection Circuit

The voltage detection circuit is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

24.4.1 Voltage Detection Register (LVIM)

This register is set to enable or disable rewriting of the Voltage Detection Level Register (LVIS) and to confirm the masking status of the LVD outputs. The LVIM register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “00H”.

Bit	Symbol	Description	Reset value
7	LVISEN ^{Note 1}	Setting of enabled/disabled overwrite for the voltage detection level register (LVIS) 0: Disable 1: Enable	-
6:3	-	Reserved	-
2	LVD_BRKEN_TIM	LVD output as TIM1/TIM16/TIM17 brake input enable control 0: Disable 1: Enable	0
1	LVIOMSK	LVD output mask status flag 0: LVD output masking is invalid. 1: LVD output masking is valid ^{Note 2}	-
0	LVIF	Voltage detection flag 0: Supply voltage (V_{DD}) $\geq$ detection voltage (V_{LVD}) or LVD is OFF. 1: Supply voltage (V_{DD}) $<$ detection voltage (V_{LVD})	-

Note 1: The setting can only be made when the Interrupt & Reset Mode is selected (option bytes with LVIMDS1 and LVIMDS0 bits set to 1 and 0, respectively). In other modes, the initial value cannot be changed.

Note 2: The LVIOMSK bit will only automatically be set to 1 during the following periods when the Interrupt & Reset Mode is selected (option bytes with LVIMDS1 and LVIMDS0 bits set to 1 and 0, respectively), thereby masking the reset or interrupt generated by LVD.

- When LVISEN = 1
- From the occurrence of an LVD interrupt until the LVD detection voltage stabilizes
- From the change of the LVILV bit's value until the LVD detection voltage stabilizes

24.4.2 Voltage Detection Level Register (LVIS)

This is a register that sets the voltage detection level.

The LVIS register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register changes to 00H/01H/81H ^{Note1}.

Bit	Symbol	Description	Reset value
7	LVIMD ^{Note 1}	Operation mode of voltage detection 0: Interrupt mode 1: Reset mode	0
6:1	-	Reserved	-
0	LVILV ^{Note 2}	LVD detection level 0: High voltage detection level (V_{LVDH}) 1: Low voltage detection level (V_{LVDL} or V_{LVD})	0

Note 1: The reset value varies depending on the setting of the reset source and option bytes. When an LVD reset occurs, this register is not cleared to “00H”.

When a reset other than LVD occurs, the values of this register are as follows:

- Option bytes for LVIMDS1, LVIMDS0=1, 0: 00H
- Option bytes for LVIMDS1, LVIMDS0=1, 1: 81H
- Option bytes for LVIMDS1, LVIMDS0=0, 1: 01H

Note 2: Write 0 only if interrupt & reset mode is selected (LVIMDS1 bit and LVIMDS0 bits for option bytes are 1 and 0 respectively).

In other cases, it cannot be set. In interrupt & reset mode, value substitution is performed automatically by generating a reset or interrupt.

Cautions:

1. To rewrite the LVIS registers, it must be done in accordance with the steps in Figure 24-6 and Figure 24-7.
2. Select the operating mode of LVD and the detection voltage of each mode (V_{LVDH} , V_{LVDL} , V_{LVD}) by option byte 000C1H). For option byte (000C1H), refer to Chapter 26 Option Byte.

24.5 Operation of Voltage Detection Circuit

24.5.1 Reset Mode Configuration

Set the operation mode (reset mode (LVIMDS1, LVIMDS0=1, 1) and the detection voltage (V_{LVD}) by option byte 000C1H. If the reset mode is set, operation starts with the following initial settings.

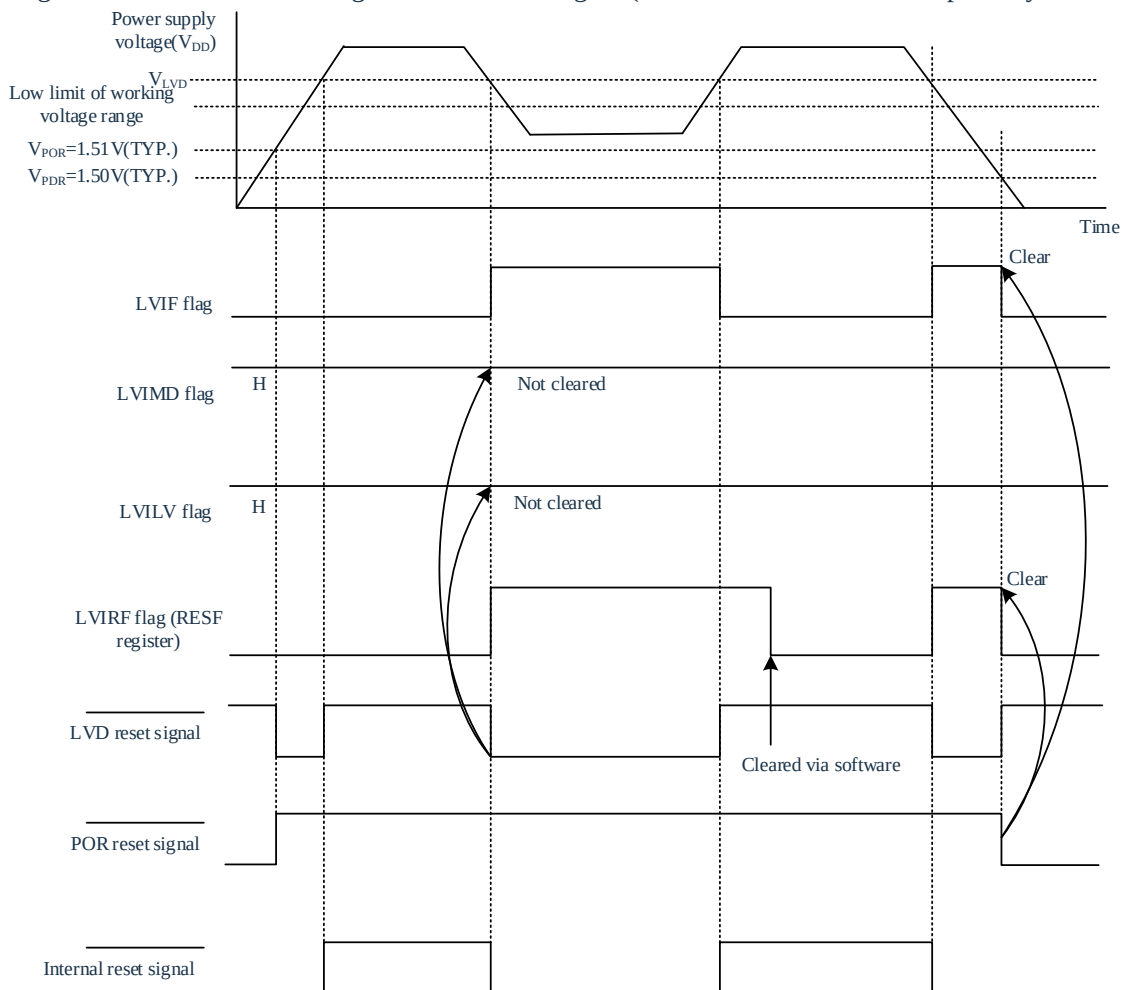
- Set the bit7 (LVISEN) of the voltage detection register (LVIM) to 0 (rewriting of the voltage detection level register (LVIS) is prohibited).
- Set the initial value of the voltage detection level register (LVIS) to 81H. The bit7 (LVIMD) is 1 (reset mode). The bit0 (LVILV) is 1 (voltage detection level: V_{LVD}).
- Operation of LVD reset mode

When the power is turned on, the reset mode (LVIMDS1, LVIMDS0 of option byte =1, 1) maintains the internal reset state of the LVD until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). If the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}), the internal reset is released.

When the operating voltage falls, an internal reset of the LVD is generated if the supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD}).

The generation timing of the internal reset signal of the LVD reset mode is shown in Figure 24-2.

Figure 24-2: Generation timing of internal reset signal (LVIMDS1, LVIMDS0 of option bytes =1, 1)



Note 1: V_{POR} : POR supply voltage rise detection voltage

Note 2: V_{PDR} : POR supply voltage fall detection voltage

24.6 Interrupt Mode Configuration

. The operation mode (interrupt mode (LVIMDS1, LVIMDS0=0, 1)) and detection voltage (V_{LVD}) are set by the option byte 000C1H. If the interrupt mode is set, operation starts in the state of the following initial settings.

- Set the bit7 (LVISEN) of the voltage detection register (LVIM) to 0 (rewriting of the voltage detection level register (LVIS) is prohibited).
- Set the initial value of the voltage detection level register (LVIS) to 01H. The bit7 (LVIMD) is 0 (interrupt mode). The bit0 (LVILV) is 1 (voltage detection level: V_{LVD}).
- Operation of LVD interrupt mode

After generating a reset, the interrupt mode (LVIMDS1, LVIMDS0 of the option byte = 0, 1) maintains the LVD's internal reset state until the supply voltage (VDD) exceeds the voltage detection level (V_{LVD}). If the supply voltage (VDD) exceeds the voltage detection level (V_{LVD}), the internal reset of the LVD is released.

The interrupt request signal (INTLVI) for the LVD is generated if the supply voltage (VDD) exceeds the voltage detection level (V_{LVD}) after the internal reset of the LVD is released. When the operating voltage drops, it must be set to the reset state by a deep sleep mode transfer or external reset before the operating voltage falls below the operating voltage range shown in the AC characteristics of the datasheet. Upon restarting operation, it must be verified that the supply voltage returns to the operating voltage range.

The generation timing of the interrupt request signal of the LVD interrupt mode is shown in Figure 24-3.

Power supply voltage (V_{DD})

V_{LVD}

Low limit of working voltage range

$V_{POR}=1.51\text{V(TYP.)}$

$V_{PDR}=1.50\text{V(TYP.)}$

Time

LVIMK flag (mask interrupt) (set by software)

Note1

H

cleared via software

LVIF flag

cleared

LVIMD flag

LVILV flag

H

INTLVI

LVIF flag

LVD reset signal

POR reset signal

Internal reset signal

V_{PDR} : POR power supply voltage fall detection voltage

24.7 Interrupt & Reset Mode Configuration

The operation mode (interrupt & reset mode (LVIMDS1, LVIMDS0=1, 0)) and detection voltage (V_{LVDH} , V_{LVDL}) are set by option byte 000C1H.

If the interrupt & reset mode is set, operation starts in the state of the following initial settings.

- Set the bit7 (LVISEN) of the voltage detection register (LVIM) to 0 (rewriting of the voltage detection level register (LVIS) is prohibited).
- Set the initial value of the voltage detection level register (LVIS) to “00H”. The bit7 (LVIMD) is 0 (interrupt mode). The bit0 (LVILV) is 0 (high voltage detection level: V_{LVDH}).
- LVD interrupt & reset mode operation

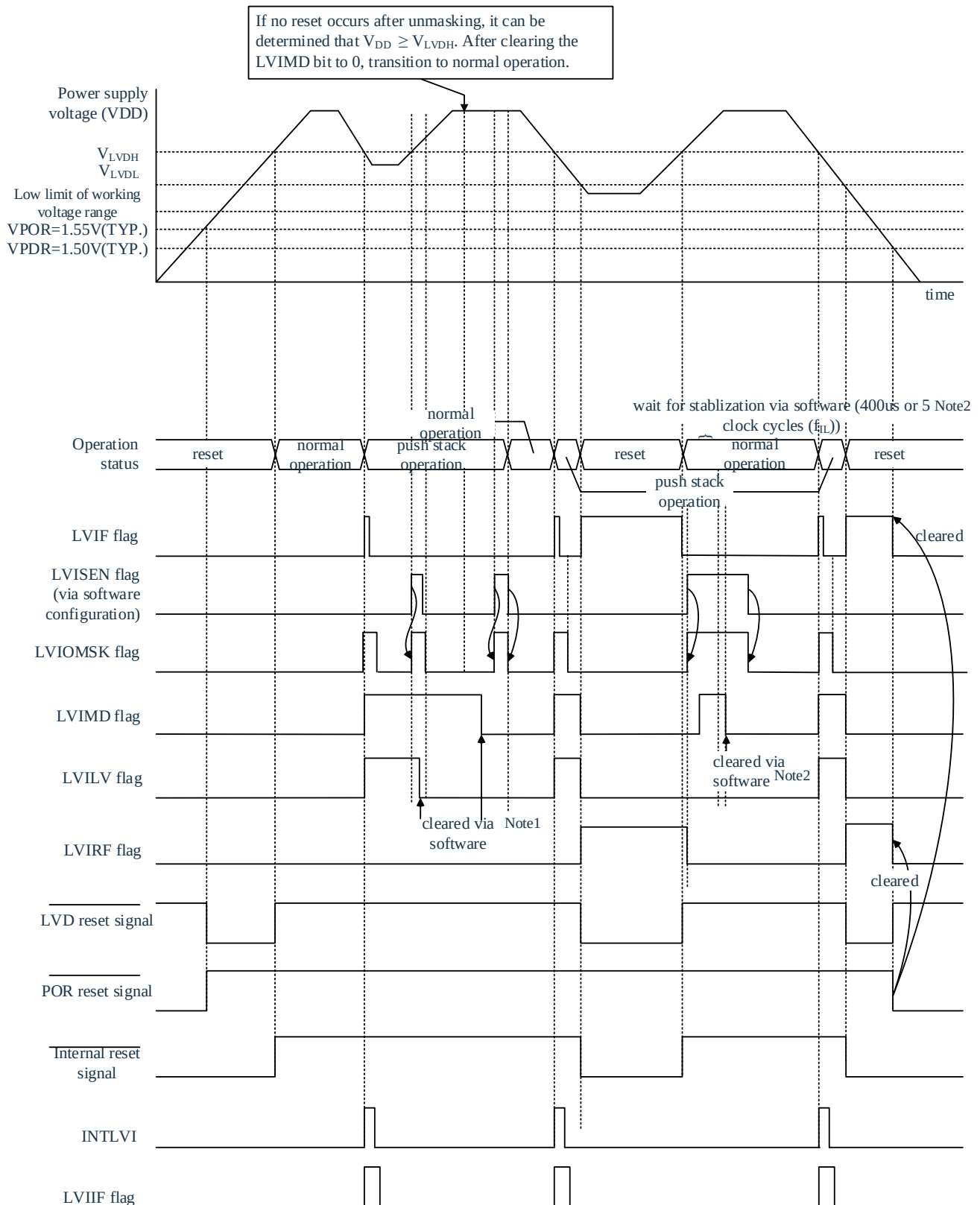
When the power is turned on, the interrupt & reset mode (LVIMDS1, LVIMDS0 of the option bytes =1, 0) maintains the internal reset state of the LVD until the supply voltage (VDD) exceeds the high voltage detection level (V_{LVDH}). If the supply voltage (VDD) exceeds the high voltage detection level (V_{LVDH}), the internal reset is released.

When the operating voltage drops, if the supply voltage (VDD) falls below the high voltage detection level (V_{LVDH}), the LVD's interrupt request signal (INTLVI) is generated and arbitrary stack processing can be performed. Thereafter, if the supply voltage (VDD) falls below the low voltage detection level (V_{LVDL}), an internal reset of the LVD is generated. However, after the occurrence of INTLVI, an interrupt request signal is not generated even if the power supply voltage (VDD) recovers to the high voltage detection voltage (V_{LVDH}) or higher in a state not lower than the low voltage detection voltage (V_{LVDL}).

When using the LVD interrupt & reset mode, you must follow the steps in the flowcharts shown in Figure 24-6: Setting procedure for confirmation/reset of operating voltage and Figure 24-7: Initial setting procedure for interrupt & reset mode.

The timing of the generation of the internal reset and interrupt signals for the LVD interrupt & reset mode is shown in Figure 24-4.

Figure 24-4 Timing of reset & interrupt signal generation (LVIMDS1, LVIMDS0 of option bytes =1, 0) (1/2)



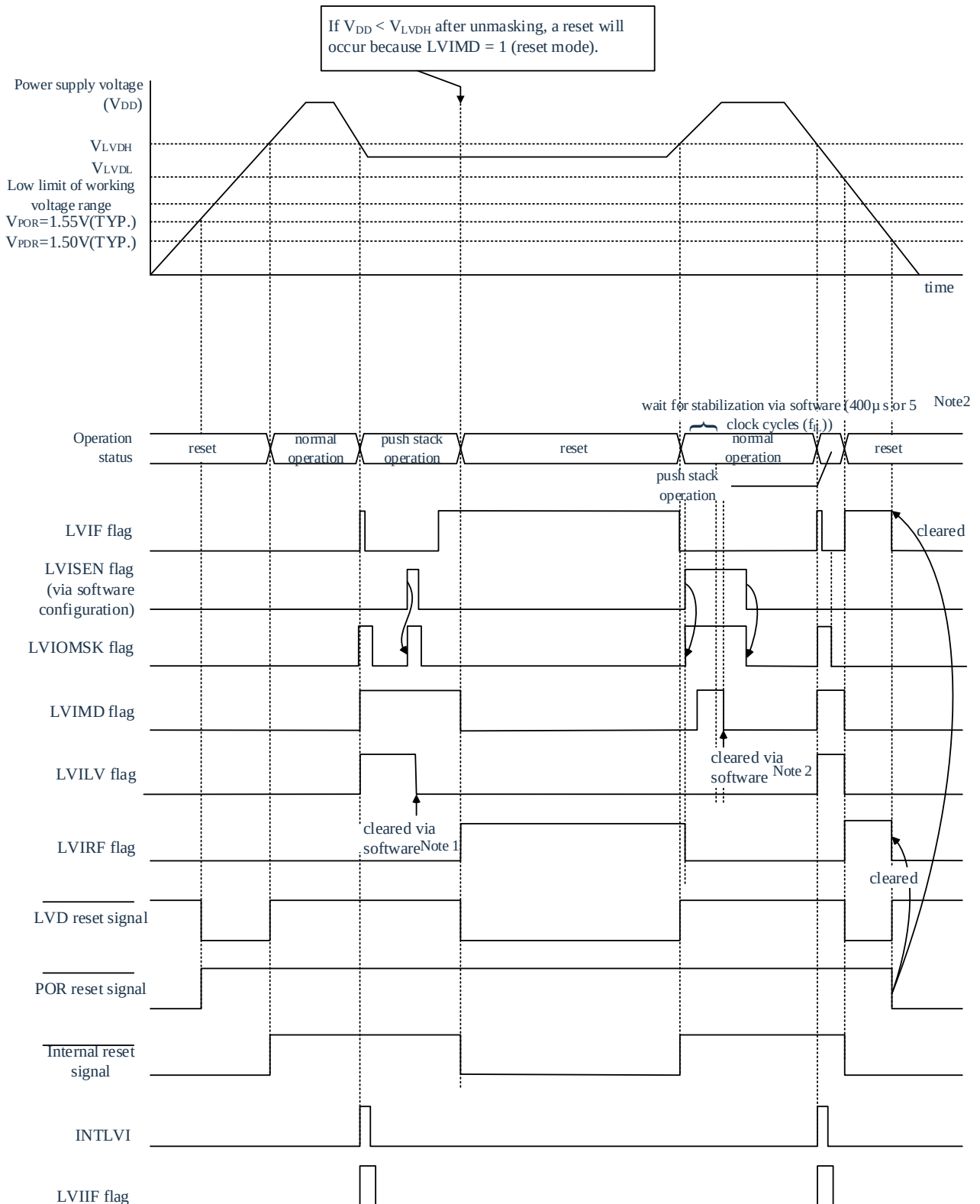
Note 1: When using interrupt & reset mode, the settings must be made after an interrupt occurs according to Figure 24-6.

Note 2: When in use & reset mode, the settings must be made according to Figure 24-7 after releasing the reset.

Note 3: V_{POR} : POR supply voltage rise detection voltage

V_{PDR} : POR supply voltage fall detection voltage

Figure 24-5 Timing of reset & interrupt signal generation (LVIMDS1, LVIMDS0 of option bytes =1, 0) (2/2)



Note 1: When using interrupt & reset mode, the settings must be made after an interrupt occurs according to Figure 24-6.

Note 2: When using interrupt & reset mode, the settings must be made according to Figure 24-7 after releasing the reset.

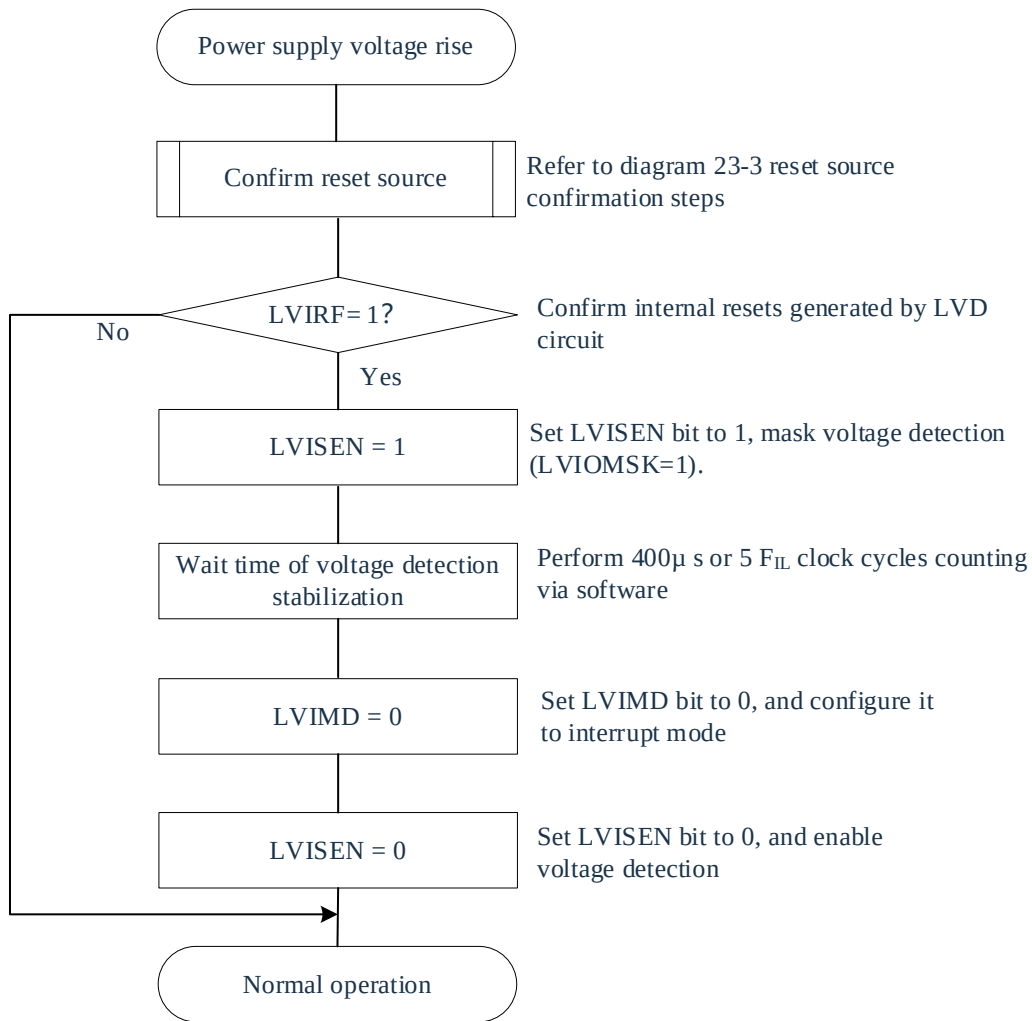
Note 3: V_{POR} : POR supply voltage rise detection voltage

V_{PDR} : POR supply voltage fall detection voltage

```
graph TD
    Start([INTLVI occurs]) --> Stacking[Stacking]
    Stacking --> LVISEN1[LVISEN = 1]
    LVISEN1 --> LVILV0[LVILV = 0]
    LVILV0 --> LVISEN0[LVISEN = 0]
    LVISEN0 --> LVIOMSK{LVIOMSK = 0}
    LVIOMSK -- No --> ResetGen([reset generation])
    LVIOMSK -- Yes --> LVDReset{LVD reset occurs}
    LVDReset -- No --> LVISEN1_2[LVISEN = 1]
    LVISEN1_2 --> LVIMD0[LVIMD = 0]
    LVIMD0 --> LVISEN0_2[LVISEN = 0]
    LVISEN0_2 --> NormalOp([Normal operation])
    LVDReset -- Yes --> ResetGen
```

The initial setup procedure for the interrupt & reset mode is shown in Figure 24-7.

Figure 24-7 Initial setting procedure for interrupt & reset mode



Remark: f_{IL} : Low-speed on-chip oscillator clock frequency

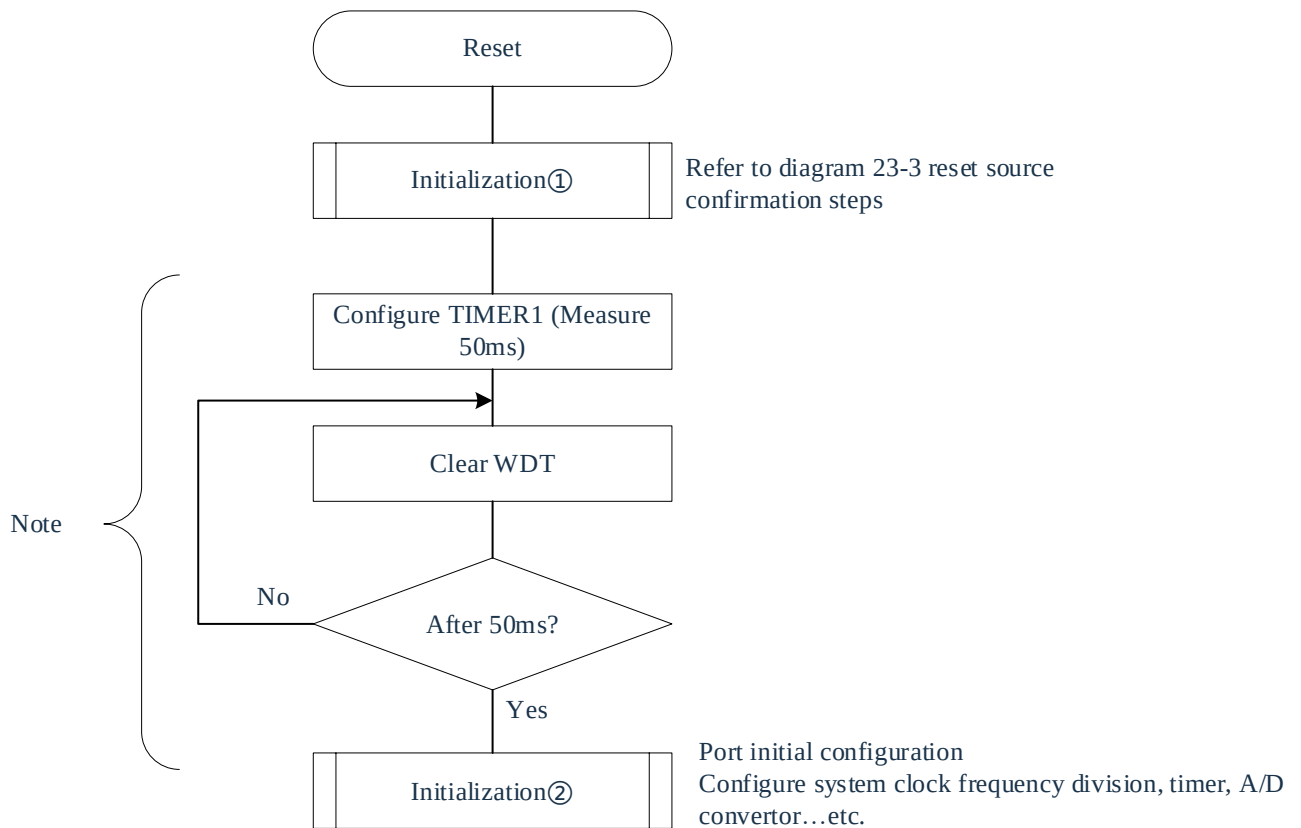
24.8 Cautions for Voltage Detection Circuit

1. Voltage fluctuations at power-on

For systems where the power supply voltage (V_{DD}) fluctuates for a certain period of time in the vicinity of the LVD detection voltage, it is possible to repeatedly enter the reset state and the reset release state. The time from reset release to the start of microcontroller operation can be arbitrarily set by the following processing.

<Processing> After releasing the reset, the initial setting of ports, etc. must be performed by using the software counter of the timer after waiting for the power supply voltage fluctuation time that varies for each system.

Figure 24-8: Example of software processing when the power supply voltage near the LVD detection voltage does not fluctuate for more than 50ms

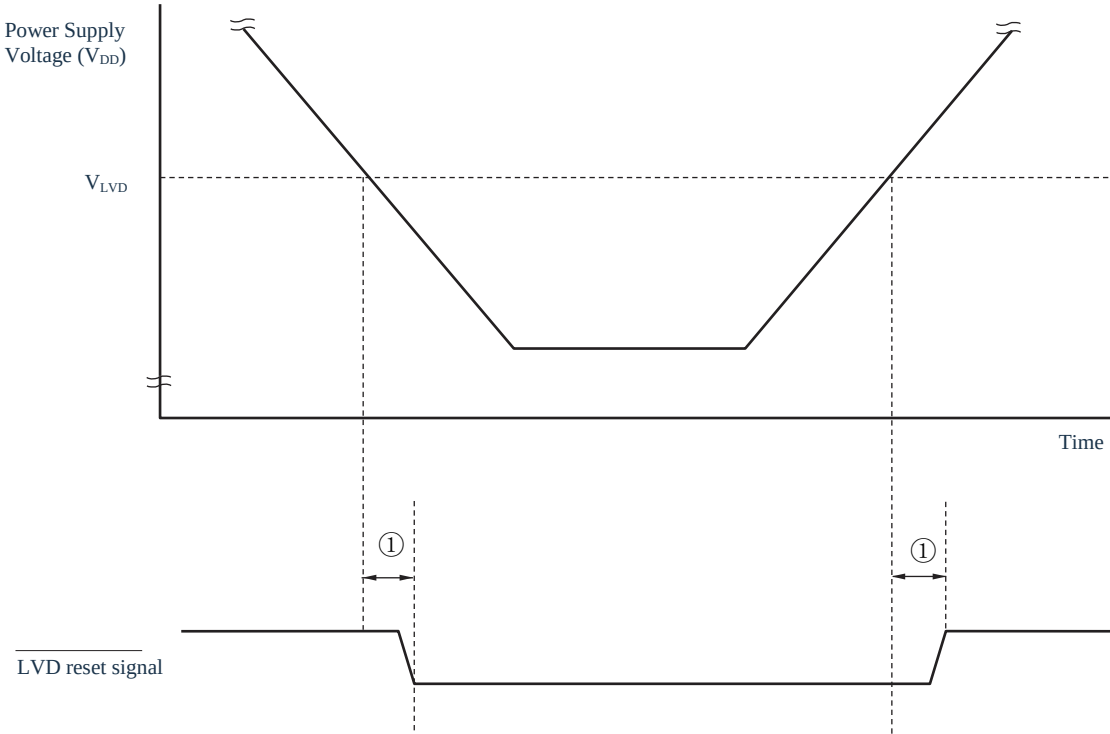


Note 1: If a reset occurs again during this period, it is not transferred to initialization processing ②.

2. Delay from generating the LVD reset source to generating or releasing the LVD reset

A delay occurs from the time the supply voltage (V_{DD}) < LVD detection voltage (V_{LVD}) is met to the time the LVD reset is generated. Similarly, a delay occurs from the time the LVD detection voltage (V_{LVD}) $\leq$ the supply voltage (V_{DD}) to the time the LVD reset is released (see figure Figure 24-9).

Figure 24-9: Delay from generation of LVD reset source to generation or release of LVD reset



① : Detection delay (300s μ s (MAX.))

3. When the power is turned on with LVD set to OFF

When LVD is set to OFF, an external reset of the RESETB pin must be used.

When performing an external reset, a low level of at least 10 μ s must be input to the RESETB pin. If an external reset is performed while the supply voltage is rising, the power must be turned on after inputting a low level to the RESETB pin, and it must be held low for at least 10 μ s over the operating voltage range shown in the AC Characteristics of the datasheet, and then input a high level.

4. Operating voltage drops when LVD is set to OFF and LVD is in interrupt mode.

If the operating voltage drops while the LVD is set to OFF and the LVD interrupt mode is set, it must be set to the reset state by a deep sleep mode transfer or external reset before the operating voltage falls below the operating voltage range shown in the AC characteristics of the datasheet. When restarting operation, it must be verified that the supply voltage returns to the operating voltage range.

Chapter 25 Safety Functions

25.1 Overview of Safety Functions

To comply with the IEC60730 safety standard, this chip incorporates the following safety features.

The purpose of these safety features is to safely stop operation through the microcontroller's self-diagnosis when a fault is detected.

1. Flash memory CRC function (high-speed CRC)

The data error of flash memory is detected by CRC operation.

- “High-speed CRC”...During the initialization program, it is possible to stop the CPU and check the entire code flash area at high speed.

2. SFR guard function

Prevents rewriting of SFR due to uncontrolled CPU.

3. A/D test function

It can perform a self-test of the A/D converter through A/D conversions using the positive (+) reference voltage, negative (–) reference voltage, analog input channel (ANI), and the internal reference voltage output of the A/D converter.

4. Digital output signal level detection function for input/output ports

When the input/output port is in output mode, the output level of the pin can be read.

25.2 Register Mapping

The following registers are used for the safety functions.

Register name	Function
• Flash memory CRC control register (CRC0CTL) • Flash memory CRC operation result register (PGCRCL)	Flash memory CRC operation function (High-speed CRC)
Special SFR guard control register (SFRGD)	SFR guard function
A/D test register (ADTES)	A/D test function
Port mode select register (PMS)	Digital output signal level detection function on input/output pins

(Flash memory CRC base address = 0x4002_1810) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
CRC0CTL	0x000	R/W	Flash memory CRC control register	0x0
PGCRCL	0x002	R/W	Flash memory CRC operation result register	0x0

(SFR base address = 0x4004_0478) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
SFRGD	0x000	R/W	SFR guard control register	0x0

(Port control base address = 0x4004_087B) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
PMS	0x000	R/W	Port mode select register	0x0

(UID base address = 0x0050_0894) RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Offset value	R/W	Description	Reset value
UID0	0x000	RO	Product unique ID bits [31:0]	-
UID1	0X004	RO	Product unique ID bits [63:32]	-
UID2	0X008	RO	Product unique ID bits [95:64]	-
UID3	0X00C	RO	Product unique ID bits [127:96]	-

25.3 Operation of Safety Functions

25.3.1 Flash Memory CRC Operation Function (High-Speed CRC)

The IEC60730 standard requires verification of the data in the flash memory and recommends CRC as a means of verification. This high-speed CRC can check the entire code flash area in the initial setup (initialization) procedure.

The high-speed CRC stops the CPU and reads 32 bits of data from the flash memory with one clock for operation. Therefore, it is characterized by a short time to complete the verification.

The CRC generation polynomial corresponds to the CRC-16-CCITT “ $X^{16} + X^{12} + X^5 + 1$ ”.

The MSB of bit31→bit0 is operated first.

25.3.1.1 Flash Memory CRC Control Register (CRC0CTL)

This is a register that sets the operational control and operational range of the high-speed CRC operator. The CRC0CTL register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register changes to “00H”.

Bit	Symbol	Description	Reset value
7	CRC0EN	Control of high-speed CRC operation 0: Stop the operation. 1: Start the operation by executing the WFE instruction.	0
6	-	Reserved	0
5:0	FEA	High-speed CRC operation range 00000: 00000H ~1FFBH(8K-4byte) 00001: 00000H ~3FFBH(16K-4byte) 00010: 00000H ~5FFBH(24K-4byte) 00011: 00000H ~7FFBH(32K-4byte) 00100: 00000H ~9FFBH(40K-4byte) 00101: 00000H ~BFFBH(48K-4byte) 00110: 00000H ~DFFBH(56K-4byte) 00111: 00000H ~FFFBH(64K-4byte) Other: Reserved	0x0

Note 1: bit6, bit5, bit4, and bit3 should be set to 0.

Note 2: The expected value of the CRC operation result for comparison must be stored in the last 4 bytes of flash memory in advance, so the operation range is the range minus 4 bytes.

25.3.1.2 Flash CRC Operation Result Register L(PGCRCL)

This is a register that stores the results of high-speed CRC operations.

The PGCRCL register is set by a 16-bit memory operation instruction.

After a reset signal is generated, the value of this register changes to “0000H”.

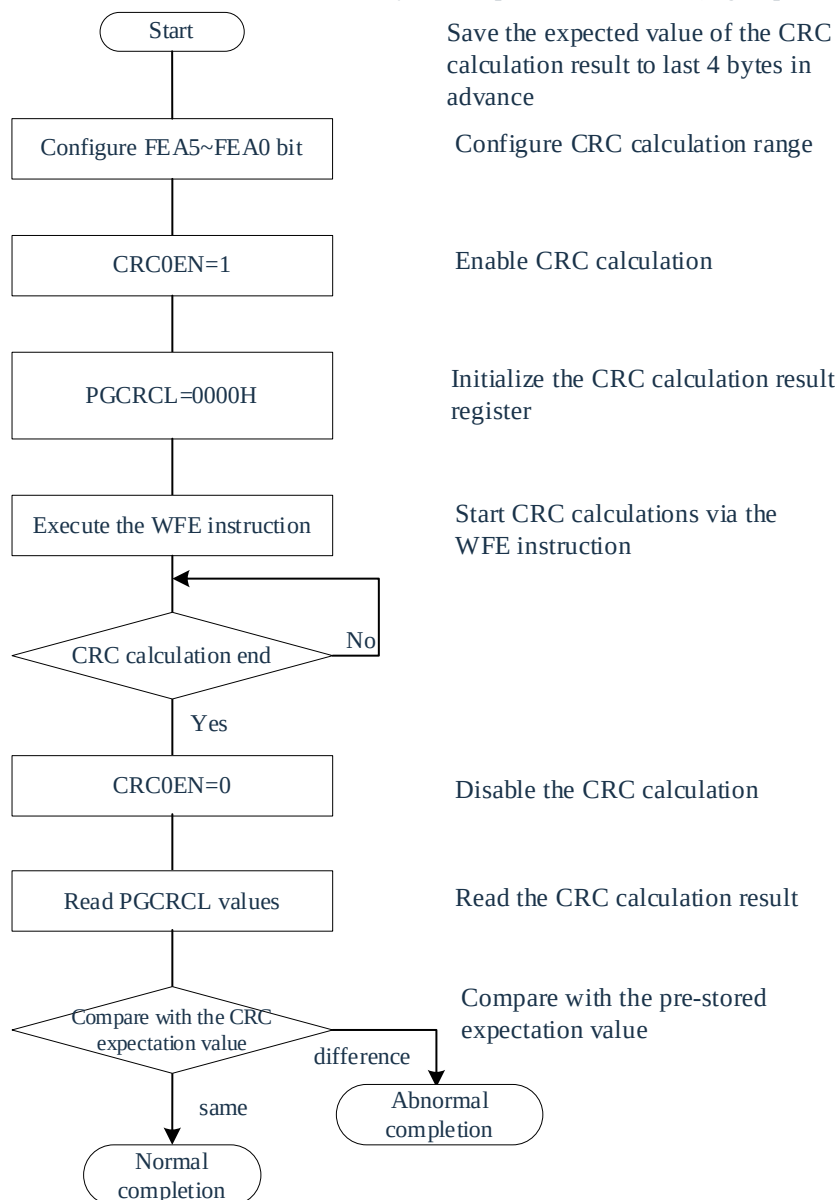
Bit	Symbol	Description	Reset value
15:0	PGCRCL	Store the high-speed CRC operation results. (0000H ~FFFFH)	0x0

Note: The PGCRCL register can only be written if the CRC0EN (bit 7 of CRC0CTL register) is 1.

The flow chart of the flash CRC operation function (high-speed CRC) is shown in Figure 25-1.

<Operation flow>

Figure 25-1: Flowchart for flash memory CRC operation function (high-speed CRC)



Note 1: Only code flash is the object of CRC operation.

Note 2: The expected value of CRC operation must be stored in the area after the operation range in the code flash memory.

25.3.2 SFR Guard Function

To ensure safety during operation, the IEC61508 standard requires that even if the CPU loses control, important SFRs (Special Function Registers) must be protected from being overwritten. The SFR protection feature is used to protect data in port functions, clock control functions, and control registers of the voltage detection circuit.

When the SFR protection feature is enabled, write operations to the protected SFR are invalid, but reading remains functional.

25.3.2.1 SFR Guard Control Register (SFRGD)

This register controls whether the SFR protection feature is enabled.

The GPORT bit and GCSC bit are used in SFR guard function.

The SFRGD register can be set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “00H”.

Bit	Symbol	Description	Reset value
7:3	Reserved	-	-
2	GPORT	Port function control register guard 0: Invalid, the port function control registers for can be read and written. 1: Valid, the port function control registers for can be read, but writing is disabled. [Guarded SFR]PMxx, PUxx, PDxx, POMxx, PMCxx, PS_xx ^{Note} .	0
1	-	Reserved	-
0	GCSC	Clock control function, voltage detection circuit control register guard 0: Invalid, the clock control function and voltage detection circuit control registers for can be read and written. 1: Valid, the clock control function and voltage detection circuit control registers for can be read, but writing is disabled. [Guarded SFR]CMC, CSC, OSTs, CKC, PERx, OSMC, LVIM, LVIS.	0

Note: Pxx (port register) is not guarded.

25.3.3 A/D Test Function

The IEC60730 standard requires testing of the A/D converter. This A/D test function confirms whether the A/D converter operates normally by performing A/D conversions on the positive (+) reference voltage, negative (–) reference voltage, analog input channel (AN), and internal reference voltage.

The following steps can be used to verify the analog multiplexer:

- 1) Select the AN_x pin as the A/D conversion object through the ADS register.
- 2) Perform A/D conversion on the AN_x pin (Conversion Result 1-1).
- 3) Select the negative (–) reference voltage of the A/D converter as the A/D conversion object through the ADS and ADICHS registers.
- 4) Perform A/D conversion on the negative (–) reference voltage of the A/D converter (Conversion Result 2-1).
- 5) Select the AN_x pin as the A/D conversion object through the ADS register.
- 6) Perform A/D conversion on the AN_x pin (Conversion Result 1-2).
- 7) Select the positive (+) reference voltage of the A/D converter as the A/D conversion object through the ADS and ADICH registers.
- 8) Perform A/D conversion on the positive (+) reference voltage of the A/D converter (Conversion Result 2-2).
- 9) Select the AN_x pin as the A/D conversion object through the ADS register.
- 10) Perform A/D conversion on the AN_x pin (Conversion Result 1-3).
- 11) Confirm that Conversion Result 1-1, Conversion Result 1-2, and Conversion Result 1-3 are the same.
- 12) Confirm that the A/D conversion result of Conversion Result 2-1 is all 0 and the A/D conversion result of Conversion Result 2-2 is all 1. Through these steps, the analog multiplexer can be selected, and it can be confirmed that the wiring is intact.

Note 1: During steps 1) to 10), if the analog input voltage is variable, other methods must be used to verify the analog multiplexer.

Note 2: Since conversion results contain errors, appropriate consideration of the error should be made when comparing the results.

25.3.3.1 A/D Input Channel Selection Register

The ADS and ADICHS registers are used to select the positive (+) reference voltage, negative (–) reference voltage, analog input channel (AN_{xx}), and internal reference voltage (0.8V) as A/D conversion objects for the A/D converter.

When used as an A/D test function, the following settings should be made:

- When measuring the zero scale, select the negative (–) reference voltage as the A/D conversion object.
- When measuring the full scale, select the positive (+) reference voltage as the A/D conversion object.

25.3.4 Digital Output Signal Level Detection Function for Input/Output Pins

The IEC60730 standard mandates the confirmation of proper I/O functionality.

The digital output signal level detection function of the input/output pin can read the digital output level of the pin when the pin is in the output mode.

25.3.4.1 Port Mode Selection Register (PMS)

This register selects whether to read the value of the port's output latch or the output level of the pin when the pin is in output mode (PMmn bit of the Port Mode Register (PMm) is 0).

The PMS register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to "00H".

Bit	Symbol	Description	Reset value
7:1	-	Reserved	-
0	PMS0	Selection of read data when the pin is in output mode 0: Read the value of the Pmn register. 1: Read the digital output level of the pin.	0

Note: m=A, B, F, n=0~15.

25.3.5 Product Unique ID Register

The product's unique ID is ideally suited for:

- Use as a serial number (e.g., USB character serial number or other terminal applications)
- Use as a password, combining this unique ID with software encryption/decryption algorithms when writing to flash memory to enhance the security of code within the flash storage.
- Use to activate a boot process with security mechanisms.

The 128-bit product unique ID provides a reference number that is unique for any BAT32 microcontroller under all conditions. The user cannot modify this ID under any circumstances.

Product unique ID register 0(UID0)

Bit	Symbol	Description	Reset value
31:0	-	Product unique ID bits [31:0], and the values are written before shipment.	-

Product unique ID register 1(UID1)

Bit	Symbol	Description	Reset value
31:0	-	Product unique ID bits [63:32], and the values are written before shipment.	-

Product unique ID register 2(UID2)

Bit	Symbol	Description	Reset value
31:0	-	Product unique ID bits [95:64], and the values are written before shipment.	-

Product unique ID register 3(UID3)

Bit	Symbol	Description	Reset value
31:0	-	Product unique ID bits [127:96], and the values are written before shipment.	-

Chapter 26 Option Byte

26.1 Functions of Option Bytes

The chip's flash memory range from 000C0H to 000C3H, and 500004H is the option byte area.

The option byte consists of the user option byte (000C0H to 000C2H) and the flash memory data protection option byte (000C3H, 500004H). Upon power-up or reset, the system will automatically reference the option byte to set specific functions. When using this product, the following functions must be configured through the option byte. For bits without configuration, the initial value cannot be changed.

Caution: Whether or not each function is used, the option byte must be configured.

26.1.1 User Option Bytes (000C0H~000C2H)

1. 000C0H

IWDG hardware enable control

2. 000C1H

LVD operating mode configuration

- Interrupt & Reset Mode
- Reset Mode
- Interrupt Mode
- LVD OFF (uses RESETB pin for external reset input).

Setting of LVD detection levels (V_{LVDH} , V_{LVDL} , V_{LVD})

Note 1: When the power voltage rises, the reset state must be maintained by the voltage detection circuit or external reset until the power voltage reaches the operating voltage range specified in the data sheet's AC characteristics. When the power voltage drops, the reset state must be set by the deep sleep mode transition, voltage detection circuit, or external reset before the power voltage falls below the operating voltage range.

Note 2: The operating voltage range depends on the setting of the user option byte (000C2H).

3. 000C2H

High-speed on-chip oscillator frequency configuration

- Select from 1.5MHz~48MHz.

26.2 Flash Memory Data Protection Option Bytes (000C3H, 500004H)

Control of on-chip debug operation

Level0: Enabling read/write/erase operations on flash memory data via debugger.

Level1: Chip full erase operations on flash data via debugger are enabled, read and write operations are disabled.

Level2: Manipulation of flash data via debugger is disabled.

26.3 Register Mapping

Register	Address	R/W	Description	Reset value
Option byte 0	0x00C0H	R/W	Watchdog Control Register	0xFF
Option byte 1	0x00C1H	R/W	LVD Status Control Register	0xFF
Option byte 2	0x00C2H	R/W	High-Speed On-Chip Oscillator Frequency Control Register	0xE6
Option byte 3	0x00C3H	R/W	On-Chip Debug Flash Data Protection Control Register 1	0xFF
Option byte 4	0x500004H	R/W	On-Chip Debug Flash Data Protection Control Register 2	0xFF

26.4 User Option Bytes

26.4.1 User Option Byte (000C0H)

Bit	Symbol	Description	Reset value
7:4	--	Reserved	0xF
3:0	IWDG_SW	IWDG hardware enable control =A/F: Control of IWDG count enable by software After CONFIG is loaded and takes effect, force enable =Other: the IWDG counting.	0xF

26.4.2 User Option Byte (000C1H)

Bit	Symbol	Description	Reset value
7:5	VPOC[2:0]	Detection voltage setting	0x7
4	-	Reserved (set to 1)	1
3:2	LVIS[1:0]	Detection voltage setting	0x2
1:0	LVIMDS[1:0]	Mode selection 10: Interrupt & Reset Mode 11: Reset Mode 01: Interrupt Mode	0x3

• LVD setting (interrupt & reset mode)

Detection voltage			Setting value of option byte								
V _{LVDH}		V _{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting			
rise	drop	drop						LVIMDS1	LVIMDS0		
1.85V	1.8V	1.8V	0	0	1	1	1	1	0		
1.95V	1.9V					1	0				
2.05V	2.0V					0	1				
3.15V	3.1V					0	0				
2.50V	2.45V	2.45V		1	0	1	1			1	0
2.60V	2.55V					1	0				
2.70V	2.65V					0	1				
3.70V	3.65V					0	0				
2.80V	2.75V	2.75V		1	1	1	1			1	0
2.90V	2.85V					1	0				
3.00V	2.95V					0	1				
4.05V	4.0V					0	0				
—			Setting values other than those mentioned above is prohibited.								

Note 1: Set the bit4 to 1.

Note 2: For details of the LVD circuit, refer to “Chapter 24 Voltage Detection Circuit”.

Note 3: The detection voltage is the TYP. value. For details, refer to LVD Circuit Characteristics in the datasheet.

- LVD setting (reset mode)

Detection voltage		Setting value of option byte						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
rise	drop						LVIMDS1	LVIMDS0
1.85V	1.8V	0	0	1	1	1	1	1
1.95V	1.9V		0	1	1	0		
2.05V	2.0V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.60V	2.55V		1	0	1	0		
2.70V	2.65V		1	0	0	1		
2.80V	2.75V		1	1	1	1		
2.90V	2.85V		1	1	1	0		
3.00V	2.95V		1	1	0	1		
3.15V	3.1V		0	1	0	0		
3.70V	3.65V		1	0	0	0		
4.05V	4.0V		1	1	0	0		
—		Setting values other than those mentioned above is prohibited.						

Note 1: Set the bit4 to 1.

Note 2: For details of the LVD circuit, refer to “Chapter 24 Voltage Detection Circuit”

Note 3: The detection voltage is the TYP. value. For details, refer to LVD Circuit Characteristics in the datasheet.

- LVD setting (interrupt mode)

Detection voltage		Setting value of option byte						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
rise	drop						LVIMDS1	LVIMDS0
1.85V	1.8V	0	0	1	1	1	0	1
1.95V	1.9V		0	1	1	0		
2.05V	2.0V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.60V	2.55V		1	0	1	0		
2.70V	2.65V		1	0	0	1		
2.80V	2.75V		1	1	1	1		
2.90V	2.85V		1	1	1	0		
3.00V	2.95V		1	1	0	1		
3.15V	3.1V		0	1	0	0		
3.70V	3.65V		1	0	0	0		
4.05V	4.0V		1	1	0	0		
—		Setting values other than those mentioned above is prohibited.						

Note 1: Set the bit4 to 1.

Note 2: For details of the LVD circuit, refer to “Chapter 24 Voltage Detection Circuit”.

Note 3: The detection voltage is the TYP. value. For details, refer to LVD Circuit Characteristics in the datasheet.

- Setting when LVD is OFF (external reset input using RESETB pin)

Detection voltage		Setting value of option byte						
V _{LVDH}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
rise	drop						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		Setting values other than those mentioned above is prohibited.						

Caution 1: Set the bit4 to 1.

Caution 2: When the supply voltage rises, the reset state must be maintained by the voltage detection circuit or external reset before the supply voltage reaches the operating voltage range shown in the AC Characteristics of the datasheet; when the supply voltage falls, it must be set to the reset state by the transfer of the sleep mode, the voltage detection circuit, or the external reset before the supply voltage falls below the operating voltage range.

The operating voltage range depends on the setting of the user option byte (000C2H).

Note 1: ×: Ignore.

Note 2: For details of the LVD circuit, refer to “Chapter 24 Voltage Detection Circuit”.

Note 3: The detection voltage is the TYP. value. For details, refer to LVD Circuit Characteristics in the datasheet.

26.4.3 User Option Byte (000C2H)

Bit	Symbol	Description	Reset value
7:5	-	Reserved (set to 1)	0x7
4:0	FRQSE[4:0]	High-speed on-chip oscillator clock frequency selection	0x06

FRQSE4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	High-speed on-chip oscillator clock frequency	
					f _{HOCO}	f _{IH}
0	0	0	0	0	48MHz	48MHz
0	0	0	0	1	48MHz	24MHz
0	0	0	1	0	48MHz	12MHz
0	0	0	1	1	48MHz	6MHz
0	0	1	0	0	48MHz	3MHz
0	0	1	0	1	48MHz	1.5MHz
Others					Setting prohibited	

Note 1: Set the bits 7 to 5 to 1.

Note 2: The operating frequency range and operating voltage range vary depending on each operating mode of the flash memory. For details, refer to AC Characteristics in the datasheet.

26.4.4 Flash Memory Data Protection Option Byte (000C3H)

Bit	Symbol	Description	Reset value
7:0	OCDEN[7:0]	Control of flash data protection	0xFF

26.4.5 Flash Memory Data Protection Option Byte (500004H)

Bit	Symbol	Description	Reset value
7:0	OCDM[7:0]	Control of flash data protection	0xFF

OCDM	OCDEN	Control of flash data protection
3C	C3	Flash memory data operations are not enabled through the debugger.
Other than 3C	C3	Flash memory data operations are enabled through the debugger for full chip erase, but read and write operations are not enabled.
Other than the above		Flash memory data read/write/erase operations are enabled through the debugger.

Note: The address 50_0004H belongs to the data flash area. If this address is used for data storage, it must be confirmed that the value will not cause an incorrect setting of the protection options.

Chapter 27 Flash Control

27.1 Description of Flash Control

This product includes a 64KByte capacity Flash memory, divided into 128 sectors, with each sector having a capacity of 512Bytes. It can be used as program memory or data memory. This module supports erase, programming, and read operations on this memory.

27.2 Structure of Flash Memory

FFFF_FFFFH	Reserved
E00F_FFFFH	Cortex-M0+ Dedicated Peripheral Resource Region
E000_0000H	Reserved
4008_FFFFH	Peripheral Resource Region
4000_0000H	Reserved
2000_1FFFFH	SRAM(8KB)
2000_0000H	Reserved
0050_05FFFH	Data Flash 1KB
0050_0200H	Reserved
0000_FFFFH	Main Flash Region (up to 64KB)
0000_0000H	

27.3 Register Mapping

(Flash control base address = 0x4002_0000)

RO: Read-Only, WO: Write-Only, R/W: Read/Write

Register	Address	R/W	Description	Reset value
FLKEY	0x004	WO	Flash Key Register	0x0
FLSR	0x00C	R/W	Flash Status Register	0x0
FLPRT	0x010	R/W	Flash Protection Register	0x0
FLCMD	0x014	R/W	Flash Operation Control Register	0x0
FLADDR	0x018	R/W	Flash Address Register	0x00ffffff
FLDATA0	0x01C	R/W	Flash Data Register 0	0xffffffff
FLCR	0x024	R/W	Flash Control Register	0x80000300
FLREGION	0x048	R/W	Flash Region Control Register	0x0

27.4 Register Description

27.4.1 Flash Operation Control Register (FLCMD)

The Flash Operation Control Register is used to select Flash operations.

Bit	Symbol	Description	Reset value
31:8	-	Reserved	-
7:0	CMD	Operation selection: 56H: Chip erase 5CH: Erase without retry AAH: Write without pre-written data	0x0

27.4.2 Flash Key Register (FLKEY)

With the help of the Flash Key Register, access to the Flash Control Register can be allowed, thus enabling programming and erase operations.

Bit	Symbol	Description	Reset value
31:0	KEY	To unlock the Flash registers and allow programming/erase operations, the following values must be programmed in sequence: (a): KEY = 0x45670123 (b): KEY = 0xCDEF89AB	0x0

27.4.3 Flash Data Register 0 (FLDATA0)

Flash Data Register, write the value to be written into this register.

Bit	Symbol	Description	Reset value
31:0	DATA0	Flash written data	0xffffffff

27.4.4 Flash Address Register (FLADDR)

Flash Address Register, write the address to be written or erased into this register.

Bit	Symbol	Description	Reset value
31:24	-	Reserved	0x0
23:0	ADDR	Address to be written or erased	0xffffffff

27.4.5 Flash Control Register (FLCR)

The Flash Control Register is used to configure and initiate Flash operations.

Bit	Symbol	Description	Reset value
31	LOCK	This bit can only be written as 1. When this bit is set to 1, it indicates that the Flash_CR register is locked. When the unlock sequence is detected, this bit will be cleared by hardware. If the unlock operation fails, this bit remains set to 1 until the next reset.	1
30:18	-	Reserved	0
17	ERRIE	Error interrupt enable: 0: Disable error interrupt 1: Enable error interrupt	0x0
16	EOPIE	Operation complete interrupt enable: 0: Disable interrupt 1: Enable interrupt	0x0
15:10	-	Reserved	0
9:8	PSIZE	Data write width: 00: X8 01: X16 10: X32 11: Disable setting	0x11
7:1	-	Reserved	0x0
0	STRT	Flash erase/write start bit: 0: Cleared by hardware after the BSY bit is cleared 1: Triggers erase or write operation (can only be set to 1 by software)	0

27.4.6 Flash Status Register (FLSR)

The Flash Status Register provides information related to ongoing programming and erase operations.

Bit	Symbol	Description	Reset value
31:17	-	Reserved	0x0
16	BSY	Busy ^{Note 1} : 0: No Flash operation is currently being executed 1: A Flash operation is in progress (This bit indicates that a Flash operation is ongoing. It is set to 1 when the operation begins and cleared when the operation ends or an error occurs.)	0
15:7	-	Reserved	0x0
6	PGPERR	Programming parallel bit error: 0: Programming parallel bits are normal 1: The code performs a write access to Flash without correctly configuring the control register, and hardware will set this bit to 1. Software must write 1 to clear it.	0
5	-	Reserved	0
4	WRPERR	Write protection error 0: Write protection is normal 1: The address for erase/programming belongs to a write-protected area in Flash, and hardware will set this bit to 1. Software must write 1 to clear it.	0
3:2	-	Reserved	0x0
1	OPERR	Operation error: 0: Operation is normal 1: A Flash operation (programming/erase/read) request was detected, but it could not be executed due to parallel bit error, alignment error, or write protection error. Hardware will set this bit to 1.	0
0	EOP	Operation complete: 0: Operation not completed 1: When one or more Flash operations (programming/erase) are successfully completed, hardware will set this bit to 1. Software must write 1 to clear it.	0

Note 1: BSY is read-only.

27.4.7 Flash Region Control Register (FLREGION)

The Flash Region Selection Register is used to select the Flash access region.

Bit	Symbol	Description	Reset value
31:8	-	Reserved	0x0
7:0	REGION	Flash region selection bit: 55H: Program flash region AAH: Data flash region	0x0

27.4.8 Flash Protection Register (FLPRT)

The Flash Protection Register is used to protect the contents of the program region and data region.

Bit	Symbol	Description	Reset value
31	PRTALL	Full Sspace protection control bit 0: No write protection for the entire space 1: Write protection for the entire space	0
30:25	-	Reserved	0x0
24	PRTDATA	Data region protection control bit: 0: No write protection for the data region 1: Write protection for the data region	0
23:16	-	Reserved	0x0
15	PRTROM15	Program region 60K-64K(F000H-FFFFH) protection control bit: 0: No write protection 1: Write protection	0
14	PRTROM14	Program region 56K-60K(E000H-EFFFH) protection control bit: 0: No write protection 1: Write protection	0
13	PRTROM13	Program region 52K-56K(D000H-DFFFH) protection control bit: 0: No write protection 1: Write protection	0
12	PRTROM12	Program region 48K-52K(C000H-CFFFH) protection control bit: 0: No write protection 1: Write protection	0
11	PRTROM11	Program region 44K-48K(B000H-BFFFH) protection control bit: 0: No write protection 1: Write protection	0
10	PRTROM10	Program region 40K-44K(A000H-AFFFH) protection control bit: 0: No write protection 1: Write protection	0
9	PRTROM9	Program region 36K-40K(9000H-9FFFH) protection control bit: 0: No write protection 1: Write protection	0
8	PRTROM8	Program region 32K-36K(8000H-8FFFH) protection control bit: 0: No write protection 1: Write protection	0
7	PRTROM7	Program region 28K-32K(7000H-7FFFH) protection control bit: 0: No write protection 1: Write protection	0
6	PRTROM6	Program region 24K-28K(6000H-6FFFH) protection control bit: 0: No write protection 1: Write protection	0
5	PRTROM5	Program region 20K-24K(5000H-5FFFH) protection control bit: 0: No write protection 1: Write protection	0
4	PRTROM4	Program region 16K-20K(4000H-4FFFH) protection control bit: 0: No write protection	0

		1: Write protection	
3	PRTROM3	Program region 12K-16K(3000H-3FFFH) protection control bit: 0: No write protection 1: Write protection	0
2	PRTROM2	Program region 8K-12K(2000H-2FFFH) protection control bit: 0: No write protection 1: Write protection	0
1	PRTROM1	Program region 4K-8K(1000H-1FFFH) protection control bit: 0: No write protection 1: Write protection	0
0	PRTROM0	Program region 0K-4K(0000H-0FFFH) protection control bit: 0: No write protection 1: Write protection	0

27.5 How to Operate the Flash

27.5.1 Programming

The write operation process is as follows:

- (1) Unlock the Flash register: Write in sequence

FLKEY=0x45670123;

FLKEY=0xCDEF89AB; After unlocking, FLCR.LOCK=0;

- (2) Disable protection for the programming region: FLPRT.PRTROMn=0(n=0-15) or FLPRT.PRTDATA=0.

- (3) Set the write mode: FLCMD=0xAA.

- (4) Set the programming region: FLREGION.

- (5) Write the target address into FLADDR.

- (6) Write the written data into FLDATA0.

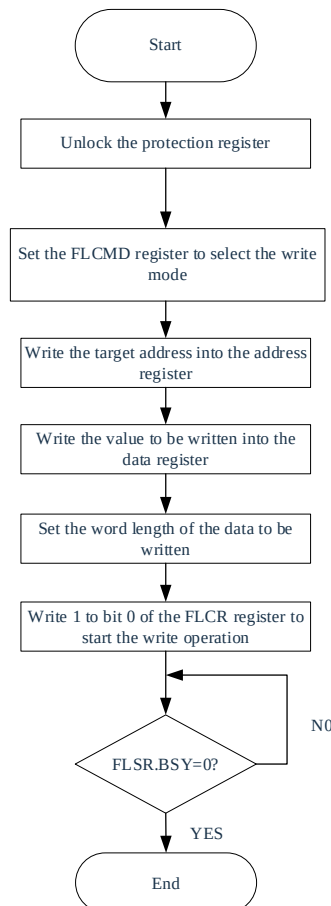
- (7) Set the data width to be written: FLCR.PSIZE.

- (8) Start programming: FLCR.STRT=1.

- (9) Software polls the status register: FLSR.BSY = 0 indicates the operation is complete, or software polls the status register: FLSR.EOP = 1, indicating the operation is complete.

- (10) Before the next operation, software writes “1” to clear FLSR.EOP.

- (11) After the operation is complete, FLCR.LOCK = 1, re-lock FLCR.



27.5.2 Sector Erase

Sector erase, the erase time is implemented by hardware. The operation process is as follows:

(1) Unlock the Flash register: Write in sequence

FLKEY=0x45670123;

FLKEY=0xCDEF89AB; After unlocking, FLCR.LOCK=0;

(2) Disable protection for the programming region: FLPRT.PRTROMn=0(n=0-15) or FLPRT.PRTDATA=0.

(3) Set the sector erase mode: FLCMD=0x5C.

(4) Set the programming region: FLREGION.

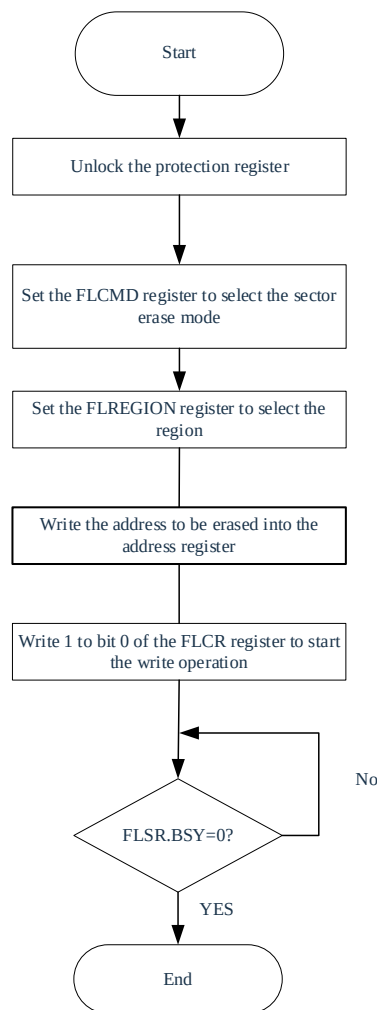
(5) Write the target address into FLADDR (write the start address of the sector to be erased).

(6) Start programming: FLCR.STRT=1.

(7) Software polls the status register: FLSR.BSY = 0 indicates the operation is complete, or software polls the status register: FLSR.EOP = 1, indicating the operation is complete.

(8) Before the next operation, software writes “1” to clear FLSR.EOP.

(9) After the operation is complete, FLCR.LOCK = 1, re-lock FLCR.



27.5.3 Chip Erase

Chip erase, the erase time is implemented by hardware. The operation process is as follows:

(1) Unlock the Flash register: Write in sequence

FLKEY=0x45670123;

FLKEY=0xCDEF89AB; After unlocking, FLCR.LOCK=0;

(2) Disable protection for the programming region: FLPRT.PRTROMn=0 (n=0-15) or

FLPRT.PRTDATA=0, FLPRT.PRTALL=0.

(3) Set the sector erase mode: FLCMD=0x56.

(4) Set the programming region: FLREGION=0x55(chip erase only erases the program flash region).

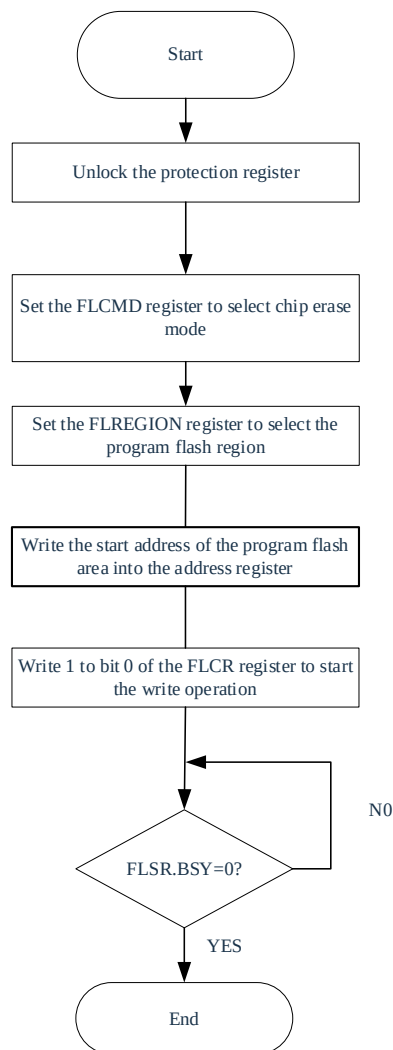
(5) Write the target address into FLADDR (write the start address of the program flash region)

(6) Start programming: FLCR.STRT=1.

(7) Software polls the status register: FLSR.BSY = 0 indicates the operation is complete, or software polls the status register: FLSR.EOP = 1, indicating the operation is complete.

(8) Before the next operation, software writes “1” to clear FLSR.EOP.

(9) After the operation is complete, FLCR.LOCK = 1, re-lock FLCR.



27.6 Cautions for Flash Operation

Flash memory has strict timing requirements for erase and programming control signals. If the timing of the control signals is incorrect, the erase and programming operations will fail. The erase and program parameters are implemented by hardware, and there is no need to set the parameter registers.

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 before continuing with the next instruction. If the operation is executed from RAM, the CPU will not stop fetching instructions, and the next instruction can be executed.

During Flash programming, if the CPU executes a deep sleep instruction, the system will wait for the programming operation to complete before entering deep sleep.

Chapter 28 Appendix Revision History

Version #	Date	Description of changes
V0.0.1	Jun. 2024	Initial release
V0.1.0	Feb. 2025	Modified the maximum frequency of the SPI Communication Unit in slave mode
V0.1.1	Mar. 2025	1) Modified UID base address 2) Modified Flash operation control register
V0.1.2	Mar. 2025	Optimized format
V0.1.3	Apr. 2025	Modified Chapters 2/4/7/8/9/11/12/15/16/17/18/27/28
V0.1.4	Apr. 2025	Modified DMA channel trigger source selection register in Section 16.4.5
V0.1.5	May 2025	Deleted content related to Clock Output (MCO)
V0.1.6	Jun. 2025	1) Deleted USRAT2 and modified DMA content 2) Modified Flash CRC control register description 3) Modify sections 12.3.8-12.3.14 register descriptions
V0.9.0	Jun. 2025	Modify Figure 16-3 control data allocation and 13.5.3 SPI status register
V0.9.1	Aug. 2025	1) Modified SPI Status Register (SPI_SR) 2) Deleted Flash Access Control Register (FLACR) 3) Modified 12.3.15 Noise Filter Enable Register 0 (NFEN0) 4) Revised 18.5.5 Analog comparator event control register names 5) Modified TIM1 Brake and Dead-Time Register descriptions