



BAT32G133B User Manual

Ultra-low power 32-bit microcontrollers based on ARM® Cortex®-M0+

V0.5.1

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 BAT32G133B 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:

BAT32G133Bxx: BAT32G133B 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	14
1.1 Overview	14
1.2 Cortex-M0+ Core Features	14
1.3 Debug Features	14
1.4 SWD Interface Pins	16
1.5 ARM Reference Documentation	17
Chapter 2 Pin Functions	18
2.1 Port Functions	18
2.2 Port Alternate Functions	20
2.3 Registers for Controlling Port Functions	21
2.3.1 Port Mode Register (PMxx)	22
2.3.2 Port register (Pxx)	23
2.3.3 Pull-Up Resistor Selection Register (PUxx)	24
2.3.4 Pull-Down Resistor Selection Register (PDxx)	25
2.3.5 Port Output Mode Register (POMxx)	26
2.3.6 Port Mode Control Register (PMCxx)	27
2.3.7 Port Set Control Register (PSETxx)	28
2.3.8 Port Clear Control Register (PCLRxx)	29
2.3.9 Port Read Register (PREADxx)	30
2.3.10 Port Alternate Function Configuration Register (PxxCFG)	31
2.4 Handling of Unused Pins	34
2.5 Special Pin Configuration	34
2.5.1 Port P00 (Default is RESETB)	34
Chapter 3 System Architecture	35
3.1 Overview	35
3.2 System Address Partitioning	36
Chapter 4 Clock Generation Circuit	38
4.1 Functions of Clock Generation Circuit	38
4.2 Structure of Clock Generation Circuit	40
4.3 Registers for Controlling Clock Generation Circuit	43
4.3.1 Clock Operation Mode Control Register (CMC)	44
4.3.2 System Clock Control Register (CKC)	46
4.3.3 Clock Operation Status Control Register (CSC)	47
4.3.4 Oscillation Stabilization Time Counter Status Register (OSTC)	49
4.3.5 Oscillation Stabilization Time Select Register (OSTS)	51
4.3.6 Peripheral Enable Registers 0, 1 (PER0, PER1)	52
4.3.7 Subsystem Clock Supply Mode Control Register (OSMC)	56
4.3.8 High-Speed On-Chip Oscillator Frequency Select Register (HOCODIV)	57
4.3.9 Low-Speed On-Chip Oscillator Frequency Selection Register (LOSCSEL) and Its Protection Register (PRCR)	58
4.3.10 External Reset Pin Reset Function Mask Register (RSTM)	59
4.4 System Clock Oscillation Circuit	60

4.4.1	X1 Oscillation Circuit	60
4.4.2	XT1 Oscillation Circuit.....	61
4.4.3	High-Speed On-Chip Oscillator	64
4.4.4	Low-Speed On-Chip Oscillator	64
4.5	Operation of Clock Generation Circuit	65
4.6	Clock Control	67
4.6.1	Example of High-Speed On-Chip Oscillator Set-Up	67
4.6.2	Example of Setting X1 Oscillation Circuit.....	69
4.6.3	Example of Setting XT1 Oscillation Circuit	70
4.6.4	State Transition Diagram of CPU Clock	71
4.6.5	Conditions Before Changing CPU Clock and Processing After Changing CPU Clock	76
4.6.6	Time Required to Switch CPU Clock and Main System Clock	78
4.6.7	Conditions Before Clock Oscillation Is Stopped.....	79
Chapter 5	Universal Timer Unit (Timer4)	80
5.1	Functions of Universal Timer Unit.....	81
5.1.1	Independent Channel Operation	81
5.1.2	Multi-Channel Linkage Operation Function	83
5.1.3	8-Bit Timer Operation Function (Channels 1 and 3 of Unit 0 Only).....	84
5.2	Structure of Universal Timer Unit.....	85
5.2.1	List of Universal Timer Unit 0 Registers.....	88
5.2.2	List of Universal Timer Unit 1 Registers.....	89
5.2.3	Timer Count Register mn (TCRmn).....	90
5.2.4	Timer Data Register mn (TDRmn).....	91
5.3	Registers for Controlling Universal Timer Unit	92
5.3.1	Peripheral Enable Register 0 (PER0)	93
5.3.2	Timer Clock Select Register m (TPSm).....	94
5.3.3	Timer Mode Register mn (TMRmn)	97
5.3.4	Timer Status Register mn (TSRmn)	102
5.3.5	Timer Channel Enable Status Register m (TEm)	103
5.3.6	Timer Channel Start Register m (TSm).....	104
5.3.7	Timer Channel Stop Register m (TTm).....	105
5.3.8	Timer Input/Output Selection Register (TIOS0)	106
5.3.9	Timer Output Enable Register m (TOEm)	107
5.3.10	Timer Output Register m (TOm).....	108
5.3.11	Timer Output Level Register m (TOLm)	109
5.3.12	Timer Output Mode Register m (TOMm)	110
5.3.13	Noise Filter Enable Register 1 (NFEN1).....	111
5.3.14	Noise Filter Enable Register 2 (NFEN2).....	112
5.3.15	Registers for Controlling Port Functions of Timer Input/Output Pins.....	113
5.4	Basic Rules of Universal Timer Unit.....	114
5.4.1	Basic Rules of Multi-Channel Linkage Operation Function	114
5.4.2	Basic Rules of 8-Bit Timer Operation Function (Channels 1 and 3 of Unit 0 Only)	116
5.5	Operation of Counter	117
5.5.1	Count Clock (f_{CLK}).....	117
5.5.2	Start Timing of Counter.....	119
5.5.3	Operation of Counter.....	120
5.6	Control of Channel Output (TOMn Pin).....	125

5.6.1	Configuration of TOMn Pin Output Circuit.....	125
5.6.2	Output Settings for TOMn Pins	126
5.6.3	Cautions for Channel Output Operation.....	127
5.6.4	One-Time Operation of TOMn Bit	131
5.6.5	Timer Interrupt and TOMn Pin Output When Counting Starts.....	132
5.7	Control of Timer Input (TIMn)	133
5.7.1	Structure of TIMn Pin Input Circuit	133
5.7.2	Noise Filter	133
5.7.3	Cautions on Channel Input Operation	134
5.8	Independent Channel Operation Function of Universal Timer Unit.....	135
5.8.1	Operation as Interval Timer/Square Wave Output.....	135
5.8.2	Operation as External Event Counter	139
5.8.3	Operation as Frequency Divider.....	142
5.8.4	Operation as Input Pulse Interval Measurement.....	145
5.8.5	Operation as Input Signal High-/Low-Level Width Measurement.....	148
5.8.6	Operation as Delay Counter	152
5.9	Multi-Channel Linkage Operation Function for Universal Timer Unit.....	155
5.9.1	Operation as Single Trigger Pulse Output Function	155
5.9.2	Operation as PWM Function	162
5.9.3	Operation as Multiple PWM Output Function	169
Chapter 6	Real-Time Clock	176
6.1	Functions of Real-Time Clock.....	176
6.2	Structure of Real-Time Clock.....	176
6.3	Registers for Controlling Real-Time Clock.....	178
6.3.1	Peripheral Enable Register 0 (PER0)	179
6.3.2	Real-Time Clock Selection Register (RTCCCL)	180
6.3.3	Real-Time Clock Control Register 0 (RTCC0)	181
6.3.4	Real-Time Clock Control Register 1 (RTCC1)	182
6.3.5	Clock Error Correction Register (SUBCUD).....	184
6.3.6	Second Count Register (SEC)	185
6.3.7	Minute Count Register (MIN)	185
6.3.8	Hour Count Register (HOUR).....	186
6.3.9	Day Count Register (DAY)	188
6.3.10	Week Count Register (WEEK).....	189
6.3.11	Month Count Register (MONTH).....	190
6.3.12	Year Count Register (YEAR)	190
6.3.13	Alarm Minute Register (ALARMWM).....	191
6.3.14	Alarm Hour Register (ALARMWH).....	191
6.3.15	Alarm Week Register (ALARMWW)	192
6.3.16	Port Mode Register and Port Register	192
6.4	Operation of Real-Time Clock	193
6.4.1	Start of Real-Time Clock Operation.....	193
6.4.2	Shifting to Sleep Mode After Starting Operation	194
6.4.3	Reading/Writing Real-Time Clock.....	195
6.4.4	Setting Alarm of Real-Time Clock.....	197
6.4.5	1Hz Output of Real-Time Clock	198
6.4.6	Example of Clock Error Correction of Real-Time Clock.....	199

Chapter 7	15-Bit Interval Timer	201
7.1	Functions of 15-Bit Interval Timer	201
7.2	Structure of 15-Bit Interval Timer	201
7.3	Registers for Controlling 15-Bit Interval Timer	202
7.3.1	Peripheral Enable Register 0 (PER0)	202
7.3.2	Real-Time Clock Selection Register (RTCCL)	203
7.3.3	15-Bit Interval Timer Control Register (ITMC)	204
7.4	Operation of 15-Bit Interval Timer	205
7.4.1	Operation Timing of 15-Bbit Interval Timer	205
7.4.2	Start of Count Operation and Re-Enter to Sleep Mode After Returned from Sleep Mode	206
Chapter 8	Clock Output/Buzzer Output Controller	207
8.1	Functions of Clock Output/Buzzer Output Controller	207
8.2	Structure of Clock Output/Buzzer Output Controller	208
8.3	Registers for Controlling Clock Output/Buzzer Output Controller	209
8.3.1	Clock Output Select Register n (CKSn)	209
8.3.2	Registers for Controlling Clock Output/Buzzer Output Port Functions	210
8.4	Operation of Clock Output/Buzzer Output Controller	211
8.4.1	Operation as Output Pin	211
8.5	Cautions for Clock Output/Buzzer Output Controller	211
Chapter 9	Watch Dog Timer	212
9.1	Functions of Watch Dog Timer	212
9.2	Configuration of Watchdog Timer	213
9.3	Registers Controlling Watch Dog Timer	214
9.3.1	Watch Dog Timer Enable Register (WDTE)	214
9.3.2	LOCKUP Control Register (LOCKCTL) and Its Protection Register (PRCR)	215
9.3.3	Watchdog Configuration Register (WDTCFG0/1/2/3)	216
9.4	Operation of Watchdog Timer	217
9.4.1	Operational Control of Watchdog Timer	217
9.4.2	Setting Overflow Time of Watchdog Timer	218
9.4.3	Setting Window Open Period of Watchdog Timer	219
9.4.4	Setting Watchdog Timer Interval Interrupt	220
Chapter 10	A/D Converter	221
10.1	Functions of A/D Converter	221
10.2	Registers for Controlling A/D Converter	223
10.2.1	Peripheral Enable Register 0 (PER0)	224
10.2.2	A/D Converter Mode Register 0 (ADM0)	225
10.2.3	A/D Converter Mode Register 1 (ADM1)	230
10.2.4	A/D Converter Mode Register 2 (ADM2)	231
10.2.5	A/D Converter Trigger Mode Register (ADTRG)	232
10.2.6	Analog Input Channel Specification Register (ADS)	233
10.2.7	12-Bit A/D Conversion Result Register (ADCR)	235
10.2.8	8-Bit A/D Conversion Result Register (ADCRH)	236
10.2.9	Conversion Result Comparison Upper Limit Setting Register (ADUL)	236
10.2.10	Conversion Result Comparison Lower Limit Setting Register (ADLL)	237
10.2.11	A/D Converter Sampling Time Control Register (ADNSMP)	238
10.2.12	A/D Sample Time Extension Register (ADSMPWAIT)	240

10.2.13	A/D Test Register (ADTES).....	241
10.2.14	A/D Status Register (ADFLG).....	242
10.2.15	A/D Charge/Discharge Control Register (ADNDIS)	243
10.2.16	Registers Controlling Port Functions of Analog Input Pins	243
10.3	Input Voltage and Conversion Results.....	244
10.4	Operation Mode of A/D Converter	245
10.4.1	Software Trigger Mode (Select mode, Continuous Conversion Mode)	245
10.4.2	Software Trigger Mode (Select Mode, Single Conversion Mode)	246
10.4.3	Software Trigger Mode (Scan mode, Sequential Conversion Mode).....	247
10.4.4	Software Trigger Mode (Scan Mode, Single Conversion Mode)	248
10.4.5	Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode).....	249
10.4.6	Hardware Trigger No-Wait Mode (Select Mode, Single Conversion Mode)	250
10.4.7	Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode).....	251
10.4.8	Hardware Trigger No-Wait Mode (Scan Mode, Single Conversion Mode)	252
10.4.9	Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode).....	253
10.4.10	Hardware Trigger Wait Mode (Select Mode, Single Conversion Mode)	254
10.4.11	Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode).....	255
10.4.12	Hardware Trigger Wait Mode (Scan Mode, Single Conversion Mode)	256
10.5	A/D Converter Setup Flowchart	257
10.5.1	Software Trigger Mode Settings.....	257
10.5.2	Hardware Trigger No-Wait Mode Settings.....	258
10.5.3	Hardware Trigger Wait Mode Settings	259
10.5.4	Settings When Selecting the Output Voltage/Internal Reference Voltage of the Temperature Sensor	260
10.5.5	Test Mode Settings	261
Chapter 11	Comparator.....	262
11.1	Functions of Comparator	262
11.2	Structure of Comparator	263
11.3	Registers for Controlling Comparator	265
11.3.1	Peripheral Enable Register 1 (PER1)	266
11.3.2	Comparator Mode Configuration Register (COMPMDR)	267
11.3.3	Comparator Filter Control Register (COMPFIR).....	268
11.3.4	Comparator Output Control Register (COMPOCR)	270
11.3.5	Comparator Built-In Reference Voltage Control Register (CVRCTL)	272
11.3.6	Comparator Built-In Reference Voltage Selection Register (CiRVM).....	273
11.3.7	Comparator 0 Input Signal Selection Control Register (CMPSEL0)	274
11.3.8	Comparator 1 Input Signal Selection Control Register (CMPSEL1)	275
11.3.9	Registers Controlling Port Functions of Analog Input Pins	275
11.4	Operation Instructions	276
11.4.1	Digital Filter of Comparator i (i=0, 1).....	278
11.4.2	Comparator i Interrupts (i=0, 1)	278
11.4.3	Event Signals Output to the Linkage Controller (EVENTC)	279
11.4.4	Output of Comparator i (i=0, 1)	280
11.4.5	Stop and Enable Comparator Clock Supply	280
Chapter 12	Programmable Gain Amplifier (PGA)	281
12.1	Functions of Programmable Gain Amplifier	281
12.2	Structure of Programmable Gain Amplifier	282

12.3	Registers for Programmable Gain Amplifier	283
12.3.1	Peripheral Enable Register 1 (PER1)	283
12.3.2	Programmable Gain Amplifier Control Register (PGAnCTL).....	284
12.3.3	Registers Controlling Port Functions of Analog Input Pins	284
12.4	Operation of Programmable Gain Amplifier	285
12.4.1	Starting Operation Steps of Programmable Gain Amplifier	285
12.4.2	Stopping Operation Steps of Programmable Gain Amplifier	286
Chapter 13	Universal Serial Communication Unit	287
13.1	Functions of Universal Serial Communication Unit	288
13.1.1	3-Wire Serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21)	288
13.1.2	UART (UART0~UART2)	289
13.1.3	Simplified I ² C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21).....	290
13.2	Structure of Universal Serial Communication Unit.....	291
13.2.1	Shift Register.....	294
13.2.2	Low 8 Bits or Low 9 Bits of Serial Data Register mn (SDRmn)	294
13.3	Registers for Controlling Universal Serial Communication Unit.....	296
13.3.1	Peripheral Enable Register 0 (PER0)	297
13.3.2	Serial Clock Select Register m (SPSm).....	298
13.3.3	Serial Mode Register mn (SMRmn).....	299
13.3.4	Serial Communication Operation Configuration Register mn (SCRmn)	301
13.3.5	Serial Data Register mn (SDRmn)	303
13.3.6	Serial Flag Clear Trigger Register mn (SIRmn)	305
13.3.7	Serial Status Register mn (SSRmn).....	306
13.3.8	Serial Channel Start Register m (SSm)	308
13.3.9	Serial Channel Stop Register m (STm)	309
13.3.10	Serial Channel Enable Status Register m (SEm).....	310
13.3.11	Serial Output Enable Register m(SOEm).....	311
13.3.12	Serial Output Register m (SOM)	312
13.3.13	Serial Output Level Register m (SOLm).....	313
13.3.14	Input Switching Control Register (ISC).....	315
13.3.15	Noise Filter Enable Register 0 (NFEN0).....	316
13.3.16	Registers Controlling Port Functions of Serial Input/Output Pins	317
13.4	Run Stop Mode.....	318
13.4.1	Stopping the Operation by Units	318
13.4.2	Stopping the Operation by Channels.....	319
13.5	3-Wire Serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21) Communication Operation..	320
13.5.1	Master Transmission.....	321
13.5.2	Master Reception.....	330
13.5.3	Master Transmission and Reception.....	338
13.5.4	Slave Transmission.....	346
13.5.5	Slave Reception.....	354
13.5.6	Slave Transmission and Reception.....	360
13.5.7	Calculation of Transfer Clock Frequency.....	369
13.5.8	Procedure for Handling Errors During 3-Wire Serial I/O Communication (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21)	371
13.6	Operation of Clock-Synchronous Serial Communication with Slave Selection Input Function	372
13.6.1	Slave Transmission.....	375

13.6.2	Slave Reception.....	385
13.6.3	Slave Transmission and Reception.....	392
13.6.4	Calculation of Transfer Clock Frequency.....	402
13.6.5	Procedure for Handling Errors in Clock-Synchronous Serial Communication with Slave Select Input Function	403
13.7	Operation of UART (UART0~UART2) Communication.....	404
13.7.1	UART Transmission	405
13.7.2	UART Reception	414
13.7.3	Calculation of Baud Rate	421
13.7.4	Handling Steps When an Error Occurs During UART (UART0~UART2) Communication	425
13.8	Operation of LIN Communication	426
13.8.1	LIN Transmission.....	426
13.8.2	LIN Reception	429
13.9	Simplified I ² C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) Communication Operation	434
13.9.1	Address Field Transmission	435
13.9.2	Data Transmission	440
13.9.3	Data Reception	443
13.9.4	Generation of Stop Condition.....	447
13.9.5	Calculation of Transfer Rate.....	448
13.9.6	Processing Steps When an Error Occurs During Simplified I ² C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) Communication	450
Chapter 14	Serial Interface IICA	451
14.1	Functions of Serial Interface IICA	451
14.2	Structure of Serial Interface IICA	454
14.3	Registers for Controlling Serial Interface IICA.....	457
14.3.1	Peripheral Enable Register 0 (PER0)	458
14.3.2	IICA Control Register n0 (IICCTLn0).....	459
14.3.3	IICA Status Register n (IICSn).....	463
14.3.4	IICA Flag Register n (IICFn)	466
14.3.5	IICA Control Register n1 (IICCTLn1).....	468
14.3.6	IICA Low Level Width Setting Register n (IICWLn)	470
14.3.7	IICA High Level Width Setting Register n (IICWHn)	470
14.3.8	Registers for Controlling Pin Port Functions of IICA.....	470
14.4	Functions of I ² C Bus Mode	471
14.4.1	Pin Structure	471
14.4.2	Setting Transfer Clock via IICWLn and IICWHn Registers	472
14.5	Definition and Control Method of I ² C bus	474
14.5.1	Start Condition	475
14.5.2	Address.....	476
14.5.3	Transfer Direction Specification	476
14.5.4	Acknowledge (ACK).....	477
14.5.5	Stop Condition.....	478
14.5.6	Wait	479
14.5.7	Wait Release Methods	481
14.5.8	Generation Timing and Waiting Control of Interrupt Requests (INTIICAn)	482
14.5.9	Detection for Address Matching.....	484
14.5.10	Error Detection.....	484

14.5.11	Extension Code	485
14.5.12	Arbitration	486
14.5.13	Wake-Up Function.....	488
14.5.14	Communication Reservation	491
14.5.15	Cautions.....	495
14.5.16	Communication Operation	496
14.5.17	Timing of I ² C Interrupt Request (INTIICAn) Generation.....	504
14.6	Timing Diagrams	525
Chapter 15	IrDA	540
15.1	Functions of IrDA	540
15.2	Registers for Controlling IrDA.....	541
15.2.1	Peripheral Enable Register 0 (PER0)	541
15.2.2	IrDA Control Register (IRCR)	542
15.3	IrDA Operation.....	543
15.3.1	IrDA Communication Operation Steps	543
15.3.2	Transmission.....	544
15.3.3	Reception.....	544
15.3.4	Selection of High-Level Pulse Width	545
15.4	Notes on Using IrDA.....	546
Chapter 16	Enhanced DMA.....	547
16.1	Functions of DMA.....	547
16.2	Structure of DMA.....	549
16.3	Registers for Controlling DMA.....	550
16.3.1	Allocation of DMA Control Data Zone and DMA Vector Table Zone.....	551
16.3.2	Control Data Allocation.....	552
16.3.3	Vector Table.....	554
16.3.4	Peripheral enable register 1 (PER1)	556
16.3.5	DMA Control Register j (DMACRj) (j=0~23).....	557
16.3.6	DMA Block Size Register j (DMBLSj) (j=0~23)	558
16.3.7	DMA Transfer Count Register j (DMACTj) (j=0~23)	559
16.3.8	DMA Transfer Count Reload Register j (DMRLDj) (j=0~23).....	560
16.3.9	DMA Source Address Register j (DMSARj) (j=0~23).....	561
16.3.10	DMA Target Address Register j (DMDARj) (j=0~23).....	561
16.3.11	DMA Start Enable Register i (DMAENi) (i=0~2)	562
16.3.12	DMAENi Set Register (DMSETi).....	564
16.3.13	DMAENi Reset Register (DMCLRi)	564
16.3.14	DMA Base Address Register (DMABAR).....	565
16.4	Operation of DMA	566
16.4.1	Boot Source	566
16.4.2	Normal Mode	567
16.4.3	Repeat Mode	570
16.4.4	Chain Transfer	573
16.5	Cautions When Using DMA.....	575
16.5.1	DMA Control Data and Vector Table Settings.....	575
16.5.2	Assignment of DMA Control Data Area and DMA Vector Table Area	575
16.5.3	Number of Execution Clocks for DMA	576
16.5.4	Response Time for DMA	577

16.5.5	Start Source for DMA	577
16.5.6	Operation in Standby Mode	578
Chapter 17 Linkage Controller (EVENTC).....		579
17.1	Functions of EVENTC	579
17.2	Structure of EVENTC	579
17.3	Control Registers.....	580
17.3.1	Output Target Selection Register n (ELSELRn) (n=00~14).....	581
17.4	Operation of EVENTC	583
Chapter 18 Interrupt Function		584
18.1	Types of Interrupt Function	584
18.2	Interrupt Source and Structure.....	585
18.3	Registers for Controlling Interrupt Function	589
18.3.1	Interrupt Request Flag Register (IF00~IF31)	589
18.3.2	Interrupt Mask Flag Register (MK00~MK31)	590
18.3.3	External Interrupt Rising Edge Enable Register (EGP0), External Interrupt Falling Edge Enable Register (EGN0).....	592
18.4	Operation of Interrupt Handling	593
18.4.1	Acceptance of Maskable Interrupt Requests	593
18.4.2	Acceptance of Non-Maskable Interrupt Requests	593
Chapter 19 Standby Function		594
19.1	Standby Function.....	594
19.2	Sleep Mode.....	595
19.2.1	Sleep Mode Configuration	595
19.2.2	Release of Sleep Mode	598
19.3	Deep Sleep Mode	599
19.3.1	Deep Sleep Mode Configuration	599
19.3.2	Release of Deep Sleep Mode.....	601
Chapter 20 Reset Function.....		603
20.1	Registers for Confirming the Reset Source	607
20.1.1	Reset Control Flag Register (RESF)	607
Chapter 21 Power-On Reset Circuit		610
21.1	Functions of Power-On Reset Circuit.....	610
21.2	Structure of Power-On Reset Circuit.....	611
21.3	Operation of Power-On Reset Circuit.....	612
Chapter 22 Voltage Detection Circuit.....		615
22.1	Functions of Voltage Detection Circuit	615
22.2	Structure of Voltage Detection Circuit	616
22.3	Registers for Controlling Voltage Detection Circuit.....	617
22.3.1	Voltage Detection Register (LVIM).....	617
22.3.2	Voltage Detection Level Register (LVIS).....	618
22.4	Operation of Voltage Detection Circuit	621
22.4.1	Settings When Used as Reset Mode	621
22.4.2	Settings When Used as Interrupt Mode.....	623
22.4.3	Settings When Used as Interrupt & Reset Mode	625
22.5	Cautions for Voltage Detection Circuit.....	630

Chapter 23	Safety Functions.....	632
23.1	Overview of Safety Functions	632
23.2	Registers Used by Safety Functions	633
23.3	Operation of Safety Functions	634
23.3.1	Flash Memory CRC Operation Function (High-Speed CRC)	634
23.3.2	CRC Operation Function (General-Purpose CRC)	637
23.3.3	RAM Parity Error Detection Function	640
23.3.4	SFR Guard Function.....	642
23.3.5	Frequency Detection Function	643
23.3.6	A/D Test Function	644
23.3.7	Digital Output Signal Level Detection Function for Input/Output Pins	646
23.3.8	Product Unique ID Register	647
Chapter 24	Temperature Sensor.....	648
24.1	Functions of Temperature Sensors.....	648
24.2	Registers Used by Temperature Sensors.....	648
24.2.1	Temperature Sensor Calibration Data Register TSCDR1.....	648
24.2.2	Temperature Sensor Calibration Data Register TSCDR2.....	648
24.3	Instructions for Using Temperature Sensors.....	649
24.3.1	How to Use the Temperature Sensor	649
24.3.2	Methods for Using the Temperature Sensor	650
Chapter 25	Option Byte.....	651
25.1	Functions of Option Bytes	651
25.1.1	User Option Byte (000C0H~000C2H/010C0H~010C2H).....	651
25.1.2	Flash Memory Data Protection Option Byte (000C3H/010C3H, 500004H~50000BH)	652
25.2	Format of User Option Byte	653
25.3	Format of Flash Memory Data Protection Option Byte	659
Chapter 26	Flash Control.....	660
26.1	Description of Flash Control	660
26.2	Structure of Flash Memory.....	660
26.3	Registers for Controlling Flash	661
26.3.1	Flash Write Protect Register (FLPROT)	661
26.3.2	Flash Operation Control Register (FLOPMD1, FLOPMD2).....	662
26.3.3	Flash Erase Control Register (FLERMD)	662
26.3.4	Flash Status Register (FLSTS)	663
26.3.5	Flash Chip Erase Time Control Register (FLCERCNT)	663
26.3.6	Flash Sector Erase Time Control Register (FLSERCNT).....	664
26.3.7	Flash Write Time Control Register (FLPROCNT).....	665
26.3.8	Flash Mode Time Control Register (FLNVSCNT/FLPRVCNT/FLERVCNT).....	666
26.4	How to Operate the Flash	668
26.4.1	Sector Erase.....	668
26.4.2	Chip Erase	669
26.4.3	Word Program	669
26.5	Flash Read	669
26.6	Cautions for Flash Operation.....	669

Appendix Version History.....	670

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.

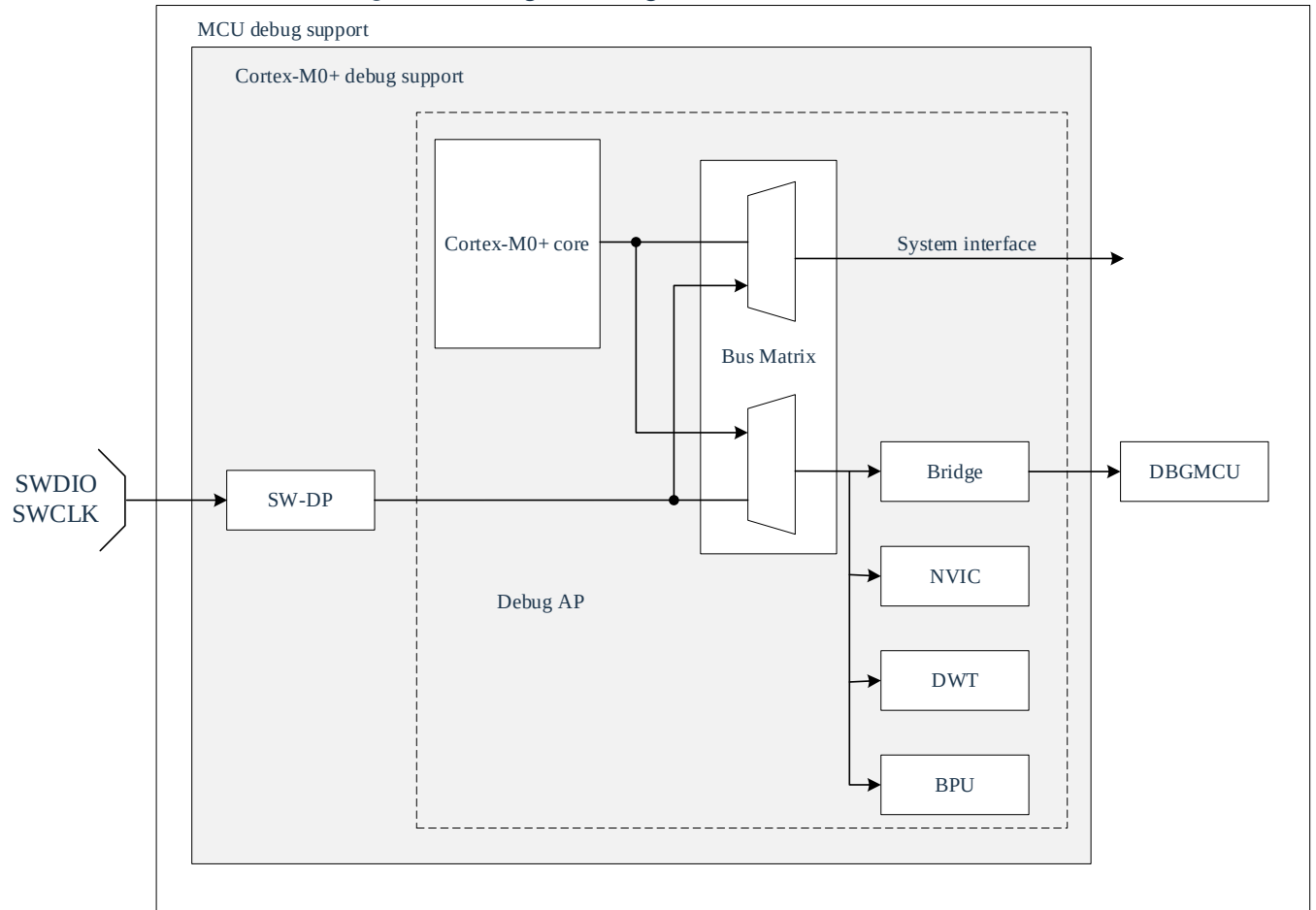
1.2 Cortex-M0+ Core Features

- The ARM Cortex-M0+ processor is a 32-bit RISC core with a 2-stage pipeline that only supports privileged mode
- 32-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

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	P21
SWDIO	Serial data input/output	P20

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

Bit No.	31	30	29	28	27	26	25	24
DBGSTOPCR	-	-	-	-	-	-	-	SWDIS
Default value	0	0	0	0	0	0	0	0

Bit No.	23	22	21	20	19	18	17	16
DBGSTOPCR	-	-	-	-	-	-	-	-
Default value	0	0	0	0	0	0	0	0

Bit No.	15	14	13	12	11	10	9	8
DBGSTOPCR	-	-	-	-	-	-	-	-
Default value	0	0	0	0	0	0	0	0

Bit No.	7	6	5	4	3	2	1	0
DBGSTOPCR	-	-	-	-	-	RESMSK	FRZEN1	FRZEN0
Default value	0	0	0	0	0	1	0	0

SWDIS	SWD debug interface disable
0	SWD debug interface enable. P20 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. Clear OCDMODE, and P20 can be used as a GPIO

RESMSK	In debug mode, Flash blank address access error reset, watchdog reset, SRAM parity check reset, and LVI reset are masked.
0	Reset is not masked.
1	Reset is masked.

FRZEN1	In the state where the debugger is connected and the CPU is in the debug state (HALED=1), the communication system peripheral module action/stop ^{Note 2}
0	Peripheral acts
1	Peripheral stops

FRZEN0	In the state where the debugger is connected and the CPU is in the debug state (HALED=1), the timer is the peripheral module action/stop ^{Note 1}
0	Peripheral acts
1	Peripheral stops

Note 1: The timer system peripheral module of the product includes: Universal timer unit Timer4.

Note 2: The communication system peripheral module of this product includes: Communication Serial Communication Unit, Serial IICA.

1.5 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

The ports are controlled through the following registers.

- Port mode register (PMxx)
- Port register (Pxx)
- Pull-up resistor selection register (PUxx)
- Pull-down resistor selection register (PDxx)
- Port output mode register (POMx)
- Port mode control register (PMCxx)
- Port set control register (PSETxx)
- Port clear control register (PCLRxx)
- Port read register (PREADxx)

When used as GPIO, the following functions are supported:

- Input floating
- Input pull-up
- Input pull-down
- Analog channel
- Open-drain output
- Push-pull output

The registers and bits assigned to each port vary depending on the port's hardware characteristics and the product. Please refer to Table 2-1 for the PMxx, Pxx, PUxx, PDxx, POMxx, PMCxx, PSETxx, PCLRxx, PREADxx registers and their bits allocated to each product and the datasheets of each product series.

Taking the 24-pin product as an example, the default initial functions of each port after power-on are as follows:

Pin name	Default function
P00	External reset input
P01	Digital IO
P02	Digital IO
P10	Digital IO
P11	Digital IO
P12	Digital IO
P13	Digital IO
P20	Digital IO
P21	Digital IO
P22	Analog IO
P23	Analog IO
P24	Analog IO
P25	Analog IO
P26	Analog IO
P30	Analog IO
P31	Analog IO
P32	Analog IO
P33	Analog IO
P34	Analog IO
P35	Digital IO
P36	Digital IO
P37	Digital IO

Ports P00~P02 and P10~P13 can only be used as digital channels.

Ports P20~P26 and P30~P37 can be used both as digital channels and as analog channels. When PMCxx is configured as 1, the port is used as an analog channel. All analog functions of this product are fixed to specific ports and do not support redirection configuration, as described in the datasheets of each product series. When PMCxx is configured as 0, the port is used as a digital channel. The digital multiplex functions of each port can be switched by configuring the PxxCFG register. For detailed configuration methods, please refer to Table 2-2 Port Alternate Function Configuration Method.

2.2 Port Alternate Functions

The peripheral I/O functions can be redirected to any port except P00 through software programming by configuring the PMxx, PMCxx, and PxxCFG registers. Please refer to 2.3.10 Port Alternate Function Configuration Register (PxxCFG) and the datasheets of each product series.

2.3 Registers for Controlling Port Functions

The ports are controlled through the following registers.

- Port mode register (PMxx)
- Port register (Pxx)
- Pull-up resistor selection register (PUxx)
- Pull-down resistor selection register (PDxx)
- Port output mode register (POMx)
- Port mode control register (PMCxx)
- Port set control register (PSETxx)
- Port clear control register (PCLRxx)
- Port read register (PREADxx)
- Port alternate function configuration register (PxxCFG)

Note: Which registers and bits are included depends on the product. For registers and bits mounted on each product, see Table 2-1.

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

Table 2-1 PMxx, Pxx, PUxx, PDxx, POMxx, PMCxx, PSETxx, PCLRxx, PREADxx registers and bits

Port		Bit name										24-pin	20-pin	16-pin
		PMxx register	Pxx register	PUxx register	PDxx register	POMxx register	PMCxx register	PSETxx register	PCLRxx register	PREADxx register	PxxCFG register			
Port 0	0	—	—	PU00	—	—	—	PSET00	PCLR00	PREAD00	—	○	○	○
	1	PM01	P01	—	—	POM01	—	PSET01	PCLR01	PREAD01	P01CFG	○	○	○
	2	PM02	P02	—	—	POM02	—	PSET02	PCLR02	PREAD02	P02CFG	○	○	○
Port 1	0	PM10	P10	PU10	—	POM10	—	PSET10	PCLR10	PREAD10	P10CFG	○	○	○
	1	PM11	P11	PU11	—	POM11	—	PSET11	PCLR11	PREAD11	P11CFG	○	○	—
	2	PM12	P12	—	—	POM12	—	PSET12	PCLR12	PREAD12	P12CFG	○	○	○
	3	PM13	P13	—	—	POM13	—	PSET13	PCLR13	PREAD13	P13CFG	○	○	○
Port 2	0	PM20	P20	PU20	PD20	POM20	PMC20	PSET20	PCLR20	PREAD20	P20CFG	○	○	○
	1	PM21	P21	PU21	PD21	POM21	PMC21	PSET21	PCLR21	PREAD21	P21CFG	○	○	○
	2	PM22	P22	PU22	PD22	POM22	PMC22	PSET22	PCLR22	PREAD22	P22CFG	○	—	—
	3	PM23	P23	PU23	PD23	POM23	PMC23	PSET23	PCLR23	PREAD23	P23CFG	○	—	—
	4	PM24	P24	PU24	PD24	POM24	PMC24	PSET24	PCLR24	PREAD24	P24CFG	○	○	—
	5	PM25	P25	PU25	PD25	POM25	PMC25	PSET25	PCLR25	PREAD25	P25CFG	○	○	○
	6	PM26	P26	PU26	PD26	POM26	PMC26	PSET26	PCLR26	PREAD26	P26CFG	○	○	○
Port 3	0	PM30	P30	PU30	PD30	POM30	PMC30	PSET30	PCLR30	PREAD30	P30CFG	○	○	○
	1	PM31	P31	PU31	PD31	POM31	PMC31	PSET31	PCLR31	PREAD31	P31CFG	○	○	○
	2	PM32	P32	PU32	PD32	POM32	PMC32	PSET32	PCLR32	PREAD32	P32CFG	○	○	○
	3	PM33	P33	PU33	PD33	POM33	PMC33	PSET33	PCLR33	PREAD33	P33CFG	○	—	—
	4	PM34	P34	PU34	PD34	POM34	PMC34	PSET34	PCLR34	PREAD34	P34CFG	○	—	—
	5	PM35	P35	PU35	PD35	POM35	PMC35	PSET35	PCLR35	PREAD35	P35CFG	○	○	○
	6	PM36	P36	PU36	PD36	POM36	PMC36	PSET36	PCLR36	PREAD36	P36CFG	○	○	—
	7	PM37	P37	PU37	PD37	POM37	PMC37	PSET37	PCLR37	PREAD37	P37CFG	○	○	—

2.3.1 Port Mode Register (PMxx)

This is a register that sets the port input/output on a bit basis. After the reset signal is generated, the values of these registers become “FFH”. When the port pins are used for alternate functions, this register must be set, and the settings should refer to 2.3.10 Port Alternate Function Configuration Register (PxxCFG).

Register address = base address + offset; the base address of the PM register is 0x40040000, and the offset is shown in the figure below.

Figure 2-1 Format of port mode register

Symbol	7	6	5	4	3	2	1	0	Offset address	After reset	R/W
PM0	1	1	1	1	1	PM02	PM01	1	0x320	FFH	R/W
PM1	1	1	1	1	PM13	PM12	PM11	PM10	0x321	FFH	R/W
PM2	1	PM26	PM25	PM24	PM23	PM22	PM21	PM20	0x322	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	0x323	FFH	R/W

PMmn	Selection of input/output modes for the Pmn port (m=0~3, n=0~7)
0	Output mode (used as an output port (output buffer ON)).
1	Input mode (used as an input port (output buffer OFF)).

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

2.3.2 Port register (Pxx)

This is the register that sets the value of the port's output latch in 1-bit units. Reading this register in input mode yields the port level, and reading in output mode outputs latch's value. After a reset signal is generated, the value of the register becomes 00H.

Register address = base address + offset address; the base address of the port register is 0x40040000, and the offset address is shown in the figure below.

Figure 2-2 Format of port register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	P02	P01	P00 ^{Note1}	0x300	00H ^{Note1}	R/W
P1	0	0	0	0	P13	P12	P11	P10	0x301	00H (output latch)	R/W
P2	0	P26	P25	P24	P23	P22	P21	P20	0x302	00H (output latch)	R/W
P3	P37	P36	P35	P34	P33	P32	P31	P30	0x303	00H (output latch)	R/W

Note 1: P00 is read-only because the P00 pin only has input mode.

Pmn	(m=0~3, n=0~7)	
	Control of output data (output mode)	Reading of input data (input mode)
0	Output 0.	Input low.
1	Output 1.	Input high.

2.3.3 Pull-Up Resistor Selection Register (PUxx)

This is the internal pull-up resistor selection register. By setting this register, a port in input mode (PMmn=1) can be pulled up in 1-bit units by using an internal pull-up resistor and setting the POMmn to 0. For ports set to output mode, independent of the setting of the pull-up resistor selection register, no internal pull-up resistors are connected. The same applies when used as an output port for alternate functions or when set to analog functions.

Register address = base address + offset address; the base address of the PU register is 0x40040000, and the offset address is shown in the figure below.

Figure 2-3 Format of pull-up resistor selection register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	0	0	PU00	0x030	01H	R/W
PU1	0	0	0	0	0	0	PU11	PU10	0x031	00H	R/W
PU2	0	PU26	PU25	PU24	PU23	PU22	PU21	PU20	0x032	03H	R/W
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30	0x033	E0H	R/W

PUmn	Selection of the internal pull-up resistor for the Pmn port (m=0~3, n=0~7)
0	Internal pull-up resistors are not connected.
1	Connect an internal pull-up resistor.

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

2.3.4 Pull-Down Resistor Selection Register (PDxx)

This is the internal pull-down resistor selection register. By setting this register, ports in input mode (PMmn=1) can be pulled down in 1-bit units by using an internal pull-up resistor and setting the POMmn to 0. For ports set to output mode, independent of the setting of the pull-down resistor selection register, no internal pull-down resistor is connected. The same applies when used as an output port for alternate function or when set to analog function.

After a reset signal is generated, the value of these registers becomes 0000H.

Register address = base address + offset address; the base address of the PD register is 0x40040000, and the offset address is shown in the figure below.

Figure 2-4 Format of pull-down resistor selection register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PD2	0	PD26	PD25	PD24	PD23	PD22	PD21	PD20	0x042	00H	R/W
PD3	PD37	PD36	PD35	PD34	PD33	PD32	PD31	PD30	0x043	00H	R/W
	PDmn	Selection of the internal pull-down resistor for the Pmn port (m=2, 3, n=0~7).									
	0	No internal pull-down resistors are connected.									
	1	Connect an internal pull-down resistor.									

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

2.3.5 Port Output Mode Register (POMxx)

This is the register that sets the output mode in 1-bit units. When communicating serially with external devices of different potentials and I²C with external devices of the same potential, the N-channel open-drain output (V_{DD}/EV_{DD} voltage tolerance) mode can be selected for the SDAxx port.

After a reset signal is generated, the value of these registers becomes 0000H.

Register address = base address + offset address; The base address of the POM register is 0x40040000, and the offset address is shown in the figure below.

Caution: For the bit that sets the N-channel open-drain output mode (V_{DD}/EV_{DD} voltage tolerance) mode (POMmn = 1), the internal pull-up/pull-down resistors are not connected.

Figure 2-5 Format of port output mode register

Symbol	7	6	5	4	3	2	1	0	Offset address	After reset	R/W
POM0	0	0	0	0	0	POM02	POM01	0	0x050	00H	R/W
POM1	0	0	0	0	POM13	POM12	POM11	POM10	0x051	00H	R/W
POM2	0	POM26	POM25	POM24	POM23	POM22	POM21	POM20	0x052	00H	R/W
POM3	POM37	POM36	POM35	POM34	POM33	POM32	POM31	POM30	0x053	00H	R/W

POMmn	Selection of the output mode of the Pmn port (m=0~3, n=0~7)
0	Normal output mode
1	N-channel open-drain output mode (V_{DD}/EV_{DD} voltage tolerance)

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

2.3.6 Port Mode Control Register (PMCxx)

The PMC registers set the port in 1-bit units for use as a digital input/output or as an analog channel. When a port pin is used for an alternate function, the port must be set to digital mode.

Register address = base address + offset address; the base address of the PMC register is 0x40040000, and the offset address is shown in the figure below.

Figure 2-6 Format of port mode control register

Symbol	7	6	5	4	3	2	1	0	Offset address	After reset	R/W
PMC2	1	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20	0x062	fcH	R/W
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30	0x063	1fH	R/W

PMCMn	Selection of digital input/output or analog input of Pmn port (m=2, 3, n=0~7)
0	Digital inputs/outputs (alternate functions other than analog inputs)
1	Analog inputs

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

2.3.7 Port Set Control Register (PSETxx)

This is the register that sets the output latch in 1-bit units. After a reset signal is generated, the value of the register becomes 00H.

Register address = base address + offset address; The base address of the register is 0x40040000, and the offset address is shown in the figure below.

Figure 2-7: Format of port set control register

Symbol	7	6	5	4	3	2	1	0	Offset address	After reset	R/W
PSET0						PSET02	PSET01	PSET00	0x080	00H	W
PSET1					PSET13	PSET12	PSET11	PSET10	0x081	00H	W
PSET2		PSET26	PSET25	PSET24	PSET23	PSET22	PSET21	PSET20	0x082	00H	W
PSET3	PSET37	PSET36	PSET35	PSET34	PSET33	PSET32	PSET31	PSET30	0x083	00H	W

PSETmn	Pmn port set control (m=0~3, n=0~7)
0	No operations
1	Set the corresponding Pmn to 1

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

2.3.8 Port Clear Control Register (PCLRxx)

This is the register that sets the output latch in 1-bit units. After a reset signal is generated, the value of the register becomes “00H”.

Register address = base address + offset address; The base address of the register is 0x40040000, and the offset address is shown in the figure below.

Figure 2-8: Format of port clear control register

Symbol	7	6	5	4	3	2	1	0	Offset address	After reset	R/W
PCLR0						PCLR02	PCLR01	PCLR00	0x090	00H	W
PCLR1					PCLR13	PCLR12	PCLR11	PCLR10	0x091	00H	W
PCLR2		PCLR26	PCLR25	PCLR24	PCLR23	PCLR22	PCLR21	PCLR20	0x092	00H	W
PCLR3	PCLR37	PCLR36	PCLR35	PCLR34	PCLR33	PCLR32	PCLR31	PCLR30	0x093	00H	W

PCLRmn	Pmn port clear control (m=0~3, n=0~7)
0	No operations
1	Clear the corresponding Pmn

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

2.3.9 Port Read Register (PREADxx)

This is a read-only register that can be read to get the port level when the port is a normal digital GPIO.

Register address = base address + offset address; the base address of the port register is 0x40040000, and the offset address is shown in the figure below.

Figure 2-9: Format of port read register

Symbol	7	6	5	4	3	2	1	0	Offset address	After reset	R/W
PREAD0						PREAD02	PREAD01	PREAD00	0x0A0	00H	R
PREAD1					PREAD13	PREAD12	PREAD11	PREAD10	0x0A1	00H	R
PREAD2		PREAD26	PREAD25	PREAD24	PREAD23	PREAD22	PREAD21	PREAD20	0x0A2	00H	R
PREAD3	PREAD37	PREAD36	PREAD35	PREAD34	PREAD33	PREAD32	PREAD31	PREAD30	0x0A3	00H	R
PREADmn	Output/input mode (m=0~3, n=0~7)										
0	The port is low.										
1	The port is high.										

Remark: The readout value of the register after reset depends on the status of each port.

2.3.10 Port Alternate Function Configuration Register (PxxCFG)

The Port Alternate Function Configuration Register (PxxCFG) allows the redirection of digital functions of peripheral I/O to ports other than P00. The reset value of the Port Alternate Function Configuration Register is 00H. The addresses of the port configuration registers are referenced in the figure below.

Register Address = Base Address + Offset.

Figure 2-10 Format of port alternate function configuration registers (PxxCFG)

Base address: 0x40040800

Register name	Offset address	R/W	Reset value
P01CFG	0x002	R/W	00H
P02CFG	0x004	R/W	00H
P10CFG	0x010	R/W	00H
P11CFG	0x012	R/W	00H
P12CFG	0x014	R/W	00H
P13CFG	0x016	R/W	00H
P20CFG	0x020	R/W	00H
P21CFG	0x022	R/W	00H
P22CFG	0x024	R/W	00H
P23CFG	0x026	R/W	00H
P24CFG	0x028	R/W	00H
P25CFG	0x02A	R/W	00H
P26CFG	0x02C	R/W	00H
P30CFG	0x030	R/W	00H
P31CFG	0x032	R/W	00H
P32CFG	0x034	R/W	00H
P33CFG	0x036	R/W	00H
P34CFG	0x038	R/W	00H
P35CFG	0x03A	R/W	00H
P36CFG	0x03C	R/W	00H
P37CFG	0x03E	R/W	00H

Configuration instructions:

- When using the port's digital alternate functions, the port must be configured in digital mode (PMCxx=0).
- For alternated input functions, the port must be configured in input mode (PMxx=1).
- For alternated output functions, the port must be configured in output mode (push-pull or open-drain) (PMxx=0).
- For bidirectional alternated functions, the port must be configured in output mode (push-pull or open-drain) (PMxx=0). In this case, the input driver is configured as a floating input mode.
- When using the GPIO or alternated function of P01 or P02 ports, ensure that the X1 oscillator mode and external clock input mode are not enabled. Refer to 4.3.1 Clock Operation Mode Control Register (CMC).
- When using the GPIO or alternated function of P12 or P13 ports, ensure that the XT1 oscillator mode and external clock input mode are not enabled. Refer to 4.3.1 Clock Operation Mode Control Register (CMC).
- P00 has no port alternating function and can only be used as a GPIO input. When using the GPIO input function of P00, the RSTM register must first be configured to mask its external reset function. Refer to 2.5.1 P00 Port (default RESETB).
- The same alternated input function cannot be configured on different ports simultaneously, and different alternated functions cannot be configured on the same port at the same time.

Table 2-2 Configuration methods for port alternate outputs

Function name	Input/output	PxxCFP	PMCxx	PMxx	POMxx	Pxx	Remark
Analog function	Input/output	6'h00	1	1	x	x	All analog functions are directed to fixed ports and cannot be configured. Please refer to the datasheet of each product series for more details.
GPIO	Input	6'h00	0	1	0	x	P00 can only be used as an external reset input or GPIO input.
	Output		0	0	0	0/1	
	N-channel open-drain output		0	0	1	0/1	
INTP0	Input	6'h02	0	1	0	x	
INTP1	Input	6'h03	0	1	0	x	
INTP2	Input	6'h04	0	1	0	x	
INTP3	Input	6'h05	0	1	0	x	
TI00	Input	6'h06	0	1	0	x	
TI01	Input	6'h07	0	1	0	x	
TI02	Input	6'h08	0	1	0	x	
TI03	Input	6'h09	0	1	0	x	
TI10	Input	6'h0a	0	1	0	x	
TI11	Input	6'h0b	0	1	0	x	
TI12	Input	6'h0c	0	1	0	x	
TI13	Input	6'h0d	0	1	0	x	
TO00	Output	6'h0e	0	0	0	x	
TO01	Output	6'h0f	0	0	0	x	
TO02	Output	6'h10	0	0	0	x	
TO03	Output	6'h11	0	0	0	x	
TO10	Output	6'h12	0	0	0	x	
TO11	Output	6'h13	0	0	0	x	
TO12	Output	6'h14	0	0	0	x	
TO13	Output	6'h15	0	0	0	x	
SCLA0	Bidirectional	6'h16	0	0	1 ^{Note1}	x	Note 1: When used as an IICA port, POMxx is automatically set to 1 and does not require software configuration.
SDAA0	Bidirectional	6'h17	0	0	1 ^{Note1}	x	
CLKBUZ0	Output	6'h18	0	0	0	x	
CLKBUZ1	Output	6'h19	0	0	0	x	
VCOUT0	Output	6'h1a	0	0	0	x	
VCOUT1	Output	6'h1b	0	0	0	x	
RTC1HZ	Output	6'h1c	0	0	0	x	
SS00	Input	6'h1f	0	1	0	x	
SCLK00/SCL00	Input	6'h20	0	1	0	x	
	Output		0	0	0	x	
SCLK01/SCL01	Input	6'h21	0	1	0	x	
	Output		0	0	0	x	
SCLK10/SCL10	Input	6'h22	0	1	0	x	
	Output		0	0	0	x	
SCLK11/SCL11	Input	6'h23	0	1	0	x	
	Output		0	0	0	x	
SCLK20/SCL20	Input	6'h24	0	1	0	x	
	Output		0	0	0	x	
SCLK21/SCL21	Input	6'h25	0	1	0	x	
	Output		0	0	0	x	
SDI00/RxD0	Input	6'h26	0	1	0	x	
SDA00	Bidirectional	6'h26	0	0	1 ^{Note2}	x	Note 2: Software configuration of POM is required.

SDI01	Input	6'h27	0	1	0	x	
SDA01	Bidirectional	6'h27	0	0	1 ^{Note2}	x	
SDI10/RxD1	Input	6'h28	0	1	0	x	
SDA10	Bidirectional	6'h28	0	0	1 ^{Note2}	x	
SDI11	Input	6'h29	0	1	0	x	
SDA11	Bidirectional	6'h29	0	0	1 ^{Note2}	x	
SDI20/RxD2	Input	6'h2a	0	1	0	x	
SDA20	Bidirectional	6'h2a	0	0	1 ^{Note2}	x	
SDI21	Input	6'h2b	0	1	0	x	
SDA21	Bidirectional	6'h2b	0	0	1 ^{Note2}	x	
SDO00/TxD0	Output	6'h2c	0	0	0	x	
SDO01	Output	6'h2d	0	0	0	x	
SDO10/TxD1	Output	6'h2e	0	0	0	x	
SDO11	Output	6'h2f	0	0	0	x	
SDO20/TxD2	Output	6'h30	0	0	0	x	
SDO21	Output	6'h31	0	0	0	x	

2.4 Handling of Unused Pins

The handling of each unused port is as following Table 2-3.

Table 2-3 Handling of each unused pin

Pin name	Input/Output	Recommended connection method when not in use
P00	Input	V _{DD} is connected directly or through a resistor. (default: RESETB)
P01~P02	Input/output	Input: V _{DD} or V _{SS} is connected independently by a resistor.
P10~P13		Output: Set to open circuit.
P20~P26		Input: EV _{DD} or EV _{SS} is connected independently by a resistor.
P30~P37		Output: Set to open circuit.

Remark: For products without EV_{DD}, EV_{SS} pins, EV_{DD} must be replaced with V_{DD} and EV_{SS} with V_{SS}.

2.5 Special Pin Configuration

2.5.1 Port P00 (Default is RESETB)

By default, P00 is used as an external reset input pin, and a system reset occurs when it is low. If the GPIO input function of P00 is to be used, the RSTM register must first be configured to mask its external reset function.

Base address: 0x40020400

Symbol	7	6	5	4	3	2	1	0	Offset address	After reset	R/W
RSTM	0	0	0	0	0	0	0	RSTM	0x007	00H	R/W

RSTM	P00 pin external reset function mask
0	P00 as an external reset pin
1	P00 as a GPIO input pin

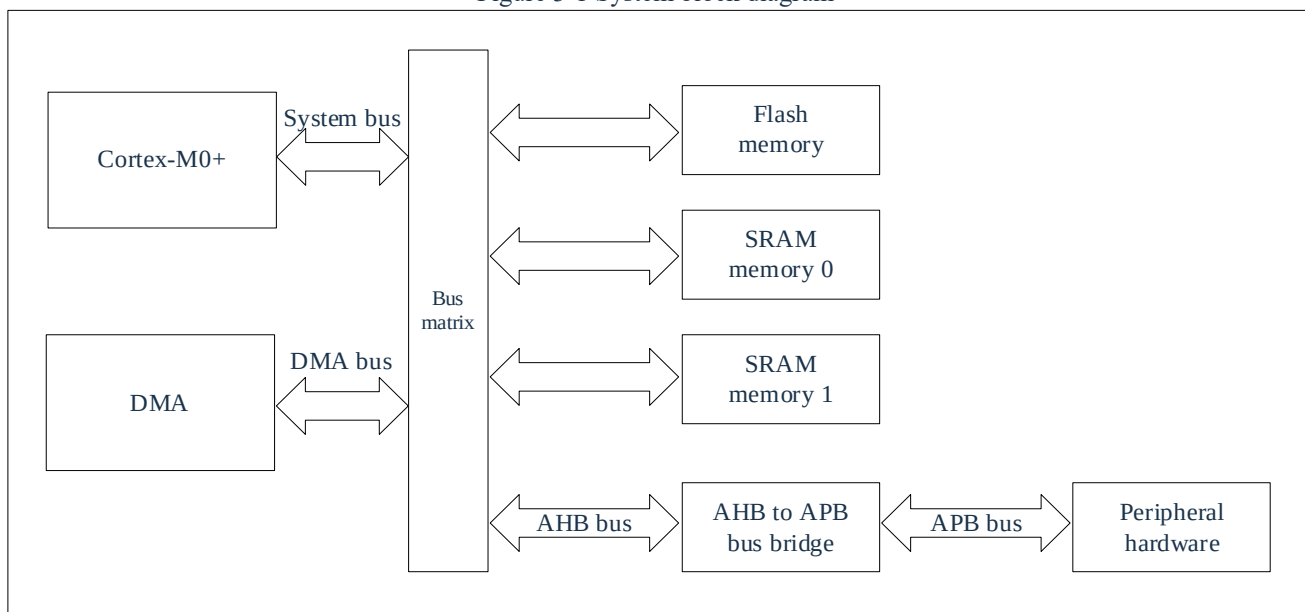
Chapter 3 System Architecture

3.1 Overview

This product system consists of the following components:

- 2 AHB buses (Master):
 - Cortex-M0+
 - Enhanced DMA
- 5 AHB buses (Slaves):
 - Flash memory
 - SRAM memory 0
 - SRAM memory 1
 - AHB to APB bridge, contains all APB interface peripherals

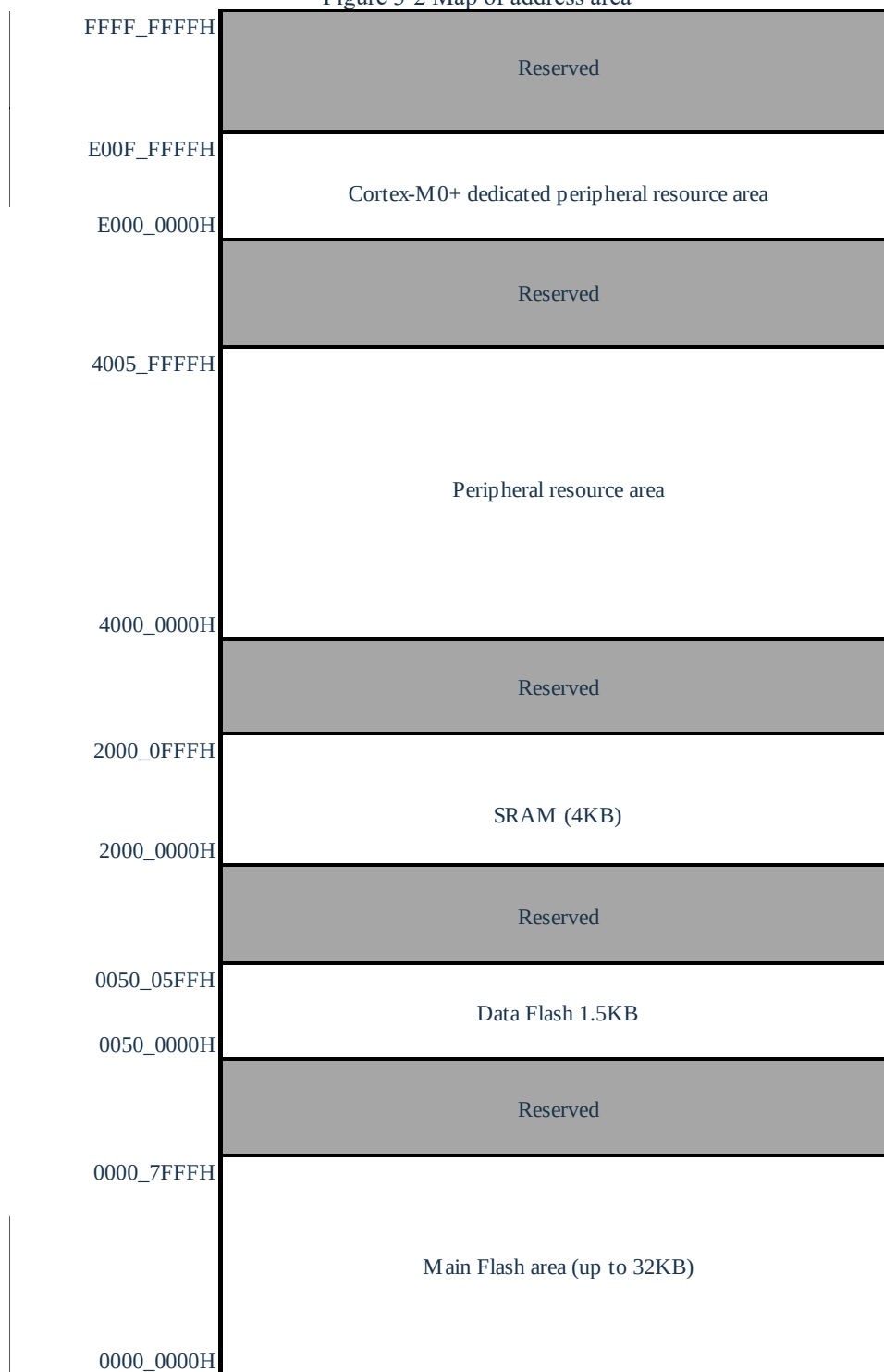
Figure 3-1 System block diagram



- System bus: This bus connects the system bus (peripheral bus) of the Cortex-M0+ core to the bus matrix, which coordinates access between the core and DMA.
- DMA bus: The bus connects the AHB master interface of the DMA to a bus matrix that coordinates CPU and DMA access to SRAM, flash and peripherals.
- Bus matrix: The bus matrix coordinates the access arbitration between the kernel system bus and the DMA master bus, and the arbitration adopts a fixed priority, and the DMA priority is high.
- AHB to APB Bridge: AHB to APB Bridge provides a synchronous connection between the AHB and APB buses. Refer to for address mapping of the different peripherals connected to each bridge Table 3-1.

3.2 System Address Partitioning

Figure 3-2 Map of address area



Peripheral address assignment

Table 3-1 Peripheral register bank start address

Start address	Peripheral	Remark
0x4000_0000 - 0x4000_4FFF	Reserved	
0x4000_5000 - 0x4000_5FFF	DMA	
0x4000_6000 - 0x4000_6FFF	Interrupt control	
0x4000_7000 - 0x4001_8FFF	Reserved	
0x4001_9000 - 0x4001_9FFF	Reserved	
0x4001_A000 - 0x4001_FFFF	Reserved	
0x4002_0000 - 0x4002_03FF	Flash control	
0x4002_0400 - 0x4002_0FFF	Clock control	
0x4002_1000 - 0x4002_1001	Watchdog timer	
0x4002_1002 - 0x4002_1800	Reserved	
0x4002_1800 - 0x4002_1BFF	High-speed CRC	See Chapter 23 Safety Function for details
0x4002_1C00 - 0x4002_1FFF	Clock control	
0x4002_2000 - 0x4003_FFFF	Reserved	
0x4004_0000 - 0x4004_0FFF	GPIO	
0x4004_1100 - 0x4004_19FF	Serial communication unit	
0x4004_1A00 - 0x4004_1CFF	Serial interface IICA	
0x4004_1D00 - 0x4004_1FFF	Timer array unit 0	
0x4004_2000 - 0x4004_21FF	Timer array unit 1	
0x4004_2200 - 0x4004_25FF	Reserved	
0x4004_2600 - 0x4004_29FF	Reserved	
0x4004_2A00 - 0x4004_2BFF	Reserved	
0x4004_2C00 - 0x4004_2FFF	Reserved	
0x4004_3000 - 0x4004_31FF	Reserved	
0x4004_3200 - 0x4004_32FF	Universal CRC	See Chapter 23 Safety Function for details
0x4004_3300 - 0x4004_33FF	Reserved	
0x4004_3400 - 0x4004_37FF	Linkage controller	
0x4004_3800 - 0x4004_3BFF	Comparator	
0x4004_3C00 - 0x4004_3FFF	Reserved	
0x4004_4000 - 0x4004_46FF	IrDA	
0x4004_4700 - 0x4004_4AFF	Reserved	
0x4004_4B00 - 0x4004_4EFF	Reserved	
0x4004_4F00 - 0x4004_4FFF	Real time clock	
0x4004_5000 - 0x4004_53FF	AD converter	
0x4004_5400 - 0x4004_5AFF	Reserved	
0x4004_5B00 - 0x4004_5BFF	Interrupt control	
0x4008_0000 - 0x4008_01FF	Reserved	
0x4008_0200 - 0xDFFF_FFFF	Reserved	

Chapter 4 Clock Generation Circuit

The pins for connecting the resonator for the main system clock/external clock input, and the pins for connecting the resonator for the subsystem clock/external clock input, may vary depending on the product.

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 three types of system clock and clock oscillation circuits.

(1) Main system clock

① X1 oscillation circuit

The X1 pin and X2 pin can oscillate a clock with $F_X=1$ to 20MHz by connecting the X1 pin and the X2 pin to the resonator, or the oscillation can be stopped by entering the deep sleep mode or setting the MSTOP bit (bit 7 of the clock operation status control register (CSC)).

② High-speed on-chip oscillator (high-speed OCO)

It can oscillate from $F_{HOCO}=64\text{MHz}$, 48MHz, 32MHz, 24MHz, 16MHz, 12MHz, 8MHz, 6MHz, 4MHz, 3MHz, 2MHz and 1MHz (TYP.) via option bytes (000C2H). When using the high-speed on-chip oscillator clock as the USB clock, $f_{HOCO} = 48\text{MHz}$ must be selected. After the reset is released, the CPU must start running with this high-speed on-chip oscillator clock. The oscillator can be stopped by entering deep sleep mode or setting the HIOSTOP bit (bit 0 of the CSC register). The frequency set by the option byte can be changed by the high-speed on-chip oscillator frequency selection register, HOCODIV. For frequency settings, refer to Figure 4-10: Format of the High-Speed On-Chip Oscillator Frequency Selection Register (HOCODIV).

In addition, an external main system clock ($f_{EX} = 1$ to 20MHz) can be provided through the EXCLK/X2/P02 pins, and the input of the external main system clock can be disabled by entering deep sleep mode or setting the MSTOP bit.

The switching between the high-speed system clock (either the X1 clock or the external main system clock) and the high-speed on-chip oscillator clock is controlled by setting the MCM0 bit (bit 4 of the System Clock Control Register, CKC).

(2) Subsystem clock

- XT1 oscillating circuit

The XT1 pin and XT2 pin are connected to a 32.768kHz resonator to oscillate the clock with $f_{XT}=32.768\text{kHz}$ and to stop the oscillation by setting a XTSTOP bit (bit6 of the clock operation status control register (CSC)).

In addition, the external subsystem clock ($f_{EXS}=32.768\text{kHz}$) can be provided by the EXCLKS/XT2/P13 pin, and the input of the external subsystem clock can be set invalid by setting the XTSTOP bit.

(3) Low speed on-chip oscillator clock (low speed OCO)

It can make the clock oscillate at $f_{IL}=30\text{kHz}/15\text{kHz}$.

A low-speed on-chip oscillator clock can not be used as a CPU clock.

The SysTick timer use a low-speed on-chip oscillator clock as an external reference clock.

Only the following peripheral hardware can operate using the low-speed on-chip oscillator clock:

- Watchdog timer
- Real-time clock
- 15-bit interval timer

When bit4 (WDTON) of the option byte (000C0H) or bit4 (WUTMMCK0) of the subsystem clock supply mode control register (OSMC) is 1, the low-speed on-chip oscillator oscillates.

However, the low-speed on-chip oscillator stops oscillating if either deep sleep mode or sleep mode is entered when the WDTON bit is 1 and WUTMMCK0 bit is 0 and the bit0 (WDSTBYON) of the option byte is 0.

Caution: The low-speed on-chip oscillator clock (f_{IL}) can be selected as the count clock of the real-time clock only when a fixed period interrupt function is used.

Remarks:

- f_X : X1 clock oscillation frequency
- f_{HOCO} : High-speed on-chip oscillator clock frequency
- f_{IH} : High-speed on-chip oscillator clock frequency
- f_{EX} : External main clock frequency
- f_{XT} : XT1 clock oscillation frequency
- f_{EXS} : External subsystem clock frequency
- f_{IL} : Low-speed on-chip oscillator clock frequency

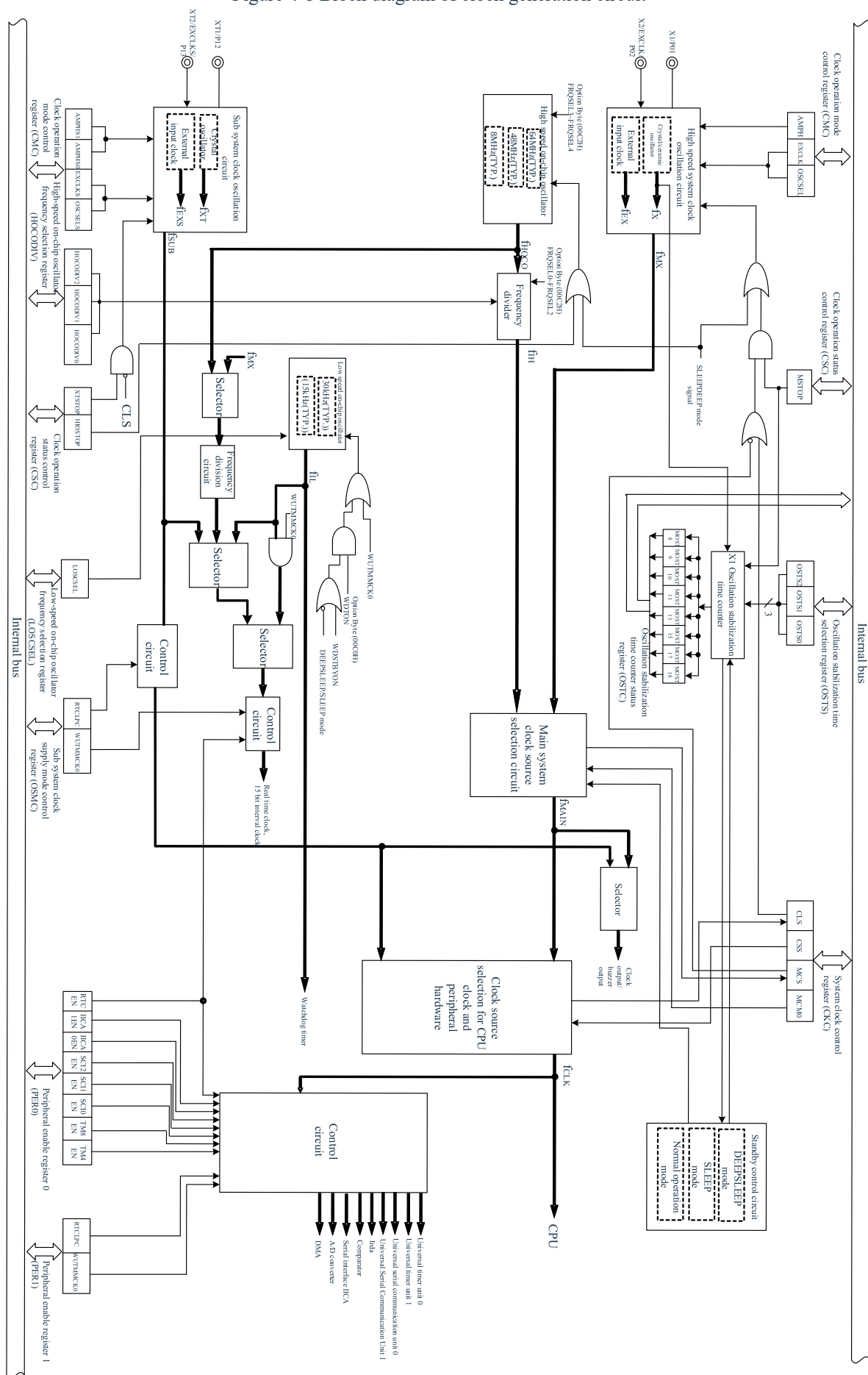
4.2 Structure of Clock Generation Circuit

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) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) Peripheral enable registers 0, 1 (PER0, PER1) Subsystem clock supply mode control register (OSMC) High-speed on-chip oscillator frequency selection register (HOCODIV) Low-speed on-chip oscillator frequency selection register (LOSCSEL) Frequency selection register protection register (PRCR) External reset pin reset function mask register (RSTM)
Oscillation circuit	X1 oscillation circuit XT1 oscillation circuit High-speed on-chip oscillator Low-speed on-chip oscillator

Figure 4-1 Block diagram of clock generation circuit



Notes	f_X	: X1 clock oscillation frequency
	f_{HOCO}	: High-speed on-chip oscillator clock frequency
	f_{IH}	: High-speed on-chip oscillator clock frequency
	f_{EX}	: External main clock frequency
	f_{MX}	: High-speed system clock frequency
	f_{MAIN}	: Main system clock frequency
	f_{XT}	: XT1 clock oscillation frequency
	f_{EXS}	: External subsystem clock frequency
	f_{SUB}	: Sub-system clock frequency
	f_{CLK}	: CPU/peripheral hardware clock frequency
	f_{IL}	: Low-speed on-chip oscillator clock frequency

4.3 Registers for Controlling Clock Generation Circuit

The clock generation circuit is controlled by the following registers.

- Clock operation mode control register (CMC)
- System clock control register (CKC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time selection register (OSTS)
- Peripheral enable registers 0, 1 (PER0, PER1)
- Subsystem clock supply mode control register (OSMC)
- High-speed on-chip oscillator frequency selection register (HOCODIV)
- Low-speed on-chip oscillator frequency selection register (LOSCSEL)
- Frequency selection protection register (PRCR)
- External reset pin function mask register (RSTM)

Note: Which registers and bits are included depends on the product. Be sure to set bits that are not mounted to their initial values.

4.3.1 Clock Operation Mode Control Register (CMC)

This is a register that sets the operation mode of the X1/P01, X2/EXCLK/P02, XT1/P12, XT2/EXCLKS/P13 pin and selects the gain of the oscillating circuit.

The CMC register can only be written 1 time by the 8-bit memory manipulation instruction after reset. The register can be read by an 8-bit memory manipulation instruction.

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

Figure 4-2 Format of clock operation mode control register (CMC)

Address: 40020400H

After reset: 00H

R/W

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS ^{Note}	OSCSELS ^{Note}	0	AMPHS1 ^{Note}	AMPHS0 ^{Note}	AMPH

EXCLK	OSCSEL	High-speed system clock pin operation mode	X1/P01 pin	X2/EXCLK/P02 pin
0	0	Port mode	Input/output port	
0	1	X1 oscillation mode	Crystal or ceramic resonator connection	
1	0	Port mode	Input/output port	
1	1	External clock input mode	Input/output port	External clock input

EXCLKS	OSCSELS	Subsystem clock pin operation mode	XT1/P12 pin	XT2/EXCLKS/P13 pin
0	0	Port mode	Input/output port	
0	1	XT1 oscillation mode	Crystal resonator connection	
1	0	Port mode	Input/output port	
1	1	External clock input mode	Input/output port	External clock input

AMPHS1	AMPHS0	Selection of oscillation modes for XT1 oscillation circuit
0	0	Low power oscillation (default)
0	1	Normal oscillation
1	0	Ultra-low power oscillation
1	1	Disable setting.

AMPH	Control of X1 clock oscillation frequency
0	$1\text{MHz} \leq f_X \leq 10\text{MHz}$
1	$10\text{MHz} < f_X \leq 20\text{MHz}$

Note: The EXCLKS bit, OSCSELS bit, AMPHS1 bit, and AMPHS0 bit are initialized only during power-on reset and remain unchanged during other resets.

Cautions:

1. After reset is released, the CMC register can only be written once with the value 1 using 8-bit memory manipulation instructions. When the CMC register is used with its initial value (00H), in order to prevent erroneous actions due to program malfunction (as writing a value other than 00H cannot be recovered), the CMC register must be set to 00H after reset is released.
2. After reset is released, and before starting the X1 or XT1 oscillation by setting the Clock Operation Status Control Register (CSC), the CMC register must be set.
3. When the X1 clock oscillation frequency exceeds 10MHz, the AMPHS bit must be set to 1.

4. The AMPHS, AMPHS1, and AMPHS0 bits must be set after reset is released and while the system is in the state where f_{IH} is selected as f_{CLK} (before switching f_{CLK} to f_{MX} or f_{SUB}).
5. The oscillation stabilization time of f_{XT} must be counted through software.
6. The system clock frequency is limited to 64MHz, but the X1 oscillation circuit's frequency is limited to 20MHz.

Remark: f_X : X1 clock oscillation frequency

4.3.2 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.

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

Figure 4-3 Format of system clock control register (CKC)

Address: 40020404H	After reset: 00H	R/W ^{Note1}						
Symbol	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	0	0
CLS	CPU/peripheral hardware clock (f_{CLK}) status							
0	Main system clock (f_{MAIN})							
1	Subsystem clock (f_{SUB})							
CSS ^{Note2}	CPU/peripheral hardware clock (f_{CLK}) selection							
0	Main system clock (f_{MAIN})							
1	Subsystem clock (f_{SUB})							
MCS	Main system clock (f_{MAIN}) status							
0	High-speed on-chip oscillator clock (f_{IH})							
1	High-speed system clock (f_{MX})							
MCM0 ^{Note2}	Main system clock (f_{MAIN}) operation control							
0	Select the high-speed on-chip oscillator clock (f_{IH}) as the main system clock (f_{MAIN}).							
1	Select the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN}).							

Notes: 1. Bit7 and bit5 are read-only bits.

2. Do not change the value of the MCM0 bit when the CSS bit is set to 1.

Remark: f_{HOCO} : High speed on-chip oscillator clock frequency

f_{IH} : High speed on-chip oscillator clock frequency

f_{MX} : High-speed system clock frequency

f_{MAIN} : Main system clock frequency

f_{SUB} : Subsystem clock frequency

Cautions: 1. Bit0~3 must be set to 0.

- The CSS bit provides the clock for the CPU and peripheral hardware. If the CPU clock is changed, the peripheral hardware clocks will also change (except for the real-time clock, 15-bit interval timer, clock output/buzzer output, and watchdog timer). Therefore, if you want to change the CPU/peripheral hardware clock, you must stop the respective peripheral functions.
- If the subsystem clock is used as the peripheral hardware clock, the operation of the A/D converter and IICA cannot be guaranteed. For the operating characteristics of the peripheral hardware, please refer to the relevant sections and data manuals of each peripheral hardware.

4.3.3 Clock Operation Status Control Register (CSC)

This is a register that controls the operation of a high-speed system clock, a high-speed on-chip oscillator clock, and a subsystem clock (except for a low-speed on-chip oscillator clock). The CSC register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “C0H”.

Figure 4-4 Format of clock operation status control register (CSC)

Address: 40020401H	After reset: C0H	R/W						
Symbol	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP

MSTOP	Operation control of high-speed system clock		
	X1 oscillation mode	External clock input mode	Port mode
0	X1 oscillator circuit runs	The EXCLK pin external clock is valid	Input/output port
1	X1 oscillator circuit stops	The EXCLK pin external clock is invalid	

XTSTOP	Operation control of subsystem system clock		
	XT1 oscillation mode	External clock input mode	Port mode
0	XT1 oscillator circuit runs	The EXCLKS pin external clock is valid	Input/output port
1	XT1 oscillator circuit stops	The EXCLKS pin external clock is invalid	

HIOSTOP	Operation control of high-speed on-chip oscillator clock	
0	The high-speed on-chip oscillator runs	
1	The high-speed on-chip oscillator stops	

Cautions:

1. After the reset is released, the CSC register must be set after setting the Clock Operation Mode Control Register (CMC).
2. After the reset is released and before setting the MSTOP bit to 0, the Oscillation Stabilization Time Select Register (OSTS) must be set. However, when using the OSTs register with its initial value, setting the OSTs register is not required.
3. To start the XT1 oscillation by setting the XSTOP bit, software must wait for the oscillation stabilization time required by the subsystem clock.
4. When you want to start XT1 oscillation by setting the XSTOP bit, you must wait for the oscillation stabilization time required by the subsystem clock through software.
5. The clock selected as the CPU/peripheral hardware clock (f_{CLK}) cannot be stopped via the CSC register.
6. For the register flags used to stop clock oscillation (when the external clock input is invalid) and the conditions before stopping, refer to Table 4-2.

Table 4-2 Clock stopping method

Clock	Condition before clock stops (invalid external clock input)	Set CSC register flag
X1 clock	The CPU/peripheral hardware clock runs at a clock other than the high-speed system clock. (CLS=0 and MCS=0, or CLS=1)	MSTOP=1
External main system clock		
XT1 clock	The CPU/peripheral hardware clock runs at a clock other than the subsystem clock. (CLS=0)	XTSTOP=1
External subsystem clock		
High-speed on-chip oscillator clock	The 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

4.3.4 Oscillation Stabilization Time Counter Status Register (OSTC)

This is a register that represents the count state of the oscillating steady-time counter of the X1 clock. The oscillation stability time of the X1 clock can be confirmed under the following circumstances:

- When the CPU clock is a high-speed on-chip oscillator clock or a subsystem clock and the oscillation of the X1 clock is started.
- When the CPU clock is a high-speed on-chip oscillator clock and the sleep mode is released after transferring to deep sleep mode in the X1 clock oscillation state.

The OSTC register can be read by an 8-bit memory manipulation instruction.

By generating a reset signal, the system enters deep sleep mode, or when the MSTOP bit (bit 7 of the Clock Status Control Register (CSC)) is set to 1, the value of this register becomes 00H.

Note: The oscillation stabilization time counter starts counting when:

- When the X1 clock starts to oscillate (EXCLK, OSCSEL=0, 1→MSTOP=0)
- When deep sleep mode is released

Figure 4-5 Format of oscillation stabilization time counter status register (OSTC)

Address: 40020402H After reset: 00H R

Symbol 7 6 5 4 3 2 1 0

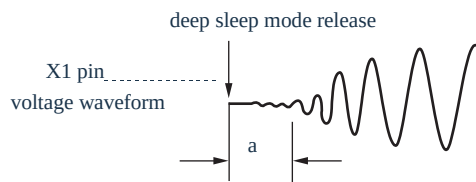
OSTC

MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18

MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18	Oscillation stabilization time status		
									$f_X=10\text{MHz}$	$f_X=20\text{MHz}$
0	0	0	0	0	0	0	0	Less than $2^8/f_X$	Less than $25.6\mu\text{s}$	Less than $12.8\mu\text{s}$
1	0	0	0	0	0	0	0	At least $2^8/f_X$	At least $25.6\mu\text{s}$	At least $12.8\mu\text{s}$
1	1	0	0	0	0	0	0	At least $2^9/f_X$	At least $51.2\mu\text{s}$	At least $25.6\mu\text{s}$
1	1	1	0	0	0	0	0	At least $2^{10}/f_X$	At least $102\mu\text{s}$	At least $51.2\mu\text{s}$
1	1	1	1	0	0	0	0	At least $2^{11}/f_X$	At least $204\mu\text{s}$	At least $102\mu\text{s}$
1	1	1	1	1	0	0	0	At least $2^{13}/f_X$	At least $819\mu\text{s}$	At least $409\mu\text{s}$
1	1	1	1	1	1	0	0	At least $2^{15}/f_X$	At least 3.27ms	At least 1.63ms
1	1	1	1	1	1	1	0	At least $2^{17}/f_X$	At least 13.1ms	At least 6.55ms
1	1	1	1	1	1	1	1	At least $2^{18}/f_X$	At least 26.2ms	At least 13.1ms

Cautions

- After the time mentioned above, you change from MOST8 to 1 and remain at 1.
- The oscillation stable time counter counts only within the oscillation stable time set by the OSTC. In the following cases, the setting value of the oscillation stability time of the OSTC register must be greater than the count value confirmed by the OSTC register.
 - When the CPU clock is a high-speed on-chip oscillator clock or a subsystem clock and the X1 clock is to start oscillating.
 - When the CPU clock is a high-speed on-chip oscillator clock and is released from deep sleep mode after shifting to deep sleep mode in the state of X1 clock oscillation (therefore, it must be noted that the OSTC register after release from deep sleep mode only sets the state within the oscillation stabilization time set by the OSTC register)
- The oscillation stabilization time of the X1 clock does not include the time before the clock starts oscillating (Figure a below).


Remark: f_X : X1 clock oscillation frequency.

4.3.5 Oscillation Stabilization Time Select Register (OSTS)

This is a register that selects the oscillation steady time of the X1 clock.

If the X1 clock is oscillated, the time set by the OSTS register is automatically waited after the X1 oscillation circuit (MSTOP=0).

If the CPU clock is switched from a high-speed on-chip oscillator clock or a subsystem clock to an X1 clock, or if the CPU clock is a high-speed on-chip oscillator clock and is switched to a deep sleep mode.

The OSTC register can be used to confirm the time set by the OSTS register.

The OSTS register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register changes to “07H”.

Figure 4-6 Format of oscillation stabilization time select register (OSTS)

Address: 40020403H	After reset: 07H	R/W						
Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time		
				$f_x=10\text{MHz}$	$f_x=20\text{MHz}$
0	0	0	$2^8/f_x$	25.6μs	12.8μs
0	0	1	$2^9/f_x$	51.2μs	25.6μs
0	1	0	$2^{10}/f_x$	102μs	51.2μs
0	1	1	$2^{11}/f_x$	204μs	102μs
1	0	0	$2^{13}/f_x$	819μs	409μs
1	0	1	$2^{15}/f_x$	3.27ms	1.63ms
1	1	0	$2^{17}/f_x$	13.1ms	6.55ms
1	1	1	$2^{18}/f_x$	26.2ms	13.1ms

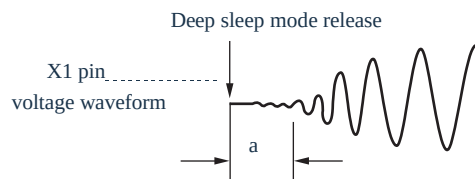
Cautions:

1. To change the setting of the OSTS register, you must make the change before the MSTOP bit of the Clock Operation Status Control Register (CSC) set to 0.
2. The oscillation stable time counter is counted only in that oscillation stable time set in the OSTS register.

In the following cases, the setting value of the oscillation stability time of the OSTS register must be greater than the count value confirmed by the OSTC register after the start of the oscillation.

- When the CPU clock is a high-speed on-chip oscillator clock or a subsystem clock and the X1 clock is to start oscillating
- When the CPU clock is a high-speed on-chip oscillator clock and is released from deep sleep mode after shifting to deep sleep mode in the state of X1 clock oscillation (therefore, it must be noted that the OSTC register after release from deep sleep mode only sets the state within the oscillation stabilization time set by the OSTS register)

3. The oscillation stable time of the X1 clock does not include the time before the clock starts to oscillate (Figure a below).



Remark: f_x : X1 clock oscillation frequency

4.3.6 Peripheral Enable Registers 0, 1 (PER0, PER1)

This is a register that sets a clock that is enabled or disabled for each peripheral hardware. Reduce power consumption and noise by stopping clock supply to unused hardware.

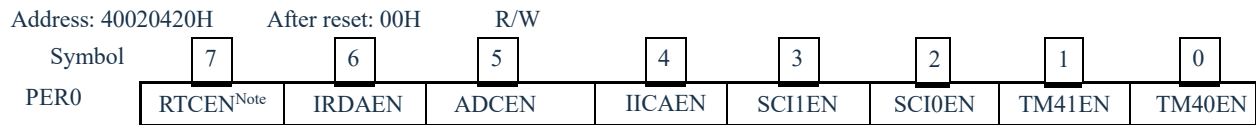
When the following peripheral functions controlled by these registers are used, the corresponding bit must be set to 1 before the initial setting of the peripheral functions.

- Real-time clock, 15-bit interval timer
- IrDA
- A/D converter
- Serial interface IICA0
- Universal serial communication unit 1
- Universal serial communication unit 0
- Universal timer unit 1
- Universal timer unit 0
- D/A converter
- Comparator
- Enhanced DMA

The PER0 register and PER1 register are set by an 8-bit memory manipulation instruction.

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

Figure 4-7 Format of peripheral enable register 0 (PER0) (1/3)



RTCEN	Control of real-time clock (RTC) and 15-bit interval timer input clock supply
0	Stop input clock supply. • SFR used by the real-time clock (RTC) and 15-bit interval timer cannot be written. • The real-time clock (RTC) and interval timer are in the reset status.
1	Enable input clock supply. • SFR used by the real-time clock (RTC) and 15-bit interval timer can be read and written.

Note: The RTCEN bit is initialized only when power-on reset, and remains unchanged during other resets.

Figure 4-7 Format of peripheral enable register 0 (2/3)

Address: 40020420H	After reset: 00H	R/W						
Symbol	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICAEN	SCI1EN	SCI0EN	TM41EN	TM40EN

IRDAEN	Control of serial interface IRDA input clock supply
0	Stop to supply the input clock. • The SFR used by the serial interface IRDA cannot be written. • Serial interface IRDA is in the reset state.
1	Supply the input clock. • The SFR used by the serial interface IRDA can be read and written.

ADCEN	Control of A/D converter input clock supply
0	Stop to supply the input clock. • The SFR used by the A/D converter cannot be written. • A/D converter is in the reset state.
1	Supply the input clock. • The SFR used by the A/D converter can be read and written.

IICA0EN	Control of serial interface IICA0 input clock supply
0	Stop to supply the input clock. • The SFR used by the serial interface IICA0 cannot be written. • Serial interface IICA0 is in the reset state.
1	Supply the input clock. • The SFR used by the serial interface IICA0 can be read and written.

SCI1EN	Control of serial communication unit 1 input clock supply
0	Stop to supply the input clock. • The SFR used by universal serial communication unit 1 cannot be written. • Universal serial communication unit 1 is in the reset state.
1	Supply the input clock. • The SFR used by the universal serial communication unit 1 can be read and written.

SCI0EN	Control of serial communication unit 0 input clock supply
0	Stop to supply the input clock. • The SFR used by universal serial communication unit 0 cannot be written. • Universal serial communication unit 0 is in the reset state.
1	Supply the input clock. • The SFR used by the universal serial communication unit 0 can be read and written.

Figure 4-7 Format of peripheral enable register 0 (PER0) (3/3)

Address: 40020420H	After reset: 00H	R/W						
Symbol	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICAEN	SCI1EN	SCI0EN	TM41EN	TM40EN

TM41EN	Control of universal timer unit 1 input clock supply
0	Stop to supply the input clock. - The SFR used by the universal timer unit 1 cannot be written. - Universal timer unit 1 is in the reset state.
1	Supply the input clock. The SFR used by the universal timer unit 1 can be read and written.

TM40EN	Control of universal timer unit 0 input clock supply
0	Stop to supply the input clock. - The SFR used by the universal timer unit 0 cannot be written. - Universal timer unit 0 is in the reset state.
1	Supply the input clock. The SFR used by the universal timer unit 0 can be read and written.

Figure 4-8 Format of peripheral enable register 1 (PER1)

Address: 4002081AH	After reset: 00H	R/W						
Symbol	7	6	5	4	3	2	1	0
PER1	0	0	PGACMPEN	0	DMAEN	0	0	0

PGACMPEN	Control of amplifier and comparator input clock supply
0	Stop to supply the input clock. - The SFR used by amplifier and comparator cannot be written. - Amplifier and comparator are in the reset state.
1	Supply the input clock. The SFR used by the amplifier and comparator can be read and written.

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.

4.3.7 Subsystem Clock Supply Mode Control Register (OSMC)

The OSMC register is a register that reduces power consumption by stopping an unwanted clock function.

If the RTCLPC bit is set to 1, it stops clocking peripheral functions other than the real-time clock and 15-bit interval timer in deep sleep mode or sleep mode where the CPU runs on the subsystem clock, thus reducing power consumption.

In addition, the real-time clock and the runtime clock of the 15-bit interval timer can be selected through the OSMC register.

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

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

Figure 4-9 Format of subsystem clock supply mode control register (OSMC)

Address: 40020423H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0

RTCLPC	Setting in deep sleep mode or sleep mode while subsystem clock is selected as CPU clock
0	Enables sub-system clocks for peripheral functions (Refer to Tables 19-1 to 19-2 for the peripheral functions that are enabled to operate).
1	Stop to supply a subsystem clock to peripheral functions other than the real-time clock and 15-bit interval timer.

WUTMMCK0	Selection of operation clock for real-time clock and 15-bit interval timer
0	The subsystem clock is the operating clock of the real-time clock and the 15-bit interval timer.
1	The low-speed on-chip oscillator clock is the operating clock of the real-time clock and the 15-bit interval timer.

4.3.8 High-Speed On-Chip Oscillator Frequency Select Register (HOCODIV)

This is a register that changes the high-speed on-chip oscillator frequency set by the option byte (000C2H). However, the frequency that can be selected varies depending on the values of the FRQSEL4 bit and FRQSEL3 bit of the option byte (000C2H).

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

After a reset signal is generated, the value of this register becomes the set value of the FRQSEL2~FRQSEL0 bit of the option byte (000C2H).

Figure 4-10 Format of high-speed on-chip oscillator frequency select register (HOCODIV)

Address: 40021C20H After reset: Setting value for the FRQSEL2~FRQSEL0 bit of the option byte (000C2H) R/W

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

HOCODIV2	HOCODIV1	HOCODIV0	Selection of high-speed on-chip oscillator frequency			
			FRQSEL4=0		FRQSEL4=1	
			FRQSEL3=0	FRQSEL3=1	FRQSEL3=0	FRQSEL3=1
0	0	0	Disable settings	$f_{IH}=48\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=64\text{MHz}$ $f_{HOCO}=64\text{MHz}$	Disable settings
0	0	1	Disable settings	$f_{IH}=24\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=32\text{MHz}$ $f_{HOCO}=64\text{MHz}$	Disable settings
0	1	0	Disable settings	$f_{IH}=12\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=16\text{MHz}$ $f_{HOCO}=64\text{MHz}$	Disable settings
0	1	1	Disable settings	$f_{IH}=6\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=8\text{MHz}$ $f_{HOCO}=64\text{MHz}$	Disable settings
1	0	0	Disable settings	$f_{IH}=3\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=4\text{MHz}$ $f_{HOCO}=64\text{MHz}$	Disable settings
1	0	1	Disable settings	Disable settings	$f_{IH}=2\text{MHz}$ $f_{HOCO}=64\text{MHz}$	Disable settings
1	1	0	Disable settings	Disable settings	$f_{IH}=1\text{MHz}$ $f_{HOCO}=64\text{MHz}$	Disable settings
Others			Disable settings			

Cautions:

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}).
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.

4.3.9 Low-Speed On-Chip Oscillator Frequency Selection Register (LOSCSEL) and Its Protection Register (PRCR)

The LOSCSEL register is the frequency selection register for the low-speed on-chip oscillator. The PRCR register is its write protection register.

The LOSCSEL and PRCR registers are set through 8-bit manipulation operation instructions. After a reset signal is generated, the value of the LOSCSEL register becomes “01H”, and the value of the PRCR register becomes “00H”.

Figure 4-11 Format of low-speed on-chip oscillator frequency selection register (LOSCSEL) and its protection register (PRCR) (1/2)

Address: 40020405H	After reset: 01H	R/W						
Symbol	7	6	5	4	3	2	1	0
LOSCSEL	0	0	0	0	0	0	0	LSOCSEL

LOSCSEL	Low-speed on-chip oscillator frequency selection
0	• 15K
1	• 30K

Figure 4-11 Format of low-speed on-chip oscillator frequency selection register (LOSCSEL) and its protection register (PRCR) (2/2)

Address: 40020406H	After reset: 00H	R/W						
Symbol	7	6	5	4	3	2	1	0
PRCR	PRTKEY[7:1]							PRCR

PRCR	Low-speed on-chip oscillator frequency selection register write protection
0	• LOSCSEL register is not writable
1	• LOSCSEL register is writable

PRTKEY[7:1]	PRCR write protection
78H	• PRCR is writable
Other	• PRCR is not writable

4.3.10 External Reset Pin Reset Function Mask Register (RSTM)

The RSTM register is the reset function masking register for the external reset pin.

This register is set through an 8-bit manipulation operation instruction.

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

Figure 4-12 Format of external reset pin reset function mask register (RSTM)

Address: 40020407H	After reset: 00H	R/W						
Symbol	7	6	5	4	3	2	1	0
RSTM	0	0	0	0	0	0	0	RSTM

RSTM	External reset pin reset function mask register
0	The reset function of the external reset pin is not masked, and P00 is used as the reset terminal.
1	The reset function of the external reset pin is masked, and P00 is used as a GPIO.

4.4 System Clock Oscillation Circuit

4.4.1 X1 Oscillation Circuit

The X1 oscillation circuit oscillates by a crystal resonator or a ceramic resonator (1 to 20MHz) connecting the X1 pin. An external clock can also be input, at which time a clock signal must be input to the EXCLK pin.

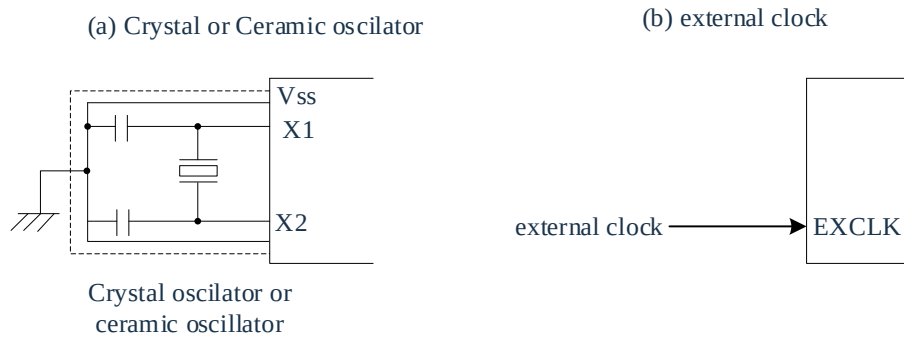
When using the X1 oscillator circuit, the bit7 and bit6 (EXCLK, OSCSEL) of the clock mode control register must be set:

- Crystal or ceramic oscillation: EXCLK, OSCSEL=0, 1
- External clock input: EXCLK, OSCSEL=1, 1

When the X1 oscillator circuit is not used, it must be set to the input port mode (EXCLK, OSCSEL=0, 0). Also, refer to the “Table 2-3 Handling of each unused pin” when not used as an input port.

An example of an external circuit for the X1 oscillation circuit is shown in Figure 4-13.

Figure 4-13 Example of an external circuit for X1 oscillation circuit



Cautions are shown on the next page.

4.4.2 XT1 Oscillation Circuit

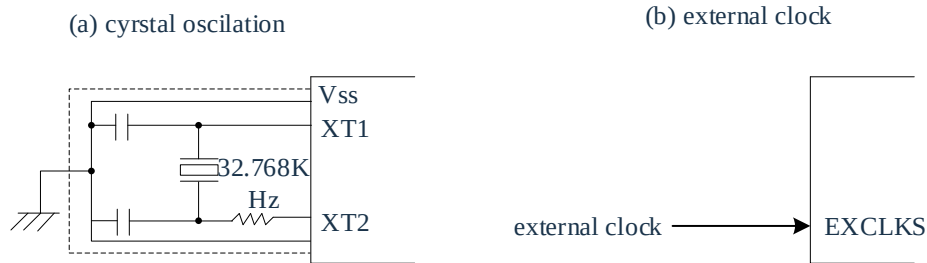
The XT1 oscillation circuit oscillates by a crystal resonator (32.768kHz (TYP.)) connecting the XT1 pin and XT2 pin. When the XT1 oscillation circuit is used, the bit4 (OSCSELS) of the clock operation mode control register (CMC) must be set to “1” to input the external clock, and the EXCLKS pin must be input.

When using the XT1 oscillator circuit, the bit5 and bit4 (EXCLKS, OSCSELS) of the clock mode control register must be set:

- Crystal oscillation: EXCLKS, OSCILS=0, 1
- External clock input: EXCLKS, OSCILS=1, 1

When the XT1 oscillator circuit is not used, it must be set to the input port mode (EXCLKS, OSCSELS=0, 0). Also, when not used as an input port, refer to “Table 2-3 Handling of each unused pin”. An example of an external circuit for the XT1 oscillation circuit is shown in Figure 4-14.

Figure 4-14 Example of an external circuit for XT1 oscillation circuit

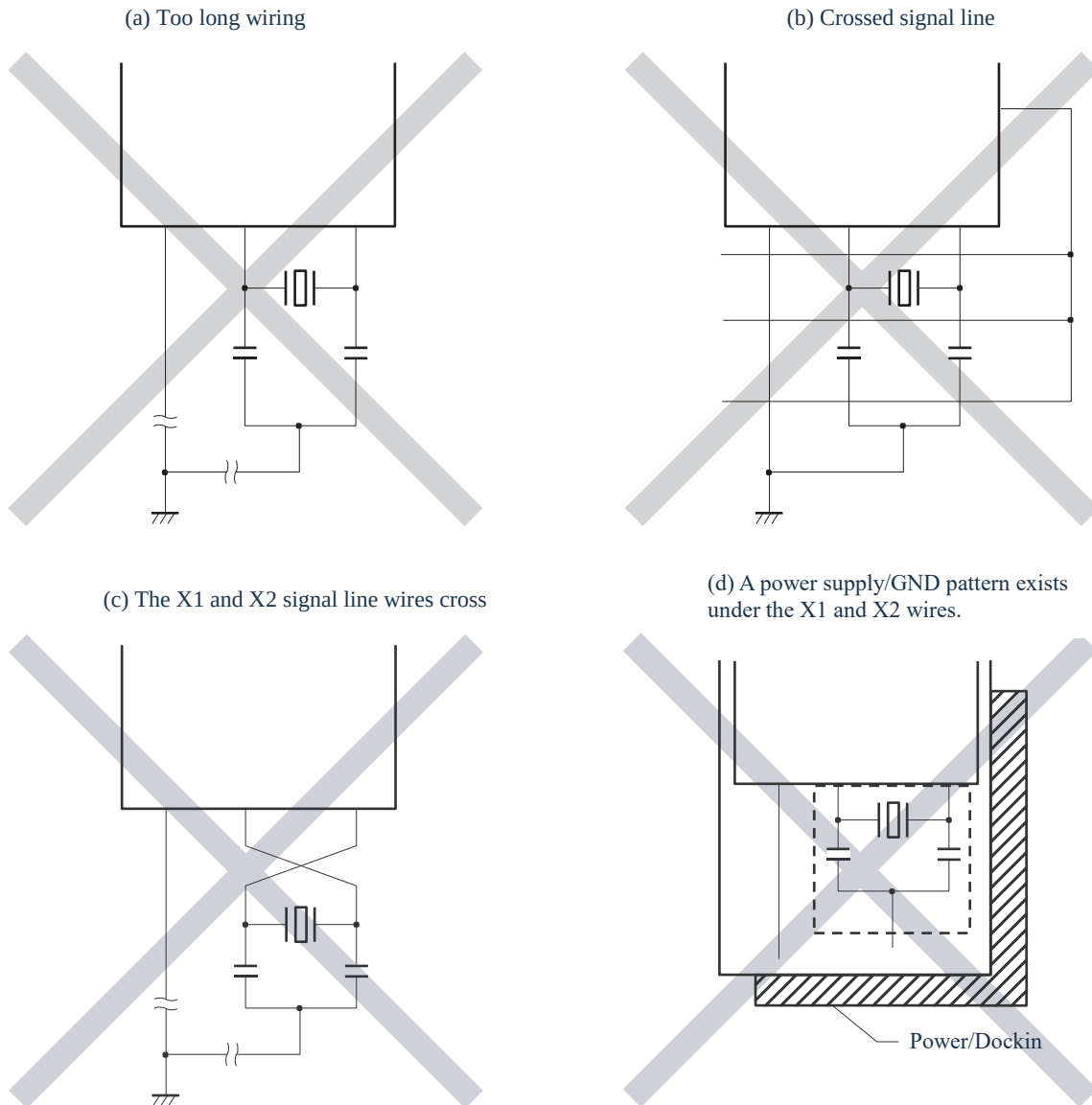


Cautions: When using the X1 oscillator circuit and the XT1 oscillator circuit, the dashed portions of Figure 4-13 and Figure 4-14 must be routed in order to avoid the effects of wiring capacitance, etc., by the following method:

- 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 V_{SS}. Do not ground the capacitor to a ground pattern through which a high current flow.
- Do not fetch signals from the oscillator.

Figure 4-15 shows examples of incorrect resonator connection.

Figure 4-15 Examples of incorrect resonator connection (1/2)

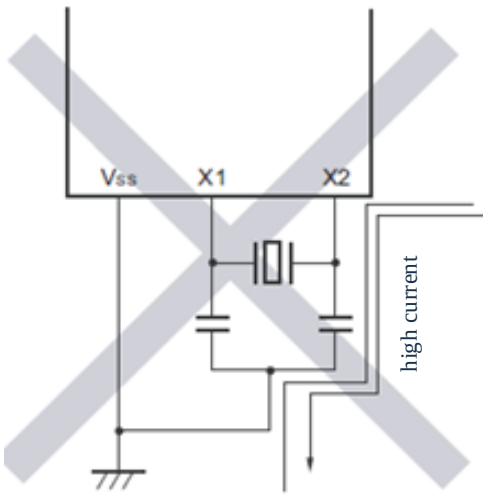


Note: In multilayer boards or double-sided boards, power or ground graphics should not be configured below the wiring area (dashed portion of the diagram) for pins X1, X2 and the resonator. The wiring must not create a capacitive component that would affect the oscillation characteristics.

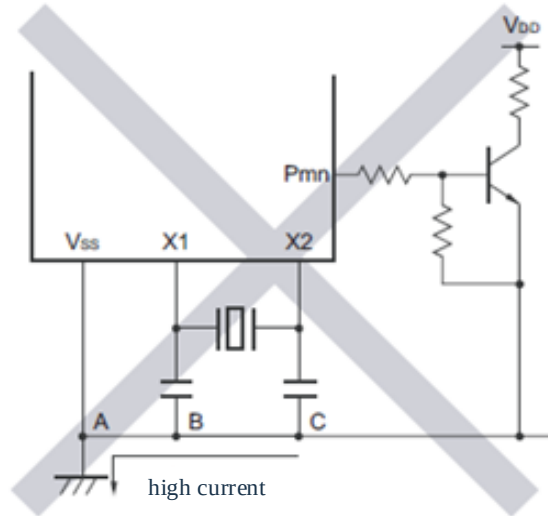
Remark: In the case of using the subsystem clock, please read with XT1 and XT2 instead of X1 and X2 respectively, and insert series resistors on the XT2 side.

Figure 4-15 Examples of incorrect resonator connection (2/2)

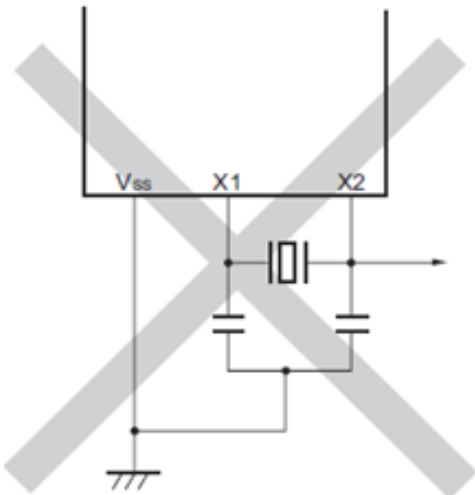
(e) Wiring near high alternating current



(f) Current flowing through ground line of oscillator (potential at points A, B, and C fluctuates)



(g) Signals are fetched



Caution: When X2 and XT1 are in parallel, the crosstalk noise of X2 will be superimposed to XT1 and cause misoperation.

Remark: In the case of using the subsystem clock, please read with XT1 and XT2 instead of X1 and X2 respectively, and insert series resistors on the XT2 side.

4.4.3 High-Speed On-Chip Oscillator

The BAT32G133B has a built-in high-speed on-chip oscillator. Frequency can be selected from 64MHz, 48MHz, 32MHz, 24MHz, 16MHz, 12MHz, 8MHz, 6MHz, 4MHz, 3MHz, 2MHz 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.

4.4.4 Low-Speed On-Chip Oscillator

The BAT32G133B has a built-in low-speed on-chip oscillator.

The low-speed on-chip oscillator clock is used as the watchdog timer, real-time clock, 15-bit interval timer clock, and an external reference clock of SysTick timer, but can not be used as a CPU clock.

The low-speed on-chip oscillator oscillates when the bit4 (WDTON) of the option byte (000C0H) or bit4 (WUTMMCK0) of the subsystem clock providing mode control register (OSMC).

The low-speed on-chip oscillator continues to oscillate when the watchdog timer stops running and the WUTMMCK0 bit is not "0". However, if the watchdog timer runs and the WUTMMCK0 bit is 0, the low-speed on-chip oscillator stops oscillating when the WDSTBYON bit is 0. When the watchdog timer runs, the low-speed on-chip oscillator clock does not stop running even if the program is out of control.

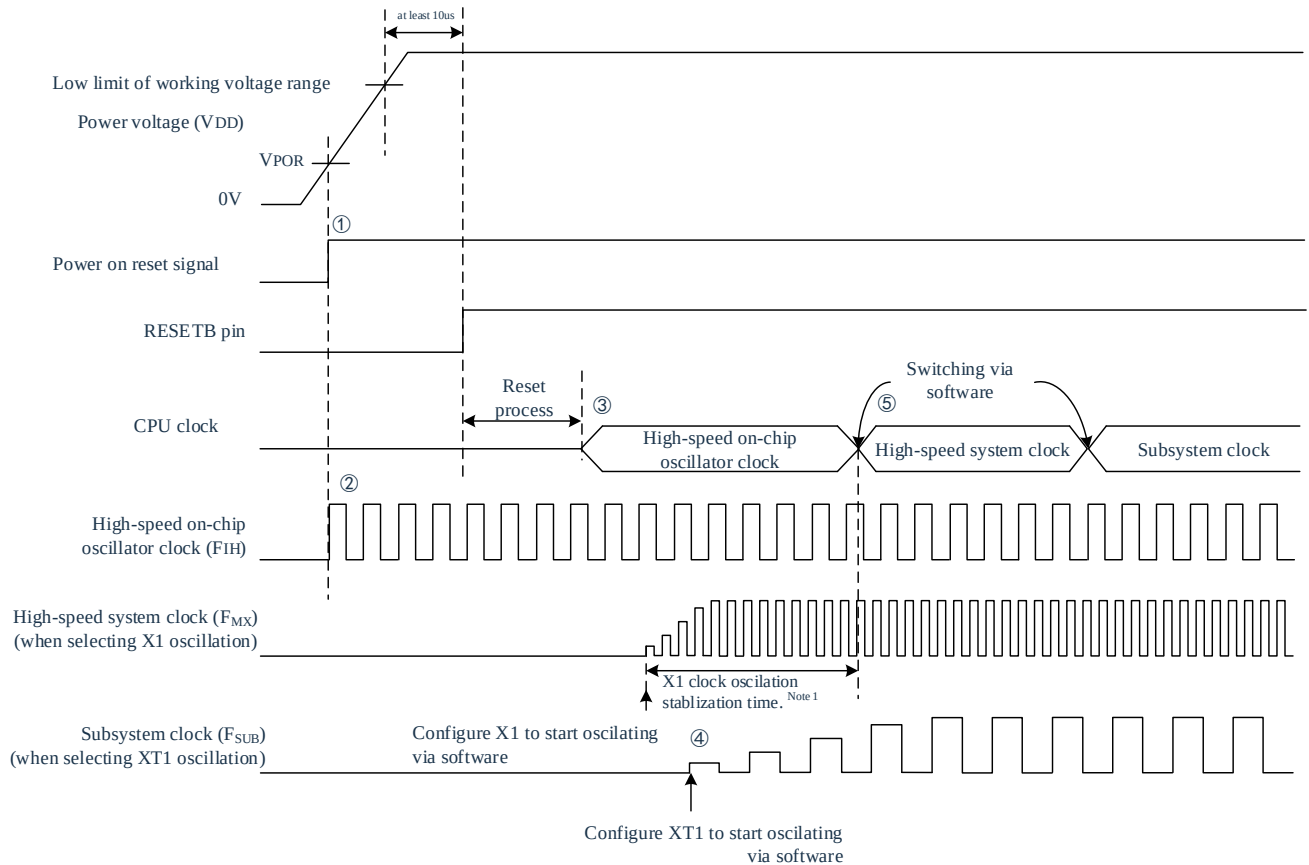
4.5 Operation of Clock Generation Circuit

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).

- Main system clock: F_{MAIN}
 - High-speed system clock: F_{MX}
 - X1 clock: F_X
 - External main system clock: F_{EX}
 - High-speed on-chip oscillator clock: F_{IH}
- Subsystem clock: F_{SUB}
 - XT1 clock: F_{XT}
 - External sub-system clock: F_{EXS}
- Low-speed on-chip oscillator clock: F_{IL}
- CPU/peripheral hardware clock: F_{CLK}

After the BAT32G133B 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 4-16.

Figure 4-16 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 “4.6.2 Example of setting X1 oscillation circuit” and “4.6.3 Example of controlling XT1 oscillation clock”).
- 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 “4.6.2 Example of setting X1 oscillation circuit” and “4.6.3 Example of controlling XT1 oscillation clock”).

Note: 1. When the reset is released, the oscillation stabilization time of the X1 clock must be confirmed by the oscillation stabilization time counter status register (OSTC).

Caution: If you use an external clock input from the EXCLK pin, there is no need for an oscillation stabilization wait time.

4.6 Clock Control

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

The CPU/peripheral hardware clock (F_{CLK}) must run at high-speed on-chip oscillator clock. High-speed on-chip oscillator frequencies can be selected from 64MHz, 48MHz, 32MHz, 24MHz, 16MHz, 12MHz, 8MHz, 6MHz, 4MHz, 3MHz, 2MHz and 1MHz by FRQSEL0~FRQSEL4 bits of option bytes (000C2H). In addition, the frequency can be changed by the high-speed on-chip oscillator register (HOCODIV).

[Settings for option bytes]

Address: 000C2H

Option bytes

	7	6	5	4	3	2	1	0
(000C2H)	1	1	1	FRQSEL4 0/1	FRQSEL3 0/1	FRQSEL2 0/1	FRQSEL1 0/1	FRQSEL0 0/1

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	High-speed on-chip oscillator frequency	
					f_{HOCO}	f_{IH}
1	0	0	0	0	64MHz	64MHz
0	1	0	0	0	48MHz	48MHz
1	0	0	0	1	64MHz	32MHz
0	1	0	0	1	48MHz	24MHz
1	0	0	1	0	64MHz	16MHz
0	1	0	1	0	48MHz	12MHz
1	0	0	1	1	64MHz	8MHz
0	1	0	1	1	48MHz	6MHz
1	0	1	0	0	64MHz	4MHz
0	1	1	0	0	48MHz	3MHz
1	0	1	0	1	64MHz	2MHz
Other than the above					Prohibited settings.	

[Setting of HOCODIV for high-speed on-chip oscillator]

Address: 0x40021C20

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

HOCODIV2	HOCODIV1	HOCODIV0	Selection of clock frequency for high-speed on-chip oscillator			
			FRQSEL4=0		FRQSEL4=1	
			FRQSEL3=0	FRQSEL3=1	FRQSEL3=0	FRQSEL3=1
0	0	0	Prohibit settings.	$f_{IH}=48\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=64\text{MHz}$ $f_{HOCO}=64\text{MHz}$	Prohibit settings.
0	0	1	Prohibit settings.	$f_{IH}=24\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=32\text{MHz}$ $f_{HOCO}=64\text{MHz}$	Prohibit settings.
0	1	0	Prohibit settings.	$f_{IH}=12\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=16\text{MHz}$ $f_{HOCO}=64\text{MHz}$	Prohibit settings.
0	1	1	Prohibit settings.	$f_{IH}=6\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=8\text{MHz}$ $f_{HOCO}=64\text{MHz}$	Prohibit settings.
1	0	0	Prohibit settings.	$f_{IH}=3\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=4\text{MHz}$ $f_{HOCO}=64\text{MHz}$	Prohibit settings.
1	0	1	Prohibit settings.	Prohibit settings.	$f_{IH}=2\text{MHz}$ $f_{HOCO}=64\text{MHz}$	Prohibit settings.
Other than the above			Prohibit settings.			

4.6.2 Example of Setting X1 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 X1 oscillation clock, set the oscillator and start oscillation by using the oscillation stabilization time select register (OSTS) and clock operation mode control register (CMC) and clock operation status control register (CSC) and wait for oscillation to stabilize by using the oscillation stabilization time counter status register (OSTC). After the oscillation stabilizes, set the X1 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 Set the OSCSEL bit of the CMC register to “1”, and set the AMPH bit to “1” when $F_X \geq 10$ MHz.

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS	0	AMPHS1	AMPHS0	AMPH
	0	1	0	0		0	0	0/1

- ② Using the OSTS register, select the oscillation stabilization time of the X1 oscillator at releasing of the deep sleep mode.

Example) Setting values when a wait of at least 102 μ s is set based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0
						0	1	0

- ③ Clear the MSTOP bit of the CSC register to start oscillating the X1 oscillator.

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

- ④ Use the OSTC register to wait for oscillation of the X1 oscillator to stabilize.

Example) Wait until the bits reach the following values when a wait of at least 102 μ s is set based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18
	1	1	1	0	0	0	0	0

- ⑤ Use the MCM0 bit of the CKC register to specify the X1 oscillation clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	0	0
	0	0	0	1				

4.6.3 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 To run only the real-time clock and 15-bit interval timer on the subsystem clock (ultra-low current consumption) when in the deep sleep mode or sleep mode during CPU operation on the subsystem clock, set the RTCLPC bit to 1.

	7	6	5	4	3	2	1	0
OSMC	RTCLPC 0/1	0	0	WUTMCK0 0	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	EXCLK 0	OSCSEL 0	EXCLKS 0	OSCSELS 1	0	AMPHS1 0/1	AMPHS0 0/1	AMPH 0

AMPHS0 and AMPHS1 bits: These bits are used to specify the oscillation mode of the XT1 oscillator.

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

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

- ④ Use the timer function or another function to wait for oscillation of the subsystem clock to stabilize by using software or timer.

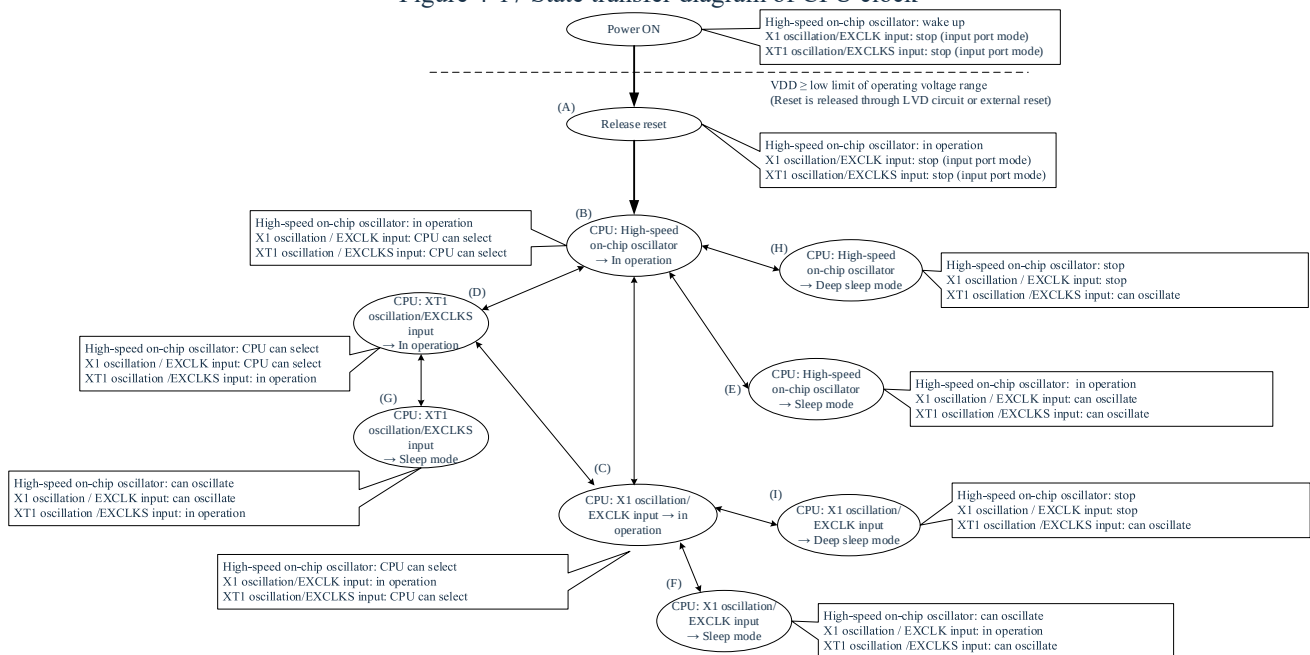
- ⑤ Use the CSS bit of the CKC register to specify the XT1 oscillation clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS 0	CSS 0	MCS 0	MCM0 1	0	0	0	0

4.6.4 State Transition Diagram of CPU Clock

The CPU clock state transition diagram of this product is as shown in Figure 4-17.

Figure 4-17 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 (1/5)

(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 high-speed system clock operation (C).

(The CPU is shifted to the high-speed on-chip oscillator clock operation (B) after the reset is released)

(Order in which SFR registers are set) →

Settings flag for SFR State transition	CMC register ^{Note1}			OSTS register	CSC register	OSTC register	CKC register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(A)→(B)→(C) (X1 clock: $1\text{MHz} \leq f_X \leq 10\text{MHz}$)	0	1	0	Note2	0	Confirmation required	1
(A)→(B)→(C) (X1 clock: $10\text{MHz} < f_X \leq 20\text{MHz}$)	0	1	1	Note2	0	Confirmation required	1
(A)→(B)→(C) (External main clock)	1	1	×	Note2	0	No confirmation required	1

Notes:

- After the reset is released, the clock operation mode control register (CMC) can only be written 1 time by 8-bit memory manipulation instructions.
- The oscillation stabilization time of the oscillation stabilization time selection register (OSTS) must be set as follows:
 - The oscillation stabilization time of the OSTC $\leq$ the oscillation stabilization time set by the OSTS register

Caution: The clock must be set after the supply voltage reaches the set clock runnable voltage (refer to the datasheet for electrical characteristics).

(3) After the reset (A) is released, the CPU is shifted to the subsystem clock operation (D).

(The CPU is shifted to the high-speed on-chip oscillator clock operation (B) after released)

(Order in which SFR registers are set) →

Settings flag for SFR State transition	CMC register ^{Note1}				CSC register	Oscillation stable waiting	CKC register
	EXCLKS	OSCSELS	AMPHS1	AMPHS0	XTSTOP		CSS
(A)→(B)→(D) (XT1 clock)	0	1	0/1	0/1	0	Yes	1
(A)→(B)→(D) (External sub-clock)	1	1	×	×	0	Yes	1

Note The clock operation mode control register (CMC) can only be written 1 time by 8-bit memory manipulation instructions after the reset is released.

Remarks: 1. ×: Ignore

- (A) to (I) of Table 4-3 corresponds to (A) to (I) of Figure 4-17.

Table 4-3 Examples of CPU clock transfer and SFR register set-up (2/5)

(4) The CPU shifts from the high-speed on-chip oscillator clock operation (B) to the high-speed system clock operation (C).

(Order in which SFR registers are set) →

Settings flag for SFR State transition	CMC register ^{Note1}			OSTS register	CSC register	OSTC register	CKC register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(B) →(C) (X1 clock: $1\text{MHz} \leq f_X \leq 10\text{MHz}$)	0	1	0	Note 2	0	Confirmation required	1
(B) →(C) (X1 clock: $10\text{MHz} < f_X \leq 20\text{MHz}$)	0	1	1	Note 2	0	Confirmation required	1
(B) →(C) (External main clock)	1	1	×	Note 2	0	No confirmation required	1

Not required if it has already been set. Not required in high-speed system clock operation.

Notes:

1. After the reset is released, the clock operation mode control register (CMC) can only be set once. If it has been set, no further action is needed.

2. The oscillation stabilization time of the oscillation stabilization time selection register (OSTS) must be set as follows:

- The expected oscillation stabilization time of the OSTC $\leq$ the oscillation stabilization time set by the OSTS register

Caution: The clock must be set after the supply voltage reaches the set clock runnable voltage (refer to the datasheet for electrical characteristics).

(5) The CPU shifts from the high-speed on-chip oscillator clock operation (B) to the subsystem clock operation (C).

(Order in which SFR registers are set) →

Settings flag for SFR State transition	CMC register ^{Note1}			CSC register	Oscillation stable waiting	CKC register
	EXCLKS	OSCSELS	AMPHS1, 0	XTSTOP		CSS
(B) →(D) (XT1 clock)	0	1	00: Low-power oscillation 01: Normal oscillation 10: Ultra-low power oscillation	0	Yes	1
(B) →(D) (External sub-clock)	1	1	×	0	Yes	1

Not required if it has already been set. Not required in subsystem clock operation

Note: After the reset is released, the clock operation mode control register (CMC) can only be written once by an 8-bit manipulation instruction. If it has been set, no further action is needed.

Remarks: 1. ×: Ignore

2. (A) to (I) of Table 4-3 corresponds to (A) to (I) of Figure 4-17.

Table 4-3 Examples of CPU clock transfer and SFR register set-up (3/5)

(6) The CPU shifts from the high-speed system clock operation (C) to the high-speed on-chip oscillator clock operation (B).

(Order in which SFR registers are set) →

State transition	Settings flag for SFR	CSC register	Oscillation stable waiting	CKC register
		HIOSTOP		MCM0
(C) → (B)		0	1us	0

Not required in high-speed on-chip oscillator clock operation

Remark The oscillation accuracy stable waiting time of the high-speed on-chip oscillator clock varies due to temperature conditions and during deep sleep mode.

(7) The CPU shifts from the high-speed system clock operation (C) to the subsystem clock operation (D).

(Order in which SFR registers are set) →

State transition	Settings flag for SFR	CSC register	Oscillation stable waiting	CKC register
		XTSTOP		CSS
(C) → (D)		0	Yes	1

Not required in subsystem clock operation

(8) The CPU shifts from the subsystem clock operation (D) to the high-speed on-chip clock operation (B).

(Order in which SFR registers are set) →

State transition	Settings flag for SFR	CSC register	Oscillation stable waiting	CKC register
		HIOSTOP		CSS
(D) → (B)		0	1us	0

Not required in high-speed on-chip oscillator clock operation

Remarks:

1. (A) to (I) of Table 4-3 corresponds to (A) to (I) of Figure 4-17.
2. The oscillation accuracy stable waiting 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 (4/5)

(9) The CPU shifts from the subsystem clock operation (D) to the high-speed system clock operation (B).

(Order in which SFR registers are set) →

Settings flag for SFR State transition	OSTS register	CSC register	OSTC register	CKC register
		MSTOP		CSS
(D) →(C) (X1 clock: $1\text{MHz} \leq f_X \leq 10\text{MHz}$)	Note	0	Confirmation required	0
(D) →(C) (X1 clock: $10\text{MHz} < f_X \leq 20\text{MHz}$)	Note	0	Confirmation required	0
(D) →(C) (External main clock)	Note	0	No confirmation required	0

Not required in high-speed system clock operation

Note: The oscillation stabilization time of the oscillation stabilization time selection register (OSTS) must be set as follows:

- The expected oscillation stabilization time of the OSTC $\leq$ the oscillation stabilization time set by the OSTS register

Caution: The clock must be set after the supply voltage reaches the set clock runnable voltage (refer to the datasheet for electrical characteristics).

- (10)• The CPU shifts from the high-speed on-chip oscillator clock operation (B) to the sleep mode (E).
- The CPU shifts from the high-speed system clock operation (C) to the sleep mode (F).
 - The CPU shifts from the subsystem clock operation (D) to the sleep mode (G).

State transition	Setting content
(B) →(E) (C) →(F) (D) →(G)	Execute WFI instructions.

Remark (A) to (I) of Table 4-3 corresponds to (A) to (I) of Figure 4-17.

Table 4-3 Examples of CPU clock transfer and SFR register set-up (5/5)

- (11)• The CPU shifts from the high-speed on-chip oscillator clock operation (B) to the deep sleep mode (H).
- The CPU shifts from the high-speed system clock operation (C) to the deep sleep mode (I).

(Setting order) →

State transition		Setting content		
(B) →(H)		Stop	—	The bit2 (SLEEPDEEP) of SCR register is set to 1, and the WFI instruction is executed.
(C) →(I)	X1 oscillator	Peripheral functions that cannot be run in deep sleep mode.	Set the OSTS register.	
	External clock		—	

Remark (A) to (I) of Table 4-3 corresponds to (A) to (I) of Figure 4-17.

4.6.5 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 (1/2)

CPU clock		Conditions before change	Processing after change
Before change	After change		
High-speed on-chip oscillator clock	X1 clock	X1 oscillation is stable. • OSCSEL=1, EXCLK=0, MSTOP=0 • After elapse of oscillation stabilization time	If the high-speed on-chip oscillator is stopped (HIOSTOP=1), the operating current can be reduced.
	External main system clock	Set the external clock of the EXCLK pin input to be valid. • OSCSEL=1, EXCLK=1, MSTOP=0	
	XT1 clock	XT1 oscillation is stable. • OSCSELS=1, EXCLKS=0, XTSTOP=0 • After elapse of oscillation stabilization time	
	External subsystem clock	Set the external clock of the EXCLKS pin input to be valid. • OSCSELS=1, EXCLKS=1, XTSTOP=0	
X1 clock	High-speed on-chip oscillator clock	Enable high-speed on-chip oscillator oscillation. • HIOSTOP=0 • After elapse of oscillation stabilization time	Can stop the oscillation of X1 (MSTOP=1).
	External main system clock	Transition is not possible.	—
	XT1 clock	XT1 oscillation is stable. • OSCSELS=1, EXCLKS=0, XTSTOP=0 • After elapse of oscillation stabilization time	Can stop the oscillation of X1 (MSTOP=1).
	External subsystem clock	Set the external clock of the EXCLKS pin input to be valid. • OSCSELS=1, EXCLKS=1, XTSTOP=0	Can stop the oscillation of X1 (MSTOP=1).
External main system clock	High-speed on-chip oscillator clock	Enable high-speed on-chip oscillator oscillation. • HIOSTOP=0 • After elapse of oscillation stabilization time	Able to disable the input of the external main system clock (MSTOP=1)
	X1 clock	Transition is not possible.	—
	XT1 clock	XT1 oscillation is stable. • OSCSELS=1, EXCLKS=0, XTSTOP=0 • After elapse of oscillation stabilization time	Able to disable the input of the external main system clock (MSTOP=1)
	External subsystem clock	Set the external clock of the EXCLKS pin input to be valid. • OSCSELS=1, EXCLKS=1, XTSTOP=0	Able to disable the input of the external main system clock (MSTOP=1)

Table 4-4 Changing CPU clock (2/2)

CPU clock		Conditions before change	Processing after change
Before change	After change		
XT1 clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is oscillating and the high-speed on-chip oscillator clock is selected as the main system clock. • HIOSTOP=0, MCS=0	Can stop the oscillation of XT1 (XTSTOP=1)
	X1 clock	The X1 oscillation is stable and the high-speed system clock is selected as the main system clock. • OSCSEL=1, EXCLK=0, MSTOP=0 • After elapse of oscillation stabilization time • MCS=1	
	External main system clock	Set the external clock of the EXCLK pin input to active and select the high-speed system clock as the main system clock. • OSCSEL=1, EXCLK=1, MSTOP=0 • MCS=1	
	External subsystem clock	Transition is not possible.	—
External subsystem clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is oscillating and the high-speed on-chip oscillator clock is selected as the main system clock. • HIOSTOP=0, MCS=0	Able to disable the input of the external subsystem clock (XTSTOP=1)
	X1 clock	X1 oscillation is stable and the high-speed system clock is selected as the main system clock. • OSCSEL=1, EXCLK=0, MSTOP=0 • After elapse of oscillation stabilization time • MCS=1	
	External main system clock	Set the external clock of the EXCLK pin input to active and select the high-speed system clock as the main system clock. • OSCSEL=1, EXCLK=1, MSTOP=0 • MCS=1	
	XT1 clock	Transition is not possible.	—

4.6.6 Time Required to Switch CPU Clock and Main System Clock

It can switch CPU clock (main system clock↔sub system clock) and main system clock (high-speed on-chip oscillator clock↔high-speed system clock) by setting bit6 and bit4 (CSS, MCM0) of system clock control register.

The actual switchover does not occur immediately after the CKC register is overridden, but several clocks continue to run with the clock before the switchover after the CKC register is changed (see Table 4-5~Table 4-7).

The CPU can be judged by the bit7 (CLS) of the CKC register whether the CPU is run with the main system clock or the sub system clock. The bit5 (MCS) of the CKC register can be used to determine whether the main system clock operates with a high-speed system clock or a high-speed on-chip oscillator clock.

If you switch the CPU clock, switch the peripheral hardware clock at the same time.

Table 4-5 Maximum time required to switch main system clock

Clock A	Switch direction	Clock B	Remark
f_{IH}	↔	f_{MX}	Refer to Table 4-6.
f_{MAIN}	↔	f_{SUB}	Refer to Table 4-7.

Table 4-6 Maximum number of clocks required for $f_{IH}↔f_{MX}$

Set value before switching		Set value after switching	
MCM0		MCM0	
		0 ($f_{MAIN}=f_{IH}$)	1 ($f_{MAIN}=f_{MX}$)
0 ($f_{MAIN}=f_{IH}$)	$f_{MX}≥f_{IH}$		2 clocks
	$f_{MX}<f_{IH}$		$2 f_{IH}/f_{MX}$ clocks
1 ($f_{MAIN}=f_{MX}$)	$f_{MX}≥f_{IH}$	$2 f_{MX}/f_{IH}$ clocks	
	$f_{MX}<f_{IH}$	2 clocks	

Table 4-7 Maximum number of clocks required for $f_{MAIN}↔f_{SUB}$

Set value before switching		Set value after switching	
CSS		CSS	
		0 ($f_{CLK}=f_{MAIN}$)	1 ($f_{CLK}=f_{SUB}$)
0 ($f_{CLK}=f_{MAIN}$)			$1+2 f_{MAIN}/f_{SUB}$ clocks
1 ($f_{CLK}=f_{SUB}$)		3 clocks	

Remarks:

1. The number of clocks in Table 4-6 and Table 4-7 is the number of CPU clocks before the switch.
2. The number of clocks in Table 4-6 and Table 4-7 is the number of clocks rounded to the decimal portion.

Example when switching the main system clock from the high-speed system clock to the high-speed on-chip oscillator clock (oscillation with $F_{IH}=8\text{MHz}$, $F_{MX}=10\text{MHz}$)

$$2F_{MX}/F_{IH}=2(10/8)=2.5→3 \text{ clocks}$$

4.6.7 Conditions Before Clock Oscillation Is Stopped

The register flag settings for stopping clock oscillations (invalid external clock input) and the conditions before stopping are as follows.

Table 4-8 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	MCS=1 or CLS=1 (The CPU is operating on a clock other than the high-speed on-chip oscillator clock)	HIOSTOP=1
X1 clock	MCS=0 or CLS=1	MSTOP=1
External main system clock	(The CPU is operating on a clock other than the high-speed system clock)	
XT1 clock	CLS=0	XTSTOP=1
External subsystem clock	The CPU is operating on a clock other than the subsystem clock)	

Chapter 5 Universal Timer Unit (Timer4)

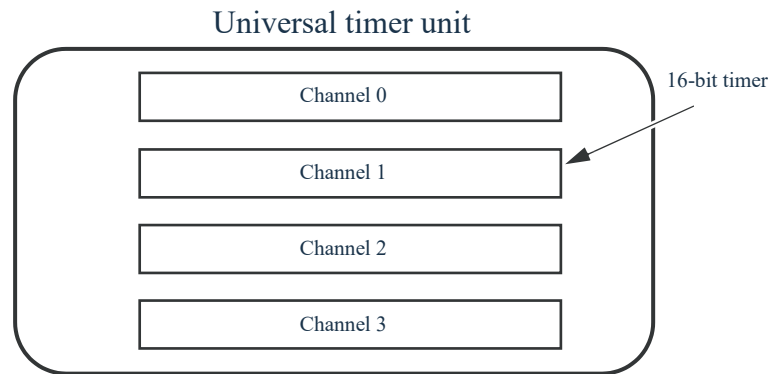
This product features two general-purpose timer units, each containing 4 channels. The pins of this product have alternated functions and can be configured to any port except P00, allowing flexible configuration of the number of channels for the general-purpose timer units.

Description:

1. The symbol m in the following part of this chapter represents the unit number. This product is equipped with two universal timers Timer4, so $m=0, 1$.
2. The symbol n in the following of this chapter represents the channel number (in this chapter $n=0\sim3$).
3. The following contents of this chapter are limited to 24-pin products for description.

Each universal timer unit has four 16-bit timers.

Each 16-bit timer is called a “channel” and can be used separately as a timer or combined with multiple channels for advanced timer functions.



For details on each feature, refer to the table below.

Independent channel operation function	Multi-channel linkage function
• Interval timer (→refer to 5.8.1) • Square wave output (→refer to 5.8.1) • External event counter (→refer to 5.8.2) • Frequency divider (→refer to 5.8.3) • Measurement of the input pulse interval (→refer to 5.8.4) • Measurement of the high and low level width of the input signal (→refer to 5.8.5) • Delay counter (→refer to 5.8.6)	• Single trigger pulse output (→refer to 5.9.1) • PWM output (→refer to 5.9.2) • Multiple PWM output (→refer to 5.9.3)

The 16-bit timers of Channel 1 and Channel 3 in Unit 0 can be used as two 8-bit timers (high and low bits). The functionalities of Channel 1 and Channel 3 as 8-bit timers are as follows:

- Interval timer (high 8-bit and low 8-bit timers)/square wave output (low 8-bit timers only)
- External event counter (low 8-bit timers only)
- Delay counter (low 8-bit timers only)

5.1 Functions of Universal Timer Unit

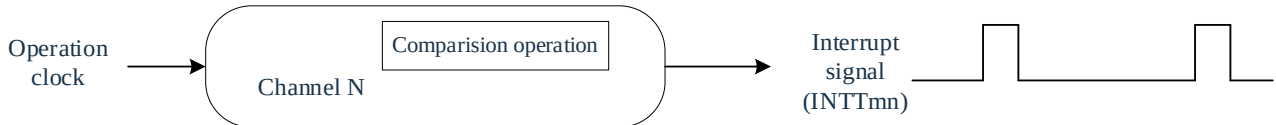
The universal timer unit has the following functions:

5.1.1 Independent Channel Operation

The independent channel operation function allows each channel to operate independently, without being affected by the operation modes of other channels.

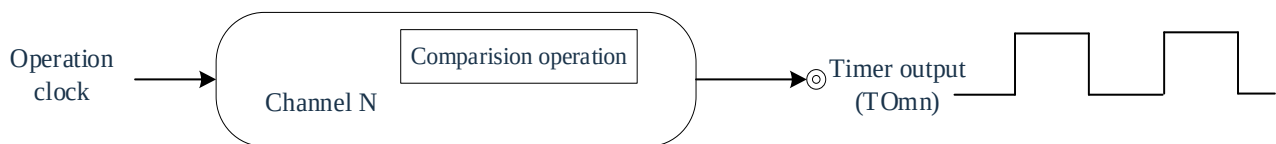
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



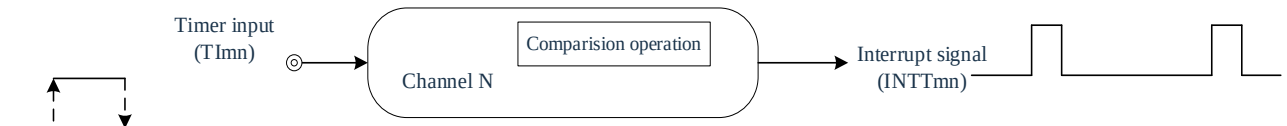
(2) Square wave output

A toggle operation is performed each time INTTMmn interrupt is generated and a square wave with a duty cycle of 50% is output from a timer output pin (TOMn).



(3) External event counter

The valid edge of the input signal of the timer input pin (TIMn) is counted, and if the specified number of times is reached, it can be used as an event counter for generating interrupts.



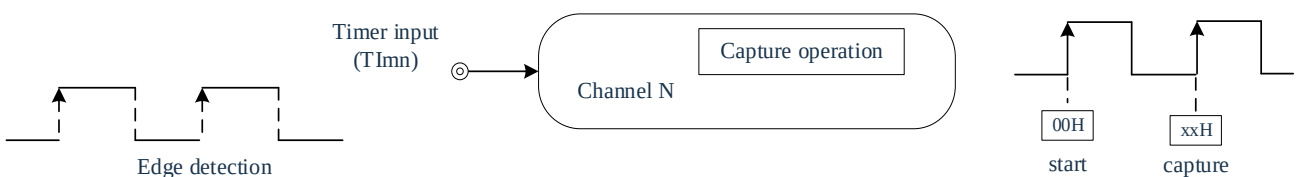
(4) Frequency divider function (channel 0 of unit 0 only)

The input clock from the timer input pin (TI00) is divided and then output from the output pin (TO00).



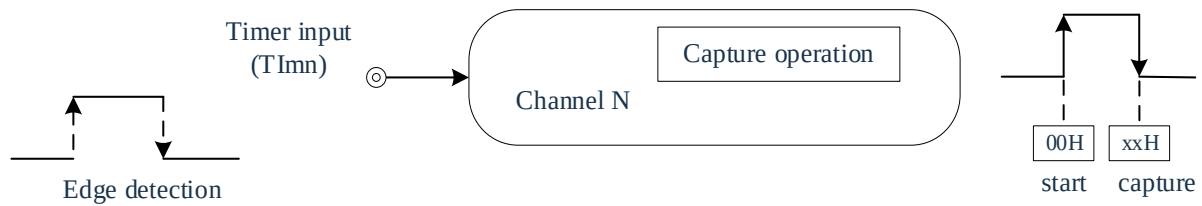
(5) Measurement of input pulse interval

The interval between input pulses is measured by starting counting at the active edge of the input pulse signal at the timer input pin (TIMn) and capturing the count value at the active edge of the next pulse.



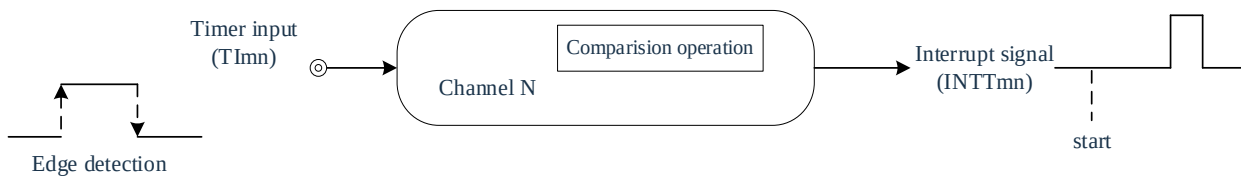
(6) Measurement of the high-/low-level width of the input signal

The high- and low-level width of the input signal is measured by starting the count on one edge of the input signal at the timer input pin (TIMn) and capturing the count value on the other edge.



(7) Delay counter

Counting begins on the active edge of the input signal to the timer input pin (TIMn) and an interrupt is generated after an arbitrary delay period.



Remark: 1. m: unit number (m=0, 1) n: channel number (n=0~3)

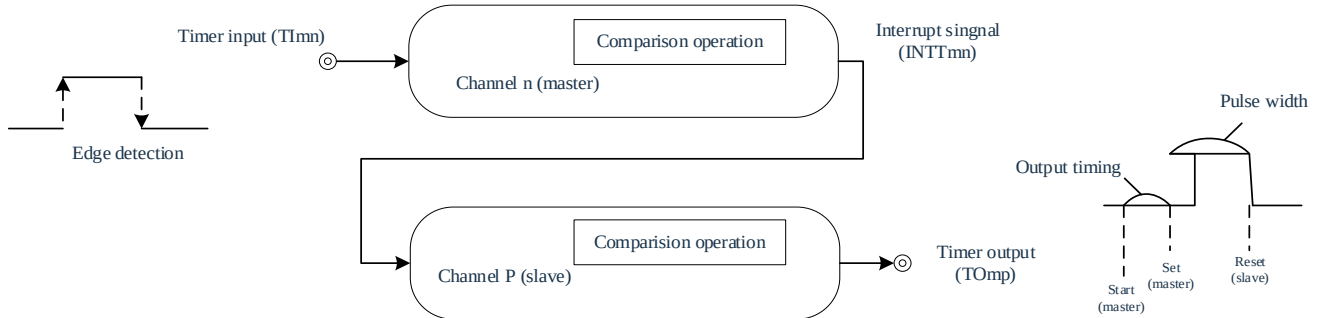
5.1.2 Multi-Channel Linkage Operation Function

The multi-channel linked operation function is a combination of a master channel (the reference timer for the master control cycle) and a slave channel (a timer that operates in compliance with the master channel).

The multi-channel linkage operation function can be used as the following modes.

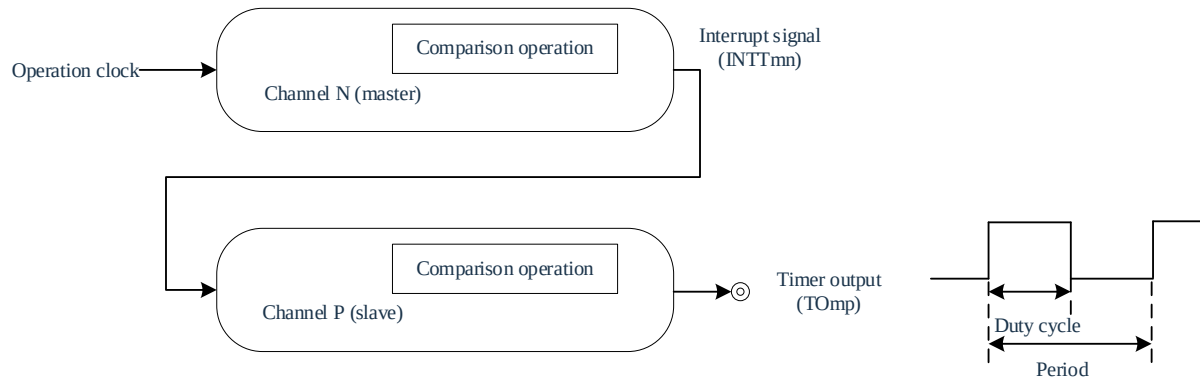
(1) Single trigger pulse output

Using the 2 channels in pairs, a single trigger pulse with arbitrary output timing and pulse width can be generated.



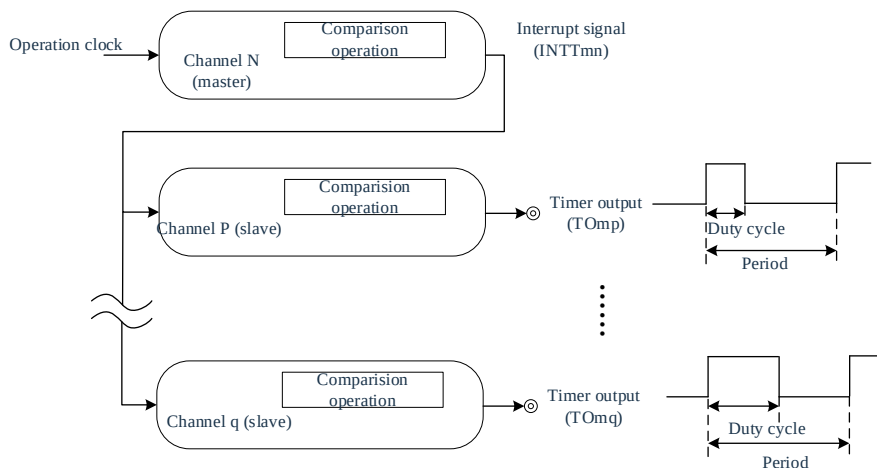
(2) PWM (Pulse Width Modulation) output

Using the 2 channels in pairs, pulses with arbitrary period and duty cycle can be generated.



(3) Multiple PWM (Pulse Width Modulation) output

The PWM function can be extended to generate up to 3 PWM signals of any duty cycle with a fixed period using one master channel and multiple slave channels.



Caution: Please refer to “5.4.1 Basic rules of multi-channel linkage operation function” for the rule details of multi-channel linkage operation function.

Remark: m: unit number (m=0, 1) n: channel number (n=0~3) p, q: slave channel number (n<p<q≤3)

5.1.3 8-Bit Timer Operation Function (Channels 1 and 3 of Unit 0 Only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels. This function can only be used for channels 1 and 3.

Caution: There are several rules for using 8-bit timer operation function.

For details, see 5.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only).

5.2 Structure of Universal Timer Unit

The universal timer unit consists of the following hardware.

Table 5-1 Structure of universal timer unit

Item	Structure
Counter	Timer count register mn (TCRmn).
Register	Timer data register mn (TDRmn)
Timer input	TI00~TI03 ^{Note1} , TI10~TI13 ^{Note1}
Output of timer	TO00~TO03 ^{Note1} , TO10~TO13 ^{Note1} , output control circuit
Control register	<Registers of unit setting section> • Peripheral enable register 0 (PER0) • Timer clock select register m (TPSm) • Timer channel enable status register m (TEm) • Timer channel start register m (TSM) • Timer channel stop register m (TTm) • Timer input select register 0 (TIOS0) ^{Note2} • Timer output enable register m (TOEm) • Timer output register m (TOM) • Timer output level register m (TOLm) • Timer output mode register m (TOMm)
	<Registers of each channel> • Timer mode register mn (TMRmn) • Timer state register mn (TSRmn) • Noise filter enable registers 1,2 (NFEN1, NFEN2) • Port mode control register (PMCxx) ^{Note 3} • Port mode register (PMxx) ^{Note 3} • Port alternate function configuration register (PxxCFG) ^{Note 3}

Note 1: The timer input/output pins of Channels 0 to 3 can be configured to any port except P00. For detailed information, refer to Section 2.3.10.

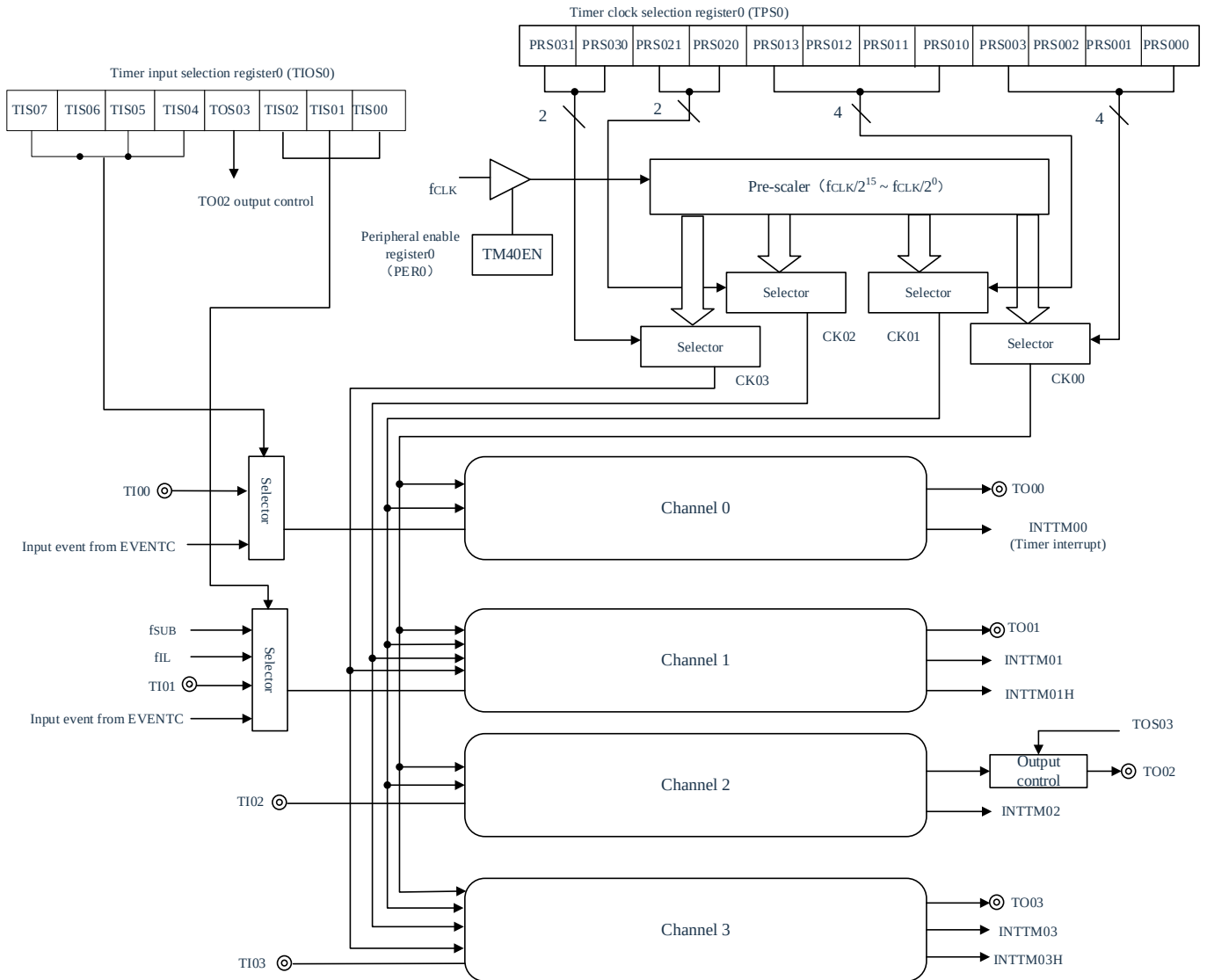
Note 2: For channel selection in Unit 0 only.

Note 3: Configuration of timer input/output pins for Channels 0 to 3. For detailed information, refer to Section 2.3.10.

Remark m: unit number (m=0, 1) n: channel number (n=0~3)

The block diagram of universal timer unit is shown in Figure 5-1.

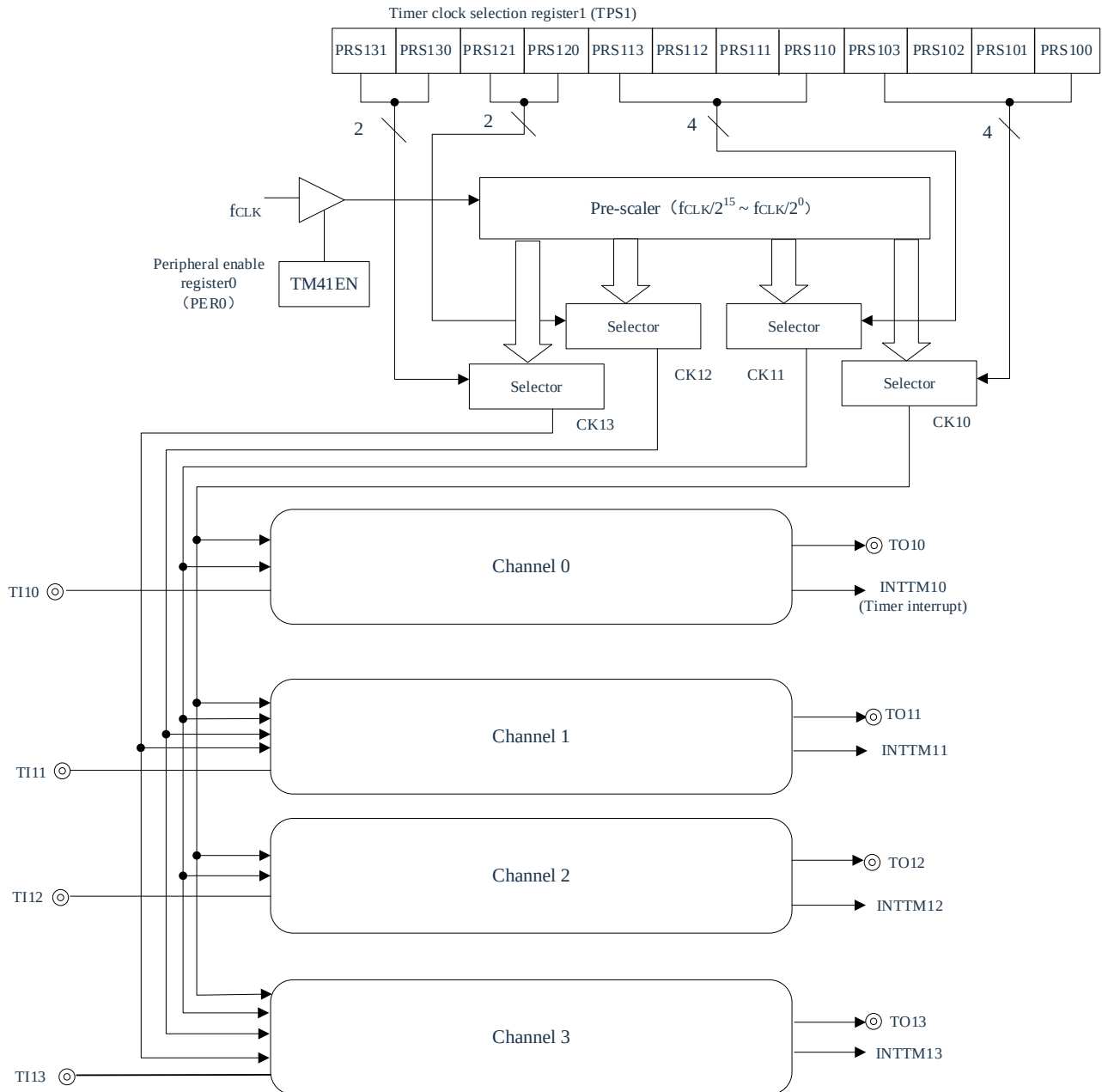
Figure 5-1 Overall block diagram of universal timer unit 0



Remark: f_{SUB} : Subsystem clock frequency

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

Figure 5-2 Overall block diagram of universal timer unit 1



5.2.1 List of Universal Timer Unit 0 Registers

Unit 0 register base address: 0x40041C00

Offset address	Register name	R/W	Bit width	Reset value
0x180	TCR00	R	16	FFFFH
0x182	TCR01	R	16	FFFFH
0x184	TCR02	R	16	FFFFH
0x186	TCR03	R	16	FFFFH
0x190	TMR00	R/W	16	0000H
0x192	TMR01	R/W	16	0000H
0x194	TMR02	R/W	16	0000H
0x196	TMR03	R/W	16	0000H
0x1A0	TSR00	R	16	0000H
0x1A0	TSR00L	R	8	00H
0x1A2	TSR01	R	16	0000H
0x1A2	TSR01L	R	8	00H
0x1A4	TSR02	R	16	0000H
0x1A4	TSR02L	R	8	00H
0x1A6	TSR03	R	16	0000H
0x1A6	TSR03L	R	8	00H
0x1B0	TE0	R	16	0000H
0x1B0	TE0L	R	8	00H
0x1B2	TS0	R/W	16	0000H
0x1B2	TS0L	R/W	8	00H
0x1B4	TT0	R/W	16	0000H
0x1B4	TT0L	R/W	8	00H
0x1B6	TPS0	R/W	16	0000H
0x1B8	TO0	R/W	16	0000H
0x1B8	TO0L	R/W	8	00H
0x1BA	TOE0	R/W	16	0000H
0x1BA	TOE0L	R/W	8	00H
0x1BC	TOL0	R/W	16	0000H
0x1BC	TOL0L	R/W	8	00H
0x1BE	TOM0	R/W	16	0000H
0x1BE	TOM0L	R/W	8	00H
0x310	TDR00	R/W	16	0000H
0x312	TDR01	R/W	16	0000H
0x312	TDR01L	R/W	8	00H
0x313	TDR01H	R/W	8	00H
0x314	TDR02	R/W	16	0000H
0x316	TDR03	R/W	16	0000H
0x316	TDR03L	R/W	8	00H
0x317	TDR03H	R/W	8	00H

5.2.2 List of Universal Timer Unit 1 Registers

Unit 1 register base address: 0x40042000

Offset address	Register name	R/W	Bit width	Reset value
0x180	TCR10	R	16	FFFFH
0x182	TCR11	R	16	FFFFH
0x184	TCR12	R	16	FFFFH
0x186	TCR13	R	16	FFFFH
0x190	TMR10	R/W	16	0000H
0x192	TMR11	R/W	16	0000H
0x194	TMR12	R/W	16	0000H
0x196	TMR13	R/W	16	0000H
0x1A0	TSR10	R	16	0000H
0x1A0	TSR10L	R	8	00H
0x1A2	TSR11	R	16	0000H
0x1A2	TSR11L	R	8	00H
0x1A4	TSR12	R	16	0000H
0x1A4	TSR12L	R	8	00H
0x1A6	TSR13	R	16	0000H
0x1A6	TSR13L	R	8	00H
0x1B0	TE1	R	16	0000H
0x1B0	TE1L	R	8	00H
0x1B2	TS1	R/W	16	0000H
0x1B2	TS1L	R/W	8	00H
0x1B4	TT1	R/W	16	0000H
0x1B4	TT1L	R/W	8	00H
0x1B6	TPS1	R/W	16	0000H
0x1B8	TO1	R/W	16	0000H
0x1B8	TO1L	R/W	8	00H
0x1BA	TOE1	R/W	16	0000H
0x1BA	TOE1L	R/W	8	00H
0x1BC	TOL1	R/W	16	0000H
0x1BC	TOL1L	R/W	8	00H
0x1BE	TOM1	R/W	16	0000H
0x1BE	TOM1L	R/W	8	00H
0x310	TDR10	R/W	16	0000H
0x312	TDR11	R/W	16	0000H
0x312	TDR11L	R/W	8	00H
0x313	TDR11H	R/W	8	00H
0x314	TDR12	R/W	16	0000H
0x316	TDR13	R/W	16	0000H
0x316	TDR13L	R/W	8	00H
0x317	TDR13H	R/W	8	00H

5.2.3 Timer Count Register mn (TCRmn)

The TCRmn register is a 16-bit read-only register that counts the count clock. The count is incremented or decremented synchronously with the rising edge of the count clock.

The operation mode is selected by the MDmn3 to MDmn0 bits of the Timer Mode Register mn (TMRmn) to switch between incremental and decremental counting (refer to 5.3.3: Timer Mode Register mn (TMRmn)).

Figure 5-3 Table of timer count register mn (TCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCRmn																

Remark: m: unit number (m=0, 1) n: channel number (n=0~3).

The count value can be read by reading the timer count register mn (TCRmn).

The count value becomes “FFFFH” in the following cases.

- When a reset signal is generated
- When clearing the TM4EN bit of the peripheral enable register 0 (PER0)
- At the end of the count of the slave channel in PWM output mode
- At the end of the count of the slave channel in delayed count mode
- At the end of counting of master/slave channels in single trigger pulse output mode
- At the end of the count of the slave channel in the multiple PWM output mode

The count value becomes “0000H” in the following cases.

- When input starts triggering in capture mode
- At the end of the capture in capture mode

Caution: Even if the TCRmn register is read, the count value is not captured to the timer data register mn(TDRmn).

As shown below, the read values of the TCRmn register vary depending on the operating mode and operating state.

Table 5-2 Read value of timer count register mn (TCRmn) in each running mode

Operation mode	Counting method	Timer count register mn (TCRmn) read value ^{Note}			
		Value if the operation mode was changed after releasing reset	Counting pause Value at (TTmn = 1)	Counting pause (TTmn=1) after changing the value of the operating mode	Wait after a single count The value at the start of the trigger
Interval timer mode	Count down	FFFFH	value when stopped	undefined	-
Capture Mode	Count up	0000H	value when stopped	undefined	-
Event counter mode	Count down	FFFFH	value when stopped	undefined	-
Single count mode	Count down	FFFFH	value when stopped	undefined	FFFFH
Capture & Single Count Mode	Count up	0000H	value when stopped	undefined	TDRmn register capture value +1

Note: It indicates the read value of the TCRmn register when channel n is in the timer stop state (TEmn=0) and the count enable state (TSmn=1). Hold this value in the TCRmn register until counting starts.

Remark: m: unit number (m=0, 1) n: channel number (n=0~3).

5.2.4 Timer Data Register mn (TDRmn)

This is a 16-bit register from which a capture function and a compare function can be selected. The capture or compare function can be switched by selecting an operation mode by using the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn).

The value of the TDRmn register can be changed at any time.

This register can be read or written in 16-bit units.

In addition, for the TDRm1 and TDRm3 registers, while in the 8-bit timer mode (when the SPLIT bits of timer mode registers m1 and m3 (TMRm1, TMRm3) are 1), it is possible to rewrite the data in 8-bit units, with TDRm1H and TDRm3H used as the higher 8 bits, and TDRm1L and TDRm3L used as the lower 8 bits.

Reset signal generation clears this register to 0000H.

Figure 5-4 Table of timer data register mn (TDRmn) (n=0, 2)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDRmn																

Figure 5-5 Table of timer data register mn(TDRmn) (n=1, 3)
(TDR01H supports 8-bit operations) (TDR01L supports 8-bit operations)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDRmn																

- i. The case where the timer data register mn (TDRmn) is used as a comparison register.
The count is decremented from the setpoint of the TDRmn register and generates an interrupt signal (INTTMmn) when the count value changes to 0000H. The value of the TDRmn register is maintained until it is rewritten.

Caution: Even if the input capture trigger signal, the TDRmn register set to the comparison function does not perform a capture operation.

- ii. The case where the timer data register mn (TDRmn) is used as the capture register.
The count value of the timer count register mn (TCRmn) is captured to the TDRmn register via input capture trigger. The effective edge of the TIMn pin can be selected as the capture trigger signal. The timing mode register mn (TMRmn) sets the selection of capture triggers.

Remark: m: unit number (m=0, 1) n: channel number (n=0~3).

5.3 Registers for Controlling Universal Timer Unit

The registers that control the universal timer unit are as follows:

- Peripheral enable register 0 (PER0)
- Timer clock select register m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register mn (TSRmn)
- Timer channel enable state register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer input/output select register (TIOS0)
- Timer output enable register m (TOEm)
- Timer output register m (TOM)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Noise filter enable register 1 (NFEN1)
- Noise filter enable register 2 (NFEN2)
- Port mode control register (PMCxx)
- Port mode register (PMxx)
- Port alternate function configuration register (PxxCFG)

Caution: The assigned registers and bits vary from product to product. The initial values must be set for unassigned bits.

Remark: m: unit number (m=0, 1) n: channel number (n=0~3).

5.3.1 Peripheral Enable Register 0 (PER0)

The PER0 register is a register that is set to enable or disable clocks to each peripheral hardware. Reduce power consumption and noise by stopping clocks to hardware that is not in use.

To use universal timer unit 0, the bit0 (TM40EN) must be set to “1”. The PER0 register is set via an 8-bit memory manipulation instruction. After generating a reset signal, the value of the PER0 register changes to “00H”.

Figure 5-6 Table for peripheral enable register 0 (PER0)

Address: 0x40020420 After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TM41EN	TM40EN

TM40EN	Control of the input clock of universal timer unit 0
0	Stop to supply the input clock. • The SFR used by universal timer unit 0 cannot be written. • Universal timer unit 0 is in the reset state.
1	Supply the input clock. • The SFR used by universal timer unit 0 can be read and written.

TM41EN	Control of the input clock of universal timer unit 1
0	Stop to supply the input clock. • The SFR used by universal timer unit 1 cannot be written. • Universal timer unit 1 is in the reset state.
1	Supply the input clock. • The SFR used by universal timer unit 1 can be read and written.

Caution: To set the universal timer unit, the following registers must be set with the TM4mEN bit at “1”. When the TM4mEN bit is “0”, the values of the Timer Array Unit's control registers are initialized, and write operations are ignored (timer input/output select register 0 (TIOS0), noise filter enable register 1 (NFEN1), noise filter enable register 2 (NFEN2), port mode control register (PMCx), port mode register (PMx), and port alternate function configuration register (PxxCFG) are excluded).

- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer output enable register m (TOEm)
- Timer output register m (TOM)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)

5.3.2 Timer Clock Select Register m (TPSm)

The TPSm register is a 16-bit register that selects the two or four common operating clocks (CKm0, CKm1, CKm2, CKm3) provided to each channel. CKm0 is selected via bits 3~0 of the TPSm register, and CKm1 is selected via bits 7~4 of the TPSm register. In addition, only channel 1 and channel 3 can select CKm2 and CKm3, and CKm2 is selected via bits 9~8 of the TPSm register, and CKm3 is selected via bits 13 and 12 of the TPSm register.

The TPSm register in timer operation can only be rewritten in the following cases.

If the PRSm00 to PRSm03 bits can be rewritten (n=0~3):

All channels for which CKm0 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 0) are stopped (TEmn = 0).

If the PRSm10 to PRSm13 bits can be rewritten (n=0~3):

All channels for which CKm2 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 1) are stopped (TEmn = 0).

If the PRSm20 and PRSm21 bits can be rewritten (n = 1, 3):

All channels for which CKm1 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 0) are stopped (TEmn = 0).

If the PRSm30 and PRSm31 bits can be rewritten (n = 1, 3):

All channels for which CKm3 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 1) are stopped (TEmn = 0).

The TPSm register is set by a 16-bit memory manipulation instruction. After a reset signal is generated, the value of the TPSm register changes to “0000H”.

Figure 5-7 Table of timer clock select register m (TPSm)(1/2)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRS m31	PRS m30	0	0	PRS m21	PRS m20	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRSmk3	PRSmk2	PRSmk1	PRSmk0	Selection of operation clock (CKmk) ^{Note} (k=0, 1)
0	0	0	0	f_{CLK}
0	0	0	1	$f_{CLK}/2$
0	0	1	0	$f_{CLK}/2^2$
0	0	1	1	$f_{CLK}/2^3$
0	1	0	0	$f_{CLK}/2^4$
0	1	0	1	$f_{CLK}/2^5$
0	1	1	0	$f_{CLK}/2^6$
0	1	1	1	$f_{CLK}/2^7$
1	0	0	0	$f_{CLK}/2^8$
1	0	0	1	$f_{CLK}/2^9$
1	0	1	0	$f_{CLK}/2^{10}$
1	0	1	1	$f_{CLK}/2^{11}$
1	1	0	0	$f_{CLK}/2^{12}$
1	1	0	1	$f_{CLK}/2^{13}$
1	1	1	0	$f_{CLK}/2^{14}$
1	1	1	1	$f_{CLK}/2^{15}$

Note: In case of changing the clock selected as F_{CLK} (changing the value of the system clock control register (CKC)), the general-purpose timer unit must be stopped (TTm=000FH). The general-purpose timer unit needs to be stopped even when the operation clock (F_{MCK}) is selected or when the active edge of the TIMn pin input signal is used.

Cautions:

1. Bits 15, 14, 11 and 10 must be set to "0".
2. If F_{CLK} (undivided) is selected as the operation clock (CKmk) and TDRnm is set to "0000H" (n=0, 1, m=0~3), the interrupt request of general-purpose timer unit cannot be used.

Remarks:

1. F_{CLK} : CPU/peripheral hardware clock frequency
2. The TPSm register selects a clock waveform that is high for only 1 F_{CLK} cycle from the rising edge. For details, refer to "5.5.1 Counting Clock (F_{TCLK})".

Figure 5-8 Table of timer clock select register m (TPSm) (2/2)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRSm31	PRSm30	0	0	PRSm21	PRSm20	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00

PRSm21	PRSm20	Selection of operation clock (CKm2) ^{Note}
0	0	$f_{CLK}/2$
0	1	$f_{CLK}/2^2$
1	0	$f_{CLK}/2^4$
1	1	$f_{CLK}/2^6$

PRSm31	PRSm30	Selection of operation clock (CKm3) ^{Note}
0	0	$f_{CLK}/2^8$
0	1	$f_{CLK}/2^{10}$
1	0	$f_{CLK}/2^{12}$
1	1	$f_{CLK}/2^{14}$

Note: The universal timer unit (TTm=000FH) must be stopped if the clock selected as F_{CLK} is changed (the value of the system clock control register (CKC) is changed). It is necessary to stop the general-purpose timer unit even when the operation clock (F_{MCK}) is selected or when the active edge of the input signal to the TIMn pin is selected.

Caution: Bits 15, 14, 11, 10 must be set to “0”.

If Channel 1 and Channel 3 are used in 8-bit timer mode and CKm2 and CKm3 are used as the operating clocks, the interval time shown in the following table can be achieved by using the interval timer function.

Table 5-3 Interval time that can be set by operation clocks CKSm2 and CKSm3

Clock		Interval time ^{Note} ($f_{CLK}=32\text{MHz}$)			
		10us	100us	1ms	10ms
CKm2	$f_{CLK}/2$	○	—	—	—
	$f_{CLK}/2^2$	○	—	—	—
	$f_{CLK}/2^4$	○	○	—	—
	$f_{CLK}/2^6$	○	○	—	—
CKm3	$f_{CLK}/2^8$	—	○	○	—
	$f_{CLK}/2^{10}$	—	○	○	—
	$f_{CLK}/2^{12}$	—	—	○	○
	$f_{CLK}/2^{14}$	—	—	○	○

Note: ○ The margin is within 5%.

Remarks:

1. F_{CLK} : CPU/peripheral hardware clock frequency
2. For details on the $f_{CLK}/2^n$ waveform selected for the TPSm register, refer to 5.5.1 Count clock (F_{TCLK}).

5.3.3 Timer Mode Register mn (TMRmn)

The TMRmn register is a register that sets the operation mode of channel n. It performs the selection of the operation clock (F_{MCK}), the selection of the count clock, the selection of master/slave, the selection of the 16-bit/8-bit timer (channel 1 and channel 3 of unit 0 only), the setting of the start trigger and the capture trigger, the selection of the effective edge of the timer input, and the operation modes (interval, capture, event counter, single count, capture & single count) settings.

It is prohibited to rewrite the TMRmn register during operation ($TE_{mn}=1$). However, bit7 and bit6 (CIS_{mn1} , CIS_{mn0}) can be rewritten during part of the function operation ($TE_{mn}=1$) (for details, refer to “5.8 Independent Channel Operation Function of Universal Timer Unit” and “5.9 Multi-Channel Operation Function of Universal Timer Unit”).

The TMRmn register is set by a 16-bit memory manipulation instruction. After a reset signal is generated, the value of the TMRmn register changes to “0000H”.

Caution: The bit11 of the TMRmn register varies from channel to channel.

TMRm2: MASTERmn bit ($n=2$)

TMRm1, TMRm3: SPLITmn bit ($n=1, 3$)

TMRm0: Fixed to 0.

Figure 5-9 Table of timer mode register mn (TMRmn) (1/4)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=2)	CKS mn1	CKS mn0	0	CCS mn	MAS TERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=1,3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=0)	CKS mn1	CKS mn0	0	CCS mn	0 ^{Note1}	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CKSmn1	CKSmn0	Selection of channel n operation clock (F _{MCK})
0	0	The operation clock CKm0 set by the timer clock selection register m (TPSm).
0	1	The operation clock CKm2 set by the timer clock selection register m (TPSm).
1	0	The operation clock CKm1 set by the timer clock selection register m (TPSm).
1	1	The operation clock CKm3 set by the timer clock selection register m (TPSm).
The operation clock (F _{MCK}) is used for edge detection circuits. The sample clock and count clock (F _{TCLK}) are generated by setting the CCSmn bit. Only Channel 1 and Channel 3 can select operation clocks CKm2 and CKm3.		

CCSmn	Selection of channel n count clock (F _{TCLK})
0	CKSmn0 bit and CKSmn1 bit specified operation clock (F _{MCK})
1	The active edge of the TIMn pin input signal Unit 0 status: Channel 0: The active edge of the input signal selected by TIS0 Channel 1: The active edge of the input signal selected by TIS0
Counting clocks (F _{TCLK}) are used in counters, output control circuits, and interrupt control circuits.	

Note 1: Bit11 is a read-only bit, fixed to “0”, ignoring write operations.

Cautions:

1. Bits 13, 5 and 4 must be set to 0.
2. To change the clock selected as F_{CLK} (change the value of the system clock control register (CKC)), the Timer Array Unit (TTm=0, 10FFH) must be stopped even if the operation clock (F_{MCK}) specified by the CKSmn0 bit and the CKSmn1 bit, or the active edge of the input signal to the TIMn pin, is selected as the count clock (F_{TCLK}).

Remark: m: unit number (m=0, 1) n: channel number (n=0~3).

Figure 5-10 Table of timer mode register mn (TMRmn) (2/4)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=2)	CKS mn1	CKS mn0	0	CCS mn	MAS TERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=1,3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=0)	CKS mn1	CKS mn0	0	CCS mn	0 ^{Note1}	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

(Bit11 of TMRmn (n=2))

MASTERmn	Selection of independent channel operation/multi-channel linked operation (slave or master) for channel n
0	Used as a slave channel for independent or multi-channel linked operation functions.
1	Used as a master control channel for the multi-channel linked operation function.

Only channels 2 can be set as the master channel (MASTERmn=1).
Channel 0 is fixed to “0” (since channel 0 is the channel with the highest bit, it is used as the master channel regardless of the setting of this bit).
For channels used as independent channel operation functions, set the MASTERmn bit to “0”.

(Bit11 of TMRmn (n=1, 3))

SPLITmn	Operation selection of 8-bit timer/16-bit timer for channel 1 and channel 3
0	Used as a 16-bit timer. (Used as a slave channel for independent channel operation or multi-channel linkage operation)
1	Used as an 8-bit timer.

STSmn2	STSmn1	STSmn0	Start trigger and capture trigger settings for channel n
0	0	0	Only software triggering is active at the start (no other trigger source is selected).
0	0	1	Use the active edge of the TIMn pin input for start triggering and capture triggering.
0	1	0	Use the rising edge and falling edge of the TIMn pin input for start triggering and capture triggering respectively.
1	0	0	Use interrupt signals from the master channel (in the case of slave channels with multi-channel linkage operation function).
Other than above			Disable setting.

Note 1: bit11 is a read-only bit, fixed to “0”, ignoring write operations.

Remark: m: unit number (m=0, 1) n: channel number (n=0~3).

Figure 5-11 Table of timer mode register mn (TMRmn) (3/4)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=2)	CKS mn1	CKS mn0	0	CCS mn	MAS TERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=1,3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=0)	CKS mn1	CKS mn0	0	CCS mn	0 ^{Note1}	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CISmn1	CISmn0	Active edge selection for TImn pins
0	0	Falling edge
0	1	Rising edge
1	0	Double edges (when measuring low level width) Start trigger: falling edge, capture Trigger: Rising edge
1	1	Double edges (when measuring high level width) Start trigger: rising edge, capture Trigger: Falling edge
When STSmn2~STSmn0 bit is not '010B' and double edge is specified, CISmn1~CISmn0 set to '10B'.		

Note: 1. Bit11 is read-only, fixed to “0”, ignoring write operations.

Remark: m: unit number (m=0, 1) n: channel number (n=0~3).

Figure 5-12 Table of timer mode register mn (TMRmn) (4/4)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=2)	CKS mn1	CKS mn0	0	CCS mn	MAS TERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=1,3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=0)	CKS mn1	CKS mn0	0	CCS mn	0 ^{Note1}	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

MD mn3	MD mn2	MD mn1	Setting of channel n operation mode	Corresponding functions	Count operation of TCR
0	0	0	Interval timer mode	Interval timer/square wave output/ Frequency divider function/PWM output (master)	Count down
0	1	0	Capture mode	Measurement of input pulse interval	Count up
0	1	1	Event counter mode	External event counter	Count down
1	0	0	Single count mode	Delay counter/single trigger pulse output/PWM output (slave)	Count down
1	1	0	Capture & Single count mode	Measurement of the high- and low-level width of the input signal	Count up
Other than the above			Settings are disabled.		
The operation of each mode varies depending on the MDmn0 bit (see the table below).					

Operation mode (Value set by the MDmn3 to MDmn1 bits (see table above))	MDmn0	Setting of starting counting and interrupt
- Interval timer mode (0, 0, 0) - Capture mode (0, 1, 0)	0	No timer interrupt is generated when counting starts (the output of the timer does not change).
	1	A timer interrupt is generated when counting starts (the output of the timer also changes).
Event counter mode (0, 1, 1)	0	No timer interrupt is generated when counting starts (the output of the timer does not change).
Single count mode ^{Note 2} (1, 0, 0)	0	The start trigger in the count operation is invalid. No interruption at this time.
	1	The start trigger in the count operation is valid ^{Note 3} . No interruption at this time.
Capture & single count mode (1, 1, 0)	0	No timer interrupt is generated when counting starts (the output of the timer does not change). The start trigger in the count operation is invalid. No interruption at this time.

Note 1: Bit11 is a read-only bit, fixed to “0”, and write is ignored.

Note 2: In single count mode, the interrupt output (INTTMmn) and TOMn output at the start of counting are not controlled.

Note 3: If a start trigger is generated during operation (TSMn=1), the counter is initialized and counting is restarted (no interrupt request is generated).

Remark: m: unit number (m=0, 1) n: channel number (n=0~3).

5.3.4 Timer Status Register mn (TSRmn)

The TSRmn register is a register that indicates the overflow status of the channel n counter.

The TSRmn register is valid only in capture mode (MDmn3~MDmn1=010B) and capture & single count mode (MDmn3~MDmn1=110B). Refer to Table 5-4 for the OVF bit changes and set/clear conditions in each operation mode.

The TSRmn register is read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSRmn register can be read with TSRmnL and 8-bit memory manipulation instructions. After a reset signal is generated, the value of the TSRmn register changes to “0000H”.

Figure 5-13 Table of timer status register mn (TSRmn)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVF

OVF	Counter overflow state for channel n
0	No overflow occurred.
1	Overflow occurred.
If the OVF bit is “1”, this flag is cleared when the next count does not overflow and the count value is captured (OVF=0).	

Remark: m: unit number (m=0, 1) n: channel number (n=0~3).

Table 5-4 OVF bit change and set/clear conditions in each operation mode

Timer operation mode	OVF bit	Set/clear conditions
• Capture mode • Capture & single count mode	clear	No overflow occurred at the capture.
	set	Overflow occurred at the capture.
• Interval timer mode • Event counter mode • Single count mode	clear	— (N/A)
	set	

Remark: Even if the counter overflows, the OVF bit does not change immediately, but changes on subsequent captures.

5.3.5 Timer Channel Enable Status Register m (TEm)

The TEm register is a register that indicates the enable or stop status of each channel timer operation.

Each of the TEm register corresponds to each of the timer channel start register m (TSm) and timer channel stop register m (TTm). If each bit of the TSm register is “1”, the corresponding bit of the TEm register is “1”. If each bit of the TTm register is “1”, the corresponding bit of the TTm register is cleared to “0”.

The TEm register is read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TEm register can be read with TEmL and 8-bit memory manipulation instructions. After a reset signal is generated, the value of the TEm register changes to “0000H”.

Figure 5-14 Table of timer channel enable status register m (TEm)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEm	0	0	0	0	TEH _{m3}	0	TEH _{m1}	0	0	0	0	0	TEm 3	TEm 2	TEm 1	TEm 0

TEH _{m3}	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 3 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEH _{m1}	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 1 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEmn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.
This bit displays whether operation of the lower 8-bit timer for TEm1 and TEm3 is enabled or stopped when channel 1 or 3 is in the 8-bit timer mode.	

Remark: m: unit number (m=0, 1) n: channel number (n=0~3).

5.3.6 Timer Channel Start Register m (TSm)

The TSm register is used to initialize the timer count register mn (TCRmn) and set the trigger for the start of counting for each channel. When the bits are set to 1, the corresponding bit in the timer channel enable status register m (TEm) is set to 1. Since TSmn, TSHm1, and TSHm3 are trigger bits, if they change to a running enabled state (TEmn, TEHm1, TEHm3 = 1), the TSmn, TSHm1, and TSHm3 bits are immediately cleared.

The TSm register is set using 16-bit memory manipulation instructions.

The lower 8 bits of the TSm register can be set using the TSmL and 8-bit memory manipulation instructions. After a reset signal is generated, the value of the TSm register becomes 0000H.

Figure 5-15 Table of timer channel start register m (TSm)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSm	0	0	0	0	TSHm3	0	TSHm1	0	0	0	0	0	TSm3	TSm2	TSm1	TSm0

TSHm3	High 8-bit timer run enable (start) trigger for channel 3 in 8-bit timer mode
0	No trigger.
1	Set TEHm3 to 1, entering the count enable state. When counting begins in the count enable state in the TCRm3 register, the interval timer mode is entered (refer to Table 5-5 for the operation from count enable state to the start of counting in the timer counter register TCRmn).

TSHm1	High 8-bit timer run enable (start) trigger for channel 1 in 8-bit timer mode
0	No trigger.
1	When counting begins in the count enable state in the TCRm1 register, the interval timer mode is entered (refer to Table 5-5 for the operation from count enable state to the start of counting in the timer counter register TCRmn).

TSmn	Channel n run enable (start) trigger
0	No trigger.
1	Set TEMn to 1, entering the count enable state. The counting behavior of the TCRmn register in the count enable state varies depending on the running mode (refer to Table 5-5 for the operation from count enable state to the start of counting in the timer counter register TCRmn). For Channels 1 and 3 in 8-bit timer mode, TSm1 and TSm3 are the run enable (start) triggers for the low 8-bit timers.

Cautions

1. Bits 15~12, 10, 8~4 must be set to "0".
2. When switching from a non-TImn pin input function to a TImn pin input function, the following waiting cycles are required after setting the timer mode register (TMRmn) and before setting the TSmn (TSHm1, TSHm3) bit to 1:
 - When the TImn pin noise filter is enabled (TNFENmn = 1): Wait for 4 clock cycles of the operating clock (f_{MCK}).
 - When the TImn pin noise filter is disabled (TNFENmn = 0): Wait for 2 clock cycles of the operating clock (f_{MCK}).

Remarks

1. The read value of the TSm register is always 0.
2. m: unit number (m=0, 1) n: channel number (n=0~3)

5.3.7 Timer Channel Stop Register m (TTm)

The TTm register is a trigger register to set the count stop of each channel.

If each bit is set to 1, the corresponding bit in the timer channel enable status register m (TEm) is cleared to 0. Since the TTmn bit, TTHm1 bit, and TTHm3 bit are trigger bits, the TTmn bit, TTHm1 bit, and TTHm3 bit are cleared immediately if the operation stop state is changed (TEmn, TEHm1, and TEHm3 = 0).

The TTm register is set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TTm register can be set by TTmL and by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the TTm register changes to “0000H”.

Figure 5-16 Table of timer channel stop register m (TTm)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTm	0	0	0	0	TTHm3	0	TTHm1	0	0	0	0	0	TTm3	TTm2	TTm1	TTm0

TTHm3	Trigger to stop operation of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	TEHm3 bit is cleared to 0 and the count operation is stopped.

TTHm1	Trigger to stop operation of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	TEHm1 bit is cleared to 0 and the count operation is stopped.

TTmn	Operation stop trigger of channel n
0	No trigger operation
1	TEmn bit clear to 0, to be count operation stop enable status. This bit is the trigger to stop operation of the lower 8-bit timer for TTm1 and TTm3 when channel 1 or 3 is in the 8-bit timer mode.

Caution: Bits 15~12, 10, 8~4 must be set to “0”.

Remark: 1. The TTm register always reads “0”.

2. m: unit number (m=0, 1) n: channel number (n=0~3).

5.3.8 Timer Input/Output Selection Register (TIOS0)

The TIOS0 register is used to make selections for the inputs and outputs of unit 0. The timer inputs for channel 0 and channel 1 and the timer output for channel 2 of unit 0 are selected. The TIOS0 register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the TIOS0 register changes to “00H”.

Figure 5-17 Table of timer input/output select register 0 (TIOS0)

Address: 0x40020474

After reset: 00H

R/W

Symbol	7	6	5	4	3	2	1	0
TIOS0	TIS07	TIS06	TIS05	TIS04	TOS03	TIS02	TIS01	TIS00

TIS07	TIS06	TIS05	Selection of timer input used for channel 0
0	0	0	Input signal for timer input pin (TI00)
Other			Settings are disabled

TIS04	Selection of timer input used by channel 0
0	Input signal selected via TIS07~TIS05
1	Event input signal for ELC

TOS03	Enable channel 2 timer output
0	Output enabled
1	Output disabled (output fixed to 0)

TIS02	TIS01	TIS00	Selection of timer input used for channel 1
0	0	0	Input signal of timer input pin (TI01)
0	0	1	Event input signal for EVENTC
0	1	0	Input signal of timer input pin (TI01)
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (FIL)
1	0	1	Subsystem clock (F _{SUB})
Other than the above			Disable setting.

Cautions:

1. The high-/low-level width of the selected timer inputs needs to be greater than or equal to $1/F_{MCK}+10ns$. Therefore, when F_{SUB} is selected as the F_{CLK} (CSS bit of the CKC register =1), the TIS02 bit cannot be set to “1”.
2. When selecting the event input signal for ELC via timer input select register 0 (TIOS0), F_{CLK} must be selected via timer clock select register 0 (TPS0).

5.3.9 Timer Output Enable Register m (TOEm)

The TOEm register is a register that sets to enable or disable the timer output of each channel.

Then channel n for which timer output has been enabled becomes unable to rewrite the value of the TOMn bit of timer output register m (TOM) described later by software, and the value reflecting the setting of the timer output function through the count operation is output from the timer output pin (TOMn).

The TOEm register is set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOEm register can be set by TOEmL and by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the TOEm register changes to “0000H”.

Figure 5-18 Table of timer output enable register m (TOEm)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEm	0	0	0	0	0	0	0	0	0	0	0	0	TOE m3	TOE m2	TOE m1	TOE m0

TOEmn	Enable/disbale timer output of channel n
0	Disable timer output. The operation of the timer is not reflected to the TOMn bit, fixed output. The TOMn bit can be written and the level set by the TOMn bit is output from the TOMn pin.
1	Enable timer output. The operation of the timer is reflected to the TOMn bit, producing an output waveform. The write of the TOMn bit is ignored.

Caution: Bits 15~4 must be set to “0”.

Remark: m: unit number (m=0, 1) n: channel number (n=0~3).

5.3.10 Timer Output Register m (TOM)

The TOM register is a buffer register for each channel timer output.

The bit value of this register is output from the output pin (TOMn) of each channel timer.

The TOMn bit of this register can be rewritten by software only when timer output is disabled (TOEmn=0). When enabling the timer output (TOEmn=1), rewrite operations via software are ignored and its value is changed only by the operation of the timer.

To use the TOMn pin as port functions, the corresponding TOMn bit must be set to “0”.

The TOM register is set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOM register can be set by TOML and by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the TOM register changes to “0000H”.

Figure 5-19 Table of timer output register m (TOM)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM	0	0	0	0	0	0	0	0	0	0	0	0	TOM3	TOM2	TOM1	TOM0

TOMn	Timer output of channel n
0	The output value of the timer is “0”.
1	The output value of the timer is “1”.

Caution: Bits 15~4 must be set to “0”.

Remark: m: unit number (m=0, 1) n: channel number (n=0~3).

5.3.11 Timer Output Level Register m (TOLm)

The TOLm register is a register that controls the output level of each channel timer.

When timer output (TOEmn=1) is enabled and the multi-channel linkage operation function (TOMmn=1) is used, the set and reset timing of the timer output signal reflects the inverse setting of each channel n performed by this register. In the master channel output mode (TOMmn=0), this register setting is invalid.

The TOLm register is set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOLm register can be set by TOLmL and by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the TOLm register changes to “0000H”.

Figure 5-20 Table of timer output level register m (TOLm)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOLm	0	0	0	0	0	0	0	0	0	0	0	0	TOLm3	TOLm2	TOLm1	0

TOLmn	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Negative output (active-low)

Caution: Bits 15~4 and bit0 must be set to “0”.

Remarks: 1. If the value of this register is rewritten while the timer is operating, the timer output logic is inverted at the next time the timer output signal changes, rather than immediately after the rewrite.

2. m: unit number (m=0, 1) n: channel number (n=0~3).

5.3.12 Timer Output Mode Register m (TOMm)

The TOMm register is a register that controls the output mode of each channel timer. When used as an independent channel operation function, the corresponding bit of the using channel should be set to “0”.

When used as a multi-channel linkage operation function (PWM output, single trigger pulse output and multiple PWM output), the corresponding bit of the master channel is “0” and the corresponding bit of the slave channel is “1”.

When the timer output (TOEmn=1) is enabled, the setting of each channel n is reflected in this register during the setting and resetting timing of the timer output signal.

The TOMm register is set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOMm register can be set by TOMmL and by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the TOMm register changes to “0000H”.

Figure 5-21 Table of timer output mode register m(TOMm)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOMm	0	0	0	0	0	0	0	0	0	0	0	0	TOM _{m3}	TOM _{m2}	TOM _{m1}	0

TOMmn	Control of channel n timer output mode
0	Master channel output mode (alternate output via timer interrupt request signal (INTTMmn))
1	Slave channel output mode (output is set via timer interrupt request signal (INTTMmn) of master channel and output is reset via timer interrupt request signal (INTTMmp) of slave channel)

Caution: Bits 15~4 and bit0 must be set to 0.

Remark: m: unit number (m=0, 1) n: channel number

n=0~3 (in master channel: n=0, 2)

p: slave channel number

n=0: p=1, 2, 3

n=2: p=3

(For details on the relationship between the master channel and the slave channel, refer to “5.4.1 Basic Rules for Multi-Channel Linkage Operation Function”.)

5.3.13 Noise Filter Enable Register 1 (NFEN1)

The NFEN1 register sets whether the noise filter is used for the input signals of the timer input pins of each channel of Unit 0. For pins that require noise removal, the corresponding bit must be set to “1” to make the noise filter effective. When the noise filter is enabled, after synchronization with the operating clock (F_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected. When the noise filter is disabled, the input signal is only synchronized with the operating clock (F_{MCK}) for the target channel^{Note}.

The NFEN1 register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the NFEN1 register changes to “00H”.

Note: For details, refer to “5.5.1(2) When the valid edge of TIMn pin input signal (CCSmn=1) is selected”, “5.5.2 Start Timing of Counter”, and “5.7 Control of Timer Input (TIMn)”.

Figure 5-22 Table of noise filter enable register 1(NFEN1)

Address: 0x40040471

Symbol	7	6	5	4	3	2	1	0
NFEN1	0	0	0	0	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN03	Usage of input signal noise filter on TI03 pin
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Usage of input signal noise filter on TI02 pin
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Usage of input signal noise filter on TI01 pin
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Usage of input signal noise filter on TI00 pin
0	Noise filter OFF
1	Noise filter ON

Remark: For the configuration of timer input/output pins for channels 0 to 3, Refer to Chapter 2 Pin Functions.

5.3.14 Noise Filter Enable Register 2 (NFEN2)

The NFEN2 register sets whether the noise filter is used for the input signals of the timer input pins of each channel of Unit 0. For pins that require noise removal, the corresponding bit must be set to “1” to make the noise filter effective. When the noise filter is enabled, after synchronization with the operating clock (F_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected. When the noise filter is disabled, the input signal is only synchronized with the operating clock (F_{MCK}) for the target channel^{Note}.

The NFEN2 register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the NFEN2 register changes to “00H”.

Note: For details, refer to “5.5.1(2) When the valid edge of TIMn pin input signal (CCSmn=1) is selected”, “5.5.2 Start Timing of Counter”, and “5.7 Control of Timer Input (TIMn)”.

Figure 5-23 Table of noise filter enable register 2 (NFEN2)

Address: 0x40040472								
Symbol	7	6	5	4	3	2	1	0
NFEN2	0	0	0	0	TNFEN13	TNFEN12	TNFEN11	TNFEN10

TNFEN13	Usage of input signal noise filter on TI13 pin
0	Noise filter OFF
1	Noise filter ON

TNFEN12	Usage of input signal noise filter on TI12 pin
0	Noise filter OFF
1	Noise filter ON

TNFEN11	Usage of input signal noise filter on TI11 pin
0	Noise filter OFF
1	Noise filter ON

TNFEN10	Usage of input signal noise filter on TI10 pin
0	Noise filter OFF
1	Noise filter ON

Remark: For the configuration of timer input/output pins for channels 0 to 3, Refer to Chapter 2 Pin Functions.

5.3.15 Registers for Controlling Port Functions of Timer Input/Output Pins

When using the universal timer unit, the timer input/output pins can be configured to any port except for P00. For detailed information, please refer to Chapter 2: Pin Functions.

When remapping the timer output pin to a specific port, you must set the corresponding bit in the port mode control register (PMCxx) and the bit of the port mode register (PMxx) to 0. Additionally, the port alternate function configuration register (PxxCFG) should be configured. In this case, the port register (Pxx) can be either 0 or 1.

Example: Configure P21 as TO00 for timer output:

Set the bit of the port mode control register 2 (PMC21) to 0.

Set the bit of the port mode register 2 (PM21) to 0.

Set the port alterante function configuration register P21CFG to 0x0e.

When using the alternated port of the timer input pin for timer input, you must set the corresponding bit in the port mode register (PMxx) to 1 and the bit in the port mode control register (PMCxx) to 0. Also, configure the port alternate function configuration register (PxxCFG). In this case, the port register (Pxx) can be either 0 or 1.

Example: Configure P20/TI00 for timer input:

Set the bit of the port mode control register 2 (PMC20) to 0.

Set the bit of the port mode register 2 (PM20) to 1.

Set the port alternate function configuration register P20CFG to 0x06.

5.4 Basic Rules of Universal Timer Unit

5.4.1 Basic Rules of Multi-Channel Linkage Operation Function

The multi-channel linkage function is a function that combines a master channel (a reference timer that counts cycles) and a slave channel (a timer that operates in compliance with the master channel), and several rules need to be observed when using it.

The basic rules of the multi-channel linkage operation function are shown below.

- (1) Only the even-number channel (channel 0, channel 2) can be set as a master channel.
- (2) Any channel other than channel 0 can be set as a slave channel.
- (3) Only the lower channel of the master channel can be set as a slave channel.

For example, when setting channel 0 as the master channel, it is possible to set the channels starting from channel 1 (channels 1 to 3) as slave channels.

- (4) Multiple slave channels can be set for 1 master channel.
- (5) When multiple master channels are used, slave channels that span the master channel cannot be set.

For example, when setting channel 0 and channel 2 as the master channel, channel 1 can be set as the slave channel of master channel 0, but channel 3 cannot be set as the slave channel of master channel 0.

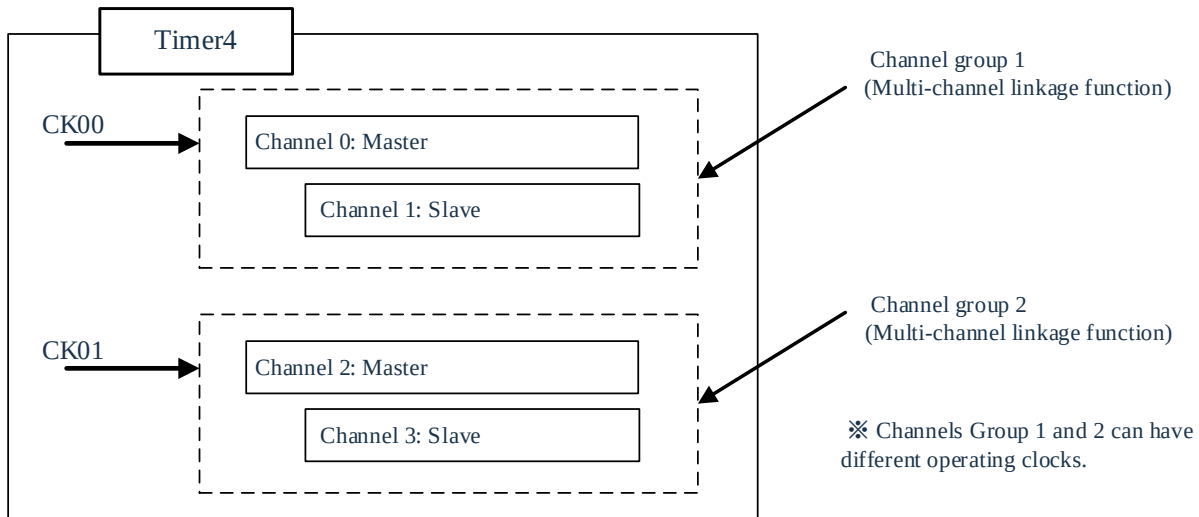
- (6) The slave channels linked to the master channel need to be set to the same operating clock. The CKSmn0 bit and CKSmn1 bit (bit15 and bit14 of Timer Mode Register mn (TMRmn)) of the slave channel linked to the master channel need to be the same setting value.
- (7) The master channel can pass the INTTMmn (interrupt), start software trigger and count clock to the lower channel.
- (8) The slave channel can use the master channel's INTTMmn (interrupt), start software trigger, and count clocks as source clocks, but cannot pass its own INTTMmn (interrupt), start software trigger, and count clocks to the lower channel.
- (9) The master channel cannot use the INTTMmn (interrupt), start software trigger and count clocks of other high master channels as source clocks.
- (10) In order to start the channels to be linked at the same time, the channel start trigger bit (TSmn) of the linked channel needs to be set at the same time.
- (11) Only all linked channels or the master channel can use the setting of the TSmn bit in the counting operation. It is not possible to use the setting of the TSmn bit of the slave channel only.
- (12) In order to stop the linked channels at the same time, the channel stop trigger bit (TTmn) of the linked channel needs to be set at the same time.
- (13) In linked operation, CKm2/CKm3 cannot be selected because the master and slave channels need the same operating clock.
- (14) The timer mode register m0 (TMRm0) has no master bit and is fixed to 0. However, since channel 0 is the highest bit channel, it can be used as the master channel during linkage operation.

The basic rules of the multi-channel linkage operation function are the rules applicable to the group of channels (a collection of master and slave channels that form a multi-channel linkage operation function).

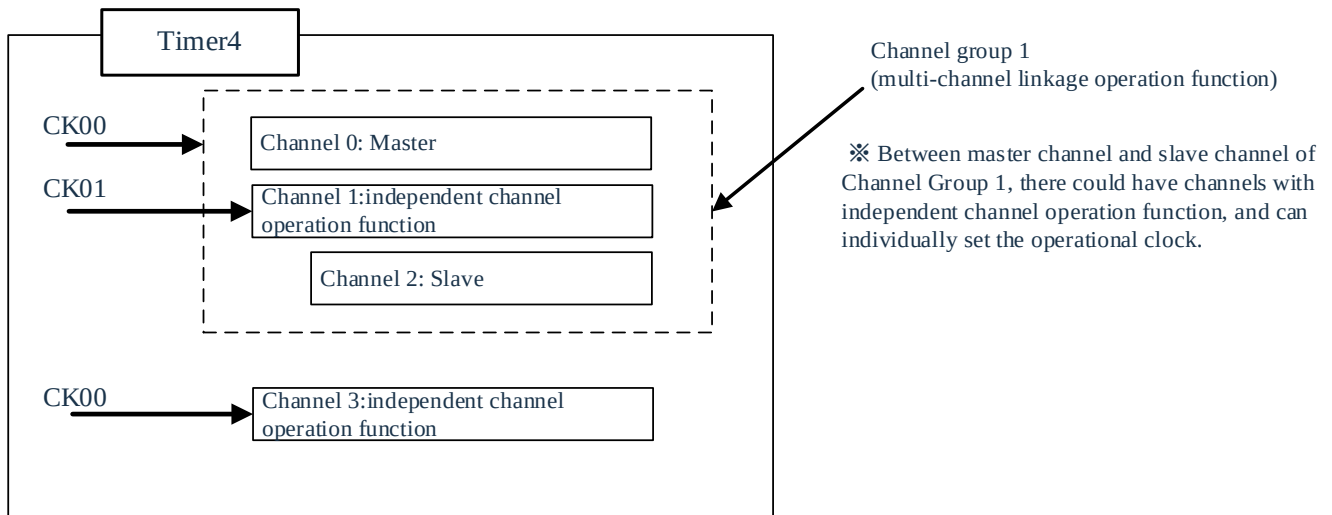
If you set 2 or more channel groups that are not linked to each other, the above basic rules do not apply to the channel groups.

Remark: m: unit number (m=0, 1) n: channel number (n=0~3).

Example 1



Example 2



5.4.2 Basic Rules of 8-Bit Timer Operation Function (Channels 1 and 3 of Unit 0 Only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.

This function can only be used for channels 1 and 3, and there are several rules for using it.

The basic rules for this function are as follows:

- (1) The 8-bit timer operation function applies only to channels 1 and 3.
- (2) When using 8-bit timers, set the SPLIT bit of timer mode register mn (TMRmn) to 1.
- (3) The higher 8 bits can be operated as the interval timer function.
- (4) At the start of operation, the higher 8 bits output INTTMm1H (an interrupt) (which is the same operation performed when MDmn0 is set to 1).
- (5) The operation clock of the higher 8 bits is selected according to the CKSmn1 and CKSmn0 bits of the lower-bit TMRmn register.
- (6) For the higher 8 bits, the TSHm1/TSHm3 bit is manipulated to start channel operation and the TTHm1/TTHm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEHm1/TEHm3 bit.
- (7) The lower 8 bits operate according to the TMRmn register settings. The following three functions support operation of the lower 8 bits:
 - Interval timer function
 - External event counter function
 - Delay count function
- (8) For the lower 8 bits, the TSm1/TSm3 bit is manipulated to start channel operation and the TTm1/TTm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEM1/TEm3 bit.
- (9) During 16-bit operation, manipulating the TSHm1, TSHm3, TTHm1, and TTHm3 bits is invalid. The TSm1, TSm3, TTm1, and TTm3 bits are manipulated to operate channels 1 and 3. The TEHm3 and TEHm1 bits are not changed.
- (10) For the 8-bit timer function, the linkage operation functions (single pulse, PWM, and multiple PWM) cannot be used.

Remark: m: unit number (m=0) n: channel number (n=1, 3)

5.5 Operation of Counter

5.5.1 Count Clock (f_{TCLK})

The count clock of the general-purpose timer unit (F_{TCLK}) can be selected by the CCSmn bit of the timer mode register mn (TMRmn) for any of the following clocks:

- The CKSmn0 bit and CKSmn1 bit specified operation clock (F_{MCK})
- The active edge of the TImn pin input signal

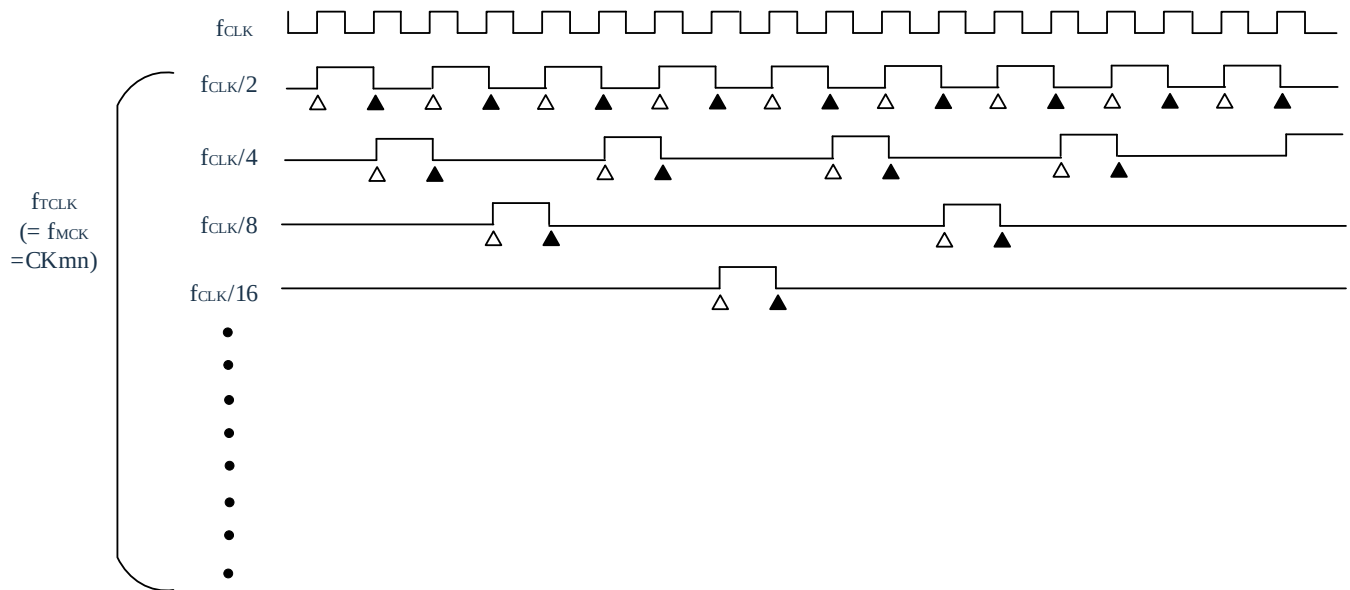
The universal timer unit is designed to operate synchronously with F_{CLK} , so the timing of the count clock (F_{TCLK}) is as follows.

(1) When operation clock (F_{MCK}) specified by the CKSmn0 and CKSmn1 bits is selected (CCSmn = 0)

According to the setting of timer clock selection register m (TPSm), the counting clock (F_{TCLK}) is $F_{CLK} \sim F_{CLK}/2^{15}$. However, when the frequency division of F_{CLK} is selected, the clock selected by TPSm register is a signal that has only 1 F_{CLK} cycle of high level from the rising edge. When F_{CLK} is selected, it is fixed to high level.

In order to obtain synchronization with F_{CLK} , timer count register mn (TCRmn) delays the counting by one F_{CLK} clock from the rising edge of the counting clock, which is called “counting at the rising edge of the counting clock” for convenience.

Figure 5-24 Timing of f_{CLK} (f_{TCLK}) and count clock (when CCSmn=0)



Remarks: 1. Δ : Rising edge of the count clock

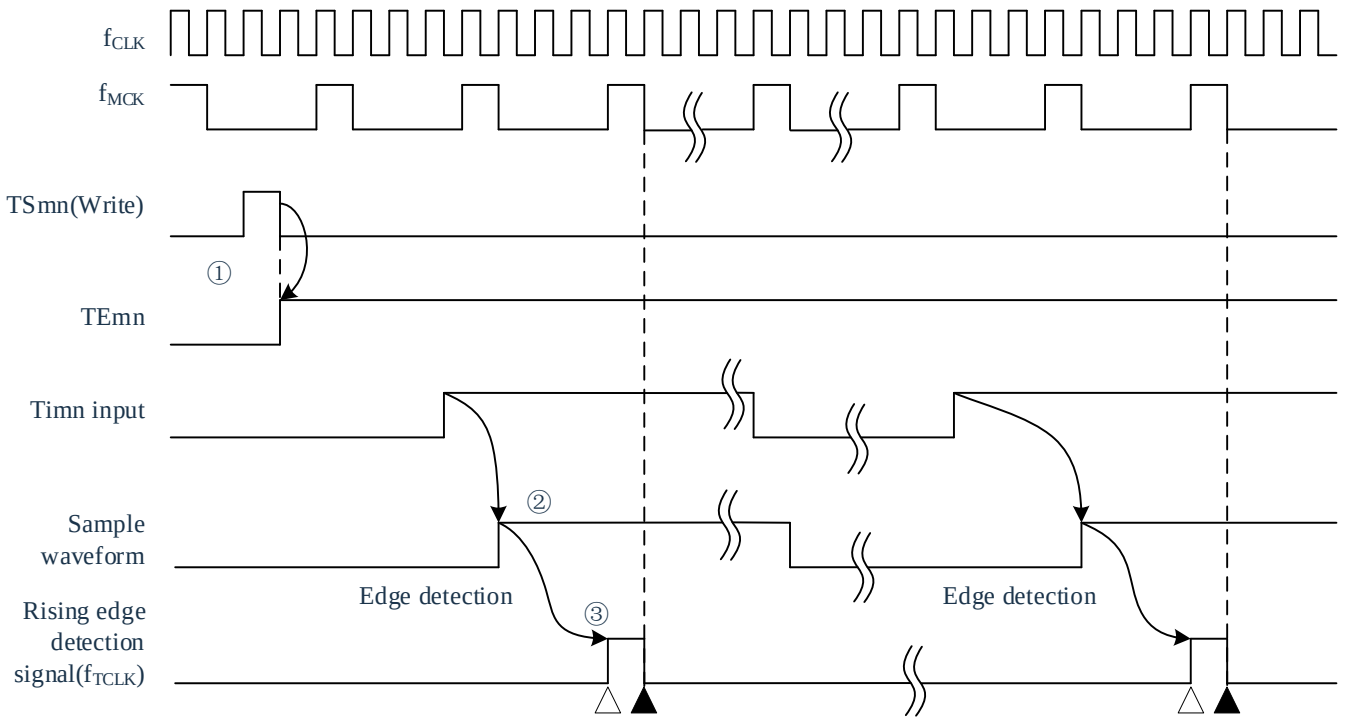
$\blacktriangle$: Synchronization, increment/decrement of counter

2. f_{CLK} : CPU/peripheral hardware clock

(2) When valid edge of input signal via the TIMn pin is selected (CCSmn = 1)

The count clock (F_{TCLK}) is a signal that detects an active edge of the TIMn pin input signal and is synchronized with the next F_{MCK} rising edge. In fact, this is a signal delayed by 1~2 F_{MCK} clocks compared to the input signal of the TIMn pin (delay 3~4 F_{MCK} clocks when using noise filters). In order to obtain synchronization with F_{CLK} , the timer count register mn (TCRmn) delays the count by one F_{CLK} time from the rising edge of the count clock, which is referred to as “counting at the effective edge of the TIMn pin input signal” for convenience.

Figure 5-25 Timing of the counting clock (f_{TCLK}) (CCSmn=1, without noise filter)



- ① Setting TSmn bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TIMn pin.
- ② The rise of input signal via the TIMn pin is sampled by F_{MCK} .
- ③ The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

Remarks: 1. Δ : Rising edge of the count clock

$\blacktriangle$: Synchronization, increment/decrement of counter

2. f_{CLK} : CPU/peripheral hardware clock

f_{MCK} : Operation clock of channel n

3. The same waveforms are used for the measurement of the input pulse interval, the high and low measurement of the input signal, the delay counter and the TIMn input for the single trigger pulse output function.

5.5.2 Start Timing of Counter

The timer count register mn (TCRmn) enters the operation enable state by setting TSmn bit of the timer channel start register m (TSM).

Execution from the counting enable state to the start of the timer count register mn (TCRmn) is shown in Table 5-5.

Table 5-5 Operation from the counting enable state to the start of the timer count register mn (TCRmn)

Timer operation mode	Operation after setting TSmn bit to "1"
• Interval timer mode	No operation is performed from the detection of the start trigger (TSmn=1) until the count clock is generated. The value of the TDRmn register is loaded into the TCRmn register by the first count clock and decremented by subsequent count clocks (refer to "5.5.3(1) Operation of the interval timer mode").
• Event counter mode	The value of the TDRmn register is loaded into the TCRmn register by writing a "1" to the TSmn bit. If the input edge of TIMn is detected, the count is decremented by the subsequent count clocks. (Refer to "5.5.3(2) Operation of the event counter mode").
• Capture mode	No operation is performed from the time the start trigger is detected until the count clock is generated. The "0000H" is loaded into the TCRmn register by the first count clock, and incremental counting is performed by the subsequent count clocks (refer to "5.5.3(3) Operation of the capture mode (input pulse interval measurement)").
• Single count mode	By writing "1" to the TSmn bit while the timer is stopped (TEmn=0), it enters the wait state for the start of the trigger. No operation is performed from the time the start trigger is detected until the count clock is generated. The value of the TDRmn register is loaded into the TCRmn register by the first count clock, and decremental counting by subsequent count clocks (refer to "5.5.3(4) Operation of the single count mode").
• Capture & single count mode	By writing "1" to the TSmn bit while the timer is stopped (TEmn=0), it enters the wait state for the start of the trigger. No operation is performed from the time the start trigger is detected until the count clock is generated. The "0000H" is loaded into the TCRmn register by the first count clock, and incremental counting is performed by the subsequent count clocks (refer to "5.5.3(5) Operation of capture & single count mode (measurement of high-level width)").

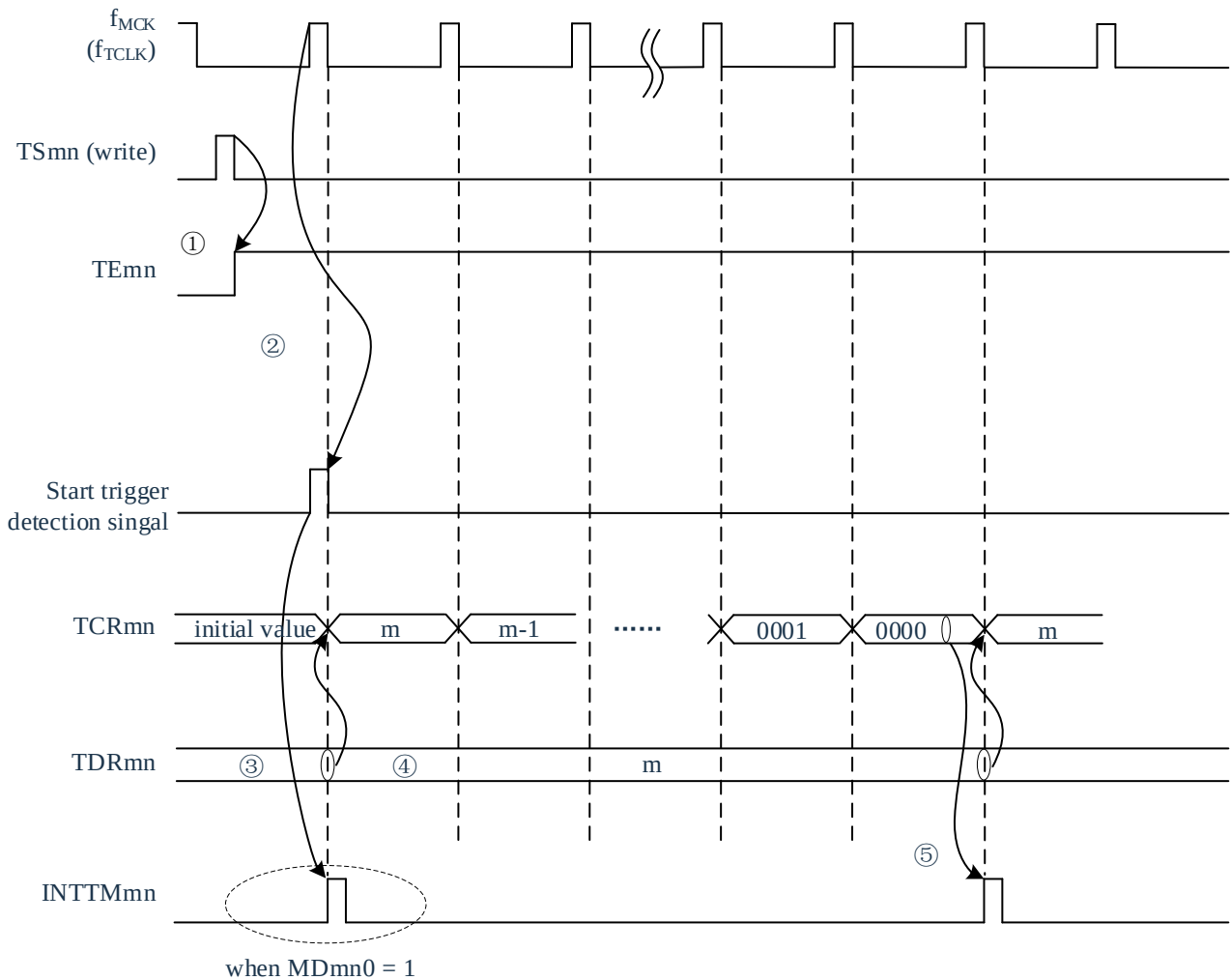
5.5.3 Operation of Counter

The following describes the counter operation for each mode.

(1) Operation of interval timer mode

- ① The operation enable state is entered by writing “1” to the TSmn bit (TEmn=1). The timer count register mn (TCRmn) remains at its initial value until a count clock is generated.
- ② A start trigger signal is generated by enabling the 1st count clock (F_{MCK}) after the operation.
- ③ When MDmn0 bit is “1”, INTTMmn is generated by the start trigger signal.
- ④ The value of timer data register mn (TDRmn) is loaded into the TCRmn register by enabling the 1st count clock after the operation, and counting starts in interval timer mode.
- ⑤ If the TCRmn register decrements to “0000H”, INTTMmn is generated by the next count clock (F_{MCK}) and continues counting after loading the value of timer data register mn (TDRmn) into the TCRmn register.

Figure 5-26 Operation timing (interval timer mode)



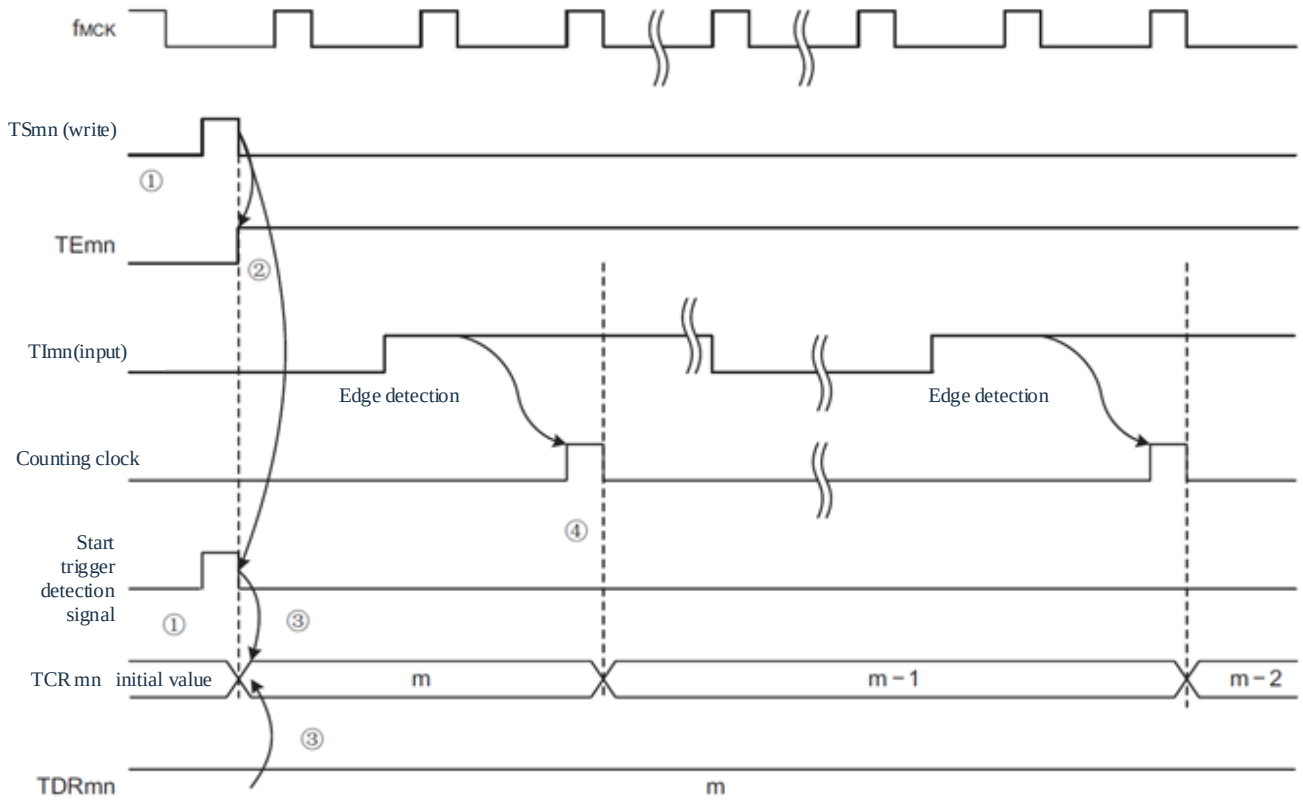
Caution: In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

Remark: F_{MCK} , the start trigger detection signal, and INTTMmn become active between one clock in synchronization with F_{CLK} .

(2) Operation of event counter mode

- ① The timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn=0).
- ② The operation enable state is enabled by writing “1” to the TSmn bit (TEmn=1).
- ③ The value of timer data register mn (TDRmn) is loaded into the TCRmn register while both the TSmn and TEmn bits are changed to “1” and counting begins.
- ④ Thereafter, the value of the TCRmn register is counted decreasingly by the count clock at the active edge of the TImn input.

Figure 5-27 Operation timing (event counter mode)

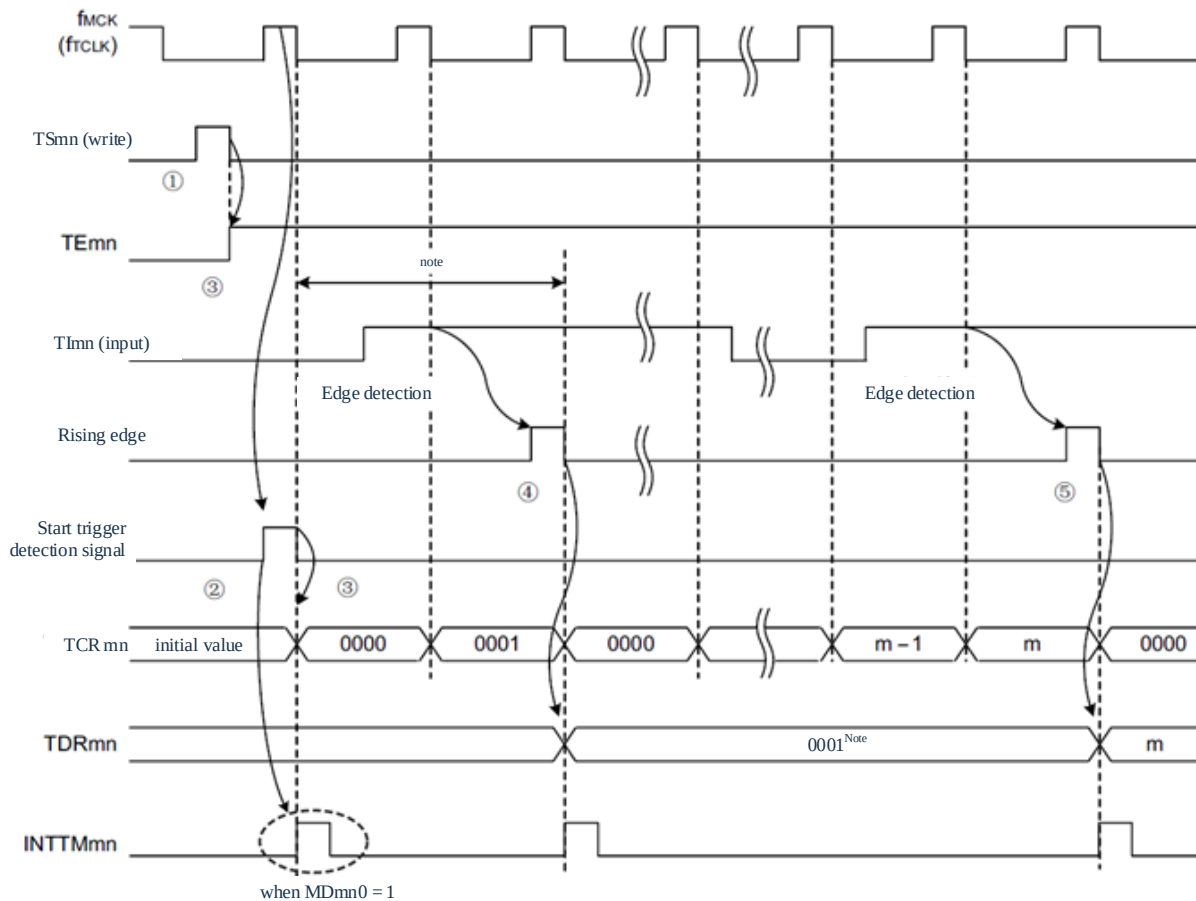


Remark: This is a timing without the noise filter. If the noise filter is used, the edge detection is delayed by 2 more F_{MCK} cycles (3~4 cycles in total) from the TImn input. The 1 cycle error is because the TImn input is not synchronized with the count clock (F_{MCK}).

(3) Operation of capture mode (interval measurement of input pulses)

- ① The operation enable state is entered by writing “1” to the TS_{mn} bit (TE_{mn}=1).
- ② The timer count register mn (TCR_{mn}) remains at its initial value until a count clock is generated.
- ③ A start trigger signal is generated by enabling the 1st count clock (F_{MCK}) after the operation. Then, the “0000H” is loaded into the TCR_{mn} register and counting starts in capture mode (INTT_{mn} is generated by the start trigger signal when MD_{mn}0 bit is “1”).
- ④ If an active edge of TI_{mn} input is detected, the value of TCR_{mn} register is captured to TDR_{mn} register and INTT_{mn} interrupt is generated. The capture value is meaningless at this point. The TCR_{mn} register continues counting from the “0000H”.
- ⑤ If an active edge of the next TI_{mn} input is detected, the value of the TCR_{mn} register is captured to the TDR_{mn} register and the INTT_{mn} interrupt is generated.

Figure 5-28 Operation timing (capture mode: interval measurement of input pulses)



Note: When the clock is input to TI_{mn} (with trigger) before the start, the count is started by detecting the trigger even if no edge is detected, so the capture value at the 1st capture (④) is not a pulse interval (in this example, 0001: 2 clock intervals) and must be ignored.

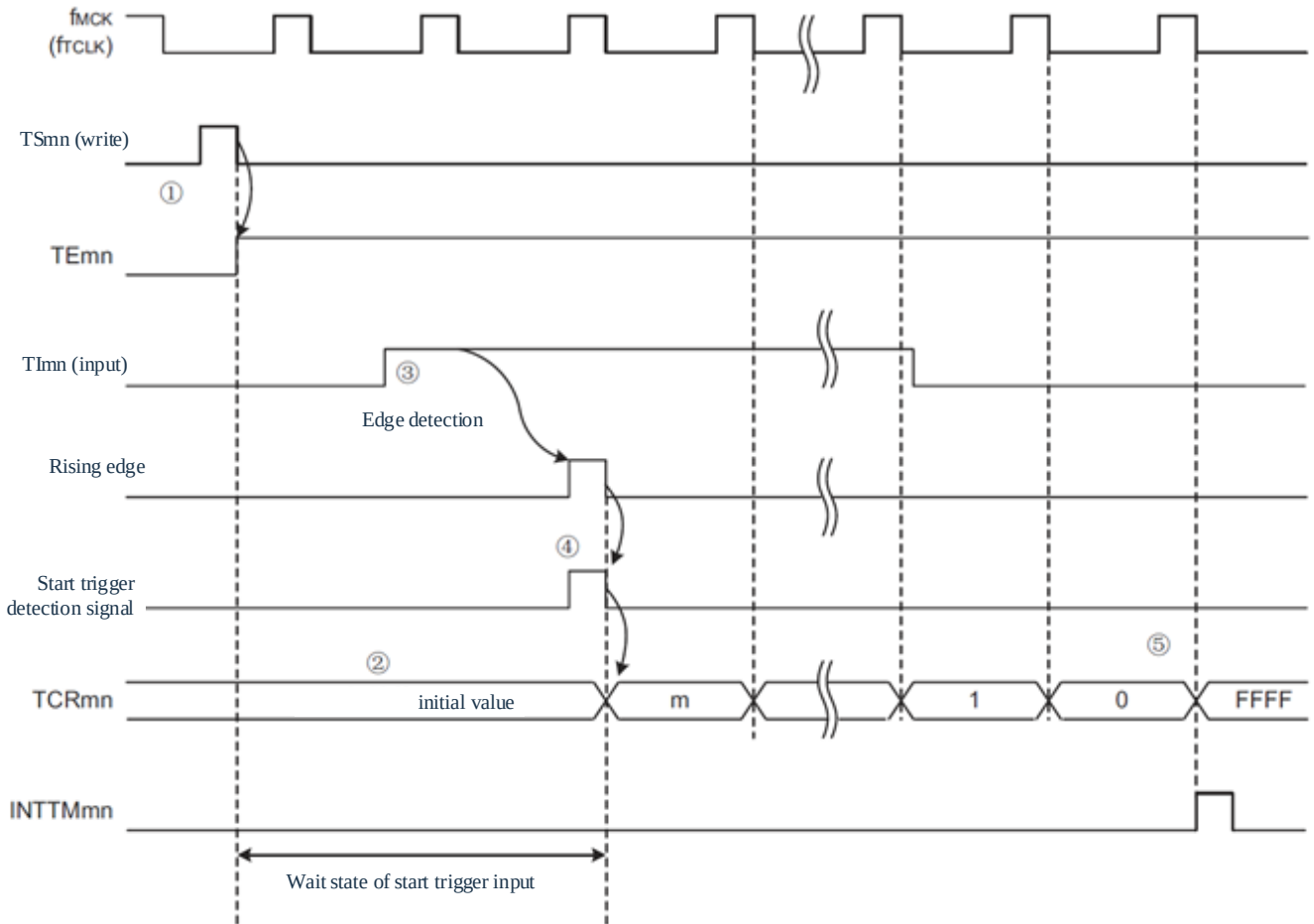
Caution: Because the 1st count clock cycle runs after the TS_{mn} bit is written and delays the start of counting before generating the count clock, an error of up to 1 clock cycle is generated. Also, if you need information about the start of the count timing, set MD_{mn}0 to “1” so that an interrupt can be generated at the start of the count.

Remark: This is a timing without the noise filter. If the noise filter is used, the edge detection is delayed by 2 more F_{MCK} cycles (3~4 cycles in total) from the TI_{mn} input. The 1 cycle error is because the TI_{mn} input is not synchronized with the count clock (F_{MCK}).

(4) Operation of single count mode

- ① The operation enable state is entered by writing “1” to the TSmn bit (TEmn=1).
- ② The timer count register mn (TCRmn) remains the initial value until a start trigger signal is generated.
- ③ Detects the rising edge of the TIMn input.
- ④ The value (m) of the TDRmn register is loaded into the TCRmn register after a start trigger signal is generated, and counting begins.
- ⑤ When the TCRmn register decrements to “0000H”, the INTTMmn interrupt is generated and the value of TCRmn register changes to “FFFFH” and stop counting.

Figure 5-29 Operation timing (single count mode)

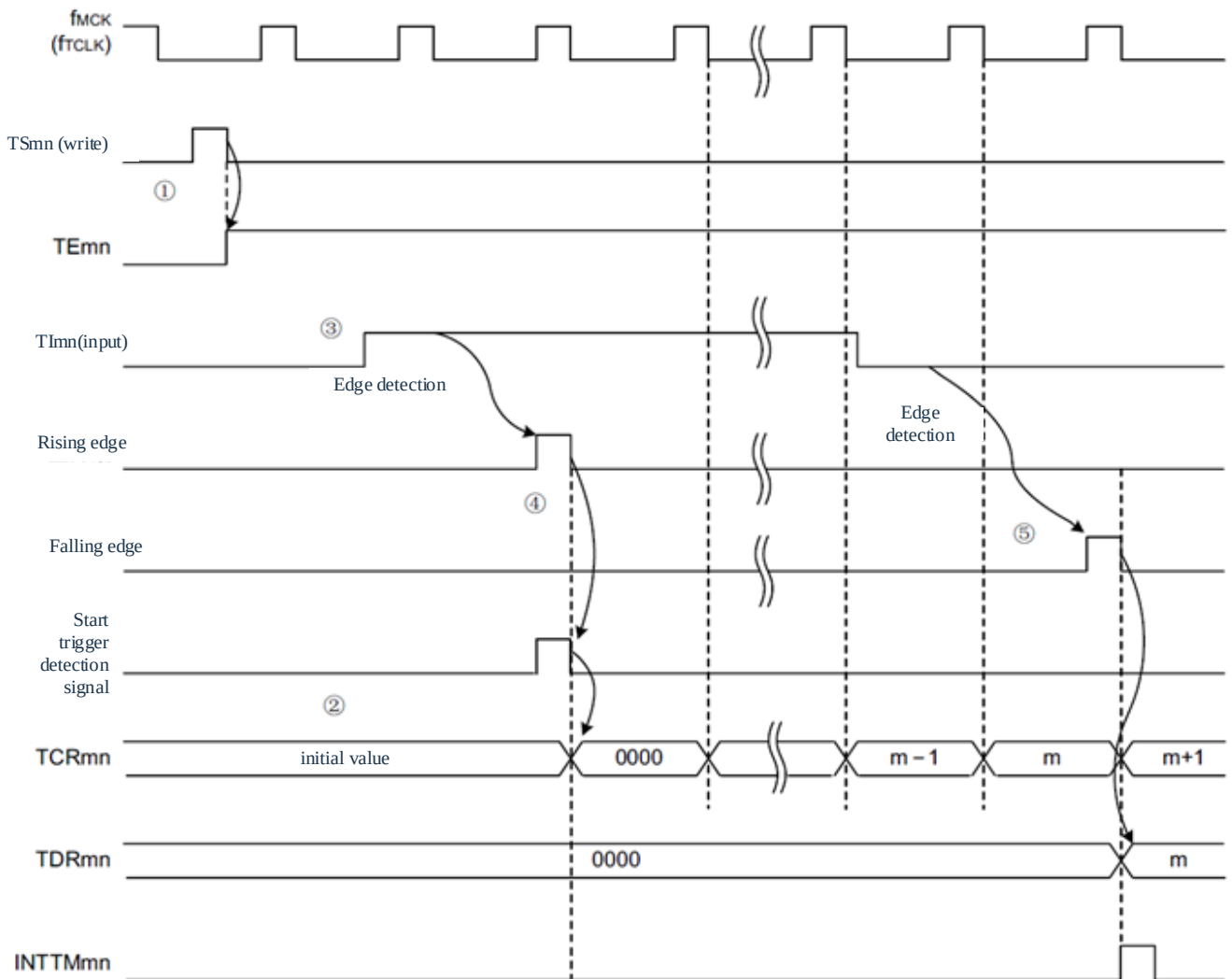


Remark: This is a timing without the noise filter. If the noise filter is used, the edge detection is delayed by 2 more F_{MCK} cycles (3~4 cycles in total) from the TIMn input. The 1 cycle error is because the TIMn input is not synchronized with the count clock (F_{MCK}).

(5) Operation of capture & single count mode (measurement of high-level width)

- ① The operation enable state is entered by writing “1” to the TSmn bit of the timer channel start register m (TSm)(TEmn=1).
- ② The timer count register mn (TCRmn) remains the initial value until a start trigger signal is generated.
- ③ Detects the rising edge of the TIMn input.
- ④ After the start trigger signal is generated, “0000H” is loaded into the TCRmn register and counting starts.
- ⑤ If the falling edge of TIMn input is detected, the value of the TCRmn register is captured to the TDRmn register and an INTTMmn interrupt is generated.

Figure 5-30 Operation timing (capture & single count mode: measurement of high-level width)

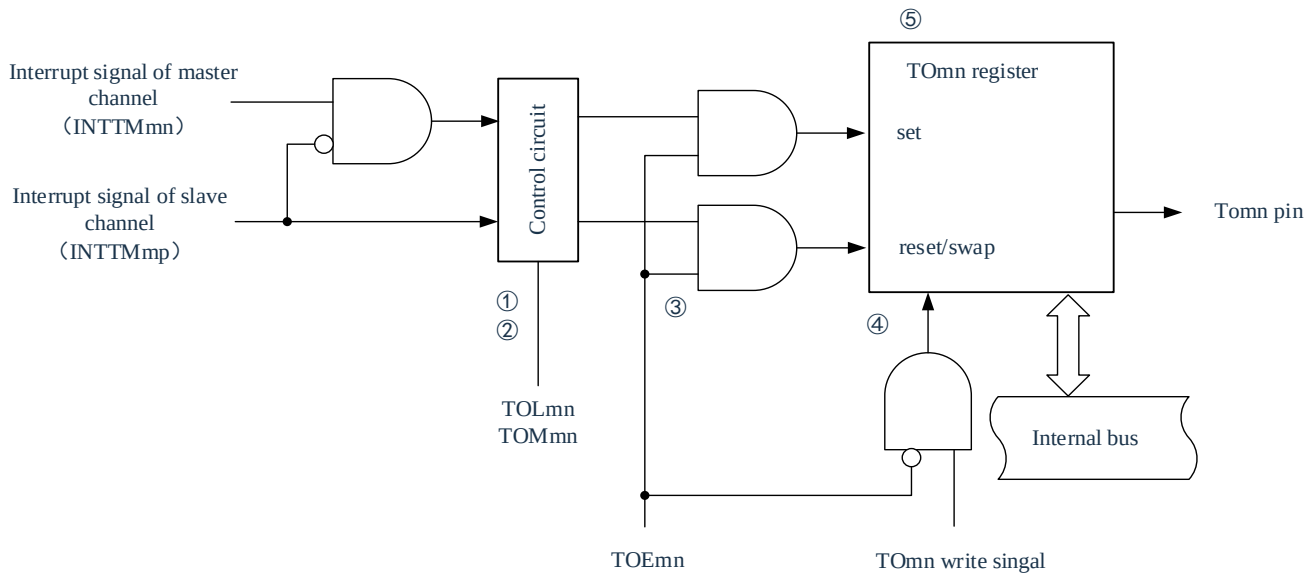


Remark: This is a timing without the noise filter. If the noise filter is used, the edge detection is delayed by 2 more F_{MCK} cycles (3~4 cycles in total) from the TIMn input. The 1 cycle error is because the TIMn input is not synchronized with the count clock (F_{MCK}).

5.6 Control of Channel Output (TOMn Pin)

5.6.1 Configuration of TOMn Pin Output Circuit

Figure 5-31 Configuration of output circuit



The following explains the output circuit of the TOMn pin.

- ① When the TOMmn bit is “0” (master channel output mode), the setting value of timer output level register m (TOLm) is ignored and only INTTMmp (slave channel timer interrupt) is passed to timer output register m (TOM).
- ② When the TOMmn bit is “1” (slave channel output mode), INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are passed to the TOM register.
At this time, the TOLm register becomes valid and the signals are controlled as follows:
When TOLmn = 0: Positive logic output (INTTMmn → set, INTTMmp → reset)
When TOLmn = 1: Negative logic output (INTTMmn → reset, INTTMmp → set)
When INTTMmn and INTTMmp are simultaneously generated, (0% output of PWM), INTTMmp (reset signal) takes priority, and INTTMmn (set signal) is masked.
- ③ In the state of enabling timer output (TOEmn=1), INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are passed to TOM register. Writing to the TOM register (TOMn write signal) is invalid.
When the TOEmn bit is “1”, the output of the TOMn pin is not changed except for the interrupt signal.
To initialize the output level of the TOMn pin, you need to write a value to the TOM register after setting it to disable the timer output (TOEmn=0).
- ④ Writing to the TOMn bit for the object channel (TOMn write signal) is valid when the timer output is disabled (TOEmn=0). When the timer output is disabled (TOEmn=0), INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are not passed to the TOM register.
- ⑤ The TOM register can be read at any time and the output level of the TOMn pin can be confirmed.

Remark: m: unit number (m= 0, 1) n: channel number n=0~3 (master channel: n=0, 2)

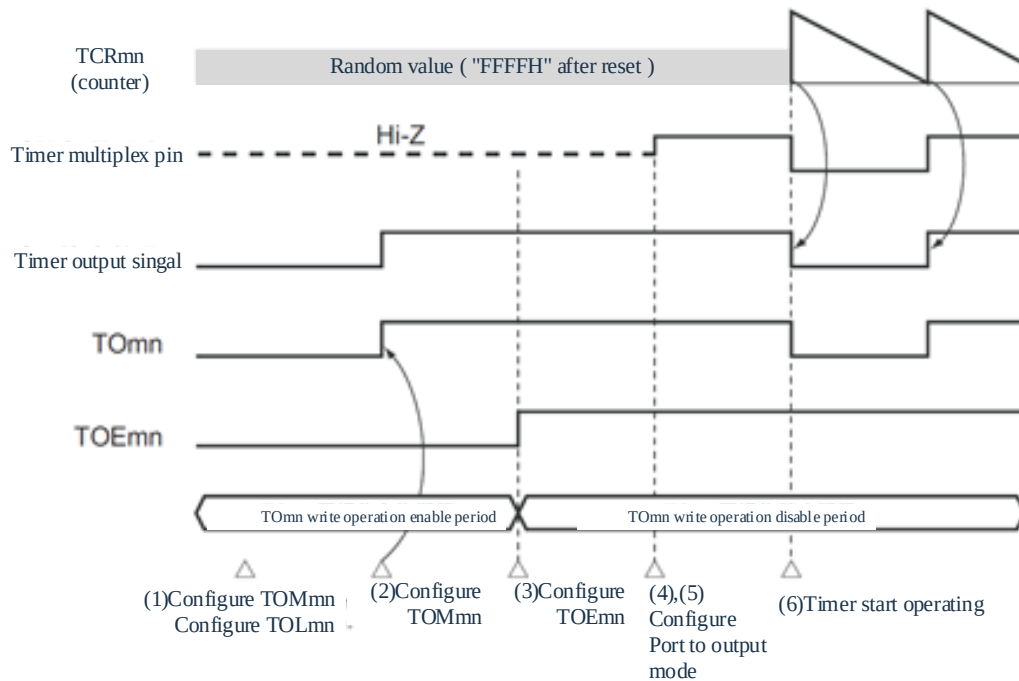
p: Slave channel number n=0: p=1, 2, 3

n=2: p=3

5.6.2 Output Settings for TOMn Pins

The following figure shows the procedure and status transition of the TOMn output pin from initial setting to timer operation start.

Figure 5-32 State change from setting timer output to start of operation



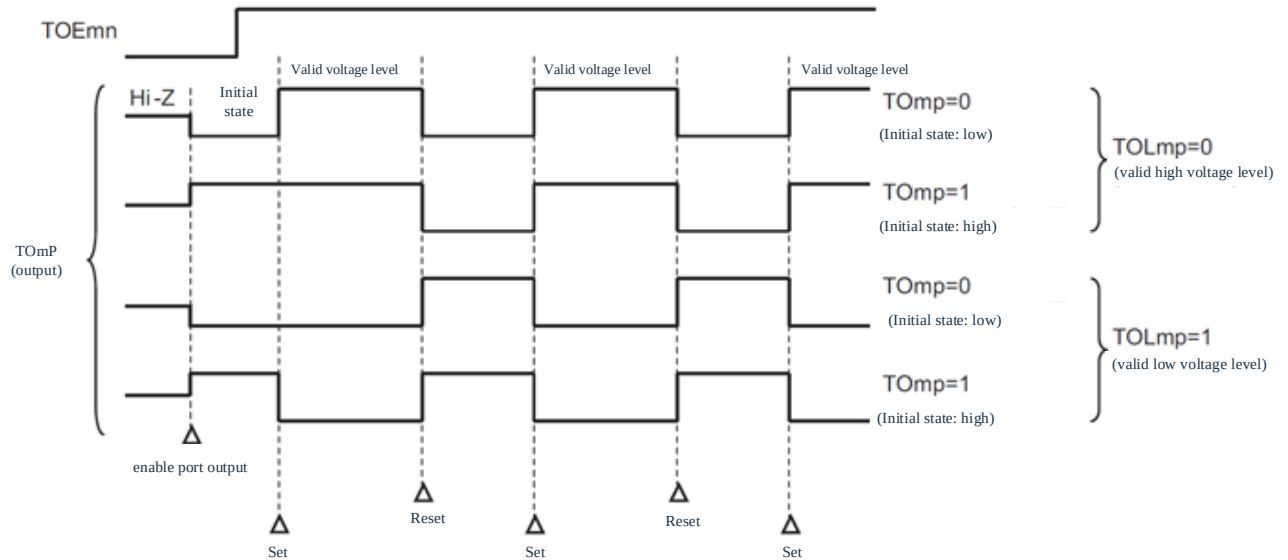
- ① Set the operation mode of the timer output.
 - TOMmn bit (0: master channel output mode, 1: slave channel output mode)
 - TOLmn bit (0: positive logic output, 1: negative logic output)
- ② The timer output signal is set to the initial state by setting the timer output register m (TOM).
- ③ Writing "1" to TOEmn bit enables timer output (writing to TOM register is disabled).
- ④ The port is set to digital input/output via the port mode control register (PMCxx)
- ⑤ Set the input/output of the port to output
- ⑥ Enable timer operation (TSmn=1).

Remark: m: unit number (m=0, 1) n: channel number (n=0~3).

- (b) When operation starts with slave channel output mode ($TOMmn = 1$) setting (PWM output))

In slave channel output mode ($TOMmn=1$), the active level depends on the setting of timer output level register m ($TOLmn$).

Figure 5-34 Output state of $TOmn$ pin at PWM output ($TOMmn=1$)



Remarks:

1. Set: The output signal from the $TOmp$ pin changes from an invalid level to a valid level.
Reset: The output signal from the $TOmp$ pin changes from a valid level to an invalid level.
- 2.m: unit number ($m=0, 1$) n: channel number ($p=1\sim3$)

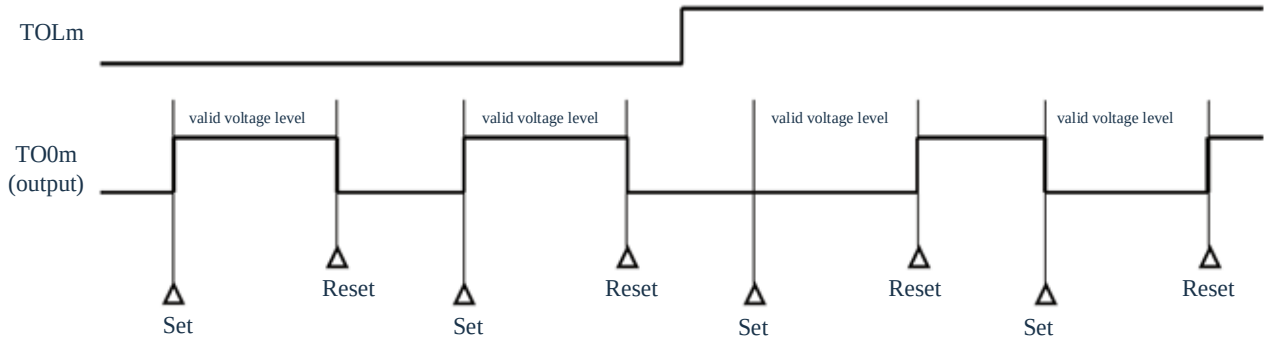
(3) Operation of TOMn pin in slave channel output mode (TOMmn = 1)

(a) When timer output level register m (TOLm) setting has been changed during timer operation

When the TOLm register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TOMn pin change condition. Rewriting the TOLm register does not change the output level of the TOMn pin.

The operation when TOMmn is set to 1 and the value of the TOLm register is changed while the timer is operating (TEmn = 1) is shown below.

Figure 5-35 Operation when the contents of the TOLm register are changed during timer operation



Remarks:

1. Set: The output signal from the TOMn pin changes from an invalid level to a valid level.
Reset: The output signal from the TOMn pin changes from a valid level to an invalid level.
2. m: unit number (m=0, 1) n: channel number (n=0~3)

(b) Set/reset timing

To achieve 0% and 100% output at PWM output, the set timing of the TOMn pin/TOMn bit when generating the master channel timer interrupt (INTTMmn) is delayed by 1 count clock via the slave channel.

When the set condition and reset condition are generated at the same time, the reset condition is given priority.

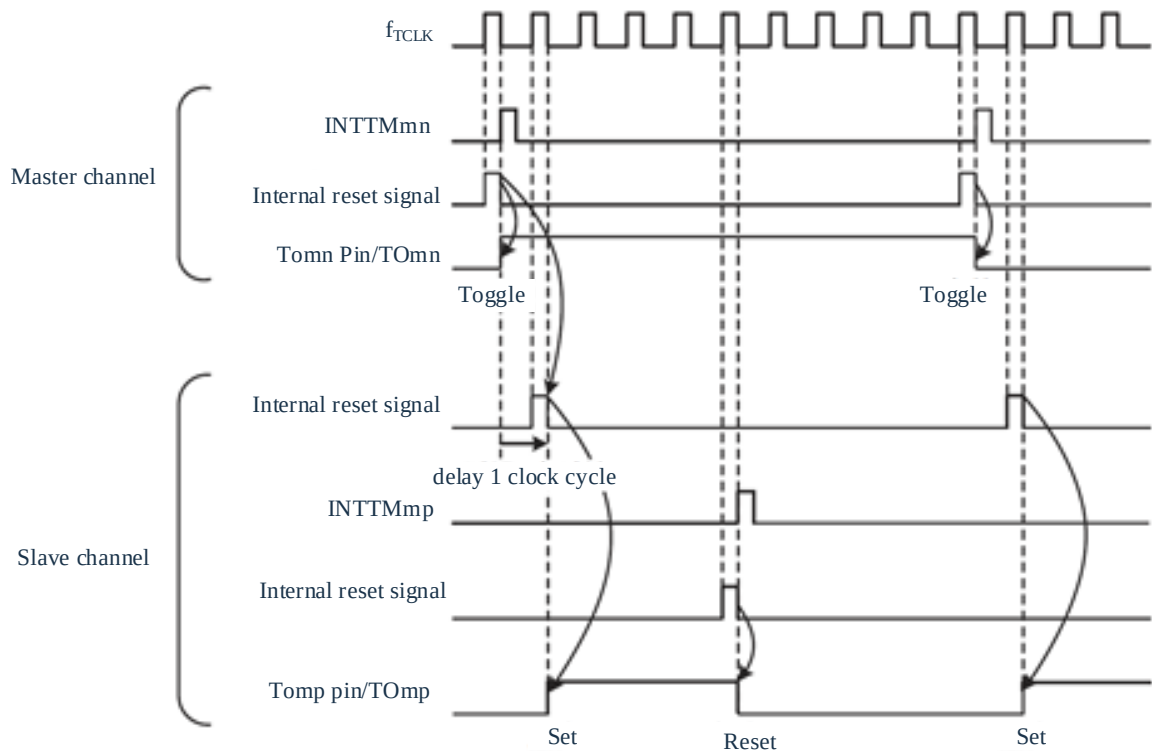
The set/reset operation status when setting the master/slave channel according to the following method is shown in Figure 5-35.

Master channel: TOEmn=1, TOMmn=0, TOLmn=0

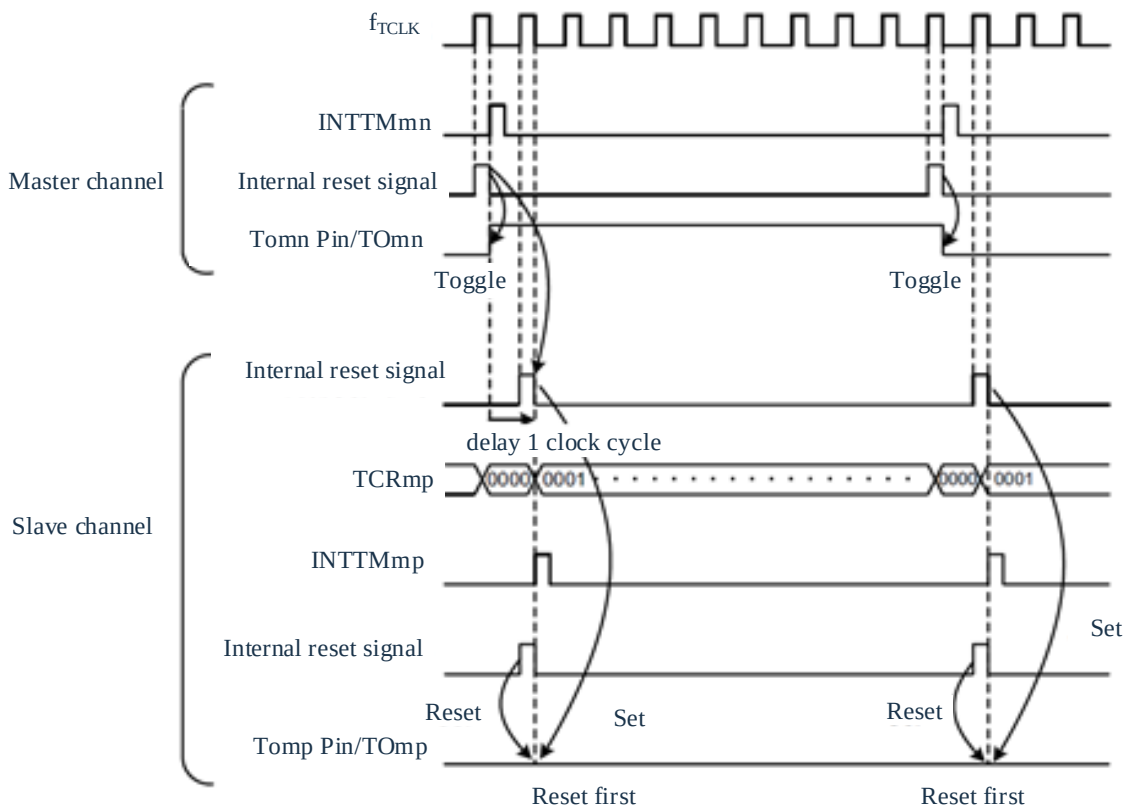
Slave channel: TOEmp=1, TOMmp=1, TOLmp=0

Figure 5-36 Set/reset timing operation status

(1) Basic operation timing



(2) Operation timing when duty cycle is 0%



Remarks:

1. Internal reset signal: TOMn pin reset/toggle signal
Internal set signal: TOMn pin set signal
2. m: unit number (m=0, 1) n: channel number n=0~3 (master channel: n=0, 2)
p: slave channel number (n=0: p=1, 2, 3 n=2: p=3)

5.6.4 One-Time Operation of TOMn Bit

Like the timer channel start register m (TSM), the timer output register m (TOM) has the set bits (TOMn) for all channels and can therefore operate the TOMn bits for all channels at once.

Figure 5-37 One-time operation example of TO0n bit

Before writing

TO0	0	0	0	0	0	0	0	0	0	0	0	TO03	TO02	TO01	TO00
												1	0	1	0

TOE0	0	0	0	0	0	0	0	0	0	0	0	TOE03	TOE02	TOE01	TOE00
												0	0	0	1

Data to be written

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

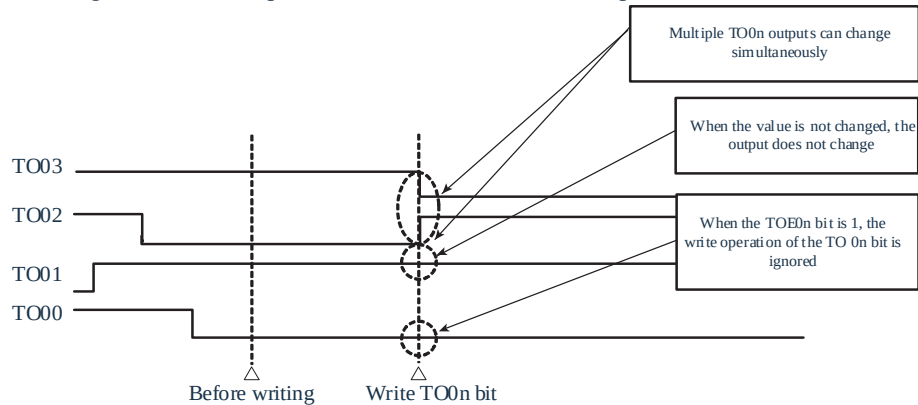
After writing

TO0	0	0	0	0	0	0	0	0	0	0	0	TO03	TO02	TO01	TO00
												0	1	1	0

Only TOMn bits can be written when the TOEmn bit is 0 and write is ignored when the TOMn bit is “1”.

TOMn (channel output) to which TOEmn = 1 is set is not affected by the write operation. Even if the write operation is done to the TOMn bit, it is ignored and the output change by timer operation is normally done.

Figure 5-38 TO0n pin status when the TO0n bit is operated at one time



Remark m: unit number (m=0, 1) n: channel number (n=0~3)

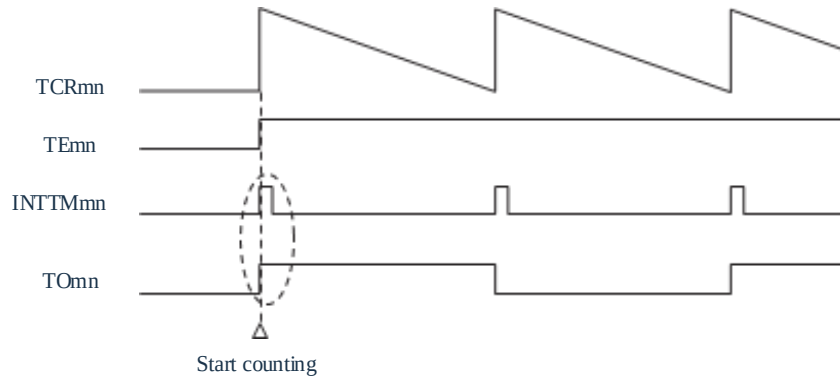
5.6.5 Timer Interrupt and TOMn Pin Output When Counting Starts

In interval timer mode or capture mode, the MDmn0 bit of timer mode register mn (TMRmn) is the bit that sets whether to generate a timer interrupt when counting starts.

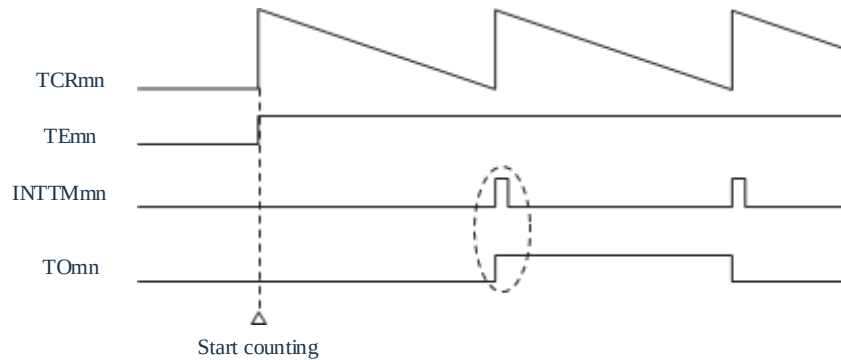
When the MDmn0 bit is 1, the start timing of the count can be known by generating a timer interrupt (INTTMmn). In other modes, the timer interrupt and TOMn output at the start of counting are not controlled. An example of operation when set to interval timer mode (TOEmn=1, TOMmn=0) is shown below.

Figure 5-39 An operation example of timer interrupt and TOMn output when counting starts

(a) When MDmn0 = 1



(b) When MDmn0 = 0



When MDmn0 bit is 1, the timer interrupt (INTTMmn) is output at the start of counting and TOMn is output alternately.

When MDmn0 bit is 0, no timer interrupt (INTTMmn) is output at the start of counting and TOMn is not changed, while INTTMmn is output and TOMn is alternately output after 1 cycle of counting.

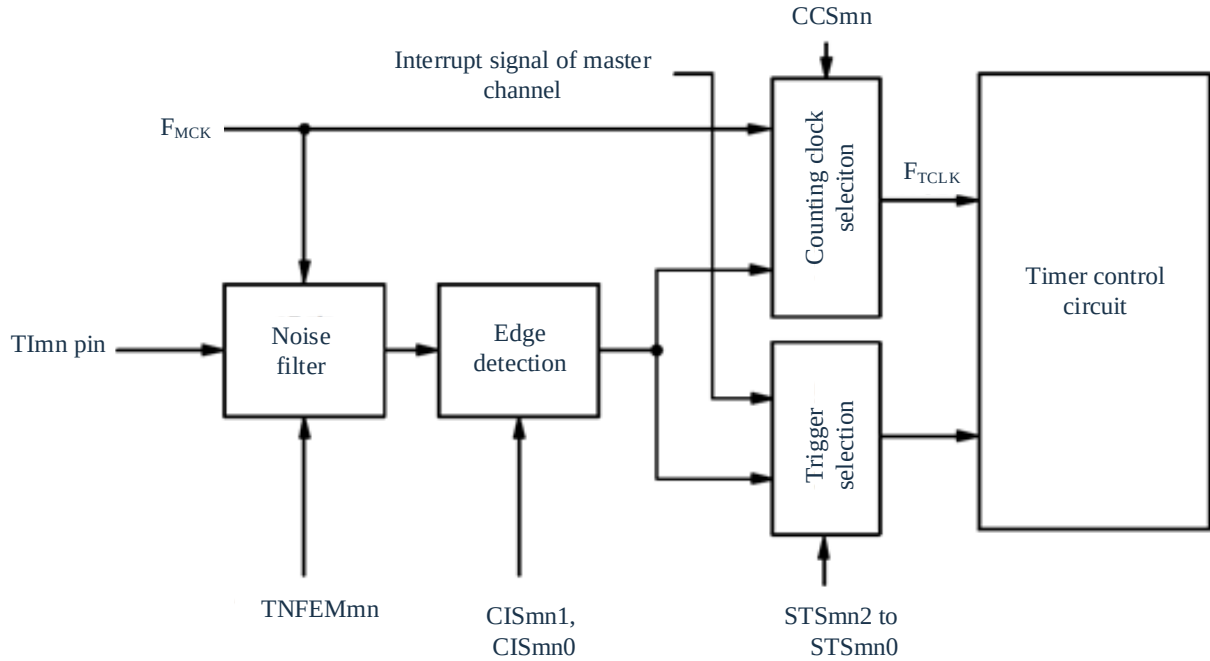
Remark: m: unit number (m=0, 1) n: channel number (n=0~3)

5.7 Control of Timer Input (TImn)

5.7.1 Structure of TImn Pin Input Circuit

The signal from the timer input pins is input to the timer control circuit via a noise filter and the edge detection circuit. For pins that need to be removed from noise, the corresponding pin noise filter must be set to enable. The block diagram of the input circuit is as follows.

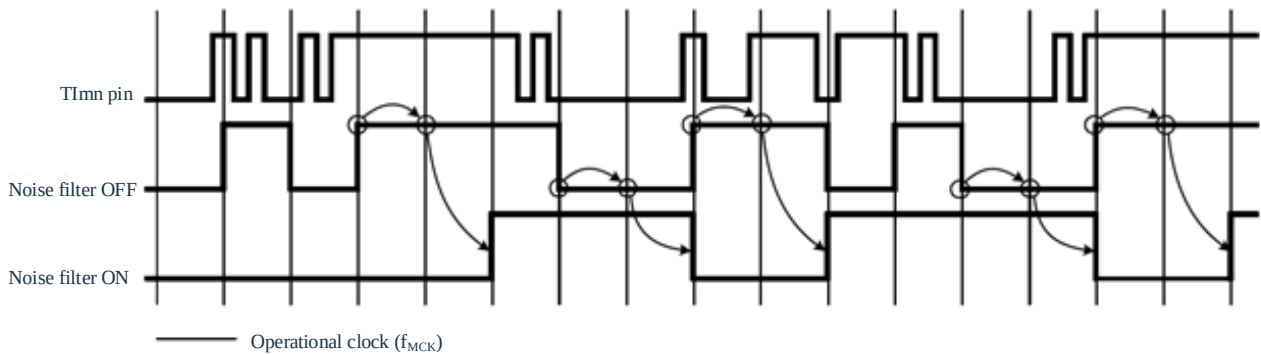
Figure 5-40 Structure of input circuit



5.7.2 Noise Filter

When the noise filter is inactive, synchronization is performed only by the operation clock (F_{MCK}) of channel n. When the noise filter is active, 2 clocks are detected after synchronization by the operation clock (F_{MCK}) of channel n. The waveform of the TM4Imn input pin after the noise filter circuit with the noise filter ON or OFF is shown below.

Figure 5-41 Sample waveform of TImn input pin with noise filter ON or OFF



Caution: The input waveform on the TImn pin is used to illustrate the operation of the noise filter is ON or OFF. For actual operation, the input must be made in accordance with the TImn input high- and low-level width shown in AC characteristics.

5.7.3 Cautions on Channel Input Operation

When set to not use the timer input pin, no operating clock is provided to the noise filter circuit. Therefore, the following wait time is required from the time set to use the timer input pin to the time the channel corresponding to the timer input pin is set to operate the enable trigger.

(1) Noise filter OFF

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are 0 and then one of them is set, wait for at least two cycles of the operating clock (F_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

(2) Noise filter ON

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are all 0 and then one of them is set, wait for at least four cycles of the operating clock (F_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

5.8 Independent Channel Operation Function of Universal Timer Unit

5.8.1 Operation as Interval Timer/Square Wave Output

(1) Interval timer

It can be used as a reference timer to generate INTTMmn (timer interrupt) at fixed intervals. The interrupt generation period can be calculated using the following equation:

$$\text{INTTMmn (timer interrupt) generation period} = \text{count clock period} \times (\text{TDRmn set value} + 1)$$

(2) Operation as square wave output

The TOMn alternates outputs while generating the INTTMmn, outputting a square wave with a 50% duty cycle. The period and frequency of the square wave can be calculated using the following equation:

$$\text{Period of square wave output from TOMn} = \text{Period of count clock} \times (\text{TDRmn set value} + 1) \times 2$$

$$\text{Frequency of square wave output from TOMn} = \text{Frequency of count clock} / \{(\text{TDRmn set value} + 1) \times 2\}$$

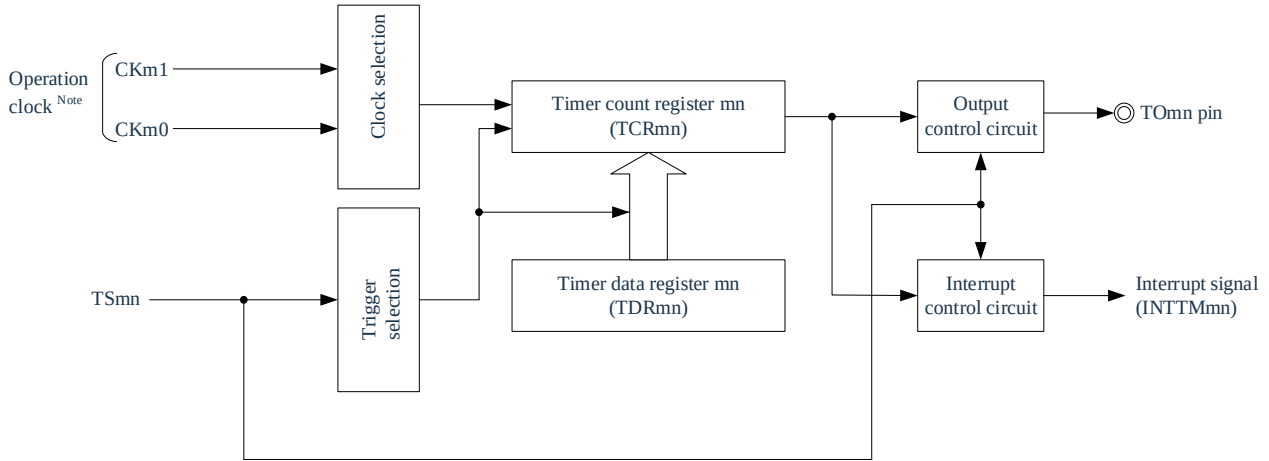
In the interval timer mode, the timer count register mn (TCRmn) is used as a decrement counter.

After setting the channel start trigger bit (TSmn, TSHm1, TSHm3) of the timer channel start register m (TSM) to 1, the value of timer data register mn (TDRmn) is loaded into the TCRmn register by the first count clock. At this time, if the MDmn0 bit of the timer mode register n (TMRmn) is 0, INTTMmn is not output and TOMn is not alternately output. If the MDmn0 bit of TMRmn register is 1, INTTMmn is output and TOMn is alternately output. Then, the TCRmn register is decremented by the count clock.

If the TCRmn becomes 0000H, the INTTMmn and TOMn are output alternately by the next count clock. At the same time, the value of TDRmn register is loaded into TCRmn register again. After that, continue the same operation.

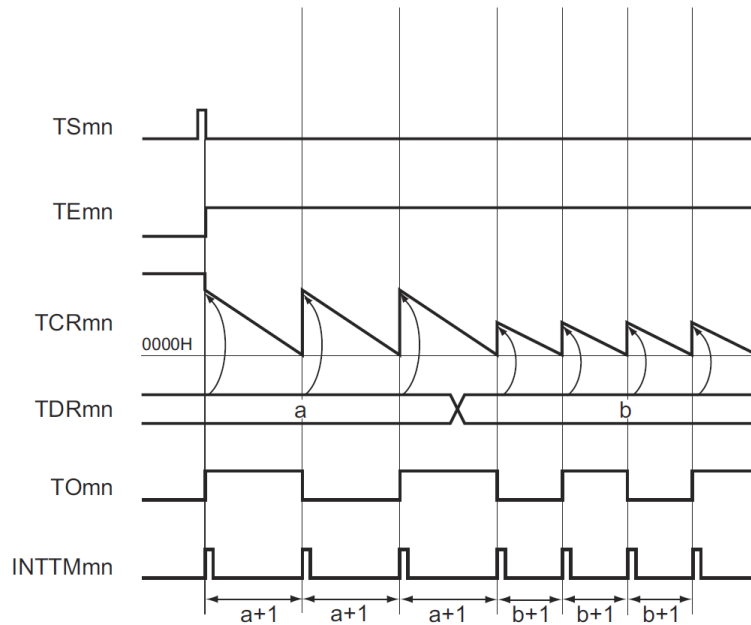
The TDRmn register can be rewritten at any time, and the rewritten TDRmn register value is valid from the next cycle.

Figure 5-42 Example of basic timing operating as an interval timer/square wave output (MDmn0=1)



Note: At channel 1 and channel 3, it is possible to select the clock from CKm0, CKm1, CKm2 and CKm3.

Figure 5-43 Example of basic timing operating as an interval timer/square wave output (MDmn0=1)

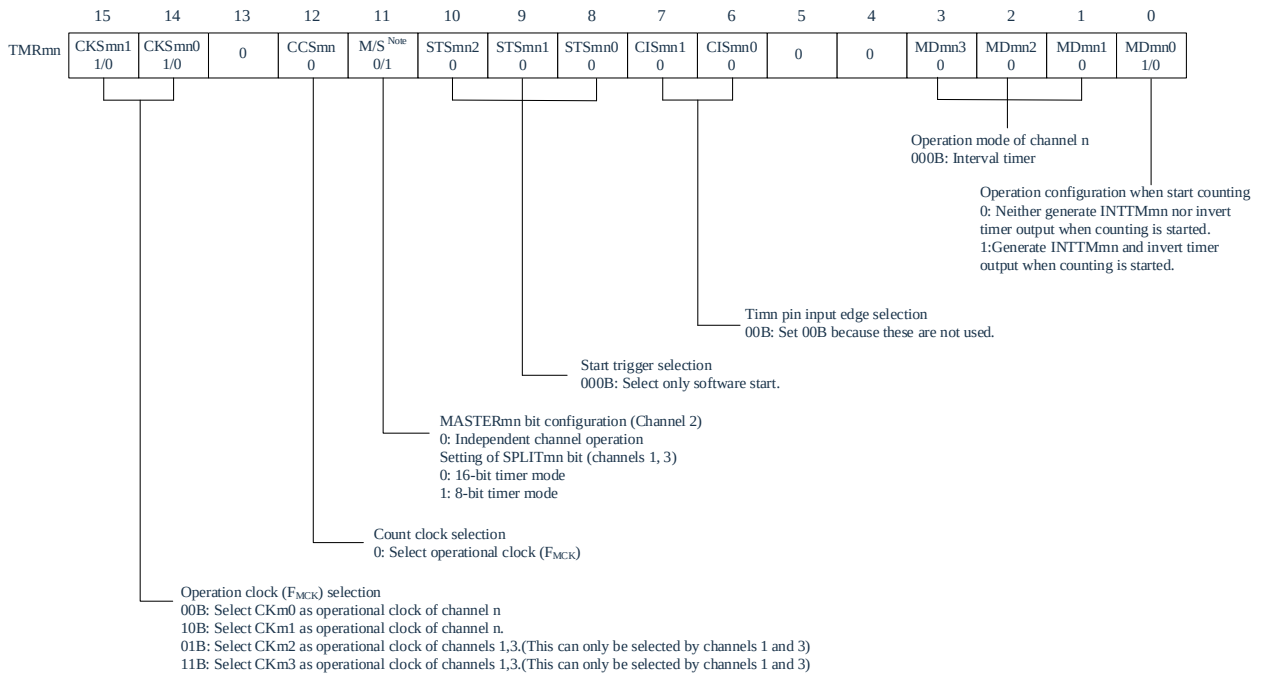


Remarks:

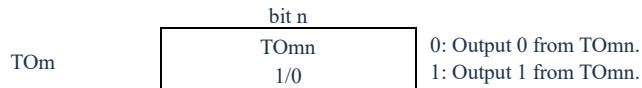
1. m: unit number (m=0, 1) n: channel number (n=0~3)
 2. TSmn: Bit n of timer channel start register m (TSm)
- TEMn: Bit n of timer channel enable status register m (TEm)
- TCRmn: Timer count register mn (TCRmn)
- TDRmn: Timer data register mn (TDRmn)
- TOMn: TOMn pin output signal

Figure 5-44 Example of register setting contents for interval timer/square wave output

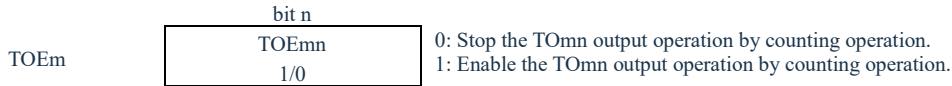
(a) Timer mode register mn (TMRmn)



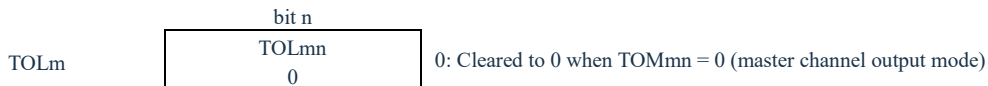
(b) Timer output register m (TOM)



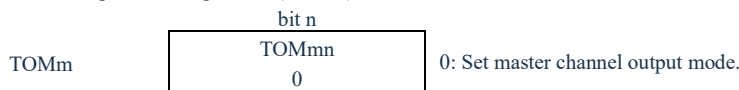
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to "0".

Remark m: unit number(m= 0, 1) n: channel number(n=0 ~3)

Figure 5-45 Procedure for interval timer/square wave output function

	Software operation	Hardware status
TAU initial settings		The input clock of timer unit m is in the stop supply state. (Stop providing clock, cannot write to each register)
	Set the TM4mEN bit of peripheral enable register 0(PER0) to "1".	The input clock of timer unit m is in the supply state. (Start clock supply, can write to each register)
	Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 ~ CKm3.	
Initial setting of channels	Set the timer mode register mn (TMRmn) (to determine the channel's operation mode). Set the interval (period) value for the timer data register mn (TDRmn).	The channel is in the stop state. (Provides clock, and consumes some power)
	Using TOMn output: Set the TOMmn bit of Timer Output Mode Register m (TOMm) to "0" (master channel output mode). Set the TOLmn bit to "0". Set the TOMn bit to determine the initial level of the TOMn output. Set the TOEmn bit to "1" and enable TOMn output. Set the Port Register and Port Mode Register to "0".	The TOMn pin is in Hi-Z output state. When the port mode register is in output mode and the port register is "0", the TOMn initial set level is output. The TOMn remains unchanged because the channel is in the stop state. The TOMn pin outputs the level set by the TOMn.
	(The TOEmn bit set to "1" only when the TOMn output is used and restarted) Set the TSmn bit to "1". Since the TSmn bit is a trigger bit, it automatically returns to "0".	The TEMn bit becomes "1" and starts counting. Load the value of the TDRmn register into the Timer Count Register mn (TCRmn). When the MDmn0 bit of TMRmn register is "1", INTTMmn is generated and TOMn is output alternately.
Restart operation	In operation	The counter (TCRmn) performs decremental counting. If the count reaches "0000H", the value of the TDRmn register is loaded into the TCRmn register again and the count continues. When TCRmn is detected as "0000H", INTTMmn is generated and TOMn is alternately output. Thereafter, repeat this operation.
	Stop operation	The TEMn bit becomes "0" and stops counting. The TCRmn register holds the count value and stops counting. The TOMn output is not initialized but remains its state. The TOMn pin outputs the level set by the TOMn bit.
	TAU stop	The output level of the TOMn pin is maintained by the port function.
	Set the TM4mEN bit of the PER0 register to "0".	The input clock of timer unit m is in the stop-providing state. Initialize all circuits and the SFR for each channel. (TOMn bit becomes "0" and TOMn pin becomes port function)

Remark m: unit number (m= 0, 1) n: channel number (n=0 ~ 3)

5.8.2 Operation as External Event Counter

It can be used as an event counter to count the active edges (external events) detected on the TIMn pin input and generate an interrupt if the specified count value is reached. The specified count value can be calculated using the following equation:

$$\text{Specified count value} = \text{TDRmn set value} + 1$$

In the event counter mode, the timer count register mn (TCRmn) is used as a decrement counter.

The value of timer data register mn (TDRmn) is loaded into the TCRmn register by setting any channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSM) to 1.

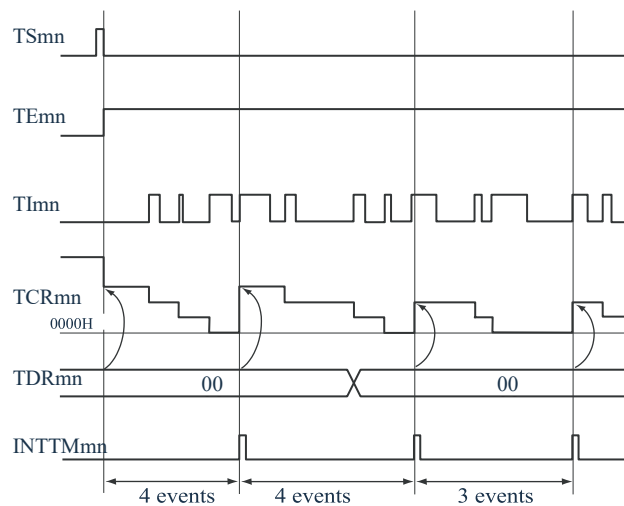
The TCRmn register decrements the count while detecting the active edge of the TIMn pin input. If TCRmn becomes “0000H”, the value of TDRmn register is loaded again and INTTMmn is output.

After that, continue the same operation.

The output must be stopped by setting the TOEmn bit of the timer output enable register m (TOEm) to 0 because the TOMn pin outputs irregular waveforms based on external events.

The TDRmn register can be rewritten at any time, and the rewritten TDRmn register value is valid for the next cycle.

Figure 5-46 Example of basic timing operating as external event counter



Remarks:

1. m: unit number (m=0, 1) n: channel number (n=0~3)
2. TSmn: Bit n of timer channel start register m (TSM)

TEmn: Bit n of timer channel enable status register m (TEM)

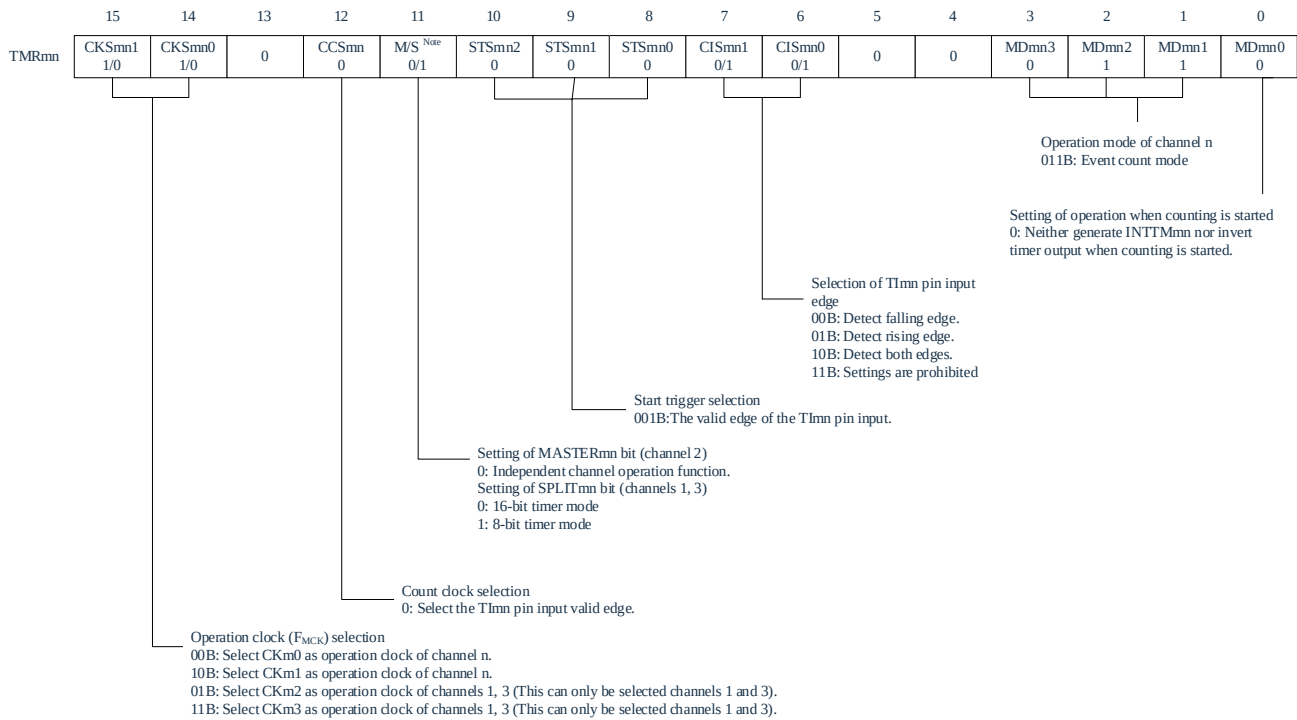
TImn: TIMn pin input signal

TCRmn: Timer count register mn (TCRmn)

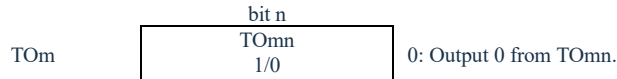
TDRmn: Timer data register mn (TDRmn)

Figure 5-47 Example of register setting contents in external event counter mode

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOM)



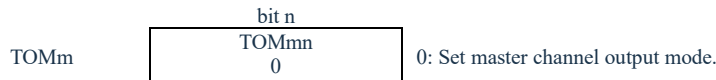
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to "0"

Remark m: unit number (m= 0, 1) n: channel number (n=0~3)

Figure 5-48 Procedure for external event counter function

	Software operation	Hardware status
Timer4 initial settings		The input clock of timer unit m is in the stop supply state. (Stop clock supply, cannot write to each register)
	Set the TM4mEN bit of peripheral enable register 0(PER0) to "1".	The input clock of timer unit m is in the supply state and the channels are in the stop state. (Start providing clock, can write to each register)
	Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 ~ CKm3.	
Initial setting of channels	Set the corresponding bit of the Noise Filter Enable Register (NFEN1) to "0" (OFF) or "1" (ON). Set the timer mode register mn (TMRmn)(to determines the operating mode of the channel). Set the count value for the timer data register mn (TDRmn). Set the TOEmn bit of the timer output enable register m (TOEm) to "0".	The channel is in the stop state. (Provides clock, and consumes some power)
Start operating	Set the TSmn bit to "1". Since the TSmn bit is a trigger bit, it automatically returns to "0".	The TEmn bit becomes "1" and starts counting. The value of the TDRmn register is loaded into the timer count register mn (TCRmn) and enter the detection wait state of the input edge of the TImn pin.
In operation	The setting of the TDRmn register can be changed at will. The TCRmn register can be read at any time. The TSRmn register is not used. The setting of the TMRmn register, the TOMmn bit, the TOLmn bit, the Tomn bit and the TOEmn bit cannot be changed.	Whenever the input edge of the TImn pin is detected, the counter (TCRmn) is decremented. If the count reaches "0000H", the value of the TDRmn register is loaded into the TCRmn register again and the count continues. When TCRmn is detected as "0000H", INTTMmn is generated. Thereafter, repeat this operation.
Stop operating	Set the TTmn bit to "1". The operation automatically returns to "0" because the TTmn bit is a trigger bit.	The TEmn bit becomes "0" and stops counting. The TCRmn register holds the count value and stops counting.
Timer4 stop	Set the TA4mEN bit of the PER0 register to "0".	The input clock of timer unit m is in the stop-providing state. Initialize all circuits and the SFR for each channel.

Restart operation

5.8.3 Operation as Frequency Divider

The clock input from the TI00 pin can be divided and used as a divider for the output of the TO00 pin.

The divided clock frequency of the TO00 output can be calculated using the following equation:

- Select rising or falling edge:
 $\text{Divider clock frequency} = \text{input clock frequency} / \{(\text{TDR00 set value} + 1) \times 2\}$
- Select both edges:
 $\text{Divider clock frequency} \approx \text{input clock frequency} / (\text{TDR00 set value} + 1)$

In the interval timer mode, the timer count register 00 (TCR00) is used as a decrement counter.

After setting the channel start trigger bit (TS00) of timer channel start register 0 (TS0) to 1, the value of timer data register 00 (TDR00) is loaded into the TCR00 register by detecting an active edge of TI00. At this time, if the MD000 bit of Timer Mode Register 00 (TMR00) is 0, INTTM00 is not output and TO00 is not output alternately; if the MD000 bit of TMR00 register is 1, INTTM00 is output and TO00 is not output alternately. If the MD000 bit of TMR00 register is 1, INTTM00 is output and TO00 is output alternately.

The TCR00 register then counts down through the active edge of the TI00 pin input. If TCR00 changes to “0000H”, TO00 performs an alternate output. At the same time, the value of the TDR00 register is loaded into the TCR00 register and counting continues.

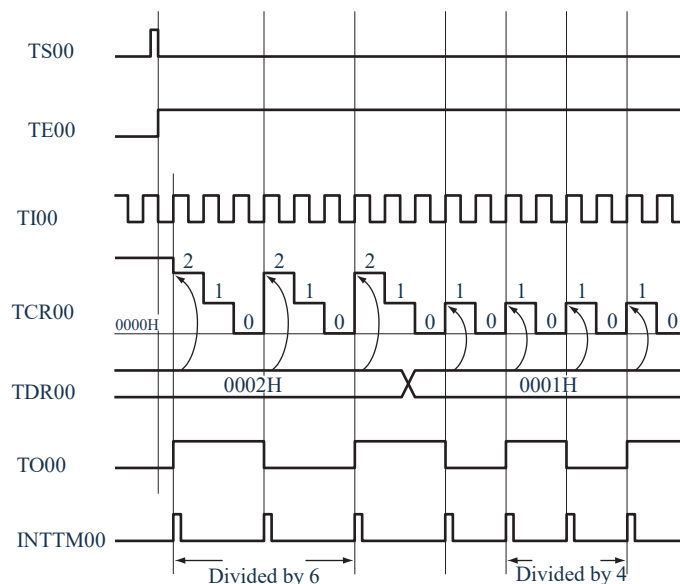
If double edge detection is selected for the TI00 pin input, the duty cycle error of the input clock affects the clock period of the TO00 output’s division.

The clock period of the TO00 output contains the sampling error of 1 run clock cycle.

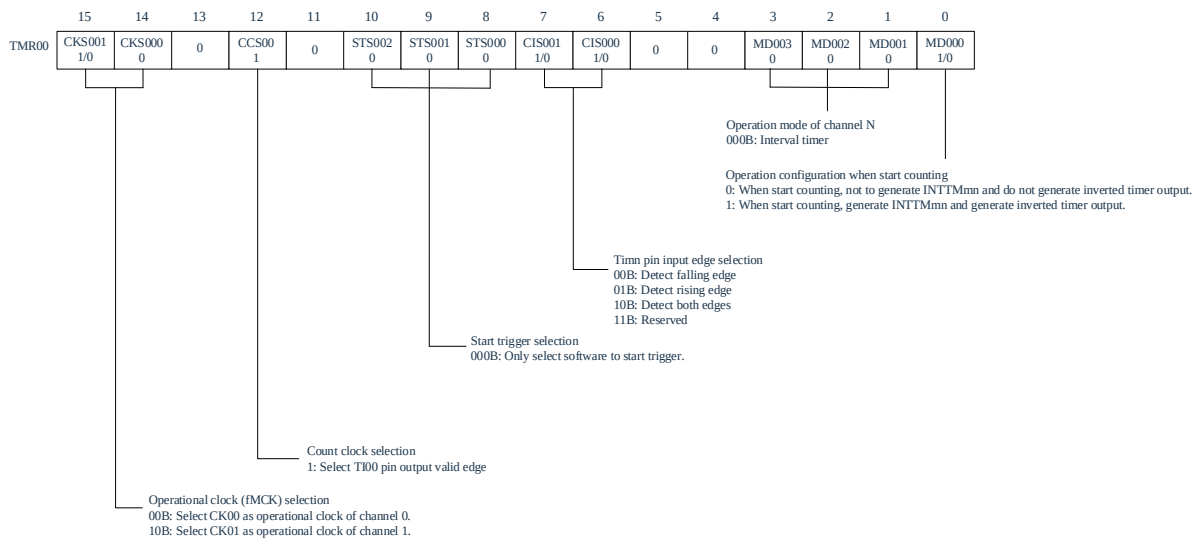
$$\text{Clock period of the TOmn output} = \text{supposed TOmn output clock period} \pm \text{operating clock period (error)}$$

The TDRmn register can be rewritten at any time, and the rewritten TDRmn register value is valid for the next cycle.

Figure 5-49 Example of basic timing operating as a frequency divider (MD000=1)



Remark: TS00: Bit 0 of timer channel start register 0 (TS0)
 TE00: Bit 0 of timer channel enable status register (TE0)
 TI00: TI00pin input signal
 TCR00: Timer count register 00 (TCR00)
 TDR00: Timer data register 00 (TDR00)
 TO00: TO00pin output signal

Figure 5-50 Example of register contents setting when operating as frequency divider
(a) Timer mode register 00 (TMR00)

(b) Timer output register 0 (TO0)

	bit 0		
TO0	TO00	0	0: Output 0 by TO00.
	1/0	1	1: Output 1 by TO00.

(c) Timer output enable register 0 (TOE0)

	bit 0		
TOE0	TOE00	0	0: Stop the TO00 output performed by the counting operation.
	1/0	1	1: Enable TO00 output performed by the counting operation.

(d) Timer output level register 0 (TOL0)

	bit 0		
TOL0	TOL00	0	0: Set to 0 in the master channel output mode (TOM0=0).
	0	0	

(e) Timer output mode register 0 (TOM0)

	bit n		
TOM0	TOM00	0	0: Set master channel output mode.
	0	0	

Figure 5-51 Procedure for frequency divider function

	Software operation	Hardware status
Timer4 initial settings		The input clock of timer unit 0 is in the stop supply state. (Stop providing clock, cannot write to each register)
	Set the TM4mEN bit of peripheral enable register 0 (PER0) to "1".	The input clock of timer unit 0 is in the supply state and the channels are in the stop state. (Start providing clock, can write to each register)
	Set the timer clock selection register 0 (TPS0). Determine the clock frequency of CK00 ~ CK03.	
Initial setting of channels	Set the corresponding bit of the Noise Filter Enable Register (NFEN1) to "0" (OFF) or "1" (ON). Set the timer mode register 00 (TMR00)(to determines the operating mode of the channel, select edge detection). Set the interval (cycles) value for the timer data register 00 (TDR00).	The channel is in the stop state. (Supply clock, and consumes some power)
	Set TOM00 bit of timer output mode register 0 (TOM0) to "0" (master control channel output mode). The TOL00 bit must be set to "0". Set the TO00 bit and determine the initial level of TO00 output. Set TOE00 bit to "1" and enable TO00 output. Set the Port Register and Port Mode Register to "0".	The TO00 pin is in Hi-Z output state. When the port mode register is in output mode and the port register is "0", the TO00 initial set level is output. Since the channel is in stop state, TO00 does not change. The TO00 pin outputs the level set by the TO00.
	Set TOE00 bit to "1" (only limited to restart operation). The TS00 bit must be set to "1". The operation automatically returns to "0" because the TS00 bit is a trigger bit.	The TE00 bit becomes "1" and starts counting. Load the value of the TDR00 register into the Timer Count Register 00 (TCR00). When the MD000 bit TMR00 register is "1", INTTM00 is generated and TO00 is output alternately.
	The setting of the TDR00 register can be changed at will. The TCR00 register can be read at any time. The TSR00 register is not used. The TO0 register and TOE0 register settings can be changed. The setting of the TMR00 register, the TOM00 bit and the TOL00 bit cannot be changed.	The counter (TCR00) performs decremental counting. If the count reaches "0000H", the value of the TDR00 register is loaded into the TCR00 register again and the count continues. When the TCR00 bit is "0000H", INTTM00 is generated and TO00 is output alternately. Thereafter, repeat this operation.
Stop operating	The TT00 bit must be set to "1". The operation automatically returns to "0" because the TT00 bit is a trigger bit. Set the TOE00 bit to "0" and set the value for the TO00 bit.	The TE00 bit becomes "0" and starts counting. The TCR00 register holds the count value and stops counting. The TO00 output is not initialized but remains its state. The TO00 pin outputs the level set by the TO00.
	To maintain the output level of the TO00 pin: Set TO00 bit to "0" after setting the value to be held for the port register. No need to maintain the output level of the TO00 pin: No need to set.	The output level of the TO00 pin is maintained by the port function.
Timer4 stop	Set the TM4mEN bit of the PER0 register to "0".	The input clock of timer unit 0 is in the stop-providing state. Initialize all circuits and the SFR for each channel. (TO00 bit becomes "0" and TO00 pin becomes port function)

Restart operation

5.8.4 Operation as Input Pulse Interval Measurement

The count value can be captured at the active edge of TIMn and the interval between TIMn input pulses can be measured. The software operation (TSmn=1) can also be set to capture the count value during the period when the TEMn bit is 1.

The pulse interval can be calculated using the following equation:

$$\text{TIMn input pulse interval} = \text{period of counting clock} \times ((10000\text{H} \times \text{TSRmn: OVF}) + (\text{TDRmn captured value} + 1))$$

Caution An operation clock error is generated because the TIMn pin input is sampled by the operation clock selected by the CKSmn bit of the Timer Mode Register mn (TMRmn).

In capture mode, the timer count register mn (TCRmn) is used as an increment counter.

If the channel start trigger bit (TSmn) of the timer channel start register m (TSM) is set to 1, the TCRmn register is incrementally counted from "0000H" by the count clock.

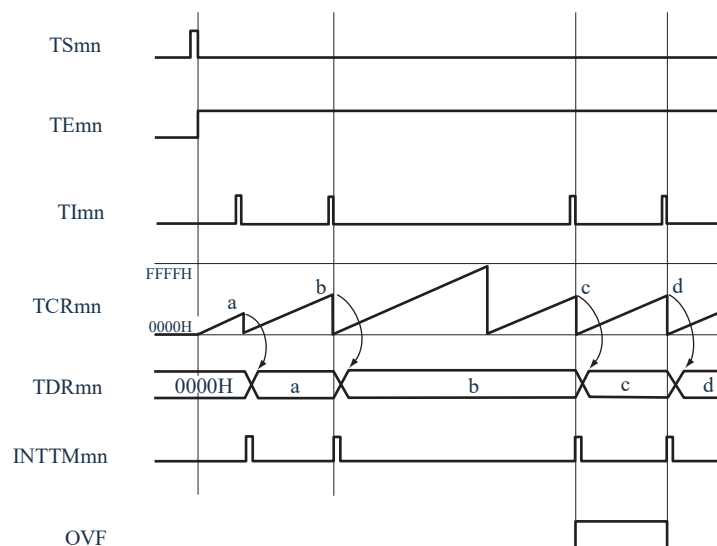
If the active edge of the TIMn pin input is detected, the count value of TCRmn register is transferred (captured) to Timer Data Register mn (TDRmn), and the TCRmn register is cleared to "0000H", and then INTTMmn is output. If the counter overflows, the OVF bit of Timer Status Register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, continue the same operation.

While capturing the count value to the TDRmn register, the OVF bit of the TSRmn register is updated according to whether or not overflow occurs during the measurement, and the overflow status of the captured value can be confirmed.

Even if the counter counts 2 or more complete cycles, the overflow is considered to have occurred and the OVF bit of the TSRmn register is set to 1. However, when two or more overflows occur, the interval value cannot be measured normally by the OVF bit.

Set the STSmn2~STSmn0 bit of the TMRmn register to "001B", and use the valid edge of TIMn for start trigger and capture trigger.

Figure 5-52 Example of basic timing operating as an input pulse interval measurement (MDmn0=0)

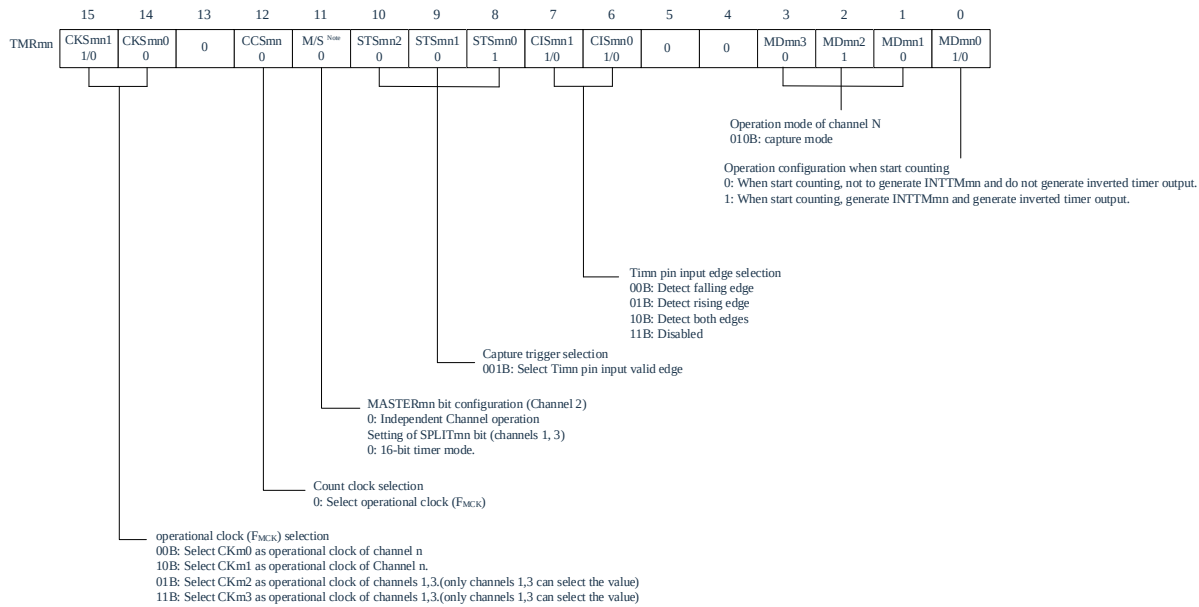


Remarks:

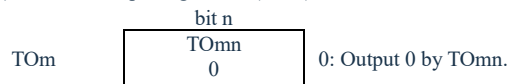
1. m: unit number (m= 0) n: channel number (n=0~3)
2. TSmn: Bit n of timer channel start register m (TSM)
- TEMn: Bit n of timer channel enable status register m (TEM)
- TImn: TIMn pin input signal
- TCRmn: Timer count register mn (TCRmn)
- TDRmn: Timer data register mn (TDRmn)
- OVF: Bit0 of timer status register mn (TSRmn)

Figure 5-53 Example of register contents setting in measuring input pulse interval

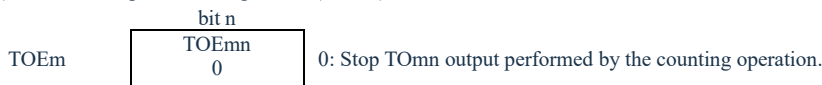
(a) Timer mode register mn (TMRmn)



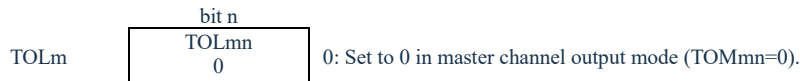
(b) Timer output register m (TOM)



(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note TMRm2: MASTERmn bit
 TMRm1, TMRm3: SPLITmn bit
 TMRm0: Fixed to "0".

Remark m: unit number (m=0) n: channel number (n=0~3)

Figure 5-54 Procedure for entering the pulse interval measurement function

	Software operation	Hardware status
Timer4 initial settings		The input clock of timer unit m is in the stop supply state. (Stop to supply clock, cannot write to each register)
	Set the TM4mEN bit of the peripheral enable register 0(PER0) to "1".	The input clock of timer unit m is in the supply state and the channels are in the stop state. (Start to supply clock, can write to each register)
	Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 ~ CKm3.	
Initial setting of channels	Set the corresponding bit of the Noise Filter Enable Register (NFEN1) to "0" (OFF) or "1" (ON). Set the timer mode register mn (TMRmn)(to determines the operating mode of the channel).	The channel is in the stop state. (Provides clock, and consumes some power)
Start operation	Set the TSmn bit to "1". Since the TSmn bit is a trigger bit, it automatically returns to "0".	The TEmn bit becomes "1" and starts counting. Clear the timer count register mn (TCRmn) to "0000H". When the MDmn0 bit of TMRmn register is "1", INTTMmn is generated.
In operation	The setting values of the CISmn1 bit and the CISmn0 bit of the TMRmn register can be changed. The TDRmn register can be read at any time. The TCRmn register can be read at any time. The TSRmn register can be read at any time. The setting of the the TOMmn bit, the TOLmn bit, the TOmn bit and the TOEmn bit cannot be changed.	The counter (TCRmn) starts incremental counting from "0000H" and transfers (captures) the count value to the timer data register mn (TDRmn) if it detects an active edge on the TImn pin input or sets TSmn bit to "1". If the counter overflows, set the OVF bit of Timer Status Register mn (TSRmn). If the counter does not overflow, the OVF bit is cleared. Thereafter, repeat this operation.
Stop operation	Set the TTmn bit to "1". The operation automatically returns to "0" because the TTmn bit is a trigger bit.	The TEmn bit becomes "0" and stops counting. The TCRmn register holds the count value and stops counting. The OVF bit of the TSRmn register remains unchanged.
Timer4 stop	Set the TM4mEN bit of the PER0 register to "0".	The input clock of timer unit m is in the stop-providing state. Initialize all circuits and the SFR for each channel.

Remark m: unit number (m= 0, 1) n: channel number (n=0~3)

5.8.5 Operation as Input Signal High-/Low-Level Width Measurement

Caution When used as a LIN-bus support function, bit1 (ISC1) of the Input Switching Control Register (ISC) must be set to 1 and RxD0 should be used instead of TIMn in the following description.

The signal width (high-/low-level width) of TIMn can be measured by starting counting at one edge of the input to the TIMn pin and capturing the count value at the other edge. The TIMn signal width of the TIMn output can be calculated using the following equation:

$$\text{Signal width of TIMn input} = \text{period of count clock} \times ((10000H \times \text{TSRmn: OVF}) + (\text{TDRmn captured value} + 1))$$

Caution Because the TIMn pin inputs are sampled by the operation clock selected by the CKSmn bit of the Timer Mode Registermn (TMRmn), an error of 1 operation clock is generated.

In the Capture & Single Count mode, the timer count register mn (TCRmn) is used as an increment counter. If the channel start trigger bit (TSmn) of the timer channel start register m(TSm) is set to 1, the TEMn bit becomes 1, and the start edge detection wait state of the TIMn pin is entered.

If the start edge of the TIMn pin input (rising edge of the TIMn pin input at the time of high-level width measurement) is detected, it is synchronized with the count clock and counts incrementally from “0000H”. Then, if an active capture edge is detected (falling edge of TIMn pin input at the time of high-level width measurement), the count value is transferred to the Timer Data Register mn (TDRmn) and INTTMmn is output at the same time. If the counter overflows, the OVF bit of the Timer Status Register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. The value of the TCRmn register changes to “Value passed to TDRmn register +1”, and the start edge detection wait state of the TIMn pin is entered. After that, continue the same operation.

While capturing the count value to the TDRmn register, the OVF bit of the TSRmn register is updated according to whether or not overflow occurs during the measurement, and the overflow status of the captured value can be confirmed.

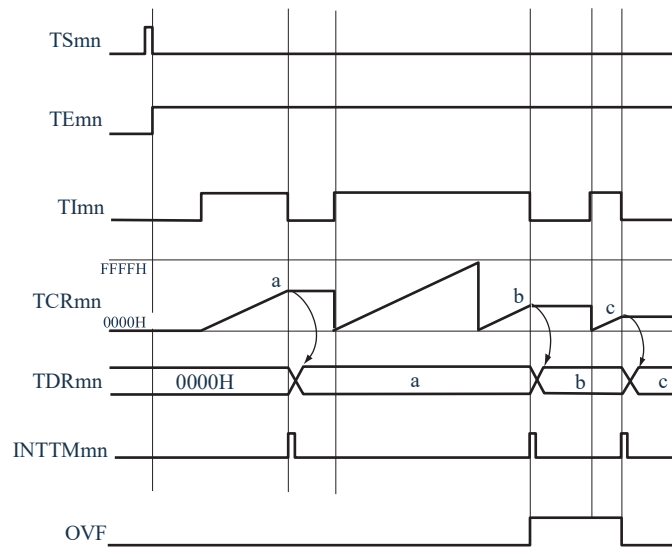
Even if the counter counts 2 or more complete cycles, the overflow is considered to have occurred and the OVF bit of the TSRmn register is set to 1. However, when two or more overflows occur, the interval value cannot be measured normally by the OVF bit.

The CISmn1 and CISmn0 bits of the TMRmn register can be used to set whether the high-level width or low-level width of the TIMn pin is to be measured. This function is designed to measure the input signal width of the TIMn pin, so the TSmn bit cannot be set to 1 during the period when the TEMn bit is 1.

CISmn1, CISmn0=10B of the TMRmn register: Measures the low-level width.

CISmn1, CISmn0=11B of the TMRmn register: Measures the high-level width.

Figure 5-55 Example of basic timing operating as high-/low-level width measurement of input signal

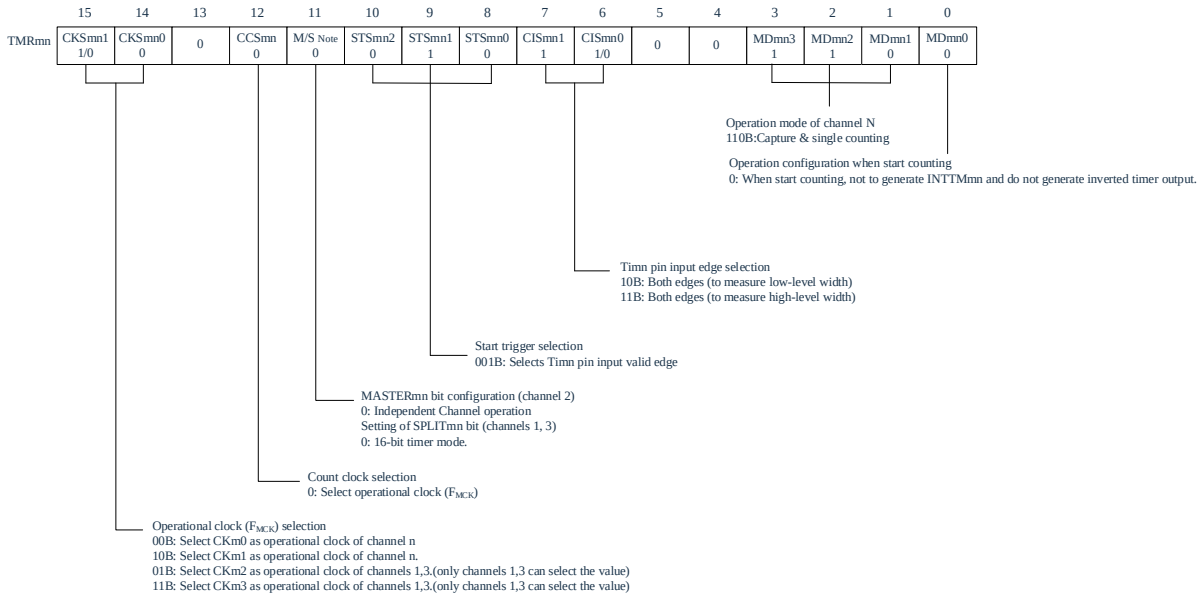


Remarks:

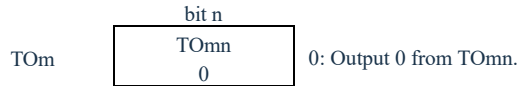
1. m: unit number (m= 0, 1) n: channel number (n=0~3)
2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 5-56 Example of register contents setting in measuring high-/low-level width of input signal

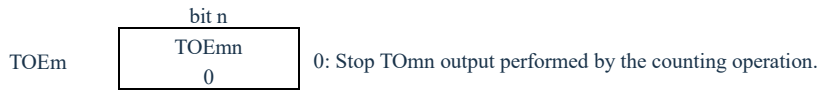
(a) Timer mode register mn (TMRmn)



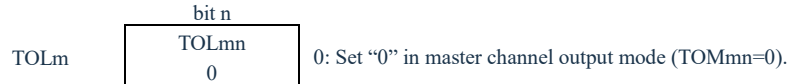
(b) Timer output register m (TOM)



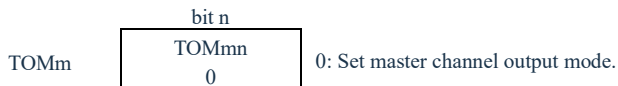
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



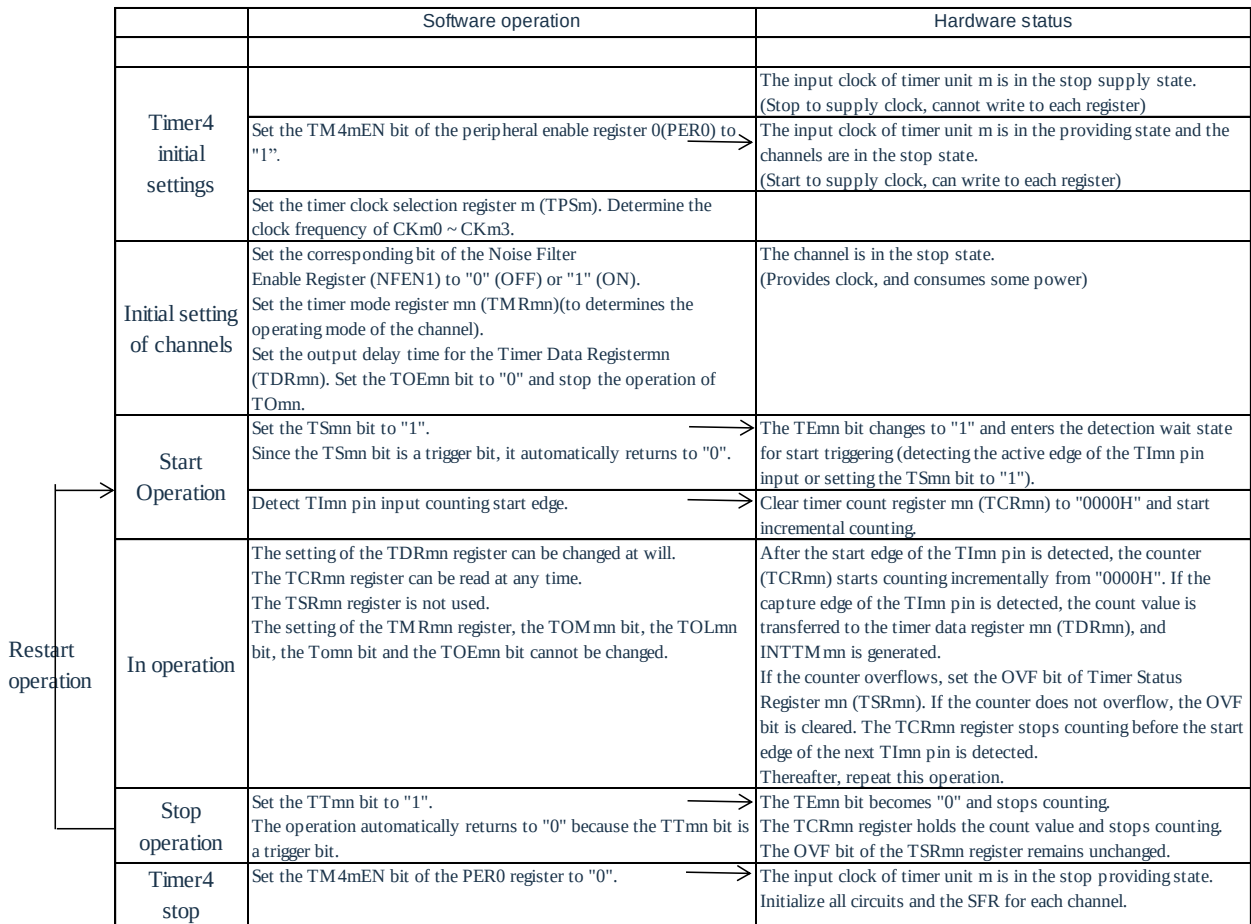
(e) Timer output mode register m (TOMm)



Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to "0".

Remark m: unit number (m=0, 1) n: channel number (n=0 ~ 3)

Figure 5-57 Procedure for high-/low-level width measurement function of input signal



Remark: m: unit number (m= 0, 1) n: channel number (n=0 ~ 3)

5.8.6 Operation as Delay Counter

The count can be decremented by the active edge detection (external event) of the TIMn pin input and INTTMMn (timer interrupt) is generated at any set interval.

During the period when the TEMn bit is 1, the TSMn bit can be set to 1 by software to start decreasing counting and generate INTTMMn (timer interrupt) at any set interval.

The interrupt generation period can be calculated using the following equation:

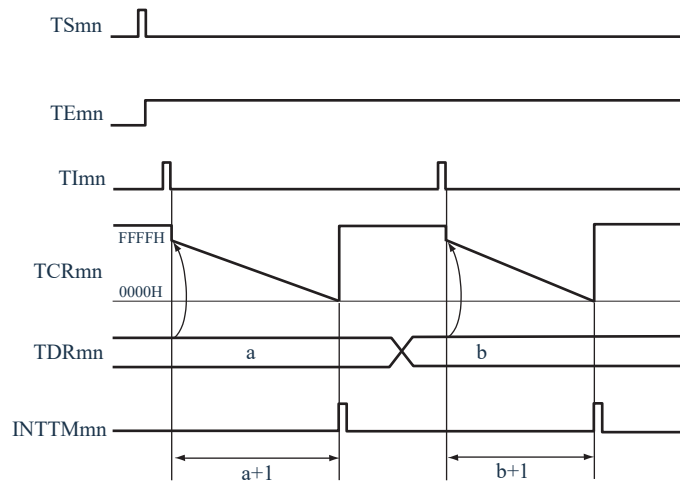
$$\text{INTTMMn (timer interrupt) generation period} = \text{counting clock period} \times (\text{TDRmn set value} + 1)$$

In the single count mode, the timer count register mn (TCRmn) is used as a decrement counter.

If the channel start trigger bit (TSMn, TSHm1, TSHm3) of the timer channel start register m(TSm) is set to 1, the TEMn bit, TEHm1 bit, TEHm3 bit become 1, and the active edge detection wait state of the TIMn pin is entered. An active edge detection via the TIMn pin input starts the TCRmn register and loads the value of the Timer Data Register mn (TDRmn). The TCRmn register counts decreasingly from the value of the loaded TDRmn register by counting the clock. If TCRmn becomes “0000H”, INTTMMn is output and counting is stopped until the next active edge of the TIMn pin input is detected.

The TDRmn register can be rewritten at any time, and the rewritten TDRmn register value is valid from the next cycle.

Figure 5-58 Example of basic timing operating as delay counter

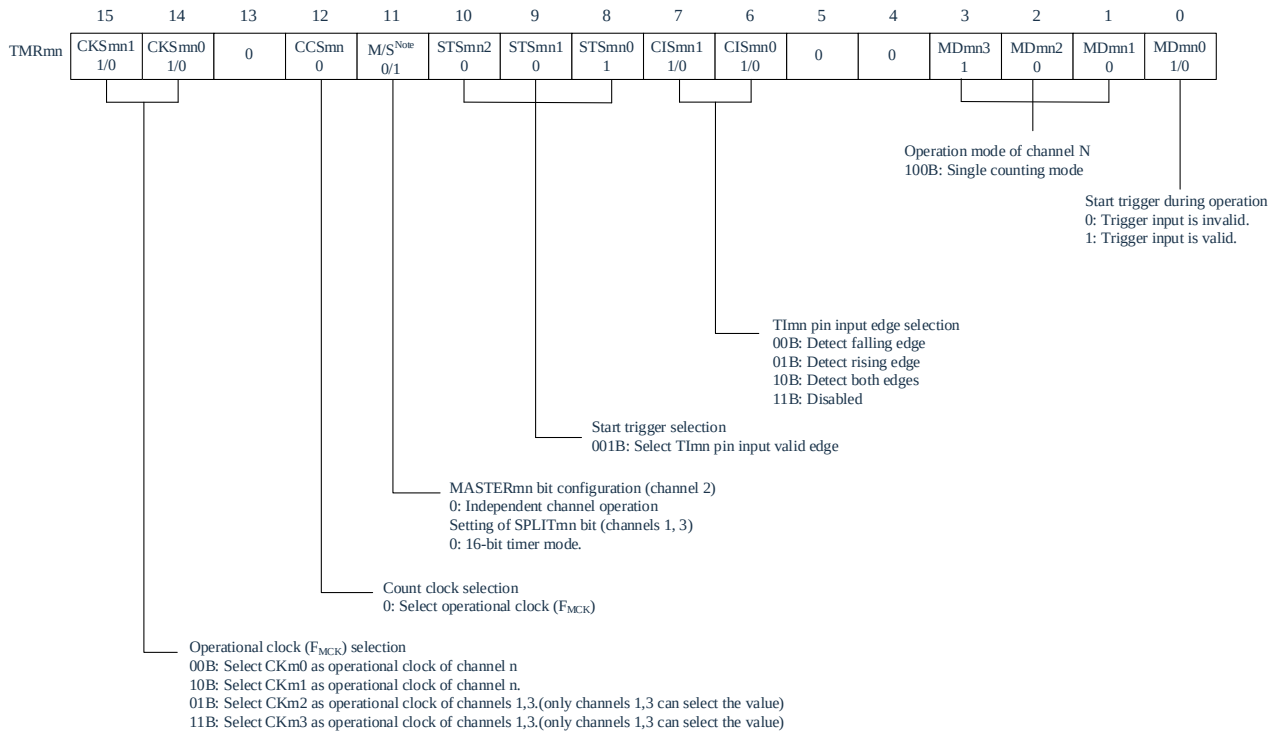


Remarks:

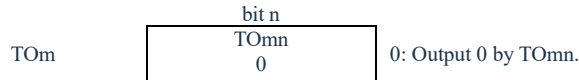
1. m: unit number (m=0, 1) n: channel number (n=0~3)
 2. TSMn: Bit n of timer channel start register m (TSm)
- TEMn: Bit n of timer channel enable status register m (TEm)
- TIMn: TIMn pin input signal
- TCRmn: Timer count register mn (TCRmn)
- TDRmn: Timer data register mn (TDRmn)

Figure 5-59 Example of register contents setting for delay counter function

(a) Timer mode register mn (TMRmn)



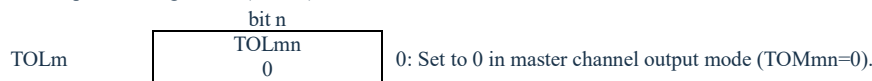
(b) Timer output register m (TOM)



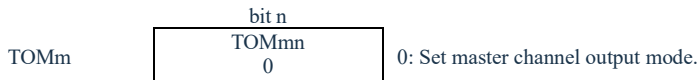
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to "0".

Remark m: unit number (m=0, 1) n: channel number (n=0~3)

Figure 5-60 Procedure for delay counter function

	Software operation	Hardware status
Timer4 initial settings		The input clock of timer unit m is in the stop supply state. (Stop providing clock, cannot write to each register)
	Set the TM4mEN bit of the peripheral enable register 0(PER0) to "1".	The input clock of timer unit m is in the providing state and the channels are in the stop state. (Start to supply clock, can write to each register)
	Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 ~ CKm3.	
Initial setting of channels	Set the corresponding bit of the Noise Filter Enable Register (NFEN1) to "0" (OFF) or "1" (ON). Set the timer mode register mn (TMRmn)(to determines the operating mode of the channel). Set the output delay time for the timer data register mn (TDRmn). Set the TOEmn bit to "0" and stop TOmn operation.	The channel is in a running stop state. (Provides clock, consumes some power)
Start Operation	Set the TSmn bit to "1". Since the TSmn bit is a trigger bit, it automatically returns to "0".	The TEmn bit turns into '1' and enter into start trigger (detect Timn pin input active edge or set TSmn bit to '1') detection waiting state.
	Start decreasing the count by detecting the next start trigger. • The active edge of the Timn pin input. • Set the TSmn bit to "1" by software.	Load the value of the TDRmn register into the Timer Count Register mn (TCRmn).
In operation	The setting of the TDRmn register can be changed at will. The TCRmn register can be read at any time. The TSRmn register is not used.	The counter (TCRmn) performs decremental counting. If TCRmn counts to "0000H", INTTMmn is generated and TCRmn is "1" until the next start trigger is detected (detecting an active edge on the Timn pin input or setting TSmn to "1"). The count is stopped when "0000H" is detected.
Stop operation	Set the TTmn bit to "1". The operation automatically returns to "0" because the TTmn bit is a trigger bit.	The TEmn bit becomes "0" and stops counting. The TCRmn register holds the count value and stops counting.
Timer4 stop	Set the TM4mEN bit of the PER0 register to "0".	The input clock of timer unit m is in the stop-providing state. Initialize all circuits and the SFR for each channel.

Restart operation

Remark m: unit number (m= 0, 1) n: channel number (n=0~3)

5.9 Multi-Channel Linkage Operation Function for Universal Timer Unit

5.9.1 Operation as Single Trigger Pulse Output Function

Using the 2 channels in pairs, a single trigger pulse with any delay pulse width can be generated from the input of the TImn pin. The delay and pulse width can be calculated using the following equation:

$$\begin{aligned}\text{Delay} &= \{\text{TDRmn (master) set value} + 2\} \times \text{counting clock period} \\ \text{Pulse width} &= \{\text{TDRmp (slave) set value}\} \times \text{counting clock period}\end{aligned}$$

In single count mode, the master channel operates and counts the delay. By detecting a start trigger, the timer count register mn (TCRmn) of the master channel starts to operate and loads the value of timer data register mn (TDRmn). The TCRmn register counts decreasingly from the value of the loaded TDRmn register by counting the clock. If TCRmn becomes “0000H”, INTTMmn is output and counting stops before the next start trigger is detected.

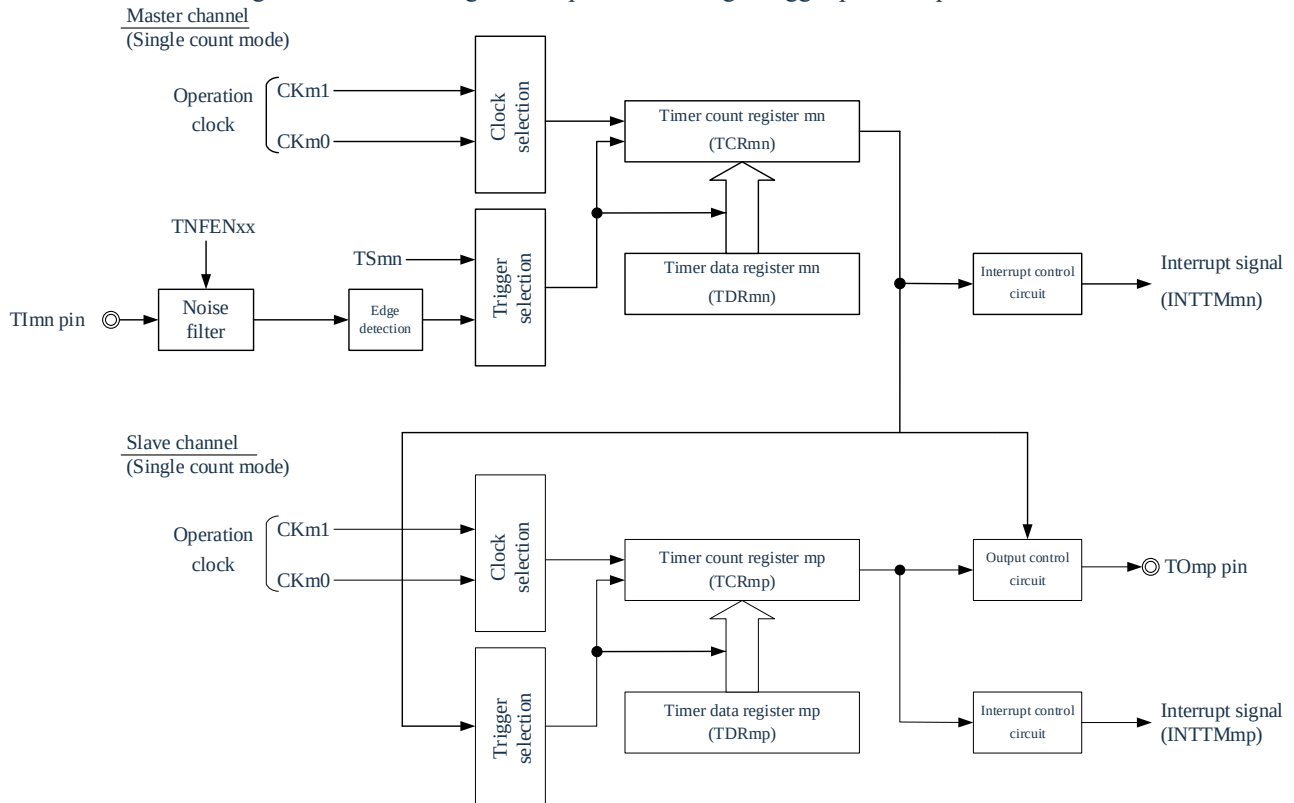
In single count mode, the slave channel operates and counts the pulse width. The INTTMmn of the master channel is used as the start trigger and the TCRmp register of the slave channel is started and loaded with the value of the TDRmp register. The TCRmp register counts decreasingly from the value of the loaded TDRmp register by counting the clock. If the count value becomes “0000H”, INTTMmp is output and counting is stopped until the next start trigger (INTTMmn of the master channel) is detected. The output level of TOmp becomes valid after INTTMmn has been generated from the master channel and after 1 count clock, if TCRmp becomes “0000H”, it becomes invalid.

The software operation (TSmn=1) can also be used as a start trigger to output a single trigger pulse without using the TImn pin input.

Caution: Because the TDRmn register of the master channel and the TDRmp register of the slave channel have different loading timings, if the TDRmn register and the TDRmp register are rewritten during counting, they may compete with the loading timings and output an abnormal waveform. The TDRmn register must be rewritten after generating INTTMmn and the TDRmp register must be rewritten after generating INTTMmp.

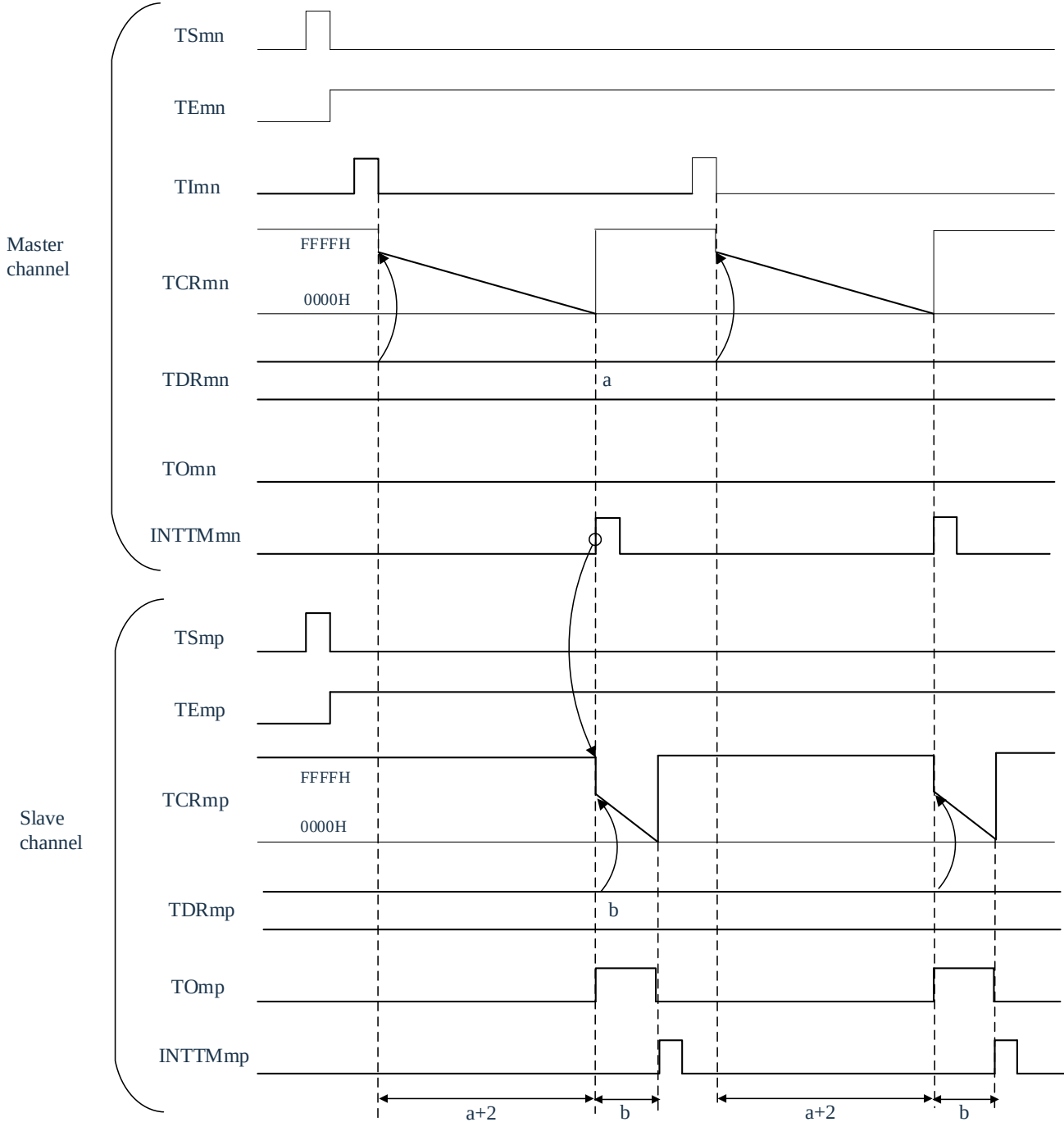
Remark: m: unit number (m=0, 1) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)

Figure 5-61 Block diagram of operation as single trigger pulse output function



Remark m: unit number (m= 0, 1) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)

Figure 5-62 Example of basic timing operating as a single trigger pulse output function



Remarks:

1. m: unit number (m= 0, 1) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)
2. TSmn, TSmp: Bits n, p of timer channel start register m (TSm)

TEmn, TEmmp: Bits n, p of timer channel enable status register m (TEm)

TImn, TImp: Input signals of TImn pin and TImp pin

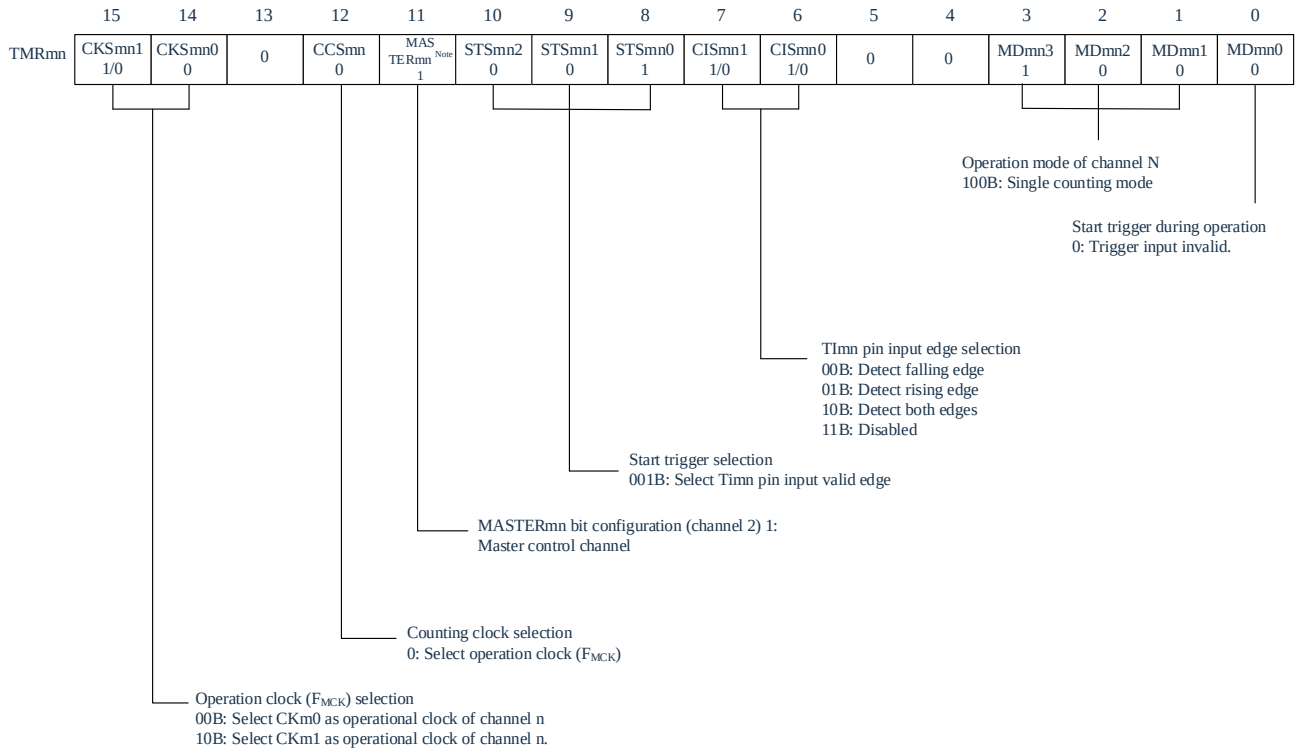
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)

TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)

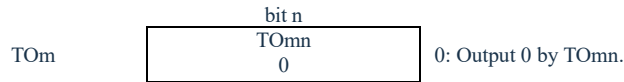
TOmn, TOmp: Output signals of TOmn pin and TOmp pin

Figure 5-63 Example of register contents setting for single trigger pulse output function (master channel)

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOM)



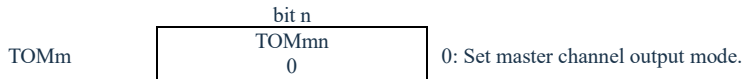
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



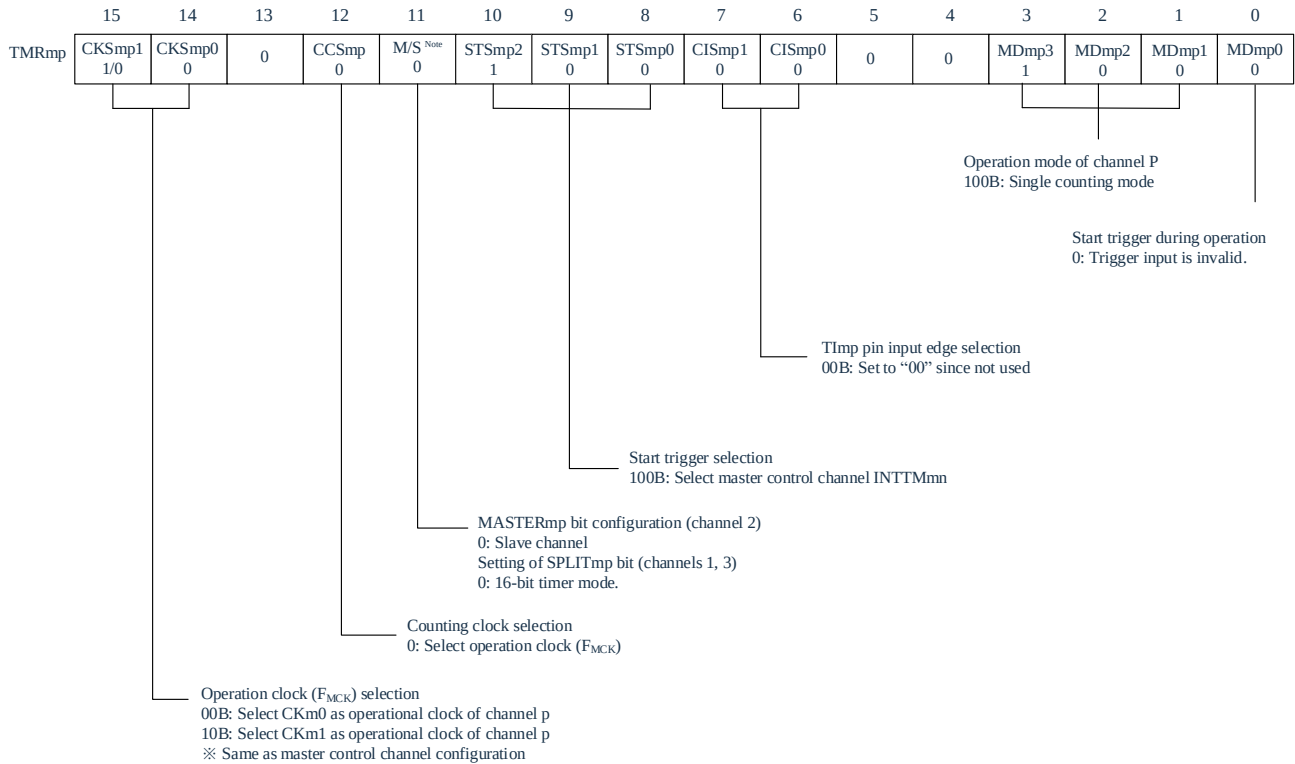
Note TMRm2: MASTERmn=1

TMRm0: Fixed to "0".

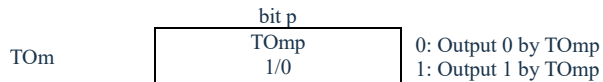
Remark m: unit number (m= 0, 1) n: master channel number (n=0, 2)

Figure 5-64 Example of register contents setting for single trigger pulse output function (slave channel)

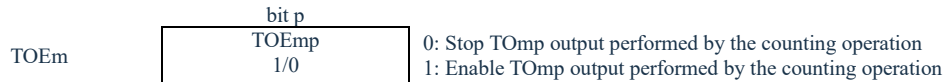
(a) Timer mode register mp (TMRmp)



(b) Timer output register m (TOm)



(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note: TMRm2: MASTERmp bit

TMRm1, TMRm3: SPLITmp bit

Remark: m: unit number (m= 0, 1) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)

Figure 5-65 Procedure for single trigger pulse output function (1/2)

	Software operation	Hardware status
Timer4 initial settings		The input clock of timer unit m is in the stop supply state. (Stop to supply clock, cannot write to each register)
	Set the TM4mEN bit of the peripheral enable register 0(PER0) to "1".	The input clock of timer unit m is in the providing state and the channels are in the stop state. (Start to supply clock, can write to each register)
	Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 ~ CKm3.	
Initial setting of channels	Set the corresponding bit of the Noise Filter Enable Register (NFEN1) to "1". Set the timer mode registers mn and mp (TMRmn, TMRmp) for the 2 channels used (to determine the operation mode of the channel). Set the output delay time for the timer data register mn (TDRmn) of the master channel, and set the pulse width for the TDRmp register of the slave channel.	The channel is in the stop state. (Provides clock, and consumes some power)
	Slave channel setting Set TOMmp bit of the timer output mode register m (TOM m) to "1" (slave channel output mode). Set the TOLmp bit. Set the TOmp bit to determine the initial level of the TOmp output. Set the TOEmp bit to "1" and enable TOmp output. Set the Port Register and Port Mode Register to "0".	The TOmp pin is in Hi-Z output state. When the port mode register is in output mode and the port register is "0", the TOmp initial set level is output. The TOmp remains unchanged because the channel is in the stop state. The TOmp pin outputs the level set by the TOmp.

Figure 5-66 Procedure for single trigger pulse output function(2/2)

	Software operation	Hardware status
Start operation	Set the TOEmp bit (slave) to "1" (restart operation only). Set the TSmn (master) and TSmp (slave) bits of the Timer Channel Start Register m (TSm) to "1" at the same time. —→ Since the TSmn bit and the TSmp bit are trigger bits, they automatically return to "0".	The TEmn and TEm bits are set to 1 and the master channel enters the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1) wait status. Counter stops operating.
	Count operation of the master channel is started by start trigger detection of the master channel • Detects the TImn pin input valid edge • Sets the TSmn bit of the master channel to 1 by software Note.	Master channel starts counting.
In operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOM and TOEm registers by slave channel can be changed.	Master channel loads the value of the TDRmn register to timer count register mn (TCRmn) by the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1), and the counter starts counting down. When the count value reaches TCRmn = 0000H, the INTTMmn output is generated, and stops counting until the next TImn pin input. The slave channel, triggered by INTTMmn of the master channel, loads the value of the TDRmp register to the TCRmp register, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Stop operation	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. —→ The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEm = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit. —→	The TOmp pin outputs the TOmp set level.
Timer4 stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. When holding the TOmp pin output level is not necessary: Setting not required.	The TOmp pin output level is held by port function.
	The TM4mEN bit of the PER0 register is cleared to 0. —→	The input clock of timer unit m is in the stop-providing state. Initialize all circuits and the SFR of each channel.

Note: The TSmn bit of the slave channel cannot be set to 1.

Remark: m: unit number (m= 0) n: master channel number (n=0, 2)

p: slave channel number (n=0: 1, 2, 3 n=2: p=3)

5.9.2 Operation as PWM Function

By using the 2 channels in pairs, pulses of any period and duty cycle can be generated. The period and duty cycle of the output pulses can be calculated using the following equations:

$$\begin{aligned} \text{Pulse period} &= \{\text{TDRmn (master) set value} + 1\} \times \text{counting clock period} \\ \text{Duty cycle [\%]} &= \{\text{TDRmp (slave) set value}\} / \{\text{TDRmn (master) set value} + 1\} \times 100 \\ 0\% \text{ output: } &\text{TDRmp (slave) set value} = 0000\text{H} \\ 100\% \text{ output: } &\text{TDRmp (slave) set value} \geq \{\text{TDRmn (master) set value} + 1\} \end{aligned}$$

Remark: When the set value of TDRmp (slave) > {Set value of TDRmn (master) + 1}, the duty cycle exceeds 100% but is 100% output.

The master channel is used as the interval timer mode. If the channel start trigger bit (TSmn) of the timer channel start register m (TSM) is set to 1, an interrupt (INTTMmn) is output, and then the set value of the timer data register mn (TDRmn) is loaded into the timer count register mn (TCRmn), and the count is decremented by the count clock. When the count reaches "0000H", the value of the TDRmn register is loaded into the TCRmn register again after the INTTMmn is output, and the count is decremented. Thereafter, this operation is repeated before setting the channel stop trigger bit (TTmn) of the timer channel stop register m (TTM) to 1.

When used as PWM function, the master channel decrements the count and the period until "0000H" is counted as the PWM output (TOmp) period. The slave channel is used in single count mode. The value of TDRmp register is loaded into TCRmp register with INTTMmn of the master channel as the start trigger, and the count is decremented until "0000H". When the count reaches "0000H", INTTMmp is output and the next start trigger (INTTMmn of the master channel) is waited.

When used as PWM function, the slave channel decrements the count and the duty cycle of the PWM output (TOmp) for the period until "0000H" is counted.

After INTTMmn is generated from the master channel and 1 clock has elapsed, the PWM output (TOmp) becomes active and it becomes invalid when the value of TCRmp register of the slave channel is "0000H".

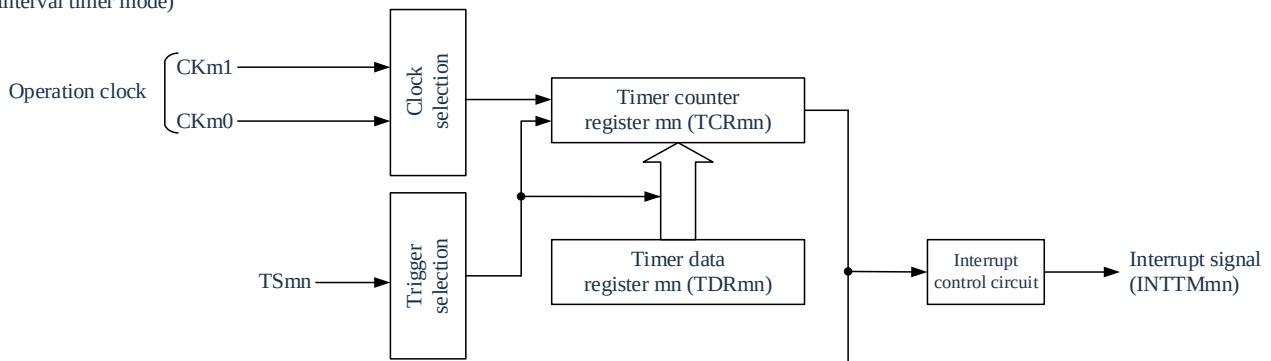
Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel, a write access is necessary two times. The timing at which the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers is upon occurrence of INTTMmn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMmn of the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.

Remark: m: unit number (m= 0, 1) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)

Figure 5-67 Block diagram of operation as PWM function

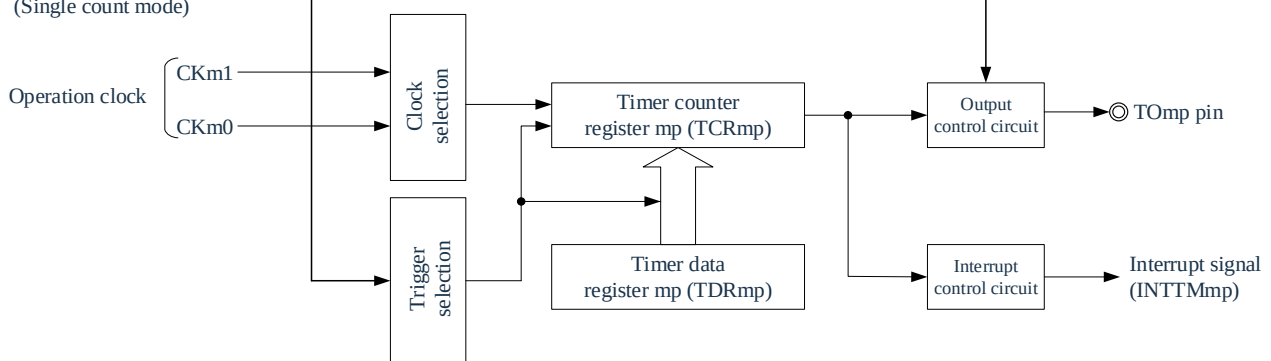
Master channel

(Interval timer mode)



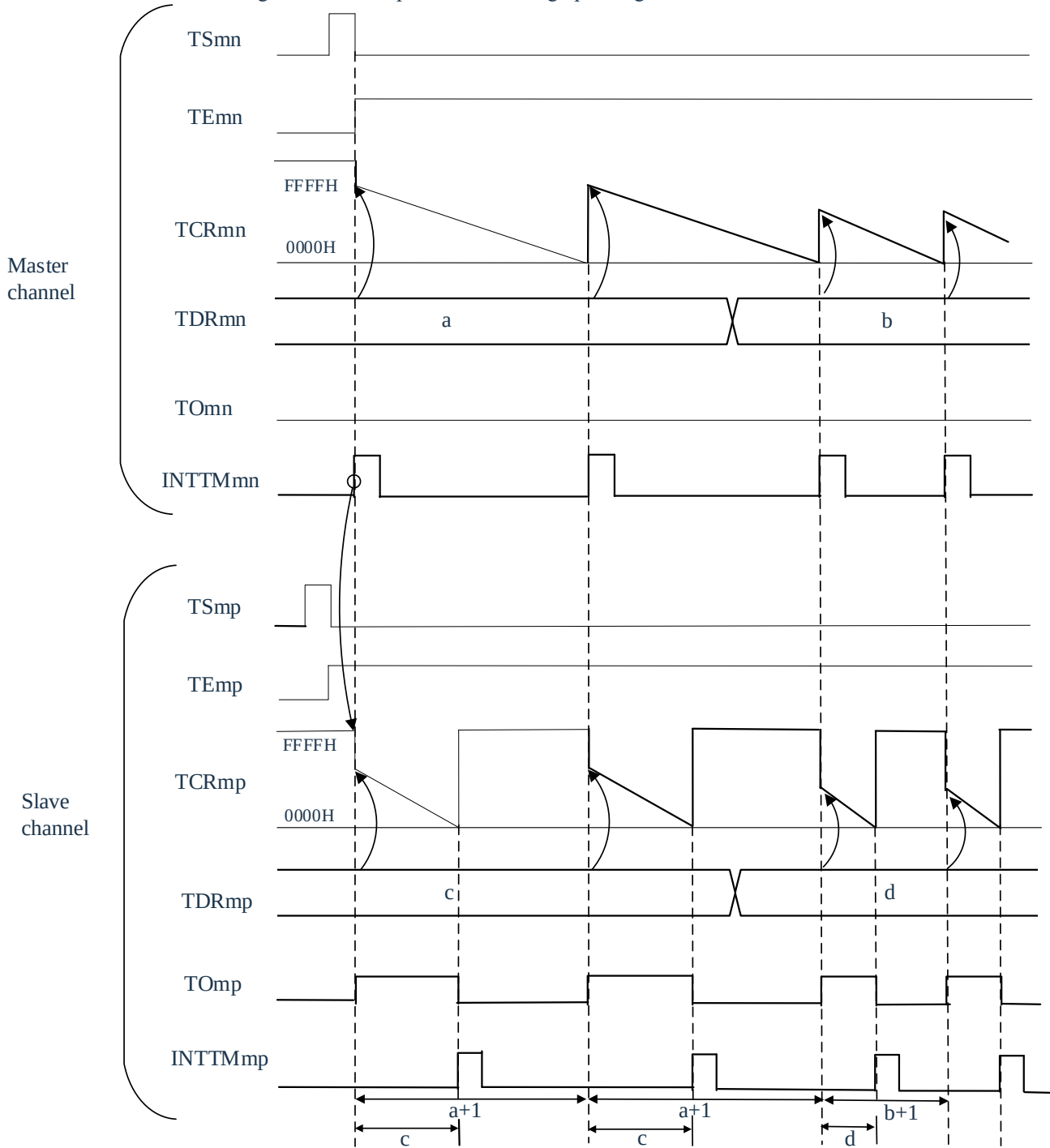
Slave channel

(Single count mode)



Remark m: unit number (m= 0, 1) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)

Figure 5-68 Example of basic timing operating as PWM function



Remarks:

1. m: unit number (m= 0, 1) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)
2. TSmn, TSmp: Bits n, p of timer channel start register m (TSm)

TEmn, TEmP: Bits n, p of timer channel enable status register m (TEm)

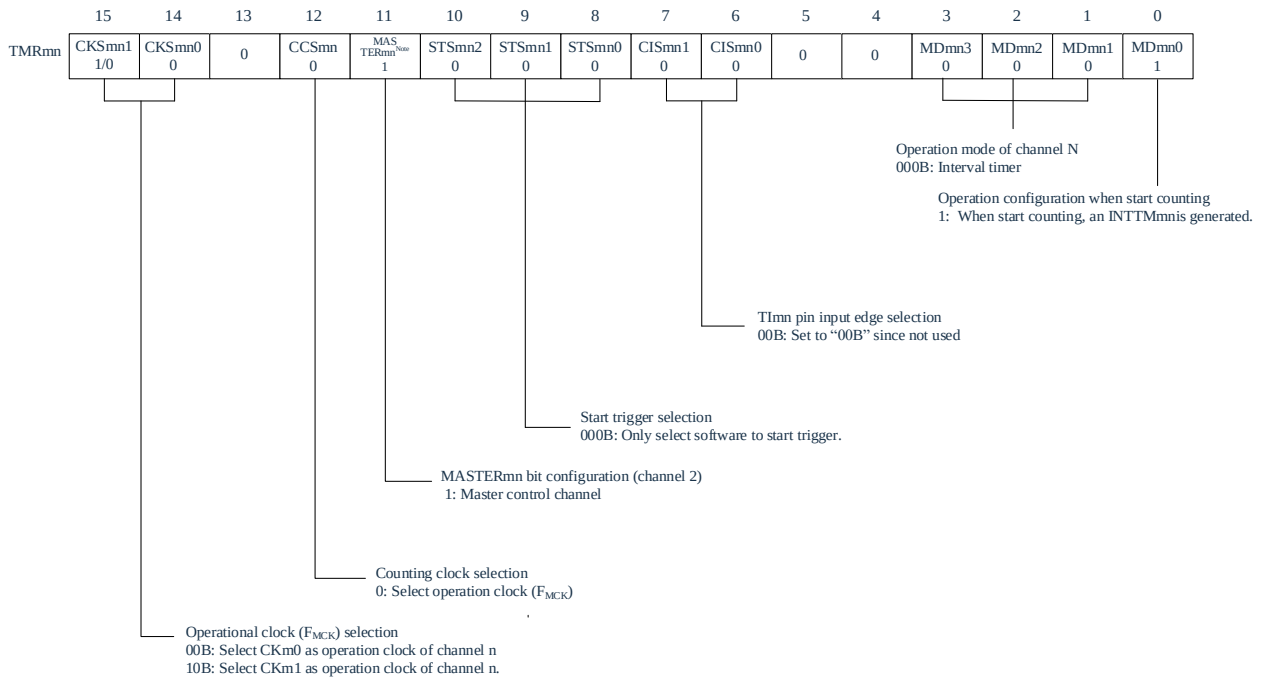
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)

TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)

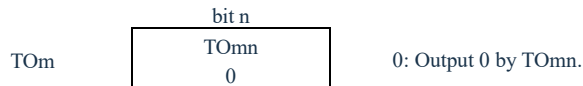
TOMn, TOMP: Output signals of TOMn pin and TOMP pin

Figure 5-69 Example of register contents setting for PWM function (master channel)

(a) Timer mode register mn (TMRmn)



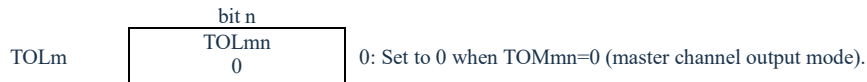
(b) Timer output register m (TOM)



(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



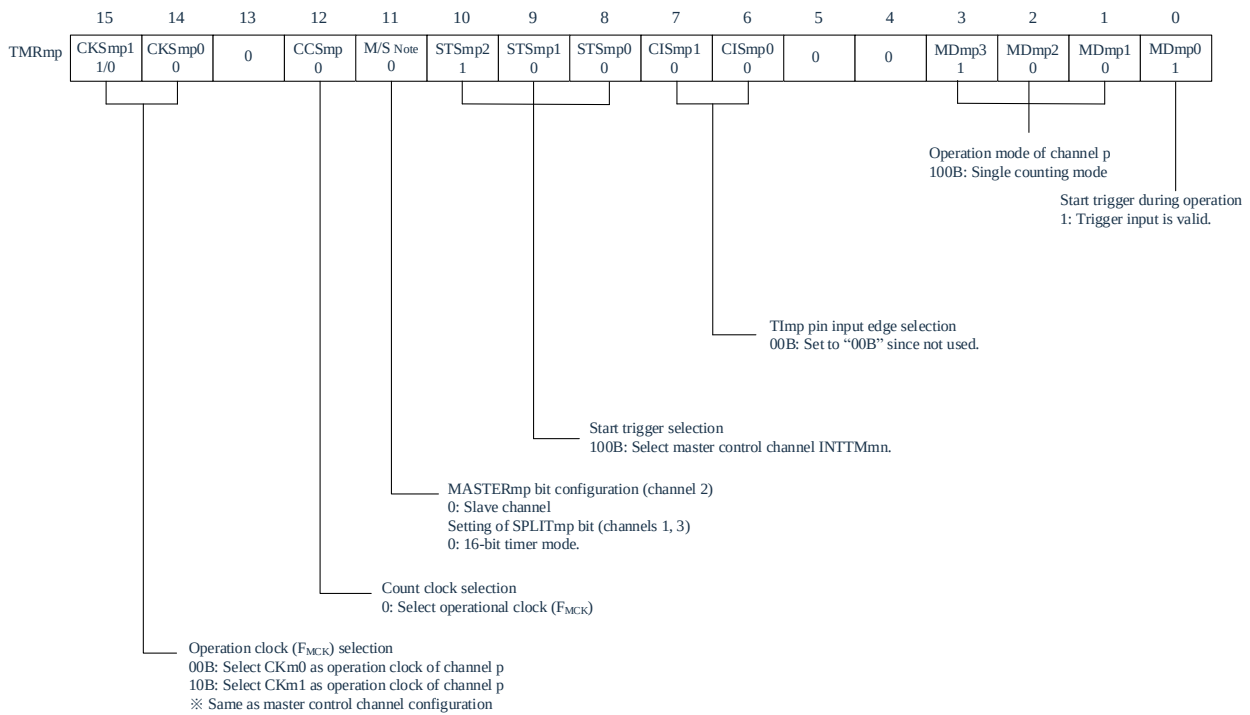
Note: TMRm2: MASTERmn=1

TMRm0: Fixed to "0".

Remark: m: unit number (m=0, 1) n: master channel number (n=0, 2)

Figure 5-70 Example of register contents setting for PWM function (slave channel)

(a) Timer mode register mp (TMRmp)



(b) Timer output register m (TOM)

bit p	
TOM	TOMp 1/0
	0: Output 0 by TOMp. 1: Output 1 by TOMp.

(c) Timer output enable register m (TOEm)

bit p	
TOEm	TOEmp 1/0
	0: Stop TOMp output performed by the counting operation. 1: Enable TOMp output performed by the counting operation.

(d) Timer output level register m (TOLm)

bit p	
TOLm	TOLmp 1/0
	0: Positive logic output (active-high) 1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

bit p	
TOMm	TOMmp 1
	1: Set the slave channel output mode.

Note: TMRm2: MASTERmp bit

TMRm1, TMRm3: SPLITmp bits

Remark: m: unit number (m= 0, 1) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)

Figure 5-71 Procedure for PWM function (1/2)

	Software operation	Hardware status
Timer4 initial settings		The input clock of timer unit m is in the stop supply state. (Stop to supply clock, cannot write to each register)
	Set the TM4mEN bit of the peripheral enable register 0(PER0) to "1".	The input clock of timer unit m is in the providing state and the channels are in the stop state. (Start to supply clock, can write to each register)
	Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 ~ CKm3.	
Initial setting of channels	Set the timer mode registers mn and mp (TMRmn, TMRmp) for the 2 channels used (to determine the operation mode of the channel). Set the interval (period) value for the timer data register mn (TDRmn) for the master channel and the duty cycle value for the TDRmp register for the slave channel.	The channel is in the stop state. (Provides clock, and consumes some power)
	Slave channel setting Set TOMmp bit of the timer output mode register m (TOMm) to "1" (slave channel output mode). Set the TOLmp bit. Set the TOmp bit to determine the initial level of the TOmp output. Set the TOEmp bit to "1" and enable TOmp output. Set the Port Register and Port Mode Register to "0".	The TOmp pin is in Hi-Z output state. When the port mode register is in output mode and the port register is "0", the initially set level of TOmp is output. The TOmp remains unchanged because the channel is in the stop state. The TOmp pin outputs the level set by the TOmp.

Figure 5-72 Procedure for PWM function (2/2)

	Software operation	Hardware status
Restart operation	Start operation Set the TOEmp bit to "1" (only limited to restart operation). Set both the TSmn bit (master) and TSmp bit (slave) of the timer channel start register m (TSM) to "1". The operation automatically returns to "0" because the TSmn and TSmp bits are trigger bits.	The TEMn and TEm bits become "1". The master channel starts counting and generates INTTMmn. With this as a trigger, the slave channel also starts counting.
	In operation The setting values of the TMRmn and TMRmp registers and the TOMmn bit, TOMmp bit, TOLmn bit, and TOLmp bit cannot be changed. Able to change the setting value of the TDRmn register and the TDRmp register after the master channel has generated INTTMmn. The TCRmn and TCRmp registers can be read at any time. The TSRmn and TSRmp registers are not used.	The master channel loads the value of the TDRmn register into the timer count register mn (TCRmn) and perform decremental counting. If TCRmn counts till "0000H", then generating INTTMmn. At the same time, load the TDRmn register value into the TCRmn register and restart decremental counting. The slave channel use INTTMmn of master channel as a trigger, load the TDRmp register value into the TCRmp register and counter start decremental counting. After INTTMmn is output from the master channel and one count clock has elapsed, the output level of TOmp is set to an active level. Then, if TCRmp counts to "0000H", it stops counting after setting the output level of TOmp to an invalid level. Thereafter, repeat this operation.
	Stop operation Set the TTmn bit (master) and TTmp bit (slave) to "1" at the same time. The operation automatically returns to "0" because the TTmn and TTmp bits are trigger bits.	TEMn, TEm = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current
	Set the TOEmp bit of slave channel to "0" and set the value for the TOmp bit.	The TOmp pin outputs the TOmp set level.
Timer4 stop	To maintain the output level of the TOmp pin: Set TOmp bit to "0" after setting the value to be held for the port register. When holding the TOmp pin output level is not necessary: No need to set.	The TOmp pin output level is held by port function.
	Set the TM4mEN bit of the PER0 register to "0".	The input clock of timer unit m is in the stop-providing state. Initialize all circuits and the SFR for each channel. (TOMn bit becomes "0" and TOmp pin becomes port function)

Remark: m: unit number (m= 0) n: master channel number (n=0, 2)

p: slave channel number (n=0: p= 1, 2, 3 n=2: p=3)

5.9.3 Operation as Multiple PWM Output Function

This is a function that extends the PWM function and uses multiple slave channels for multiple PWM outputs with different duty cycles. For example, when using 2 slave channels in pairs, the period and duty cycle of the output pulse can be calculated by using the following equation:

$$\begin{aligned} \text{Pulse period} &= \{\text{TDRmn}(\text{master}) \text{ set value} + 1\} \times \text{count clock period} \\ \text{Duty cycle1}[\%] &= \{\text{TDRmp}(\text{slave 1}) \text{ set value}\} / \{\text{TDRmn}(\text{master}) \text{ set value} + 1\} \times 100 \\ \text{Duty cycle2}[\%] &= \{\text{TDRmq}(\text{slave 2}) \text{ set value}\} / \{\text{TDRmn}(\text{master}) \text{ set value} + 1\} \times 100 \end{aligned}$$

Remark: When the set value of TDRmp (slave 1) > {the set value of TDRmn (master) + 1} or {the set value of TDRmq (slave 2)} > {the set value of TDRmn (master) + 1}, the duty cycle exceeds 100%, but is 100% output.

In interval timer mode, the timer count register mn (TCRmn) of the master channel operates and counts the period. In single count mode, the TCRmp register of slave channel 1 operates and counts the duty cycle and outputs the PWM waveform from the TOmp pin. The TCRmp register loads the value of timer data register mp (TDRmp), using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmp = “0000H”, the TCRmp outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmp becomes valid after INTTMmn has been generated from the master channel and after 1 count clock, if TCRmp becomes “0000H”, it becomes invalid.

In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in single count mode, counts the duty cycle, and outputs a PWM waveform from the TOMq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmq = “0000H”, the TCRmq register outputs INTTMmq and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of the TOMq becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmq = 0000H.

When channel 0 is used as the master channel as above, up to 3 types of PWM signals can be output at the same time.

Caution To rewrite the timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel 1 at the same time, at least 2 write accesses are required. Because the values of TDRmn register and TDRmp register are loaded into the TCRmn register and TCRmp register when the master channel generates INTTMmn, the TOmp pin cannot output the expected waveform if rewriting is performed before and after the master channel generates INTTMmn respectively. Therefore, to rewrite both the master TDRmn register and the slave TDRmp register, these two registers must be rewritten immediately after the master channel generates INTTMmn (the same applies to the TDRmq register of slave channel 2).

Remark: m: unit number (m= 0, 1) n: master channel number (n=0)
p: slave channel number q: slave channel number
 $n < p < q \leq 3$ (p and q are integers greater than n)

Figure 5-73 Block diagram of operation as multiple PWM output function (output two types of PWMs)

Master channel

(Interval timer mode)

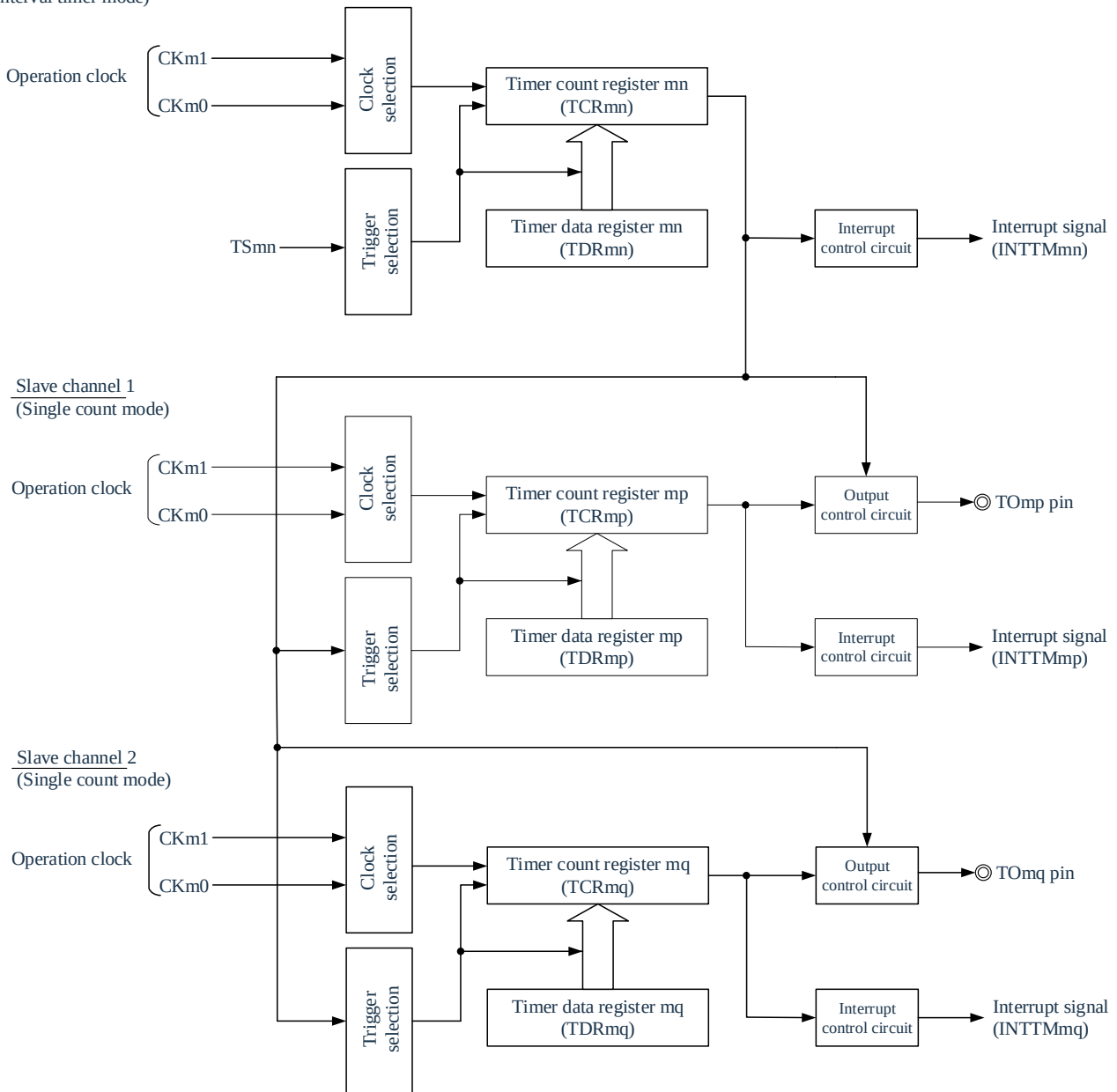
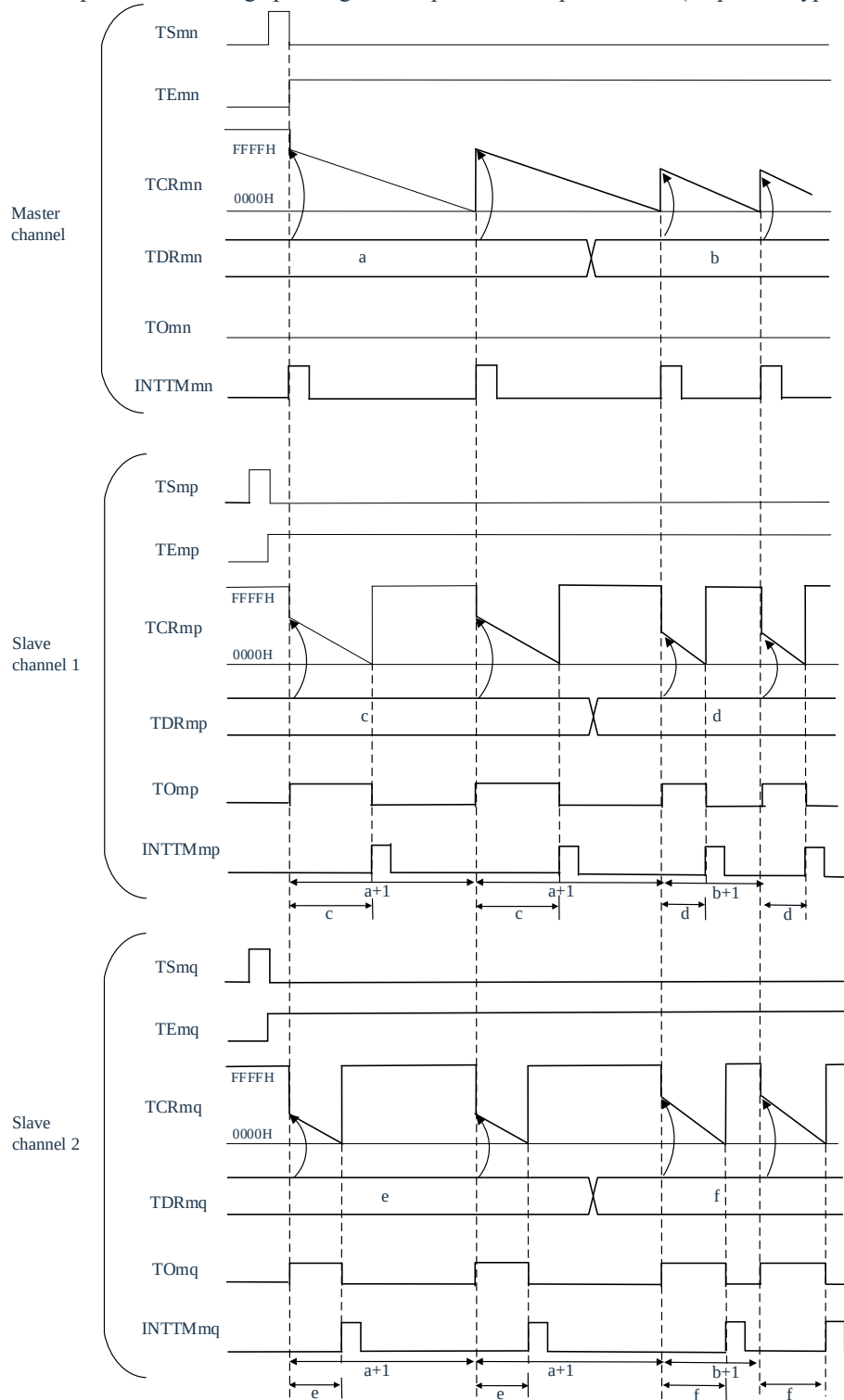


Figure 5-74 Example of basic timing operating as multiple PWM output function (output two types of PWMs)

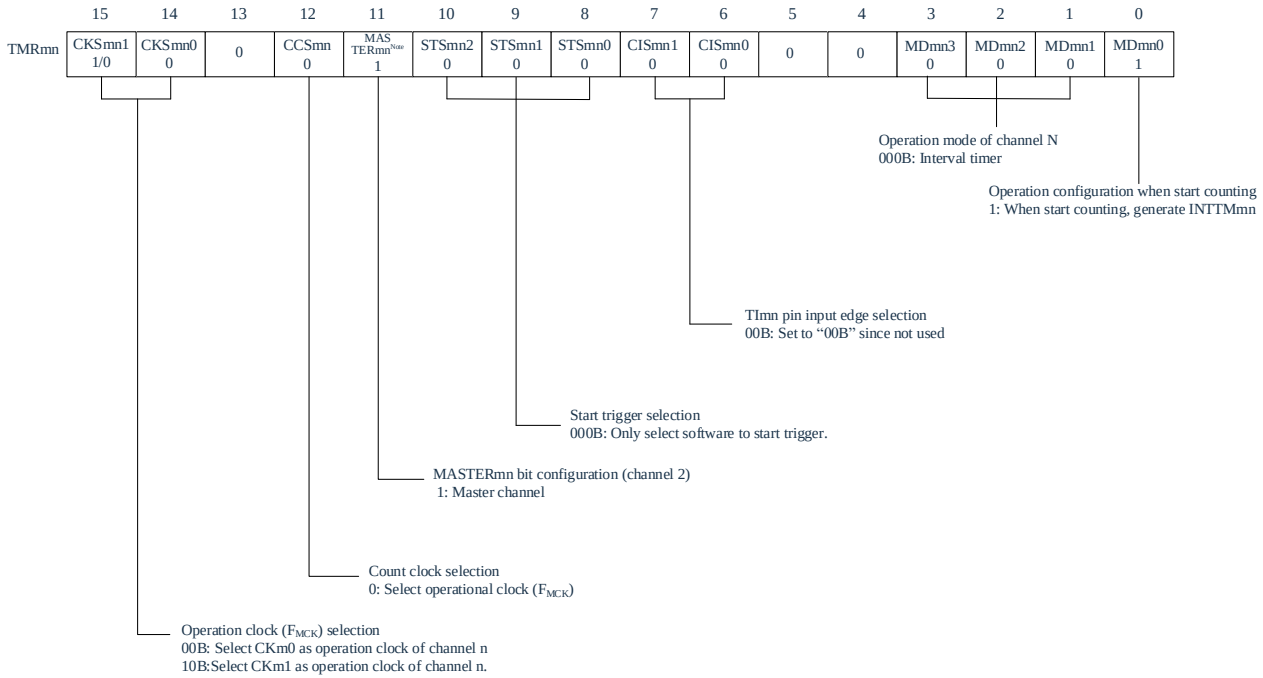


Remarks:

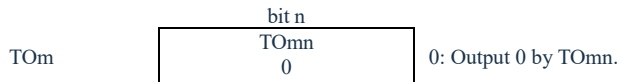
1. m: unit number (m= 0, 1) n: master channel number (n=0)
p: slave channel number q: slave channel number $n < p < q \leq 3$ (p and q are integers greater than n)
2. TSmn, TSmp, TSmq: Bits n, p and q of timer channel start register m (TSm)
TEmn, TEmmp, TEmmq: Bits n, p and q of timer channel enable status register m (TEm)
TCRmn, TCRmp, TCRmq: Timer count registers mn, mp, mq (TCRmn, TCRmp, TCRmq)
TDRmn, TDRmp, TDRmq: Timer data registers mn, mp, mq (TDRmn, TDRmp, TDRmq)
TOmn, TOmp, TOmq: TOmn, TOmp, TOmq pin output signals

Figure 5-75 Example of register contents setting for multiple PWM output function (master channel)

(a) Timer mode register mn (TMRmn)



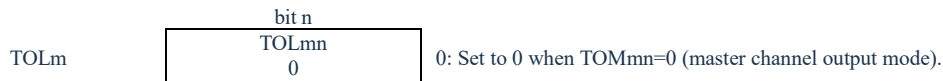
(b) Timer output register m (TOm)



(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



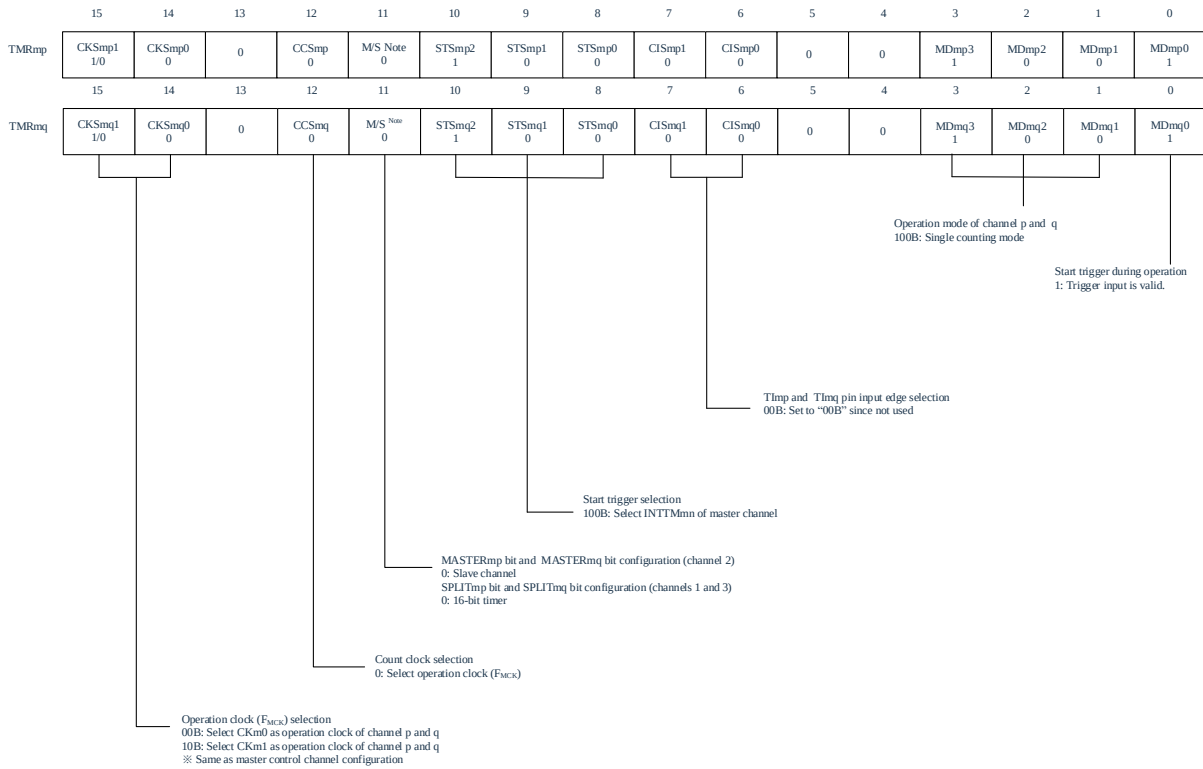
Note: TMRm2: MASTERmn=1

TMRm0: Fixed to "0".

Remark: m: unit number (m= 0, 1) n: master channel number (n=0)

Figure 5-76 Example of register contents setting for multiple PWM output function (slave channel) (output two types of PWMs)

(a) Timer mode registers mp, mq (TMRmp, TMRmq)



(b) Timer output register m (TOM)

	bit q	bit p	
TOM	TOmq 1/0	TOmp 1/0	0: Output 0 by TOmp or TOMq. 1: Output 0 by TOmp or TOMq.

(c) Timer output enable register m (TOEm)

	bit q	bit p	
TOEm	TOEmq 1/0	TOEmp 1/0	0: Stop the TOmp or TOMq output operation by counting operation. 1: Enable the TOmp or TOMq output operation by counting operation.

(d) Timer output level register m (TOLm)

	bit q	bit p	
TOLm	TOELq 1/0	TOELp 1/0	0: Positive logic output (active-high) 1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	bit q	bit p	
TOMm	TOMLq 1	TOMLp 1	1: Set the slave channel output mode.

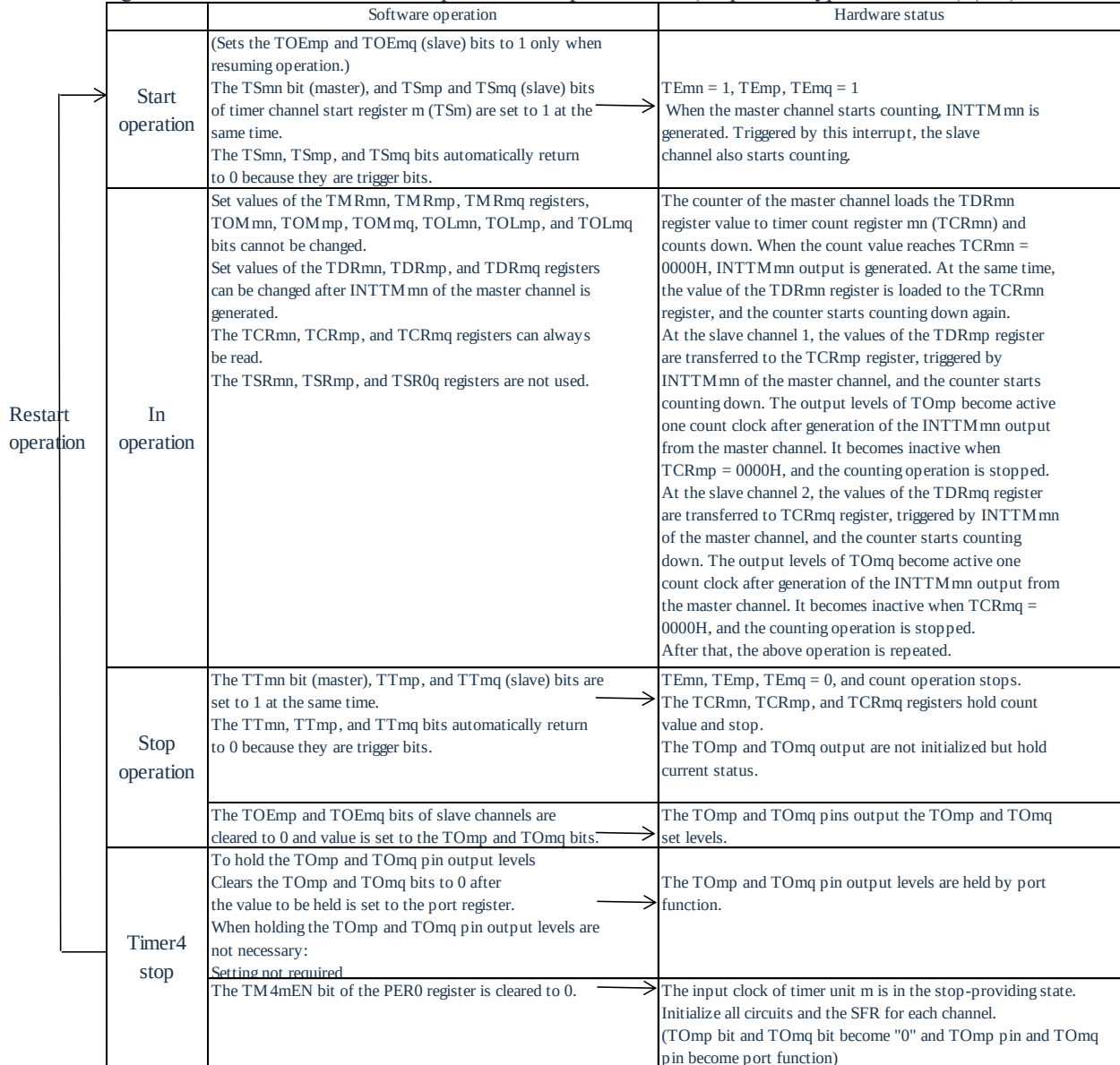
Note TMRm2: MASTERmp bit, MASTERmq bit
TMRm1, TMRm3: SPLITmp bit, SPLITmq bit

Remark m: unit number (m= 0, 1) n: master channel number (n=0)
p: Slave channel number q: Slave channel number
 $n < p < q \leq 3$ (p and q are integers greater than n)

Figure 5-77 Procedure for the multiple PWM output function (output two types of PWMs) (1/2)

	Software operation	Hardware status
Timer4 initial settings		The input clock of timer unit m is in the stop supply state. (Stop to supply clock, cannot write to each register)
	Set the TM4mEN bit of the peripheral enable register 0(PER0) to "1".	The input clock of timer unit m is in the providing state and the channels are in the stop state. (Start to supply clock, can write to each register)
	Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 and CKm1.	
Initial setting of channels	Set the timer mode registers mn, mp, (TMRmn, TMRmp,) for each channel used (to determine the channel operation mode). Set the interval (period) value for the master channel's timer data register mn (TDRmn), and set the duty cycle value for the slave channel's TDRmp register and TDRmq register.	The channel is in the stop state. (Provides clock, and consumes some power)
	Slave channel setting Set TOMmp and TOMmq bits of the timer output mode register m (TOMm) to "1" (slave channel output mode). Set the TOLmp and TOLmq bits to "0". Set the TOmp and TOmq bits and determine the initial output level of the TOmp and TOmq bits.	The TOmp pin is in Hi-Z output state. When the port mode register is in output mode and the port register is "0", the TOmp and TOmq initial set levels are output. The TOmp and TOmq remains unchanged because the channel is in the stop state.
	Set the TOEmp amd TOEmq bits to "1" and enable TOmp and TOmq output.	The TOmp pin and TOmq pin output the levels set by the TOmp and TOmq.
	Set the Port Register and Port Mode Register to "0".	

Figure 5-78 Procedure for the multiple PWM output function (output two types of PWMs) (2/2)



Remark: m: unit number (m= 0, 1) n: master channel number (n=0)

p: slave channel number q: slave channel number

$n < p < q \leq 3$ (p and q are integers greater than n)

Chapter 6 Real-Time Clock

6.1 Functions of Real-Time Clock

The real-time clock has the following functions.

- Holds counters for years, months, weeks, days, hours, minutes, and seconds up to a maximum of 99 years.
- Fixed cycle break (cycles: 0.5 seconds, 1 second, 1 minute, 1 hour, 1 day, 1 month)
- Alarm clock interrupt (alarm clock: week, hour, minute)
- 1Hz pin out capability

6.2 Structure of Real-Time Clock

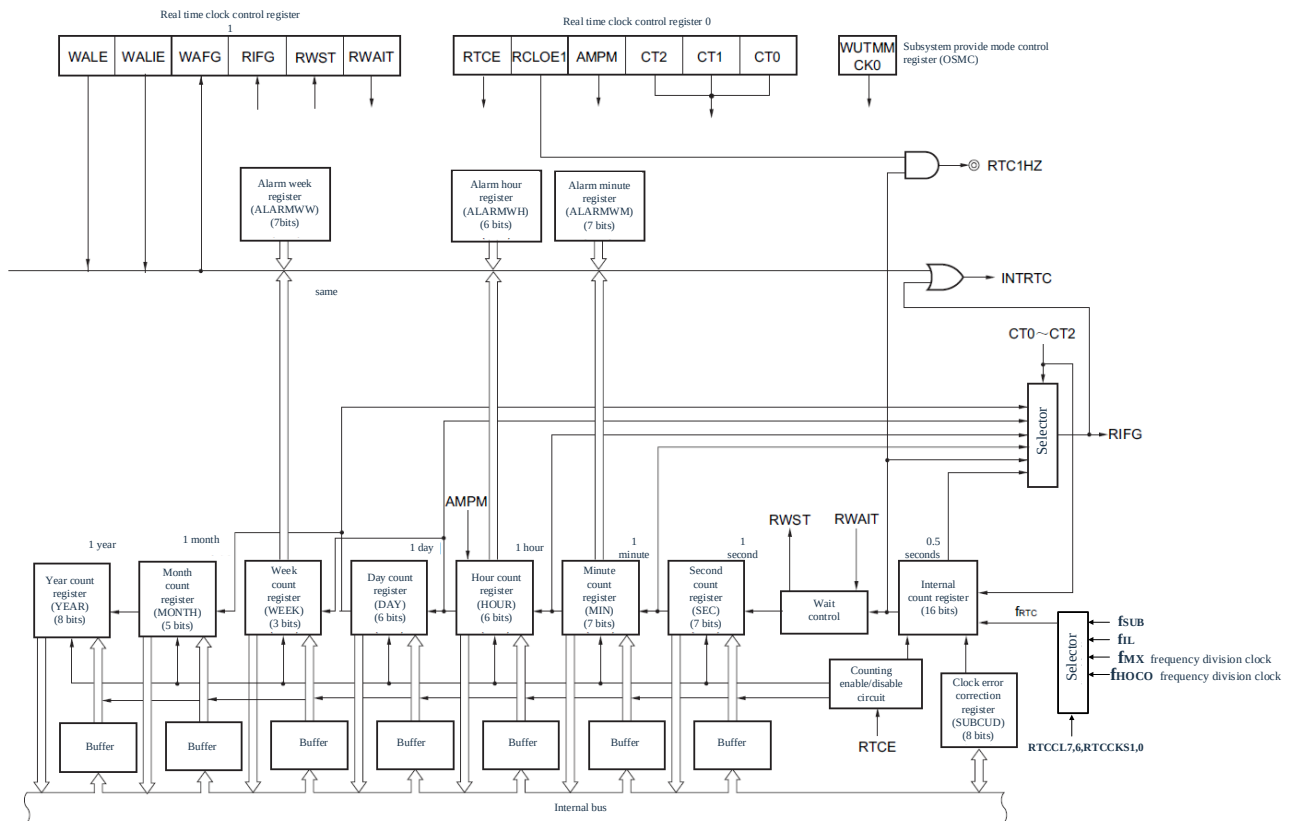
The real-time clock consists of the following hardware.

Table 6-1 Structure of real-time clock

Item	Structure
Counter	Internal counter (16-bit)
Control register	Peripheral enable register 0 (PER0.bit7)
	Real-time clock selection register (RTCCL)
	Real-time clock control register 0 (RTCC0)
	Real-time clock control register 1 (RTCC1)
	Second count register (SEC)
	Minute count register (MIN)
	Hour count register (HOUR)
	Day count register (DAY)
	Week count register (WEEK)
	Month count register (MONTH)
	Year count register (YEAR)
	Clock error correction register (SUBCUD)
	Alarm clock minute register (ALARMWM)
	Alarm clock hour register (ALARMWH)
	Alarm clock week register (ALARMWW)

Note: The reset of the above RTC control registers are only controlled by the POR reset.

Figure 6-1 Block diagram of real-time clock



Caution Count years, months, weeks, days, hours, minutes and seconds only if you select $F_{\text{mx}}/F_{\text{hoco}}$ clock ($\approx 32,768\text{KHz}$ after every week) or the secondary system clock ($F_{\text{SUB}}=32.768\text{kHz}$) as the running clock for the real-time clock. When a low-speed internal oscillator clock ($F_{\text{IL}}=30\text{kHz}$) is selected, only a fixed cycle interrupt function is used.

The fixed cycle interrupt interval when selecting F_{IL} is calculated using the following equation:

$$\text{Fixed period (value selected by the RTCC0 register)} \times F_{\text{SUB}}/F_{\text{IL}}$$

6.3 Registers for Controlling Real-Time Clock

The real-time clock is controlled through the following registers.

- Peripheral enable register 0 (PER0)
- Real-time clock selection register (RTCCL)
- Real-time clock control register 0 (RTCC0)
- Real-time clock control register 1 (RTCC1)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Clock error correction register (SUBCUD)
- Alarm clock minute register (ALARMWM)
- Alarm clock hour register (ALARMWH)
- Alarm clock week register (ALARMWW)
- Port mode register (PMxx)
- Port mode control register (PMCxx)
- Port alternate configuration register (PxxCFG)

6.3.1 Peripheral Enable Register 0 (PER0)

The PER0 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.

The bit7 (RTCEN) must be set to “1” when using real-time clocks. The PER0 register is set by an 8-bit memory manipulation instruction. After the reset signal is generated, the value of this register changes to “00H”.

Figure 6-2 Format of peripheral enable register 0 (PER0)

Address: 0x40020420	After reset: 00H	R/W						
Symbol	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TM41EN	TM40EN
RTCEN	Control of an input clock of a real-time clock (RTC) and a 15-bit interval timer							
0	Stop to supply the input clock. • SFR used by the real-time clock (RTC) and the 15-bit interval timer cannot be written. • The real-time clock (RTC) and the 15-bit interval timer are in the reset status.							
1	Supply the input clock. • SFR used by the real-time clock (RTC) and the 15-bit interval timer can be read and written.							

Cautions:

- If you want to use the real-time clock, you must first set the RTCEN bit to 1 while the counting clock (F_{RTC}) oscillation is stable, and then set the following registers. When the RTCEN bit is 0, the write operation of the real-time clock control register is ignored, and the read values are initial (except RTCCL, port mode register, and port register).

 - Real-time clock control register 0 (RTCC0)
 - Real-time clock control register 1 (RTCC1)
 - Second count register (SEC)
 - Minute count register (MIN)
 - Hour count register (HOUR)
 - Day count register (DAY)
 - Week count register (WEEK)
 - Month count register (MONTH)
 - Year count register (YEAR)
 - Clock error correction register (SUBCUD)
 - Alarm clock minute register (ALARMWM)
 - Alarm clock hour register (ALARMWH)
 - Alarm clock week register (ALARMWW)
- By setting the RTCLPC bit in the Subsystem Clock Supply Mode Control Register (OSMC) to 1, the subsystem clock can be stopped for peripheral functions other than the real-time clock and 15-bit interval timer in deep sleep mode or sleep mode running with the subsystem clock.

6.3.2 Real-Time Clock Selection Register (RTCCL)

A real-time clock and a count clock of a 15-bit interval timer (F_{RTC}) can be selected through RTCCL.

Figure 6-3 Format of real-time clock selection register (RTCCL)

Address: 0x4004047C

After reset: 00H

R/W

Symbol

7

6

5

4

3

2

1

0

RTCCL

RTCCL7	RTCCL6	0	0	0	0	RTCKS1	RTCKS0

RTCCL7	Selection of real-time clock, 15-bit interval timer count clock source
0	Select high-speed system clock (F_{MX})
1	Select high-speed on-chip oscillator (F_{HOCO})

RTCKS1	RTCKS0	RTCCL6	Selection of operation clock for real time clock, count clock of 15-bit interval timer
0	0	0/1	Subsystem clock (F_{SUB})
0	1		Low-speed internal oscillator clock (F_{IL}) (WUTMMCK0 must set to 1)
1	0	0	Main clock F_{MAX}/F_{HOCO} (via RTCCL7 selection)/1952
1	0	1	Main clock F_{MAX}/F_{HOCO} (via RTCCL7 selection)/1464
1	1	0	Main clock F_{MAX}/F_{HOCO} (via RTCCL7 selection)/488
1	1	1	Main clock F_{MAX}/F_{HOCO} (via RTCCL7 selection)/244

6.3.3 Real-Time Clock Control Register 0 (RTCC0)

This is an 8-bit register that sets the start or stop of real-time clock operation, the control of RTC1HZ pins, the 12/24-hour system and fixed cycle interrupts.

The RTCC0 register is set by an 8-bit memory manipulation instruction. After the reset signal is generated, the value of this register changes to “00H”.

Figure 6-4 Format of real-time clock control register 0 (RTCC0)

Address: 0x40044F5D

After reset: 00H

R/W

Symbol

RTCC0

	7	6	5	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1 ^{Note}	0	AMPM	CT2	CT1	CT0

RTCE	Real-time clock operation control
0	Stop the counter running.
1	Start the counter running.

RCLOE1	Output control of RTC1HZ pin
0	Disable RTC1HZ pin output (1Hz).
1	Enable RTC1HZ pin output (1Hz).

AMPM	Selection of 12-hour system/24-hour system
0	12-hour system (indicates morning or afternoon).
1	24-hour system

- To change the value of the AMPM bit, the RWAIT bit (bit 0 of the Real Time Clock Control Register 1 (RTCC1)) must be set to 1 and then rewritten. If the value of the AMPM bit is changed, the value of the hour count register (HOUR) changes to the corresponding value of the set time system.
- The time bits are represented as shown in Table 11-2.

CT2	CT1	CT0	Selection of fixed cycle interrupt (INTRTC)
0	0	0	The fixed-cycle interrupt function is not used.
0	0	1	Once every 0.5 seconds (synchronized with seconds accumulation)
0	1	0	Once every 1 second (synchronized with seconds accumulation)
0	1	1	Once every minute (00 seconds per minute).
1	0	0	Once every hour (00 minutes and 00 seconds per hour).
1	0	1	Once a day (00:00:00 per day).
1	1	×	Once a month (1st of each month at 00:00:00 a.m.).

To change the value of bits CT2~CT0 while the counter is running (RTCE=1), you must do the rewrite after setting INTRTC to disable interrupt processing via the interrupt mask flag register, and you must clear the RIFG flag and RTCIF flag after the rewrite, and then set it to enable interrupt processing.

Cautions

1. When the RTCE bit is “1”, the RCLOE1 bit cannot be changed.
2. When the RTCE bit is “0”, 1Hz is not output even if the RCLOE1 is set to “1”.

Remark ×: Ignore

6.3.4 Real-Time Clock Control Register 1 (RTCC1)

This is an 8-bit register that controls the alarm clock interrupt function and counter wait. The RTCC1 register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes “00H”.

Figure 6-5 Format of real-time clock control register 1 (RTCC1) (1/2)

Address: 0x40044F5E

After reset: 00H

R/W

Symbol

RTCC1

	7	6	5	4	3	2	1	0
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	Operation control of the alarm clock
0	Consistent operation is invalid.
1	Consistent operation is valid.
To set the WALE bit when the counter is running (RTCE=1) and the WALIE bit is 1, you must set INTRTC to disable interrupt processing through the interrupt mask flag register before the rewrite and clear the WAFG flag and RTCIF flag after the rewrite. To set each alarm register (WALIE flag of RTCC1 register, alarm minute register (ALARMWM), alarm hour register (ALARMWH) and the alarm week register (ALARMWW)), the WALE bit must be set to 0 (invalid for consistent operation).	

WALIE	Operation control of the alarm clock interrupt (INTRTC) function
0	No consistent alarm interruptions.
1	Generate consistent alarm interruptions.

WAFG	Alarm clock detection status flag
0	The alarm clock is inconsistent.
1	Consistent alarms detected.
This is a status flag that indicates that an alarm clock has been detected consistently. It is only valid when the WALE bit is 1 and changes to 1 after one FRTC clock has elapsed and the alarm is detected. Clear this flag by writing 0 to it. Writing a 1 is not valid.	

Figure 6-5 Format of real-time clock control register 1 (RTCC1) (2/2)

RIFG	Fixed-cycle interrupt status flag
0	No fixed-cycle interruptions are generated.
1	Generate fixed-cycle interrupts.
This is a status flag that indicates a fixed-cycle interrupt. When a fixed-cycle interrupt is generated, this flag is 1. Clear this flag by writing 0 to it. Writing a 1 is not valid.	

RWST	Wait status flag for the real-time clock
0	The counter is running.
1	It is in read-write mode for the counter.
This is the state that indicates whether the setting of the RWAIT bit is valid. The count value must be read and written after confirming that this flag is 1.	

RWAIT	Wait control of the real-time clock
0	Set to counter run.
1	Set the SEC to YEAR counter to stop running and enter the read/write mode of the counter
This bit controls the operation of the counter. To read and write a count value, you must write 1 to this bit. Because the internal counter (16-bit) continues to run, the read and write must end within 1 second and then return to 0. The minimum time required from the RWAIT bit set to 1 to the time the count value can be read and written (RWST=1) is 1 FRTC clock. If an internal counter (16 bits) overflows when the RWAIT bit is 1, the overflow state is maintained and the count is incremented after the RWAIT bit becomes 0. However, when the second count register is written, the overflow state that occurs is not maintained.	

Remarks:

- Fixed-cycle interrupts and alarm-consistent interrupts use the same interrupt source (INTRTC). In the case of using these two interrupts at the same time, when INTRTC occurs, you can determine which interrupt occurred by acknowledging the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG).
- If writing to the second count register (SEC), then clear the internal counter (16-bit).

6.3.5 Clock Error Correction Register (SUBCUD)

This is a register that can correct the clock speed with high accuracy by changing the overflow value (reference value: 7FFFH) from the internal counter (16 bits) to the second count register (SEC).

The SUBCUD register is set by a 16-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes “0000H”.

Figure 6-6 Format of clock error correction register (SUBCUD)

Address: 0x40044F34H	After reset: 0000H	R/W						
Symbol	15	14	13	12	11	10	9	8
SUBCUD	DEV	0	0	F12	F11	F10	F9	F8
	7	6	5	4	3	2	1	0
	F7	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of clock error correction timings
0	Corrects clock error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects clock error only when the second digits are at 00 (every 60 seconds).

Writing to the SUBCUD register at the following timing is prohibited.

- When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H
- When DEV = 1 is set: For a period of SEC = 00H

F12	Setting of clock error correction value
0	Increases by $\{(F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) - 1\} \times 2$
1	Decreases by $\{(F11 /F10/ F9 /F8/ F7 /F6/ F5 /F4/ F3 /F2/ F1 /F0) + 1\} \times 2$

When
 $(F12, F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1), (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$, the clock error is not corrected.
 Range of correction value: (F12=0) 2, 4, 6, 8, ..., 8186, 8188
 (F12=1) -2, -4, -6, -8, ..., -8186, -8188

Caution: “/” indicates the inverted value of each bit.

The range of value that can be corrected by using the clock error correction register (SUBCUD) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	-12496.9ppm~12496.9ppm	-4165.6ppm to 4165.6ppm
Maximum quantization error	±1.53ppm	±0.51ppm
Minimum resolution	±3.05ppm	±1.02ppm

Remark If a correctable range is -4165.6 ppm or lower and 4165.6 ppm or higher, set 0 to DEV.

6.3.6 Second Count Register (SEC)

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds. It counts up when the internal counter (16-bit) overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of F_{RTC} later. Set a decimal value of 00 to 59 to this register in BCD code.

The SEC register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes “00H”.

Figure 6-7 Format of second count register (SEC)

Address: 0x40044F52 After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

Caution When it reads or writes from/to the register while the counter is in operation ($RTCE = 1$), follow the procedures described in 6.4.3 Reading/Writing Real-Time Clock.

Remark The internal counter (16-bit) is cleared when the second count register (SEC) is written.

6.3.7 Minute Count Register (MIN)

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes. It counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of F_{RTC} later. Even if the second count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 59 to this register in BCD code.

The MIN register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes “00H”.

Figure 6-8 Format of minute count register (MIN)

Address: 0x40044F53 After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

Caution When it reads or writes from/to the register while the counter is in operation ($RTCE = 1$), follow the procedures described in 6.4.3 Reading/Writing Real-Time Clock.

6.3.8 Hour Count Register (HOUR)

The HOUR register is an 8-bit register that takes a value of 00 to 23 or 01 to 12 and 21 to 32 (decimal) and indicates the count value of hours. It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of F_{RTC} later. Even if the minute count register overflows while this register is being written, this register ignores the overflow and is set to the value written.

Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using bit 3 (AMPM) of real-time clock control register 0 (RTCC0).

If the AMPM bit value is changed, the values of the HOUR register change according to the specified time system. The HOUR register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes “12H”.

However, if the AMPM bit is set to “1” after a reset, the value of this register will become “00H”.

Figure 6-9 Format of hour count register (HOUR)

Address: 0x40044F54 After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Cautions:

1. Bit 5 (HOUR20) of the HOUR register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).
2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 6.4.3 Reading/Writing Real-Time Clock.

Table 6-2 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 6-2 Displayed time digits

24-Hour Display (AMPM = 1)		12-Hour Display (AMPM = 0)	
Time	HOUR Register	Time	HOUR Register
0	00H	12 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	12 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

The HOUR register value is set to 12-hour display when the AMPM bit is “0” and to 24-hour display when the AMPM bit is “1”.

In 12-hour display, the fifth bit of the HOUR register indicates AM/PM. 0 for AM and 1 for PM.

6.3.9 Day Count Register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days. It counts up when the hour counter overflows. This counter counts as follows:

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of F_{RTC} later. Even if the hour count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 31 to this register in BCD code.

The DAY register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes “01H”.

Figure 6-10 Format of day count register (DAY)

Address: 0x40044F56H	After reset: 01H		R/W					
Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

Caution When it reads or writes from/to the register while the counter is in operation ($RTCE = 1$), follow the procedures described in 6.4.3 Reading/Writing Real-Time Clock.

6.3.10 Week Count Register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays. It counts up in synchronization with the day counter.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of F_{RTC} later. Set a decimal value of 00 to 06 to this register in BCD code.

The WEEK register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes “00H”.

Figure 6-11 Format of week count register (WEEK)

Address: 0x40044F55H				After reset: 00H		R/W		
Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Caution 1. The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the week count register (WEEK) automatically. After reset release, set the week count register as follow:

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

- When it reads or writes from/to the register while the counter is in operation ($RTCE = 1$), follow the procedures described in 6.4.3 Reading/Writing Real-Time Clock.

6.3.11 Month Count Register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months. It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of F_{RTC} later. Even if the day count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 12 to this register in BCD code.

The MONTH register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes “01H”.

Figure 6-12 Format of month count register (MONTH)

Address: 0x40044F57H	After reset: 01H			R/W				
Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

Caution When it reads or writes from/to the register while the counter is in operation ($RTCE = 1$), follow the procedures described in 6.4.3 Reading/Writing Real-Time Clock.

6.3.12 Year Count Register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years. It counts up when the month count register (MONTH) overflows.

Values 00, 04, 08, ..., 92, and 96 indicate a leap year. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of F_{RTC} later. Even if the MONTH register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 99 to this register in BCD code. The YEAR register is set by an 8-bit memory manipulation instruction.

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

Figure 6-13 Format of year count register (YEAR)

Address: 0x40044F58H	After reset: 00H		R/W					
Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

Caution When it reads or writes from/to the register while the counter is in operation ($RTCE = 1$), follow the procedures described in 6.4.3 Reading/Writing Real-Time Clock.

6.3.13 Alarm Minute Register (ALARMWM)

This register is used to set minutes of alarm.

The ALARMWM register can be set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes “00H”.

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 6-14 Format of alarm minute register (ALARMWM)

Address: 0x40044F5AH	After reset: 00H	R/W					
Symbol	7	6	5	4	3	2	1 0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2 WM1

6.3.14 Alarm Hour Register (ALARMWH)

This register is used to set hours of alarm.

The ALARMWH register can be set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes “12H”.

However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 6-15 Format of alarm hour register (ALARMWH)

Address: 0x40044F5BH	After reset: 12H	R/W					
Symbol	7	6	5	4	3	2	1 0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2 WH1

Caution Bit 5 (WH20) of the ALARMWH register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

6.3.15 Alarm Week Register (ALARMWW)

This register is used to set date of alarm.

The ALARMWW register can be set by an 8-bit memory operation instruction. After a reset signal is generated, the value of this register becomes “00H”.

Figure 6-16 Format of alarm week register (ALARMWW)

Address: 0x40044F5CH

After reset: 00H

R/W

Symbol

7

6

5

4

3

2

1

0

ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

Here is an example of setting the alarm.

Time of alarm	Day							12-hour display				24-hour display			
	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Hour	Hour	Minute	Minute	Hour	Hour	Minute	Minute
	W	W	W	W	W	W	W	10	1	10	1	10	1	10	1
	0	1	2	3	4	5	6								
Every day 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Mon~Fri 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Mon, Wed, Fri 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

6.3.16 Port Mode Register and Port Register

For a 1Hz output, the RTC1HZ output pin can be alternated to any port except P00.

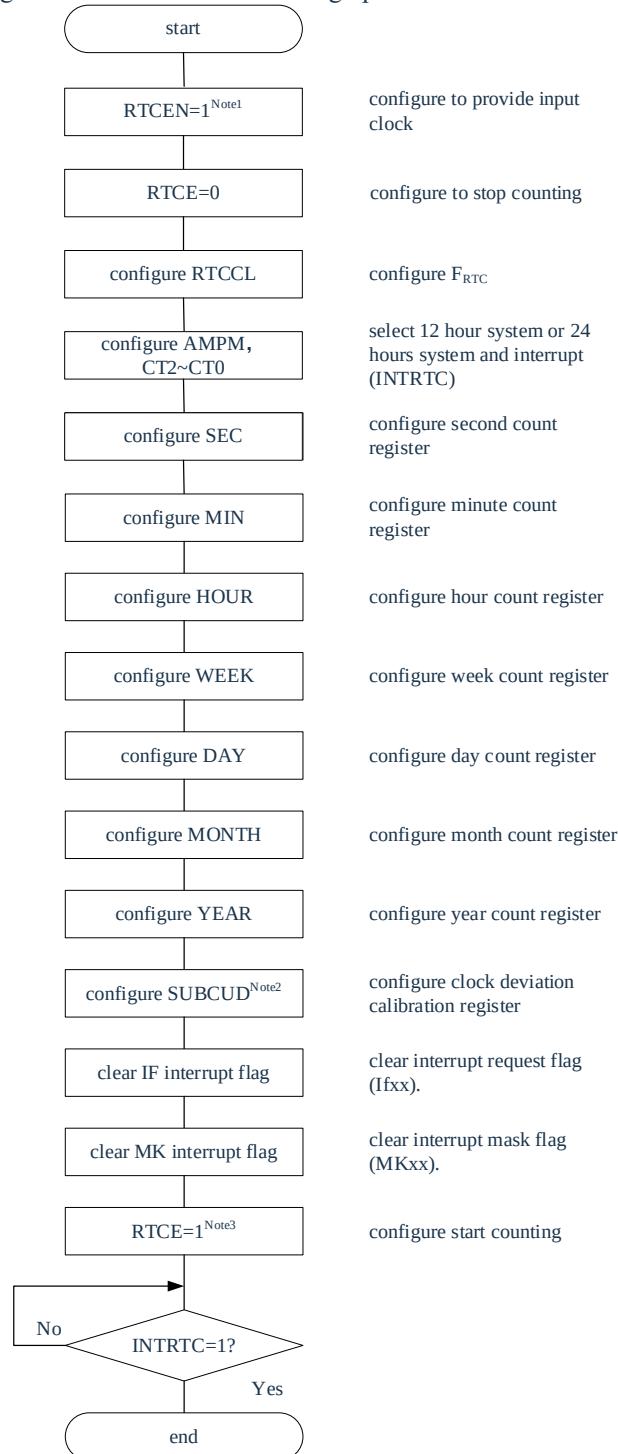
- Configure the output port's PxxCFG to “0x1c”.
- Set the corresponding port mode control register (PMCxx) bits and the port mode register (PMxx) bits to 0.

For detailed information, please refer to Chapter 2: Pin Functions, Section 2.3.10 Port Alternate Function Configuration Register (PxxCFG).

6.4 Operation of Real-Time Clock

6.4.1 Start of Real-Time Clock Operation

Figure 6-17 Procedure for starting operation of real-time clock



Notes

1. Set the RTCEN bit to 1, while oscillation of the count clock (F_{RTC}) is stable.
2. Set up the register only if the clock error must be corrected. For details about how to calculate the correction value, see “6.4.6 Example of Clock Error Correction of Real-Time Clock”.
3. Confirm the procedure described in 6.4.2 Shifting to Sleep Mode After Starting Operation when shifting to sleep mode without waiting for INTRTC = 1 after RTCE = 1.

6.4.2 Shifting to Sleep Mode After Starting Operation

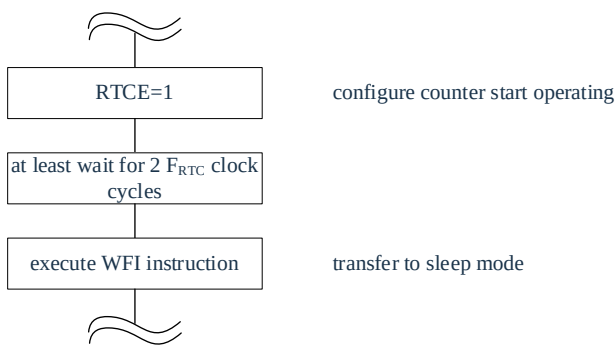
Perform some of the following processing when shifting to sleep mode immediately after setting the RTCE bit to 1.

However, after setting the RTCE bit to 1, this processing is not required when shifting to sleep mode after the INTRTC interrupt has occurred.

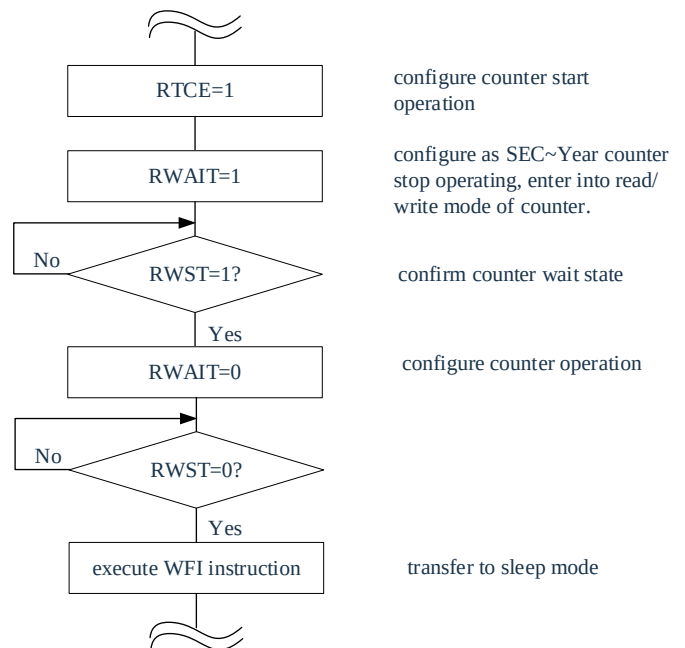
- Shifting to sleep mode when at least two cycles of the count clock (FRTC) have elapsed after setting the RTCE bit to 1 (see Figure 6-18, Example 1).
- Checking by polling the RWST bit to become 1, after setting the RTCE bit to 1 and then setting the RWAIT bit to 1. Afterward, setting the RWAIT bit to 0 and shifting to sleep mode after checking again by polling that the RWST bit has become 0 (see Figure 6-18, Example 2).

Figure 6-18 Procedure for shifting to sleep/deep sleep mode after setting RTCE bit to 1

Example1



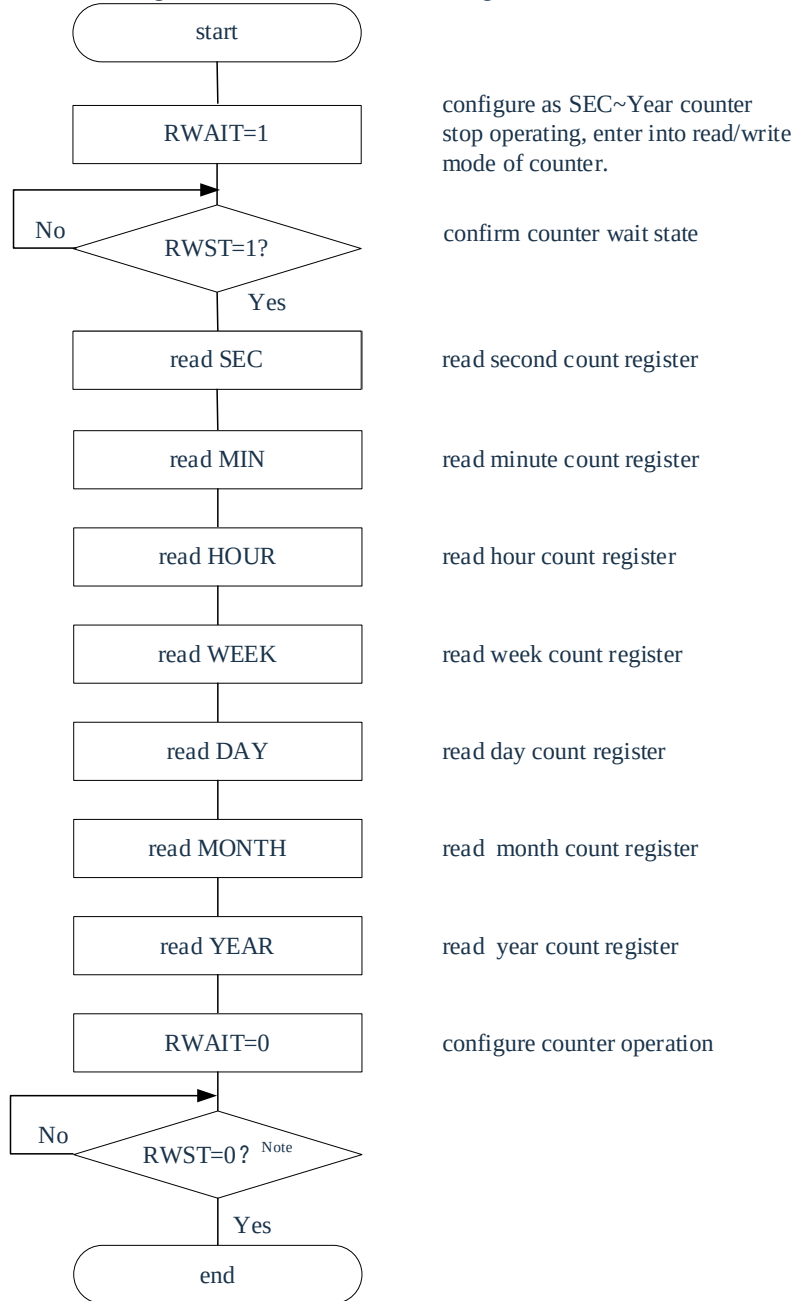
Example 2



6.4.3 Reading/Writing Real-Time Clock

Read or write the counter after setting “1” to RWAIT first. Set RWAIT to “0” after completion of reading or writing the counter.

Figure 6-19 Procedure for reading real-time clock

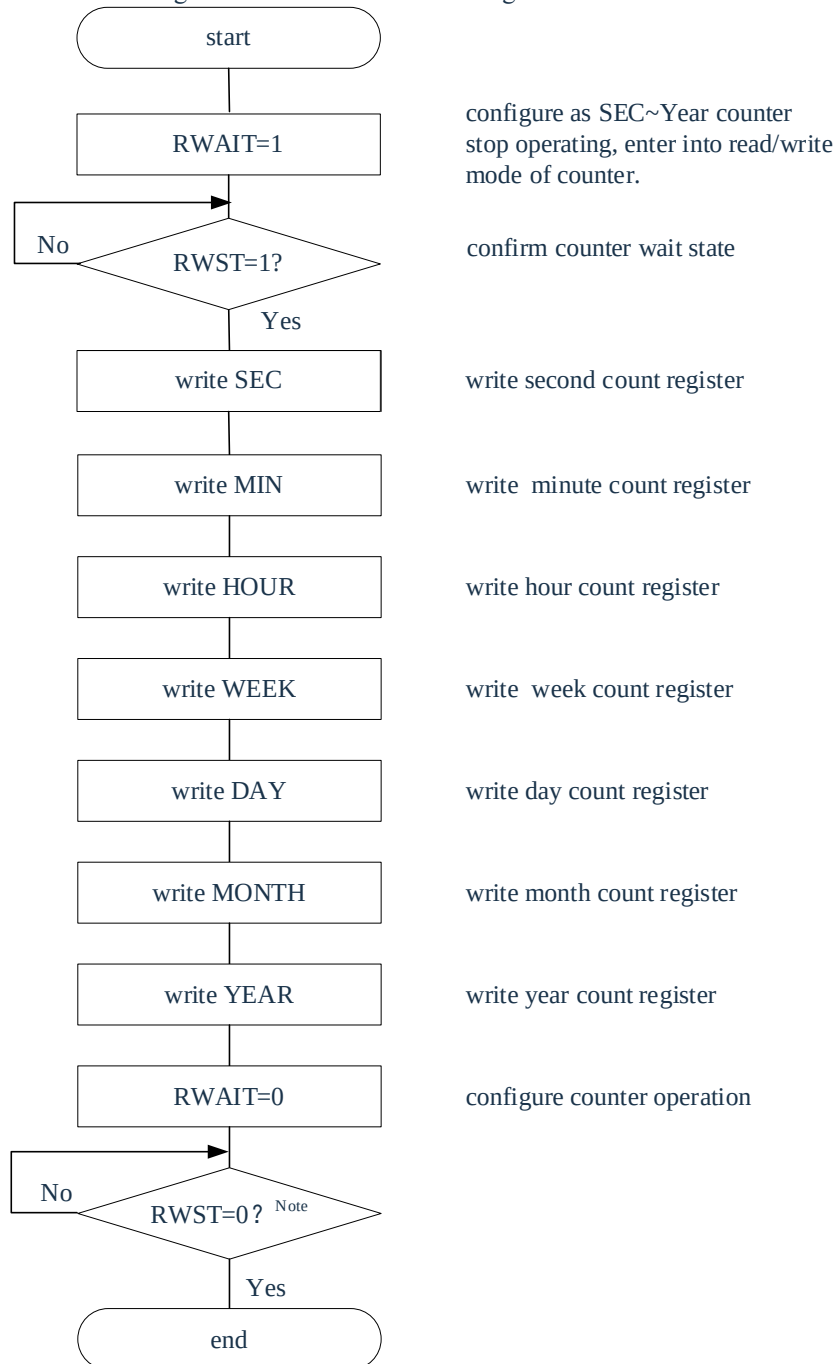


Note Be sure to confirm that RWST = 0 before setting sleep mode.

Caution Complete the series of process of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read in any sequence. All the registers do not have to read and only some registers may be read.

Figure 6-20 Procedure for reading real-time clock



Note Be sure to confirm that RWST = 0 before setting sleep mode.

Cautions

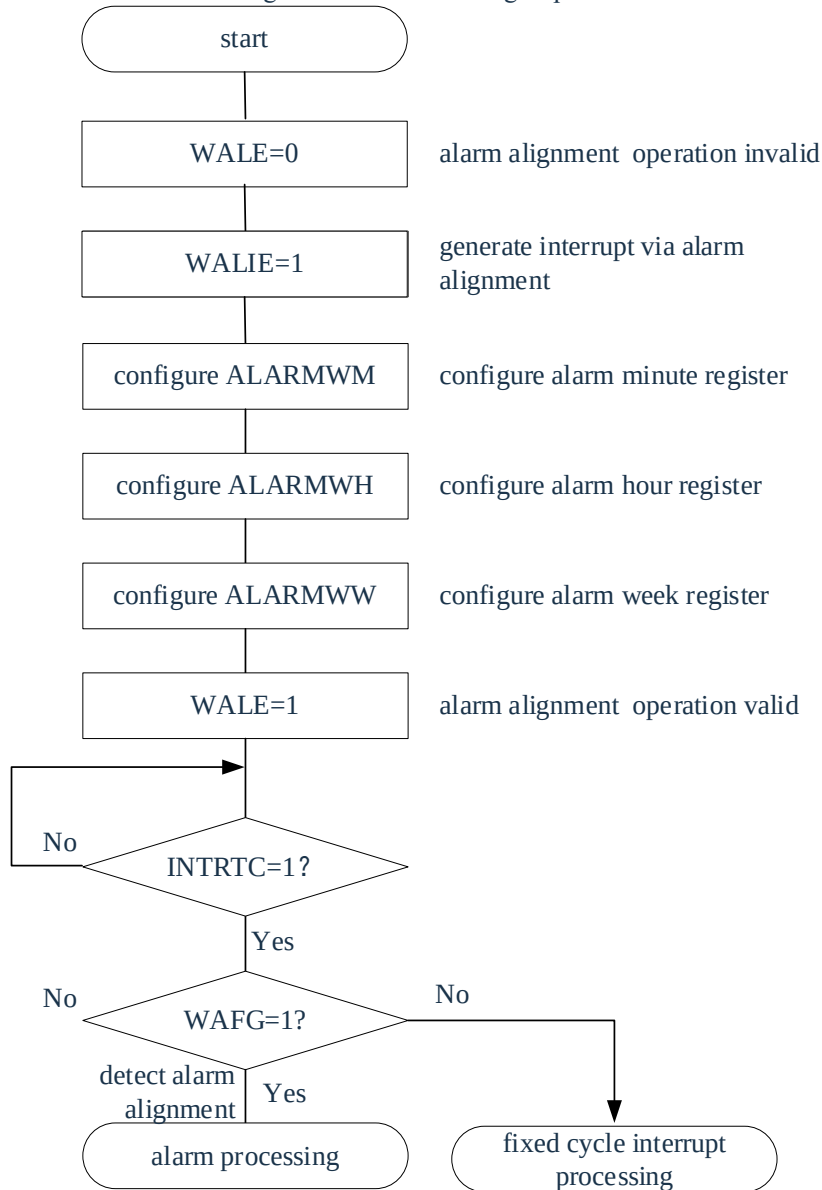
1. Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.
2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR register while the counter operates (RTCE = 1), rewrite the values after disabling interrupt servicing INTRTC by using the interrupt mask flag register, and the WAFG, RIFG, and RTCIF flags should be cleared after the rewrite.

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read in any sequence. All the registers do not have to read and only some registers may be read.

6.4.4 Setting Alarm of Real-Time Clock

Set alarm time after setting 0 to WALE (alarm operation invalid) first.

Figure 6-21 Alarm setting steps

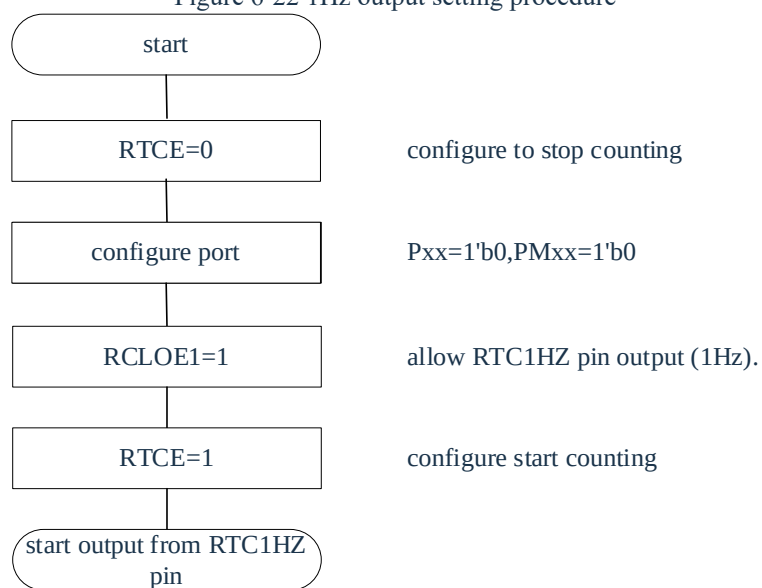


Remarks

1. There is no limit to the order of write operations for alarm minute registers (ALARMWWM), alarm hour registers (ALARMWH), and alarm week registers (ALARMWW).
2. Fixed-cycle interrupts and alarm-consistent interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

6.4.5 1Hz Output of Real-Time Clock

Figure 6-22 1Hz output setting procedure



Caution 1. First set the RTCEN bit to 1, while oscillation of the count clock (F_{SUB}) is stable.

6.4.6 Example of Clock Error Correction of Real-Time Clock

The clock can be corrected with high accuracy when it is slow or fast, by setting a value to the clock error correction

Example of calculating the correction value

register.

The correction value used when correcting the count value of the internal counter (16-bit) is calculated by using the following equation: If a correctable range is -4165.6 ppm or lower and 4165.6 ppm or higher, set 0 to DEV.

(When DEV=0)

Correction value^{Note}=Number of correction counts in 1 minute/3=(Oscillation frequency/Target frequency-1)×32768×60/3

(When DEV=1)

Correction value^{Note}=Number of correction counts in 1 minute=(Oscillation frequency/Target frequency-1) ×32768×60

Note: The correction value is the clock error correction value calculated by using bits 12 to 0 of the clock error correction register (SUBCUD).

(When F12=0) correction value = {(F11,F10,F9,F8,F7,F6,F5,F4,F3,F2,F1,F0)-1}×2

(When F12=0) correction value = - {(/F11,/F10,/F9,/F8,/F7,/F6,/F5,/F4,/F3,/F2,/F1,/F0)+1}×2

When (F12~F0)=(*,0,0,0,0,0,0,0,0,0,0,*), clock error correction is not performed. “*” is 0 or 1.

/F12~/F0 are bit-inverted values (“000000000011” when “111111111100”).

Remarks:

1. The correction value is 2,4,6,8,.....,8186, 8188 or -2,-4,-6,-8,.....,-8186,-8188.
2. The oscillation frequency is the value of the counting clock (FRTC), and can be calculated by the following equation:
The output frequency of the RTC1HZ pin×32768 when the clock error correction register is set to its initial value (00H).
3. The target frequency is the frequency resulting after correction performed by using the clock error correction register.

Correction example

Example of correcting from 32767.4Hz to 32768Hz (32767.4Hz+18.3ppm)

[Measuring the oscillation frequency]

When the watch error correction register (SUBCUD) is the initial value ("0000H"), the oscillation frequency of each product is measured by outputting a signal of approximately 1Hz from the RTC1HZ pin^{Note}.

Note: For the setting of RTC1Hz output, please refer to "6.4.5 1Hz output of real-time clock".

[Calculating the correction value]

(When the output frequency of the RTC1HZ pin is 0.9999817Hz)

Oscillation frequency = $32768 \times 0.9999817 \approx 32767.4\text{Hz}$

Suppose the target frequency is 32768Hz (32767.4Hz+18.3ppm) and DEV=1. An equation for calculating the correction value when the DEV bit is 1 is applied.

$$\begin{aligned}\text{Correction value} &= \text{Number of correction counts in 1 minute} = (\text{Oscillation frequency} / \text{Target frequency} - 1) \times 32768 \times 60 \\ &= (32767.4 / 32768 - 1) \times 32768 \times 60 \\ &= -36\end{aligned}$$

[Calculating the values to be set to (F12~F0)]

(When the correction value = -36)

If the correction value is less than 0 (when faster), assume F12=1. Calculating is based on correction values (F11~F0).

$$-\{(/F11 \sim /F0) - 1\} \times 2 = -36$$

$$(/F11 \sim /F0) = 17$$

$$(/F11 \sim /F0) = (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1)$$

$$(F11 \sim F0) = (1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0)$$

Therefore, the correction from 32767.4Hz to 32768Hz (32767.4Hz + 18.3ppm) is as follows:

If by DEV=1 and correction value=-36 (bit12~0 of the SUBCUD register: 1,1,1,1,1,1,1,0,1,1,1,0) to set the correction register, you can correct it to 32768Hz (0ppm).

Chapter 7 15-Bit Interval Timer

7.1 Functions of 15-Bit Interval Timer

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

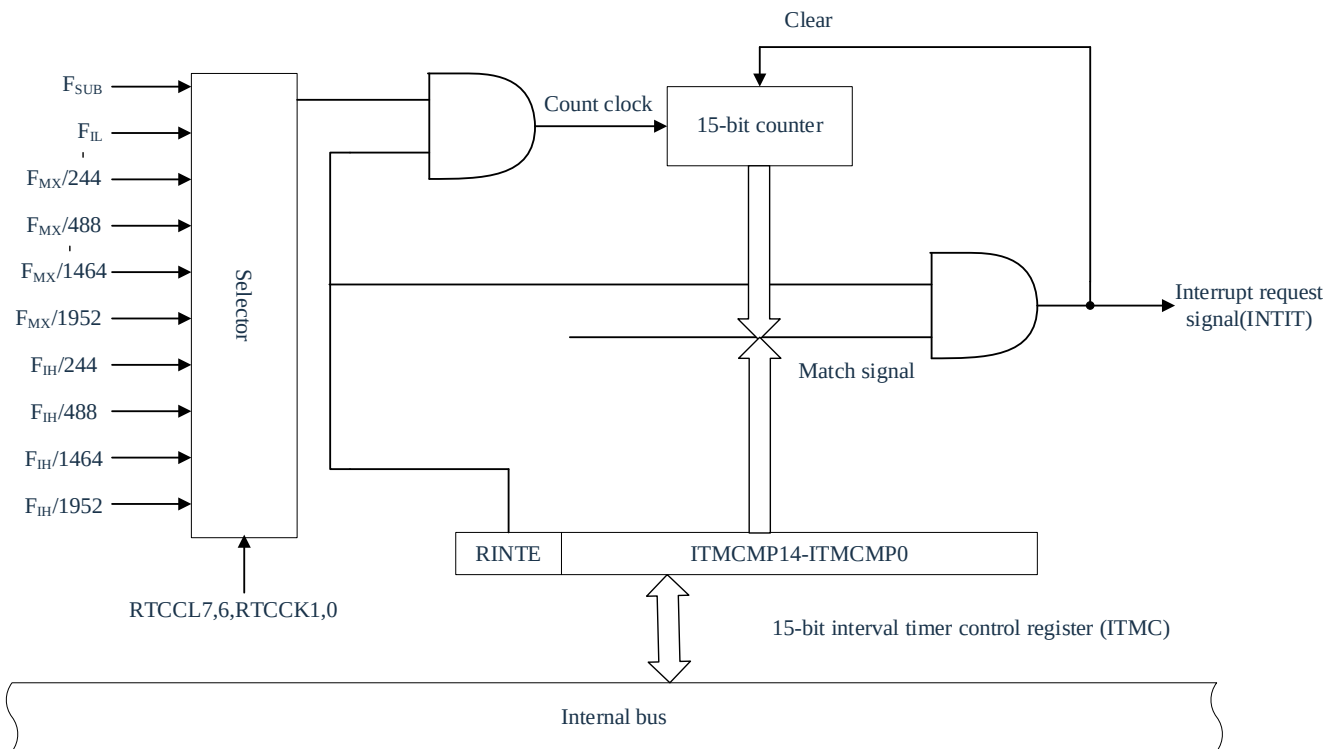
7.2 Structure of 15-Bit Interval Timer

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

Table 7-1 Structure of 15-bit interval timer

Item	Structure
Counter	15-bit counter
Control register	Peripheral enable register 0 (PER0)
	Real-time clock selection register (RTCCL)
	15-bit interval timer control register (ITMC)

Figure 7-1 Block diagram of 15-bit interval timer



7.3 Registers for Controlling 15-Bit Interval Timer

The 15-bit interval timer is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Real-time clock selection register (RTCCL)
- 15-bit interval timer control register (ITMC)

7.3.1 Peripheral Enable Register 0 (PER0)

The PER0 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.

When using a 15-bit interval timer, bit7 (RTCEN) must be set to 1. The PER0 register is set by an 8-bit memory manipulation instruction. After the reset signal is generated, the value of this register changes to “00H”.

Figure 7-2 Format of peripheral enable register 0 (PER0)

Address: 0x40020420	After reset: 00H		R/W					
Symbol	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICA0EN	SCI1EN	SCI0EN	TM41EN	TM40EN
RTCEN	Control of an input clock of a real-time clock (RTC) and a 15-bit interval timer							
0	Stop to supply the input clock. •The SFR used by the Real Time Clock (RTC) and the 15-bit interval timer cannot be written. •The real-time clock (RTC) and 15-bit interval timer are in the reset state.							
1	Supply the input clock. •The SFR used by the Real Time Clock (RTC) and the 15-bit interval timer can be read and written.							

7.3.2 Real-Time Clock Selection Register (RTCCL)

The real-time clock and the count clock (F_{RTC}) of the 15-bit interval timer can be selected via RTCCL.

Figure 7-3 Format of real-time clock selection register (RTCCL)

Address: 0x4004047C

After reset: 00H

R/W

Symbol	7	6	5	4	3	2	1	0
RTCCL	RTCCL7	RTCCL6	0	0	0	0	RTCKS1	RTCKS0

RTCCL7	Selection of clock source for real time clock, counting clock for 15-bit interval timer
0	Selects high-speed system clock (F_{MX})
1	Selects high-speed on-chip oscillator (F_{HOCO})

RTCKS1	RTCKS0	RTCCL6	Selection of operating clocks for real-time clocks, counting clocks for 15-bit interval timer
0	0	0/1	Subsystem clock (F_{SUB})
0	1		Low-speed internal oscillator clock (F_{IL}) (WUTMMCK0 must be set to 1).
1	0	0	Main clock F_{MAX}/F_{HOCO} (selected via RTCCL7)/1952
1	0	1	Main clock F_{MAX}/F_{HOCO} (selected via RTCCL7)/1464
1	1	0	Main clock F_{MAX}/F_{HOCO} (selected via RTCCL7)/488
1	1	1	Main clock F_{MAX}/F_{HOCO} (selected via RTCCL7)/244

7.3.3 15-Bit Interval Timer Control Register (ITMC)

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”.

Figure 7-4 Format of 15-bit interval timer control register (ITMC)

Address: 0x40044F50

After reset: 7FFFH

R/W

Symbol 15

14~0

ITMC

RINTE

ITCMP14~ITCMP0

RINTE	15-bit interval timer operation control
0	Count operation stopped (count clear)
1	Start operation of counters.

ITCMP14~ITCMP0	Specification of the 15-bit interval timer compare value
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 7FFFH is specified for ITCMP14~ITCMP0 - ITCMP14~ITCMP0=0001H, count clock: $F_{SUB}=32.768\text{KHz}$ $1/32.768[\text{KHz}] \times (1+1) = 0.06103515625[\text{ms}] \approx 61.03[\mu\text{s}]$ - ITCMP14~ITCMP0=7FFFH, count clock: $F_{SUB}=32.768\text{KHz}$ $1/32.768[\text{KHz}] \times (32767+1) = 1000[\text{ms}]$	

Cautions:

- Before changing the RINTE bit from “1” to “0”, use the interrupt mask flag register to disable the INTIT interrupt servicing. When the operation starts (from “0” to “1”) again, clear the ITIF flag, and then enable the interrupt servicing.
- The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit.
- After shifting from sleep mode to normal operation mode, if you want to set the ITMC register and enter sleep mode again, you must confirm that the write value of the ITMC register is reflected or set the ITMC registers or wait that more than one clock of the count clock has elapsed before shifting to sleep mode.
- Only change the setting of the ITCMP14 to ITCMP0 bits when RINTE = 0. However, it is possible to change the settings of the ITCMP14 to ITCMP0 bits at the same time as when changing RINTE from “0” to “1” or “1” to “0”.

7.4 Operation of 15-Bit Interval Timer

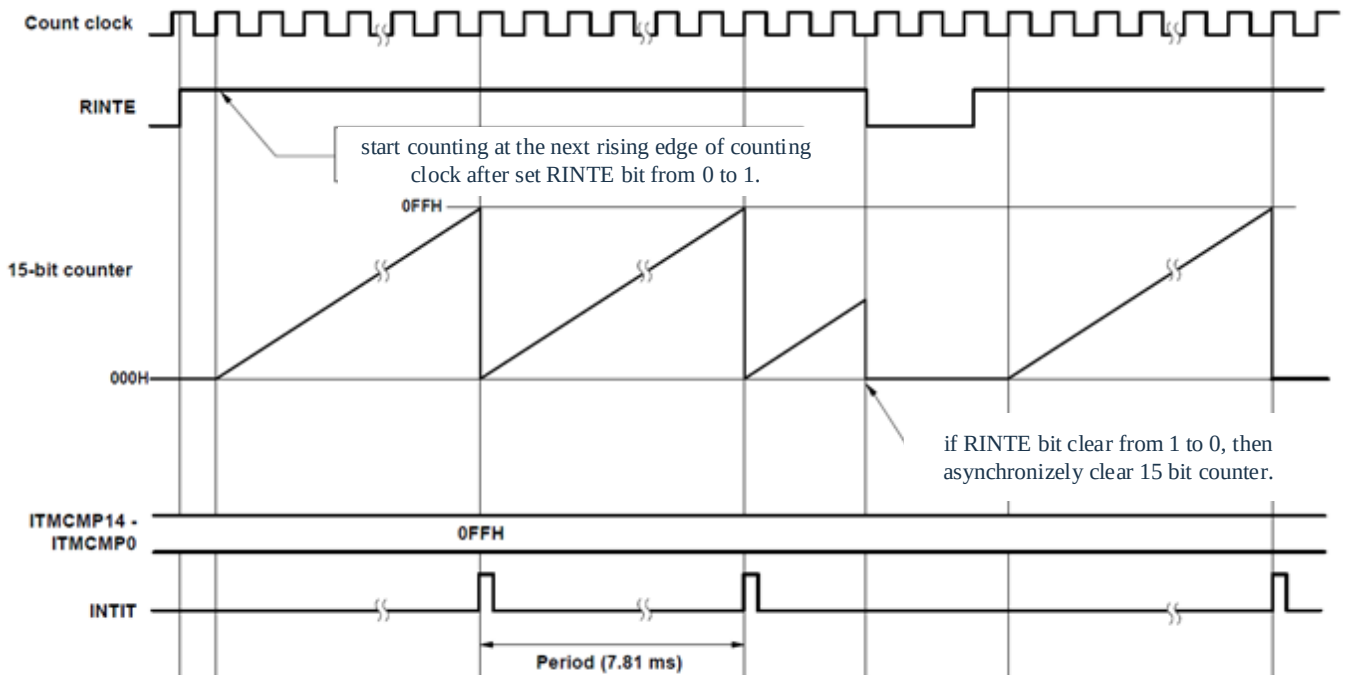
7.4.1 Operation Timing of 15-Bbit Interval Timer

The count value specified for the ITCMP14 to ITCMP0 bits is used as an interval to operate a 15-bit interval timer that repeatedly generates interrupt requests (INTIT). When the RINTE bit is set to “1”, the 15-bit counter starts counting.

When the 15-bit counter value matches the value specified for the ITCMP14 to ITCMP0 bits, the 15-bit counter value is cleared to 0, counting continues, and an interrupt request signal (INTIT) is generated at the same time.

The basic operation of the 15-bit interval timer is as follows Figure 7-5.

Figure 7-5 Operation timing of 15-bit interval timer
(ITCMP14~ITCMP0=0FFH, count clock: $F_{SUB}=32.768\text{KHz}$)



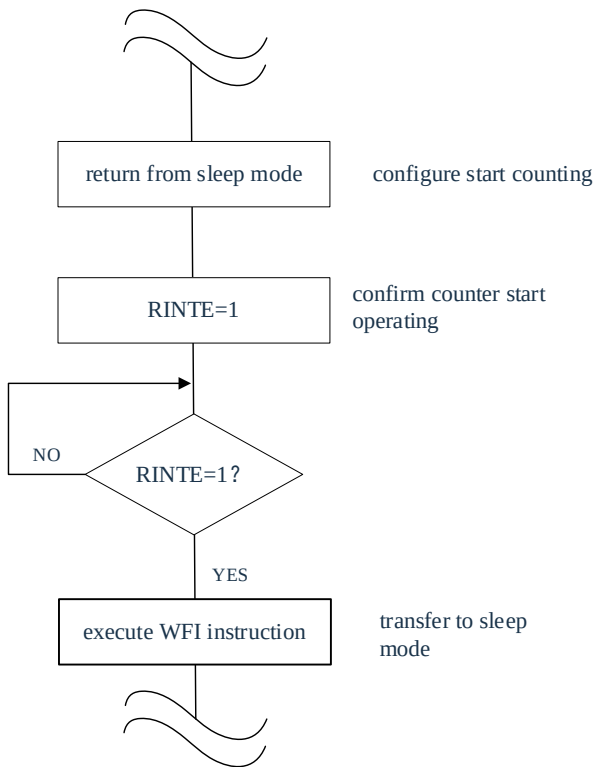
7.4.2 Start of Count Operation and Re-Enter to Sleep Mode After Returned from Sleep Mode

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.

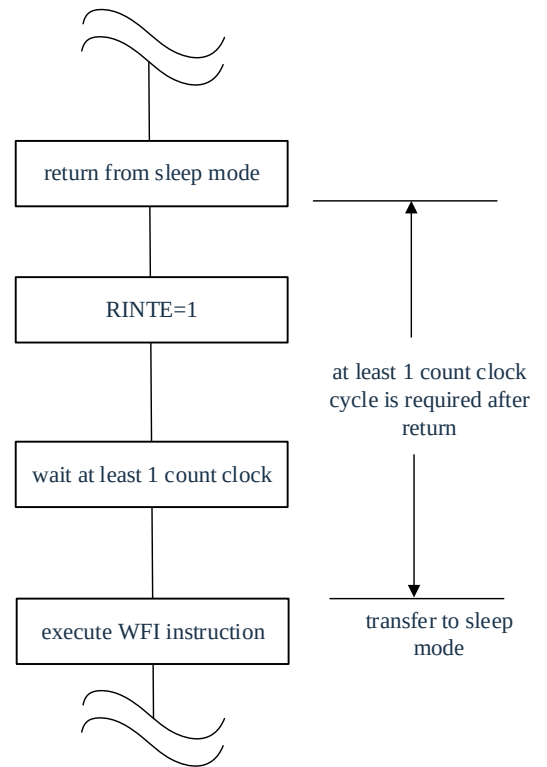
After setting RINTE to 1, confirm by polling that the RINTE bit has become 1, and then enter sleep mode (see Example 1 in the figure below).

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 8 Clock Output/Buzzer Output Controller

Caution The following section of this chapter focuses on 24-pin products.

8.1 Functions of Clock Output/Buzzer Output Controller

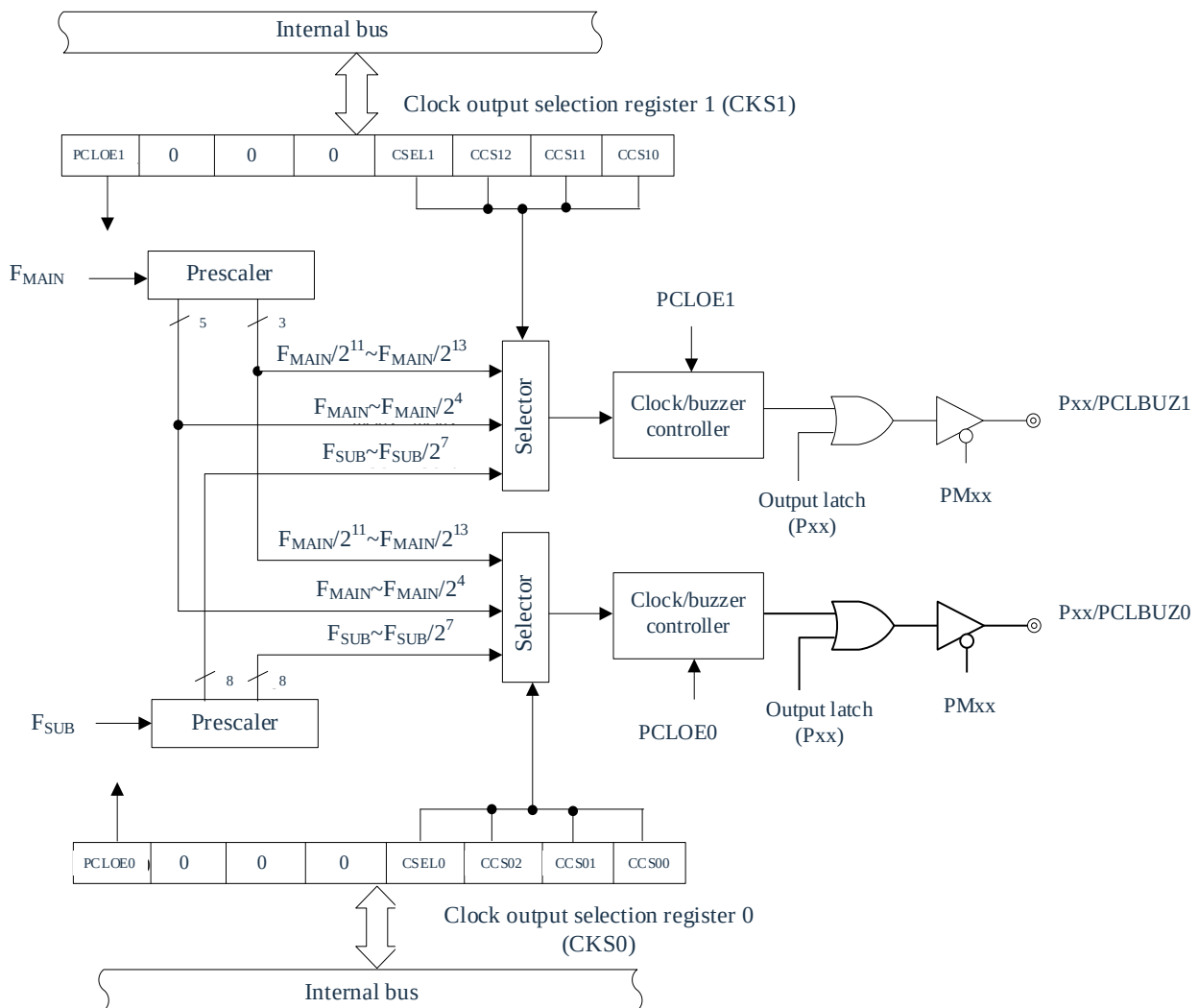
Clock output is the function of outputting the clock provided to the peripheral IC, and buzzer output is the function of outputting the buzzer frequency square wave. This product can be selected to be used as clock output or buzzer output from any pin other than P00. The PCLBUZn pin outputs the clock selected by the clock output selection register n (CKSn).

The block diagram of the clock output/buzzer output controller is shown in Figure 8-1.

Caution When the RTCLPC bit of the subsystem clock supply mode control register (OSMC) is set to 1 and the CPU is running in Sleep mode with the sub-system clock (F_{SUB}), the sub-system clock (F_{SUB}) cannot be output from the PCLBUZn pin.

Remark n=0, 1

Figure 8-1 Block diagram of clock output/buzzer output controller



Note For the frequencies that can be output from PCLBUZ0 and PCLBUZ1 pins, please refer to “AC Characteristics” in the data sheet.

8.2 Structure of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller consists of the following hardware.

Table 8-1 Registers for controlling clock output/buzzer output controller

Item	Structure
Control register	Clock output select register n (CKSn) Port mode control register (PMCxx) Port mode register (PMxx) Port alternate control register (PxxCFG)

8.3 Registers for Controlling Clock Output/Buzzer Output Controller

8.3.1 Clock Output Select Register n (CKSn)

These registers set output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZn), and set the output clock.

Select the clock to be output from the PCLBUZn pin by using the CKSn register. The CKSn register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes “00H”.

Figure 8-2 Format of clock output select register n (CKSn)

Address: 0x40040FA5 (CKS0), 0x40040FA6 (CKS1) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn pin output enable/disable specification
0	Output disable (default)
1	Output enable

CSELn	CCSn2	CCSn1	CCSn0	PCLBUZn pin output clock selection
0	0	0	0	F _{MAIN}
0	0	0	1	F _{MAIN} /2
0	0	1	0	F _{MAIN} /2 ²
0	0	1	1	F _{MAIN} /2 ³
0	1	0	0	F _{MAIN} /2 ⁴
0	1	0	1	F _{MAIN} /2 ¹¹
0	1	1	0	F _{MAIN} /2 ¹²
0	1	1	1	F _{MAIN} /2 ¹³
1	0	0	0	F _{SUB}
1	0	0	1	F _{SUB} /2
1	0	1	0	F _{SUB} /2 ²
1	0	1	1	F _{SUB} /2 ³
1	1	0	0	F _{SUB} /2 ⁴
1	1	0	1	F _{SUB} /2 ⁵
1	1	1	0	F _{SUB} /2 ⁶
1	1	1	1	F _{SUB} /2 ⁷

Note: Use the output clock within a range of 16 MHz. For details, please refer to “AC Characteristics” in the data sheet.

Cautions:

1. Change the output clock after disabling clock output (PCLOEn = 0).
2. To shift to deep sleep mode when the main system clock is selected (CSELn = 0), set PCLOEn = 0 before executing the WFI instruction. When the subsystem clock is selected (CSELn = 1), PCLOEn = 1 can be set because the clock can be output while the RTCLPC bit of the subsystem clock supply mode control (OSMC) register is set to 0 and moreover while deep sleep mode is set.
3. When the RTCLPC bit of the Oscillator Subsystem Mode Control Register (OSMC) is set to 1 and the CPU is running in Sleep mode with the sub-system clock (F_{SUB}), the sub-system clock (F_{SUB}) cannot be output from the PCLBUZn pin.

Remarks:

1. n=0, 1
2. F_{MAIN}: Main system clock frequency
F_{SUB}: Subsystem clock frequency

8.3.2 Registers for Controlling Clock Output/Buzzer Output Port Functions

This product allows the clock output/buzzer output function to be alternated to any port except P00. The port alternate function configuration register (PxxCFG), the port mode register (PMxx), and the port mode control register (PMCxx) must be properly set. For detailed information, refer to Chapter 2: Pin Functions.

For the alternated port configured as a clock output/buzzer output pin, the corresponding bit in the port mode register (PMxx) and the port mode control register (PMCxx) must be set to 0.

(Example) Using P20 as the clock output/buzzer output (PCLBUZ0):

Set the bit PM20 in the port mode register 2 to 0.

Set the bit PMC20 in the port mode control register 2 to 0.

Set the port alternate function configuration register P20CFG to 0x18.

8.4 Operation of Clock Output/Buzzer Output Controller

One pin can be used as clock output or buzzer output.

The PCLBUZ0 pin outputs a clock/buzzer selected by the clock output select register 0 (CKS0).

The PCLBUZ1 pin outputs a clock/buzzer selected by the clock output select register 1 (CKS1).

8.4.1 Operation as Output Pin

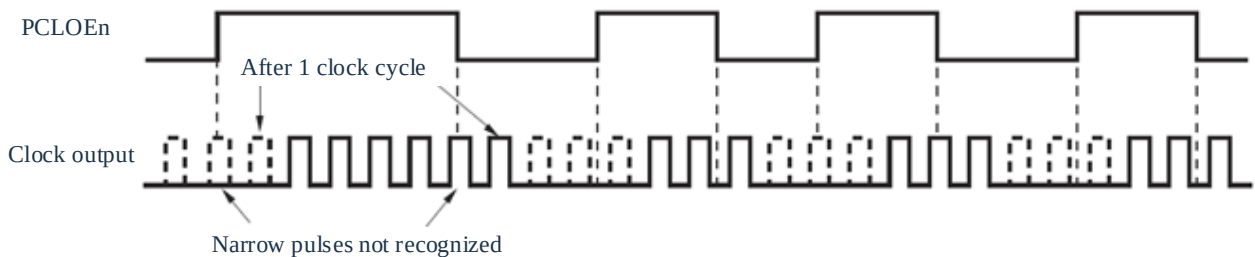
The PCLBUZn pin is output as the following procedure.

- ① Set the port alternate function configuration register (PxxCFG), and ensure that the corresponding bits in the port mode register (PMxx) and port mode control register (PMCxx) for the pin used as the PCLBUZ0 pin are set to 0.
- ② Use the clock output selection register (CKSn) for the PCLBUZn pin, and select the output frequency using bits 0 to 3 (CCSn0-CCSn2, CSELn) of the CKSn register (the output should be in the disabled state).
- ③ Set bit 7 (PCLOEn) of the CKSn register to 1 to enable the clock/buzzer output.

Remarks:

1. The control circuit for the clock output starts or stops the clock output one clock cycle after the PCLOEn bit is set to enable or disable the clock output. At this time, no narrow pulses are output. The timing for enabling or stopping the output through the PCLOEn bit and the clock output is shown in Figure 8-3.
2. n=0, 1

Figure 8-3 CLKBUZn pin output clock timing



8.5 Cautions for Clock Output/Buzzer Output Controller

When the main system clock is selected as the PCLBUZn output (CSELn=0), if the device transitions to deep sleep mode within 1.5 clock cycles of the PCLBUZn pin's output clock after setting the stop output (PCLOEn=0), the output width of PCLBUZn becomes narrower.

Chapter 9 Watch Dog Timer

9.1 Functions of Watch Dog Timer

The counting operation of the watchdog timer is set by the option byte (000C0H). The watchdog timer operates on the low-speed on-chip oscillator clock (F_{IL}). The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- When a bitwise operation is performed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to the WDTE register
- If data is written to the WDTE register during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of the RESF register, see Chapter 20 Reset Function. When 75% of the overflow time + $1/2 F_{IL}$ is reached, an interval interrupt can be generated.

9.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 9-1 Configuration of watchdog timer

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

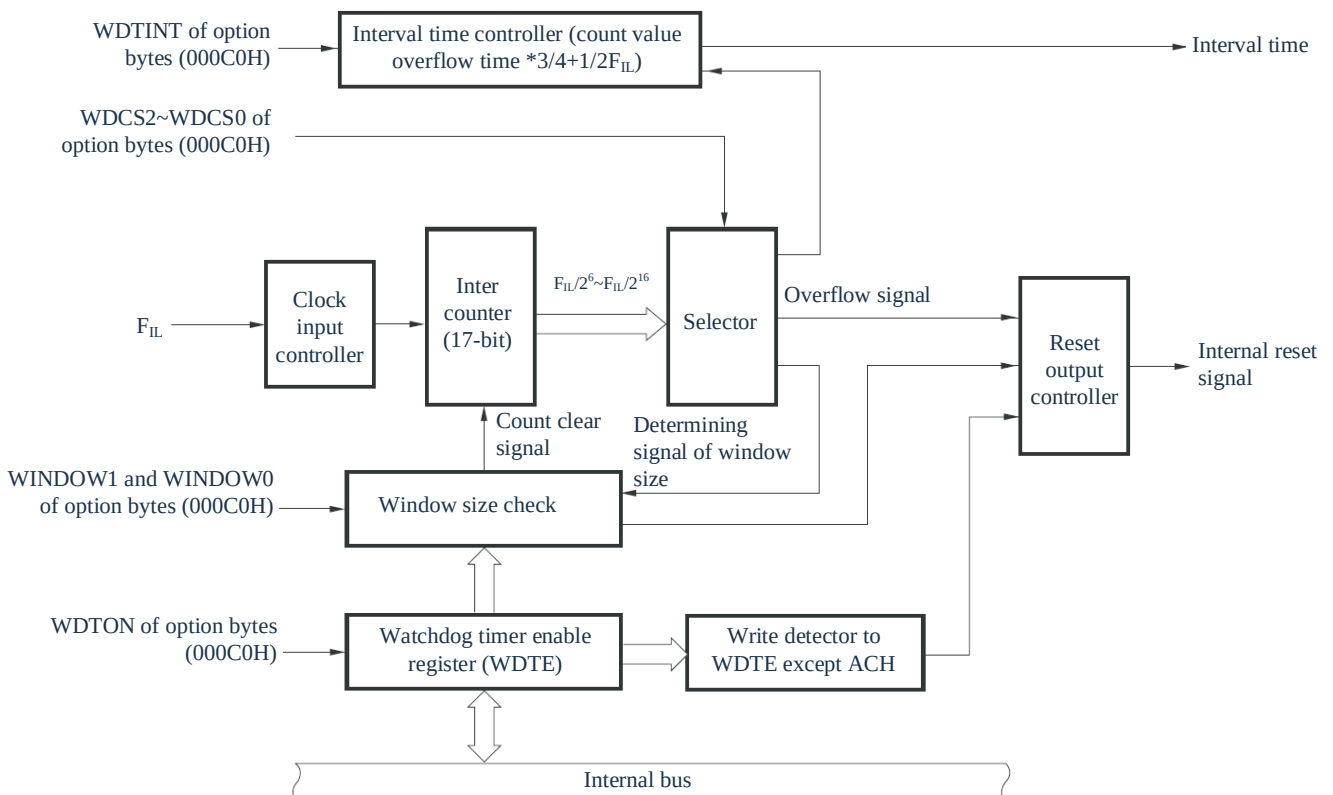
The operation of the counter is controlled by the option byte as well as the setting of the overflow time, the window opening period and the interval interrupt.

Table 9-2 Setting of option bytes and watchdog timer

Setting of watch dog timer	Option byte (000C0H)
Watchdog timer interval interrupt	bit7 (WDTINT)
Window open period	bit6 and bit5 (WINDOW1, WINDOW0)
Controlling counter operation of watchdog timer	bit4 (WDTON)
Overflow time of watchdog timer	bit3~1 (WDCS2~WDCS0)
Controlling counter operation of watchdog timer (in sleep mode)	bit0 (WDSTBYON)

Remark: For option byte, see Chapter 25 Option Byte.

Figure 9-1 Block diagram of watchdog timer



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

9.3 Registers Controlling Watch Dog Timer

The watchdog timer is controlled through the watchdog timer enable register (WDTE).

9.3.1 Watch Dog Timer Enable Register (WDTE)

Writing “ACH” to the WDTE register clears the watchdog timer counter and starts counting again. The WDTE register is set by an 8-bit memory manipulation instruction. Reset signal generation sets this register to “9AH” or “1AH”^{Note}.

Figure 9-2 Format of watch dog timer enable register (WDTE)

Address: 0x40021001	After reset: 9AH/1AH ^{Note}	R/W						
Symbol	7	6	5	4	3	2	1	0
WDTE								

Note: The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

WDTON bit setting value	WDTE register reset value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Caution 1: If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.

Caution 2: When a bitwise operation is performed on the WDTE register, an internal reset signal is generated.

Caution 3: The value read from the WDTE register is 9AH/1AH (this differs from the written value (ACH)).

9.3.2 LOCKUP Control Register (LOCKCTL) and Its Protection Register (PRCR)

The LOCKCTL register is a configuration register for controlling the Cortex-M0+ LockUp function to operate the watchdog timer, and PRCR is its write-protect register.

The LOCKCTL and PRCR registers are set by 8-bit memory manipulation instructions.

After a reset signal is generated, the value of the LOCKCTL register becomes “01H”, and the value of the PRCR register becomes “00H”.

Figure 9-3 Format of LOCKUP control register (LOCKCTL) and its protection register (PRCR)

Address: 4002040CH

After reset: 01H

R/W

LOCKCTL

0	0	0	0	0	0	0	lockup_rst

lockup_rst	LOCKUP function configuration
0	• LOCKUP does not trigger a WDT reset
1	• LOCKUP triggers a WDT reset

Address: 40020406H

After reset: 00H

R/W

PRCR

PRTKEY[7:1]		PRCR

PRCR	LOCKUP control register write protection
0	• LOCKCTL register is not writable
1	• LOCKCTL register is writable

PRTKEY[7:1]	PRCR write protection
78H	• PRCR is writable
Other	• PRCR is not writable

Address: 40020406H	After reset: 00H	R/W												
PRCR	<table><tr><td colspan="2">PRTKEY[7:1]</td><td>PRCR</td></tr><tr><td>PRCR</td><td colspan="2">LOCKUP control register write protection</td></tr><tr><td>0</td><td colspan="2">• LOCKCTL register is not writable</td></tr><tr><td>1</td><td colspan="2">• LOCKCTL register is writable</td></tr></table>		PRTKEY[7:1]		PRCR	PRCR	LOCKUP control register write protection		0	• LOCKCTL register is not writable		1	• LOCKCTL register is writable	
PRTKEY[7:1]		PRCR												
PRCR	LOCKUP control register write protection													
0	• LOCKCTL register is not writable													
1	• LOCKCTL register is writable													
<table><tr><td>PRTKEY[7:1]</td><td>PRCR write protection</td></tr><tr><td>78H</td><td>• PRCR is writable</td></tr><tr><td>Other</td><td>• PRCR is not writable</td></tr></table>			PRTKEY[7:1]	PRCR write protection	78H	• PRCR is writable	Other	• PRCR is not writable						
PRTKEY[7:1]	PRCR write protection													
78H	• PRCR is writable													
Other	• PRCR is not writable													

9.3.3 Watchdog Configuration Register (WDTCFG0/1/2/3)

The WDTCFGx configuration register is a register that forces the watchdog timer to operate or not.

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

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

Figure 9-4 WDTCFG configuration register (WDTCFG0/1/2/3)

Address: 40020408H	After reset: 00H	R/W
WDTCFG0	WDTCFG0	
Address: 40020409H	After reset: 00H	R/W
WDTCFG1	WDTCFG1	
Address: 4002040AH	After reset: 00H	R/W
WDTCFG2	WDTCFG2	
Address: 4002040BH	After reset: 00H	R/W
WDTCFG3	WDTCFG3	

WDTCFG0	WDTCFG1	WDTCFG2	WDTCFG3	Configuration of the watchdog timer function
0x1A	0x2B	0x3C	0x4D	• Watchdog timer forced stop
Other				• Watchdog timer forced start

9.4 Operation of Watchdog Timer

9.4.1 Operational Control of Watchdog Timer

1. When using the watchdog timer, set the following items by option byte (000C0H):
 - The bit 4 (WDTON) of the option byte (000C0H) must be set to 1 to enable the watchdog timer count to operate (the counter starts operating after the reset is released) (refer to Chapter 25 Option Byte for details).
2. After the reset is released, the watchdog timer starts counting.
3. After starting counting and before the overflow time set by the option byte, writing “ACH” to the watchdog timer enable register (WDTE) clears the watchdog timer and starts counting again.
4. Thereafter, writes to WDTE registers after the second time after the reset must be performed while the window is open. If you write the WDTE register while the window is closed, an internal reset signal is generated.
5. If you do not write “ACH” to the WDTE register and exceed the overflow time, an internal reset signal is generated. An internal reset signal is generated if:
 - When a bitwise operation is performed on the WDTE register.
 - If data other than “ACH” is written to the WDTE register

Note 1: When data is written to the watchdog timer enable register (WDTE) for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

Note 2: After “ACH” is written to the WDTE register, an error of up to 2 F_{IL} clocks may occur before the watchdog timer is cleared.

Note 3: The watchdog timer can be cleared immediately before the count value overflows.

Note 4: As shown below, the watchdog timer operates in sleep or deep sleep mode depending on the set value of bit0 (WDSTBYON) of the option byte (000C0H).

	WDSTBYON=0	WDSTBYON=1
Sleep mode	Stop operation of watchdog timer.	Continue operation of watchdog timer.
Deep sleep mode		

When the WDSTBYON bit is “0”, restart the watchdog timer count after the sleep or deep sleep mode released. At this point, the counter is cleared to “0” and the count begins.

When releasing from deep sleep mode and operating with the X1 oscillator clock, the CPU begins operation after the oscillator stabilization time.

If the time from releasing deep sleep mode to the watchdog timer overflow is short, the watchdog overflow may occur during the oscillator stabilization time, triggering a reset. Therefore, after releasing deep sleep mode via an interval interrupt, if the system is running with the X1 oscillator clock and the watchdog timer is cleared, you must account for this scenario where the watchdog timer is cleared after the oscillator stabilization time, and the overflow time should be appropriately set.

9.4.2 Setting Overflow Time of Watchdog Timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H).

If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to the watchdog timer enable register (WDTE) during the window open period before the overflow time. The following overflow times can be set.

Table 9-3 Setting of overflow time of watchdog timer

WDCS2	WDCS1	WDCS0	Overflow time of watchdog timer (When $F_{IL}=40\text{kHz}$ (Max.))
0	0	0	$2^6/F_{IL}$ (1.6ms)
0	0	1	$2^7/F_{IL}$ (3.2ms)
0	1	0	$2^8/F_{IL}$ (6.4ms)
0	1	1	$2^9/F_{IL}$ (12.8ms)
1	0	0	$2^{11}/F_{IL}$ (51.2ms)
1	0	1	$2^{13}/F_{IL}$ (204.8ms)
1	1	0	$2^{14}/F_{IL}$ (409.6ms)
1	1	1	$2^{16}/F_{IL}$ (1638.4ms)

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

9.4.3 Setting Window Open Period of Watchdog Timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows:

- If “ACH” is written to the watchdog timer enable register (WDTE) during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to the WDTE register during the window close period, an abnormality is detected and an internal reset signal is generated.

Caution: When data is written to the WDTE register for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

The window open period can be set is as follows.

Table 9-4 Setting window open period of watchdog timer

WINDOW1	WINDOW0	Window open period of watchdog timer
0	-	Settings are disabled.
1	0	75%
1	1	100%

Caution: When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of the WINDOW1 and WINDOW0 bits.

If the overflow time is set to $2^9/F_{IL}$, the window close time and open time are as follows.

	Setting of window open period	
	75%	100%
Window close time	0~6.4ms	None
Window open time	6.4~12.8ms	0~12.8ms

<When window open period is 75%>

- Overflow time: $2^9/F_{IL}(\text{MAX.})=2^9/40\text{kHz}(\text{MAX.})=12.8\text{ms}$
- Window close time: $0\sim 2^9/F_{IL}(\text{MIN.})\times(1-0.75)=0\sim 2^9/20\text{kHz}\times 0.25=0\sim 6.4\text{ms}$
- Window open time: $2^9/F_{IL}(\text{MIN.})\times(1-0.75)\sim 2^9/F_{IL}(\text{MAX.})=6.4\sim 12.8\text{ms}$

9.4.4 Setting Watchdog Timer Interval Interrupt

Depending on the setting of bit 7 (WDTINT) of an option byte (000C0H), an interval interrupt (INTWDTI) can be generated when $75\% + 1/2F_{IL}$ of the overflow time is reached.

Table 9-5 Setting of watchdog timer interval interrupt

WDTINT	Use of watchdog timer interval interrupt
0	Interval interrupt is not used.
1	Interval interrupt is generated when $75\% + 1/2F_{IL}$ of the overflow time is reached.

Caution: When releasing deep sleep mode and operating with the X1 oscillator clock, the CPU begins operation after the oscillator stabilization time. If the time from exiting deep sleep mode to the watchdog timer overflow is short, the watchdog overflow may occur during the oscillator stabilization time, causing a reset. Therefore, after releasing deep sleep mode through an interval interrupt, if the system is to run with the X1 oscillator clock and clear the watchdog timer, since the watchdog timer is cleared only after the oscillator stabilization time, this scenario must be considered when setting the overflow time.

Remark: The watchdog timer continues counting even after a watchdog interval interrupt is generated (until “ACH” is written to the watchdog timer enable register (WDTE)). If “ACH” is not written to the WDTE register before the overflow time, an internal reset signal is generated.

Chapter 10 A/D Converter

The number of analog input channels of the A/D converter varies from product to product.

Pins	20 pins	24 pins
Analog input channels	11ch	15ch
	(ANI0, ANI1, ANI4~ANI9, ANI12~ANI14)	(ANI0~ANI14)

10.1 Functions of A/D Converter

The A/D converter is a device that converts analog inputs into digital values. It can control up to 18 analog channels for A/D conversion (15 external input channels and 3 internal channels).

The A/D converter has the following features:

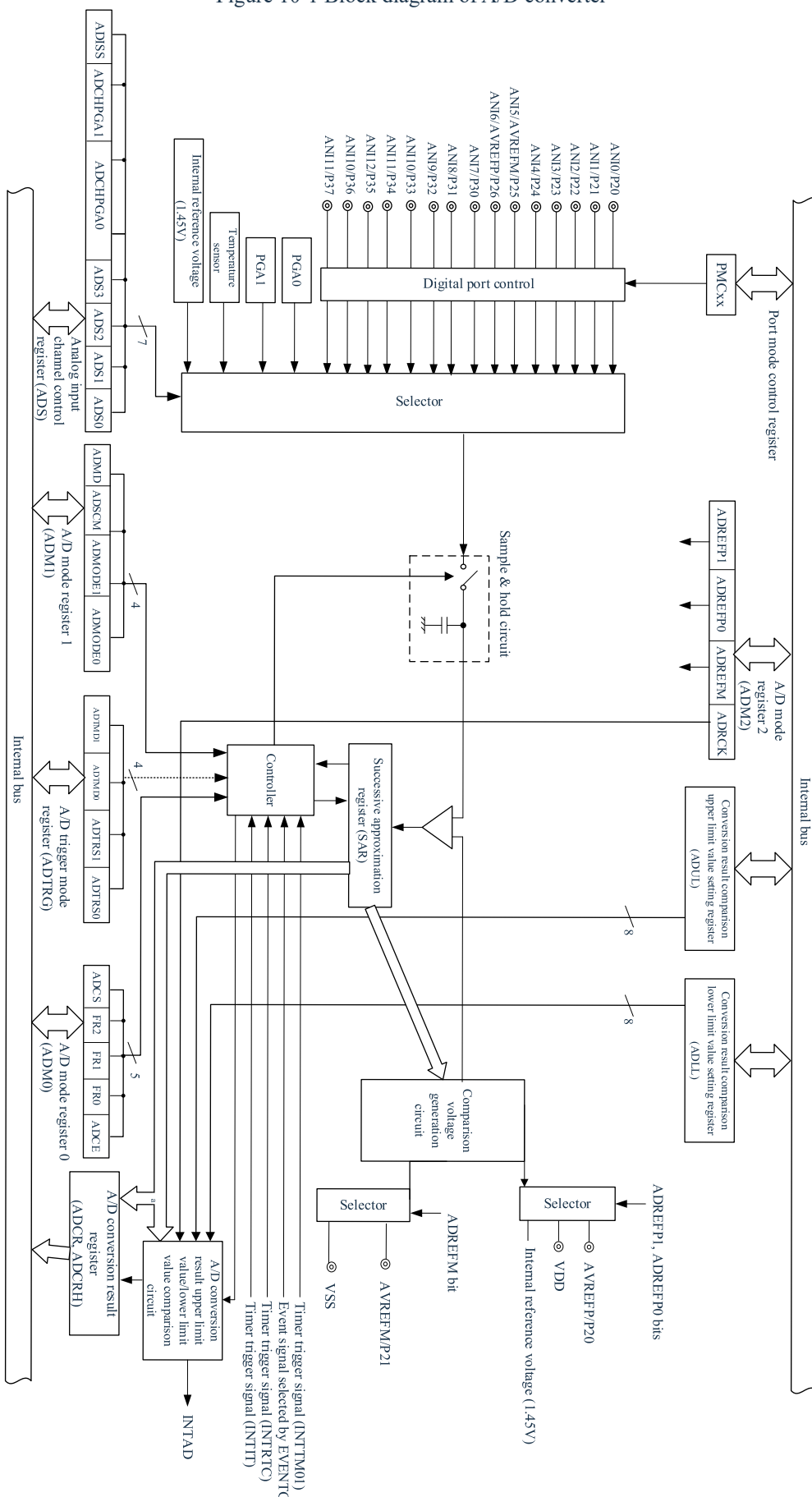
- A/D conversion with 12-bit resolution

It selects one analog input channel from ANI0 to ANI14, PGA0, PGA1, or the temperature sensor, and performs 12-bit resolution A/D conversion repeatedly. After each A/D conversion, an interrupt request (INTAD) is generated (depending on the selected mode).

Various A/D conversion modes are set by the combination of modes as described below.

Trigger mode	Software trigger	Start the conversion with software operations.
	Hardware trigger no-wait mode	Conversion is started by detecting a hardware trigger.
	Hardware trigger wait mode	In the conversion standby state where the A/D power supply is cut off, the power supply is turned on by detecting a hardware trigger, and the conversion starts automatically after the A/D power supply stabilization waiting time.
Channel selection mode	Select mode	Select 1 channel of analog input for A/D conversion.
	Scan mode	Perform A/D conversion of 4 channels of analog input in sequence. Can select 4 consecutive channels from ANI0 to ANI14 as analog inputs.
Conversion mode	Single conversion mode	Perform 1 A/D conversion for the selected channel.
	Consecutive conversion mode	Perform continuous A/D conversion for the selected channel until stopped by the software.
Sampling time	Sampling clock 5.5~255 ADCLKs	The sampling time can be selected via the ADNSMP register, which uses 13.5 conversion clocks (f_{AD}) by default.

Figure 10-1 Block diagram of A/D converter



10.2 Registers for Controlling A/D Converter

The A/D converter is controlled by the following registers:

Base address:

CSC_BASE = 4002_0420H;

ADC_BASE = 4004_5000H;

PORT_BASE = 4004_0000H

Register name	Register description	R/W	Reset value	Register address
PER0	Peripheral enable register 0	R/W	00H	CSC_BASE+20H
ADM0	A/D converter mode register 0	R/W	00H	ADC_BASE+00H
ADM1	A/D converter mode register 1	R/W	00H	ADC_BASE+02H
ADM2	A/D converter mode register 2	R/W	00H	ADC_BASE+04H
ADTRG	A/D converter trigger mode register	R/W	00H	ADC_BASE+06H
ADS	Analog input channel specification register	R/W	00H	ADC_BASE+08H
ADLL	Conversion result comparison lower limit setting register	R/W	00H	ADC_BASE+0AH
ADUL	Conversion result comparison upper limit setting register	R/W	00H	ADC_BASE+0BH
ADNSMP	A/D converter sampling time control register	R/W	0dH	ADC_BASE+0CH
ADCR	12-bit A/D conversion result register	R	0000H	ADC_BASE+0EH
ADCRH	8-bit A/D conversion result register	R	00H	ADC_BASE+0FH
ADTES	A/D test register	R/W	00H	ADC_BASE+10H
ADNDIS	A/D converter charge/discharge control register	R/W	00H	ADC_BASE+11H
ADSMPWAIT	A/D converter sampling time extension control register	R/W	00H	ADC_BASE+15H
ADFLG	A/D hardware module status register	R	00H	ADC_BASE+16H
PMC2	Port mode control register 2	R/W	FCH	PORT_BASE+62H
PMC3	Port mode control register 3	R/W	1FH	PORT_BASE+63H

10.2.1 Peripheral Enable Register 0 (PER0)

The PER0 register is the register that sets whether to enable or disable the supply of clocks to each peripheral hardware. Reduce power consumption and noise by stopping clocks to hardware that is not in use.

To use the A/D converter, bit5 (ADCEN) must be set to 1.

The PER0 register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes “00H”.

Figure 10-2 Format of peripheral enable register 0 (PER0)

Reset value: 00H	R/W							
	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICA0EN	SCI1EN	SCI0EN	TM41EN	TM40EN

ADCEN	Control of input clock for A/D converter
0	Stop input clock supply. - SFR used by the A/D converter cannot be written. - The A/D converter is in the reset status.
1	Enable input clock supply. SFR used by the A/D converter can be read and written.

Caution 1. When setting the A/D converter, be sure to set the following registers first while the ADCEN bit is set to 1. If ADCEN = 0, the values of the A/D converter control registers are cleared to their initial values and writing to them is ignored (except for port mode control register (PMCxx)).

- A/D converter mode register 0 (ADM0)
- A/D converter mode register 1 (ADM1)
- A/D converter mode register 2 (ADM2)
- A/D converter trigger mode register (ADTRG)
- Analog input channel specification register (ADS)
- Conversion result comparison lower limit setting register (ADLL)
- Conversion result comparison upper limit setting register (ADUL)
- A/D converter sampling time control register (ADNSMP)
- 12-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- A/D test register (ADTES)
- A/D converter charge/discharge control register (ADNDIS)
- A/D converter sampling time extension control register (ADSMPWAIT)
- A/D hardware module status register (ADFLG)

10.2.2 A/D Converter Mode Register 0 (ADM0)

This register sets the clock for A/D converter, and starts/stops conversion. The ADM0 register is set by an 8-bit memory manipulation instruction.

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

Figure 10-3 Format of A/D converter mode register 0 (ADM0)

Reset value: 00H R/W

	7	6	5	4	3	2	1	0
ADM0	ADCS	0	FR2	FR1	FR0	0	0	ADCE

ADCS	A/D conversion operation control
0	Stop conversion [When reading] Stop conversion/standby status
1	Enable conversion [When reading] Software trigger mode: Conversion operation status Hardware trigger wait mode: A/D power supply stabilization wait status+conversion operation status

ADCE	A/D voltage comparator operation control
0	Stop A/D voltage comparator operation.
1	Enable A/D voltage comparator operation.

Notes:

- For details of the FR2~FR0 bits and A/D conversion, please refer to “Table 10-3 A/D Conversion Time Selection”.
- It takes 1 μ s 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 μ s 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 1 μ s is guaranteed by design.

Cautions:

- Change the FR2~FR0 bits while conversion is stopped (ADCS = 0).
- Do not set the ADCS bit to 1 and the ADCE bit to 0.
- Do not change the ADCS and ADCE bits from 0 to 1 by using an 8-bit manipulation instruction. You must follow the procedure in “10.5 A/D converter setup flowchart”.

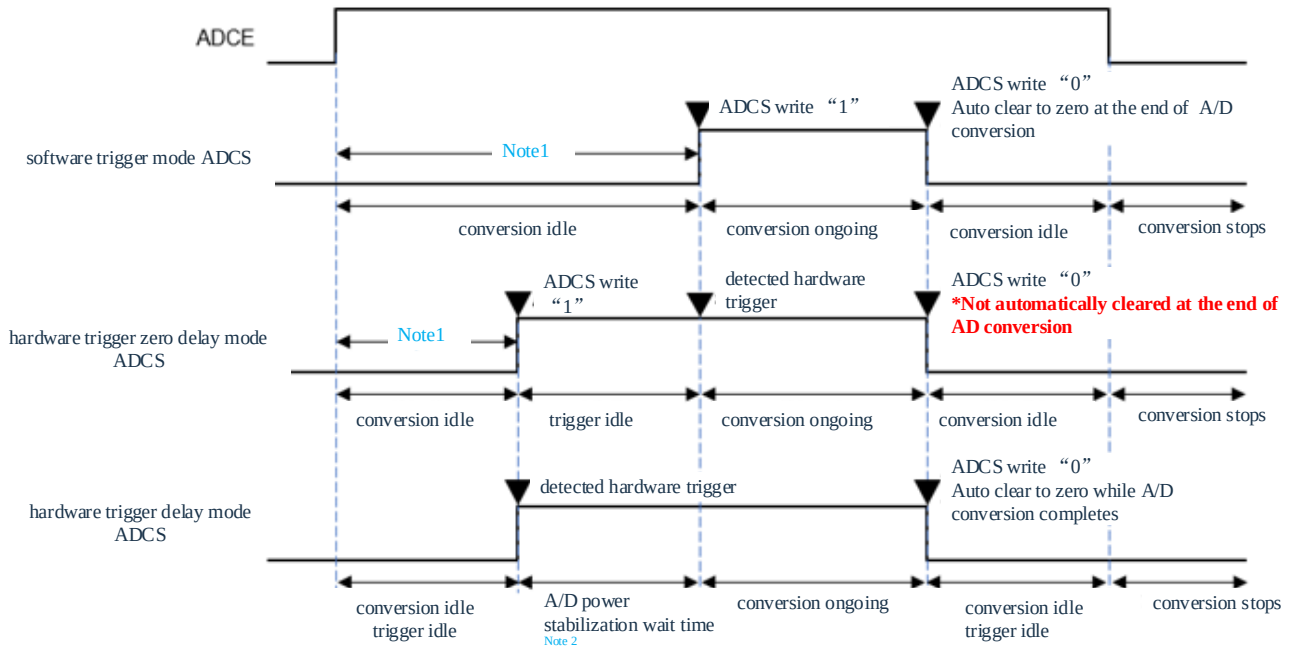
Table 10-1 ADCS bit and ADCE bit settings

ADCS	ADCE	A/D conversion operation
0	0	Conversion stop state
0	1	Conversion idle state
1	0	Prohibit settings.
1	1	Conversion operating state

Table 10-2 ADCS bit set and clear conditions

A/D conversion mode			Set condition	Clear condition
Software trigger	Select mode	Continuous conversion mode	When write “1” to the ADCS bit	When writing “0” to the ADCS bit
		Single conversion mode		- When writing “0” to the ADCS bit - Automatically clears to “0” 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 - Automatically clears to “0” when the set 4-channel conversion is completed.
Hardware trigger no-wait mode	Select mode	Continuous conversion mode	When write “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 triggered by input hardware	When writing “0” to the ADCS bit
		Single conversion mode		- When writing “0” to the ADCS bit - Automatically clears to “0” 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 - Automatically clears to “0” when the set 4-channel conversion is completed.

Figure 10-4 Diagram of using various modes of A/D



Note 1: In software trigger mode or hardware trigger no-wait mode, it takes at least 1us (TBD) to rise from the ADCE bit to the ADCS bit to stabilize the internal circuitry.

Note 2: In hardware trigger wait mode, the A/D power supply stabilization time of 1us is guaranteed by design.

Cautions:

1. To use the hardware trigger wait mode, setting the ADCS bit to 1 is prohibited (it is automatically switched to 1 when a hardware trigger signal is detected). However, the ADCS bit can be set to 0 to set the standby state for A/D conversion.
2. The ADCE bit must be overridden when the ADCS bit is 0 (Stop Transition/Transition Standby).
3. To end the A/D conversion, the hardware trigger interval must be set at least to the following time:
Hardware trigger no-wait mode: $2 F_{CLK} \text{ clocks} + \text{A/D conversion time}$
Hardware trigger wait mode: $2 F_{CLK} \text{ clocks} + \text{A/D power supply stable wait time} + \text{A/D conversion time}$

Remark: f_{CLK} : CPU/peripheral hardware clock frequency

Table 10-3 Selection of A/D conversion time (1/2)

(1) No A/D power stabilization wait time (software trigger mode/hardware trigger no wait mode)

A/D converter mode register 0 (ADM0)			A/D converter mode register 1 (ADM1) ^{Note 1}		Mode	Frequency of conversion clock ADCLK (F _{AD})	12-bit resolution conversion time ^{Note2}	
							ADC conversion time = (number of sample clocks + number of successive comparison clocks)/ F _{AD}	
FR2	FR1	FR0	ADMODE[1]	ADMOD[0]			ADC conversion clock number (13.5 sample clocks)	ADC conversion time (13.5 sample clocks)
0	0	0	0	0	High-speed conversion mode	f _{CLK} /32	45 ADCLK (13.5 sample clocks +31.5 successive comparison clocks)	45/ f _{AD}
0	0	1				f _{CLK} /16		
0	1	0				f _{CLK} /8		
0	1	1				f _{CLK} /4		
1	0	0				f _{CLK} /2		
1	0	1				f _{CLK} /1		
0	0	0	1	1	Low-current mode	f _{CLK} /32	54 ADCLK (13.5 sample clocks +40.5 successive comparison clocks)	54/ f _{AD}
0	0	1				f _{CLK} /16		
0	1	0				f _{CLK} /8		
0	1	1				f _{CLK} /4		
1	0	0				f _{CLK} /2		
1	0	1				f _{CLK} /1		

Note 1. To override the FR2~FR0 bits and ADMODE[1:0] bits into different data, it must be done in the conversion stop state (ADCS=0).

Note 2. Time required to perform an ADC conversion = (number of sample clocks + number of successive comparison clocks) / F_{AD}

The number of sampling clocks can be adjusted through the ADNSMP register, the default is 13.5 ADCLK, the number of successive comparison clocks is determined by the conversion mode, the high-speed conversion mode is 31.5 ADCLK, the low current mode is 40.5 ADCLK, the fastest clock supported by the ADCLK is 64MHz in the high-speed conversion mode, the fastest clock supported by the ADCLK is 27MHz in the low current mode. For actual use, please configure the conversion mode and conversion clock frequency according to the “AC Characteristics” in the datasheet.

Remark f_{CLK} : CPU/peripheral hardware clock frequency

Table 10-4 Selection of A/D conversion time (2/2)
(2) With A/D power stabilization wait time (hardware trigger wait mode^{Note1})

A/D converter mode register 0 (ADM0)			A/D converter mode register 1 (ADM1)		Mode	Frequency of conversion clock ADCLK (F _{AD})	A/D power supply stabilization wait time	ADC conversion clock number	A/D power stabilization Waiting time +ADC conversion time ^{Note 2}
FR2	FR1	FR0	ADM0[1]	ADM0[0]					
0	0	0	0	0	High-speed conversion mode	f _{CLK} /32	1us	45 ADCLK (13.5 sample clocks +31.5 successive comparison clocks)	1us +45/f _{AD}
0	0	1				f _{CLK} /16			
0	1	0				f _{CLK} /8			
0	1	1				f _{CLK} /4			
1	0	0				f _{CLK} /2			
1	0	1				f _{CLK} /1			
0	0	0	1	1	Low-current mode	f _{CLK} /32	1us	54 ADCLK (13.5 sample clocks +40.5 successive comparison clocks)	1us +54/f _{AD}
0	0	1				f _{CLK} /16			
0	1	0				f _{CLK} /8			
0	1	1				f _{CLK} /4			
1	0	0				f _{CLK} /2			
1	0	1				f _{CLK} /1			

Note 1. In hardware trigger wait mode, the power supply settling time is guaranteed by the hardware design and does not need to be set. And in continuous conversion mode, the A/D power stabilization wait time occurs only after the hardware trigger is detected for the first time (see Table 10-4).

Note 2. Time required for ADC conversion after hardware triggering =1us + (number of sample clocks + number of successive comparison clocks)/F_{AD}. The number of sample clocks can be adjusted through the ADNSMP register, the default is 13.5 ADCLK, the number of successive comparison clocks is determined by the conversion mode, it is 31.5 ADCLK in high speed conversion mode, and 40.5 ADCLK in low current mode, the fastest clock supported by ADCLK is 64MHz in high speed conversion mode, and the fastest clock supported by ADCLK is 27MHz in low current mode. For actual use, configure the conversion mode and conversion clock frequency according to the “AC Characteristics” in the datasheet.

Cautions

1. To override the FR2~FR0 bits and ADM0[1:0] bits into different data, it must be done in the conversion stop state (ADCS=0).
2. The conversion time in the hardware trigger wait mode includes the A/D power stabilization wait time after the hardware trigger is detected.

Remark: f_{CLK}: CPU/peripheral hardware clock frequency

10.2.3 A/D Converter Mode Register 1 (ADM1)

This is a register that sets the A/D conversion mode.

The ADM1 register is set via an 8-bit memory manipulation instruction.

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

Figure 10-5 Format of A/D converter mode register 1 (ADM1)

Reset value: 00H R/W

	7	6	5	4	3	2	1	0
ADM1	ADMD	0	0	0	ADSCM	0	ADMODE1	ADMODE0

ADMD	Setting of the A/D conversion channel selection mode
0	Select mode
1	Scan mode

ADSCM	Setting of the A/D conversion mode
0	Continuous conversion mode
1	Single conversion mode

ADMODE1	ADMODE0	A/D conversion mode
0	0	High-speed conversion mode (ADCLK fastest clock is 64MHz)
1	1	Low current mode (ADCLK fastest clock is 27 MHz)
Other		Setting disabled

Caution Bits 6 to 4, 2 must be set to “0”.

Cautions

1. To override the ADM1 register, it must be done in the conversion stop state (ADCS=0).
2. To end the A/D conversion normally, the hardware trigger interval must be set at least to the following time:
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 stable wait time+A/D conversion time
3. For A/D conversion, the number of successive comparison clocks is determined by the conversion mode, which is 31.5 ADCLK for high speed conversion mode and 40.5 ADCLK for low current mode, and the fastest clock supported by ADCLK is 64 MHz for high speed conversion mode and 27 MHz for low current mode. For actual use, please configure the conversion mode and conversion clock frequency according to the “AC Characteristics” in the datasheet.

Remark 1. f_{CLK} : CPU/peripheral hardware clock frequency

10.2.4 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 becomes “00H”.

Figure 10-6 Format of A/D converter mode register 2 (ADM2)

Reset value: 00H R/W

	7	6	5	4	3	2	1	0
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	0	CHRDE	0

ADREFP1	ADREFP0	Selection of positive (+) voltage references for A/D converters
0	0	Provided by V _{DD} .
0	1	Provided by the AV _{REFP} external terminal.
1	0	Provided by internal reference voltage (1.45V).
1	1	Setting disabled.

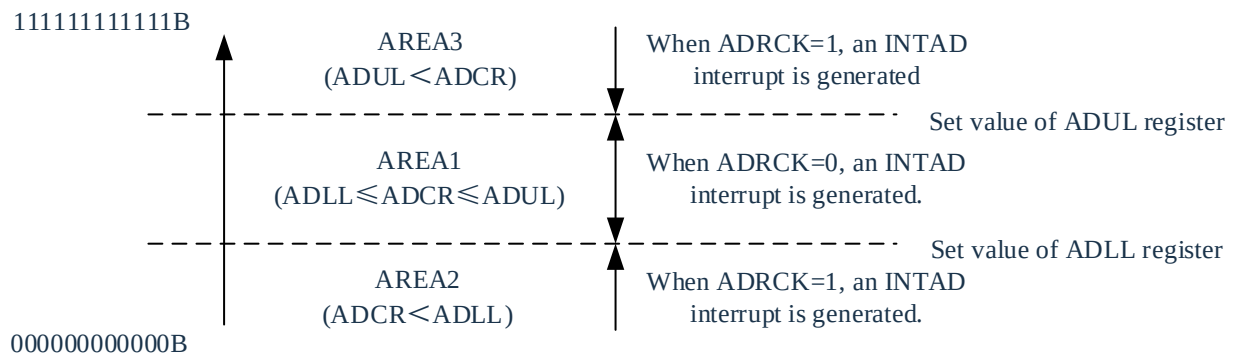
ADREFM	Selection of negative (-) voltage references for A/D converters
0	Provided by V _{SS} .
1	Provided by the AV _{REFM} external terminal.

ADRCK	Checking of upper and lower limit values of conversion results
0	When ADLL register ≤ ADCR register ≤ ADUL register (AREA1), an interrupt signal (INTAD) is generated.
1	When ADCR register < ADLL register (AREA2) or ADUL register < ADCR register (AREA3), an interrupt signal (INTAD) is generated.
The range of interrupt signal (INTAD) generated from AREA1 to AREA3 is shown in Figure 15-8.	

CHRDE	Output enable for channel identification during A/D converter scan mode
0	In scanning mode, the channel number is not identified in the conversion results
1	In scanning mode, the high four bits of the converted result ([15:12] of the ADCR register) are the channel numbers for this result

Figure 10-7 Range of interrupt signal generation for the ADRCK bit

ADCR register value
(A/D conversion result)



Cautions

1. To override the ADM2 register, it must be done in the conversion stop state (ADCS=0).
2. When using AV_{REFP} and AV_{REFM}, the port must be set to analog port mode (PMCxx=1).

Remark: When INTAD does not occur, the A/D conversion results are not saved to the ADCR register and the ADCRH register.

10.2.5 A/D Converter Trigger Mode Register (ADTRG)

This is a register that sets the A/D conversion trigger mode and the hardware trigger signal.

The ADTRG register is set via an 8-bit memory manipulation instruction.

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

Figure 10-8 Format of A/D converter trigger mode register (ADTRG)

Reset value: 00H R/W

	7	6	5	4	3	2	1	0
ADTRG	ADTMD1	ADTMD0	0	0	0	0	ADTRS1	ADTRS0

ADTMD1	ADTMD0	Selection of A/D conversion trigger modes
0	0	Software trigger mode
0	1	
1	0	Hardware trigger no-wait mode
1	1	Hardware trigger wait mode

ADTRS1	ADTRS0	Selection of hardware trigger signals
0	0	The counting end of timer channel 1 or the capture end of interrupt signal (INTTM01)
0	1	The event signal selected by ELC
1	0	Real-time clock interrupt signal (INTRTC)
1	1	Interval timer interrupt signal (INTIT)

Cautions

1. To override the ADTRG register, it must be done in the conversion stop state (ADCS=0, ADCE=0).
2. To end the A/D conversion normally, the hardware trigger interval must be set at least to the following time:
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 stable wait time + A/D conversion time

Remark 1. f_{CLK} : CPU/peripheral hardware clock frequency

10.2.6 Analog Input Channel Specification Register (ADS)

This is a register that specifies the analog voltage input channel for A/D converter.

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

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

Figure 10-9 Format of analog input channel specification register (ADS)

Reset value: 00H R/W

	7	6	5	4	3	2	1	0
ADS	ADISS	0	ADCHPGA1	ADCHPGA0	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADM1.ADMD=0)

ADS register setting value							CH selection
ADISS	ADCHPGA1	ADCHPGA0	ADS3	ADS2	ADS1	ADS0	
0	0	0	0	0	0	0	ANI0(P20)
0	0	0	0	0	0	1	ANI1(P21)
0	0	0	0	0	1	0	ANI2(P22)
0	0	0	0	0	1	1	ANI3(P23)
0	0	0	0	1	0	0	ANI4(P24)
0	0	0	0	1	0	1	ANI5(P25)
0	0	0	0	1	1	0	ANI6(P26)
0	0	0	0	1	1	1	ANI7(P30)
0	0	0	1	0	0	0	ANI8(P31)
0	0	0	1	0	0	1	ANI9(P32)
0	0	0	1	0	1	0	ANI10(P33)
0	0	0	1	0	1	1	ANI11(P34)
0	0	0	1	1	0	0	ANI12(P35)
0	0	0	1	1	0	1	ANI13(P36)
0	0	0	1	1	1	0	ANI14(P37)
0	0	0	1	1	1	1	ANI0~ANI14 channels OFF
1	0	0	0	0	0	0	Temperature sensor output voltage Note 1
1	0	0	0	0	0	1	Internal reference voltage (1.45V)
0	1	0	0	0	0	0	PGA1
0	0	1	0	0	0	0	PGA0
Other settings are prohibited							

Note 1: If the internal reference voltage (1.45V) is selected as the reference voltage for comparator 0 or comparator 1, the temperature sensor output cannot be selected.

○ Scan mode (ADM1.ADMD=1)

ADISS	ADS3	ADS2	ADS1	ADS0	Analog input channel			
					Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
0	0	1	0	1	ANI5	ANI6	ANI7	ANI8
0	0	1	1	0	ANI6	ANI7	ANI8	ANI9
0	0	1	1	1	ANI7	ANI8	ANI9	ANI10
0	1	0	0	0	ANI8	ANI9	ANI10	ANI11
0	1	0	0	1	ANI9	ANI10	ANI11	ANI12
0	1	0	1	0	ANI10	ANI11	ANI12	ANI13
0	1	0	1	1	ANI11	ANI12	ANI13	ANI14
Others					Settings are disabled.			

Cautions

1. Bit4, bit5 and bit6 must be set to 0 in scan mode.
2. For the port set as analog input by PMCx register, it can be designated as analog input for A/D conversion by ADS.
3. For pins set as digital input/output by the Port Mode Control Register (PMCxx), they cannot be set by the ADS register.
4. To rewrite the ADISS bit, it must be done in the conversion stop state (ADCS=0, ADCE=0).
5. When AV_{REFP} is used as the positive (+) reference voltage for the A/D converter, ANI6 cannot be selected as the A/D converter channel.
6. When AV_{REFM} is used as the negative (-) reference voltage for the A/D converter, ANI5 cannot be selected as the A/D converter channel.
7. After setting the ADISS bit to 1, the result of the first conversion cannot be used. For details of the setting procedure, refer to 10.5.4 Setting for Selecting the Temperature Sensor Output Voltage/Internal Reference Voltage.
8. The ADISS bit cannot be set to 1 when transferring to deep sleep mode or when transferring to sleep mode while the CPU is running at the sub-system clock.

10.2.7 12-Bit A/D Conversion Result Register (ADCR)

This is a 16-bit register that stores the A/D conversion result, and this register is read-only. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR)^{Note}.

The high 4 bits of this register are fixed to 0 when the mode is selected, and the channel number of the conversion result can be configured by ADM2.CHRDE=1 in scan mode.

The ADCR register can be read by a 16-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes “0000H”.

Note: If the value of the A/D conversion result is not within the set value range of the A/D conversion result comparison function (set by the ADRCK bit and the ADUL/ADLL register, refer to Figure 10-7), the A/D conversion results are not saved.

Figure 10-10 Format of 12-bit A/D conversion result register (ADCR)

Reset value: 0000H R

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCR	ADCH3	ADCH2	ADCH1	ADCH0	ADCR[11:0]											

Cautions

1. If only 8-bit resolution A/D conversion results are required, the higher 8 bits of the conversion result can be read by the ADCRH register.
2. When 16 bits of access are made to the ADCR register, the higher 12 bits of the conversion result can be read sequentially from bit11.

○Select mode (ADM1.ADMD=0)

ADCH0~3 read value is fixed at 4'b0000

○Scan mode (ADM1.ADMD=1) and ADM2.CHRDE=1, the relationship between the readout values of ADCH0~3 and the converted channels is as follows

ADCH3	ADCH2	ADCH1	ADCH0	Conversion channel ID
0	0	0	0	ANI0(P20)
0	0	0	1	ANI1(P21)
0	0	1	0	ANI2(P22)
0	0	1	1	ANI3(P23)
0	1	0	0	ANI4(P24)
0	1	0	1	ANI5(P25)
0	1	1	0	ANI6(P26)
0	1	1	1	ANI7(P30)
1	0	0	0	ANI8(P31)
1	0	0	1	ANI9(P32)
1	0	1	0	ANI10(P33)
1	0	1	1	ANI11(P34)
1	1	0	0	ANI12(P35)
1	1	0	1	ANI13(P36)
1	1	1	0	ANI14(P37)

10.2.8 8-Bit A/D Conversion Result Register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 12-bit resolution are stored^{Note}.

The ADCRH register can be read by an 8-bit memory manipulation instruction.

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

Note: If the value of the A/D conversion result is not within the set value range of the A/D conversion result comparison function (set by the ADRCK bit and the ADUL/ADLL register), the A/D conversion results are not saved.

Figure 10-11 Format of 8-bit A/D conversion result register (ADCRH)

Reset value: 00H R

	7	6	5	4	3	2	1	0
ADCRH								

Caution Read the conversion result following conversion completion before writing to the ADM0, ADS registers. Otherwise, you may not read the correct conversion results.

10.2.9 Conversion Result Comparison Upper Limit Setting 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) (shown in Figure 10-7 Range of interrupt signal generation for the ADRCK bit). 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”.

Notes

1. Only the higher 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.

Figure 10-12 Format of conversion result comparison upper limit setting register (ADUL)

Reset value: FFH R/W

	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

10.2.10 Conversion Result Comparison Lower Limit Setting 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) (shown in Figure 10-7 Range of interrupt signal generation for the ADRCK bit).

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”.

Figure 10-13 Format of conversion result comparison lower limit setting register (ADLL)

Reset value: 00H R/W

	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

Cautions

1. Only the higher 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.

10.2.11 A/D Converter Sampling Time Control Register (ADNSMP)

This register controls the A/D sampling time.

The ADNSMP register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes “00H”.

Figure 10-14 Format of A/D converter sampling time control register (ADNSMP)

Reset value: 0dH R/W



Sample clock number setting:

ADNSMP[7:0]	Sample time	Remark
8'h05	5.5 ADCLK	
8'h06	6.5 ADCLK	
8'h07	7.5 ADCLK	
8'h08	8.5 ADCLK	
8'h09	9.5 ADCLK	
8'h0a	10.5 ADCLK	
8'h0b	11.5 ADCLK	
8'h0c	12.5 ADCLK	
8'h0d	13.5 ADCLK	Default value
8'h0e	14.5 ADCLK	
8'h0f	15.5 ADCLK	
8'h10	16.5 ADCLK	
8'h11	17.5 ADCLK	
8'h12	18.5 ADCLK	
8'h13	19.5 ADCLK	
8'h14	20.5 ADCLK	
.....		
8'hff	255.5 ADCLK	

Caution: To rewrite the ADNSMP register, it must be in the conversion stop status (ADCS=0).

Time required to perform an ADC conversion:

High-speed conversion mode: ADC conversion time = (number of sampling clocks + number of successive comparison clocks
(31.5))/F_{AD}

Low-current conversion mode: ADC conversion time = (number of sampling clocks + number of successive comparison clocks
(40.5))/F_{AD}

The number of AD sampling clocks can be adjusted by the ADNSMP register, and the default value is 13.5 ADCLK. The number of successive comparison clocks are determined by the conversion mode, which is 31.5 ADCLK for high speed conversion mode and 40.5 ADCLK for low current mode.

Under different conditions, the sampling time of each channel should be guaranteed:

Sampling time calculation equations: Number of sampling clocks /f_{AD} ≥ recommended sampling time

A/D conversion mode	AVDD[V]	ANIX[ns]	PGA/ temperature sensors/internal reference voltage [ns]
High speed conversion	4.5~5.5	211	633
	2.7~5.5	250	750
	2.4~5.5	422	1266
Low current conversion	2.7~5.5	500	759
	2.4~5.5	844	1281
	1.8~5.5	1688	2563

Caution: For actual use, please configure the conversion mode, the number of sampling clocks and the conversion clock frequency according to the AC Characteristics requirement in the data sheet.

10.2.12 A/D Sample Time Extension 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”.

Figure 10-15 Format of A/D sample time extension register (ADSMPWAIT)

Reset value: 00H R/W

	7	6	5	4	3	2	1	0
ADSMPWAIT	0	0	0	0	0	0	0	ADSMPWAIT

ADSMTWIT	A/D conversion target
0	When it is 0, the A/D sampling time is set directly by the ADNSMP register
1	When it is 1 and the sampling time of A/D is arbitrarily prolonged, the A/D conversion target is controlled continuously by ADNSMP after changing from 1 to 0.

Note: Set ADSMPWAIT=1 in the conversion stop state (ADCS=0), and rewrite ADSMPWAIT to 0 when (ADCS=1).

10.2.13 A/D Test Register (ADTES)

This register is used to set the test mode of the A/D converter.

The ADTES register is set by an 8-bit memory manipulation instruction. After the reset signal is generated, the value of this register changes to “00H”.

Figure 10-16 Format of A/D test register (ADTES)

Reset value: 00H R/W

	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	ADTES2	ADTES1	ADTES0

ADTES2	ADTES1	ADTES0	A/D operation mode
0	0	0	Normal conversion
0	0	1	Self-diagnostic testing for 0 code
0	1	1	Self-diagnostic testing of half code
1	0	1	Self-diagnostic testing of full codes
Other than above			Disable settings.

Caution Bits 7 to 3 must be set to 0.

10.2.14 A/D Status Register (ADFLG)

This register represents the state of the A/D converter.

The ADFLG register is read by an 8-bit memory manipulation instruction.

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

Figure 10-17 Format of A/D status register (ADFLG)

Reset value: 00H R

	7	6	5	4	3	2	1	0
ADFLG	0	0	0	ADFLG4	ADFLG3	ADFLG2	ADFLG1	ADFLG0

ADFLG4	A/D transition state
0	A/D conversion is not complete when single conversion mode
1	End of conversion in single conversion mode (auto cleared after 2 ADCLKs) ADFLG4 keeps 1'b0 during continuous conversion mode

ADFLG3	A/D transition state
0	1 ADCLK before non-A/D conversion ends
1	1 ADCLK before A/D conversion ends (auto cleared after 1 ADCLK)

ADFLG2	A/D transition state
0	2 ADCLK before non-A/D conversion ends
1	2 ADCLK before A/D conversion ends (auto cleared after 1 ADCLK)

ADFLG1	A/D transition state
0	Non-successive compare period
1	Successive compare period

ADFLG0	A/D transition state
0	Non-A/D sampling period
1	A/D sampling period

10.2.15 A/D Charge/Discharge Control Register (ADNDIS)

The register is used to control the charging and discharging operation and time of the A/D converter.

The ADNDIS register is read and written by an 8-bit memory manipulation instruction.

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

Figure 10-18 Format of A/D charge/discharge control register (ADNDIS)

Reset value: 00H W

	7	6	5	4	3	2	1	0
ADNDIS	0	0	0	ADNDIS4	ADNDIS3	ADNDIS2	ADNDIS1	ADNDIS0

ADNDIS [4]	Charge and discharge control
1'b0	Discharge
1'b1	Charging

ADNDIS [3:0]	Charge and discharge time
4'b0000	No charging or discharging
4'b0010	2x ADCLK
4'b0011	3x ADCLK
4'b0100	4x ADCLK
4'b0101	5x ADCLK
4'b0110	6x ADCLK
.....	
4'b1111	15x ADCLK

Note: Disable setting charge/discharge time for 1 ADCLK, i.e. ADNDIS[3:0]=4'b0001

10.2.16 Registers Controlling Port Functions of Analog Input Pins

When using the ANI0~ANI14 pins as the analog inputs to an A/D converter, the ports must be configured as analog channels by setting the corresponding Port Mode Control Register (PMCxx) bits to “1”. For details, please refer to “Chapter 2 Pin Functions”.

10.3 Input Voltage and Conversion Results

The analog input voltage at the analog input pin (ANI0~ANI14) and the theoretical A/D conversion result (12-bit A/D Conversion Result Register (ADCR)) are related by the following expressions.

$$ADCR = \text{INT} \left(\frac{V_{AIN}}{AV_{REF}} \times 4096 + 0.5 \right)$$

or

$$(ADCR - 0.5) \times \frac{AV_{REF}}{4096} \leq V_{AIN} < (ADCR + 0.5) \times \frac{AV_{REF}}{4096}$$

INT(): A function that returns the integer portion of a numeric value in parentheses

V_{AIN} : Analog input voltage

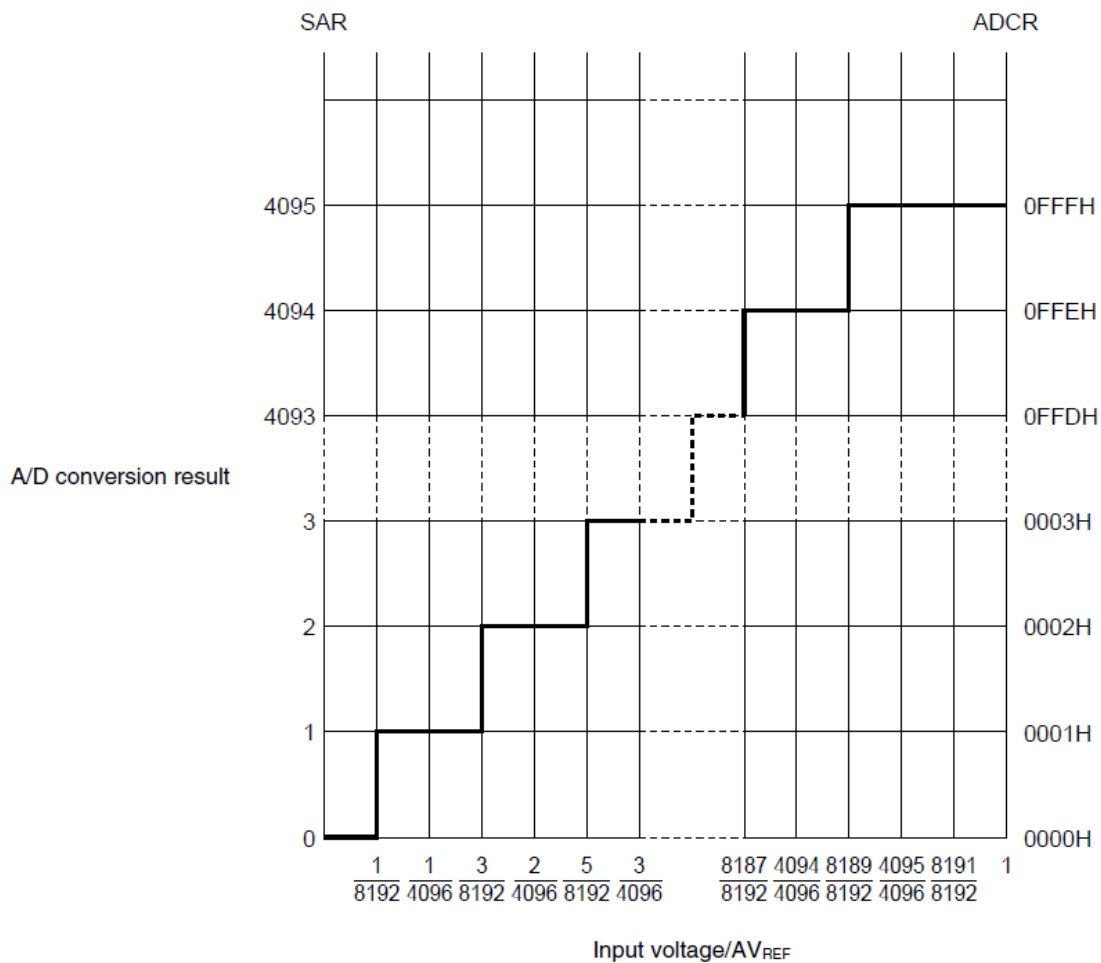
AV_{REF} : AV_{REF} pin voltage

ADCR: The value of the A/D conversion result register (ADCR)

SAR: Successive approximation register

The relationship between the analog input voltage and the A/D conversion results is shown in the following figure.

Figure 10-19 Analog input voltage vs. A/D conversion results



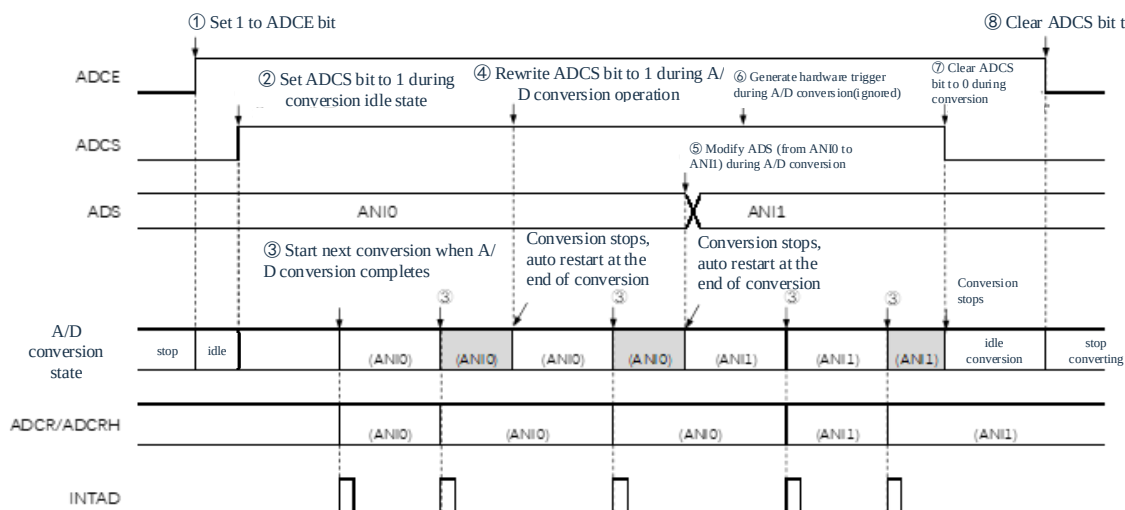
10.4 Operation Mode of A/D Converter

The A/D converter conversion operations are described below. For the setting of each mode, please refer to “10.5 A/D converter setup flowchart”.

10.4.1 Software Trigger Mode (Select mode, Continuous 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.

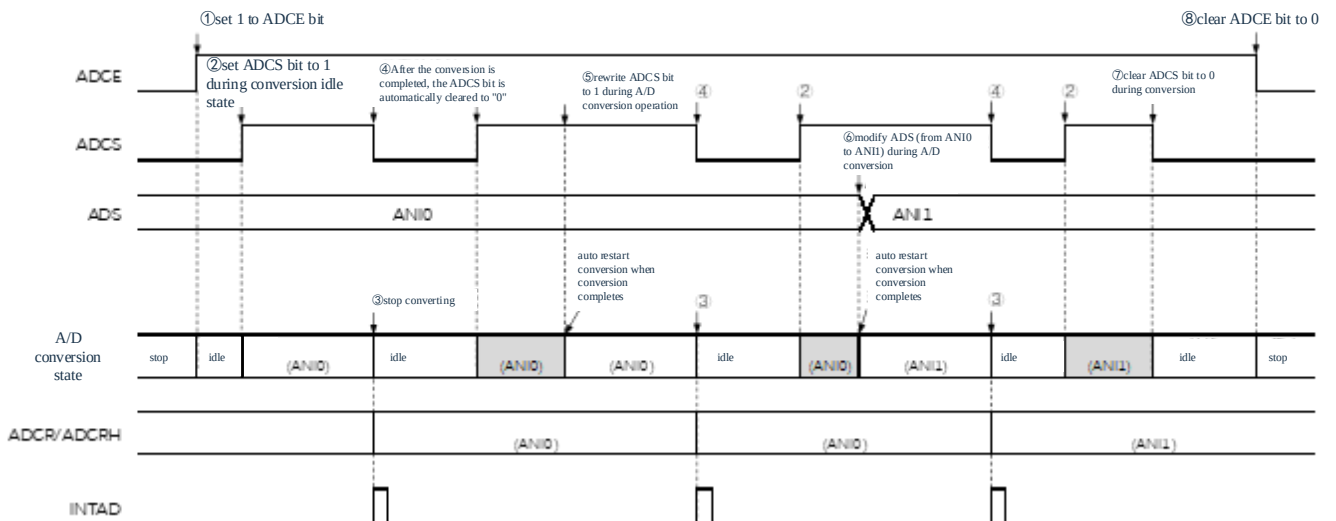
Figure 10-20 Example of software trigger mode (select mode, sequential conversion mode) operation timing



10.4.2 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.

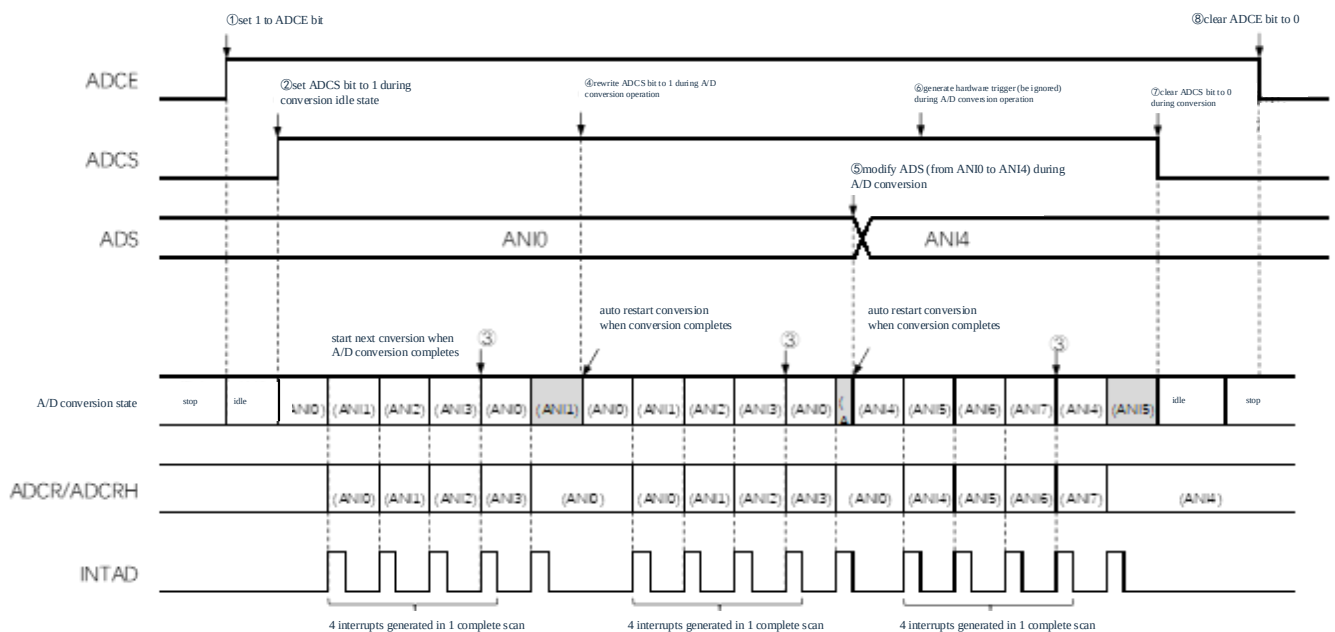
Figure 10-21 Example of software trigger mode (select mode, single conversion mode) operation timing



10.4.3 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.

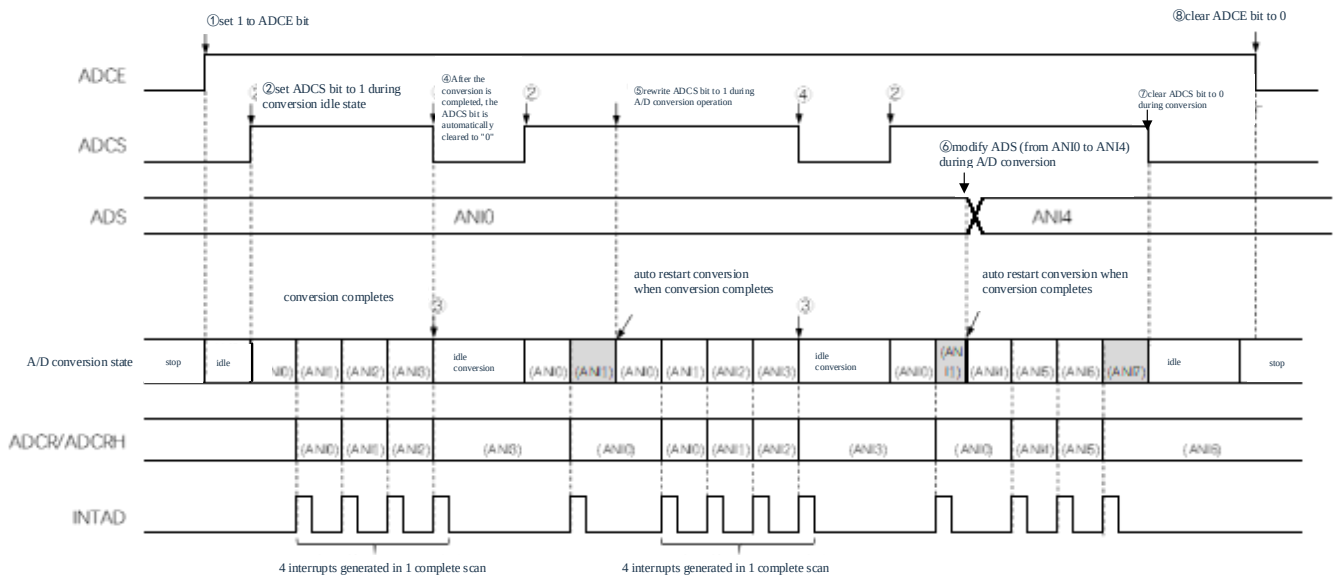
Figure 10-22 Example of software trigger mode (scan mode, sequential conversion mode) operation timing



10.4.4 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.

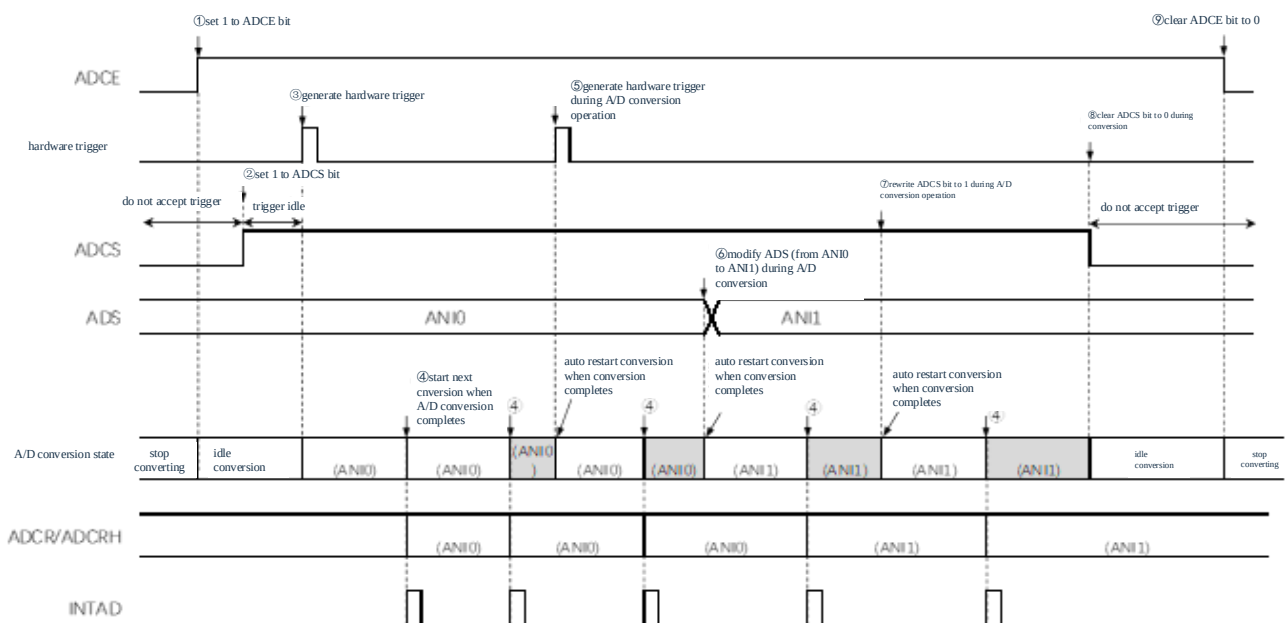
Figure 10-23 Example of software trigger mode (scan mode, single conversion mode) operation timing



10.4.5 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.

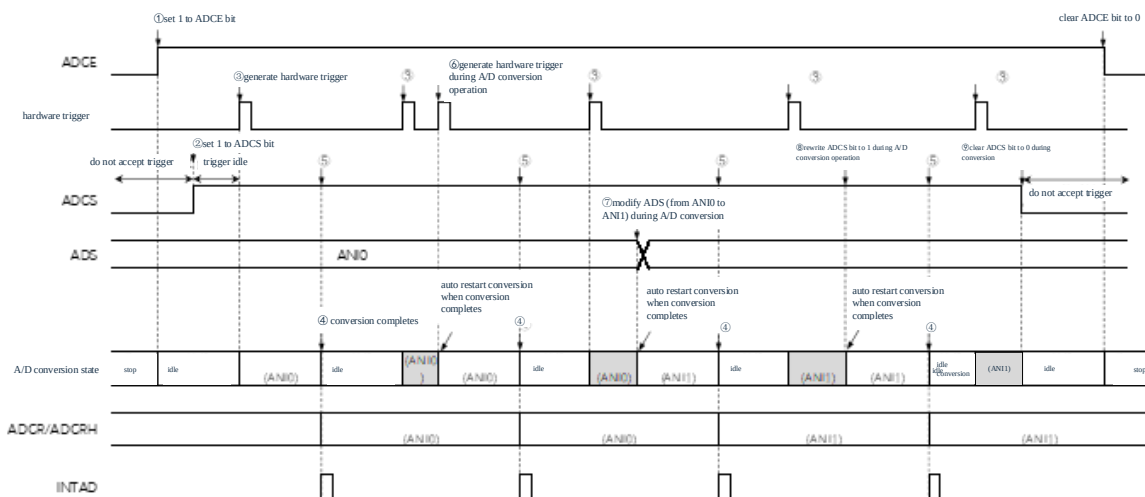
Figure 10-24 Example of hardware trigger no-wait mode (select mode, sequential conversion mode) operation timing



10.4.6 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.

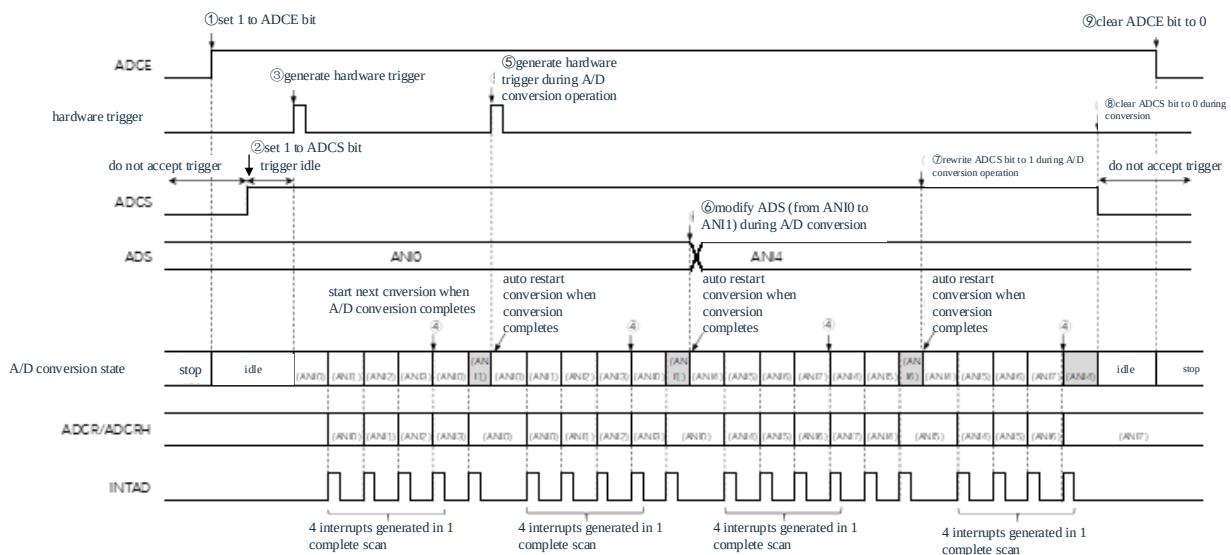
Figure 10-25 Example of hardware trigger no-wait mode (select mode, single conversion mode) operation timing



10.4.7 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.

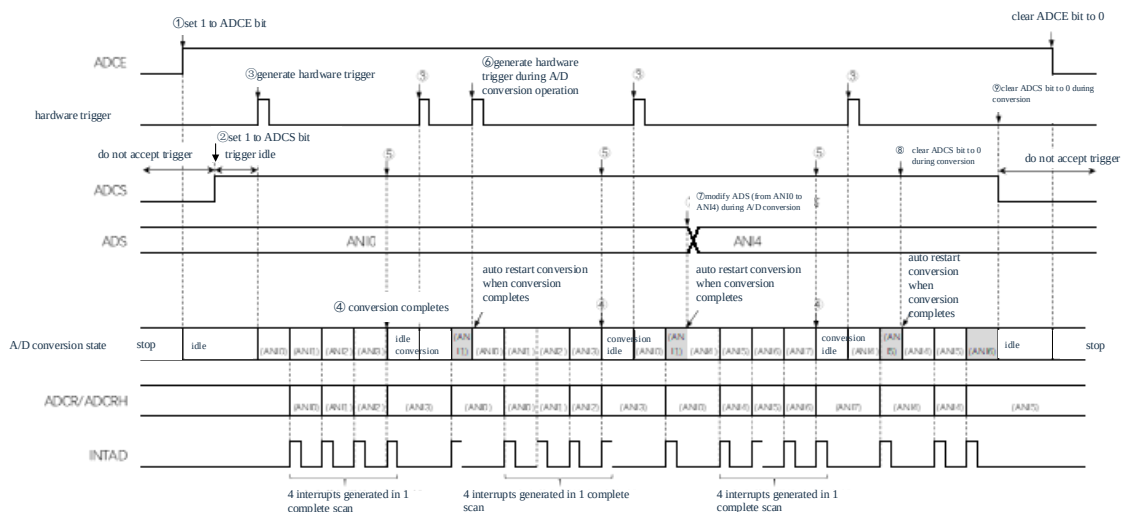
Figure 10-26 Example of hardware trigger no-wait mode (scan mode, sequential conversion mode) operation timing



10.4.8 Hardware Trigger No-Wait 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 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.
- ⑤ After A/D conversion of the four channels 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 at the first channel.
- 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 first channel 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 at the first channel.
- 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.

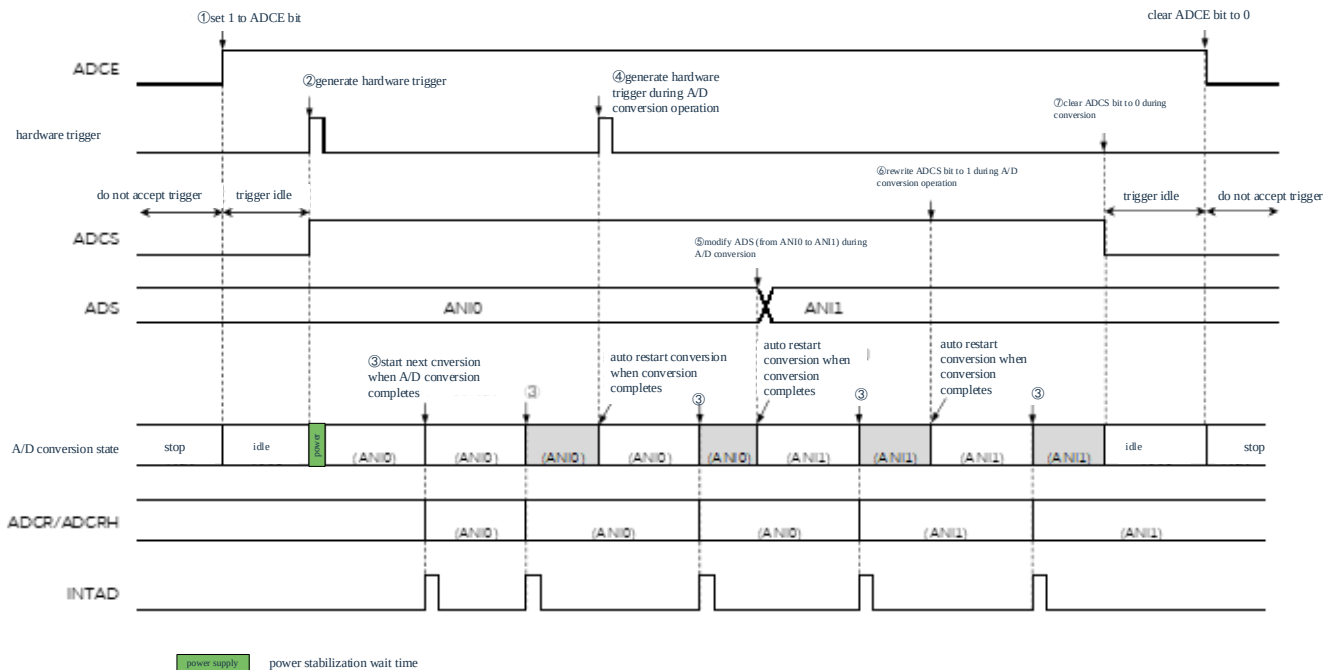
Figure 10-27 Hardware trigger no-wait mode (scan mode, single conversion mode)



10.4.9 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.

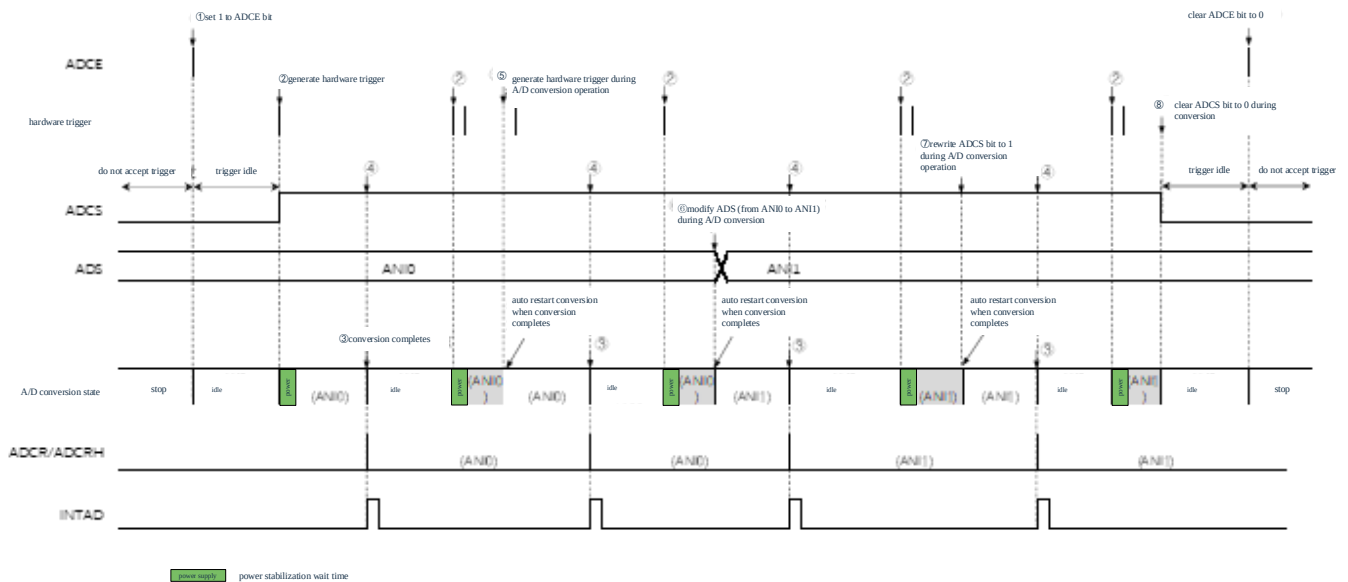
Figure 10-28 Example of hardware trigger wait mode (select mode, sequential conversion mode) operation timing



10.4.10 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.

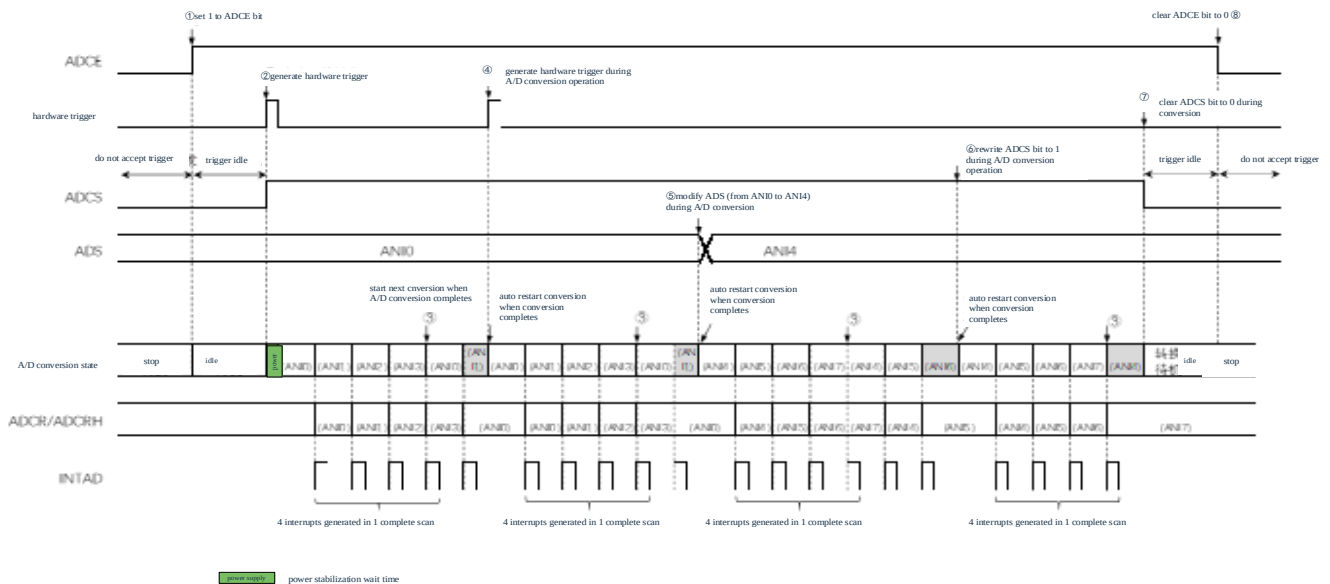
Figure 10-29 Example of hardware trigger wait mode (select mode, single conversion mode) operation timing



10.4.11 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.

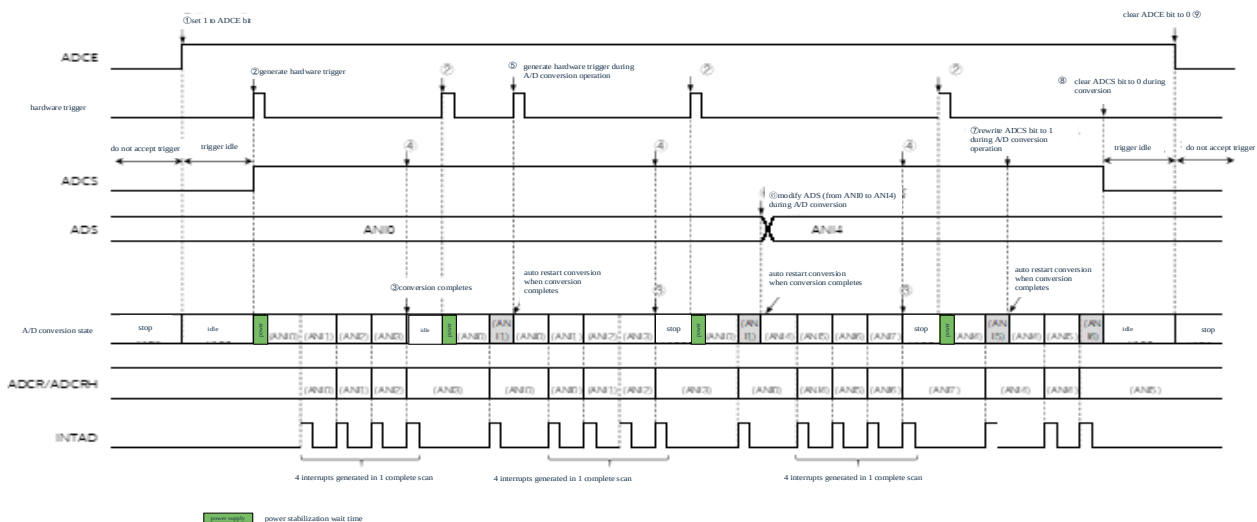
Figure 10-30 Example of hardware trigger wait mode (scan mode, sequential conversion mode) operation timing



10.4.12 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 10-31 Example of hardware trigger wait mode (scan mode, single conversion mode) operation timing

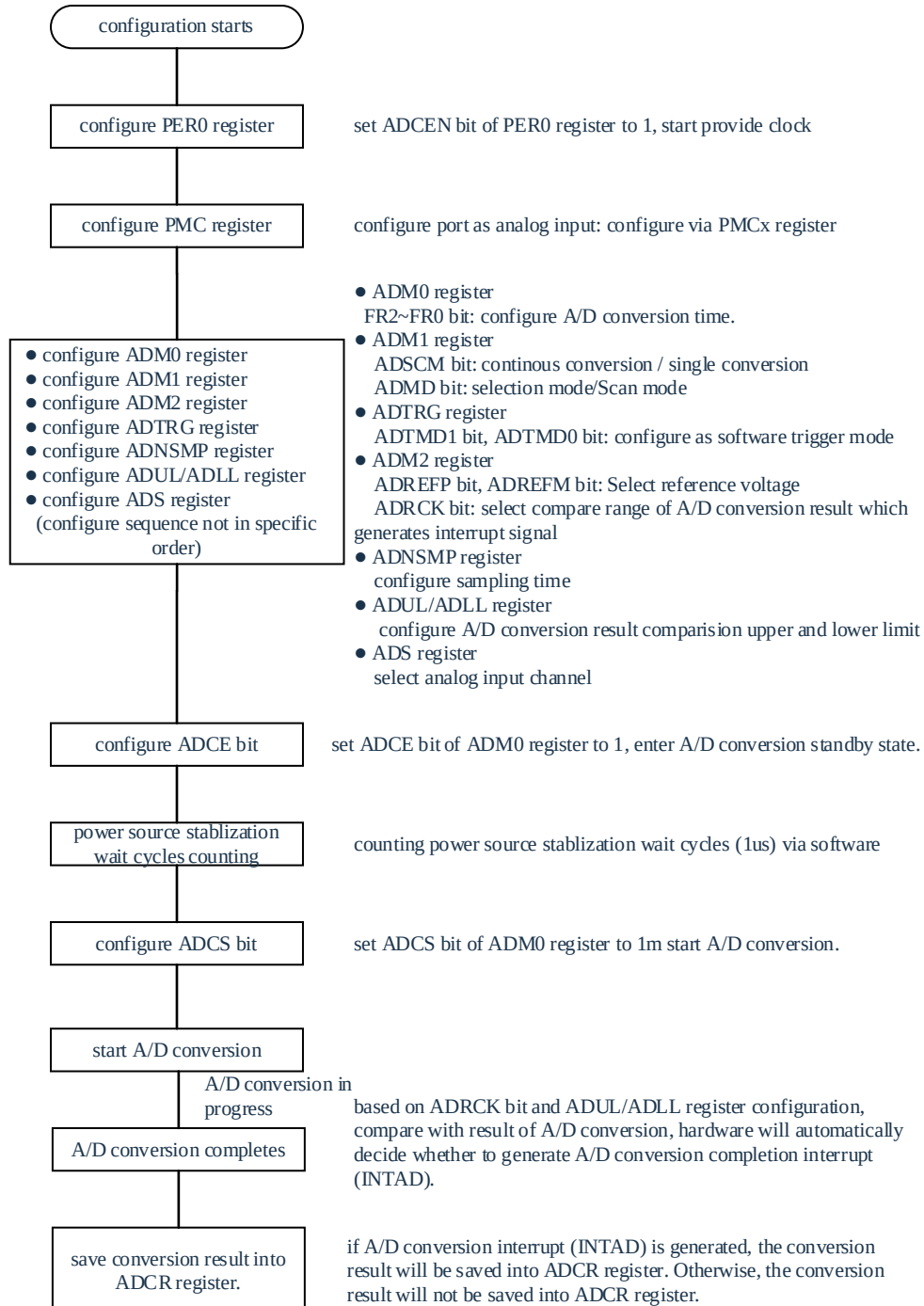


10.5 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

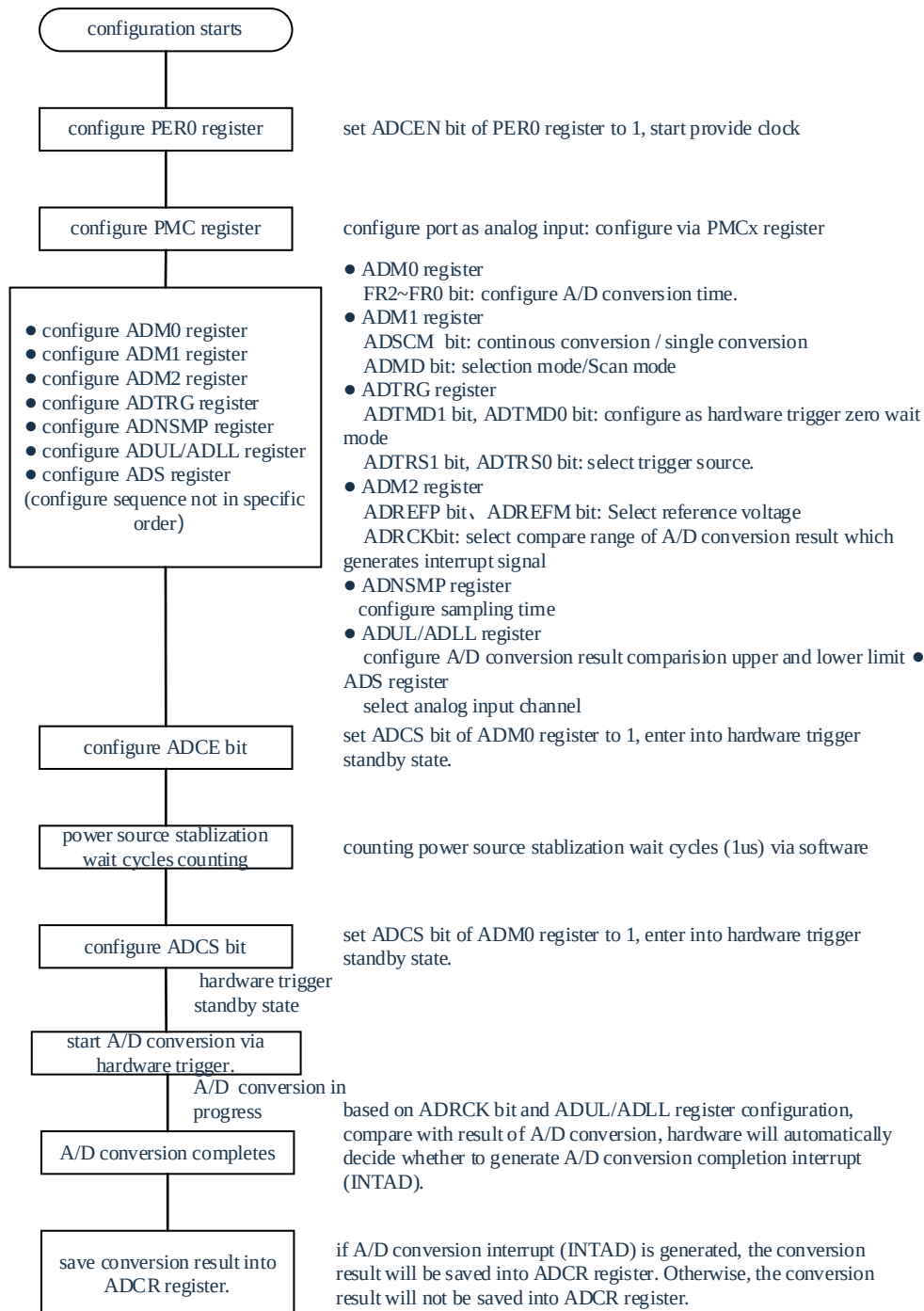
10.5.1 Software Trigger Mode Settings

Figure 10-32 Software trigger mode settings



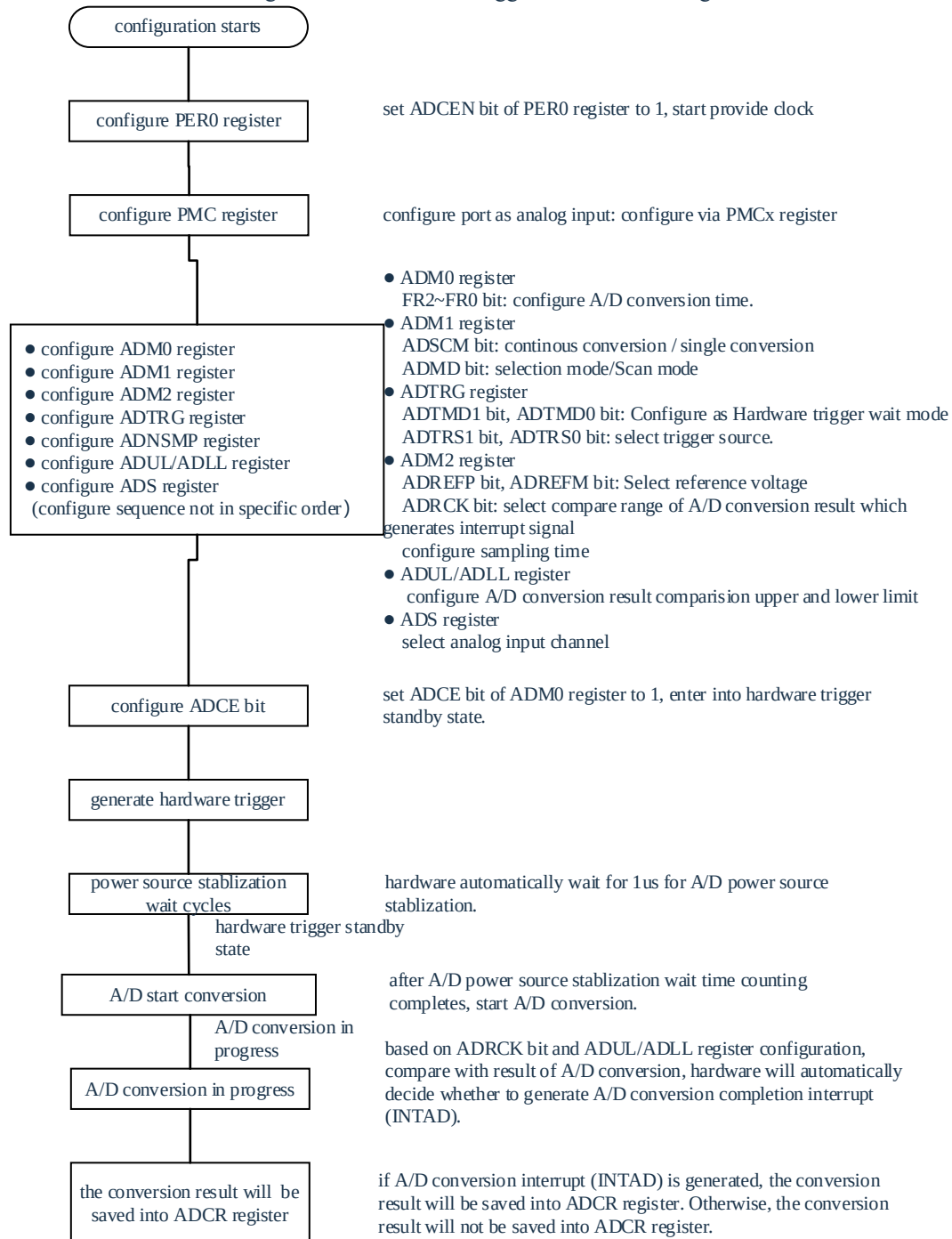
10.5.2 Hardware Trigger No-Wait Mode Settings

Figure 10-33 Hardware trigger no-wait mode settings



10.5.3 Hardware Trigger Wait Mode Settings

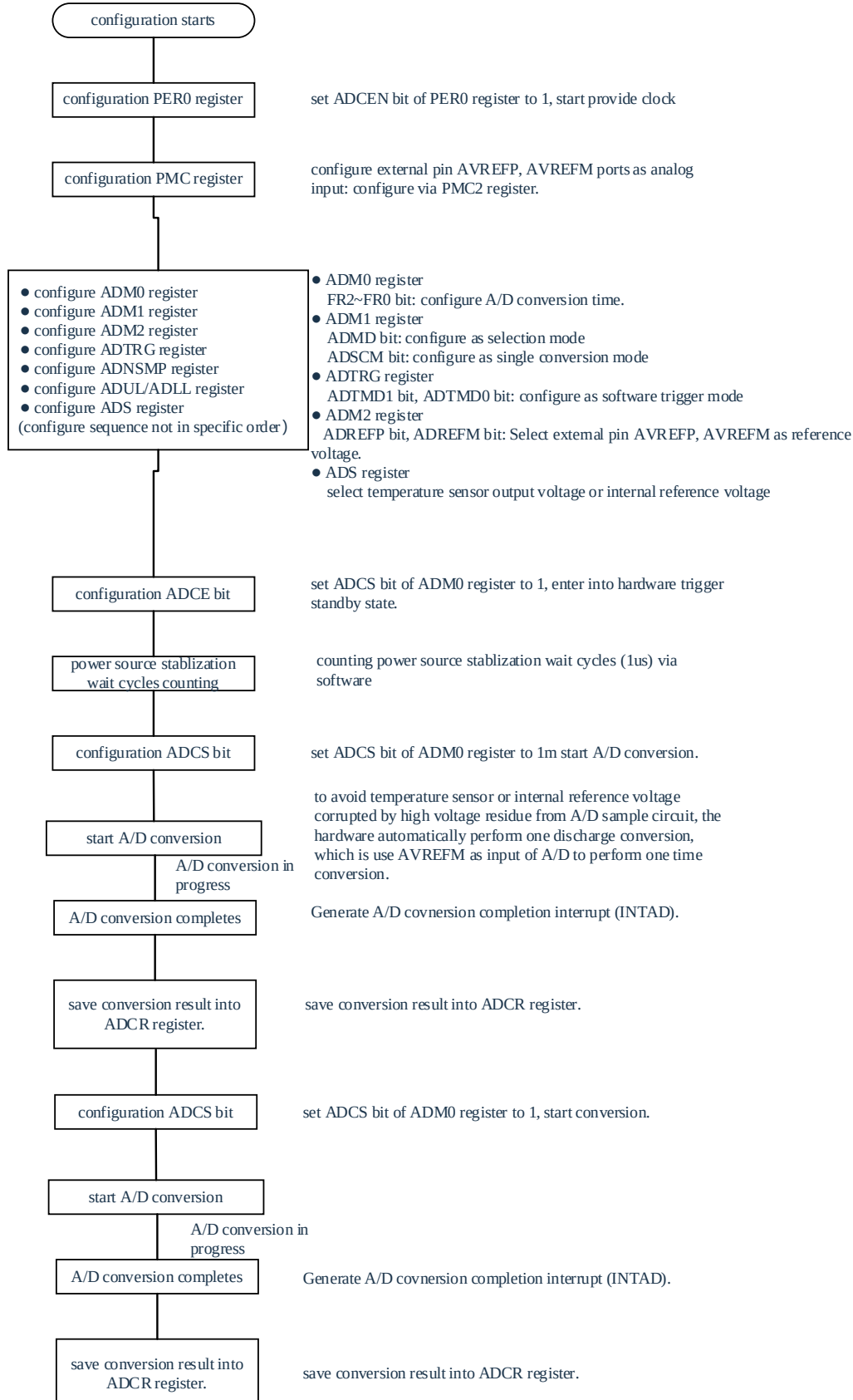
Figure 10-34 Hardware trigger wait mode settings



10.5.4 Settings When Selecting the Output Voltage/Internal Reference Voltage of the Temperature Sensor

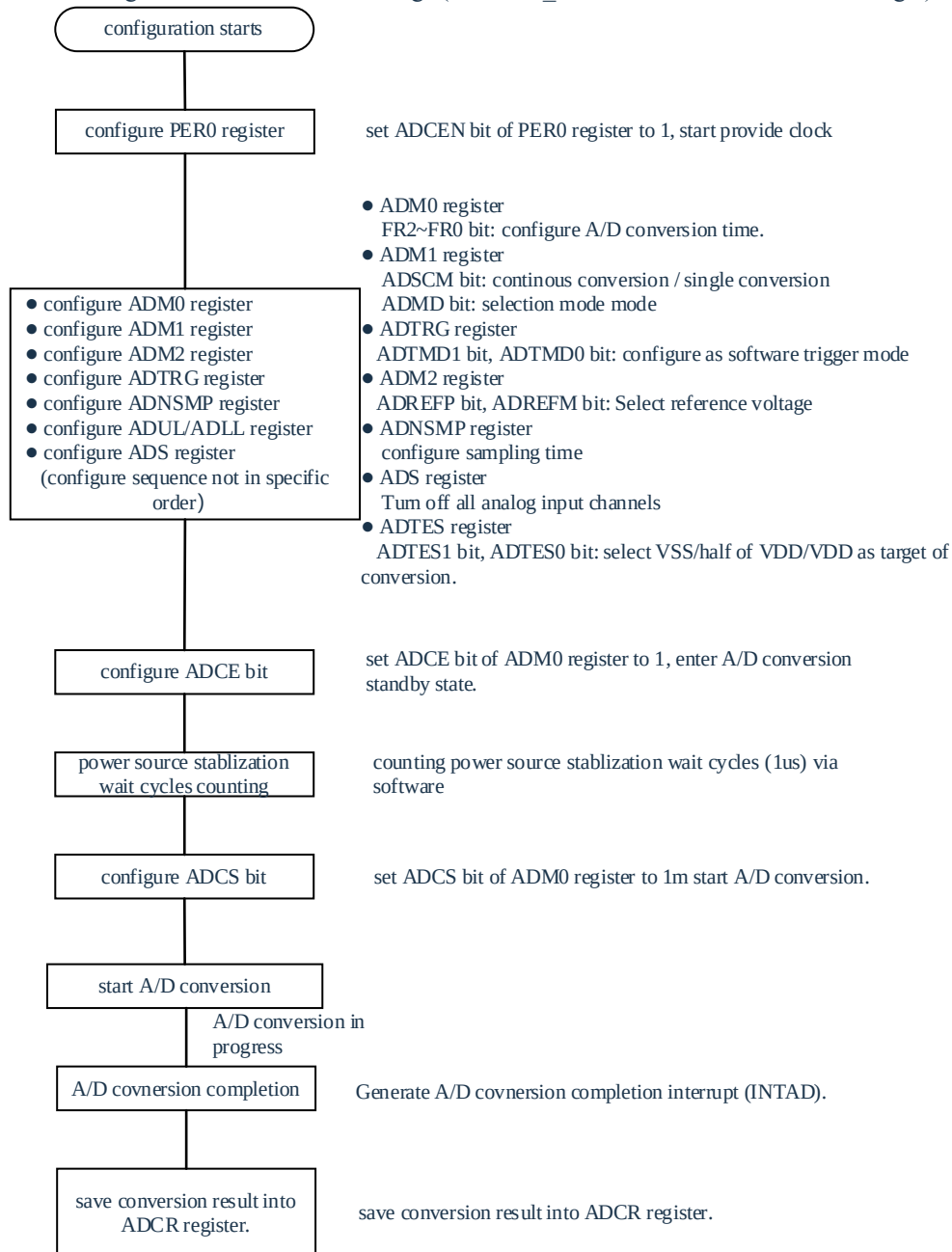
(Take software trigger mode, single conversion mode for example)

Figure 10-35 Settings when selecting the output voltage/internal reference voltage of the temperature sensor



10.5.5 Test Mode Settings

Figure 10-36 Test mode settings (VSS/half_VDD/VDD as the conversion target)



Chapter 11 Comparator

This product has a built-in 2-channel comparator.

11.1 Functions of Comparator

The comparator has the following functions:

- The input pin of the CMP1 can select an external port, internal reference voltage, and internal DAC reference voltage.
- The comparison results of comparator 0 and comparator 1 can be output by pins (VCOUT0, VCOUT1).

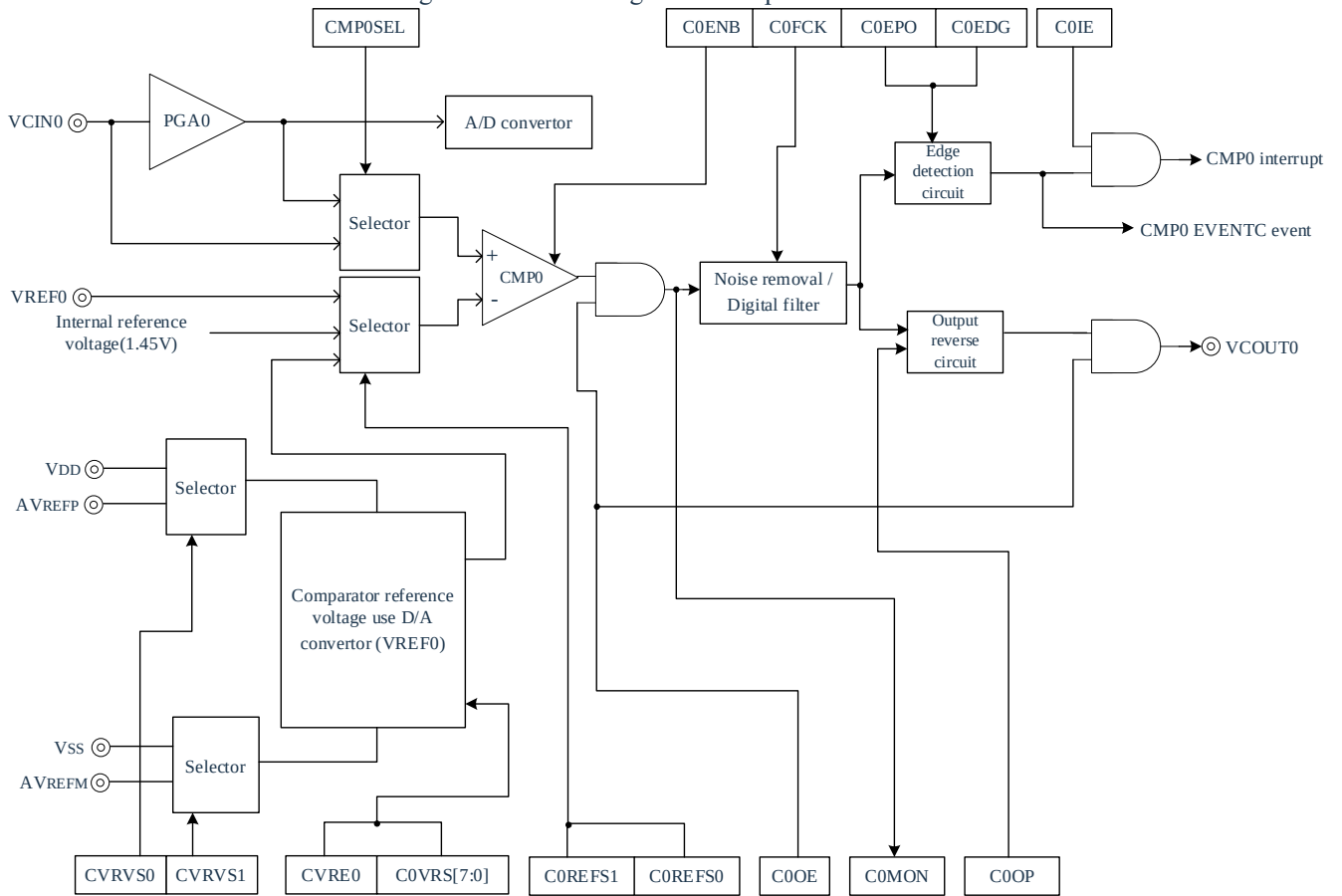
Table 11-1 Comparator feature summary

Item	Content
CMP	• 2-channel comparators (CMP0 and CMP1) • The comparator negative can select a reference voltage: CMP0 negative can select from external pin inputs, built-in reference voltage of CMP0 and internal reference voltage (1.45V) CMP1 negative can select from external pin inputs (4), built-in reference voltage of CMP1 and internal reference voltage (1.45V) • The negative internal reference voltage can be set (256 steps) • The CMP0 positive can select the PGA output • The CMP1 positive can select the external PGA pin inputs (4) • When the positive input voltage > the negative input voltage, the output level is high • When the positive input voltage < the negative input voltage, the output level is low • The digital filter width can be selected • Output reverse function • The comparison result can be output from the pins (VCOUT0, VCOUT1) • An effective edge of the comparator output can be detected and an interrupt signal is generated • Combined with Timer4 to output TIMER WINDOW

11.2 Structure of Comparator

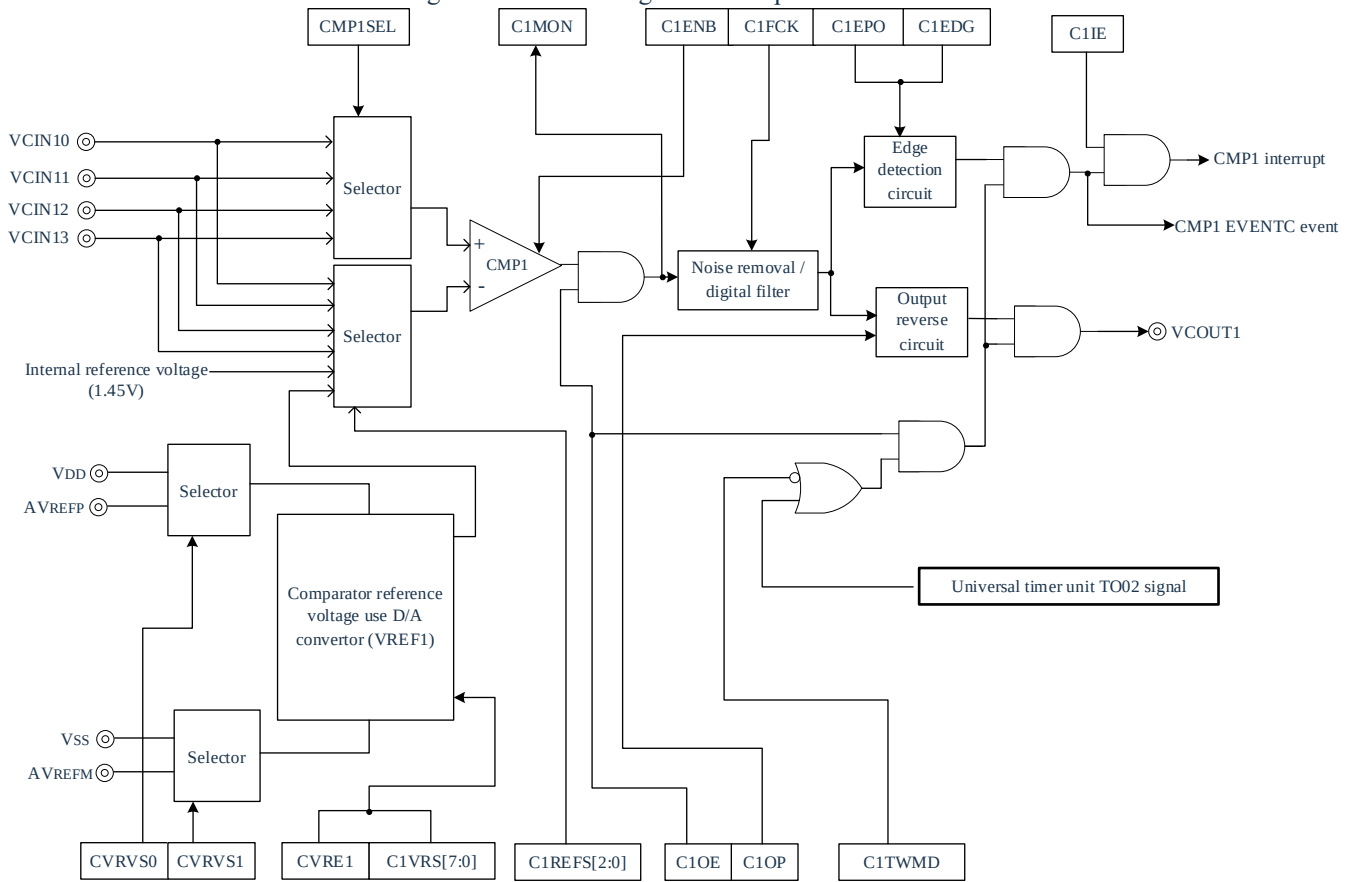
The block diagram of Comparator 0 is shown in Figure 11-1.

Figure 11-1 Block diagram of comparator 0



The block diagram of Comparator 1 is shown in Figure 11-2.

Figure 11-2 Block diagram of comparator 1



11.3 Registers for Controlling Comparator

The registers that control the comparator are shown in Table 11-2.

Table 11-2 Registers for controlling comparator

Register name	Symbol
Peripheral enable register 1	PER1
Comparator mode configuration register	COMPMDR
Comparator filter control register	COMPFIR
Comparator output control register	COMPOCR
Comparator built-in reference voltage control register	CVRCTL
Comparator built-in reference voltage selection register 0	C0RVM
Comparator built-in reference voltage selection register 1	C1RVM
Comparator 0 input selection control register	CMPSEL0
Comparator 1 input selection control register	CMPSEL1
Port mode control register	PMCxx
Port mode register	PMxx
Port alternate configuration register	PxxCFG

11.3.1 Peripheral Enable Register 1 (PER1)

The PER1 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.

Set bit5 (PGACMPEN) to 1 when using the comparators.

The PER1 register is set by an 8-bit memory manipulation instruction.

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

Figure 11-3 Format of peripheral enable register 1(PER1)

Address: 0x4002081A

After reset: 00H

R/W

Symbol

PER1

7	6	5	4	3	2	1	0
0	0	PGACMPEN	0	DMAEN	0	0	0

PGACMPEN	Control of comparator input clock
0	Stop input clock supply • The SFR used by the comparator cannot be written. • The comparator is in the reset state.
1	Enable input clock supply • The SFR used by the comparator can be read and written.

Caution: To set the comparator, the PGACMPEN bit must first be set to 1.

If PGACMPEN=0, writing to a control register of the comparator is ignored, and all read values are initial values (except for Port Registers (PMxx, PMxx and PxxCFG)).

11.3.2 Comparator Mode Configuration Register (COMPMDR)

The COMPMDR register is a register that sets the comparator action enable/disable and detects the comparator output.

The CiENB bit is forbidden to be set to 0 when the comparator output is allowed (CiOE bit of COMPOCR register is set to 1).

Setting the CiENB bit to 1 (i=0, 1) is prohibited in the following cases:

- The CMP negative input selects the built-in reference voltage while the built-in reference voltage action stops (the CVREi bit of the CVRCTL register is 0)
- The input of the CMP0 selects the output of the PGA, and when the PGA action stops (the CMPSEL0 bit of the CVRCTL register is 1 and the PGAEN bit of the PGAEN register is 0)

The COMPMDR register is set by an 8-bit memory manipulation instruction. After the reset signal is generated, the value of this register changes to “00H”.

Figure 11-4 Format of comparator mode configuration register (COMPMDR)

Address: 40043840H

After reset: 00H

R/W

Symbol

COMPMDR

7	6	5	4	3	2	1	0
C1MON	0	0	C1ENB	C0MON	0	0	C0ENB

C1MON	Comparator 1 monitor flag ^{Notes 1, 2}
0	IVCMP1<Reference voltage of comparator 1, or comparator 1 is not running.
1	IVCMP1> Reference voltage of comparator 1

C1ENB	Enable comparator 1 operation
0	Disable comparator 1 operation.
1	Enable comparator 1 operation.

C0MON	Comparator 0 monitor flag ^{Note 1,2}
0	IVCMP0<Reference voltage of comparator 0, or comparator 0 is not running.
1	IVCMP0>Reference voltage of comparator 0

C0ENB	Enable comparator 0 operation
0	Disable comparator 0 operation
1	Enable comparator 0 operation

Notes

1. It becomes 0 (initial value) immediately after the reset is released, or an undefined value if both the C0ENB bit and the C1ENB bit are set to 0 after the operation of the comparator is allowed.
2. Ignore the write value for this bit.

11.3.3 Comparator Filter Control Register (COMPFIR)

The COMPFIR register is a control register for the digital filter. The COMPFIR register is set by an 8-bit memory manipulation instruction.

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

Figure 11-5 Format of comparator filter control register (COMPFIR)

Address: 40043841H

After reset: 00H

R/W

Symbol

7

6

5

4

3

2

1

0

COMPFIR

C1EDG	C1EPO	C1FCK1	C1FCK0	C0EDG	C0EPO	C0FCK1	C0FCK0

C1EDG	Comparator 1 edge detection selection ^{Note 1}
0	An interrupt request is generated by single edge detection of the comparator 1.
1	An interrupt request is generated by bilateral edge detection of the comparator 1.

C1EPO	Comparator 1 edge polarity switching ^{Note 1}
0	An interrupt request is generated by the rising edge of the comparator 1.
1	An interrupt request is generated by the falling edge of the comparator 1.

C1FCK1	C1FCK0	Comparator 1 filter selection ^{Note 1}
0	0	Comparator 1 has no filter.
0	1	Comparator 1 has a filter and samples it by f_{CLK} .
1	0	Comparator 1 has a filter and samples it by $f_{CLK}/8$.
1	1	Comparator 1 has a filter and samples it by $f_{CLK}/32$.

C0EDG	Comparator 0 edge detection selection ^{Note 2}
0	An interrupt request is generated by single edge detection of comparator 0.
1	An interrupt request is generated by bilateral edge detection of comparator 0.

C0EPO	Comparator 0 edge polarity switching ^{Note 2}
0	An interrupt request is generated by the rising edge of the comparator 0.
1	An interrupt request is generated by the falling edge of the comparator 0.

C0FCK1	C0FCK0	Comparator 0 filter selection ^{Note 2}
0	0	Comparator 0 has no filter.
0	1	Comparator 0 has a filter, and samples it by f_{CLK} .
1	0	Comparator 0 has a filter, and samples it by $f_{CLK}/8$.
1	1	Comparator 0 has a filter, and samples it by $f_{CLK}/32$.

Notes:

- If you modify the C1FCK1~C1FCK0 bits, the C1EPO bit, and the C1EDG bit, it may trigger an interrupt request for comparator 1 and an event signal to be output to EVENTC. These bits must be modified after setting the ELSELR14 register of EVENTC (which does not link the output of comparator 1) to 0. Additionally, the interrupt request flag register (IF) must be cleared to 0.

If the C1FCK1~C1FCK0 bits are changed from “00B” (comparator 1 without a filter) to any other value (comparator 1 with a filter), it is required to sample the filter’s output four times before using the interrupt request of comparator 1 or the event

signal to EVENTC.

2. If you modify the C0FCK1~C0FCK0 bits, the C0EPO bit, and the C0EDG bit, it may trigger an interrupt request for comparator 0 and an event signal to be output to EVENTC. These bits must be modified after setting the ELSELR13 register of EVENTC (which does not link the output of comparator 0) to 0. Additionally, the interrupt request flag register (IF) must be cleared to 0.

If the C0FCK1~C0FCK0 bits are changed from “00B” (comparator 0 without a filter) to any other value (comparator 0 with a filter), it is required to sample the filter’s output four times before using the interrupt request of comparator 0 or the event signal to EVENTC.

11.3.4 Comparator Output Control Register (COMPOCR)

The COMPOCR register is a control register used to set the polarity of the comparator output, enable/disable the output, and enable/disable the interrupt output.

The CiOE bit (output enable) in the COMPOCR register should not be set to 1 (output enabled) in the following cases (i = 0, 1):

When the comparator operation is stopped (CiENB bit of the COMPMDR register is 0), the CMP negative input is selected as the built-in reference voltage.

When the built-in reference voltage operation is stopped (CVREi bit of the CVRCTL register is 0), the CMP0 input is selected as the output of the PGA.

When the PGA operation is stopped (CMPSEL0 bit of the CVRCTL register is 1 and PGAEN bit of the PGAEN register is 0), the COMPOCR register is set by an 8-bit memory manipulation instruction.

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

Figure 11-6 Format of comparator output control register (COMPOCR)

Address: 40043842H

After reset: 00H

R/W

Symbol

7

6

5

4

3

2

1

0

COMPOCR

C1OTWMD	C1OP	C1OE	C1IE	0	C0OP	C0OE	C0IE

C1OTWMD	Comparator 1 TIMER WINDOW output mode control bit ^{Note 1}
0	Comparator 1 normal output mode (controlled by C1OE)
1	Comparator 1 TIMER WINDOW output mode (co-controlled by TO02 and C1OE)

C1OP	Selection of VCOUT1 output polarity
0	Comparator 1 output from VCOUT1
1	Inverted output of comparator 1 from VCOUT1

C1OE	Enable VCOUT1 pin output ^{Note 2,3}
0	Disable comparator1 VCOUT1 pin output
1	Enable comparator1 VCOUT1 pin output

C1IE	Enable Comparator 1 interrupt request ^{Note 4}
0	Disable the comparator 1 interrupt request
1	Enable the comparator 1 interrupt request

C0OP	Selection of VCOUT0 output polarity
0	Comparator 0 output from VCOUT0
1	Inverted output of comparator 0 from VCOUT0

C0OE	Enable VCOUT0 pin output ^{Notes 5,6}
0	Disable comparator0 VCOUT0 pin output
1	Enable comparator0 VCOUT0 pin output ^{Notes 4, 8}

C0IE	Enable Comparator 0 interrupt request ^{Note 7}
0	Disable comparator 0 interrupt request
1	Enable comparator 0 interrupt request

Note 1 When Comparator 1 is used in TIMER WINDOW mode, bit 7 (C1EDG) of the COMPFIR register must be set to 1. The

C1OE and C1OTWMD bits cannot be set simultaneously. You must first set the C1OTWMD bit and then set the C1OE bit to 1.

- Note 2 Modifying the C1OE bit may trigger an interrupt request and an EVENTC event for Comparator 1. Before modifying this bit, make sure to set the ELSELR14 register of EVENTC to 0 (disconnect it from the output of Comparator 1). After modifying the C1OE bit, initialize the IF bit in the interrupt request flag register (no interrupt request).
- Note 3 The result of Comparator 1 can be output to any pin except P00. To select a specific pin, set the PxxCFG register of that pin to 0x1b, set PMxx to 1'b0, and PMCxx to 1'b0.
- Note 4 Modifying the C0OE bit may trigger an interrupt request and an EVENTC event for Comparator 0. Before modifying this bit, make sure to set the ELSELR13 register of EVENTC to 0 (disconnect it from the output of Comparator 0). After modifying the C0OE bit, initialize the IF bit in the interrupt request flag register (no interrupt request).
- Note 5 If you change C1IE from 0 (disable interrupt request) to 1 (enable interrupt request), the IF bit in the interrupt request flag register may change to 1 (indicating an interrupt request). Therefore, ensure that you clear the IF bit to 0 in the interrupt request flag register before using the interrupt.
- Note 6 The result of Comparator 0 can be output to any pin except P00. To select a specific pin, set the PxxCFG register of that pin to 0x1b, set PMxx to 1'b0, and PMCxx to 1'b0. For detailed information, please refer to Chapter 2: Pin Functions, Section 2.3.7.
- Note 7 If you change C0IE from 0 (disable interrupt request) to 1 (enable interrupt request), the IF bit in the interrupt request flag register may change to 1 (indicating an interrupt request). Therefore, ensure that you clear the IF bit to 0 in the interrupt request flag register before using the interrupt.

11.3.5 Comparator Built-In Reference Voltage Control Register (CVRCTL)

The CVRCTL register is used to enable/stop the operation of the built-in reference voltage for the comparator.

The CVRCTL register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to “00H”.

Note: When the built-in reference voltage operation is stopped ($CVREi = 0$), the CVRVSi bit of the CVRCTL register should be rewritten.

Figure 11-7 Format of comparator built-in reference voltage control register (CVRCTL)

Address: 40043843H	After reset: 00H	R/W						
Symbol	7	6	5	4	3	2	1	0
CVRCTL	0	0	CVRE1	CVRVS1	0	0	CVRE0	CVRVS0

CVRE1	Control bit with built-in reference voltage 1
0	Disable operation of built-in reference voltage 1
1	Enable operation of built-in reference voltage 1

CVRVS1	Ground side selection bit with built-in reference voltage
0	Internal reference voltage ground side selects Vss
1	Internal reference voltage ground side selects AV_{REFM} ^{Note 1}

CVRE0	Built-in reference voltage 0 control bit
0	Disable operation of built-in reference voltage 0
1	Enable operation of built-in reference voltage 0

CVRVS0	Built-in reference voltage power supply side selection bit
0	Built in reference voltage power supply side selects V_{DD} with
1	Built in reference voltage power supply side selects AV_{REFP} ^{Note 2}

Note 1 The P25 pin is shared between AV_{REFM} and VCIN13. Therefore, when the P25 pin is used as the input signal for CMP1, setting the CVRVs1 bit to 1 is prohibited.

Note 2 The P26 pin is shared between AV_{REFP} and VCIN12. Therefore, when the P26 pin is used as the input signal for CMP1, setting the CVRVs0 bit to 1 is prohibited.

11.3.6 Comparator Built-In Reference Voltage Selection Register (CiRVM)

The CiRVM register is a register that sets the built-in reference voltage of the comparator.

When the built-in reference voltage stops (CVREi=0), rewrite the CiRVM register.

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

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

Figure 11-8 Format of comparator built-in reference voltage selection register i (CiRVM)

Address: 400438434H(C0RVM), 400438435H(C1RVM), After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CiRVM	CiRVS7	CiRVS6	CiRVS5	CiRVS4	CiRVS3	CiRVS2	CiRVS1	CiRVS0

CiRV S7	CiRV S6	CiRV S5	CiRV S4	CiRV S3	CiRV S2	CiRV S1	CiRV S0	Setting of built-in reference voltage of comparator
0	0	0	0	0	0	0	0	{{(AVREFP or VDD)/256}x0
0	0	0	0	0	0	0	1	{{(AVREFP or VDD)/256}x1
0	0	0	0	0	0	1	0	{{(AVREFP or VDD)/256}x2
0	0	0	0	0	0	1	1	{{(AVREFP or VDD)/256}x3
...								...
1	1	1	1	1	1	0	0	{{(AVREFP or VDD)/256}x252
1	1	1	1	1	1	0	1	{{(AVREFP or VDD)/256}x253
1	1	1	1	1	1	1	0	{{(AVREFP or VDD)/256}x254
1	1	1	1	1	1	1	1	{{(AVREFP or VDD)/256}x255

11.3.7 Comparator 0 Input Signal Selection Control Register (CMPSEL0)

The CMPSEL0 register is a selection register for the input signal of the positive end and the negative end of the comparator 0.

When comparator 0 stops (C0ENB=0), override the CMPSEL0 register.

The CMPSEL0 register is set by an 8-bit memory manipulation instruction. After the reset signal is generated, the value of this register changes to “00H”.

Figure 11-9 Format of comparator 0 input signal selection control register (CMPSEL0)

Address: 4004384AH	After reset: 00H	R/W							
Symbol	7	6	5	4	3	2	1	0	
CMPSEL0	CMP0SEL	0	0	0	0	0	C0REFS1	C0REFS0	

CMP0SEL	Comparator 0 positive input signal selection bit
0	Select external pin (VCIN0 pin)
1	Select PGA output signal

C0REFS1	C0REFS0	Comparator 0 negative input signal selection bit
0	0	Select the built-in reference voltage VREF0
0	1	Select the internal reference voltage (1.45V)
1	0	Select an external pin (IVREF0 pin)
1	1	Disable settings

11.3.8 Comparator 1 Input Signal Selection Control Register (CMPSEL1)

The CMPSEL1 register is a selection register for the input signal of the positive and the negative end of the comparator 1.

When comparator 1 stops (C1ENB=0), rewrite the CMPSEL1 register.

The CMPSEL1 register is set by an 8-bit memory manipulation instruction.

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

Figure 11-10 Format of comparator 1 input signal selection control register (CMPSEL1)

Address: 4004384BH

After reset: 00H

R/W

Symbol

7

6

5

4

3

2

1

0

CMPSEL1

CMPSEL1	CMPSEL0	0	0	0	COREFS1	COREFS1	COREFS0

CMPSEL1	CMPSEL0	Comparator 1 positive input signal selection bit
0	0	Select external pin (VCIN10 Pin)
0	1	Select external pin (VCIN11 Pin)
1	0	Select external pin (VCIN12 Pin)
1	1	Select external pin (VCIN13 Pin)

COREFS2	COREFS1	COREFS0	Negative input signal selection bit of comparator 1
0	0	0	Select the built-in reference voltage VREF1
0	0	1	Select internal reference voltage (1.45V)
0	1	0	Select external pin (VCIN10 Pin)
0	1	1	Select external pin (VCIN11 Pin)
1	0	0	Select external pin (VCIN12 Pin)
1	0	1	Select external pin (VCIN13 Pin)
1	1	0	Disable settings
1	1	1	Disable settings

Caution: When switching the analog input of CMP1, to prevent shoot-through current between the two input signals, the switching interval must be greater than 3μs.

11.3.9 Registers Controlling Port Functions of Analog Input Pins

When using the VCIN0 pin, VCIN10-VCIN13 pins, and the VREF0 pin as the analog inputs for the comparator, the corresponding bits in the Port Mode Register (PMxx) and the Port Mode Control Register (PMCxx) for each port must be set to 1.

When using the VCOUT0 and VCOUT1 functions, the Port Alternate Function Configuration Register (PxxCFG), Port Mode Register (PMxx), and Port Mode Control Register (PMCxx) must be set accordingly. For detailed information, refer to Chapter 2: Pin Functions, Section 2.3.10.

11.4 Operation Instructions

Comparator 0 and Comparator 1 can be run independently. The setting method is the same as operation.

CMP0 and PGA can be combined and linked.

Table 11-3 shows the setting procedure for independent operation and linkage of the comparator.

Table 11-3 Set-up steps for comparator-related registers

Step	Register	Bit	Set value
1	PGACTL	PGAVG0/1/2	Select gain ^{Note 3}
2	PGACTL	PVRVS	0 (Vss pin selection) ^{Note 3}
3	PGACTL	PGAEN	1 (Enable operation) ^{Note 3}
4	Wait for PGA stability time (minimum 10μs)		
5	COMPSELi	CMP0SEL/CMP1SELi	Comparator i positive input selection
6	COMPSELi	CiREFS	Comparator i negative input selection
7	CiRVM	CiRVSn	Set the value of the built-in reference voltage
8	CVRCTL	CVRVSi	Select power and ground with built-in reference voltage
9	CVRCTL	CVREi	1 (built-in reference voltage i allowed to run)
10	Wait for stability time of reference voltage (minimum 20μs)		
11	Set VCIN0, VCIN1x, IVREF0 pin (input), PGAI (input) ^{Note 3} to the analog input function. Function selection for PMCxx, VCIN0 pins, VCi and IVREFi pins. Set the PMCxx bit to 1 (analog input). Set the PMxx bit to 1 (input mode).		
12	COMPMDR	CiENB	1 (Enable operation)
13	Wait for the stability time of the comparator (minimum 3μs)		
14	COMPFIR	CiFCK	Select a sampling clock using or without a digital filter.
		CiEOP, CiEDG	Select the edge detection criteria (rising, falling, or bilateral) for the interrupt request.
15	COMPOCR	CiOP, CiOE	Set the output of the VCOUTi (select polarity, set enable or disable output). Refer to 11.4.4 Output of Comparator i (i=0, 1)
		CiIE	Set the output that enable or disable the interrupt request. Refer to 11.4.4 Output of Comparator i (i=0, 1)
		C1OTWMD	Set TIMER WINDOW output enable/disable for Comparator 1
16	MKxx ^{Note 1}	MKL	When using interrupts: select mask interrupt.
17	IFxx ^{Note 1}	IFL	When using interrupts: 0 (non-interrupt request: initialization) ^{Note 2}

Note 1 MKxx and IFxx are comparator interrupt control registerd, see Chapter 18 Interrupt Function for details.

Note 2 After the comparator is configured and before it reaches a stable operating state, unwanted interrupt requests may occur.

Therefore, the interrupt request flag must be initialized.

Note 3 Comparator 0 must be set when interacting with PGA

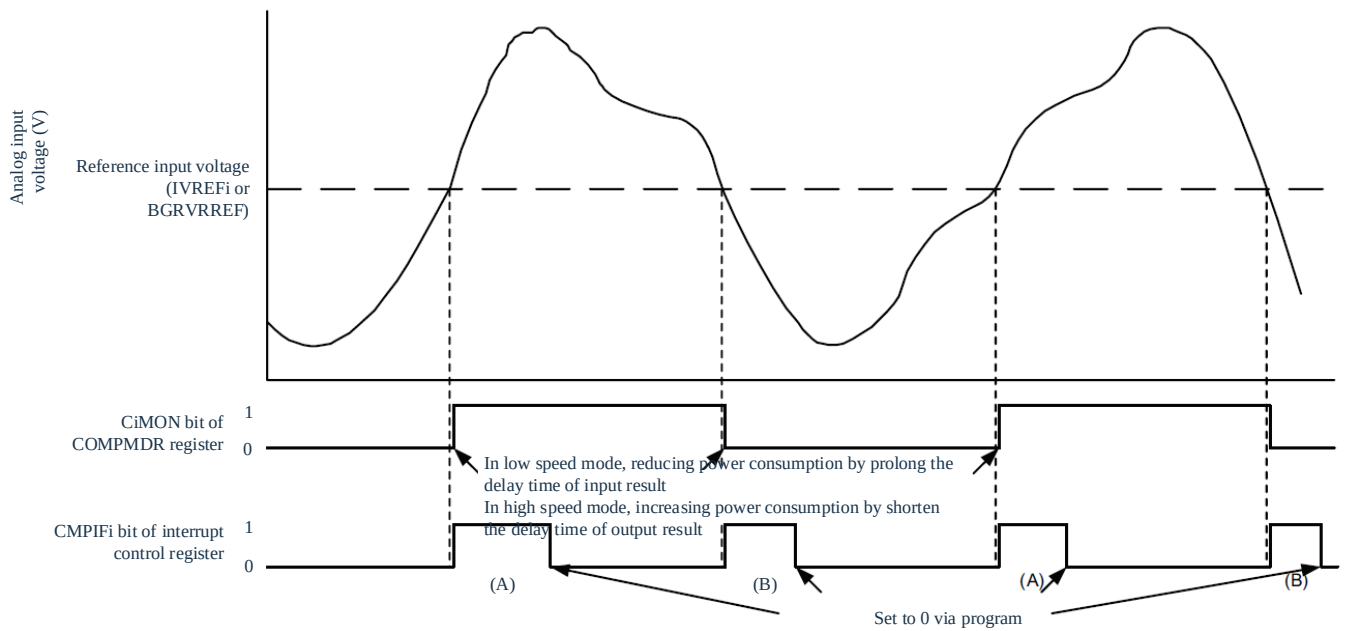
Remark i=0, 1, n=0-7, x=0-3

An example of the operation of comparator i ($i = 0, 1$) is shown in Figure 11-11. In basic mode, when the analog input voltage is higher than the reference input voltage, the CiMON bit in the COMPMDR register is set to 1. When the analog input voltage is lower than the reference input voltage, the CiMON bit is set to 0.

To use the comparator i interrupt, the CiIE bit of the COMPOCR register must be set to 1 (enable interrupt request). At this time, if the comparison result changes, an interrupt request of the comparator i is generated. For more information on interrupt requests, refer to 11.4.2 Comparator i interrupt ($i=0, 1$).

Figure 11-11 Operation example of comparator i ($i=0, 1$) (basic mode)

- Basic mode operation example

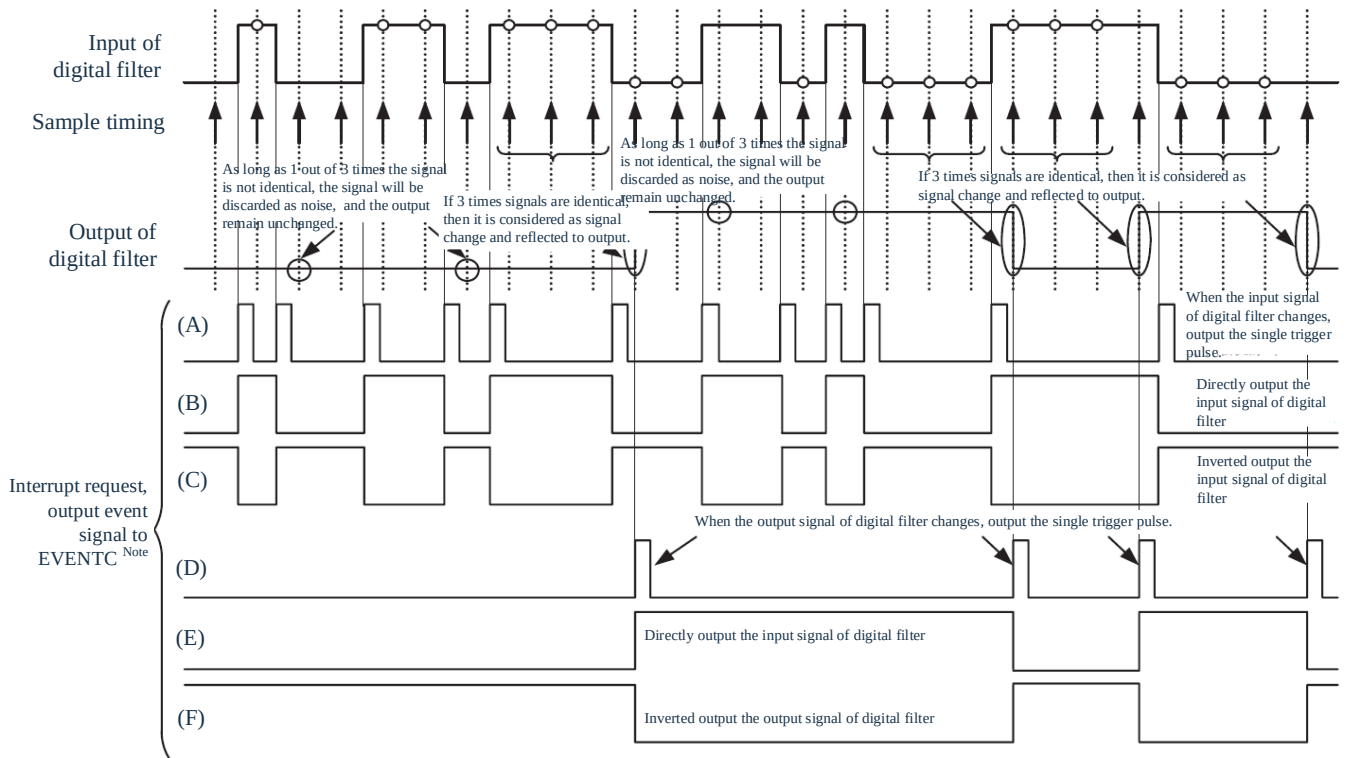


Caution The above figure represents the case where the CiFCK1-CiFCK0 bits in the COMPFIR register are set to “00B” (no filter) and the CiEDG bit is set to 1 (both edges). When the CiEDG bit is 0 and the CiEPO bit is 0 (rising edge), the CMPIFi is triggered only by changes in (A). When the CiEDG bit is 0 and the CiEPO bit is 1 (falling edge), the CMPIFi is triggered only by changes in (B).

11.4.3 Event Signals Output to the Linkage Controller (EVENTC)

An event signal output to the COMPFIR is generated by detecting the output edge of the digital filter set by the EVENTC register. However, unlike an interrupt request, the event signal is always output to the EVENTC regardless of the CiIE bit of the COMPOCR register. You must set the selection of the event output target and the stop of the event link through the EVENTC's ELSELR13 register and ELSELR14 register.

Figure 11-14 Operation of digital filters, interrupt requests and output of event signals to EVENTC



Note When the CiIE bit ($i = 0, 1$) is 1, the interrupt request and the event signal output to EVENTC have the same waveform.
When the CiIE bit ($i = 0, 1$) is 0, only the interrupt request is fixed to 0.

The waveforms (A), (B), and (C) represent the case when the CiFCK bits ($i=0, 1$) in the COMPFIR register are set to “00B” (no digital filter). The waveforms (D), (E), and (F) represent the case when the CiFCK bits ($i=0, 1$) are set to “01B”, “10B”, or “11B” (with digital filter). (A) and (D) correspond to the case where the CiEDG bit is 1 (both edges). (B) and (E) correspond to the case where the CiEDG bit is 0 and the CiEPO bit is 0 (rising edge). (C) and (F) correspond to the case where the CiEDG bit is 0 and the CiEPO bit is 1 (falling edge).

11.4.4 Output of Comparator i (i=0, 1)

The CiOE bit of the COMPOCR register can be used to set whether the comparison result of the comparator is output to an external pin, and the CiOP bit of the COMPOCR register can be used to set the output polarity (positive or negative output). Refer to “11.3.4 Comparator Output Control Register (COMPOCR)” for the corresponding register settings and comparator outputs.

To output the comparator's comparison results to the VCOUTi pin, you must set the port as follows (after reset, the port defaults to the input state):

- ① Set the mode of the comparator (Steps 2-5 of Table 11-3 Set-up steps for comparator-related registers).
- ② Set the VCOUTi output of the comparator (set the COMPOCR register, select polarity, and allow output).
- ③ Set the port mode control register corresponding to the output pin of VCOUTi to 0.
- ④ Set the bit of the port register corresponding to the output pin of VCOUTi to 0.
- ⑤ Set the port direction register corresponding to the output pin of VCOUTi to output (output from the pin).

11.4.5 Stop and Enable Comparator Clock Supply

In the case of stopping the comparator clock by setting the peripheral admission register 1 (PER1), you must follow these steps:

- ① Set the CiENB bit of the COMPMDR register to 0 (to stop the operation of the comparator).
- ② Set the IF bit of the interrupt request flag register to 0 (clear the interrupts that are not needed before the comparator stops running).
- ③ Set the PGACMPEN bit of PER1 register to 0.

If the clock is stopped by setting the PER1 register, all internal registers of the comparator are initialized. Therefore, when the comparator is to be used again, the registers must be set according to the steps in Table 11-3, which outlines the register settings for the comparator.

Cautions:

1. If the comparator n reference voltage selection bit (CnVRF) in the comparator mode register (COMPMDR) is set to 1 (indicating the comparator n reference voltage is the internal reference voltage of 1.45V), the output of the temperature sensor cannot be converted by the A/D converter.
2. If DMA transfer is allowed in the following conditions, the DMA transfer will begin and an interrupt will be generated after the transfer is completed. Therefore, the DMA trigger must be enabled after confirming the comparator's monitoring flag (CnMON) as needed:
 - When it is set to generate an interrupt request through the comparator's single-edge detection (CnEDG = 0) and the rising edge of the comparator (CnEPO = 0), and when $IVCMP > IVREF$ (or the internal reference voltage of 1.45V).
 - When it is set to generate an interrupt request through the comparator's single-edge detection (CnEDG = 0) and the falling edge of the comparator (CnEPO = 1), and when $IVCMP < IVREF$ (or the internal reference voltage of 1.45V).(n=0, 1)

Chapter 12 Programmable Gain Amplifier (PGA)

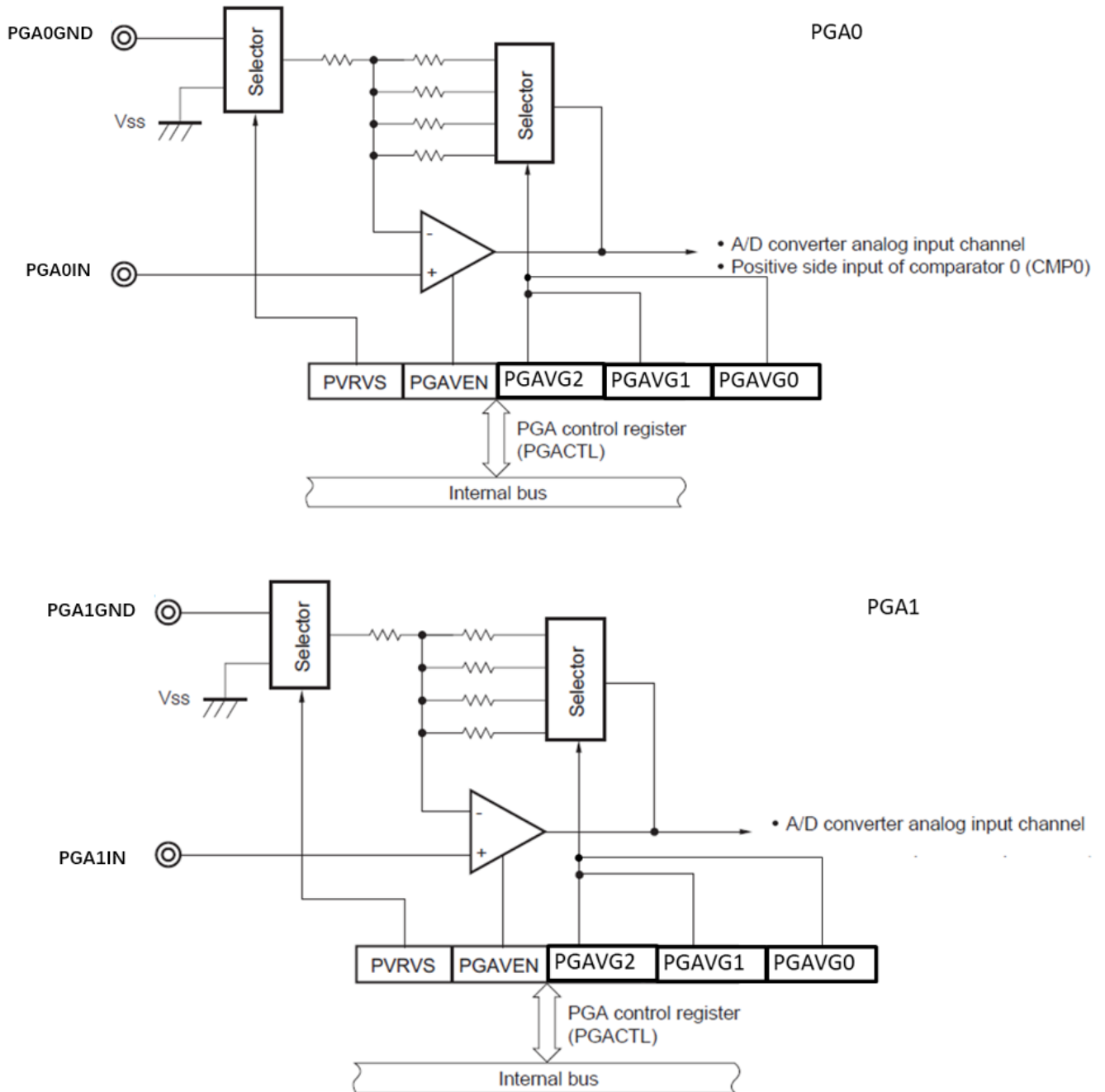
12.1 Functions of Programmable Gain Amplifier

This product has two built-in programmable gain amplifiers (PGA0 and PGA1) with the following functions.

- 7 choices of amplification gain per PGA: 4x, 8x, 10x, 12x, 14x, 16x, 32x.
- External pin can be selected as the ground for the PGA negative feedback resistor
- The output of PGA0 can be selected as an analog input for the A/D converter or an analog input on the positive side of comparator 0 (CMP0)
- The output of PGA1 can be selected as an analog input for the A/D converter

12.2 Structure of Programmable Gain Amplifier

Figure 12-1 Block diagram of programmable gain amplifier



12.3 Registers for Programmable Gain Amplifier

Table 12-1 Registers for controlling programmable gain amplifier

Register name	Symbol
Peripheral enable register 1	PER1
Programmable gain amplifier control register	PGA0CTL
Programmable gain amplifier control register	PGA1CTL
Port mode control register	PMCxx
Port mode register	PMxx

12.3.1 Peripheral Enable Register 1 (PER1)

The PER1 register is a register that sets a clock that is allowed or prohibited to supply to each peripheral hardware.

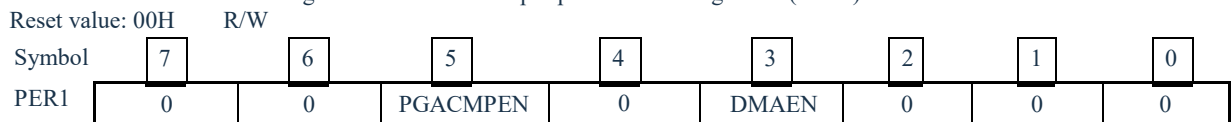
Reduce power consumption and noise by stopping clock supply to unused hardware.

The bit5 (PGACMPEN) of this PER1 register must be set to 1 when using a programmable gain amplifier.

The PER1 register is set by an 8-bit memory manipulation instruction.

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

Figure 12-2 Format of peripheral enable register 1 (PER1)



PGACMPEN	Control of input clock of comparator/programmable gain amplifier
0	Stop input clock supply. Comparator or programmable gain amplifier registers are not writable Comparator or programmable gain amplifier is in the reset status
1	Enable input clock supply. Comparator or programmable gain amplifier registers are readable and writable.

Caution

Before configuring the comparator or programmable gain amplifier (PGA) registers, ensure that the PGACMPEN bit is set to 1.

If PGACMPEN = 0, write operations to the comparator or PGA control registers are ineffective, and all read values will be the default values (except for the port mode registers (PMXX) and port registers (PXX)).

12.3.2 Programmable Gain Amplifier Control Register (PGAnCTL)

The PGA0CTL and PGA1CTL registers are used to control the programmable gain amplifier start, stop and amplification.

The PGA0CTL and PGA1CTL registers can be set by 1-bit or 8-bit memory manipulation instructions. This register is reset to 00H after a reset signal is generated.

Figure 12-3 Format of PGA control register (PGAnCTL)

Reset value: 00H

R/W

	7	6	5	4	3	2	1	
PGAnCTL	PGAnEN	-	-	-	PVRVS Note	PGAnV G2	PGAnV G1	PGAnV G0

n=0,1

PGAnEN	Programmable gain amplifier operation control
0	Stop the amplifier operation
1	Enable the amplifier operation

PVRVS	Selection of feedback resistor ground
0	Select Vss
1	Select PGAnGND pin

PGAnVG2	PGAnVG1	PGAnVG0	PGAn gain
0	0	0	4 x
0	0	1	8 x
0	1	0	10 x
0	1	1	12 x
1	0	0	14 x
1	0	1	16 x
1	1	0	32 x
Others			Disable settings

Caution: With PGAnEN set to 1, the programmable gain amplifier requires a 10us stabilization time to operate.

12.3.3 Registers Controlling Port Functions of Analog Input Pins

When using the PGA0IN pin, PGA1IN pin, PGA0GND pin, and PGA1GND pin as analog inputs to a programmable gain amplifier, you must set the bits in the Port Mode Register (PMxx) and the bits in the Port Digital/Analog Control Register (PMCxx) corresponding to each port to “1”.

12.4 Operation of Programmable Gain Amplifier

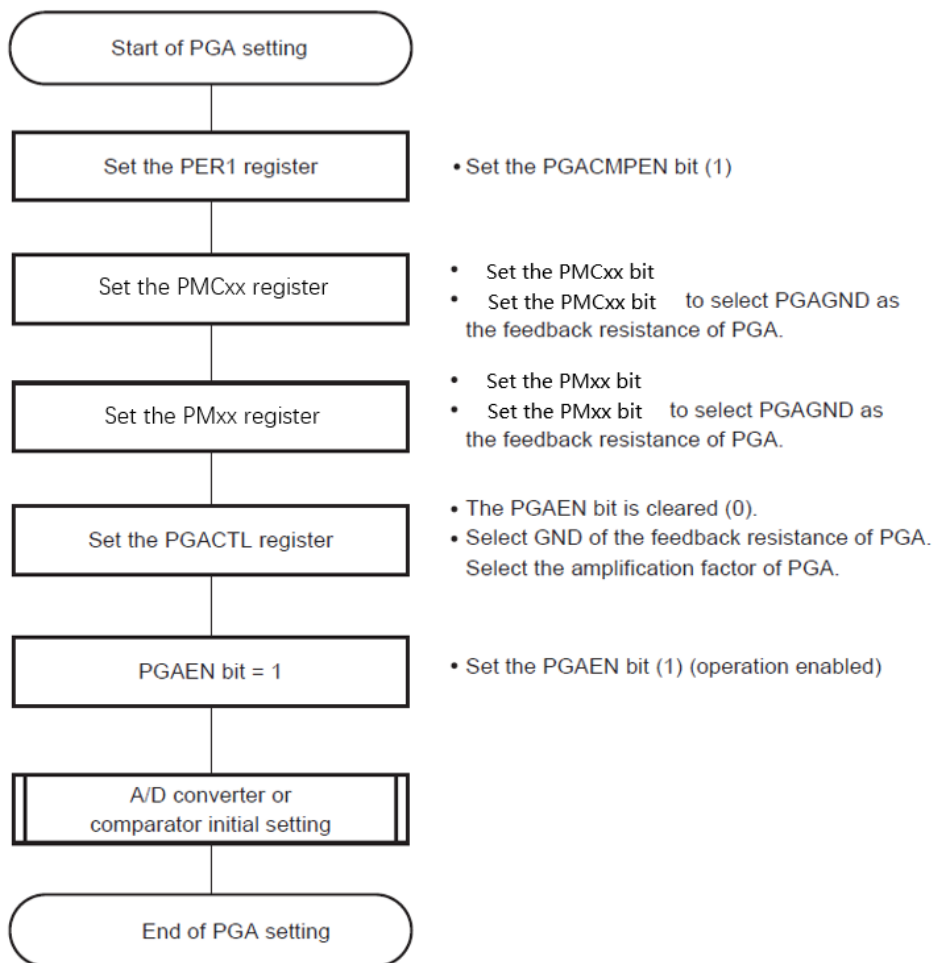
The analog voltage input from the PGAIN pin is amplified with seven choices of amplification gain: 4x, 8x, 10x, 12x, 14x, 16x, and 32x.

The amplified voltage may be used for analog input of the A/D converter and positive input signal of the comparator 0 (CMP 0).

The steps for starting and stopping the programmable gain amplifier are as follows.

12.4.1 Starting Operation Steps of Programmable Gain Amplifier

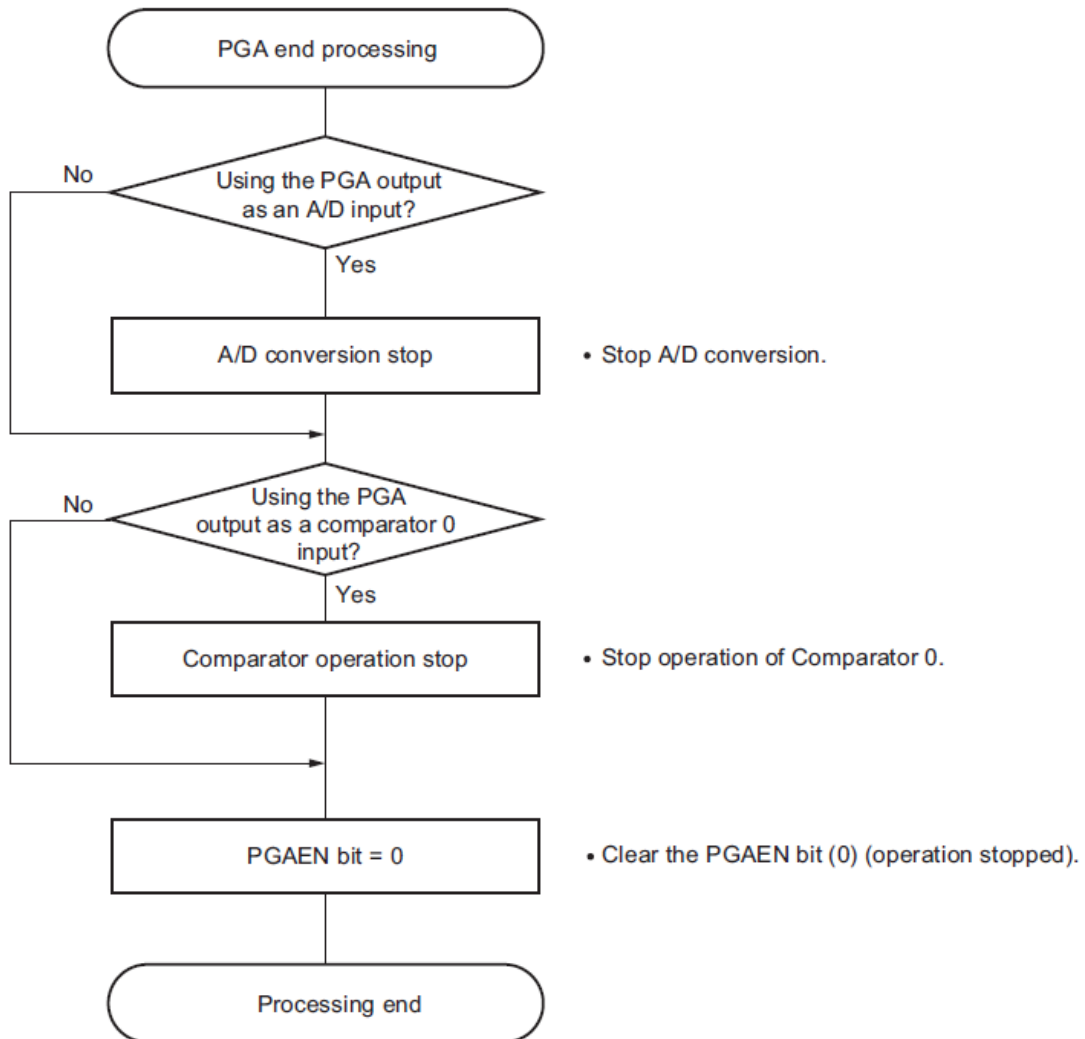
Take PGA0 as an example, set up as follows:



Caution 1. A PGA stabilization time of 10us is required after setting the PGAEN bit to 1. The A/D conversion is then initiated.

12.4.2 Stopping Operation Steps of Programmable Gain Amplifier

Take PGA0 as an example, set up as follows:



Cautions

1. When restarting the PGA and A/D converter or amplifier, a PGA stabilization time of 10 μ s is required after setting the PGAEN bit to 1.
2. Even if the PGA operation is stopped, A/D conversion and comparator operation can be performed using the shoot-through pins.

Chapter 13 Universal Serial Communication Unit

The Universal Serial Communication Unit has four serial channels in Unit 0 and two serial channels in Unit 1, each of which can realize 3-wire serial (SSPI), UART, and simplified I²C communication functions.

The functions of each channel supported by this product are assigned as follows.

Unit	Channel	Used as SSPI	Used as UART	Used as simplified I ² C
0	0	SSPI00 (Slave selection input function supported)	UART0	IIC00
	1	SSPI01		IIC01
	2	SSPI10	UART1	IIC10
	3	SSPI11		IIC11
1	0	SSPI20	UART2	IIC20
	1	SSPI21		IIC21

When UART0 is used for channel 0 and channel 1 of Unit 0, SSPI00 and SSPI01 cannot be used, but SSPI10, UART1, and IIC10 for channel 2 and channel 3 can be used.

Caution The following section of this chapter describes the unit and channel structure of the 24-pin product.

13.1 Functions of Universal Serial Communication Unit

The functions of each serial interface supported by this product are shown below.

13.1.1 3-Wire Serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21)

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 “13.5 3-wire serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21) communication operation”.

[Transmit and receive data]

- 7-bit or 8-bit data length
- Phase control of transmitting and receiving data
- MSB/LSB first

[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 ^{Note}

Master communication: $\text{Max.f}_{\text{CLK}}/2$ (SSPI00 only)

$\text{Max.f}_{\text{CLK}}/4$

Slave communication: $\text{Max.f}_{\text{MCK}}/6$

[Interrupt function]

- Transfer end interrupt, buffer empty interrupt

[Error detection flag]

- Overflow error

Note It must be used within the scope of the SCLK Cycle Time (t_{KCY}) characteristics. Please refer to the data sheet for details.

13.1.2 UART (UART0~UART2)

This function performs asynchronous communication using two lines: Transmit Data (TxD) and Receive Data (RxD). Using these two communication lines, data is transmitted and received asynchronously with other communication partners according to data frames (composed of a start bit, data, parity bit, and stop bit) at 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 setting examples, see “13.7 Operation of UART (UART0~UART2) Communication”.

[Transmit and receive data]

- 7-bit, 8-bit or 9-bit data length ^{Note}
- MSB/LSB first
- Level settings for transmitting and receiving data, selection of reverse phase
- Append parity function for parity bits, parity check function
- Append stop bits

[Interrupt function]

- Transfer end interrupt, buffer empty interrupt
- Error interrupt caused by frame error, parity error, or overflow error

[Error detection flag]

- Frame error, parity error, overflow error

13.1.3 Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21)

This is a function that synchronizes clock communication with multiple devices through a total of 2 lines of serial clock (SCL) and serial data (SDA). Because this simplified I²C is designed for single communication with EEPROM, flash memory, A/D converters, etc., it is only used as a master device.

For start and stop conditions, AC specifications must be adhered to and processed by software while operating the control registers. For specific setup examples, refer to “13.9 Simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21).”

[Transmit and receive data]

- Master transmission, master reception (limited to single master functions).
- ACK output function ^{Note}, ACK detection function
- 8 bits of data length (when transmitting the address, specify the address with a high 7 bits, and use the lowest bit for R/W control).
- Generation of start and stop conditions

[Interrupt function]

- Transfer end interruption

[Error detection flag]

- ACK error, overflow error.

※[Features not supported by simplified I²C]

- Slave transmission, slave reception
- Arbitration failure detection function
- Wait for detection function

Note When receiving the last data, if you write “0” to the SOEmn bit (serial output enable register m (SOEm)) to stop the output of the serial communication data, the ACK is not output. For details, please refer to “13.9.3 Processing Flow”.

Note When using the full-featured I²C-bus, refer to Chapter 14, Serial Interface IICA.

13.2 Structure of Universal Serial Communication Unit

The universal serial communication unit consists of the following hardware.

Table 13-1 Structure of universal serial communication unit

Item	Structure
Shift register	8-bit or 9-bit ^{Note 1}
Buffer register	Serial data register mn (SDRmn) low 8 bits or 9 bits ^{Notes 1, 2}
Serial clock input/output	SCLK00, SCLK01, SCLK10, SCLK11, SCLK20, SCLK21 pins (for 3-wire serial I/O) SCL00, SCL01, SCL10, SCL11, SCL20, SCL21 pins (for simplified I ² C)
Serial data input	SDI00, SDI01, SDI10, SDI11, SDI20, SDI21 pins (for 3-wire serial I/O) RxD0, RxD1, RxD2 pins (for UART)
Serial data output	SDO00, SDO01, SDO10, SDO11, SDO20, SDO21 pins (for 3-wire serial I/O) TxD0, TxD1, TxD2 pins (for UART)
Serial data input/output	SDA00, SDA01, SDA10, SDA11, SDA20, SDA21 pins (for Simplified I ² C)
Slave Select Input	SS00 pin (for 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 (PxxCFG) • Port output mode register (POMxx) • Port mode register (PMxx) • Port register (Pxx)

Notes 1. The number of bits used as shift registers and buffer registers varies by unit and channel.

- mn = 00, 01: low 9-bit
- Others: low 8-bit

2. Depending on the communication mode, the lower 8 bits of the serial data register mn (SDRmn) can be read and written with the following SFR name.

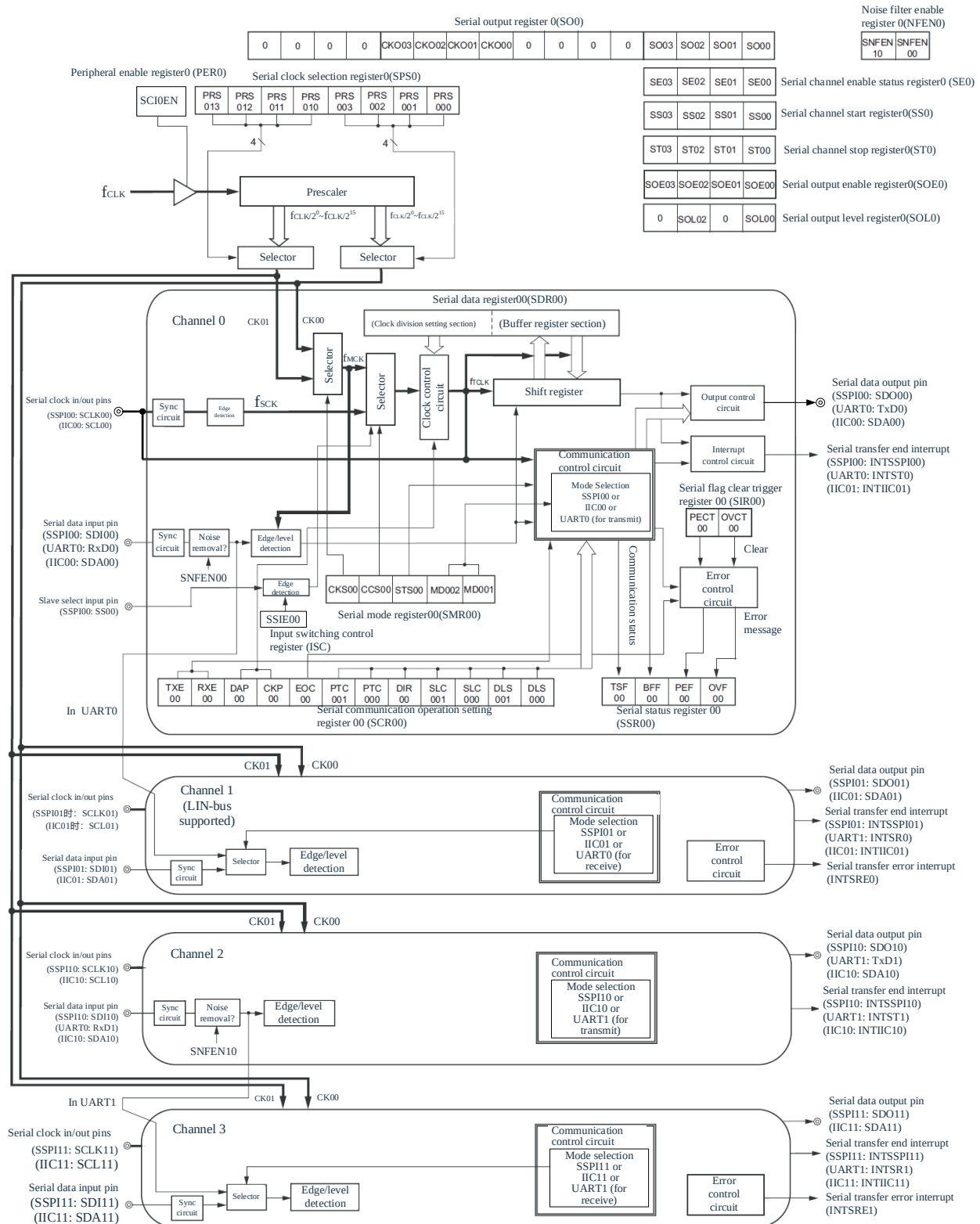
- SSPIp communication... SIOp (SSPIp Data Register)
- UARTq reception... RXDq (UARTq Receive Data Register)
- UARTq transmission... TXDq (UARTq Transmit Data Register)
- IICr communication... SIOr (IICr Data Register)

Remark m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)

q: UART number (q=0~2) r: IIC number (r=00, 01, 10, 11, 20, 21)

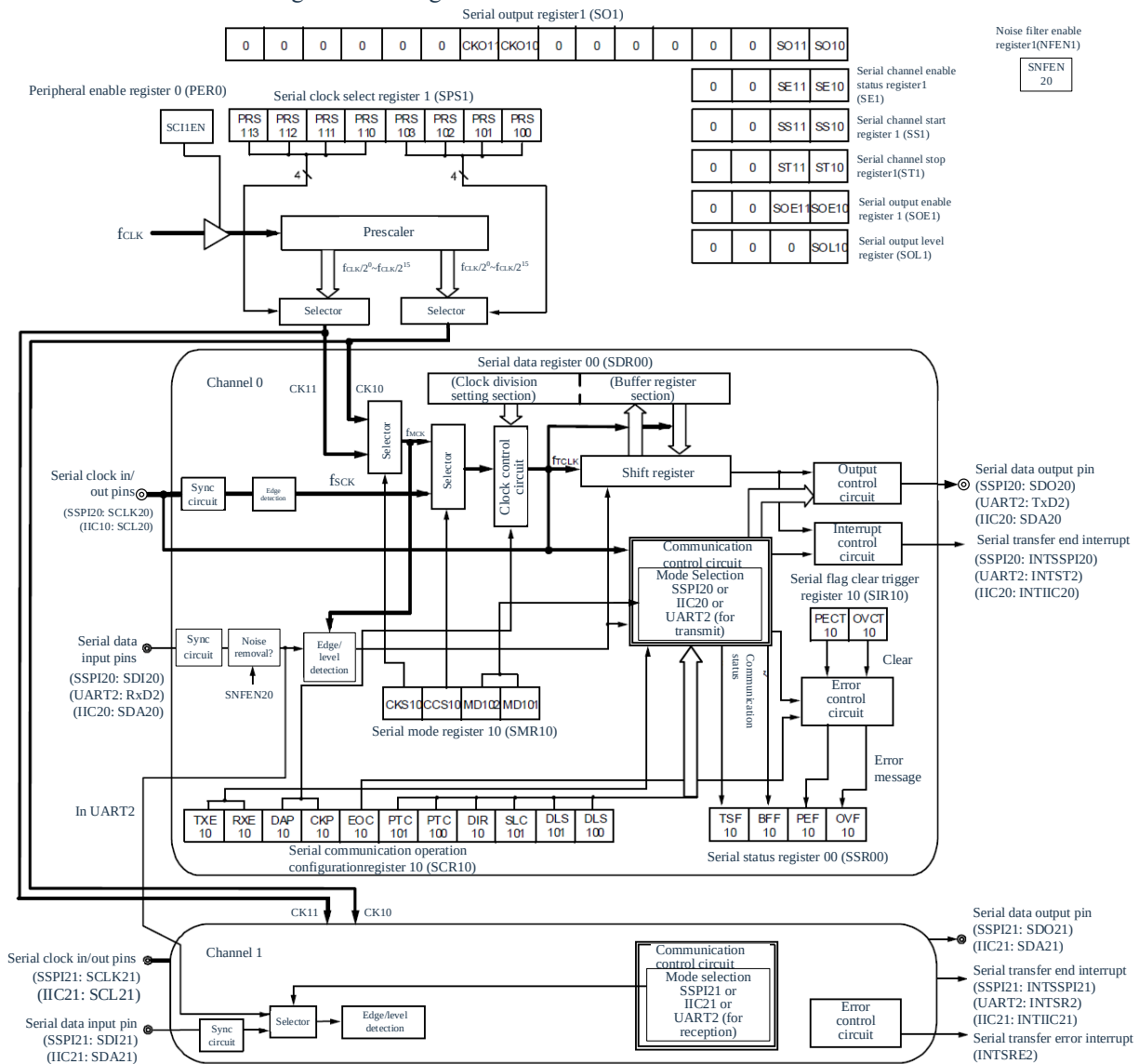
A block diagram of universal serial communication unit 0 is shown in Figure 13-1.

Figure 13-1 Diagram of universal serial communication unit 0



A block diagram of a universal serial communication unit 1 is shown in Figure 13-2.

Figure 13-2 Diagram of universal serial communication unit 1

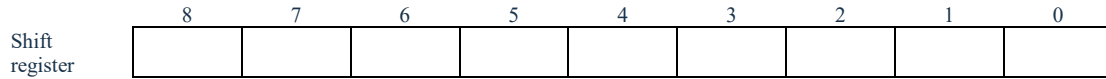


13.2.1 Shift Register

This is a 9-bit register that converts parallel and serial to and from each other.

For UART communication at 9 bits of data length, use 9 bits (bit0 to 8) ^{Note 1}. Converts the input data of the serial input pin into parallel data when receiving data; When data is sent, the value to this register will be transferred as serial data from the serial output pin output ^{Note 1}. Shift registers cannot be manipulated directly through the program.

To read and write data from shift registers, use the low 8 or 9 bits of the serial data register mn (SDRmn).



13.2.2 Low 8 Bits or Low 9 Bits of Serial Data Register mn (SDRmn)

The SDRmn register is the transmit and receive data register (16 bits) for channel n.

Bits 8-0 (low 9 bits) or bits 7-0 (low 8 bits) are used as the transmit and receive buffer registers, while bits 15-9 are used as the clock divider register for the operating clock (f_{MCK}).

During data reception, the parallel data converted by the shift register is stored in the low 8 bits or low 9 bits. During data transmission, the data to be transmitted to the shift register is set in the low 8 bits or low 9 bits.

Regardless of the data output sequence, the data saved in the low 8 bits or low 9 bits is determined by the settings of bits 0 and 1 (DLSmn0, DLSmn1) in the serial communication control register mn (SCRmn), as follows:

- 7-bit data length (saved in bit0~6 of the SDRmn register).
- 8-bit data length (saved in bit0~7 of the SDRmn register).
- 9-bit data length (saved in bit0~8 of the SDRmn register) ^{Note 1}

The SDRmn register can be read or written in 16-bit units.

Depending on the communication mode, the following SFR names can be used to read or write the low 8 bits or low 9 bits of the SDRmn register in 8-bit units ^{Note 2}.

- SSPIp communication... SDIOp (SSPIp Data Register)
- UARTq reception... RXDq (UARTq Receive Data Register)
- UARTq transmission... TXDq (UARTq Transmit Data Register)
- IICr communication... SDIOOr (IICr Data Register)

After the reset signal is generated, the value of the SDRmn register changes to “0000H”.

Notes

1. Only UART0 supports 9-bit data length.
2. When the operation is stopped (SEmn=0), it is forbidden to override SDRmn [7:0] via 8-bit memory manipulation instructions (otherwise, the SDRmn [15:9] is cleared).

Remarks

1. After the reception ends, bits from bit0 to 8 that exceed the length of the data are “0”.
2. m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)
q: UART number (q=0~2) r: IIC number (r=00, 01, 10, 11, 20, 21)

Figure 13-3 Format of serial data register mn (SDRmn) (mn=00, 01, 10, 11)

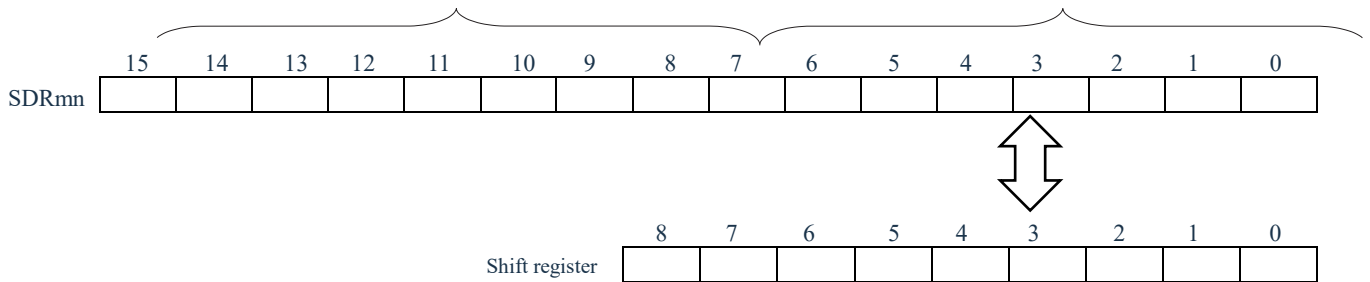
Address: 40041310H (SDR00), 40041312H (SDR01) After reset: 0000H

R/W

40041748H (SDR10), 4004174AH (SDR11)

40041311H (SDR00)

40041310H (SDR00)



Remark For the functions of the higher 7 bits of the SDRmn register, refer to 13.3 Registers for Controlling Universal Serial Communication Unit.

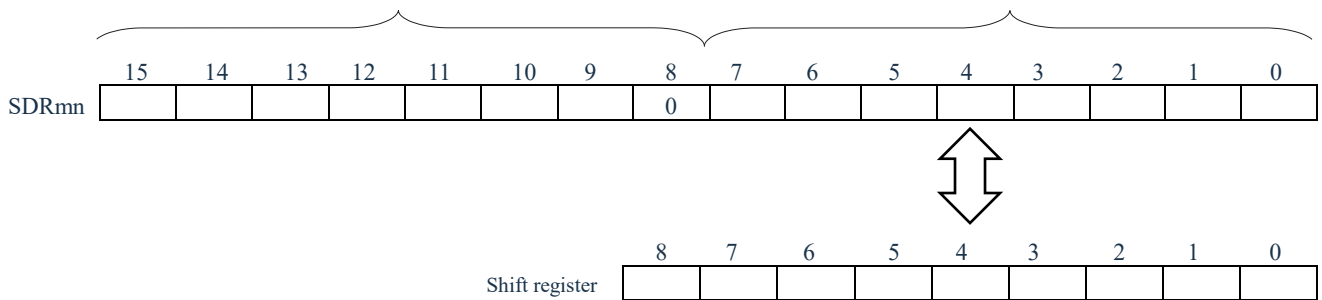
Figure 13-4 Format of serial data register mn (SDRmn) (mn=02, 03)

Address: 40041314H (SDR02), 40041316H (SDR03) After reset: 0000H

R/W

40041315H (SDR02)

40041314H (SDR02)



Caution Bit8 must be set to “0”.

For the functions of the higher 7 bits of the SDRmn register, refer to 13.3 Registers for Controlling Universal Serial Communication Unit.

13.3 Registers for Controlling Universal Serial Communication Unit

The registers that control the universal serial communication unit are as follows:

- Peripheral enable register 0 (PER0)
- Serial clock selection register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation configuration register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SDIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOM)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port alternate function configuration register (PxxCFG)
- Port output mode register (POMx)
- Port mode register (PMx)
- Port register (Px)

Remark m: unit number (m=0, 1) n: channel number (n=0~3).

13.3.1 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 supply to unused hardware, power consumption and noise can be reduced.

To use the Universal Serial Communication Unit 0 (SCI0), bit 2 (SCI0EN) must be set to 1.

To use the Universal Serial Communication Unit 1 (SCI1), bit 3 (SCI1EN) must be set to 1.

The PER0 register is set by an 8-bit memory manipulation instruction.

After the reset signal is generated, the value of the PER0 register changes to “00H”.

Figure 13-5 Format of peripheral enable register 0 (PER0)

Address: 0x40020420		After reset: 00H		R/W				
Symbol	7	6	5	4	3	2	1	0
PER0	RTCCEN	IRDAEN	ADCEN	IICA0EN	SCI1EN	SCI0EN	TM41EN	TM40EN

SCI _m EN	Control of input clock supply of the universal serial communication unit m
0	Stop input clock supply • SFR used by the universal serial communication unit m cannot be written. • Universal serial communication unit m is in the reset status.
1	Enable input clock supply • SFR used by the universal serial communication unit m can be read and written.

Caution 1. To set the universal serial communication unit m, the following registers must first be set when the SCI_mEN bit is “1”. When the SCI_mEN 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 (PxxCFG), and port output mode register (POMx), port mode registers (PMx), port mode control registers (PMCx), and port registers (Px).

- Serial clock selection register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation configuration register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOM)

13.3.2 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”.

Figure 13-6 Format of serial clock selection register m (SPSm)

Address: 40041126H (SPS0), 40041566H (SPS1)

After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00

PRSmk3	PRSmk2	PRSmk1	PRSmk0	Selection of the operating clock (CKmk) ^{Note}
0	0	0	0	f_{CLK}
0	0	0	1	$f_{CLK}/2$
0	0	1	0	$f_{CLK}/2^2$
0	0	1	1	$f_{CLK}/2^3$
0	1	0	0	$f_{CLK}/2^4$
0	1	0	1	$f_{CLK}/2^5$
0	1	1	0	$f_{CLK}/2^6$
0	1	1	1	$f_{CLK}/2^7$
1	0	0	0	$f_{CLK}/2^8$
1	0	0	1	$f_{CLK}/2^9$
1	0	1	0	$f_{CLK}/2^{10}$
1	0	1	1	$f_{CLK}/2^{11}$
1	1	0	0	$f_{CLK}/2^{12}$
1	1	0	1	$f_{CLK}/2^{13}$
1	1	1	0	$f_{CLK}/2^{14}$
1	1	1	1	$f_{CLK}/2^{15}$

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”.

Remarks

1. f_{CLK} : CPU/peripheral hardware clock frequency
2. m: unit number (m=0, 1)
3. k=0, 1

13.3.3 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”.

Figure 13-7 Format of serial mode register mn (SMRmn) (1/2)

Address: 40041110H (SMR00)~40041116H (SMR03) After reset: 0020H

R/W

40041550H (SMR10)~40041552H (SMR11)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn _n Note1	0	SISmn ₀ Note1	1	0	0	MDmn2	MDmn1	MDmn0

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.	

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.	

STSmn _{Note1}	Selection of start triggering sources
0	Only software triggers are valid (selected in SSPI, UART Send, 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”.	

Note1 SMR01, SMR03, SMR11 registers only.

Caution Set bits 13~9, 7, 4, and 3 (SMR00, SMR02, SMR10 registers are bit13~6, 4, 3) to 0, and set the bit5 to 1.

Remark m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)

q: UART number (q=0~2) r: IIC number (r=00, 01, 10, 11, 20, 21)

Figure 13-7 Format of serial mode register mn (SMRmn) (2/2)

Address: 40041110H (SMR00)~40041116H (SMR03) After reset: 0020H

R/W

40041550H (SMR10)~40041552H (SMR11)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STSm n ^{Note1}	0	SISmn0 Note1	1	0	0	MD mn2	MD mn1	MD mn0

SISmn0 ^{Note1}	Level inversion control of received data for channel n in UART mode
0	Detect the falling edge as the starting bit. The input communication data is not reversed.
1	Detect the rising edge as the starting point. The input communication data is reversed.

MDmn2	MDmn1	Setting of the channel n operating mode
0	0	SSPI mode
0	1	UART mode
1	0	Simplified I ² C mode
1	1	Disable settings

MDmn0	Channel n interrupt source selection
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register).
In continuous transmission, if the MDmn0 bit is “1” and the data for SDRmn is empty, the next transmitted data is made.	

Note 1 SMR01, SMR03, SMR11, registers only.

Caution Set bits 13 to 9, 7, 4, 3 (bits 13~6, 4, 3 of SMR00, SMR02, SMR10 registers) to “0”, and set the bit5 to 1”.

Remark m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)
q: UART number (q=0~2) r: IIC number (r=00, 01, 10, 11, 20, 21)

13.3.4 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 “0087H”.

Figure 13-8 Format of serial communication operation configuration register mn (SCRmn) (1/2)

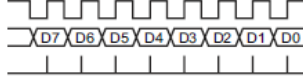
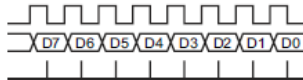
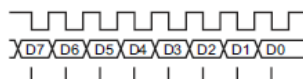
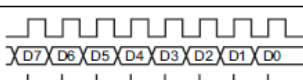
Address: 40041118H (SCR00)~4004111EH (SCR03) After reset: 0087H

R/W

40041558H (SCR10)~4004155AH (SCR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn mn	RXEmn mn	DAPmn mn	CKPmn mn	0	EOCmn mn	PTCmn1 mn1	PTCmn0 mn0	DIRmn mn	0	SLCmn1 n1Note1	SLCmn0 mn0	0	1	DLSmn1 n1Note2	DLSmn0 mn0

TXEmn	RXEmn	Setting of the channel n operating mode
0	0	Disable communication.
0	1	Receive only.
1	0	Transmit only.
1	1	Enable transmission and reception.

DAPmn	CKPmn	Data and clock phase selection in SSPI mode	Type
0	0		1
0	1		2
1	0		3
1	1		4

In UART mode and simplified I²C mode, set the DAPmn bit and CKPmn bit to 0.

EOCmn	Mask control of error interrupt signal (INTSREx (x=0 to 3))
0	Disable the generation of error interrupt INTSREx (generate INTSRx).
1	Enable the generation of error interrupt INTSREx (no INTSRx is generated when an error occurs).

The EOCmn bit must be set to 0 in SSPI mode and Simplified I²C mode or in UART transmission ^{Note 3}.

Notes 1. Limited to SCR00, SCR02, SCR10 registers only.

2. Limited to SCR00 register and SCR01 register, the others are fixed as “1”.

3. When the EOCmn bit is “0” and SSPImn is not used, it is possible to generate an error interrupt INTSREn.

Caution Bit 3, 6, and 11 must be set to 0 (bit 5 of SCR01, SCR03, and SCR11 registers must be set to 0), and bit 2 must be set to 1.

Remark m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)

Figure 13-8 Format of serial communication operation configuration register mn (SCRmn) (2/2)

Address: 40041118H (SCR00)~4004111EH (SCR03) After reset: 0087H

R/W

40041558H (SCR10)~4004155AH (SCR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n ^{Note1}	SLC mn0	0	1	DLSm n ^{Note2}	DLS mn0

PTCmn1	PTCmn0	Setting of parity bits in UART mode	
		Transmission	Reception
0	0	Parity bits are not output.	There is no parity at the time of reception.
0	1	Output parity ^{Note 3}	Parity is not judged.
1	0	Output even parity.	Judge even parity.
1	1	Output odd parity.	Judge odd parity.

In SSPI mode and Simplified I²C mode, both the PTCmn1 bit and the PTCmn0 bit must be set to “0”.

DIRmn	Selection of data transfer order in SSPI and UART modes
0	Perform MSB-priority input/output.
1	Perform LSB-priority input/output.

In Simplified I²C mode, the DIRmn bit must be “0”.

SLCmn1 ^{Note1}	SLCmn0	Setting of the stop bit in UART mode
0	0	No stop bits
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits (limited to mn=00, 02, 10).
1	1	Disable settings.

If a transfer end interrupt is selected, an interrupt occurs after all stop bits have been transferred.
When UART is received or in Simplified I²C mode, it must be set to 1 stop bit (SLCmn1, SLCmn0=0, 1).
In SSPI mode, it must be set to no stop bit (SLCmn1, SLCmn0=0, 0).
When UART is transmitted, it must be set to 1 bit (SLCmn1, SLCmn0=0, 1) or 2 bits (SLCmn1, SLCmn0=1, 0).

DLSmn1 ^{Note2}	DLSmn0	Setting of data length in SSPI and UART modes
0	1	Data length of 9 bits (saved in bit0~8 of the SDRmn register) (selectable only in UART mode).
1	0	Data length of 7 bits (saved in bit0 to 6 of the SDRmn register).
1	1	Data length of 8 bits (bit0 to 7 in the SDRmn register).
Other		Disable settings.

In Simplified I²C mode, both the DLSmn1 bit and the DLSmn0 bit must be set to “1”.

Notes 1. Limited to SCR00, SCR02, SCR10 registers only.

2. Limited to SCR00 register and SCR01 register, the others are fixed as “1”.

3. Nothing to do with the content of the data, always appended “0”.

Caution Bits 3, 6, and 11 must be set to 0 (also bit 5 of the SCR01, SCR03, and SCR11 registers must be set to 0), and bit 2 must be set to 1.

Remark m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)

13.3.5 Serial Data Register mn (SDRmn)

The SDRmn register is a 16-bit register used for transmitting and receiving data on channel n.

The low 9 bits (bits 8–0) of SDR00, SDR01, SDR10, and SDR11, or the low 8 bits (bits 7–0) of SDR02 and SDR03, serve as the transmit and receive buffer registers. The high 7 bits (bits 15–9) are used as the clock divider settings for the operating clock (f_{MCK}).

If the serial mode register mn (SMRmn) has the CCSmn bit set to 0, the clock divider for the operating clock set by bits 15–9 of the SDRmn register is used as the transfer clock.

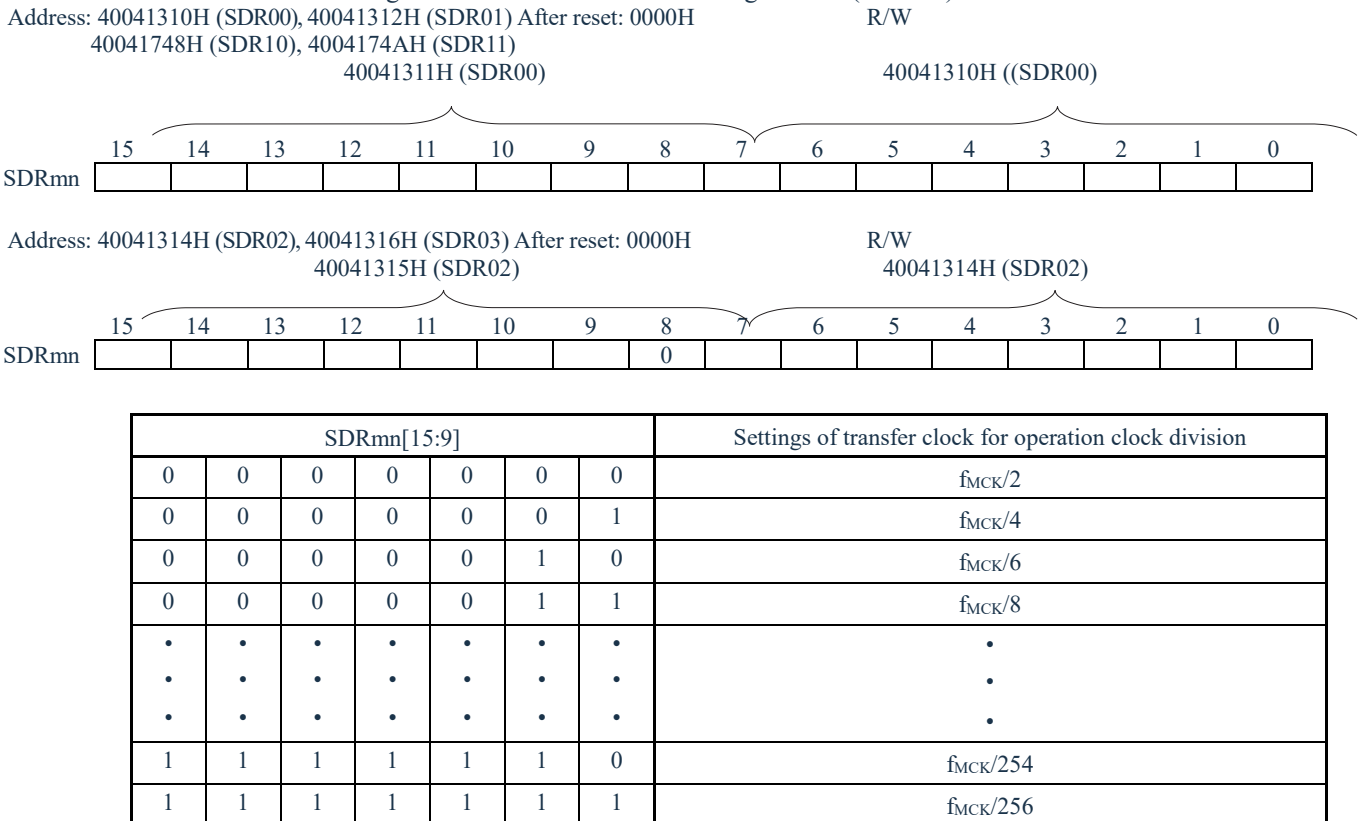
If CCSmn is set to 1, bits 15–9 of SDRmn must be set to “0000000B”. The input clock to the SCLKp pin, f_{SCLK} (for SSPI slave transfer mode), is used as the transfer clock.

The low 8 or 9 bits of the SDRmn register are used as the transmit and receive buffer registers. During data reception, parallel data converted by the shift register is saved in the low 8 or 9 bits. During data transmission, the data to be transmitted to the shift register is set in the low 8 or 9 bits.

The SDRmn register can be read or written in 16-bit units. However, the high 7 bits can only be read or written when the operation is stopped ($SEmn = 0$). During operation ($SEmn = 1$), only the low 8 or 9 bits of the SDRmn register can be written, and the value read from the high 7 bits is always 0.

After the reset signal is generated, the value of the SDRmn register changes to “0000H”.

Figure 13-9 Format of serial data register mn (SDRmn)



Notes

1. Bit8 of the SDR02 and SDR03 registers must be set to “0”.
2. When using UART, it is forbidden to set SDRmn [15:9] to “0000000B” and “0000001B”.
3. When using Simplified I²C, it is forbidden to set SDRmn[15:9] to “0000000B”, and the SDRmn [15:9] setting value must be greater than or equal to “0000001B”.
4. When the operation is stopped ($SEmn=0$), it is forbidden to override SDRmn [7:0] via 8-bit memory manipulation instructions

(otherwise, SDRmn [15:9] is all cleared “0”).

Remarks 1. For the function of the SDRmn register with the low 8 or 9 bits, refer to “13.2 Structure of universal serial communication unit

2. m: unit number (m=0, 1) n: channel number (n=0~3).

13.3.6 Serial Flag Clear Trigger Register mn (SIRmn)

This is the trigger register used to clear each error flag of channel n.

If you set the bits (FECTmn, PECTmn, OVCTmn) to “1”, the corresponding bits (FEFmn, PEFmn, OVFmn) to “0”. Because the SDIRmn register is a trigger register, if the corresponding bit of the SSRmn register is cleared, the SDIRmn register is also cleared immediately.

The SIRmn register is set by a 16-bit memory manipulation instruction.

The low 8 bits of the SIRmn register can be set with SIRmnL and via 8-bit memory manipulation instructions.

After the reset signal is generated, the value of the SIRmn register changes to “0000H”.

Figure 13-10 Format of serial flag clear trigger register mn (SIRmn)

Address: 40041108H (SIR00)~4004110EH (SIR03) After reset: 0000H R/W

40041548H (SIR10)~4004154AH (SIR11)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECT _{mn} ^{Note1}	PEC _{Tmn}	OVC _{Tmn}

FECT _{mn} ^{Note1}	Channel n-frame error flag clear trigger
0	No cleared.
1	Clear the FEFMN bit of the SSRmn register to “0”.

PECT _{mn}	Channel n parity error flag clear trigger
0	No cleared.
1	Clear the PEFmn bit of the SSRmn register to “0”.

OVCT _{mn}	Channel n overflow error flag clear trigger
0	No cleared.
1	Clear the OVFmn bit of the SSRmn register to “0”.

Note 1 Restricted to SIR01, SIR03, SIR11 registers only.

Caution Bit 15 to 3 (bit 15 to 2 for SIR00, SIR02 and SIR10 registers) must be set to 0.

Remarks

- 1.m: unit number (m=0, 1) n: channel number (n=0~3).
2. The read value of the SIRmn register is always “0000H”.

13.3.7 Serial Status Register mn (SSRmn)

The SSRmn register represents the communication status and error conditions for channel n. The errors indicated include frame errors, parity errors, and overflow errors.

The SSRmn register is read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSRmn register can be read with SSRmnL and via 8-bit memory manipulation instructions.

After the reset signal is generated, the value of the SSRmn register changes to “0000H”.

Figure 13-11 Format of serial status register mn (SSRmn) (1/2)

Address: 40041100H (SSR00)~40041106H (SSR03) After reset: 0000H R
40041540H (SSR10)~40041542H (SSR11)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn ^{Note1}	PEF mn	OVF mn

TSFmn	Indication flag for channel n communication state
0	Communication stop state or communication standby state
1	Communication operation state
[Clear condition]	
- When STmn of STm register is set to 1 (communication stopped state) or SSmn bit of SSm register is set to 1 (communication standby state) - When communication ends	
[Set condition]	
When communication begins	

BFFmn	Indication flag for channel n buffer register
0	The SDRmn register does not hold valid data.
1	The SDRmn register holds valid data.
[Clear condition]	
- When the transmitted data from the SDRmn register to the shift register is transferred during the transmit process - When the received data is finished reading from the SDRmn register during the receive process - When the STmn bit of STmn register is set to 1 (communication stopped state) or the SSm bit of the SSm register is set to 1 (communication enable state)	
[Set condition]	
- When writing data to the SDRmn register in the state where the TXEmn bit of the SCRmn register is 1 (transmit mode, transmit and receive mode in each communication mode). - When the received data is saved 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	

Note 1 For SSR01, SSR03, SSR11 registers only.

Remark m: unit number (m=0, 1) n: channel number (n=0~3).

Figure 13-11 Format of serial status register mn (SSRmn) (2/2)

Address: 40041100H (SSR00)~40041106H (SSR03)

After reset: 0000H R

40041540H (SSR10)~40041542H (SSR11)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn ^{Note1}	PEF mn	OVF mn

FEFmn ^{Note1}	Detection flag for channel n frame errors
0	No errors occurred.
1	An error occurred (in UART reception).
[Clear condition] • When writing “1” to the FECTmn bit of the SIRmn register [Accent condition]. • When no stop bit is detected at the end of UART reception	

PEFmn	Detection flag for channel n parity errors
0	No errors occurred.
1	An error occurred (in UART reception) or the 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 the parity and parity bits of the data sent at the end of the UART reception are different (parity errors). • When it is sent by I ² C and when the ACK receiving timing slave does not return an ACK signal (no ACK detected).	

OVFmn	Detection flag for channel n overflow error
0	No errors occurred.
1	An error occurred.
[Clear condition] • When writing “1” to the OVCTmn bit of the SIRmn register [Set condition] • In the state where the RXEmn bit of the SCRmn register is “1” (receive mode, transmit and receive mode in each communication mode), although the received data is saved in the SDRmn register, but when there is no read receive data and write send data or write down a received data • When data is not ready to be sent during slave send or slave send and receive in SSPI mode.	

Note 1 For SSR01, SSR03, SSR11 registers only.

Caution 1. If you write the SDRmn register when the BFFmn bit is “1”, the saved transmit or receive data is corrupted and an overflow error is detected (OVEmn=1).

Remark m: unit number (m=0, 1) n: channel number (n=0~3).

13.3.8 Serial Channel Start Register m (SSm)

The SSm register is a trigger register that sets the communication/start count enabled for each channel.

If a “1” is written to each bit (SSmn), the corresponding bit (SEmn) of the serial channel enable register m (SEm) will be set to “1” (allowing operation). Since the SSmn bit is a trigger bit, if the SEmn bit is “1”, the SSmn bit will be cleared immediately.

The SSm register is set by a 16-bit memory manipulation instruction.

The low 8 bits of the SSm register can be set using the SSmL and through 8-bit memory manipulation instructions.

After the reset signal is generated, the value of the SSm register changes to “0000H”.

Figure 13-12 Format of serial channel start register m (SSm)

Address: 40041122H (SS0)	After reset: 0000H								R/W							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00

Address: 40041562H (SS1)	After reset: 0000H								R/W							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS11	SS10

SSmn	Trigger at the beginning of channel n operation															
0	No triggering.															
1	Sets the SEmn bit to 1 and shift to communication standby state ^{Note} .															

Note If the SSmn bit is set to “1” during communication, the communication is stopped and enters standby state. At this point, the values of the control register and shift register, the SCLKmn pin and the SDOmn pin, the FEFmn flag, the PEFmn flag, and the OVFn flag remain in state.

Cautions

1. Bits 15 to 4 of SS0 register and bits 15 to 2 of SS1 register must be set to 0.
2. In UART reception, the SSmn bit must be set to 1 after at least 4 f_{MCK} clock cycles have elapsed, following the setting of the RXEmn bit in the SCRmn register to 1.

Remarks

1. m: unit number (m=0, 1) n: channel number (n=0~3).
2. The read value of the SSm register is always “0000H”.

13.3.9 Serial Channel Stop Register m (STm)

The STm register is a trigger register that sets the communication/stop count allowed for each channel.

If a “1” is written to each bit (STmn), the corresponding bit (SEmn) in the serial channel enable status register m (SEm) is cleared to “0” (stop status). 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 via a 16-bit memory manipulation instruction.

The low 8 bits of the STm register can be set with STmL and via 8-bit memory manipulation instructions.

After the reset signal is generated, the value of the STm register changes to “0000H”.

Figure 13-13 Format of serial channel stop register m (STm)

Address: 40041124H (ST0)	After reset: 0000H								R/W							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	ST03	ST02	ST01	ST00

Address: 40041564H (ST1)	After reset: 0000H								R/W							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST11	ST10

STmn	Stop trigger for channel n operation														
0	No triggering.														
1	Clear the SEmn bit to “0” to stop the communication from running.														

Note The control register and shift register values, the SCLKmn pin and SDOmn pin, and the FEFmn flag, PEFmn flag, and OVFn flag hold status.

Caution Bit15~4 of the ST0 register and bit15~2 of the ST1 register must be set to “0”.

Remarks

1. m: unit number (m=0, 1) n: channel number (n=0~3).
2. The read value of the STm register is always “0000H”.

13.3.10 Serial Channel Enable Status Register m (SEm)

The SEm register is used to confirm the allow or stop status of serial transmission and reception for each channel.

If a “1” is written to each of the serial start allow register m (SSm), the corresponding bit is set to 1. If you write 1 to each bit of the serial channel stop register m (STm), the corresponding bit is cleared to 0.

For channel n that is enabled to run, the value of the CKOmn bit (the serial clock output of channel n) 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 clock pin.

For a stopped channel n, the value of the CKOmn bit of the SOM register can be set by software and output from the serial clock pin. Thus, an arbitrary waveform such as a start condition or a stop condition can be generated by software.

The SEm register is read via a 16-bit memory manipulation instruction.

The lower 8 bits of the SEm register can be read with SEML and via 8-bit memory manipulation instructions.

After the reset signal is generated, the value of the SEm register changes to “0000H”.

Figure 13-14 Format of serial channel enable status register m (SEm)

Address: 40041120H (SE0)	After reset: 0000H								R							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	SE03	SE02	SE01	SE00

Address: 40041560H (SE1)	After reset: 0000H								R							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE11	SE10

SEmn	Indication of the enable or stop state of channel n operation														
0	Run stop status														
1	Run enable status														

Remark m: unit number (m=0, 1) n: channel number (n=0~3).

13.3.11 Serial Output Enable Register m(SOEm)

The SOEm register settings allow or stop the output of serial communication for 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 that stops the serial output, the value of the SOMn bit of the SOM register can be set by software and output from the serial data output pin. Thus, an arbitrary waveform such as a start condition or a stop condition can be generated by software.

The SOEm register is set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOEm register can be set with SOEmL and by 8-bit memory manipulation instructions.

After the reset signal is generated, the value of the SOEm register changes to “0000H”.

Figure 13-15 Format of serial output enable register m (SOEm)

Address: 4004112AH	After reset: 0000H								R/W							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	SOE 03	SOE 02	SOE 01	SOE 00

Address: 4004156AH	After reset: 0000H								R/W							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE 11	SOE 10

SOE mn	Enable or stop of channel n serial output														
0	Stop the output of serial communication.														
1	Enable the output of serial communication.														

Note Bits 15~4 of the SOE0 register and bits 15~2 of the SOE1 register must be set to “0”.

Caution m: unit number (m=0, 1) n: channel number (n=0~3).

13.3.12 Serial Output Register m (SOM)

The SOM register is a buffer register for the 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_n bit of this register is output from the serial clock output pin of channel n.

The SOMn bit of this register can only be rewritten by software when serial output is disabled (SOEm_n=0). When serial output (SOEm_n=1) is allowed, the value of the SOMn bit of this register can only be changed by serial communication by software overwriting.

The CKOm_n bit of this register can only be rewritten by software only when the channel is stopped (SEm_n=0). When allowing the channel to run (SEm_n=1), the value of the CKOm_n bit of this register can only be changed by serial communication by overriding the software.

To use the serial interface pins for non-serial interface functions such as port functions, the corresponding CKOm_n bit and SOMn bit must be set to “1”.

The SOM register is set by a 16-bit memory manipulation instruction.

After the reset signal is generated, the value of the SOM register changes to “0F0FH”.

Figure 13-16 Format of serial output register m (SOM)

Address: 40041128H

After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	CKO 03	CKO 02	CKO 01	CKO 00	0	0	0	0	SO 03	SO 02	SO 01	SO 00

Address: 40041568H

After reset: 0303H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	0	0	CKO 11	CKO 10	0	0	0	0	0	0	SO 11	SO 10

CKO mn	Serial clock output for channel n
0	The output value of the serial clock is “0”.
1	The output value of the serial clock is “1”.

SO mn	Serial data output for channel n
0	The output value of the serial data is “0”.
1	The output value of the serial data is “1”.

Caution Bits 15~12 and bits 7~4 of the SO0 register must be set to “0”.

Bits 15~10 and bits 7~2 of the SO1 register must be set to “0”.

Remark m: unit number (m=0, 1) n: channel number (n=0~3).

13.3.13 Serial Output Level Register m (SOLm)

The SOLm register is used to set the inverted data output level for each channel.

This register can be set only in UART mode. In SSPI mode and Simplified I²C mode, the corresponding bit must be set to “0”. The inversion setting for each channel (n) will be reflected in the pin output only when serial output is enabled (SOEmn = 1). When serial output is disabled (SOEmn = 0), the value of the SOmn bit is directly output.

The SOLm register is set by a 16-bit memory manipulation instruction.

The low 8 bits of the SDOLm register can be set with SOLmL and by 8-bit memory manipulation instructions.

After the reset signal is generated, the value of the SOLm register changes to “0000H”.

Figure 13-17 Format of serial output level register m (SOLm)

Address: 40041134H	After reset: 0000H								R/W							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL 02	0	SOL 00

Address: 40041574H	After reset: 0000H								R/W							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL 10

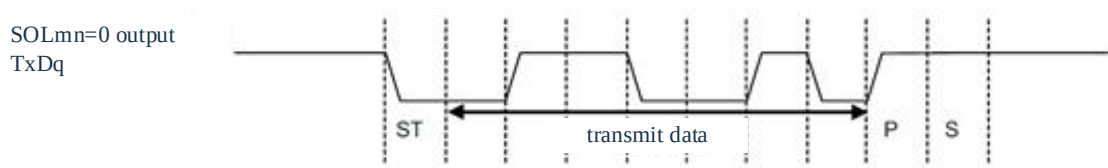
SOL mn	Selection for inverting the data output level for channel n in UART mode
0	The communication data is output directly.
1	The communication data is output inversely.

Caution Remark Bits 15 to 3 and bit1 of the SOL0 register and the bits 15 to 1 of the SOL1 register must be set to “0”.
m: unit number (m=0, 1) n: channel number (n=0, 2).

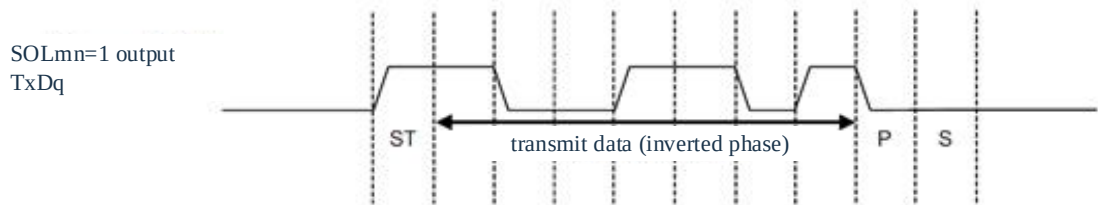
When UART is transmitted, an example of the level inversion of the transmitted data is shown in Figure 13-18.

Figure 13-18 Example of level inversion of transmitted data

(a) Non-inverting output (SOLmn=0)



(b) Inverting output (SOLmn=1)



Remark m: unit number (m=0, 1) n: channel number (n=0, 2).

13.3.14 Input Switching Control Register (ISC)

In UART0-based LIN-bus communication, the ISC register's ISC1 and ISC0 bits are used to coordinate external interrupts and the timer array unit. If bit 0 is set to 1, the serial data input (RxD0) pin's input signal is selected as the external interrupt input (INTP0), enabling wake-up signal detection via the INTP0 interrupt.

If bit 1 is set to 1, the serial data input (RxD0) pin's input signal is selected as the timer's input, allowing wake-up signal detection through the timer and measuring the low-level width of the break field and the pulse width of the synchronization field.

The SS1E00 bit controls the channel 0 SS00 pin input in the SSPI00 communication slave mode. During a high-level input on the SS00 pin, even if the serial clock is received, no transmission or reception occurs. When the SS00 pin is input with a low level, and if the serial clock is received, transmission and reception will be carried out according to the settings of the various modes.

After the reset signal is generated, the value of the ISC register changes to "00H".

Figure 13-19 Format of input switching control register (ISC)

Address: 40040473H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	SSIE00	0	0	0	0	0	ISC1	ISC0

SSIE00	Setting of SS00 input of channel 0 in slave mode for SSPI00 communication
0	The SS00 pin input is invalid.
1	The SS00 pin input is valid.

ISC1	Input switching of channel 3 of timer Timer4
0	Use the input signal from the TI03 pin as the input to the timer (usually running).
1	Use the input signal from the RxD0 pin as the input to the timer (detects the wake-up signal and measures the low-level width of the break field and the pulse width of the sync field).

ISC0	Input switching for external interrupt (INTP0)
0	Use the input signal from the INTP0 pin as the input for an external interrupt (usually in operation).
1	Use the input signal from the RxD0 pin as the input for the external interrupt (detect wake-up signal).

Caution Bits 6~0 must be set to "0".

13.3.15 Noise Filter Enable Register 0 (NFEN0)

The NFEN0 register sets whether the noise filter is used for the input signal of the serial data input pins of each channel.

For pins used for SSPI or simplified I²C communication, the corresponding bit must be “0” to invalidate the noise filter.

For the pins used for UART communication, the corresponding bit must be set to “1” to make the noise filter active.

When the noise filter is active, detect whether the two clocks are consistent after synchronization through the running clock (f_{MCK}) of the target channel; When the noise filter is invalid, synchronization is performed only through the running clock (f_{MCK}) of the target channel.

The NFEN0 register is set by an 8-bit memory manipulation instruction.

After the reset signal is generated, the value of the NFEN0 register changes to “00H”.

Figure 13-20 Format of noise filter enable register 0 (NFEN0)

Address: 40040470H

After reset: 00H

R/W

Symbol

7

6

5

4

3

2

1

0

NFEN0

0	0	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN20	RxD2 pin noise filter is used or not
0	Noise filter OFF
1	Noise filter ON
When used as the RxD2 pin, SNFEN20 must be set to “1”.	
When used as a function other than the RxD2 pin, the SNFEN20 must be set to “0”.	

SNFEN10	RxD1 pin noise filter is used or not
0	Noise filter OFF
1	Noise filter ON
When used as the RxD1 pin, SNFEN10 must be set to “1”.	
When used as a function other than the RxD1 pin, the SNFEN10 must be set to “0”.	

SNFEN00	RxD0 pin noise filter is used or not
0	Noise filter OFF
1	Noise filter ON
When used as the RxD0 pin, SNFEN00 must be set to “1”.	
When used as a function other than the RxD0 pin, the SNFEN00 must be set to “0”.	

Caution Bits 7~5, 3, and 1 must be set to “0”.

13.3.16 Registers Controlling Port Functions of Serial Input/Output Pins

When using a universal serial communication unit, the control registers for the alternated 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 Pin Functions.

When using the alternated port of the serial data output pin or the serial clock output pin as the serial data output or serial clock output, the bits of the corresponding port mode control register (PMCxx) and the bits of the port mode register (PMxx) corresponding to each port are “0”. In this case, the bits of the port register (Pxx) can be “0” or “1”.

In addition, when used for N-channel open-drain output mode, the bit of the port output mode register (POMxx) corresponding to each port must be “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 bits of the Port Mode Register (PMxx) corresponding to each port to “1” and set the bits of the Port Mode Control Register (PMCxx) to “0”. In this case, the Port Register (Pxx) bits can be “0” or “1”.

13.4 Run Stop Mode

Each serial interface of the universal serial communication unit has a stop-and-run mode. Serial communication is not possible in run-stop mode, so power consumption can be reduced. In addition, pins for the serial interface can be used as port functions in run-stop mode.

13.4.1 Stopping the Operation by Units

The peripheral enable register 0 (PER0) is to set stop operation in units.

The PER0 register is a register that sets the clock to be allowed or disallowed to be supplied to each peripheral hardware. Reduce power consumption and noise by providing a clock to hardware that is not in use.

To stop universal serial communication unit 0, the bit2 (SCI0EN) must be set to “0”. To stop universal serial communication unit 1, the bit3 (SCI1EN) must be set to “0”.

Figure 13-21 Configuration of peripheral enable register 0 (PER0) when stopping operation by unit

(a) Peripheral Enable Register 0 (PER0).... Only the corresponding bit of SCIm to be stopped is set to “0”.

	7	6	5	4	3	2	1	0
PER0	RTCEN ×	IRDAEN ×	ADCEN ×	IICA0EN ×	SCI1EN 0/1	SCI0EN 0/1	TM41EN ×	TM40EN ×

SCIm input clock control
0: Stop the input clock supply
1: Enable the input clock supply

Note 1 When the SCImEN bit is “0”, the write operation of the control register of the universal serial communication unit m is ignored, and the read values are all initial. However, the following registers are excluded:

- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port alternate function configuration register (PxxCFG)
- Port output mode register (POMx)
- Port mode register (PMx)
- Port register (Px)

Remark ×: This is an unused bit in the universal serial communication unit (depends on the settings of other peripheral functions).
0/1: Set “0” or “1” according to the user’s purpose.

13.4.2 Stopping the Operation by Channels

Stopping operation by channels is set by each of the following registers.

Figure 13-22 Setting of each register when stopping the operation by channels

(a) Serial channel stop register m (STm)... This is a register that sets the communication/stop count allowed for each channel.

	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 Note 0/1	STm2 Note 0/1	STm1 0/1	STm0 0/1

1: Clear the SEMn bit “0” and stop the communication operation

※Because the STmn bit is the trigger bit, the STmn bit is immediately cleared when the SEMn bit is “0”.

(b) Serial channel enable status register m (SEM)... This register represents the running or stopped state of data transmission and reception for each channel.

	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 Note 0/1	SEM2 Note 0/1	SEM1 0/1	SEM0 0/1

0: Run stop status

※The SEM register is a read-only status register that stops operation through the STm register. For channels that have stopped running, the value of the CKOm_n bit of the SOM register can be set by software.

(c) Serial output enable register m (SOEm)... This is a register that sets the serial communication output that enables or stops each channel.

	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 Note 0/1	SOEm2 Note 0/1	SOEm1 0/1	SOEm0 0/1

0: Stopping the output by serial communication operation

※ For channels that have stopped the serial output, the value of the SOM_n bit of the SOM register can be set by software.

(d) Serial output register m (SOM)... This is the buffer register for the serial output of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM	0	0	0	0	CKOm3 Note 0/1	CKOm2 Note 0/1	CKOm1 0/1	CKOm0 0/1	0	0	0	0	SOM3 Note 0/1	SOM2 Note 0/1	SOM1 0/1	SOM0 0/1

1: The output value of the serial clock is 1. 1: The output value of serial data is 1.

※When using the corresponding pin for each channel as a port function, the corresponding CKOm_n bit and SOM_n bit must be set to “1”.

Note Limited to universal serial communication unit 0 only.

Remarks

- Unit number (m=0, 1) n: channel number (n=0~3).
- : Cannot be set (set initial value). 0/1: Set “0” or “1” according to the user’s purpose.

13.5 3-Wire Serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21) Communication Operation

This is a clock synchronization communication function implemented by a total of 3 lines of serial clock (SCLK) and serial data (SDI and SDO).

[Transmit and receive data]

- 7-bit or 8-bit data length
- Phase control of data transmission and reception
- MSB/LSB first

[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 ^{Note}

Master communication: $\text{Max.f}_{\text{CLK}}/2$ (SSPI00 only).

Master communication: $\text{Max.f}_{\text{CLK}}/4$

Slave communication: $\text{Max.f}_{\text{MCK}}/6$

[Interrupt function]

- Transfer end interrupt, buffer empty interrupt

[Error detection flag]

- Overflow error

Note It must be used within the scope of the SCLK Cycle Time (t_{KCY}) characteristics. Please refer to the data sheet for details.

Channels 0 to 3 of SCI0 and channels 0 to 1 of SCI1 support 3-wire serial I/O SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21) channels.

3-wire serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21) has the following 6 types of communication operation:

- Master transmission (see 13.5.1)
- Master reception (see 13.5.2)
- Master transmission and reception (refer to 13.5.3)
- Slave transmission (see 13.5.4)
- Slave reception (see 13.5.5)
- Slave transmission and reception (see 13.5.6)

13.5.1 Master Transmission

Master transmission refers to the operation of this product outputs the transfer clock and sends data to other devices.

3-wire serial I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
Target channel	Channel 0 of SCI0	Channel 1 of SCI0	Channel 2 of SCI0	Channel 3 of SCI0	Channel 0 of SCI1	Channel 1 of SCI1
Used pin	SCLK00, SDO00	SCLK01, SDO01	SCLK10, SDO10	SCLK11, SDO11	SCLK20, SDO20	SCLK21, SDO21
Interrupt	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	Can select transfer end interrupt (single transfer mode) or buffer empty interrupt (continuous transfer mode).					
Error detection flag	None					
Transfer data length	7 or 8 bits					
Transfer rate Note	Max. $f_{CLK}/2$ [Hz] (SSPI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK} : System clock frequency					
Data phase	It can be selected via the DAPmn bit of the SCRmn register. • DAPmn=0: The data output starts 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 via the CKPmn bit of the SCRmn register. • CKPmn=0: Non-reverse • CKPmn=1: Reverse					
Data direction	MSB first or LSB first					

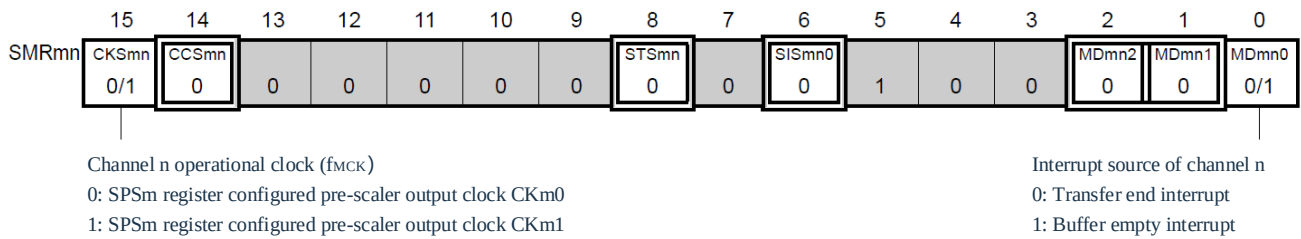
Note It must be used within the scope of the peripheral functional characteristics that meet this condition and meet the electrical characteristics (refer to the data sheet).

Remark m: unit number (m=0, 1) n: channel number (n=0~3) mn=00~03, 10~11.

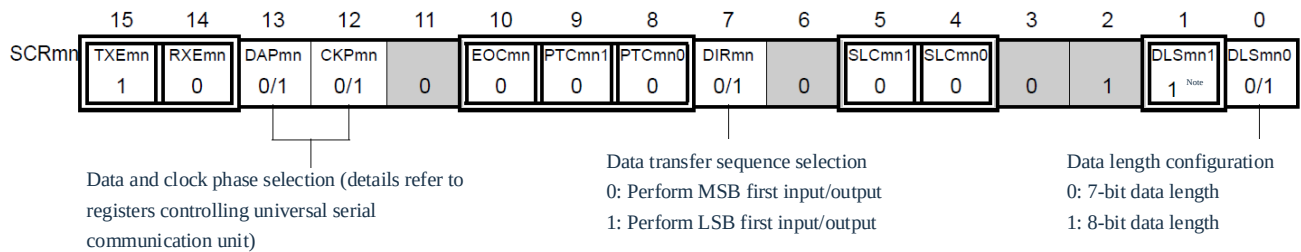
(1) Register setting

Figure 13-23 3 wire serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21)
Example of register settings when the master is transmitted

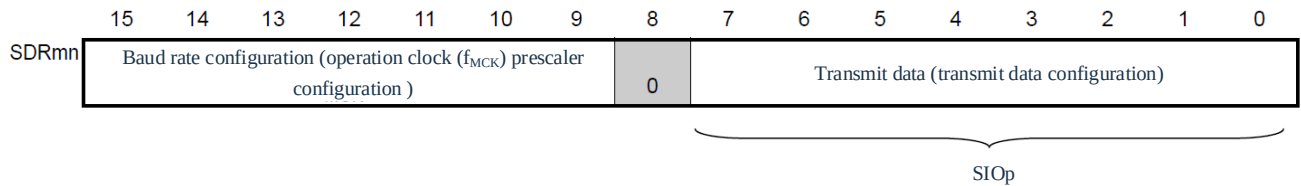
(a) Serial mode register mn (SMRmn)



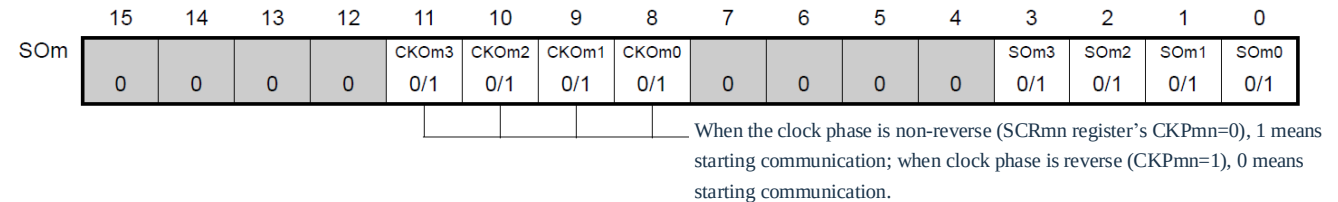
(b) Serial communication operation configuration register mn (SCRmn)



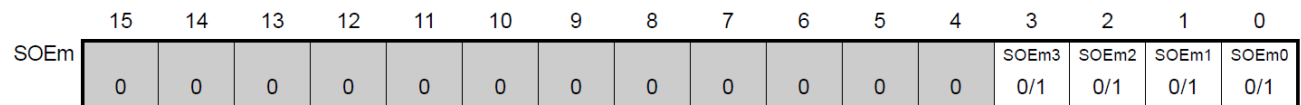
(c) Serial data register mn (SDRmn)



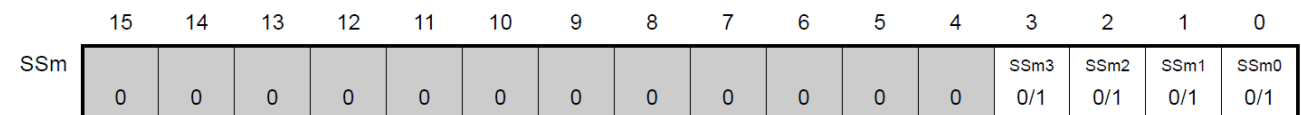
(d) Serial output register m (SOm) only configure the target channel bits



(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.



Note: Limited to SCR00, SCR01 register, and others are fixed to "1".

Remarks

1. m: unit number (m=0, 1) n: channel number (n=0~3) mn=00~03, 10~11.
2. : Cannot be set (set initial value). 0/1: Set "0" or "1" according to the user's purpose.

(2) Operation steps

Figure 13-24 Initial setup steps for master transmission

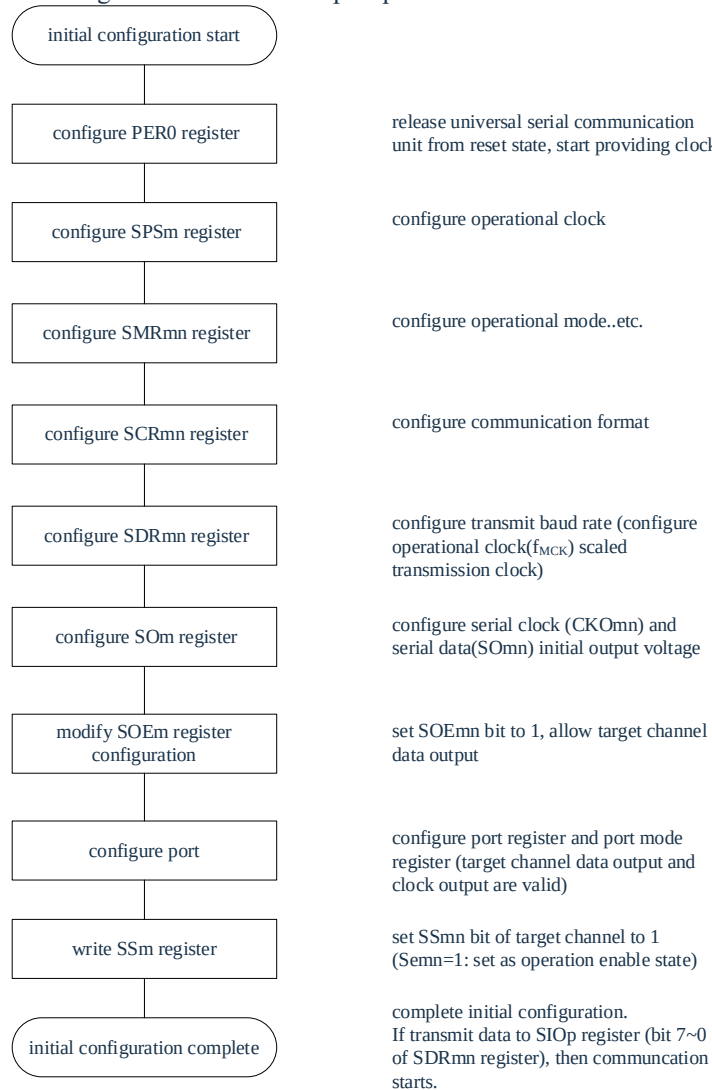


Figure 13-25 Stop steps for master transmission

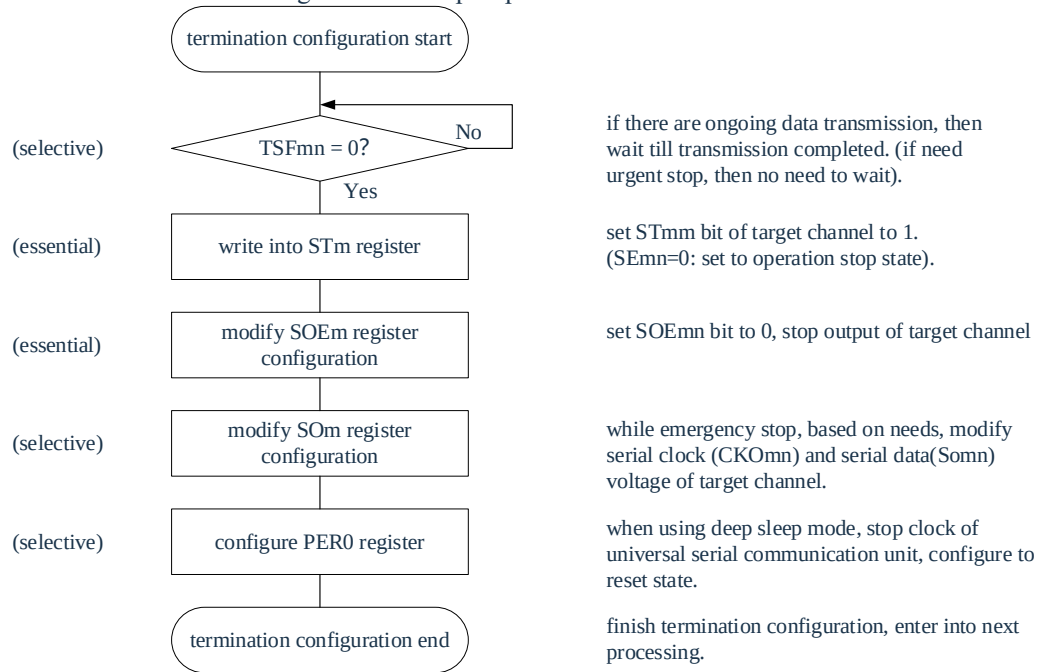
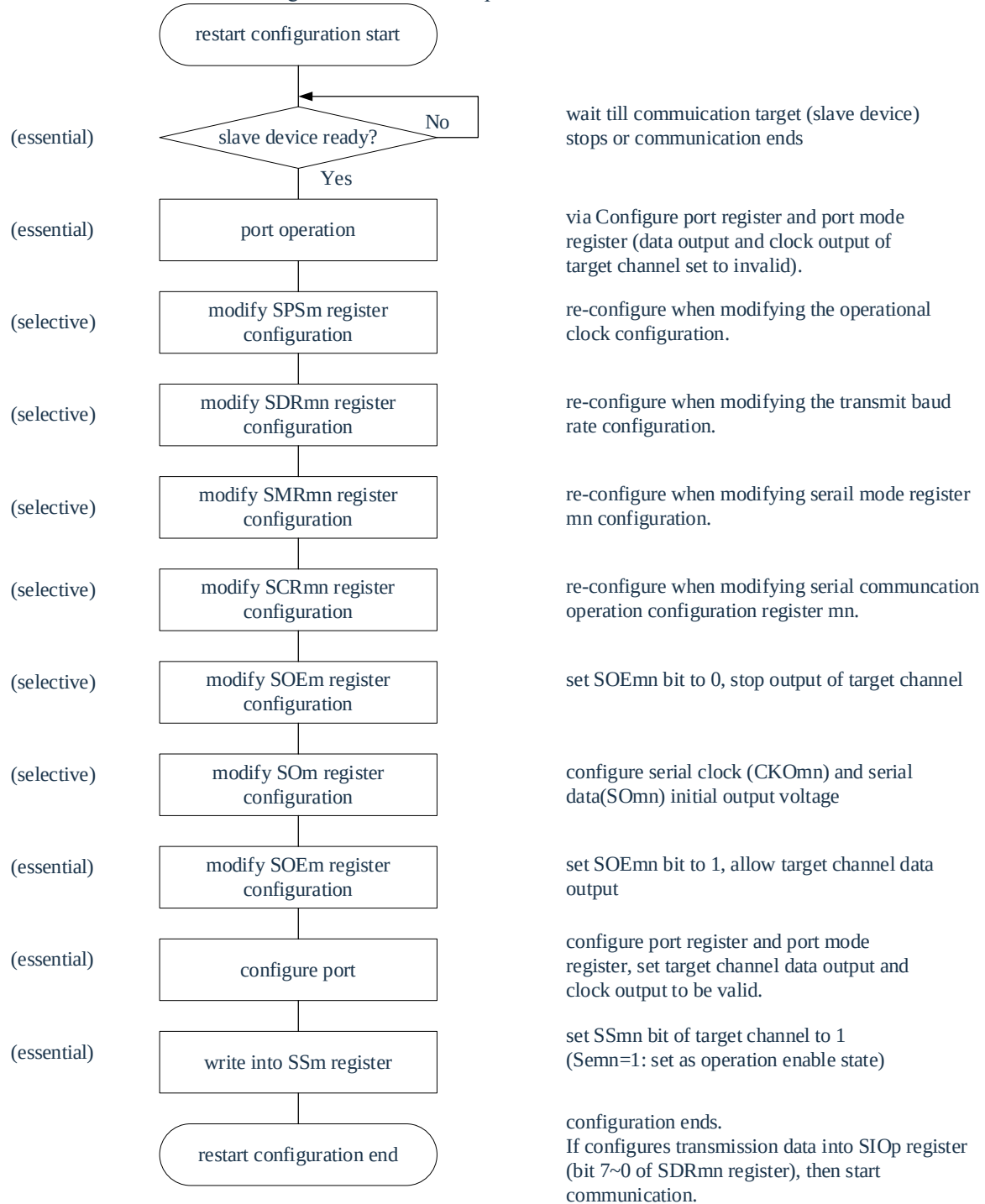


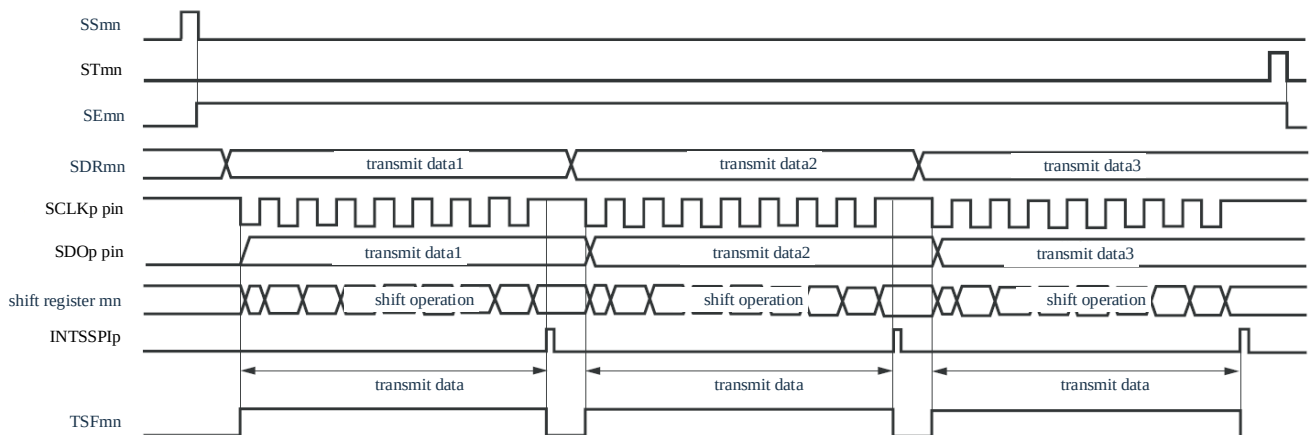
Figure 13-26 Restart steps for master transmission



Remark 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.

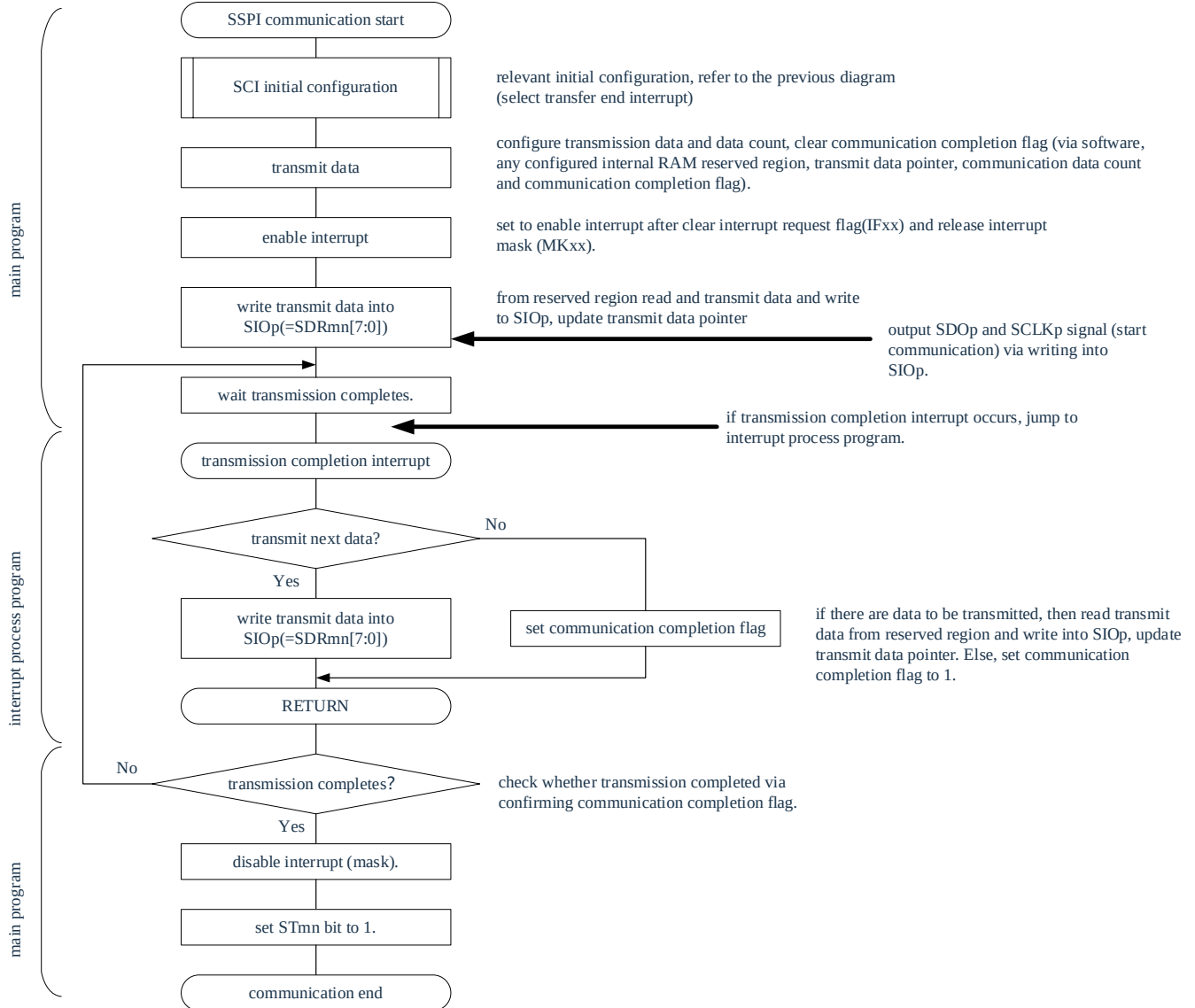
(3) Processing flow (single transmit mode)

Figure 13-27 Timing diagram of master transmission (single transmit mode) (type 1: DAPmn=0, CKPmn=0)



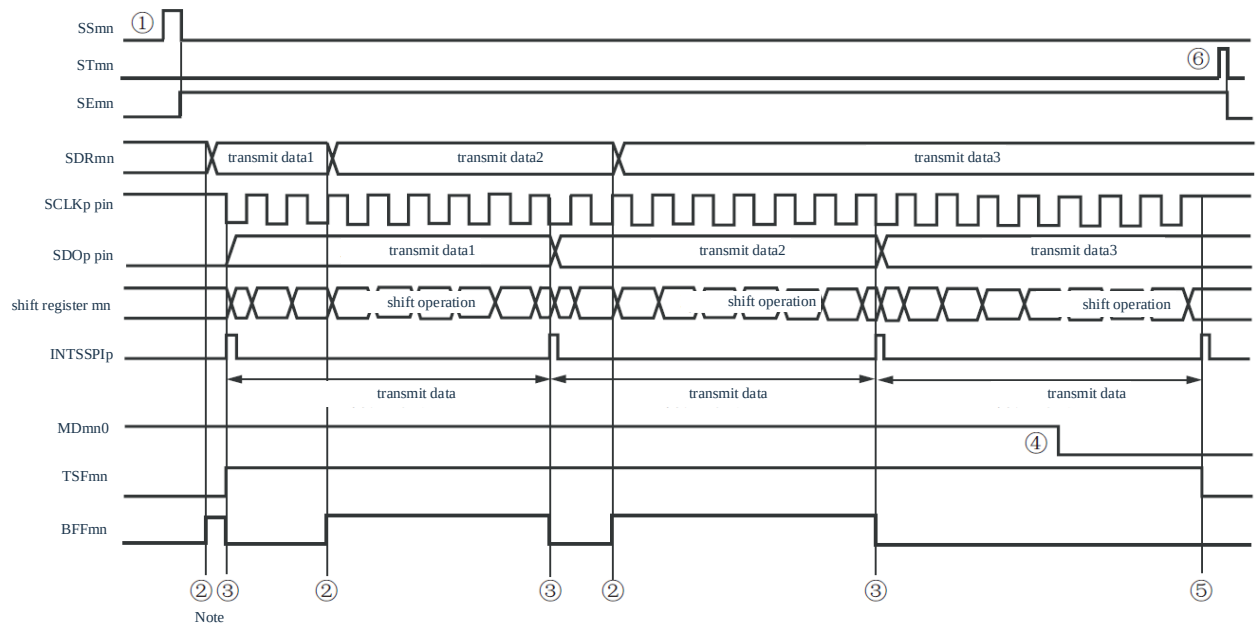
Remark m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)

mn=00~03, 10~11

Figure 13-28 Flow of master transmission (single transmit mode)


(4) Processing flow (continuous transmit mode)

Figure 13-29 Timing diagram of the master transmission (continuous transmit mode)
(type 1: DAPmn=0, CKPmn=0)

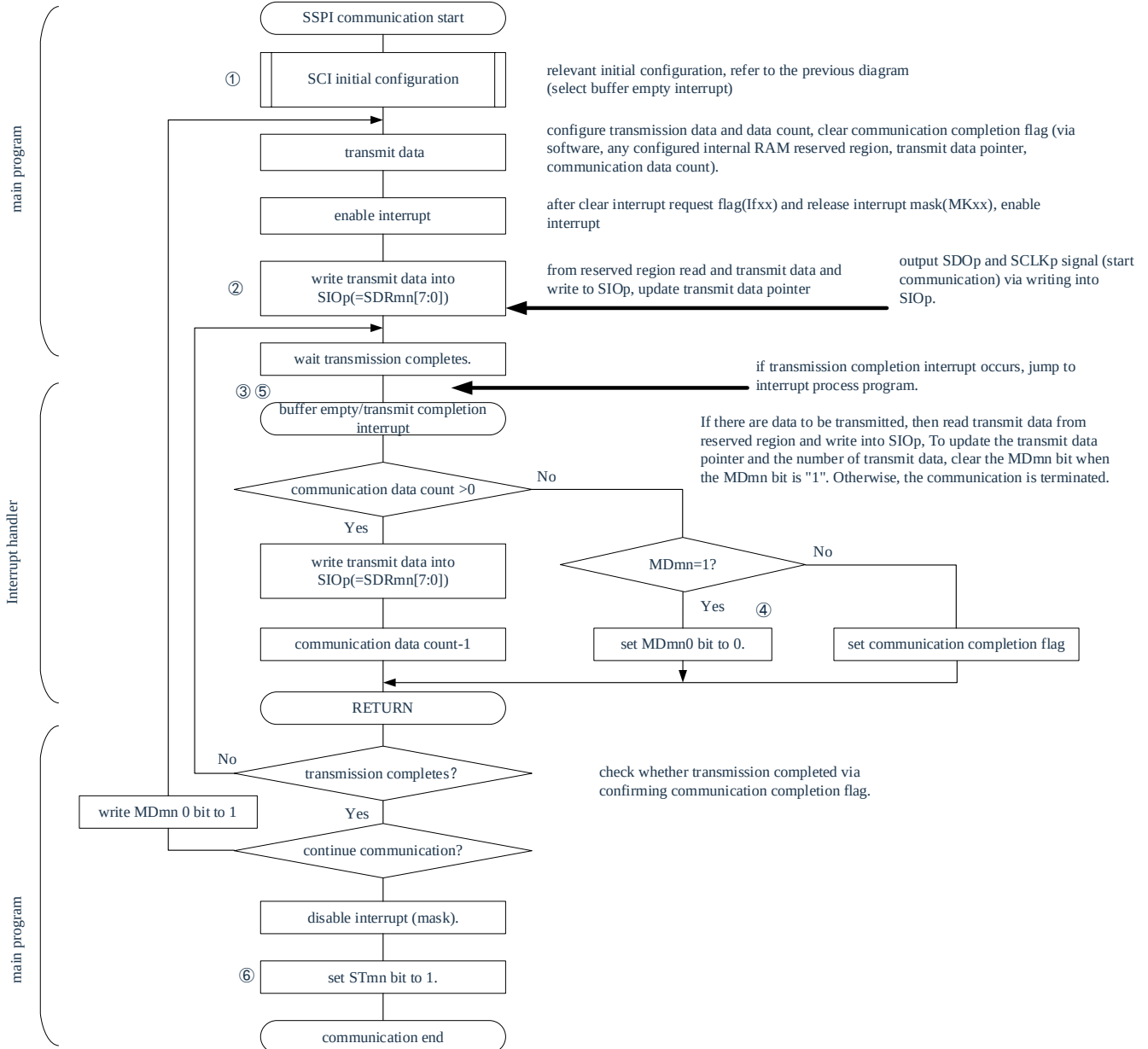


Note If the BFFmn bit of the Serial Status Register mn (SSRmn) is 1 (when valid data is stored in the Serial Data Register mn (SDRmn)), writing data to the SDRmn register will overwrite the transmitted data.

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, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)
mn=00~03, 10~11

Figure 13-30 Flowchart of master transmission (continuous transmit mode)



13.5.2 Master Reception

Master reception refers to the operation of this product outputs transfer clock and receives data from other devices.

3-wire serial I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
Target channel	Channel 0 of SCI0	Channel 1 of SCI0	Channel 2 of SCI0	Channel 3 of SCI0	Channel 0 of SCI1	Channel 1 of SCI1
Used pin	SCLK00, SDO00	SCLK01, SDO01	SCLK10, SDO10	SCLK11, SDO11	SCLK20, SDO20	SCLK21, SDO21
Interrupt	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
Error detection flag	Can select transfer complete interrupt (single transfer mode) or buffer empty interrupt (continuous transfer mode).					
Transfer data length	Only the overflow error detection flag (OVFmn)					
Transfer rate ^{Note}	7 or 8 bits					
Data phase	Max. $f_{CLK}/2$ [Hz](SSPI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK} : System clock frequency					
Clock phase	It can be selected via the DAPmn bit of the SCRmn register. • DAPmn=0: The data output starts 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 first or LSB first					

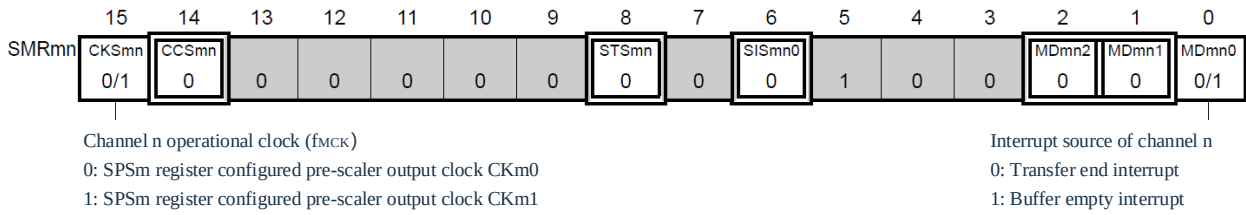
Note It must be used within the scope of the peripheral functional characteristics that meet this condition and meet the electrical characteristics (refer to the data sheet).

Remark m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21) mn=00~03, 10~11.

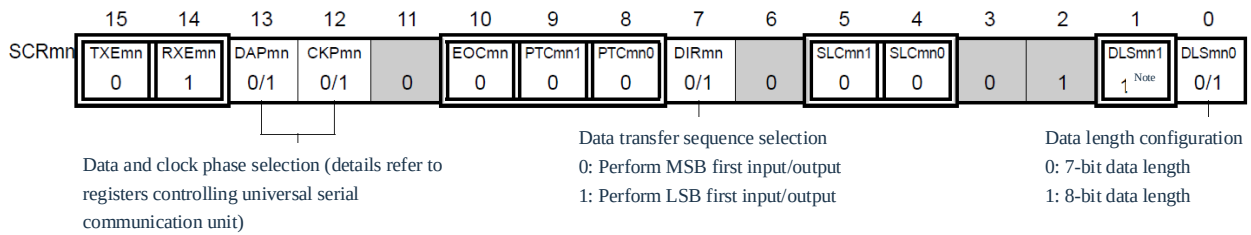
(1) Register setting

Figure 13-31 3-wire serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21)
Example of register setting content when the master receives

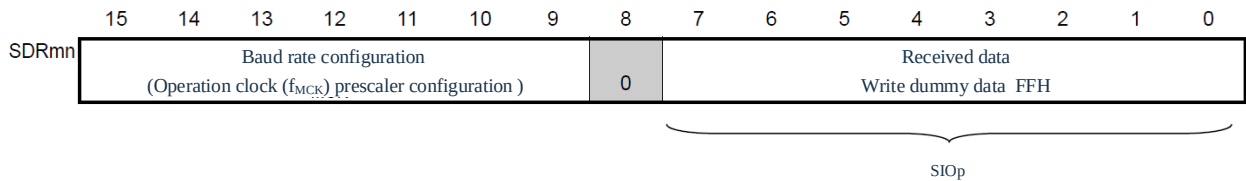
(a) Serial mode register mn (SMRmn)



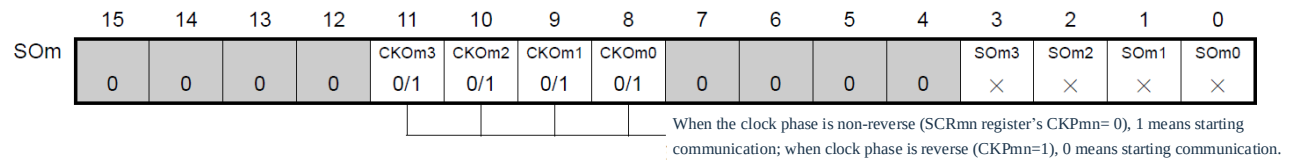
(b) Serial communication operation configuration register mn (SCRmn)



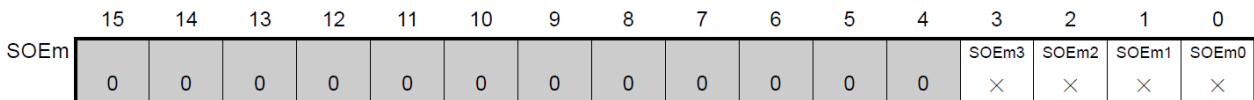
(c) Serial data register mn (SDRmn) (low 8 bits: SIOp)



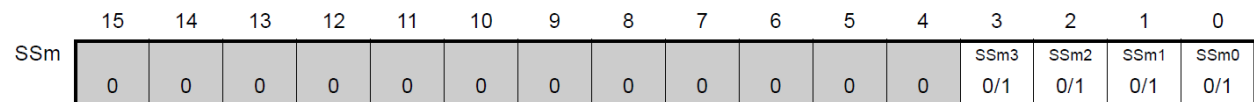
(d) Serial output register m (SOM).... only configure the target channel bits



(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.



Note Limited to SCR00 register and SCR01 register, others are fixed as "1".

Remark1.m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)

mn=00~03, 10~11

2. : 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) Operation steps

Figure 13-32 Initial setup steps for master reception

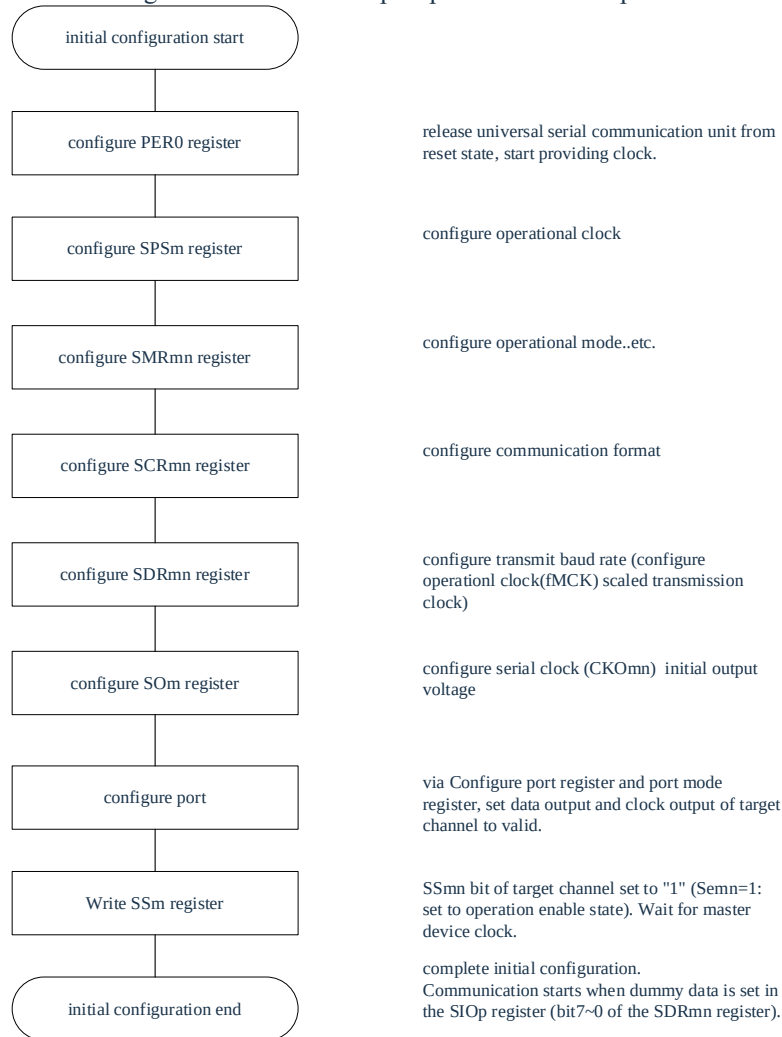


Figure 13-33 Stop steps for master reception

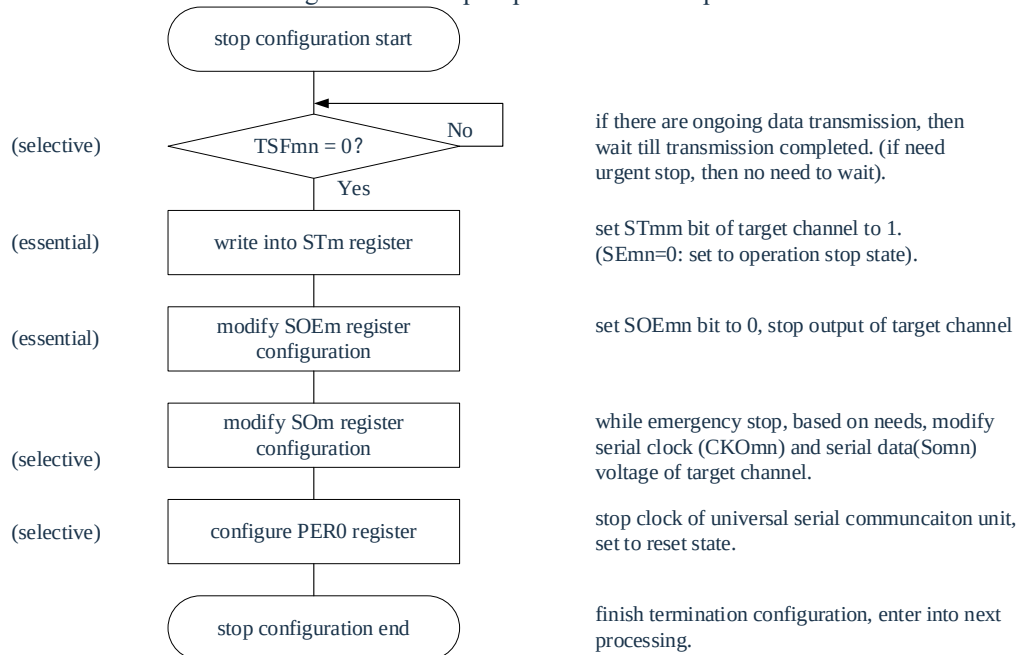
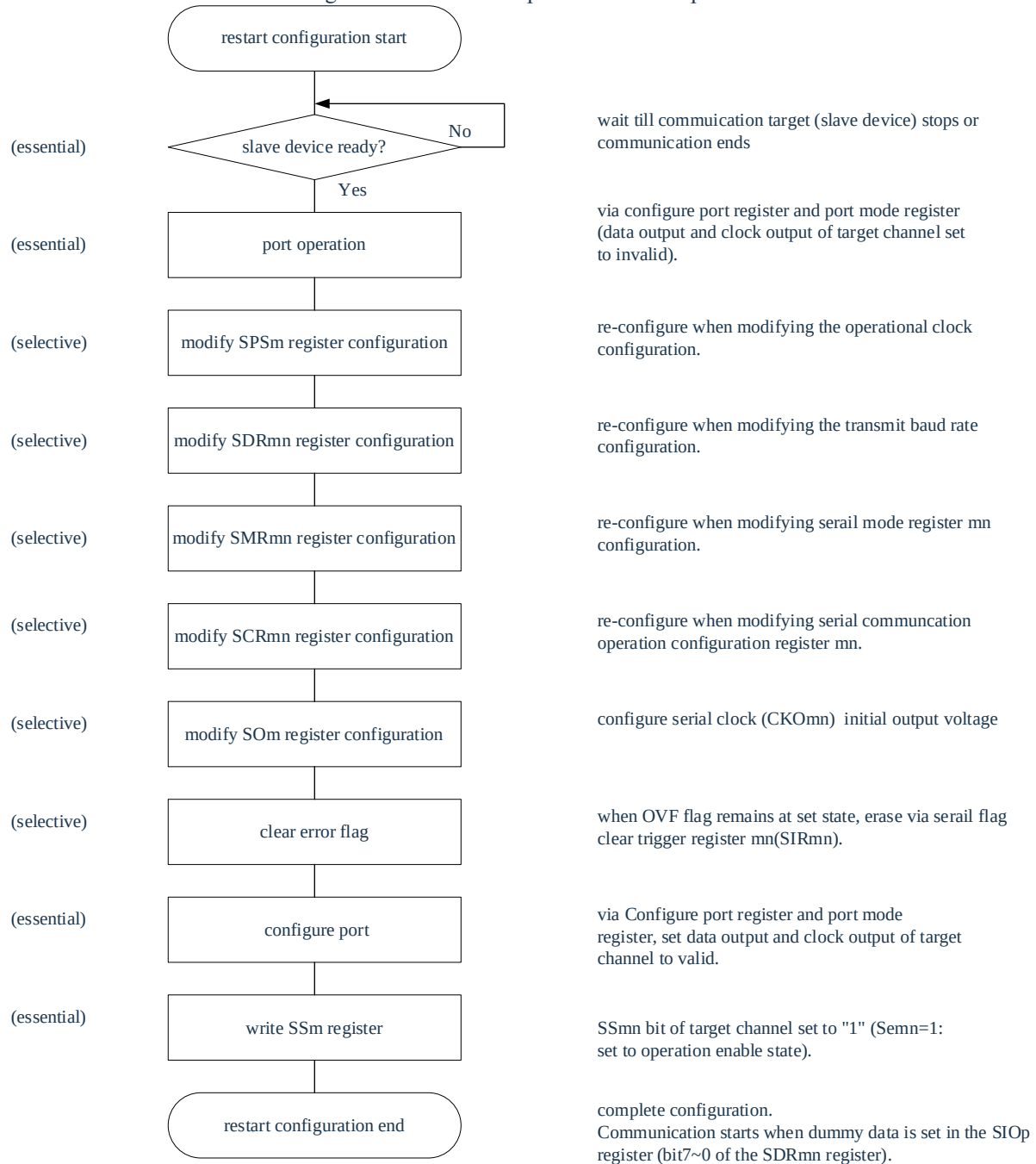


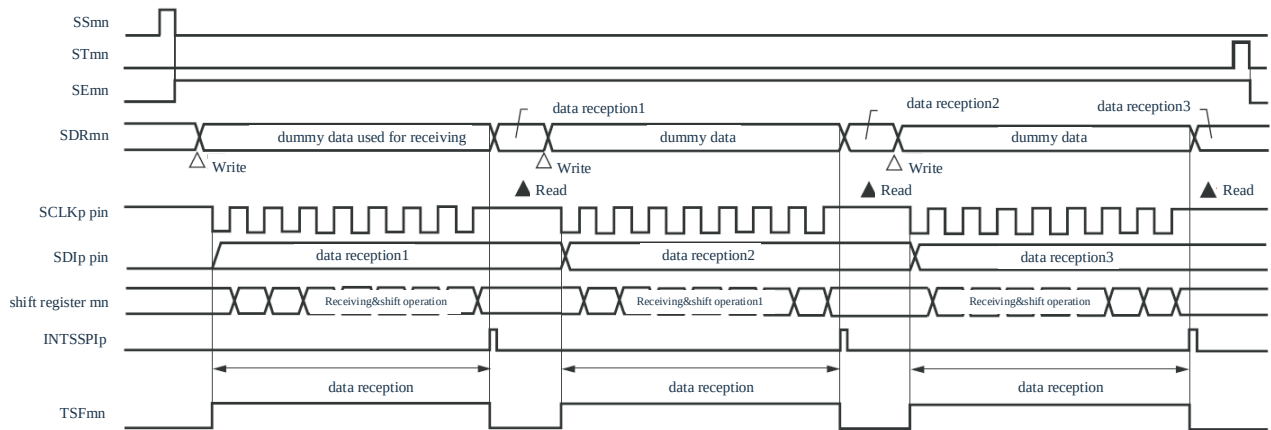
Figure 13-34 Restart steps for master reception



Remark 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)

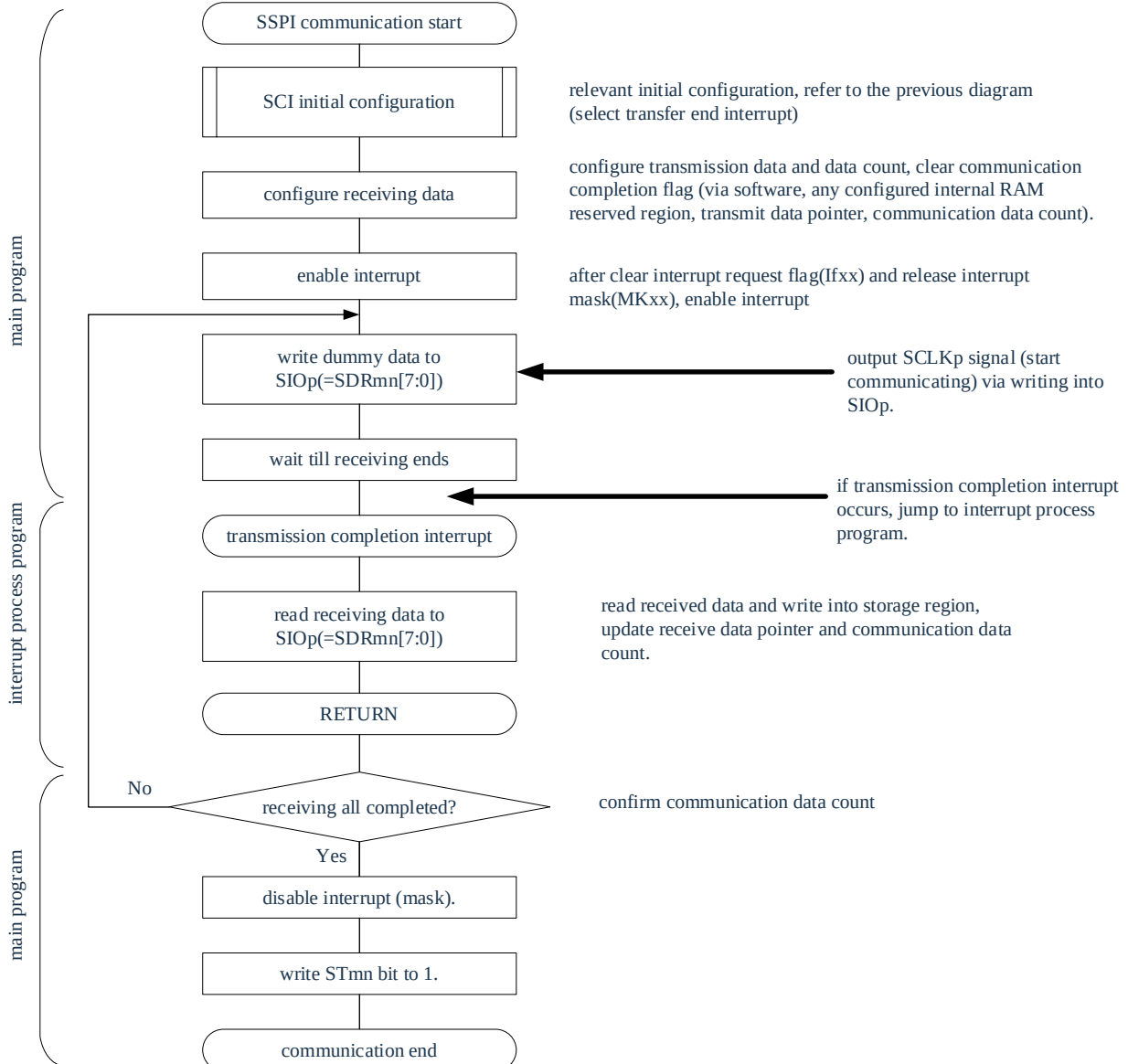
Figure 13-35 Timing diagram of master reception (single receive mode) (type 1: DAPmn=0, CKPmn=0)



Remark m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)

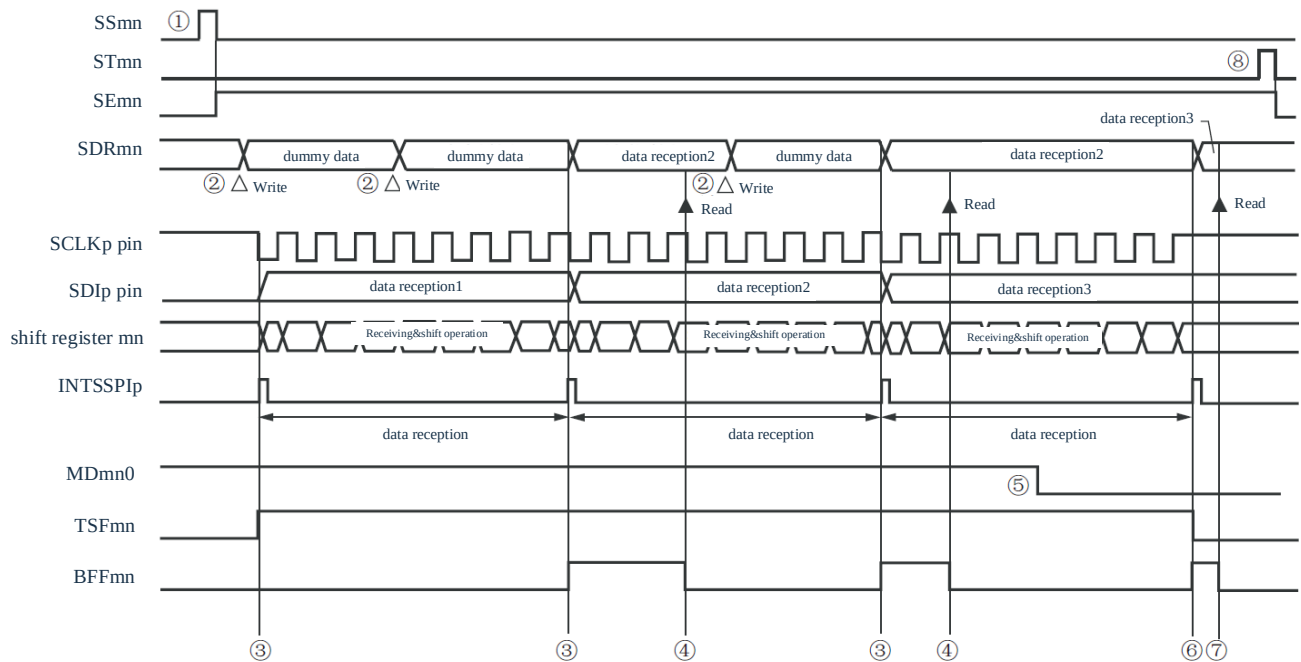
mn=00~03, 10~11

Figure 13-36 Flowchart of master reception (single receive mode)



(4) Processing flow (continuous receive mode)

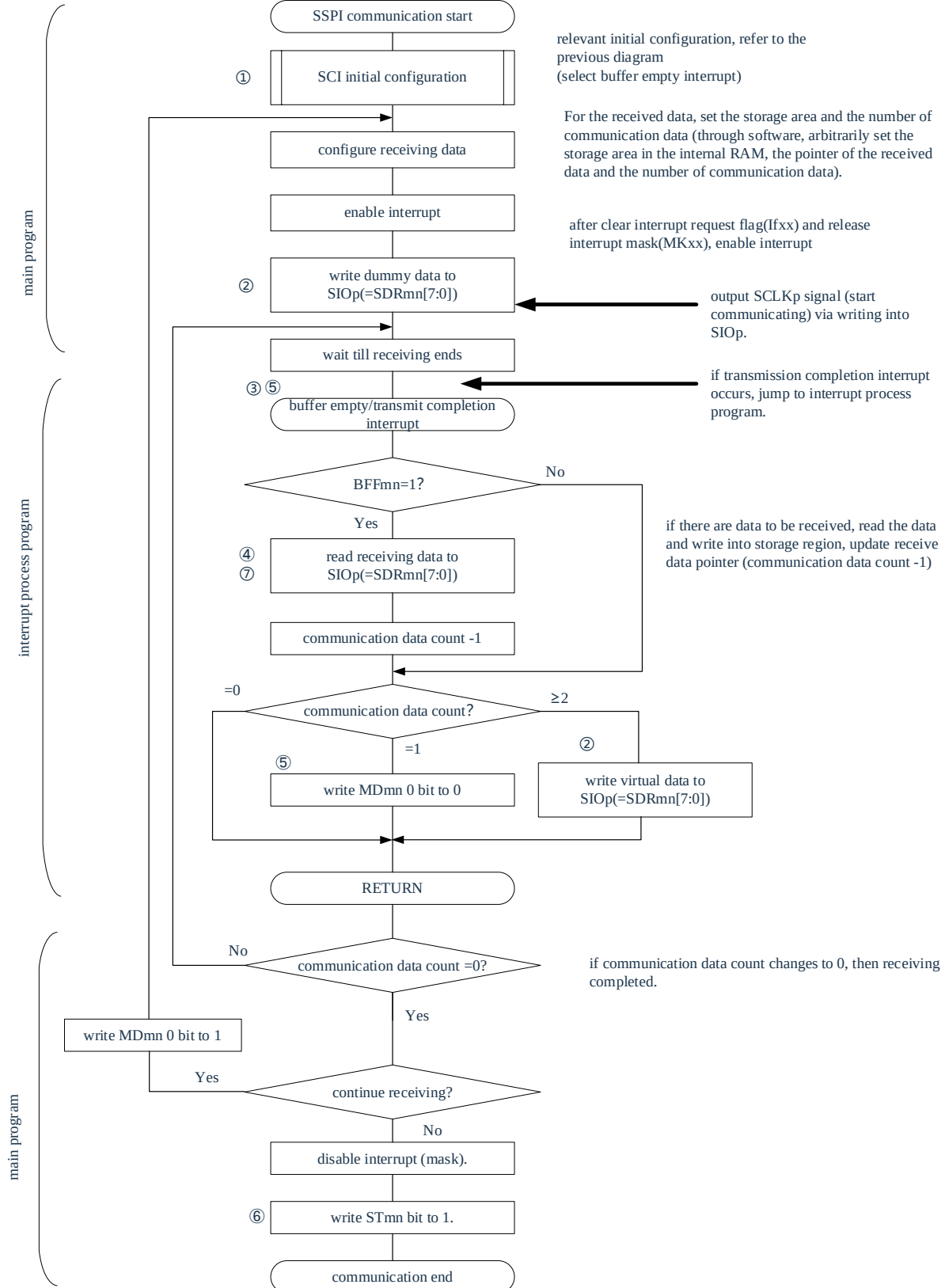
Figure 13-37 Timing diagram of master reception (continuous receive mode)
(type 1: DAPmn=0, CKPmn=0).



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. ① to ⑧ in the figure correspond to the ① to ⑧ in “Figure 13-38 Flowchart of master reception (continuous receive mode)”.

2. m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)
mn=00~03, 10~11

Figure 13-38 Flowchart of master reception (continuous receive mode)


Remark ① to ⑧ in the diagram correspond to ① to ⑧ in “Figure 13-37 Timing diagram of master reception (continuous receive mode)”.

13.5.3 Master Transmission and Reception

Master transmission and reception refers to the operation where this product outputs the transfer clock and performs data transmission and reception with other devices.

3-wire serial I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
Target channel	Channel 0 of SCI0	Channel 1 of SCI0	Channel 2 of SCI0	Channel 3 of SCI0	Channel 0 of SCI1	Channel 1 of SCI1
Used pin	SCLK00, SDI00, SDO00	SCLK01, SDI01, SDO01	SCLK10, SDI10, SDO10	SCLK11, SDI11, SDO11	SCLK20, SDI20, SDO20	SCLK21, SDI21, SDO21
Interrupt	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	Can select transfer end interrupt (single transfer mode) or buffer empty interrupt (continuous transfer mode).					
Error detection flag	Only the overflow error detection flag (OVFmn)					
Transfer data length	7 or 8 bits					
Transfer rate ^{Note}	Max. $f_{CLK}/2$ [Hz](limited to SSPI00), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK} : System clock frequency					
Data phase	It can be selected via the DAPmn bit of the SCRmn register. • DAPmn=0: The data output starts 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 first or LSB first					

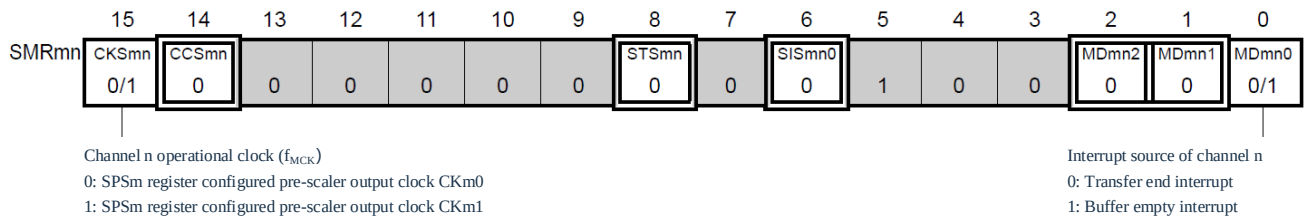
Note It must be used within the scope of the peripheral functional characteristics that meet this condition and meet the electrical characteristics (refer to the data sheet).

Remark m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)
mn=00~03, 10~11

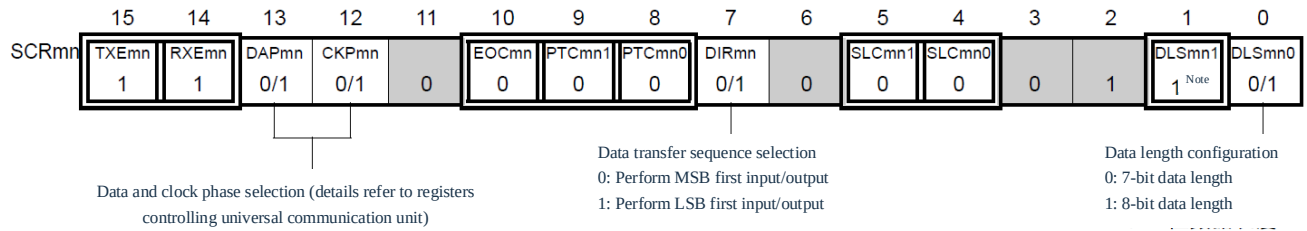
(1) Register setting

Figure 13-39 3-wire serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21)
Example of register settings during master transmission and reception

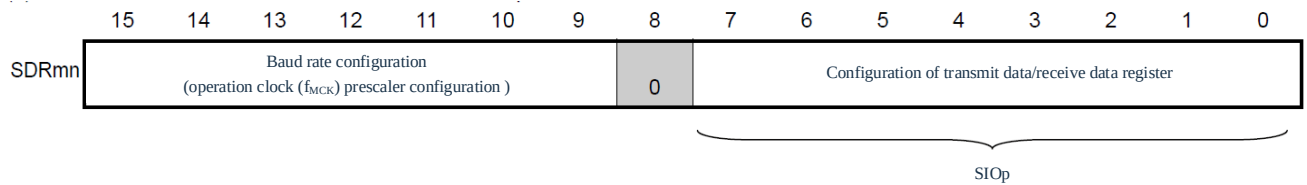
(a) Serial mode register mn (SMRmn)



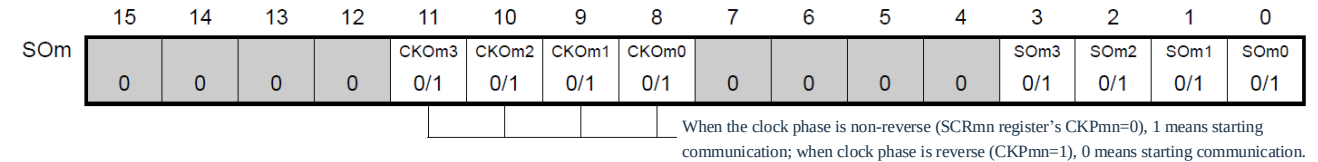
(b) Serial communication operation configuration register mn (SCRmn)



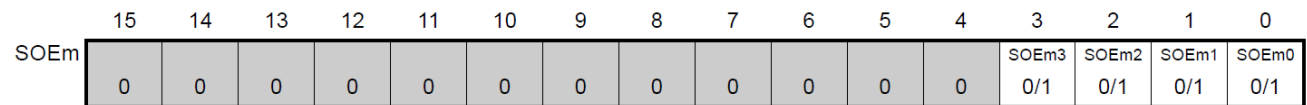
(c) Serial data register mn (SDRmn) (low 8 bits: SIOp)



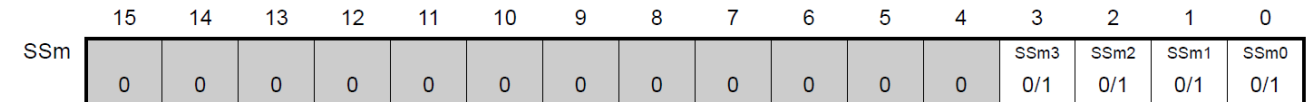
(d) Serial output register m (SOM) only configure target channel bits



(e) Serial output enable register m (SOEm).....only set the set bit of the target channel to 1.



(f) Serial channel start register m (SSm) only set the set bit of the target channel to 1.



Note Limited to SCR00 register and SCR01 register, the others are fixed as "1".

Caution 1. m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21) mn=00~03, 10~11

2. : Fixed in SSPI master T&R modes. : Cannot be set (initial value).

0/1: Set "0" or "1" according to the user's purpose.

(2) Operation steps

Figure 13-40 Initial setup steps for master transmission and reception



Figure 13-41 Stop steps for master transmission and reception

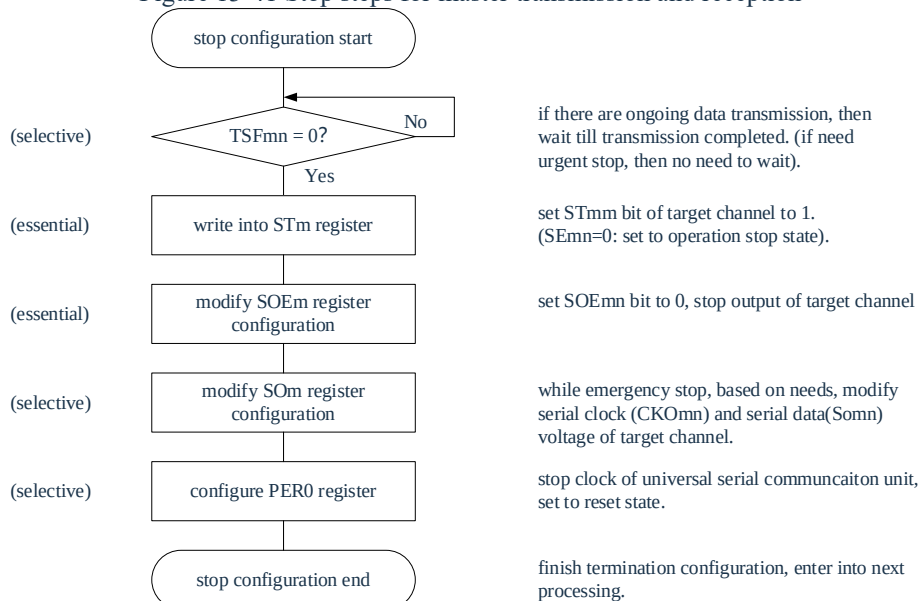
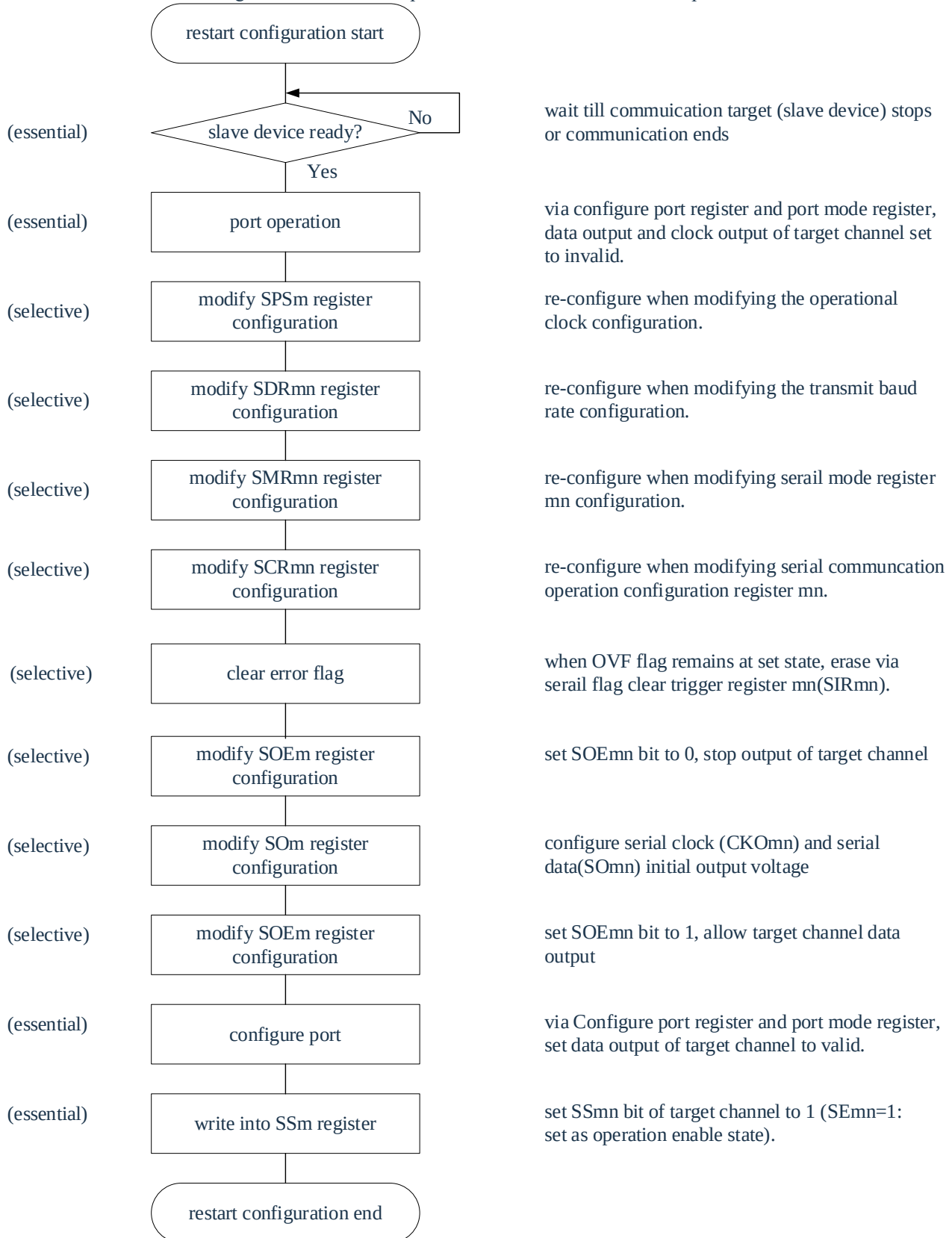
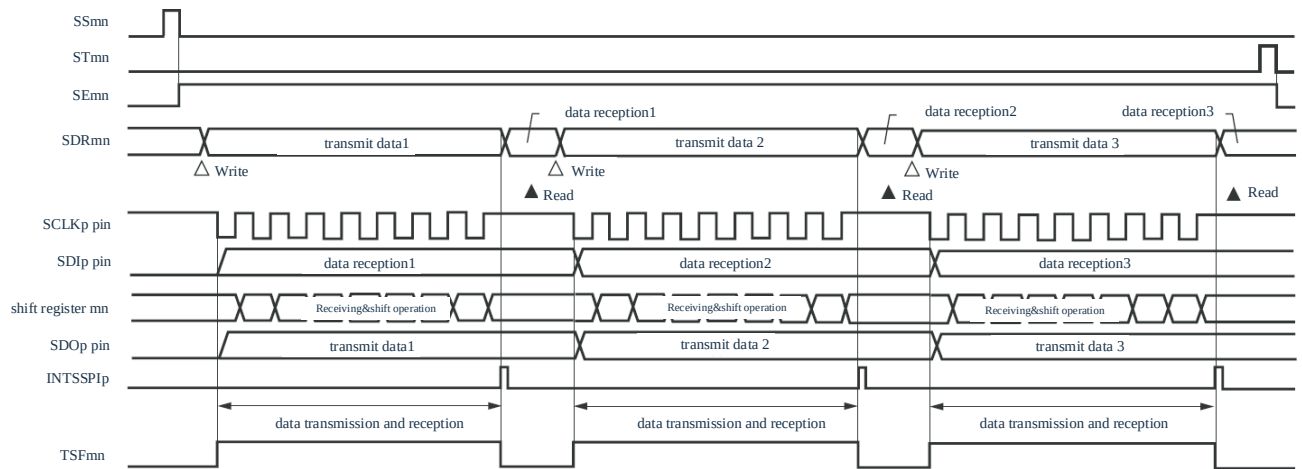


Figure 13-42 Restart steps for master transmission and reception



(3) Processing flow (single transmit and receive mode)

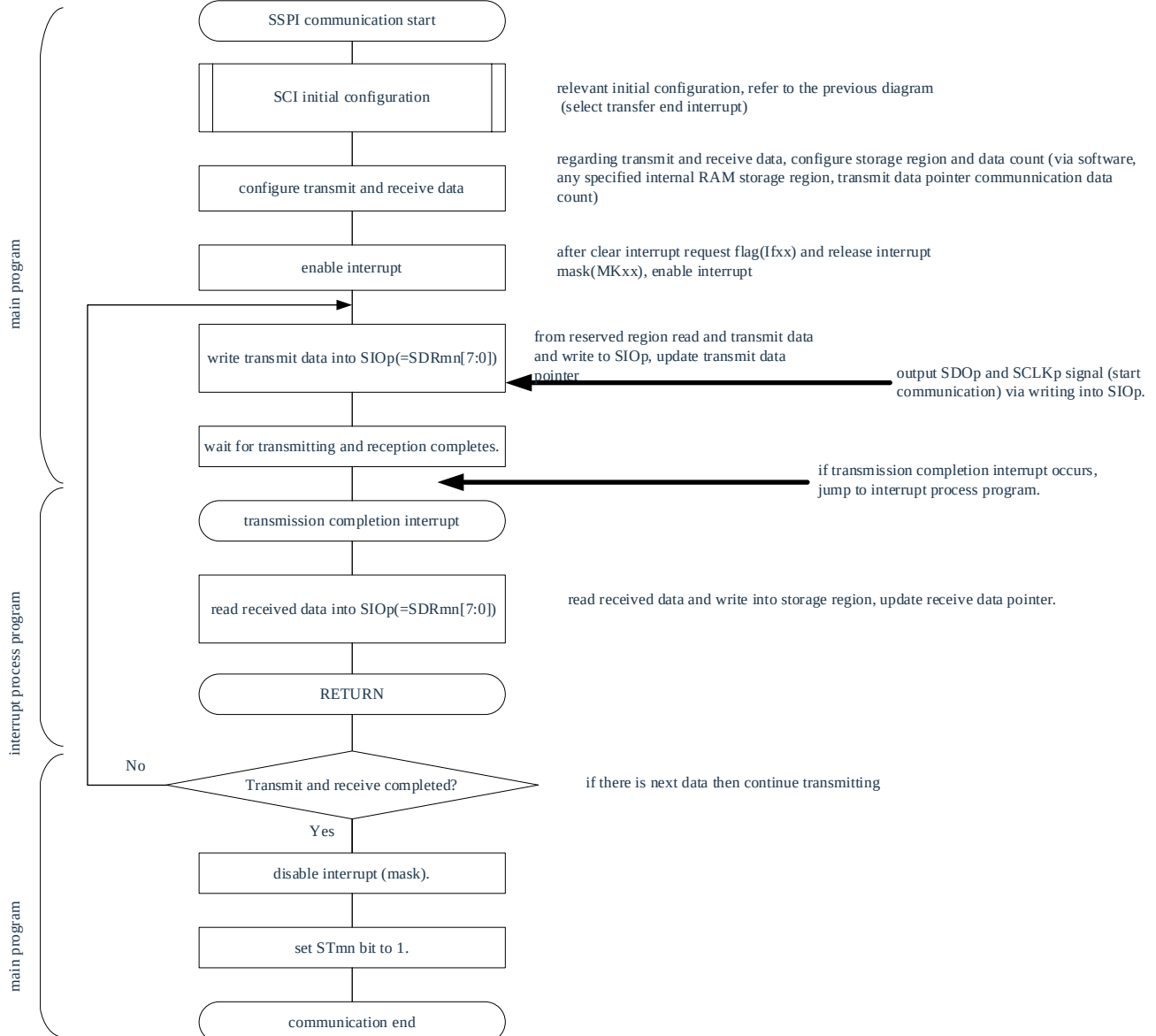
Figure 13-43 Timing diagram of master transmission and reception (single transmit and receive mode)
(Type 1: DAPmn=0, CKPmn=0)



Remark m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)

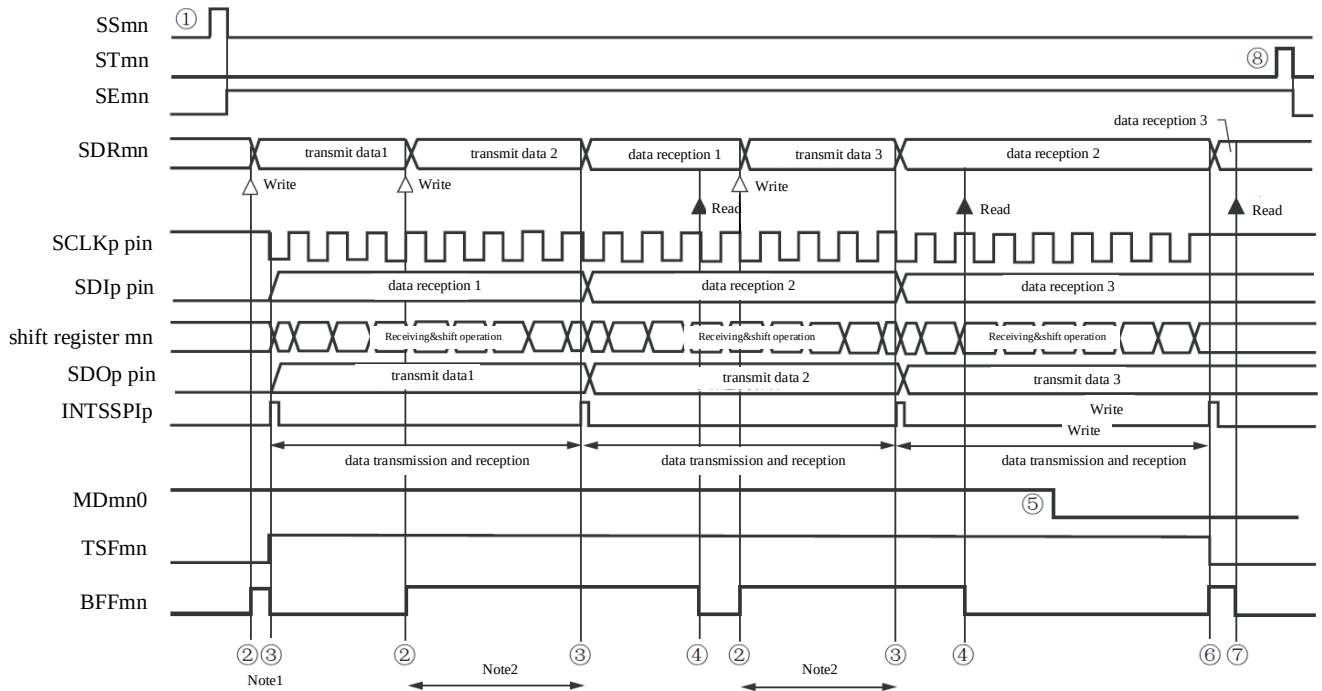
mn=00~03, 10~11.

Figure 13-44 Flowchart of master transmission and reception (single transmit and receive mode)



(4) Processing flow (continuous transmit and receive mode)

Figure 13-45 Timing diagram of master transmission and reception (continuous transmit and receive mode)
(type 1: DAPmn=0, CKPmn=0)



Notes

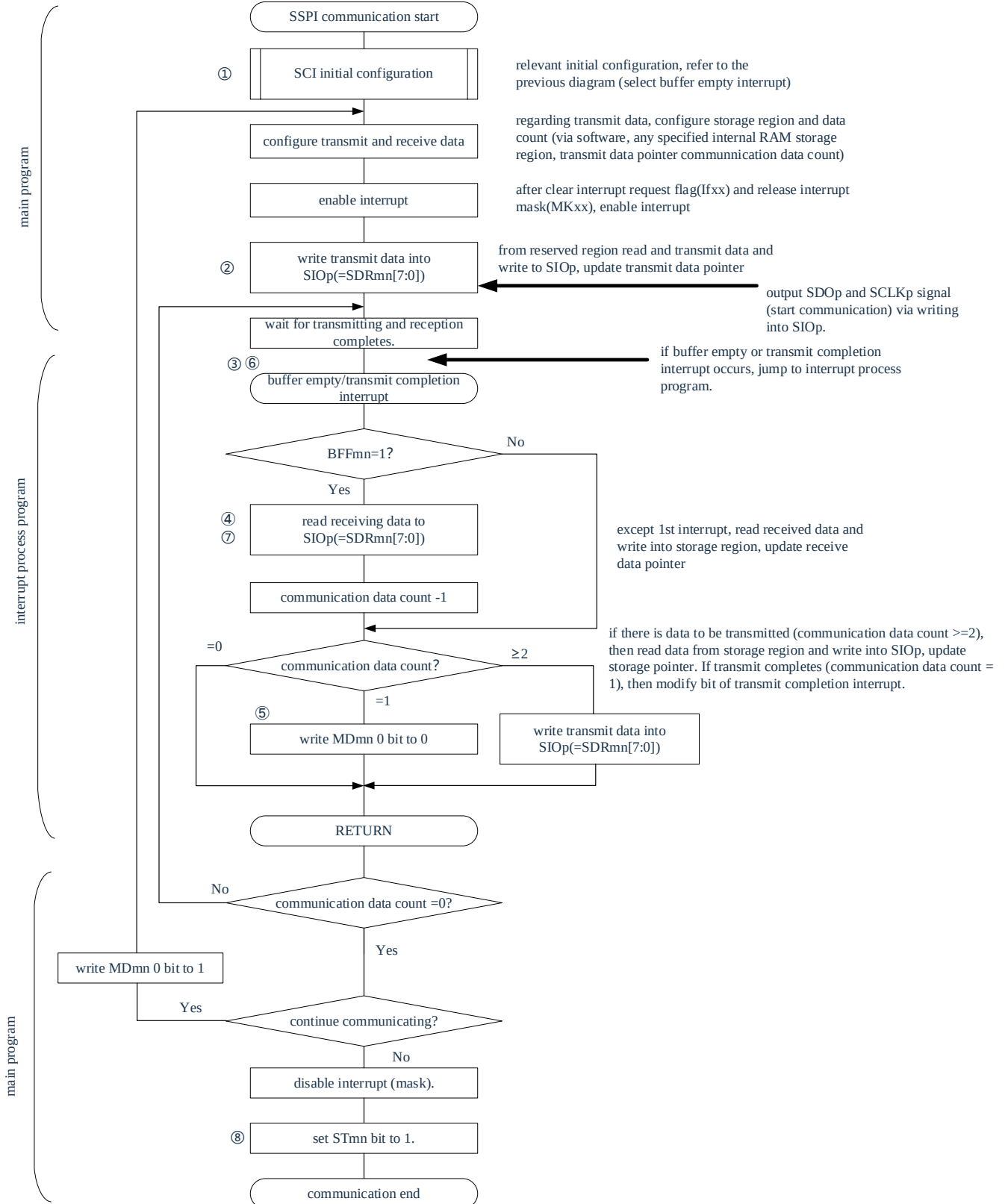
1. If the BFFmn bit of the serial status register mn (SSRmn) is "1" (valid data is saved in the serial data register mn (SDRmn) to write the send data to the SDRmn register, and rewrite the sent data.
2. If the SDRmn register is read during this period, the transmitted data can be read. At this point, 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.

Remarks 1. ① to ⑧ in the figure correspond to ① to ⑧ in the "Flowchart of Figure 13-46)".

2.m: unit number (m=0, 1) n: Channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)
mn=00~03, 10~11

Figure 13-46 Flowchart of master transmission and reception (continuous transmit and receive mode)



Remark ① to ⑧ correspond to ① to ⑧ in “Figure 13-45 Timing diagram of master transmission and reception (continuous transmit and receive mode)”.

13.5.4 Slave Transmission

Slave transmission is the operation of the BAT32G133B microcontroller to transmit data to other devices in the state of inputting the transfer clock from other devices.

3-wire serial I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
Target channel	Channel 0 of SCI0	Channel 1 of SCI0	Channel 2 of SCI0	Channel 3 of SCI0	Channel 0 of SCI1	Channel 1 of SCI1
Used pin	SCLK00, SDO00	SCLK01, SDO01	SCLK10, SDO10	SCLK11, SDO11	SCLK20, SDO20	SCLK21, SDO21
Interrupt	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
Error detection flag	Can select transfer end interrupt (single transfer mode) or buffer empty interrupt (continuous transfer mode).					
Transfer data length	Only the overflow error detection flag (OVFmn)					
Transfer rate	7 or 8 bits					
Data phase	Max. $f_{MCK}/6$ [Hz] ^{Note1, 2}					
Clock phase	It can be selected via the DAPmn bit of the SCRmn register. • DAPmn=0: The data output starts when the serial clock starts running. • DAPmn=1: Start data output half a clock before the serial clock starts running.					
Data direction	It can be selected via the CKPmn bit of the SCRmn register. • CKPmn=0: Non-reverse • CKPmn=1: Reverse					
	MSB first or LSB first					

Notes 1. The maximum transfer rate is $f_{MCK}/6$ [Hz] because the external serial clocks input from pins SCLK00, SCLK01, SCLK10, SCLK11, SCLK20, and SCLK21 are sampled internally and then used.

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).

Remarks 1. f_{MCK} : operation clock frequency of the target channel

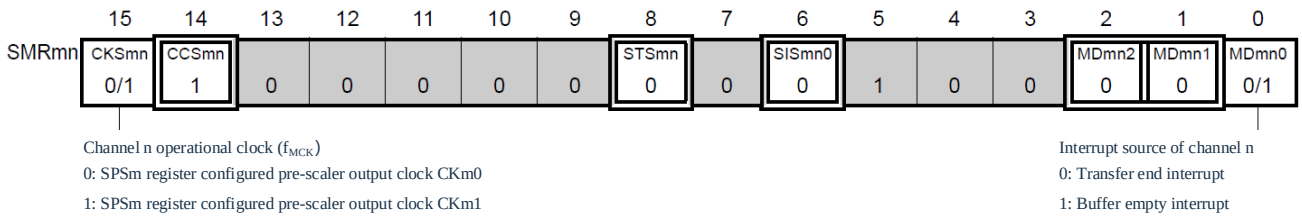
2. m: unit number (m=0, 1) n: channel number (n=0~3) mn=00~03, 10~11.

(1) Register setting

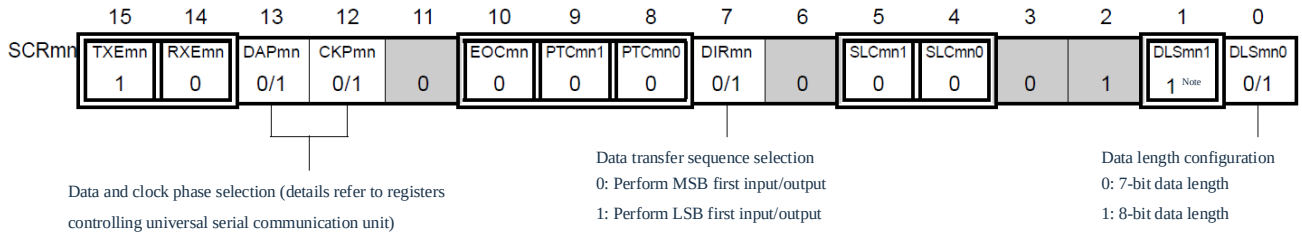
Figure 13-47 3-wire serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21)

Example of register settings for slave transmission

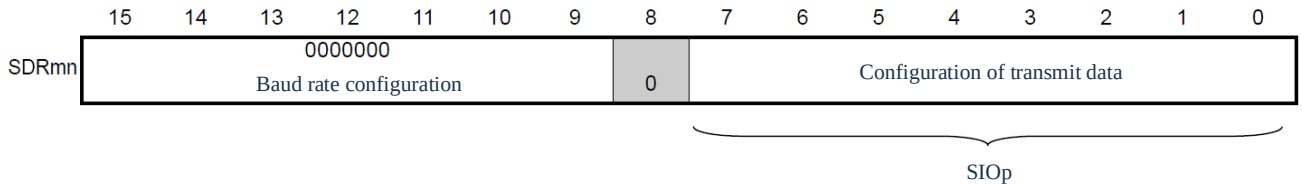
(a) Serial mode register mn (SMRmn)



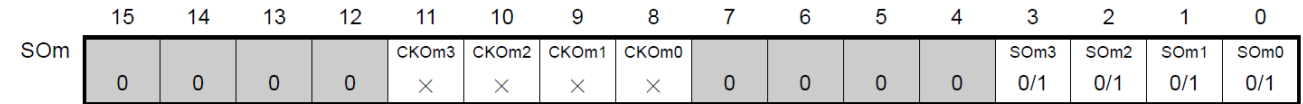
(b) Serial communication operation configuration register mn (SCRmn)



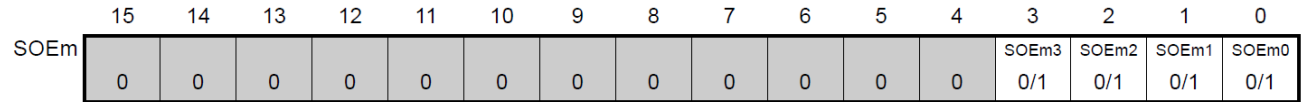
(c) Serial data register mn (SDRmn) (low 8 bits: SIOp)



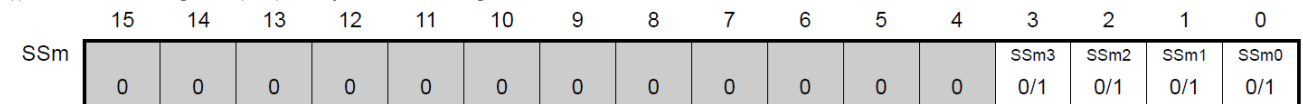
(d) Serial output register m (SOM) Only configure target channel bits



(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.



Note Limited to SCR00 register and SCR01 register, the others are fixed as “1”.

Remarks 1.m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)

mn=00~03, 10~11

2. : Fixed in SSPI slave send 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) Operation steps

Figure 13-48 Initial setup steps for slave transmission



Figure 13-49 Stop steps for slave transmission

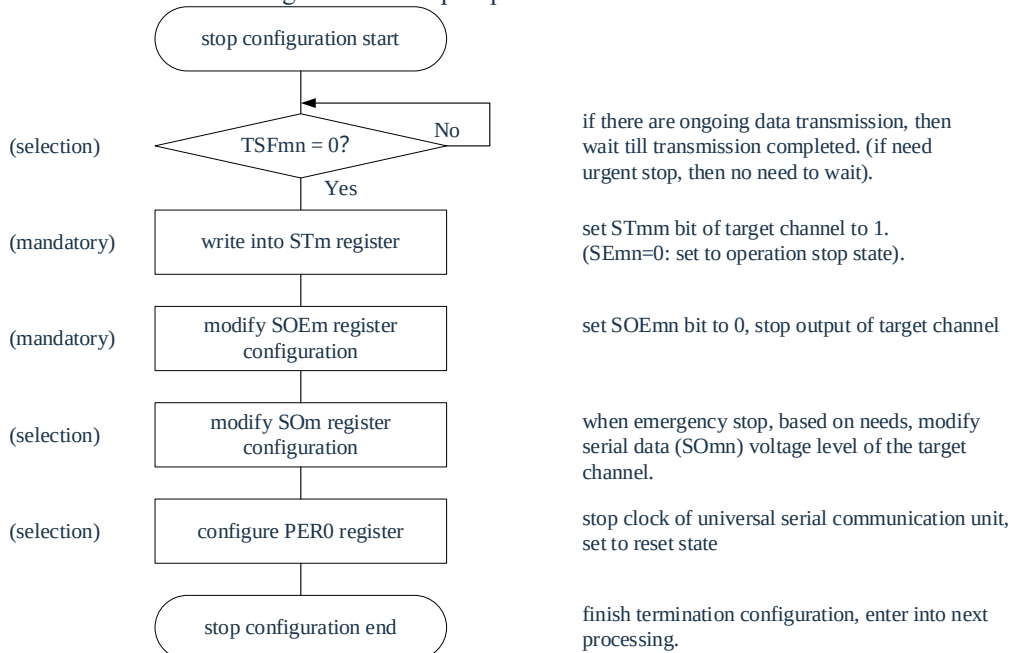
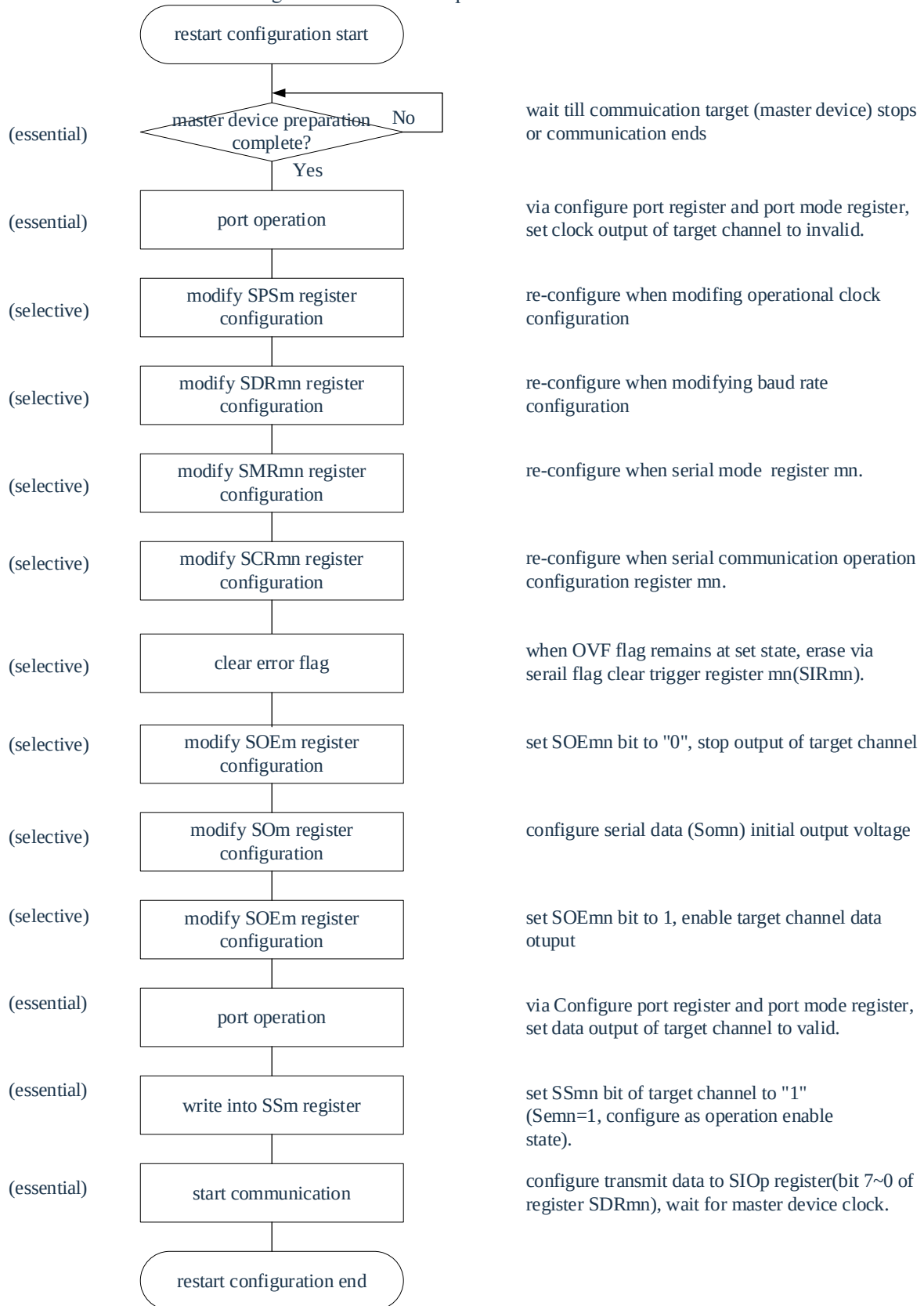


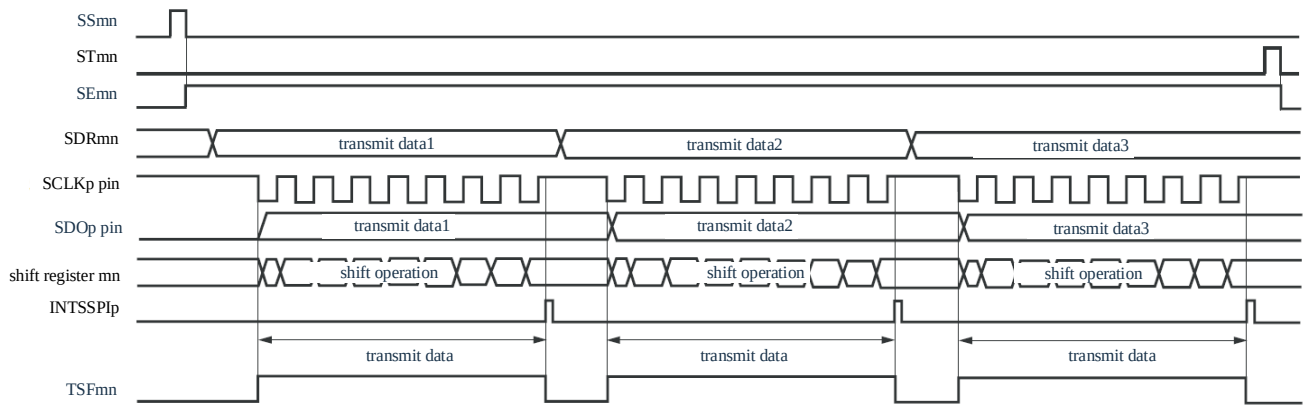
Figure 13-50 Restart steps for slave transmission



Remark 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.

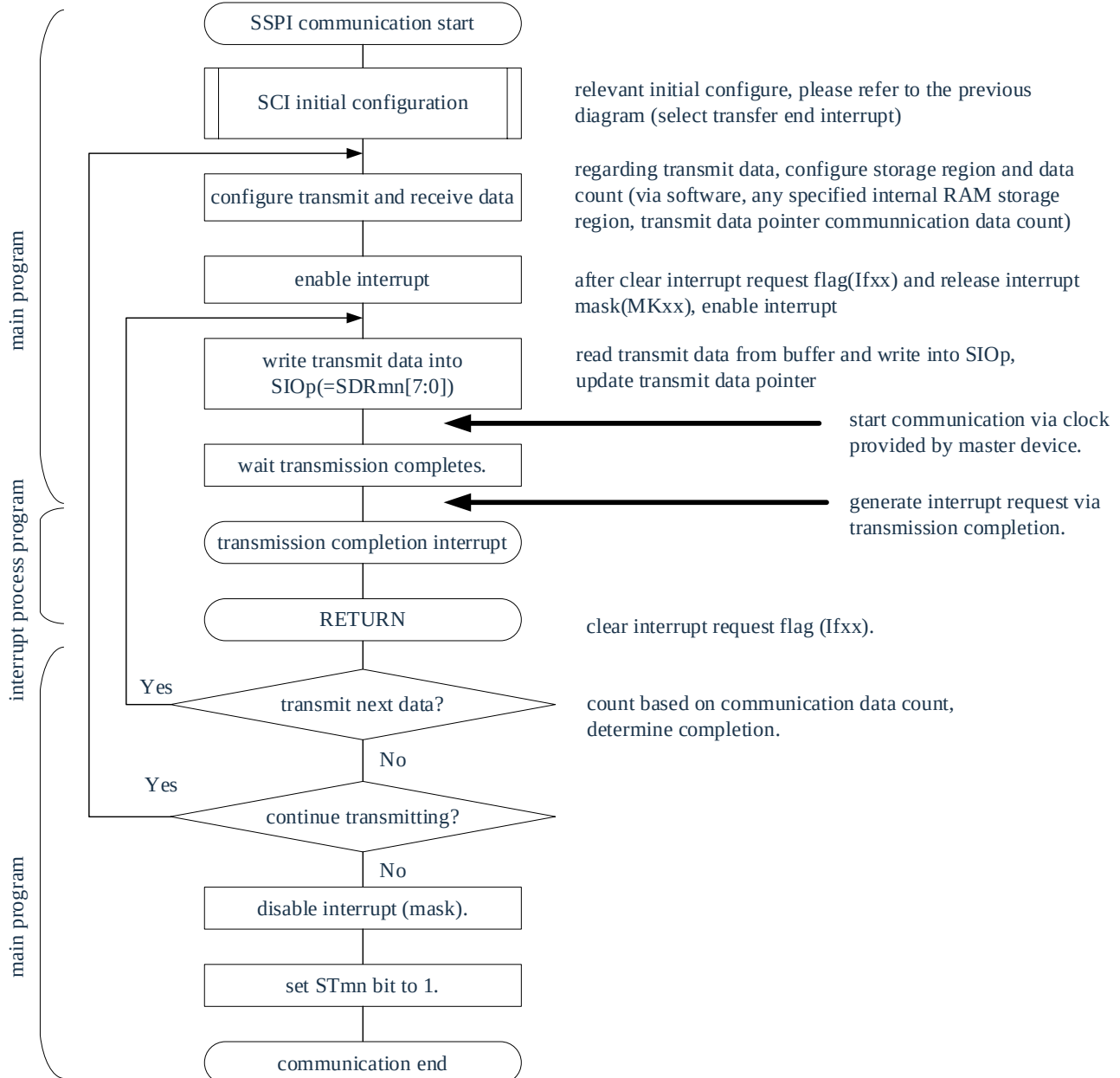
(3) Processing flow (single transmit mode)

Figure 13-51 Timing diagram of slave transmission (single transmit mode) (type 1: DAPmn=0, CKPmn=0)



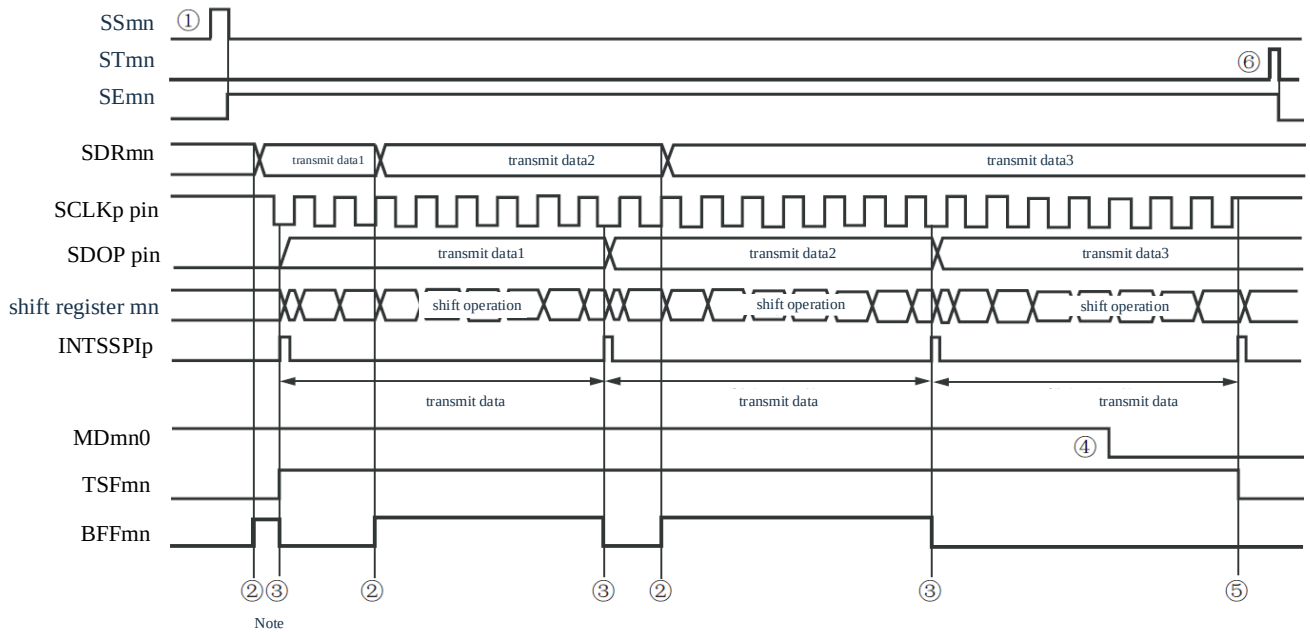
Remark m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)
mn=00~03, 10~11

Figure 13-52 Flowchart of slave transmission (single transmit mode)



(4) Processing flow (continuous transmit mode)

Figure 13-53 Timing diagram of slave transmission (continuous transmit mode) (type 1: DAPmn=0, CKPmn=0)

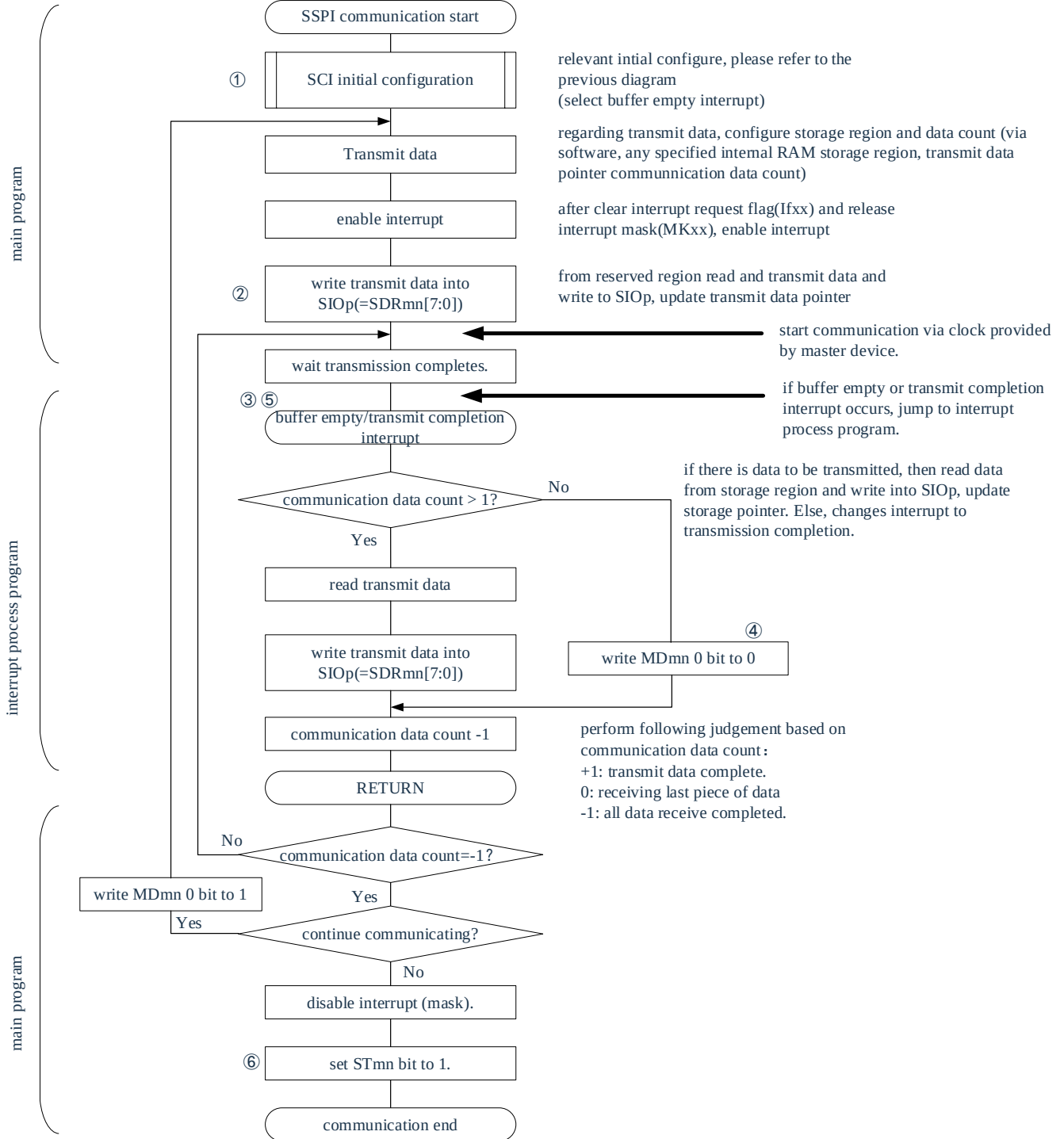


Note If the BFFmn bit of the serial status register mn (SSRmn) is "1" (when valid data is saved in the serial data register mn (SDRmn)) is given The SDRmn register writes the transmitted data and overrides the transmitted data.

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, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)
mn=00~03, 10~11.

Figure 13-54 Flowchart of slave transmission (continuous transmit mode)



Remark ① to ⑥ in the diagram correspond to ① to ⑥ in “Figure 13-53 Timing diagram of slave transmission (continuous transmit mode)”.

13.5.5 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.

3-wire serial I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
Target channel	Channel 0 of SCI0	Channel 1 of SCI0	Channel 2 of SCI0	Channel 3 of SCI0	Channel 0 of SCI1	Channel 1 of SCI1
Used pin	SCLK00, SDI00	SCLK01, SDI01	SCLK10, SDI10	SCLK11, SDI11	SCLK20, SDI20	SCLK21, SDI21
Interrupt	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
Error detection flag	Limited to transfer end interrupts (disable setting buffer empty interrupts).					
Transfer data length	Only the overflow error detection flag (OVFmn).					
Transfer rate	7 or 8 bits					
Transfer rate	Max.f _{MCK} /6[Hz] ^{Note1, 2}					
Data phase	It can be selected via the DAPmn bit of the SCRmn register. • DAPmn=0: The data output starts 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 via the CKPmn bit of the SCRmn register. • CKPmn=0: Non-reverse • CKPmn=1: Reverse					
Data direction	MSB first or LSB first					

Notes 1. The maximum transfer rate is $f_{MCK}/6$ [Hz] because the external serial clocks input from pins SCLK00, SCLK01, SCLK10, SCLK11, SCLK20, and SCLK21 are sampled internally and then used.

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).

Remarks 1. f_{MCK} : Operation clock frequency of the target channel

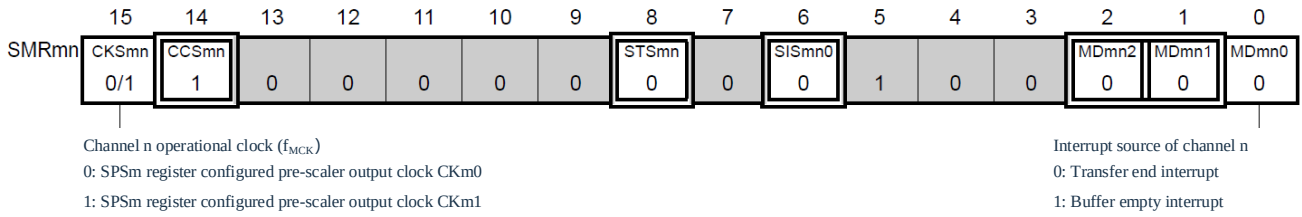
2. m: unit number (m=0, 1) n: channel number (n=0~3) mn=00~03, 10~11.

(1) Register setting

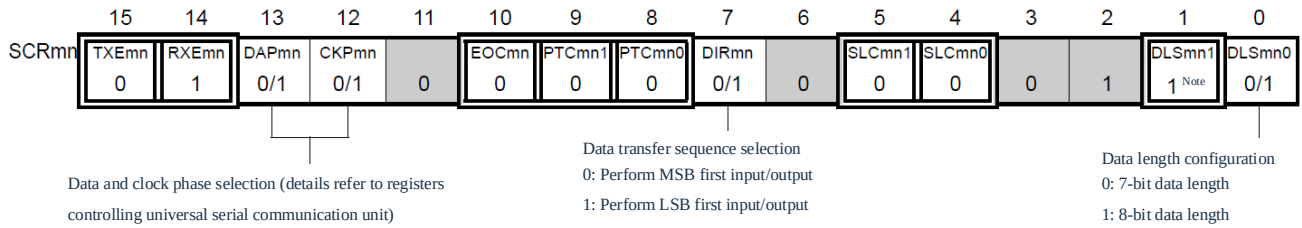
Figure 13-55 3-wire serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21)

Example of register settings for slave reception

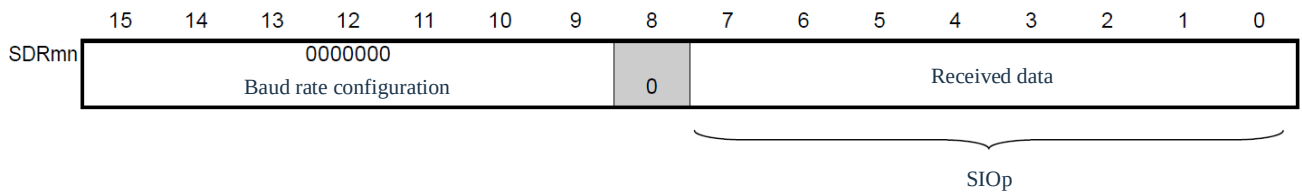
(a) Serial mode register mn (SMRmn)



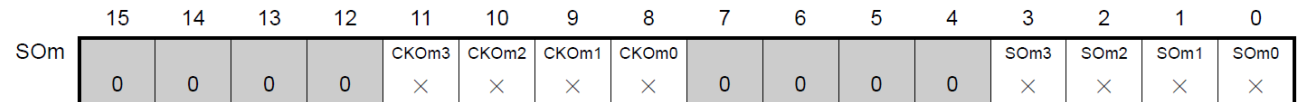
(b) Serial communication operation configuration register mn (SCRmn)



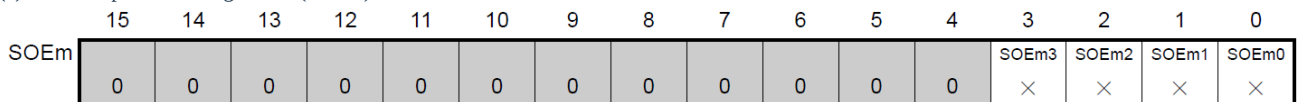
(c) Serial data register mn (SDRmn) (low 8 bits: SIOp)



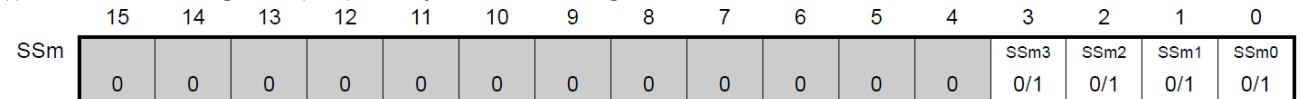
(d) Serial output register m (Som) Not used in this mode.



(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.



Note Limited to SCR00 register and SCR01 register, the others are fixed as “1”.

Remark 1.m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21, 30, 31)
mn=00~03, 10~11

- Fixed in Slave 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) Operation steps

Figure 13-56 Initial setup steps for slave reception

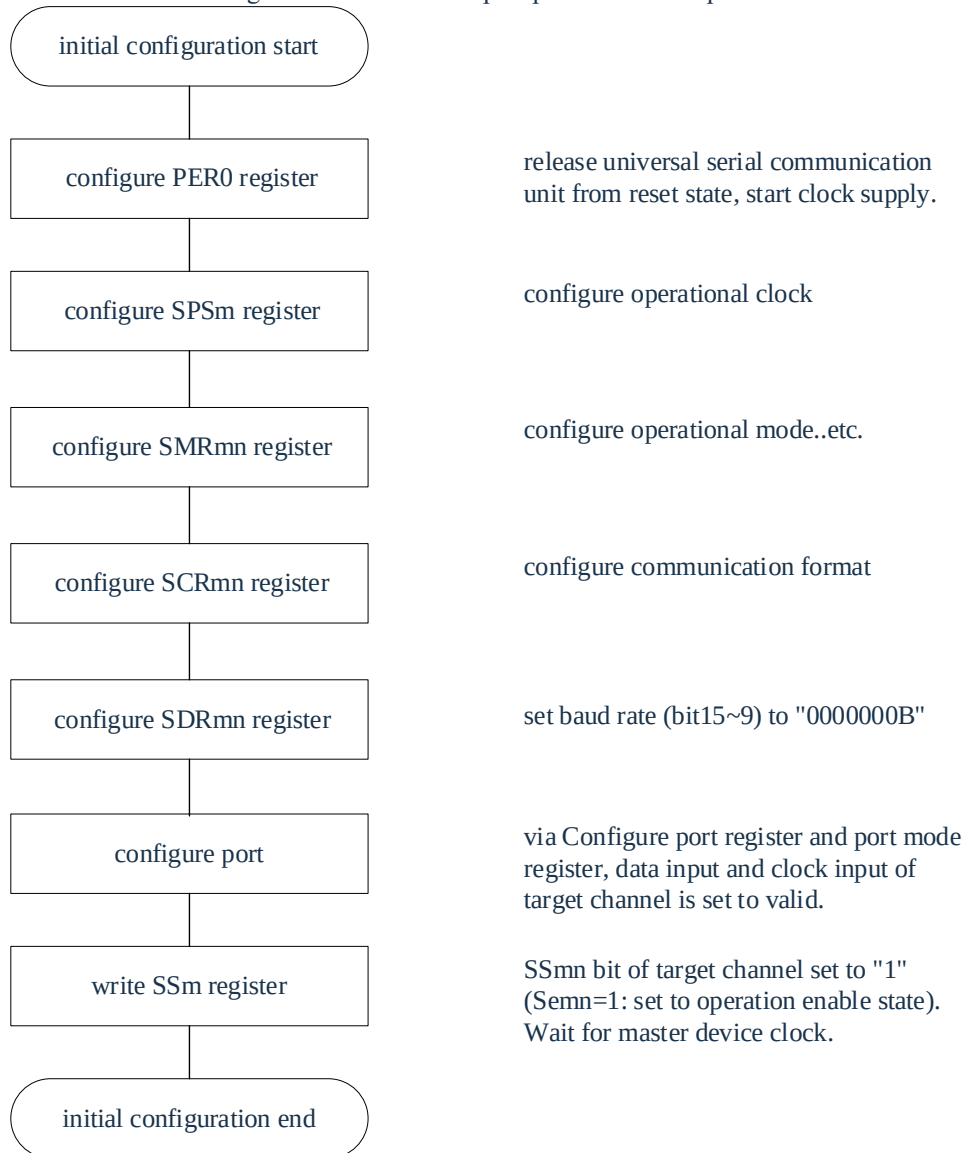


Figure 13-57 Stop steps for slave reception

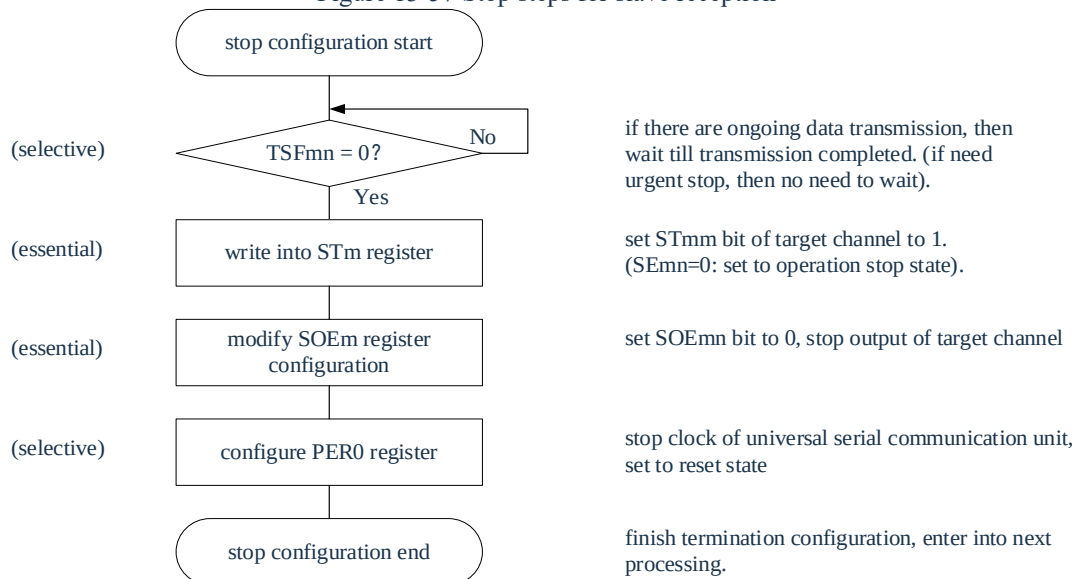
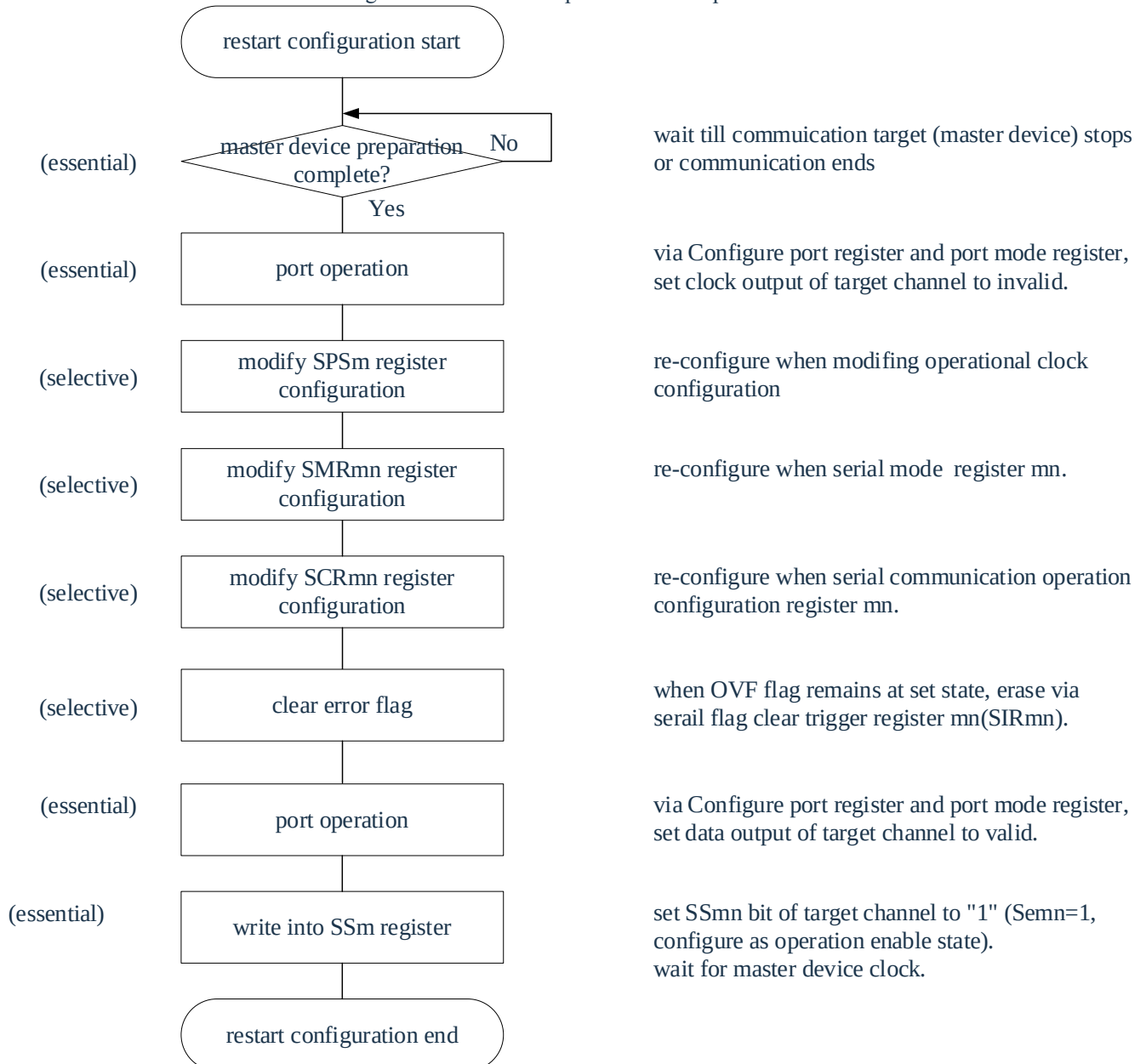


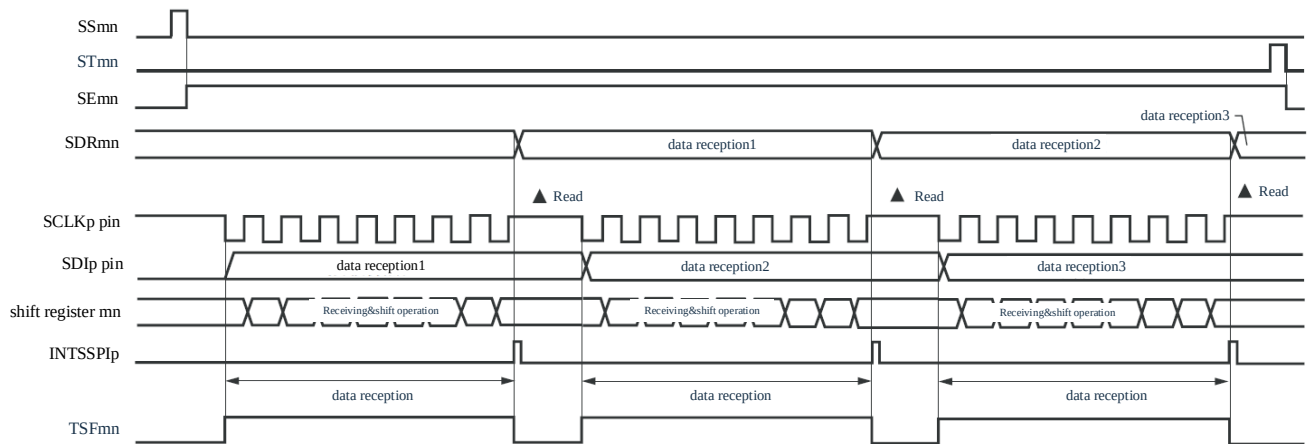
Figure 13-58 Restart steps for slave reception



Remark 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.

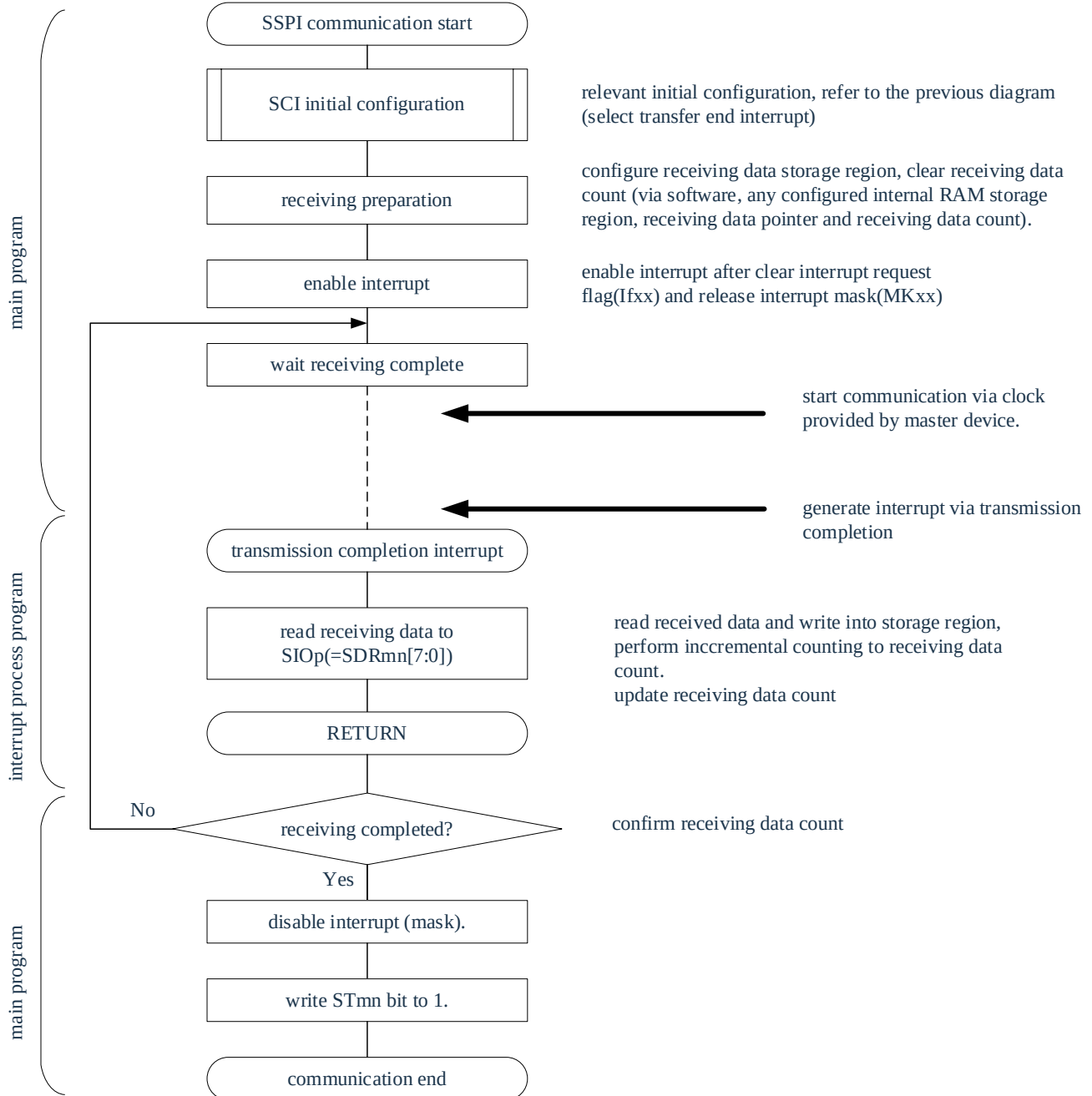
(3) Processing flow (single receive mode)

Figure 13-59 Timing diagram of slave reception (single receive mode) (type 1: DAPmn=0, CKPmn=0)



Remark m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21, 30, 31)
mn=00~03, 10~11.

Figure 13-60 Flowchart of slave reception (single receive mode)



13.5.6 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.

3-wire serial I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
Target channel	Channel 0 of SCI0	Channel 1 of SCI0	Channel 2 of SCI0	Channel 3 of SCI0	Channel 0 of SCI1	Channel 1 of SCI1
Used pin	SCLK00, SDI00, SDO00	SCLK01, SDI01, SDO01	SCLK10, SDI10, SDO10	SCLK11, SDI11, SDO11	SCLK20, SDI20, SDO20	SCLK21, SDI21, SDO21
Interrupt	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	Can select transfer end interrupt (single transfer mode) or buffer empty interrupt (continuous transfer mode).					
Error detection flag	Only the overflow error detection flag (OVFmn).					
Transfer data length	7 or 8 bits					
Transfer rate	Max. $f_{MCK}/6$ [Hz] ^{Note1, 2}					
Data phase	It can be selected via 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 via the CKPmn bit of the SCRmn register. • CKPmn=0: Non-reverse • CKPmn=1: Reverse					
Data direction	MSB first or LSB first					

Notes 1. The maximum transfer rate $f_{MCK}/6$ [Hz] is used because the external serial clock input from SCLK00, SCLK01, SCLK10, SCLK11, SCLK20, and SCLK21 pins is sampled internally and then used.

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).

Remarks 1. f_{MCK} : Operation clock frequency of the target channel

2. m: unit number (m=0, 1) n: channel number (n=0~3) mn=00~03, 10~11

(1) Register setting

Figure 13-61 3-wire serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21)
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			DLSmn1	DLSmn0
1	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1 Note	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
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (low 8 bits: SIOp)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0000000								0	Configuration of transmit data/receive data register						
Baud rate configuration									SIOp						

(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

Note: Limited to SCR00 register and SCR01 register, the others are fixed as “1”.

Caution: Data must be sent to the SIOp register settings before the master device starts the output clock.

Remarks 1. m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)

mn=00~03, 10~11

2. : Fixed in SSPI slave T&R mode.

 : Cannot be set (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) Operation steps

Figure 13-62 Initial setup steps for slave transmission and reception



Caution Before the master device starts to output the clock, the SIOp register must be set to transmit data.

Figure 13-63 Stop steps for slave transmission and reception

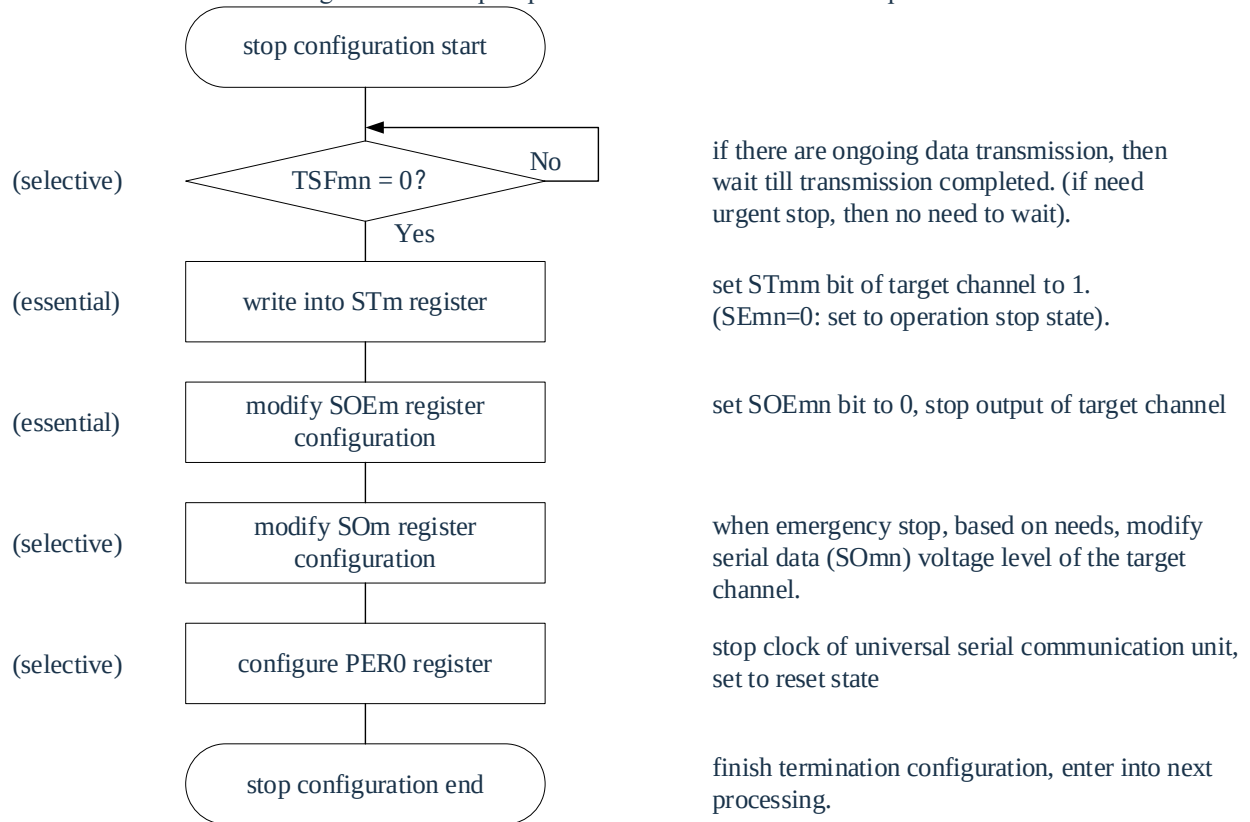
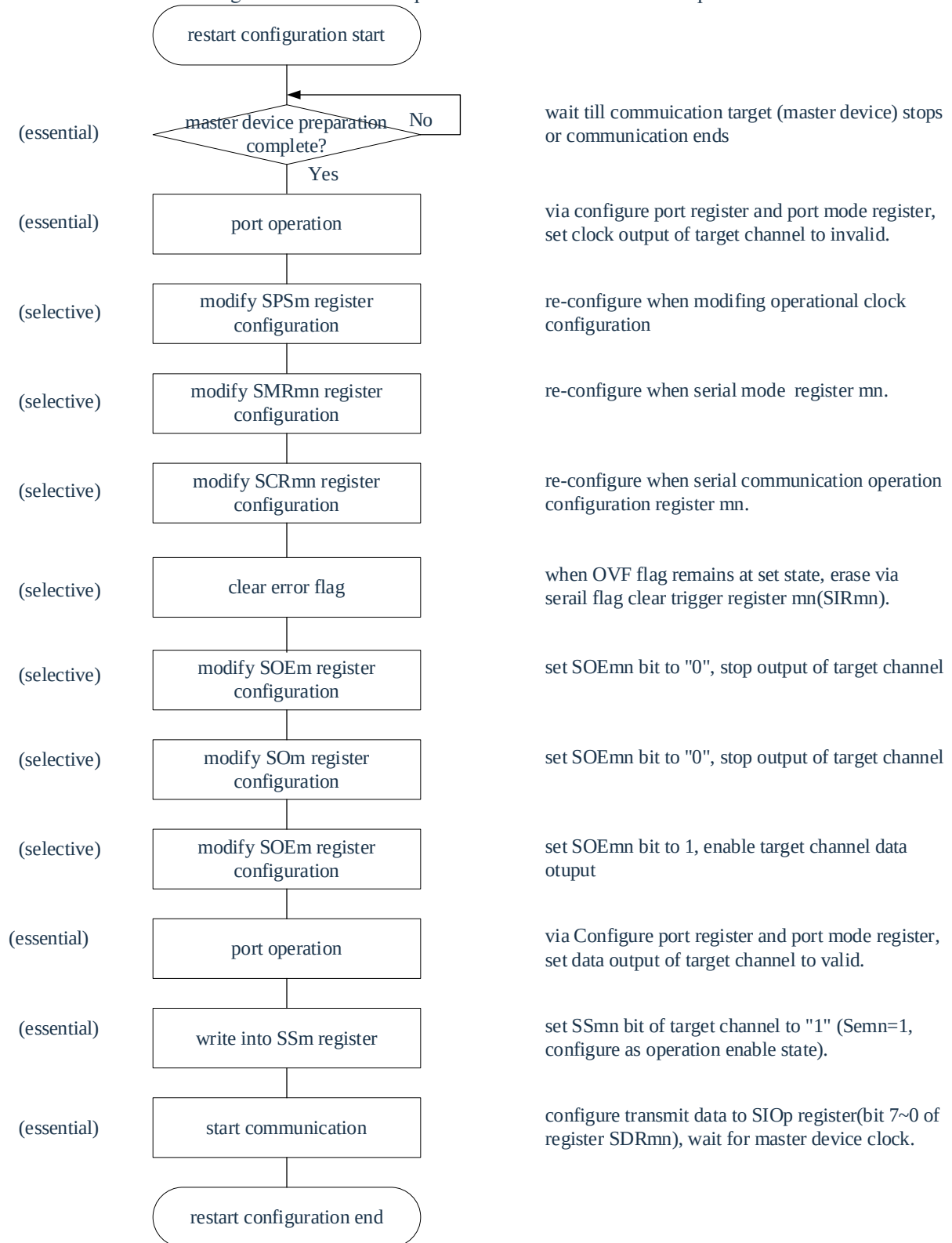


Figure 13-64 Restart steps for slave transmission and reception

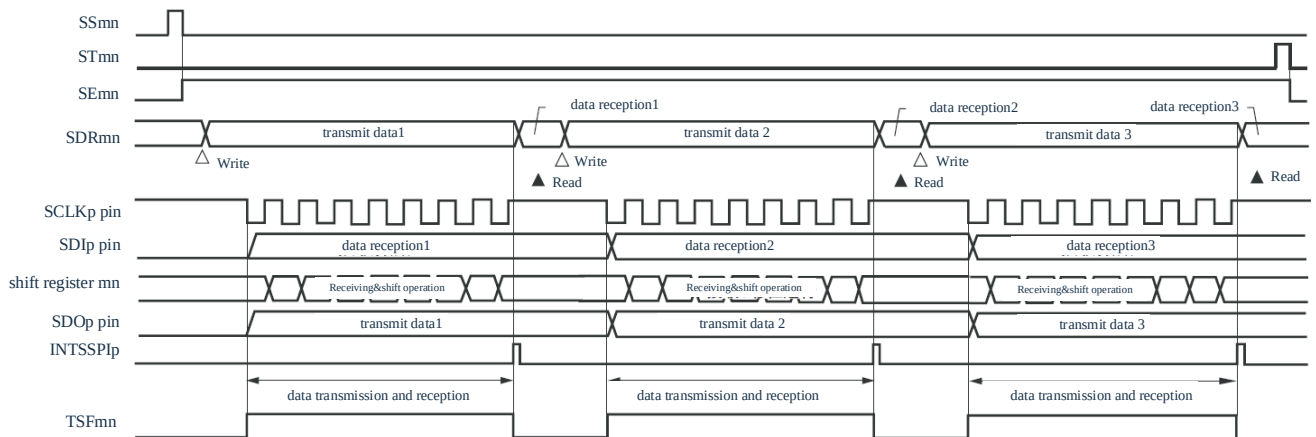


Cautions 1. Before the master device starts to output the clock, data must be sent to the SIOp register settings.

2. 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.

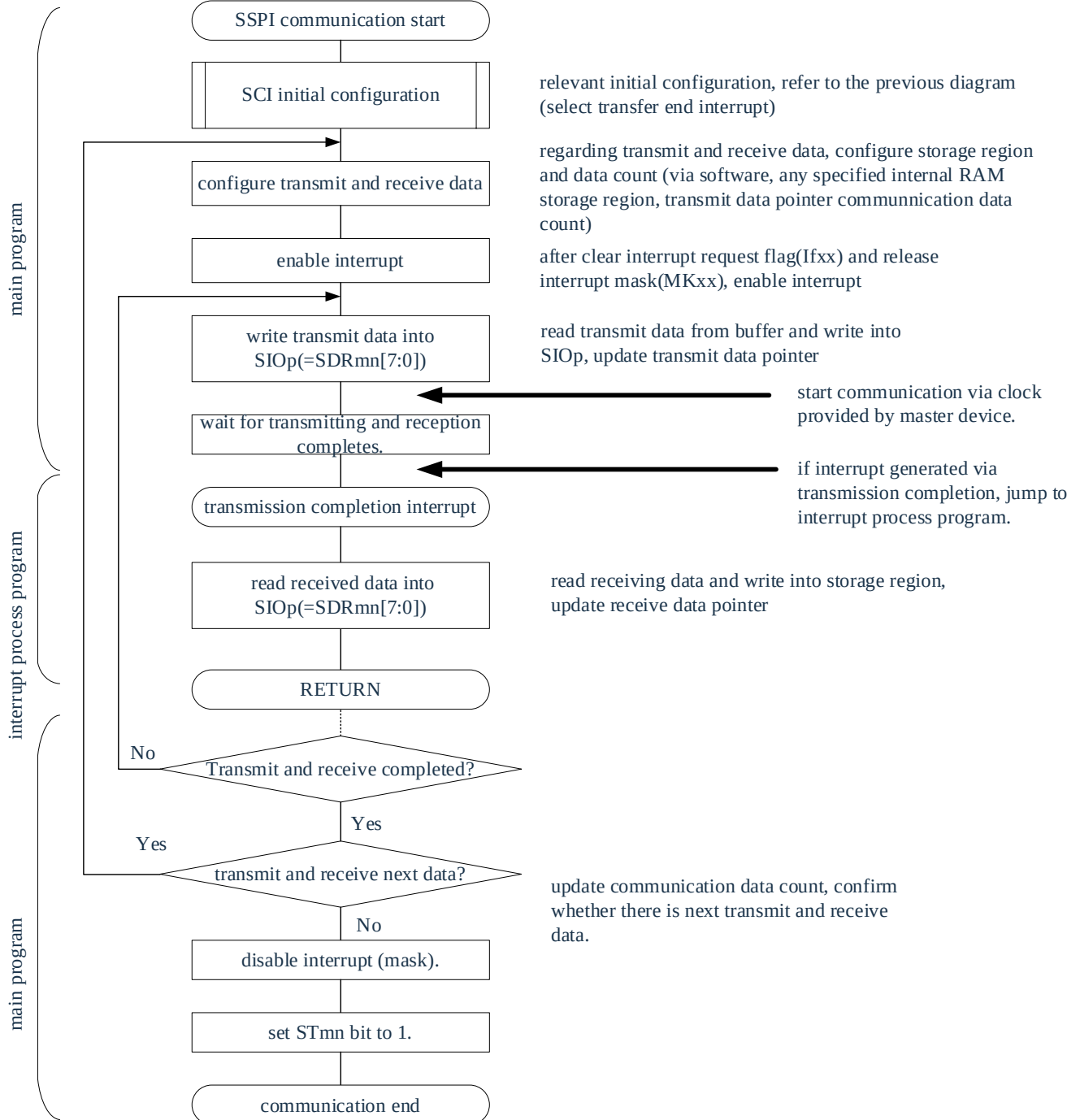
(3) Processing flow (single transmit and receive mode)

Figure 13-65 Timing diagram of slave transmission and reception (single transmit and receive mode)
(type 1: DAPmn=0, CKPmn=0)



Remark m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21)
mn=00~03, 10~11

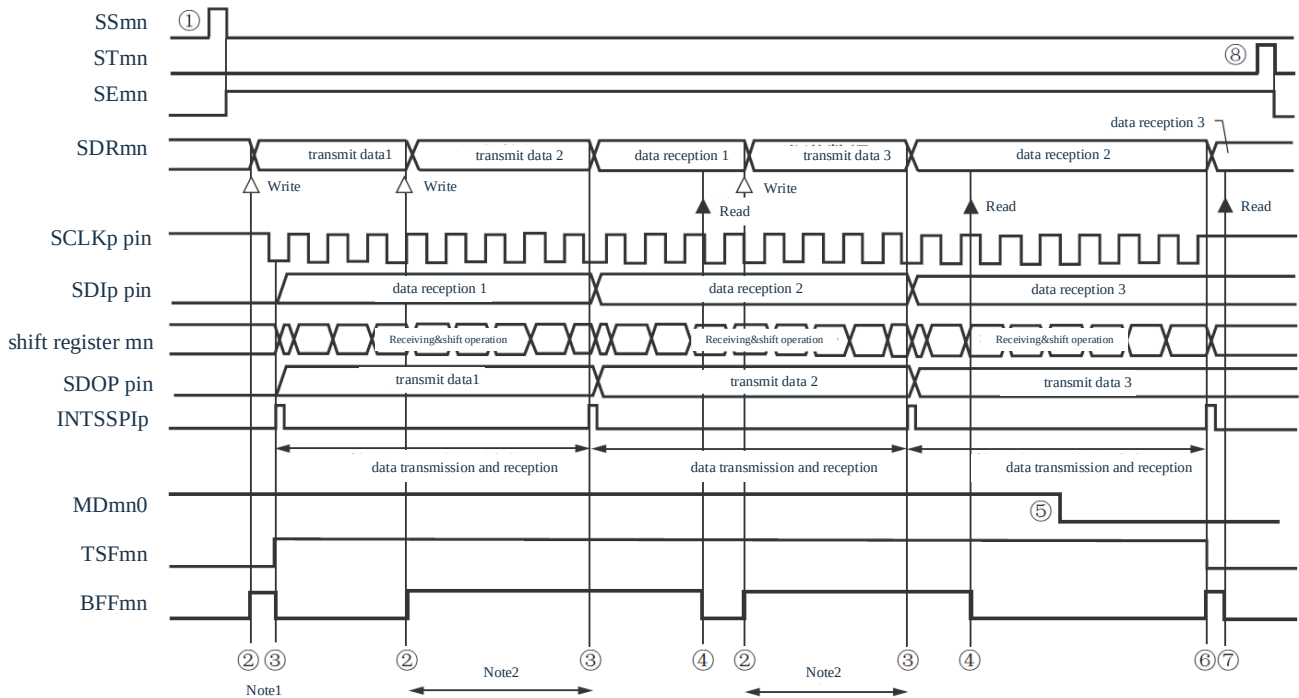
Figure 13-66 Flowchart of slave transmission and reception (single transmit and receive mode)



Caution Data must be sent to the SIOp register settings before the master device starts the output clock.

(4) Processing flow (continuous send and receive mode)

Figure 13-67 Timing diagram of slave transmission and reception (continuous transmit and receive mode)
(type 1: DAPmn=0, CKPmn=0)



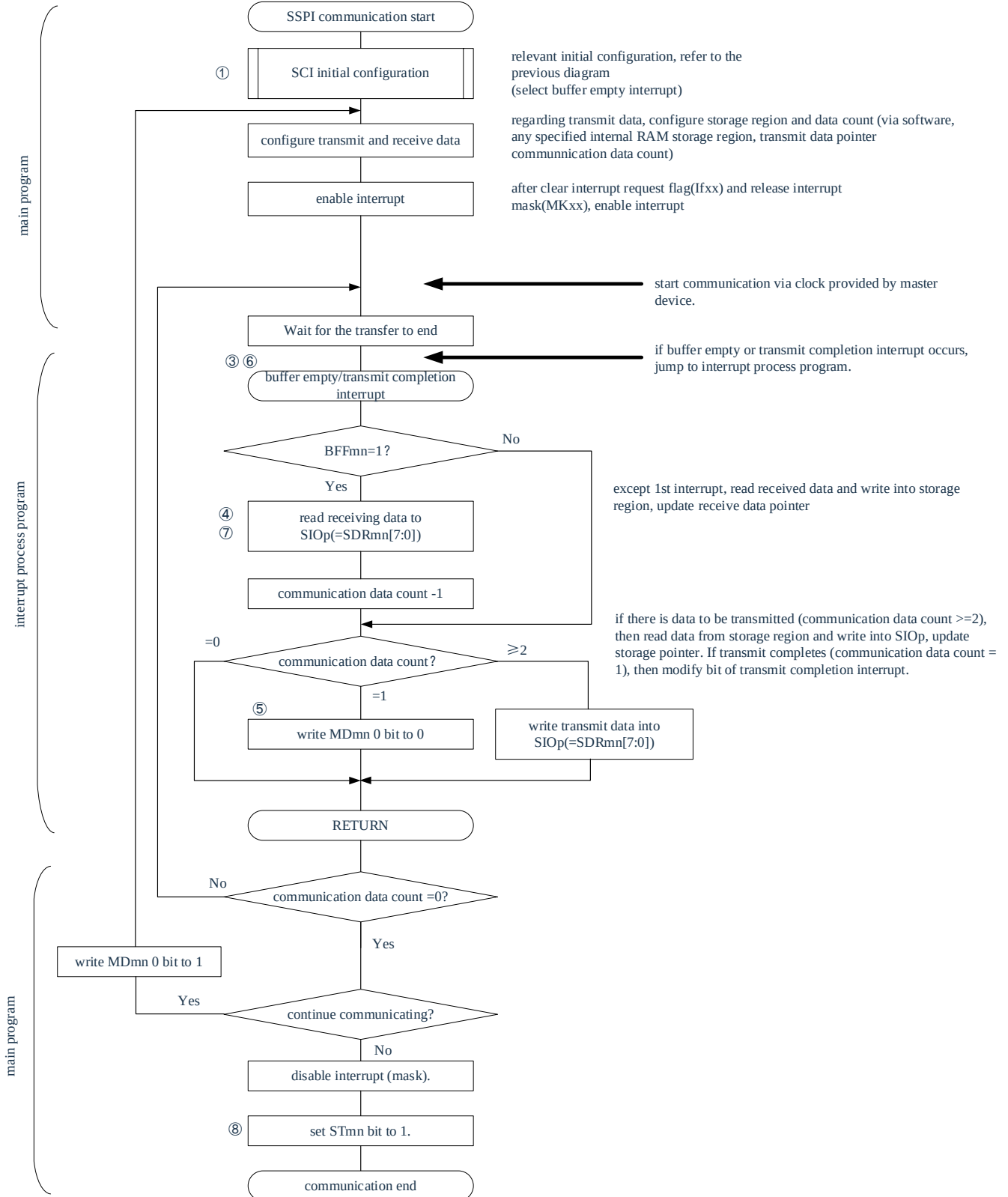
- Notes
1. If the BFFmn bit of the serial status register mn (SSRmn) is “1” (valid data is saved in the serial data register mn (SDRmn) to write the send data to the SDRmn memory, and override the sent data.
 2. If the SDRmn register is read during this period, the transmitted data can be read. At this point, the transfer run 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.

Remarks

1. ① to ⑧ in the diagram correspond to ① to ⑧ in “Figure 13-68 Flowchart of slave transmission and reception (continuous transmit and receive mode)”.
2. m: unit number (m=0, 1) n: channel number (n=0~3) p: SSPI number (p=00, 01, 10, 11, 20, 21) mn=00~03, 10~11

Figure 13-68 Flowchart of slave transmission and reception (continuous transmit and receive mode)



Caution Data must be sent to the SIOp register settings before the master device starts the output clock.

Remark ① to ⑧ correspond to ① to ⑧ in “Figure 13-67 Timing diagram of slave transmission and reception (continuous transmit and receive mode)”.

13.5.7 Calculation of Transfer Clock Frequency

3-wire serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21) communication transmission clock frequency can be calculated using the following calculation equations.

(1) Master device

$$(\text{Transfer clock frequency}) = \{ \text{Operation clock frequency of the target channel}(f_{\text{MCK}}) \} \div (\text{SDRmn}[15:9] + 1) \div 2 [\text{Hz}]$$

(2) Slave device

$$(\text{Transfer clock frequency}) = \{ \text{Serial clock (SCLK) frequency provided by the master device} \} \text{Note} [\text{Hz}]$$

Note The maximum enable transmit clock frequency is $f_{\text{MCK}}/6$.

Remark Since the value of SDRmn[15:9] is the value of bit15 to 9 (0000000B to 1111111B) of serial data register mn (SDRmn), it is 0 to 127.

The operation clock (F_{MCK}) depends on the bit15 (CKSmn) of the serial clock selection register m (SPSm) and the serial mode register mn (SMRmn).

Table 13-2 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} =32MHz in operation
0	X	X	X	X	0	0	0	0	f _{CLK}	32MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	16MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	8MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	4MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	2MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	1MHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	500kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	250kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	125kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	62.5kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	31.25kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	15.63kHz
	X	X	X	X	1	1	0	0	f _{CLK} /2 ¹²	7.81kHz
	X	X	X	X	1	1	0	1	f _{CLK} /2 ¹³	3.91kHz
	X	X	X	X	1	1	1	0	f _{CLK} /2 ¹⁴	1.95kHz
	X	X	X	X	1	1	1	1	f _{CLK} /2 ¹⁵	977Hz
1	0	0	0	0	X	X	X	X	f _{CLK}	32MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	16MHz
	0	0	1	0	X	X	X	X	f _{CLK} /2 ²	8MHz
	0	0	1	1	X	X	X	X	f _{CLK} /2 ³	4MHz
	0	1	0	0	X	X	X	X	f _{CLK} /2 ⁴	2MHz
	0	1	0	1	X	X	X	X	f _{CLK} /2 ⁵	1MHz
	0	1	1	0	X	X	X	X	f _{CLK} /2 ⁶	500kHz
	0	1	1	1	X	X	X	X	f _{CLK} /2 ⁷	250kHz
	1	0	0	0	X	X	X	X	f _{CLK} /2 ⁸	125kHz
	1	0	0	1	X	X	X	X	f _{CLK} /2 ⁹	62.5kHz
	1	0	1	0	X	X	X	X	f _{CLK} /2 ¹⁰	31.25kHz
	1	0	1	1	X	X	X	X	f _{CLK} /2 ¹¹	15.63kHz
	1	1	0	0	X	X	X	X	f _{CLK} /2 ¹²	7.81kHz
	1	1	0	1	X	X	X	X	f _{CLK} /2 ¹³	3.91kHz
	1	1	1	0	X	X	X	X	f _{CLK} /2 ¹⁴	1.95kHz
	1	1	1	1	X	X	X	X	f _{CLK} /2 ¹⁵	977Hz

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).

Remarks 1. X: Ignore

2. m: unit number (m=0, 1) n: channel number (n=0~3) mn=00~03, 10~11.

13.5.8 Procedure for Handling Errors During 3-Wire Serial I/O Communication (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21)

In 3-wire serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21), the processing steps when an error occurs during communication are shown in Figure 13-69.

Figure 13-69 Steps to handle when an overflow 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 overflow 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, 1) n: channel number (n=0~3) mn=00~03, 10~11.

13.6 Operation of Clock-Synchronous Serial Communication with Slave Selection Input Function

Channel 0 of SCI0 is the channel that supports clock-synchronous serial communication with the slave select input function.

[Transmit and receive data]

- Data length of 7 or 8bits
- Phase control of data transmission and reception
- 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: $\text{Max.f}_{\text{MCK}}/6$

[Interrupt function].

- Transfer end interrupt, buffer empty interrupt

[Error detection flag].

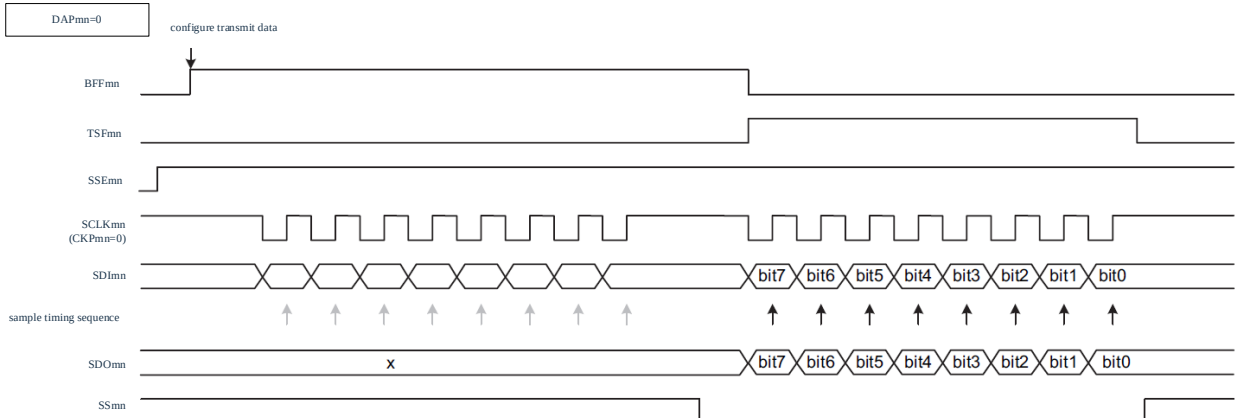
- Overflow error

Note it must be used within the scope of the SCLK Cycle Time (t_{KCY}) characteristics. Please refer to the data sheet for details.

The slave select input function operates in three types of communication:

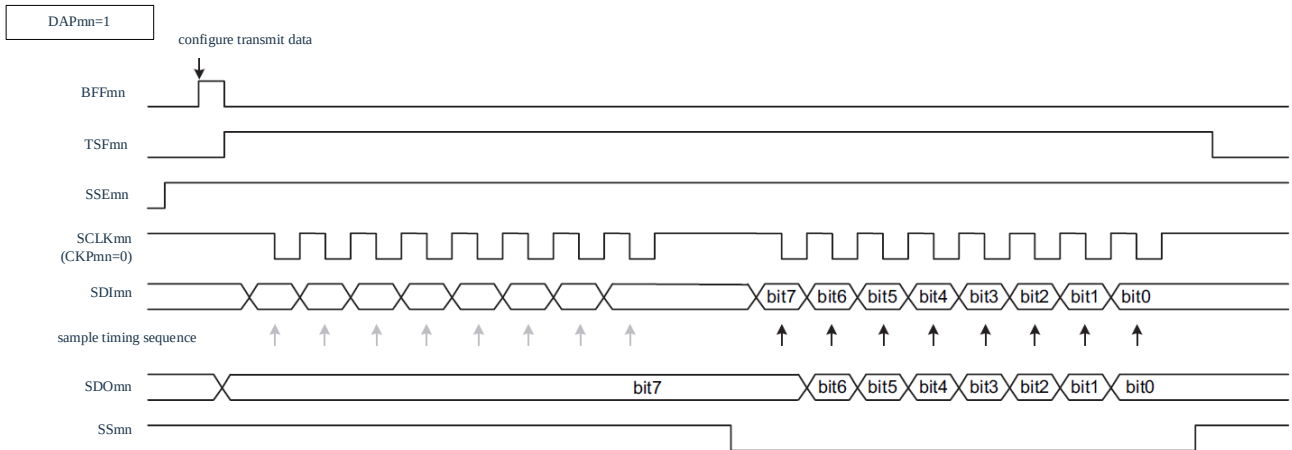
- Slave transmission (see 13.6.1).
- Slave reception (see 13.6.2).
- Slave transmission and reception (see 13.6.3).

Figure 13-71 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).

13.6.1 Slave Transmission

Slave transmission refers to the operation where this product transmits data to other devices while receiving the transfer clock from other devices.

Slave select input function	SSPI00
Target channel	Channel 0 for SCI0
Used pin	SCLK00, SDO00, SS00
Interrupt	INTSSPI00 Can select transfer end interrupt (single transfer mode) or buffer empty interrupt (continuous transfer mode).
Error detection flag	Only the overflow error detection flag (OVFmn).
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{MCK}/6[Hz]$ ^{Note1, 2}
Data phase	It can be selected via the DAPmn bit of the SCRmn register. • DAPmn=0: The data output starts 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 via the CKPmn bit of the SCRmn register. • CKPmn=0: Non-reverse • CKPmn=1: Reverse
Data direction	MSB first or LSB first
Slave select input function	Operation of the slave selection function can be selected.

Notes 1. Because the external serial clock of the SCLK00 pin input is used internally, the maximum transfer rate is $f_{MCK}/6[Hz]$.

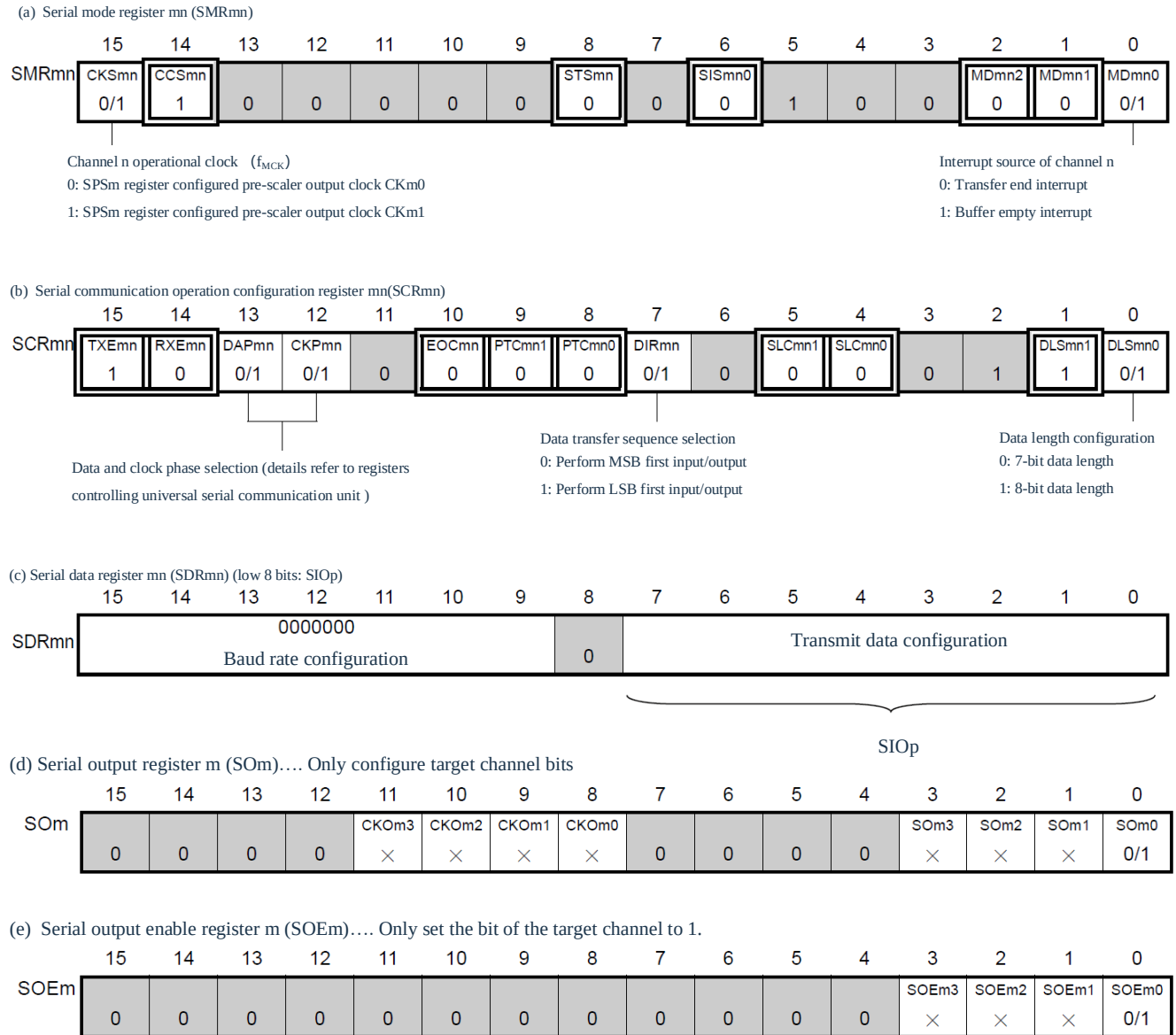
2. It must be used within the range of peripheral functional characteristics (refer to the datasheet) that fulfill this condition and satisfy the electrical characteristics.

Remarks 1. f_{MCK} : Operation clock frequency of the target channel

2. m: unit number (m=0) n: channel number (n=0).

(1) Register setting

Figure 13-72 Example of register settings when slave select input function (SSPI00) slave transmits (1/2)



Remarks 1. m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

2. : Fixed in SSPI slave send 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 13-72 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													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	×	×	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						ISC1	ISC0
	0/1	0	0	0	0	0	0/1	0/1

0: SS00 pin input invalid
1: SS00 pin input valid

Remarks 1. m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

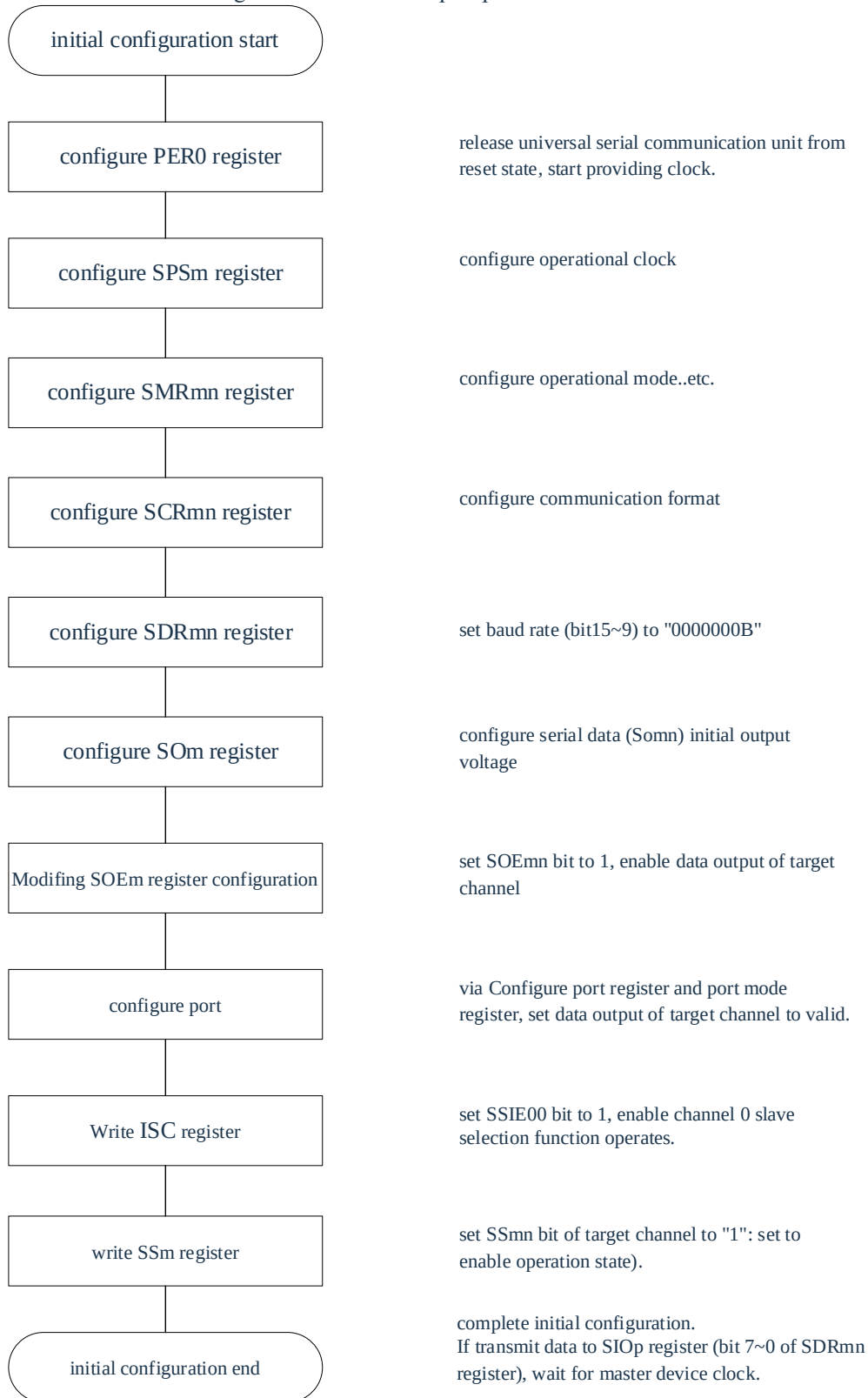
2. : Fixed in SSPI slave transmit mode. : Cannot be set (initial value).

×: 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.

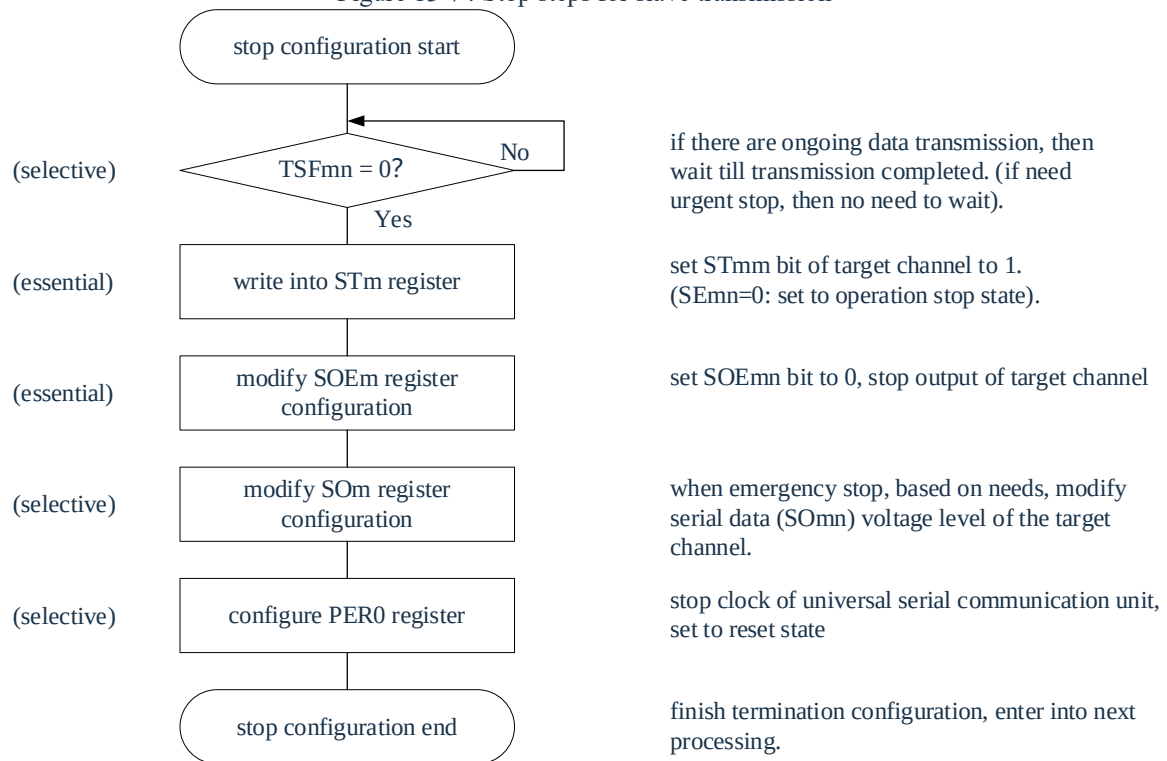
(2) Operation steps

Figure 13-73 Initial setup steps for slave transmission



Remark m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

Figure 13-74 Stop steps for slave transmission

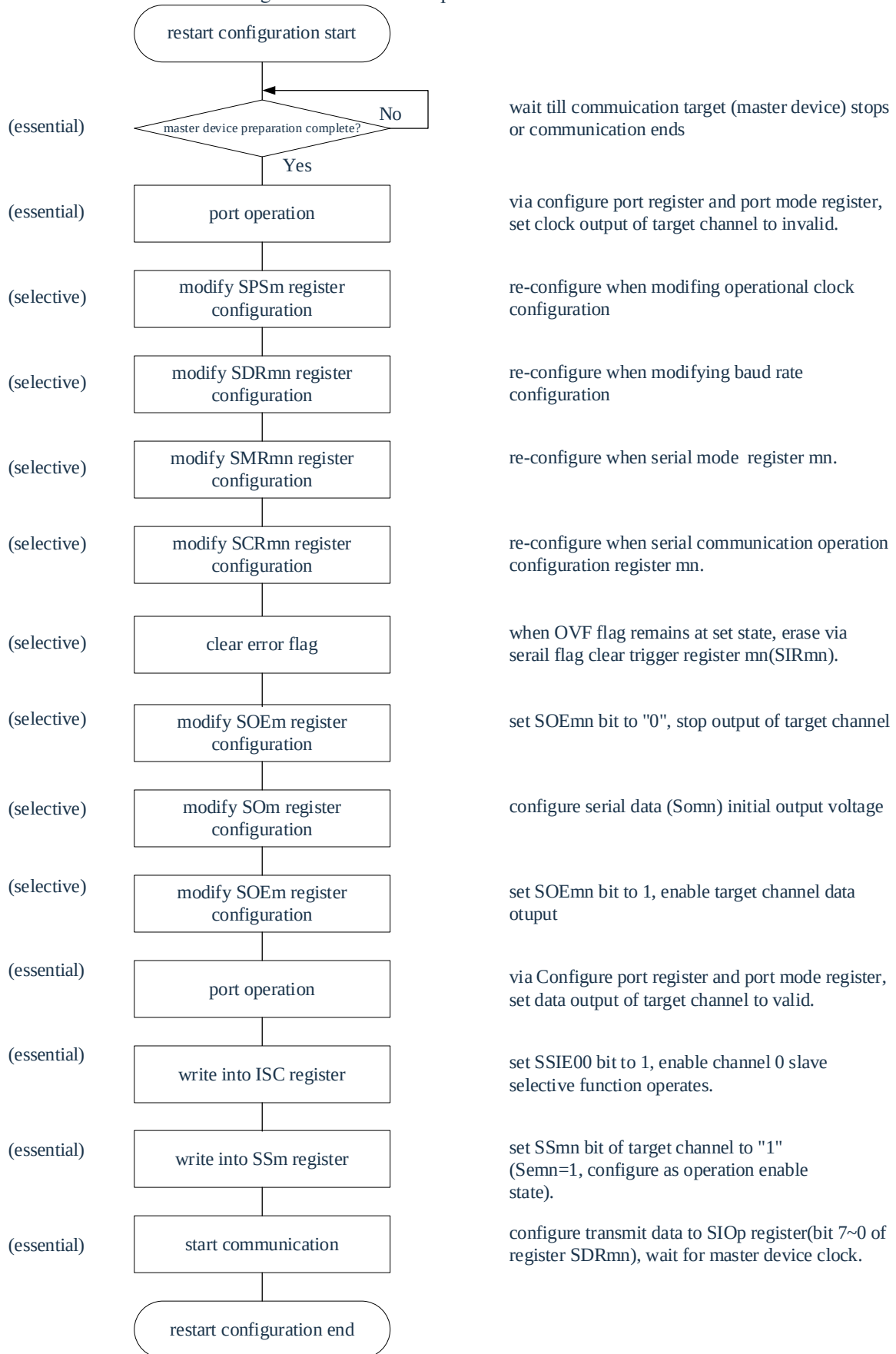


Remark m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

Notes 1. If you override PER0 in the abort setting to stop the clock, you must wait until the communication target (master device) stops or the communication is over to make the initial setting instead of starting the setting again.

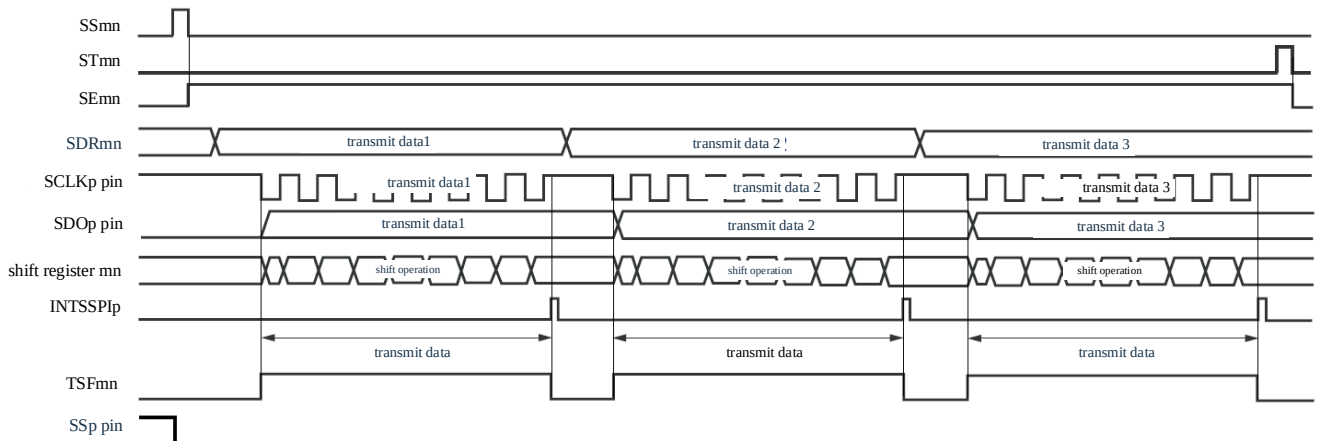
2. m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00).

Figure 13-75 Restart steps for slave transmission



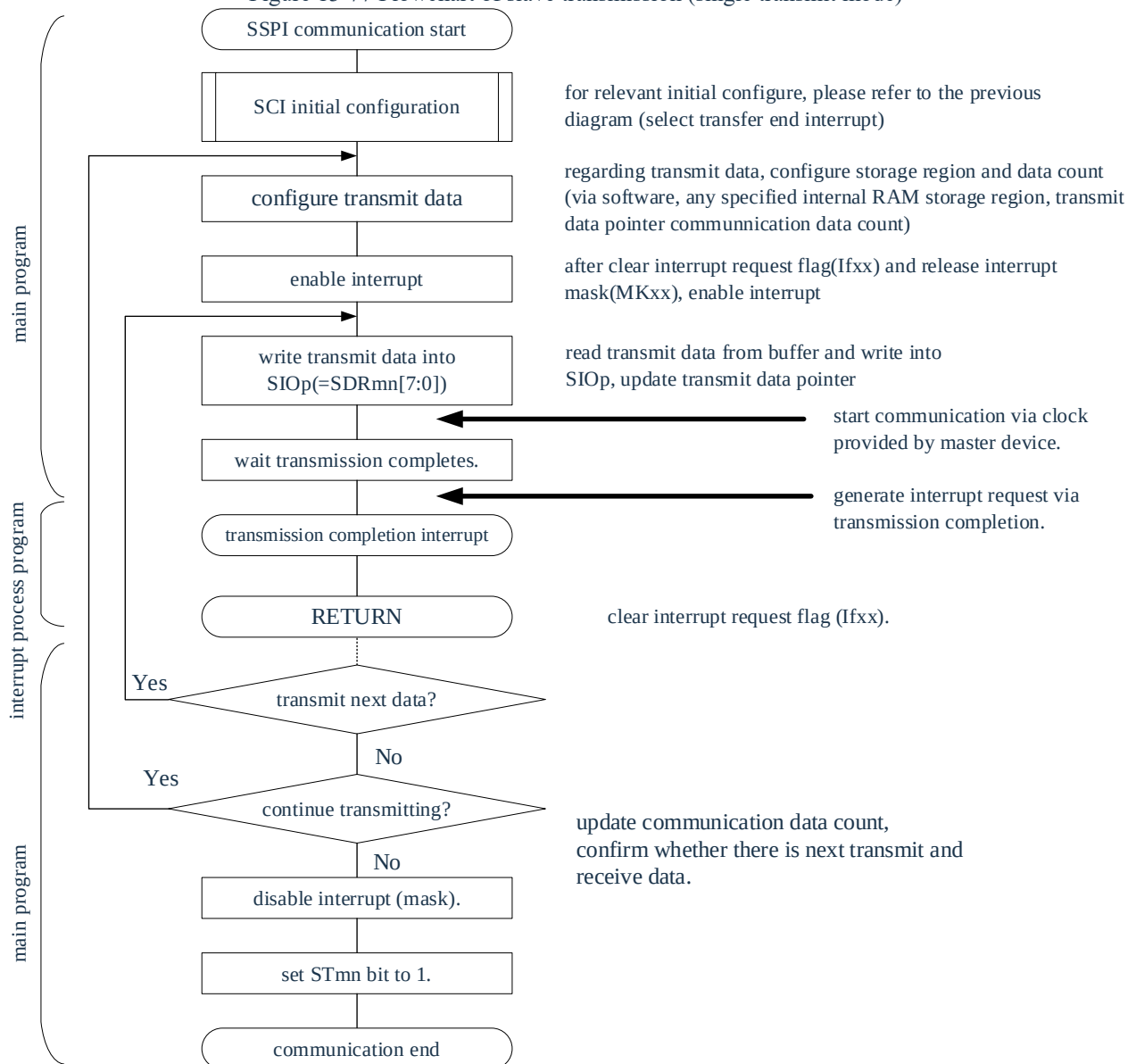
(3) Processing flow (single transmit mode)

Figure 13-76 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)

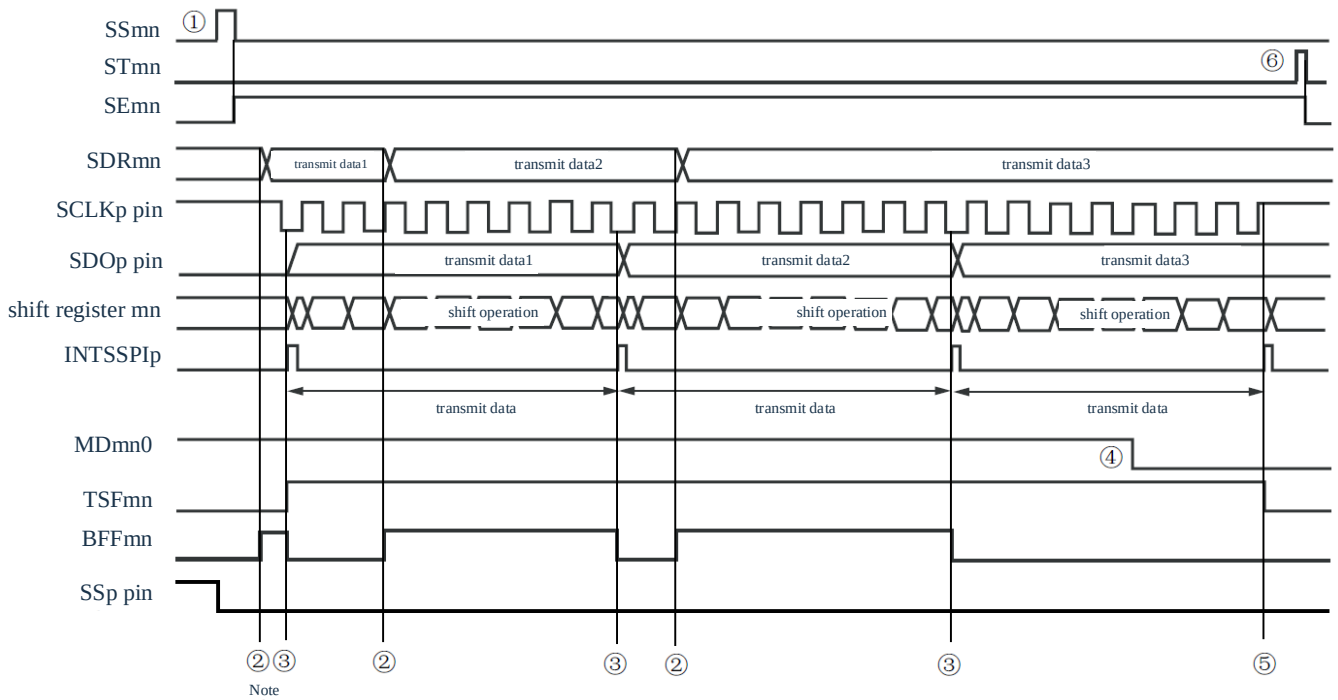
Figure 13-77 Flowchart of slave transmission (single transmit mode)



Remark m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

(4) Processing flow (continuous transmit mode)

Figure 13-78 Timing diagram of slave transmit (continuous transmit mode) (type 1: DAPmn=0, CKPmn=0)

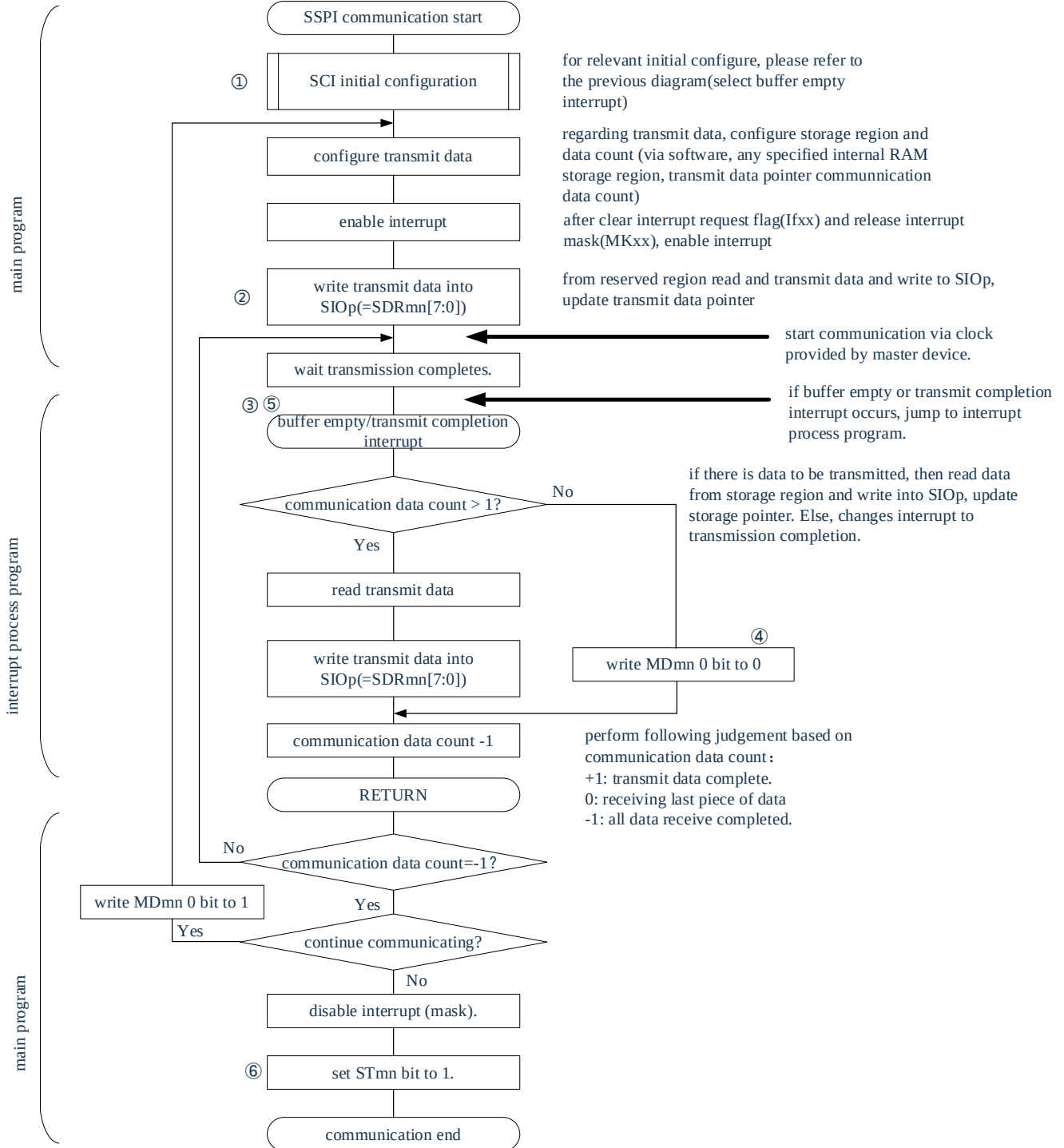


Note If the BFFmn bit in the Serial Status Register mn (SSRmn) is 1 (indicating that valid data is stored in the Serial Data Register mn (SDRmn)), writing data to the SDRmn register will overwrite the transmission data.

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) p: SSPI number (p=00)

Figure 13-79 Flowchart of slave transmission (continuous send mode)



Remarks 1. ① to ⑥ correspond to ① to ⑥ in “Figure 13-78 Timing diagram of slave transmit (continuous transmit mode)”.

2. m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

13.6.2 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.

Slave select input function	SSPI00
Target channel	Channel 0 for SCI0
Used pin	SCLK00, SDI00, SS00
Interrupt	INTSSPI00 Limited to transmit complete interrupts (disable setting buffer empty interrupts).
Error detection flag	Only the overflow error detection flag (OVFmn).
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{MCK}/6$ [Hz] ^{Note 1, 2}
Data phase	It can be selected via the DAPmn bit of the SCRmn register. • DAPmn=0: The data output starts 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 via the CKPmn bit of the SCRmn register. • CKPmn=0: Non-reverse • CKPmn=1: Reverse
Data direction	MSB first or LSB first
Slave select input function	Select the operation of the Slave select input function.

Notes

1. The maximum transfer rate is $f_{MCK}/6$ [Hz] because the external serial clock input to the SCLK00 pin is sampled internally and then used.
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).

Remarks 1. f_{MCK} : Operation clock frequency of the target channel
2. m: unit number (m=0) n: channel number (n=0).

(1) Register setting

Figure 13-80 Example of register setting contents for slave selection input function (SSPI00) slave reception (1/2)

(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

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			DLSmn1	DLSmn0
0	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	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
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (low 8 bits: SIOp)

SDRmn

1514131211109876543210

0000000

Baud rate configuration

0

Receive data

SIOp

(d) 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
				CKOm3	CKOm2	CKOm1	CKOm0					SOM3	SOM2	SOM1	SOM0
0	0	0	0	×	×	×	×	0	0	0	0	×	×	×	×

(e) 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
												SOEm3	SOEm2	SOEm1	SOEm0
0	0	0	0	0	0	0	0	0	0	0	0	×	×	×	×

Remarks 1. m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

2. : Fixed in slave 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 13-81 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													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	×	×	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						ISC1	ISC0
	0/1	0	0	0	0	0	0/1	0/1

0: SS00 pin input invalid
1: SS00 pin input valid

Remarks 1.m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

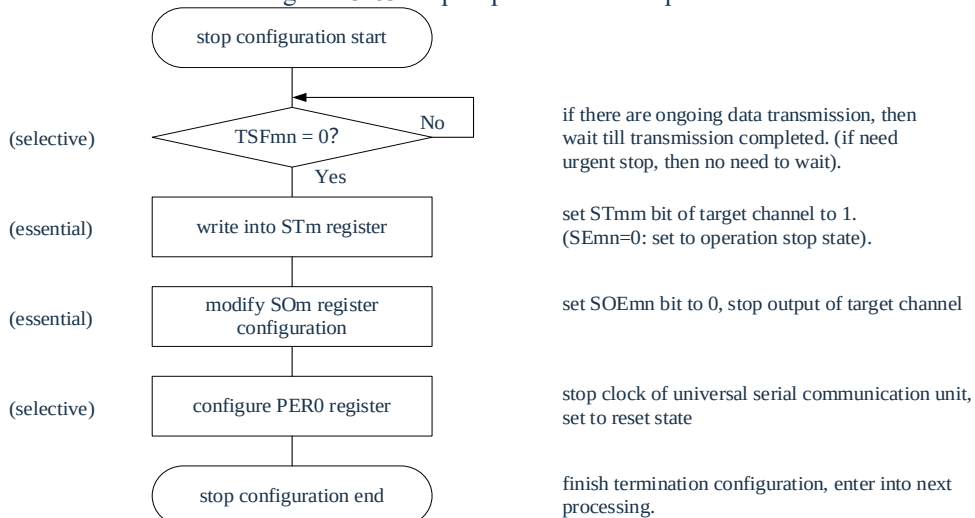
2. ☐: Fixed in slave receive mode. ☐: Cannot be set (initial value).
 ×: This is the bit that cannot be used in this mode (set the initial value if it is not used in other modes either).
 0/1: Set “0” or “1” according to the user’s purpose.

(2) Operation steps

Figure 13-82 Initial setup steps for slave reception

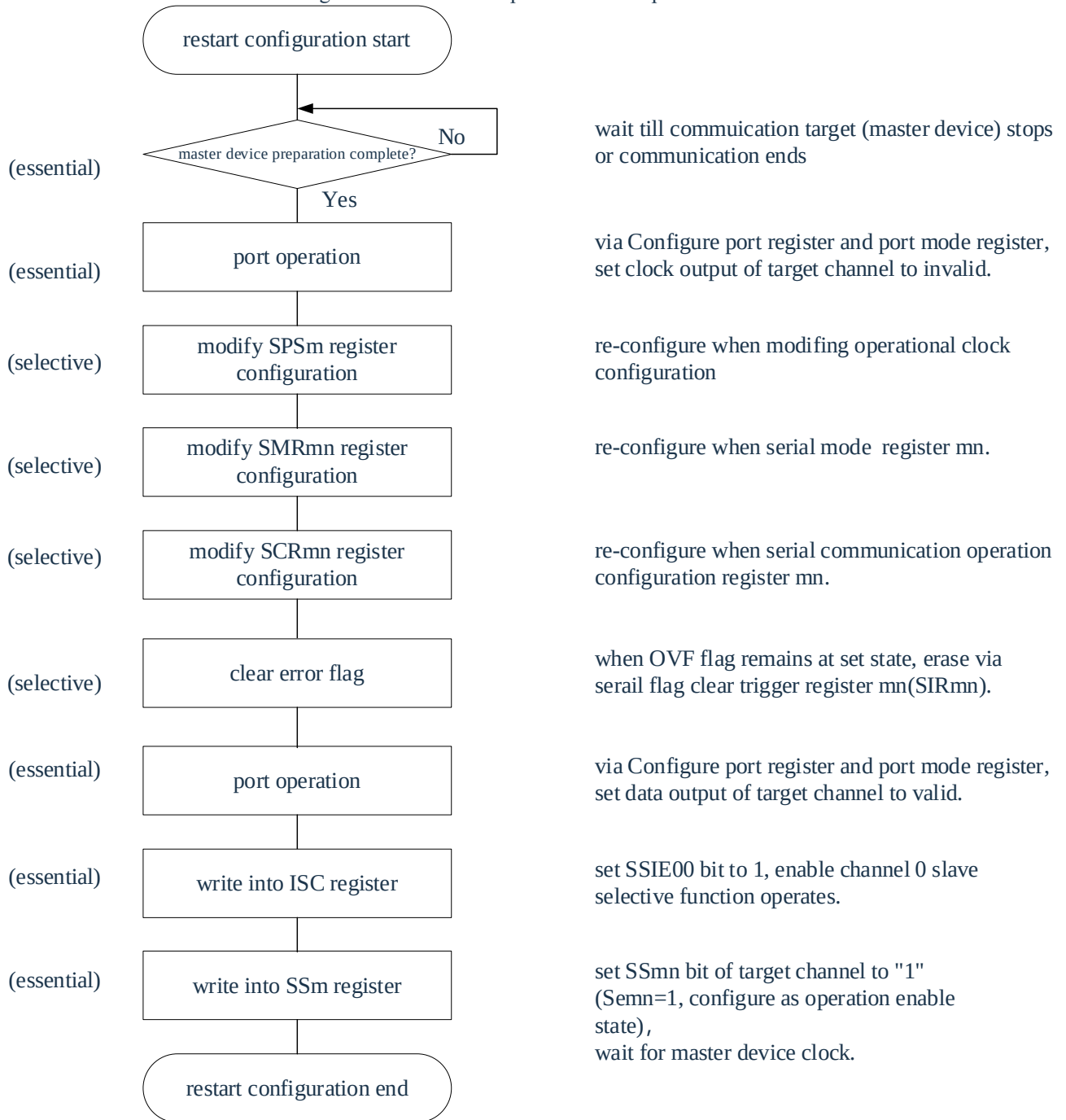


Figure 13-83 Stop steps for slave reception



Remark m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

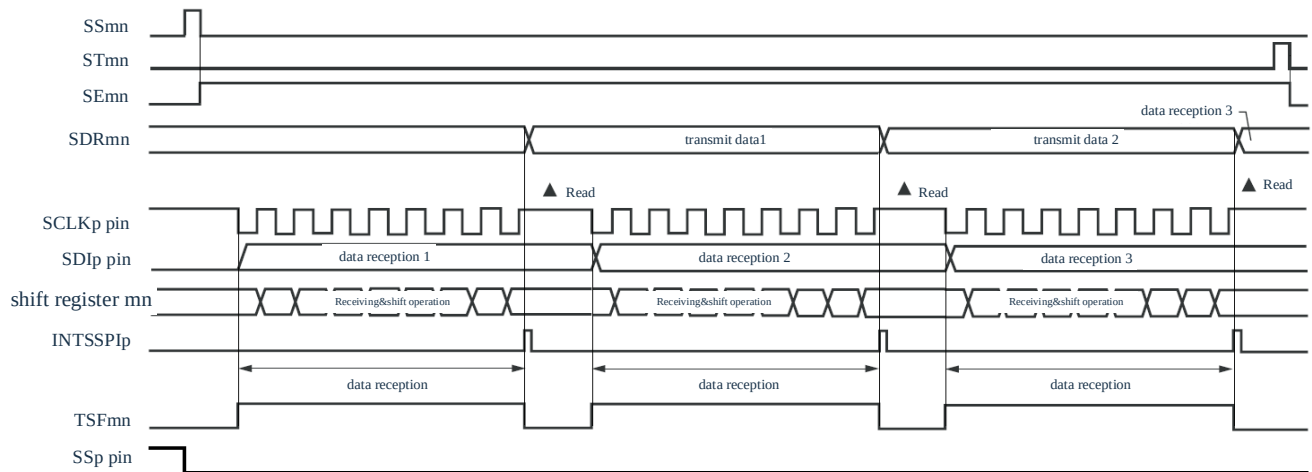
Figure 13-84 Restart steps for slave reception



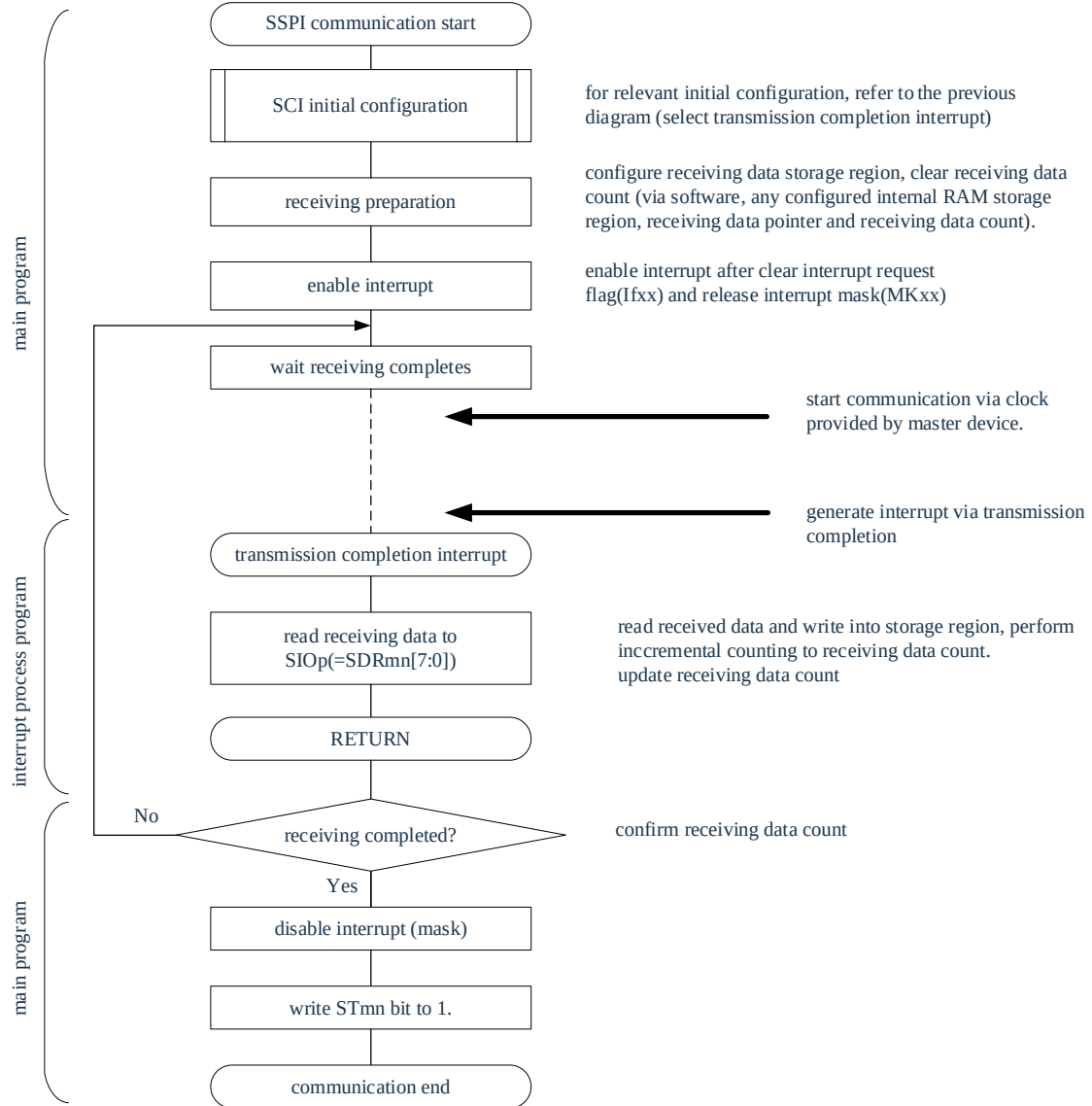
Remark m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

(3) Processing flow (single receive mode)

Figure 13-85 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 13-86 Flowchart of slave reception (single receive mode)


13.6.3 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.

Slave select input function	SSPI00
Target channel	Channel 0 for SCI0
Used pin	SCLK00, SDI00, SDO00, SS00
Interrupt	INTSSPI00 Can select transfer end interrupt (single transfer mode) or buffer empty interrupt (continuous transfer mode).
Error detection flag	Only the overflow error detection flag (OVFmn).
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{MCK}/6$ [Hz] ^{Note 1, 2}
Data phase	It can be selected via the DAPmn bit of the SCRmn register. • DAPmn=0: The data output starts 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 via the CKPmn bit of the SCRmn register. • CKPmn=0: Non-reverse • CKPmn=1: Reverse
Data direction	MSB first or LSB first
Slave select input function	Operation of the slave select input function can be selected.

Notes

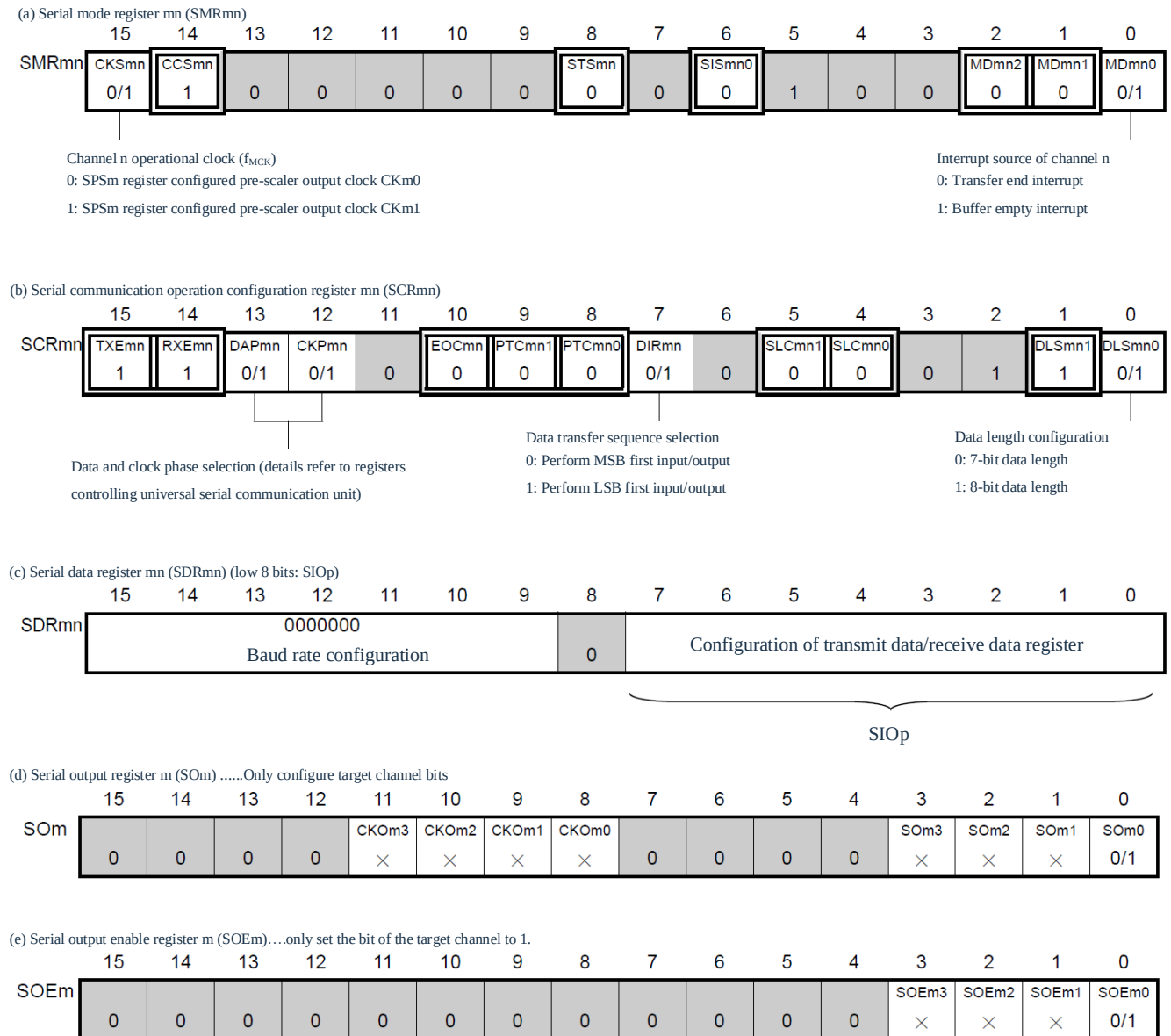
1. The maximum transfer rate is $f_{MCK}/6$ [Hz] because the external serial clock input to the SCLK00 pin is sampled internally and then used.
2. It must be used within the scope of the peripheral functional characteristics that meet this condition and meet the electrical characteristics (refer to the data sheet).

Remarks 1. f_{MCK} : Operation clock frequency of the target channel

2. m: unit number (m=0) n: channel number (n=0).

(1) Register setting

Figure 13-87 Example of register setting contents for slave selection input function (SSPI00) slave transmission and reception (1/2)



Caution Data must be sent to the SIOp register settings before the master device starts the output clock.

Remarks 1. m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

2. : Fixed in slave 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 13-87 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	SSm3 ×	SSm2 ×	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

Caution Data must be sent to the SIOp register settings before the master device starts the output clock.

Remarks 1. m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

2. ☐: Fixed in slave receive mode.

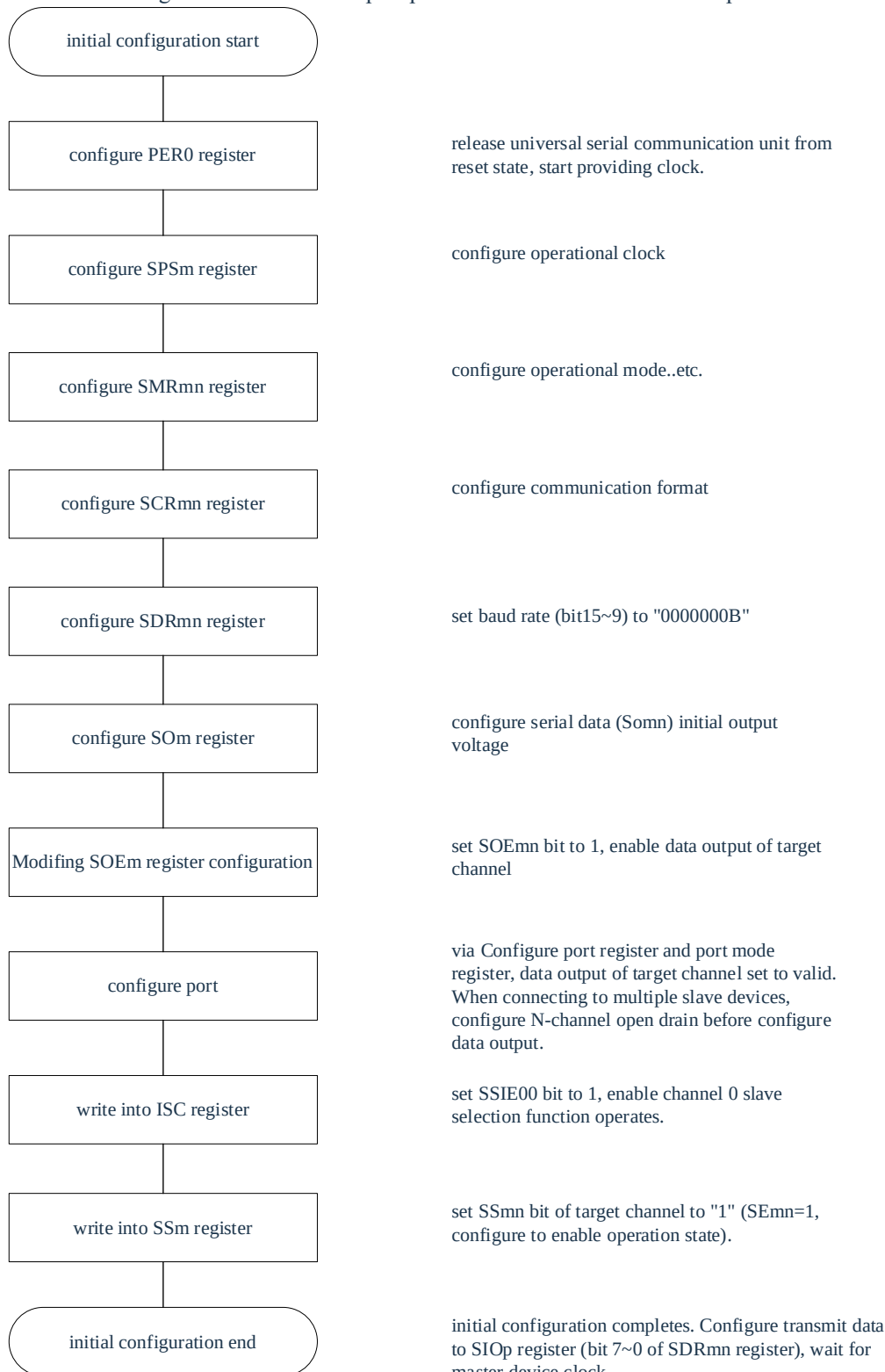
☐: Cannot be set (initial value).

×: 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) Operation steps

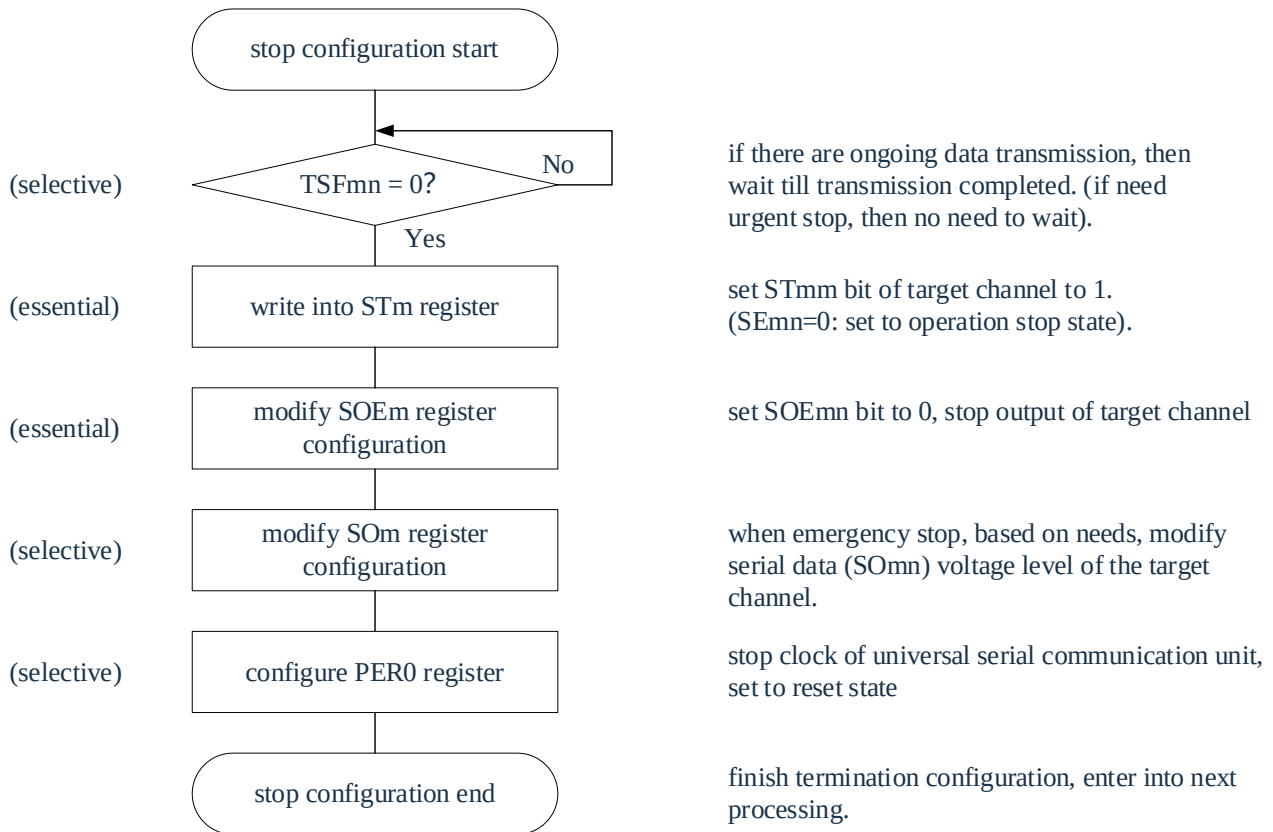
Figure 13-88 Initial setup steps for slave transmission and reception



Caution Data must be sent to the SIOp register settings before the master device starts the output clock.

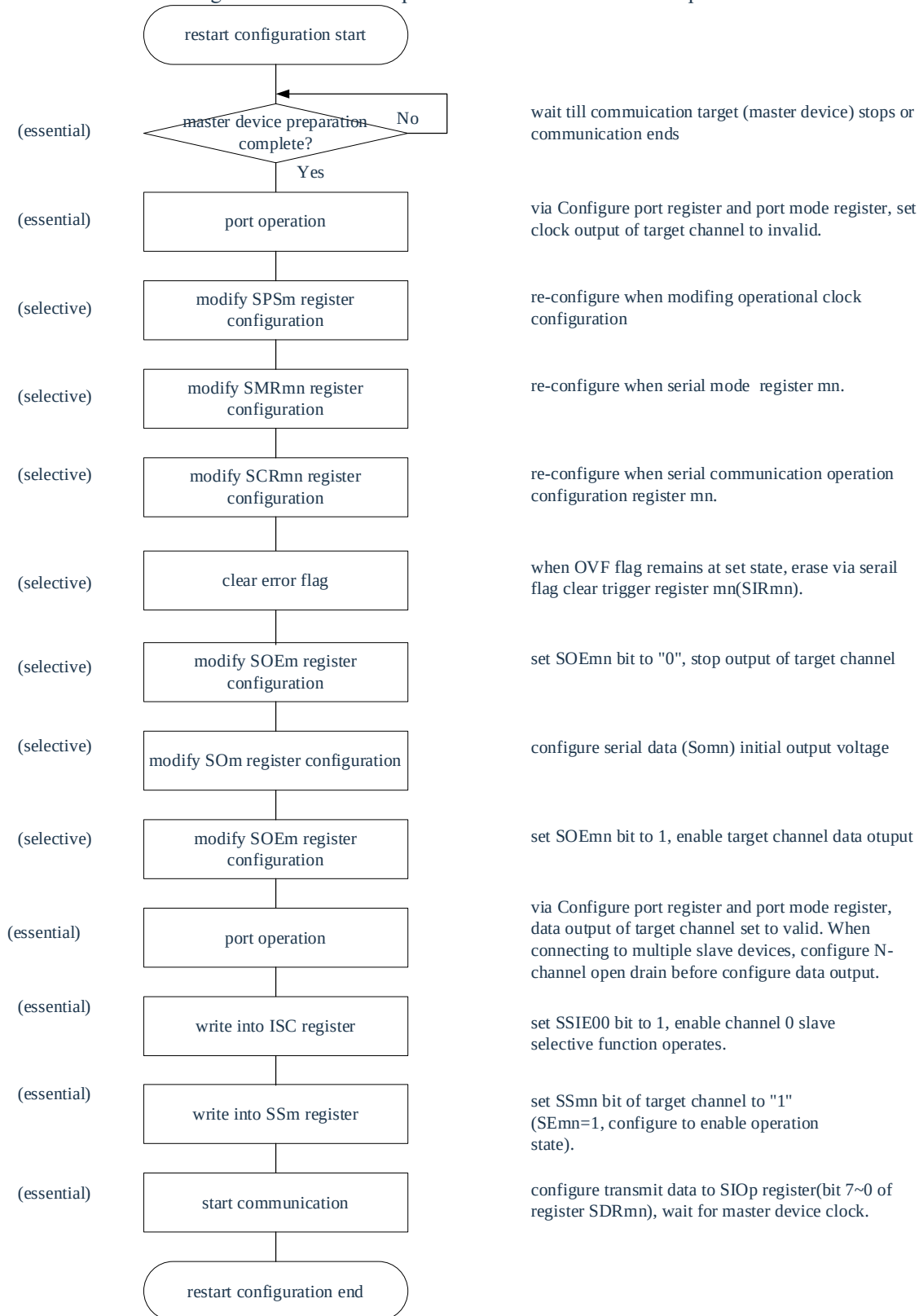
Remark m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

Figure 13-89 Stop steps of slave transmission and reception



Remark 1. m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

Figure 13-90 Restart steps of slave transmission and reception

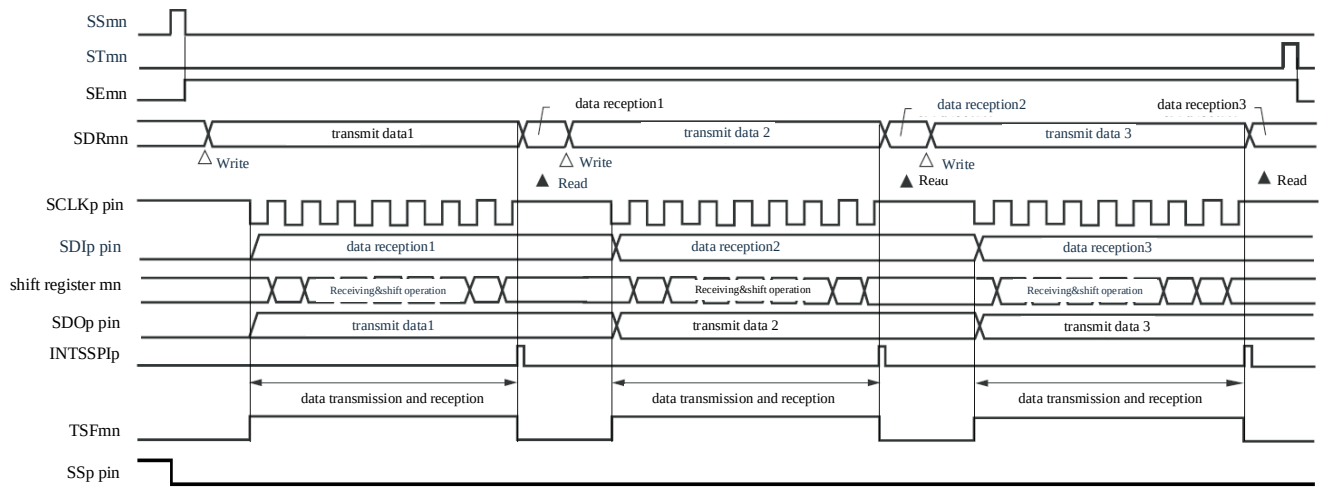


Cautions

1. Before the master device starts to output the clock, data must be sent to the SIOp register settings.
2. If you override PER0 in the abort setting to stop the clock, you must wait until the communication target (master device) stops or the communication is over to make the initial setting instead of starting the setting again.

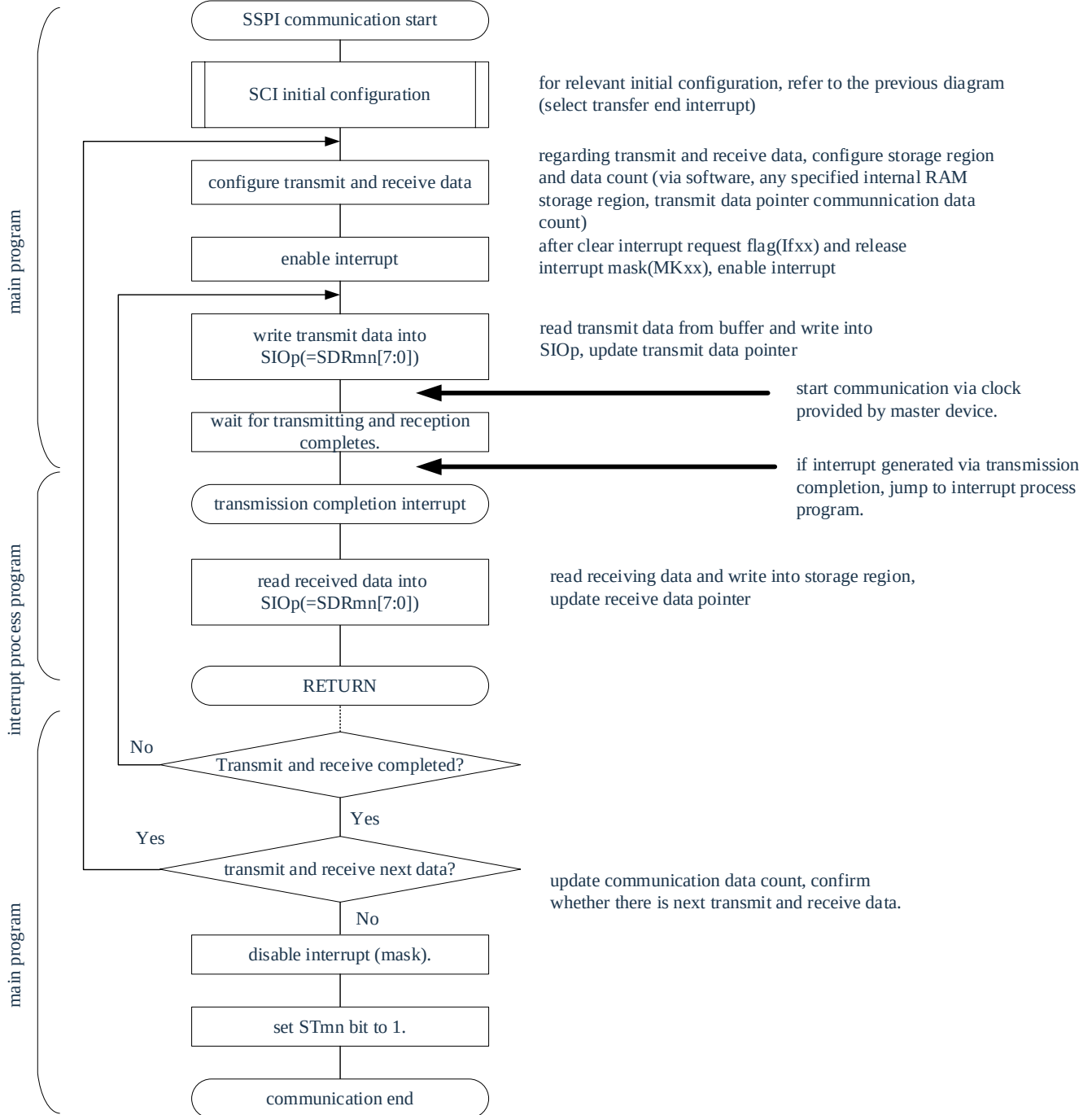
(3) Processing flow (single transmit and receive mode)

Figure 13-91 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 13-92 Flowchart of slave transmission and reception (single transmit and receive mode)

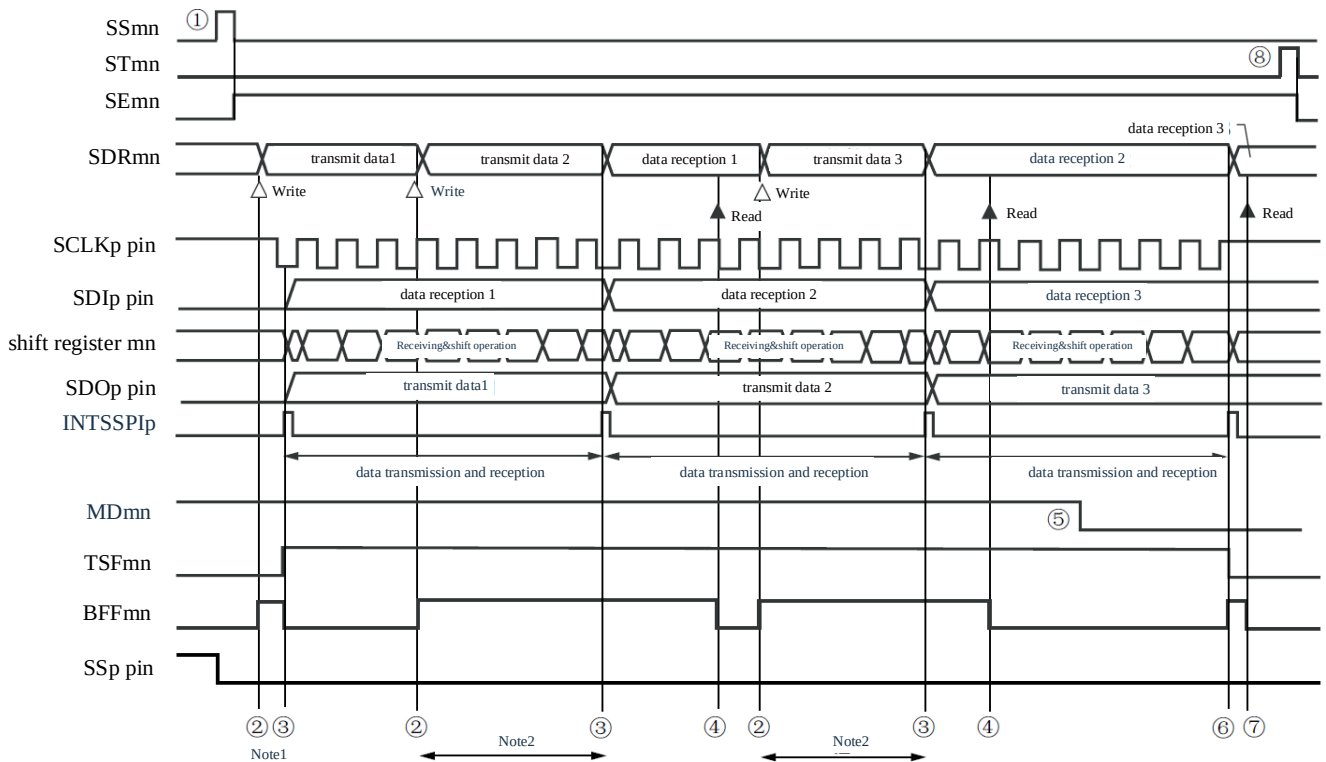


Caution Data must be sent to the SIOp register settings before the master device starts the output clock.

Remark m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

(4) Processing flow (continuous transmit and receive mode)

Figure 13-93 Timing diagram of slave transmission and reception (continuous transmit and receive mode)
(type 1: DAPmn=0, CKPmn=0)



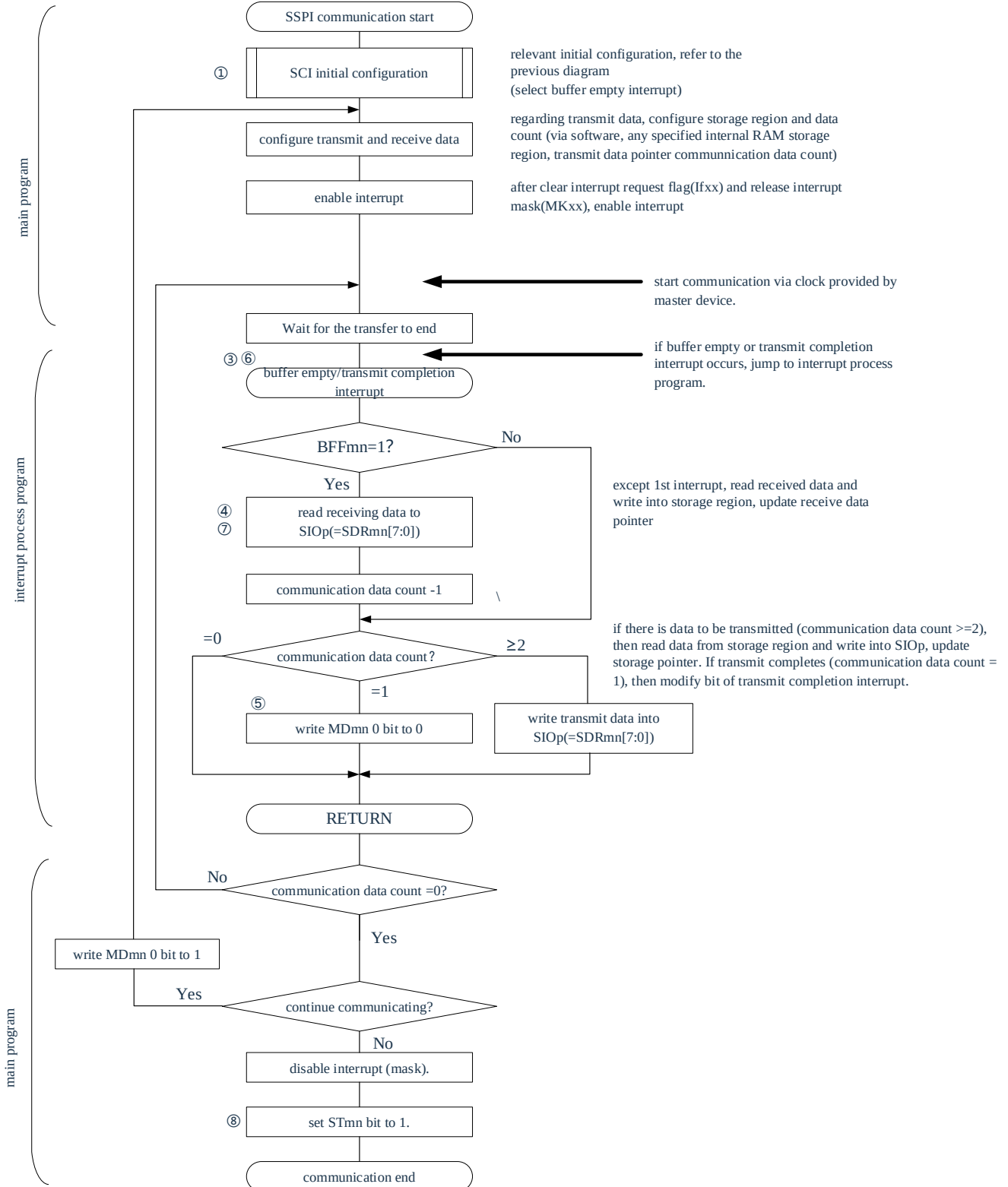
- Notes 1. If the BFFmn bit of the serial status register mn (SSRmn) is “1” (valid data is saved in the serial data register mn (SDRmn) to write the send data to the SDRmn register, and rewrite the sent data.
2. If the SDRmn register is read during this period, the transmitted data can be read. At this point, the transfer run 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.

Remarks 1. ① to ⑧ in the diagram correspond to ① to ⑧ in “Figure 13-94 Flowchart of slave transmission and reception (continuous transmit and receive mode)”.

2. m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

Figure 13-94 Flowchart of slave transmission and reception (continuous transmit and receive mode)



Caution Data must be sent to the SIOp register settings before the master device starts the output clock.

Remarks 1. ① to ⑧ correspond to ① to ⑧ in “Figure 13-93 Timing diagram of slave transmission and reception (continuous transmit and receive mode)”.

2. m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

13.6.4 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.

(1) Slave device

$$(\text{Transfer clock frequency}) = \{\text{Serial clock (SCLK) frequency provided by the master device}\}^{\text{Note}}[\text{Hz}]$$

Note The maximum enable transfer clock frequency is $f_{\text{MCK}}/6$.

Remark m: unit number (m=0) n: channel number (n=0) p: SSPI number (p=00)

Table 13-3 Slave select input function operation clock selection

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} =32MHz in operation
0	X	X	X	X	0	0	0	0	f _{CLK}	32MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	16MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	8MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	4MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	2MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	1kHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	500kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	250kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	125kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	62.5kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	31.25kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	15.63kHz
	X	X	X	X	1	1	0	0	f _{CLK} /2 ¹²	7.81kHz
	X	X	X	X	1	1	0	1	f _{CLK} /2 ¹³	3.91kHz
	X	X	X	X	1	1	1	0	f _{CLK} /2 ¹⁴	1.95kHz
	X	X	X	X	1	1	1	1	f _{CLK} /2 ¹⁵	977Hz

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.

Remarks 1. X: Ignore

2. m: unit number (m=0) n: channel number (n=0).

13.6.5 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 13-95.

Figure 13-95 Handling steps when an overflow error occurs

Software operation	Hardware status	Comments
Read the serial data register mn (SDRMN). →	The BFF m n bit of the SSRm n register is "0" and channel n is in receivable status.	This is to prevent overflow 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).

13.7 Operation of UART (UART0~UART2) Communication

This is a function that communicates asynchronously through a total of two lines: serial data transmission (TxD) and serial data reception (RxD). Using these two communication lines, data that are transmitted and received asynchronously (using internal baud rate) with other communicating parties in data frames (consisting of start bits, data, parity bits, and stop bits) are used to send and receive data. Full-duplex asynchronous UART communication can be achieved by using two channels: one dedicated for transmission (even channels) and the other dedicated for reception (odd channels).

[Transmit and receive data]

- 7-bit, 8-bit or 9-bit data length^{Note}
- 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 overflow error

[Error detection flag]

- Frame error, parity error, overflow error

Note that only UART0 supports 9-bit data length.

UART0 uses channel 0 and channel 1 of SCI0.

UART1 uses channel 2 and channel 3 of SCI0.

UART2 uses channel 0 and channel 1 of SCI1.

Each channel can select any function for use, and other functions cannot operate except for the selected one.

For example, when UART0 is used on Channel 0 and Channel 1 of Unit 0, SSPI00 and IIC01 cannot be used. However, while UART0 is in use, Channels 2 and 3 on different channels can use SSPI10, UART1, or IIC10.

Caution When used as UART, the transmitter (even channels) and receiver (odd channels) can only be used for UART.

UART has the following communication modes:

- UART transmission (see 13.7.1).
- UART reception (see 13.7.2).

13.7.1 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.

UART	UART0	UART1	UART2
Target channel	Channel 0 of SCI0	Channel 2 of SCI0	Channel 0 of SCI1
Used pin	TxD0	TxD1	TxD2
Interrupt	INTST0	INTST1	INTST2
	Transfer end interrupt (single transmit mode) or buffer empty interrupt (continuous transfer mode) can be selected		
Error detection flag	None		
Transfer data length	7-bit, 8-bit or 9-bit ^{Note 1}		
Transfer rate	$\text{Max. } f_{\text{MCK}}/6[\text{bps}](\text{SDRmn}[15:9] \geq 2)$, $\text{Min. } f_{\text{CLK}}/(2 \times 2^{15} \times 128)[\text{bps}]$ ^{Note2}		
Data phase	Non-reverse output (default: high). Reverse output (default: low).		
Parity bits	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 first or LSB first		

Notes 1. Only UART0 supports 9-bit data length.

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).

Remarks 1. f_{MCK} : Operating clock frequency of the target channel

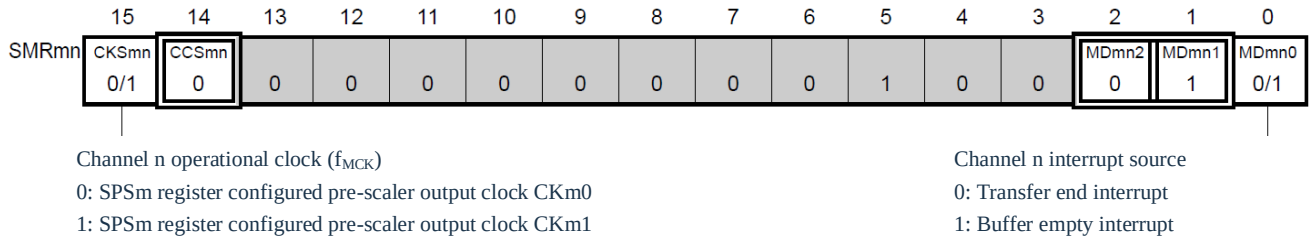
f_{CLK} : System clock frequency

2. m: unit number (m=0, 1) n: channel number (n=0, 2) mn=00, 02, 10.

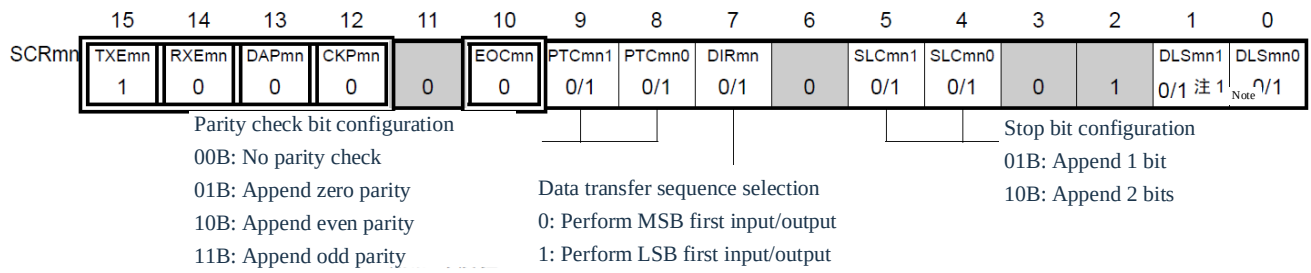
(1) Register setting

Figure 13-96 Example of register settings when UART is sent by UART (UART0~UART2) (1/2)

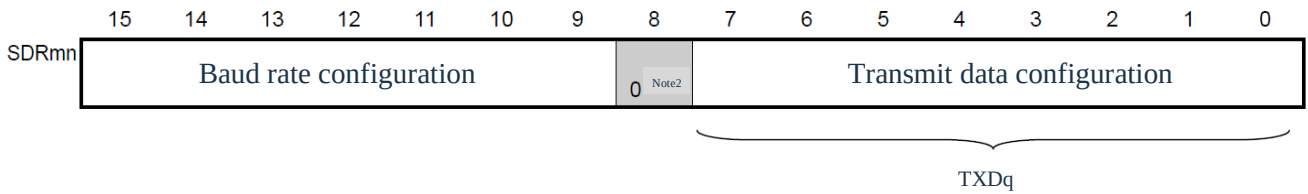
(a) Serial mode register mn (SMRmn)



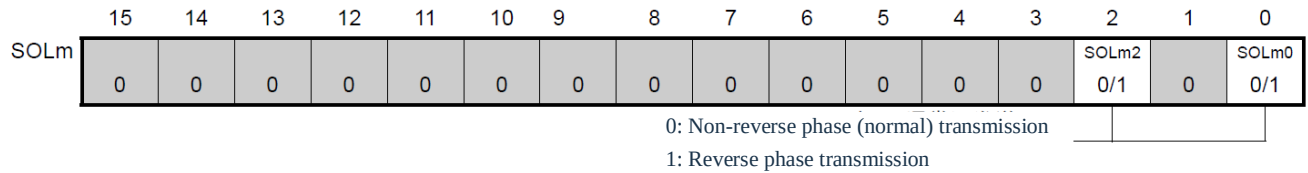
(b) Serial communication operation configuration register mn (SCRmn)



(c) Serial data register mn (SDRmn) (low 8 bits: TXDq)



(d) Serial output voltage register m (SOLm) Only configure target channel bits.



Notes 1. Limited to SCR00 registers, other fixed as "1".

2. When communicating with a 9-bit data length, bits 0 to 8 of the SDRm0 register are used for setting the transmitted data. Only UART0 can perform 9-bit data length communication.

Remarks 1.m: unit number (m=0, 1) n: channel number (n=0, 2) q: UART numbers (q=0~2) mn=00, 02, 10

2. : Fixed in UART transmit 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 13-96 UART Example of register settings when UART is sent by UART (UART0~UART2) (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	CKOm3	CKOm2	CKOm1	CKOm0	0	0	0	0	SOM3	SOM2	SOM1	SOM0
	0	0	0	0	×	×	×	×	0	0	0	0	×	0/1 ^{Note}	×	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	SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1	×	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	SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1	×	0/1

Note 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.

Remarks 1. m: unit number (m=0, 1) n: channel number (n=0, 2) q: UART numbers (q=0~2)mn=00, 02, 10

2. ☐ Fixed in UART transmit 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) Operation steps

Figure 13-97 Initial setup steps for UART transmission

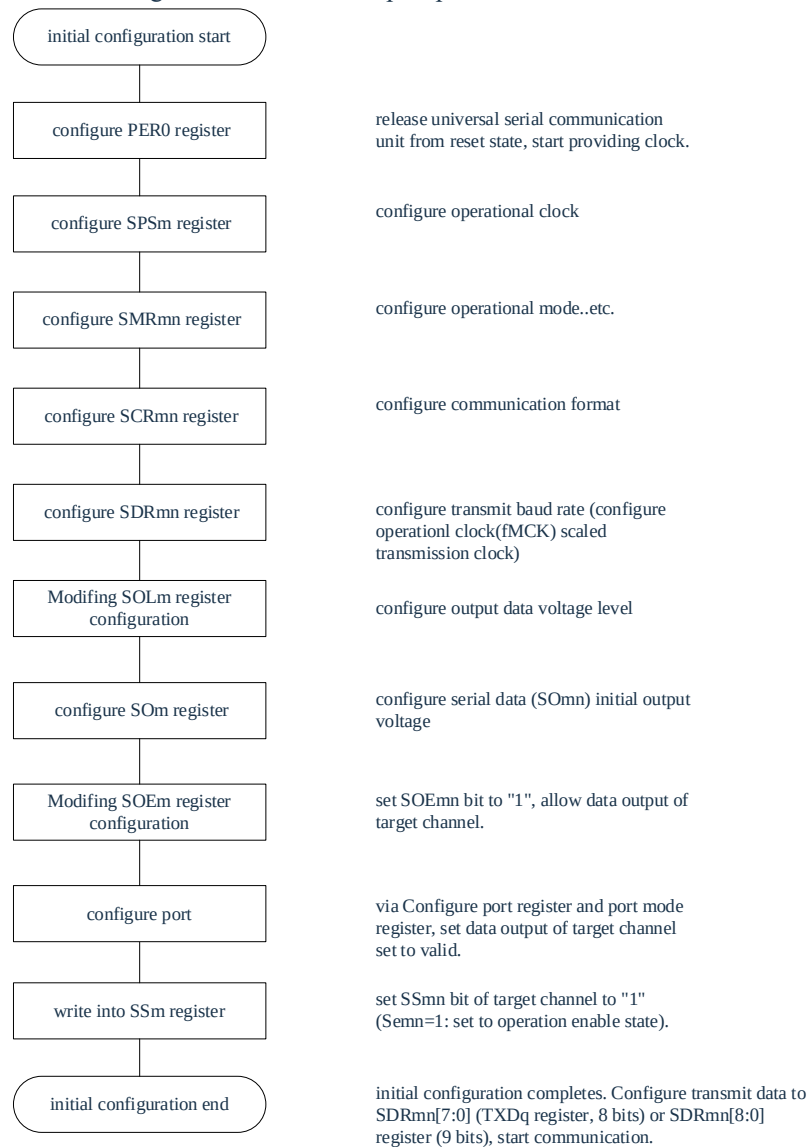


Figure 13-98 Stop steps for UART transmission

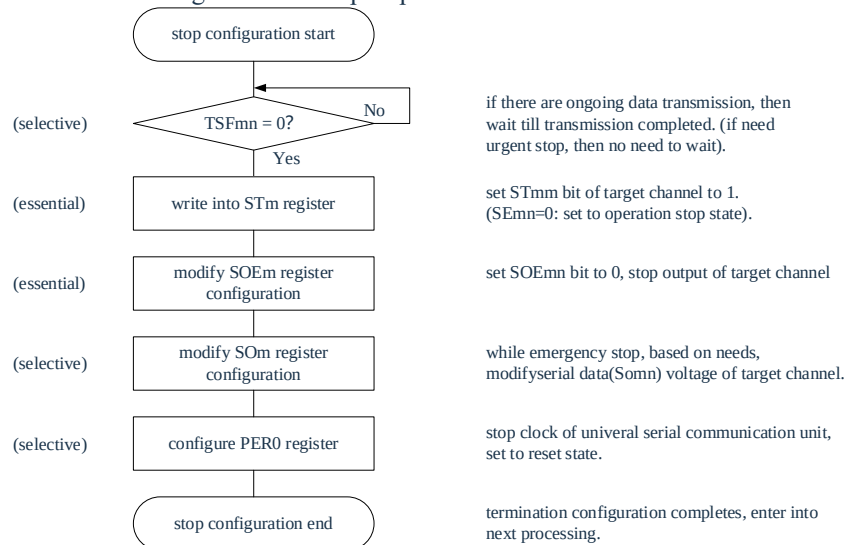
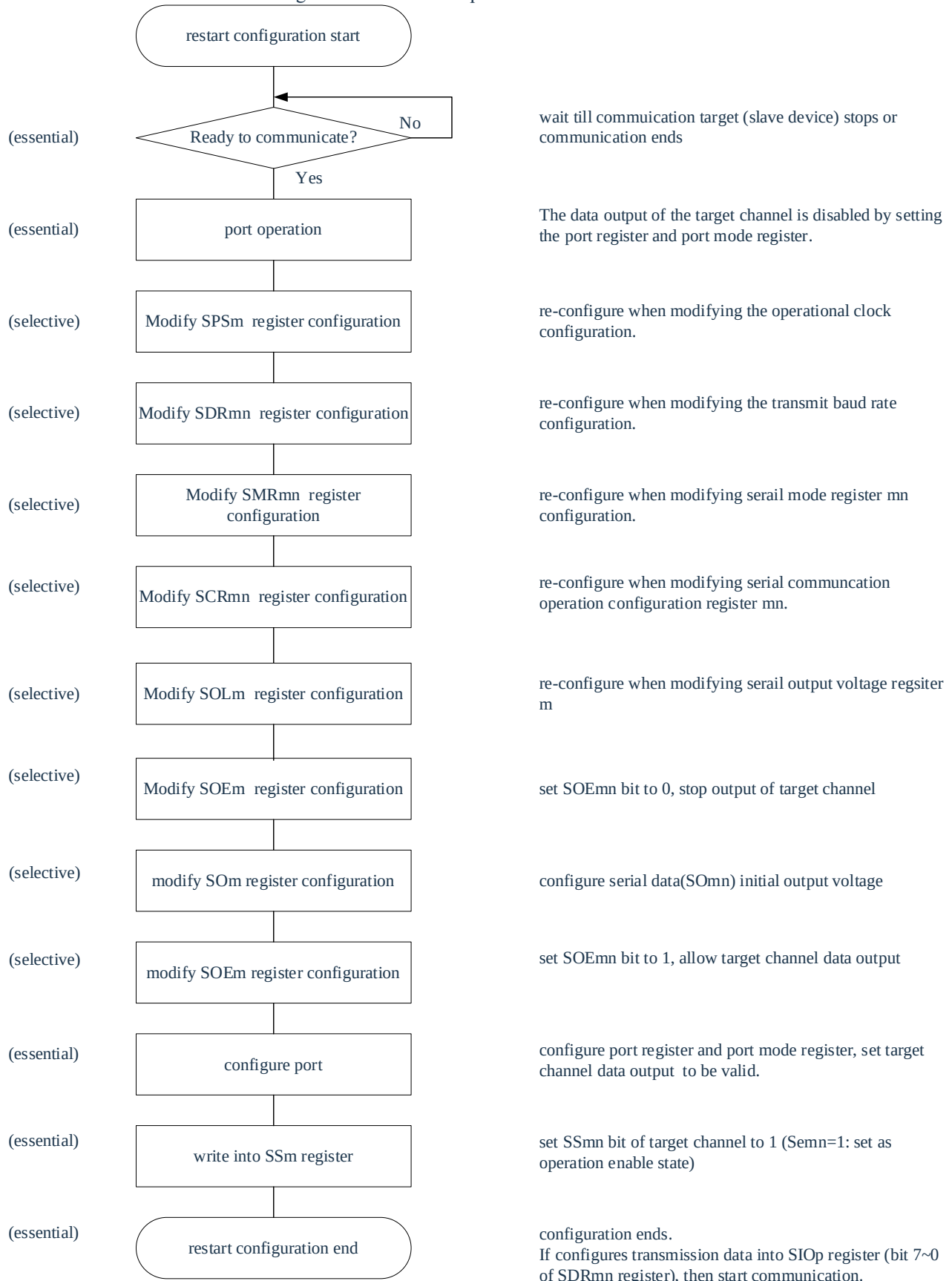


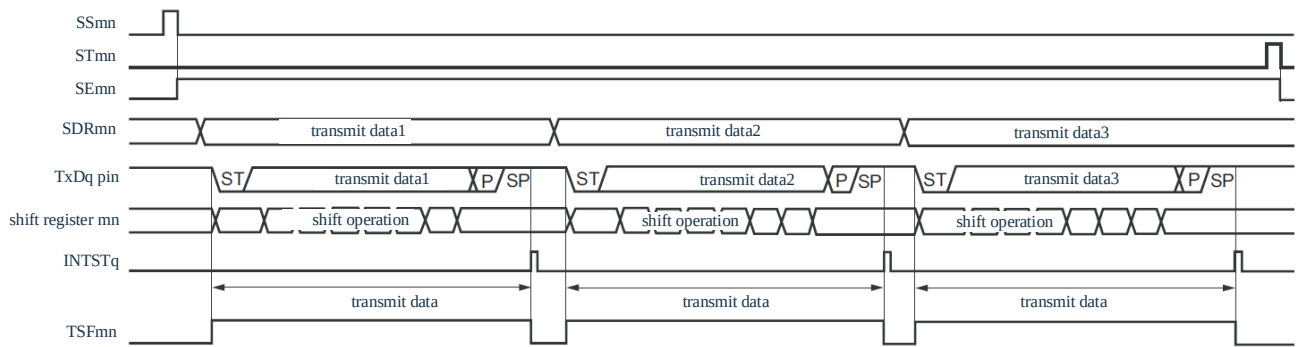
Figure 13-99 Restart steps for UART transmission



Remark 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.

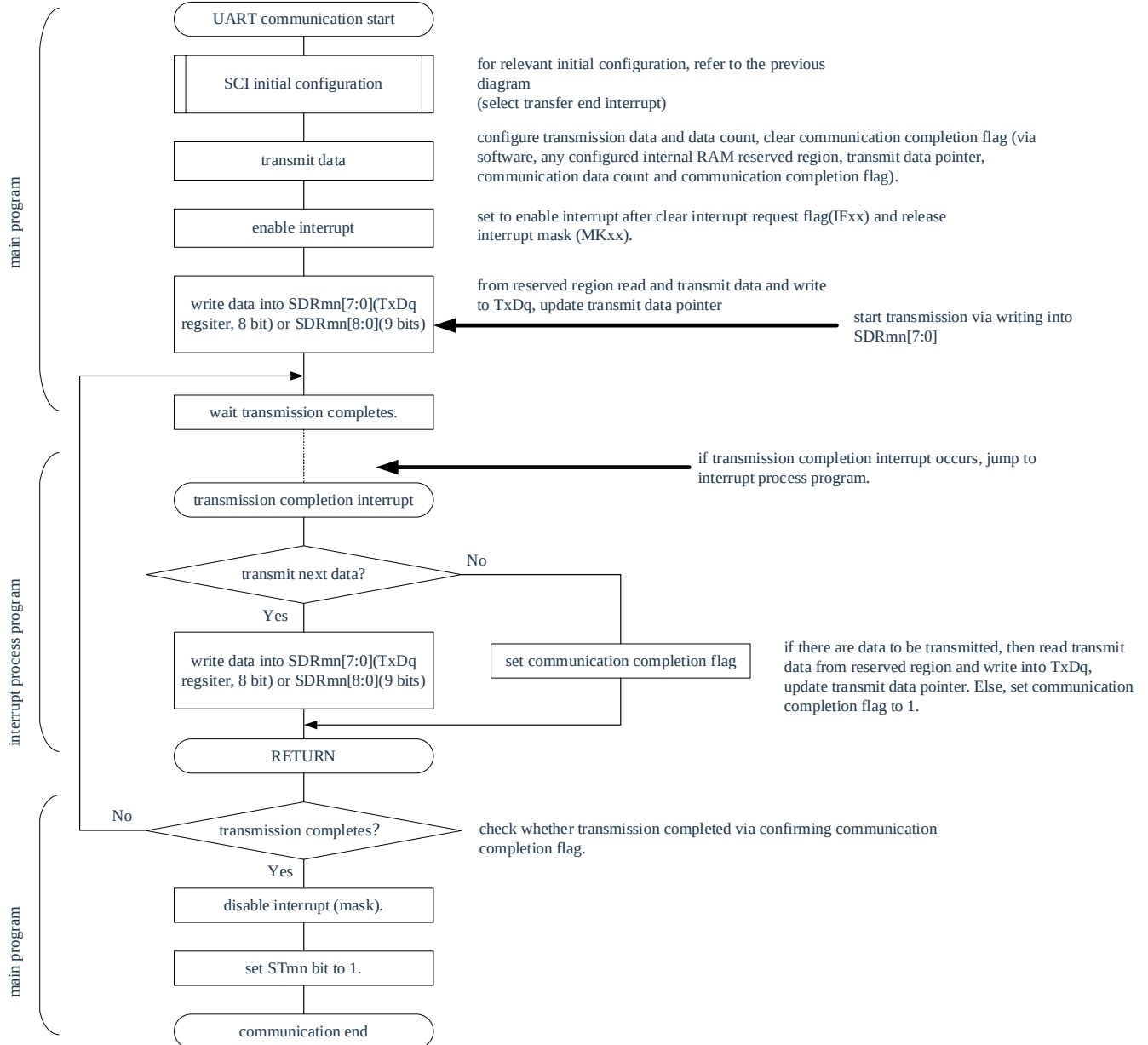
(3) Processing flow (single transmit mode)

Figure 13-100 Timing diagram of UART transmission (single transmit mode)



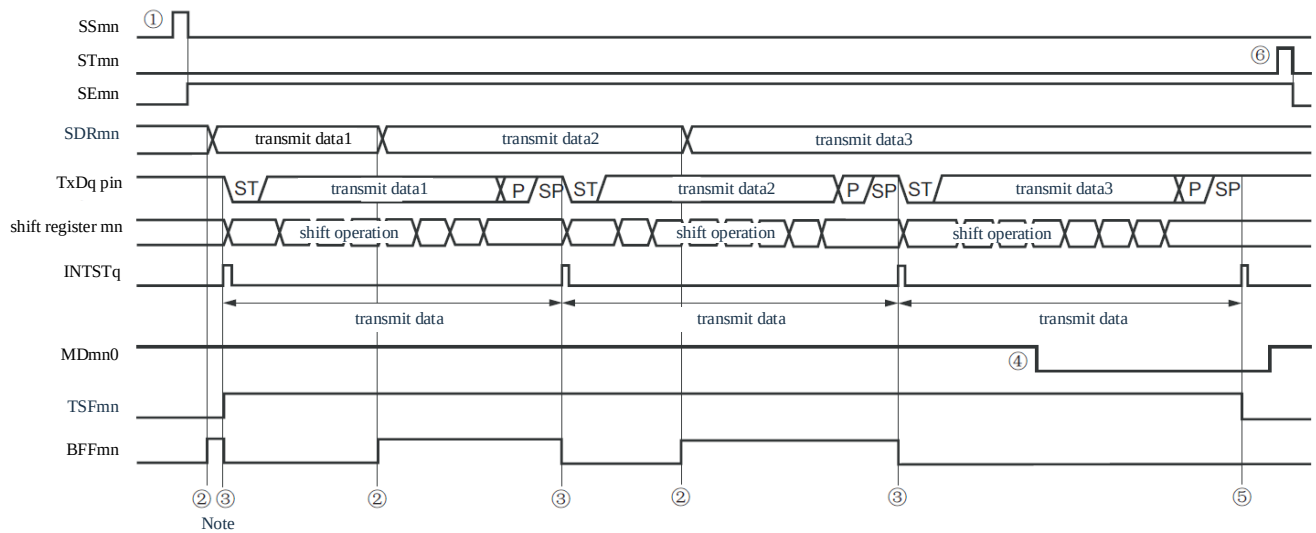
Remark m: unit number (m=0, 1) n: channel number (n=0, 2) q: UART numbers (q=0~2) mn=00, 02, 10

Figure 13-101 Flowchart of UART transmission (single transmit mode)



(4) Processing flow (continuous transmit mode)

Figure 13-102 Timing diagram of UART transmission (continuous transmit mode)

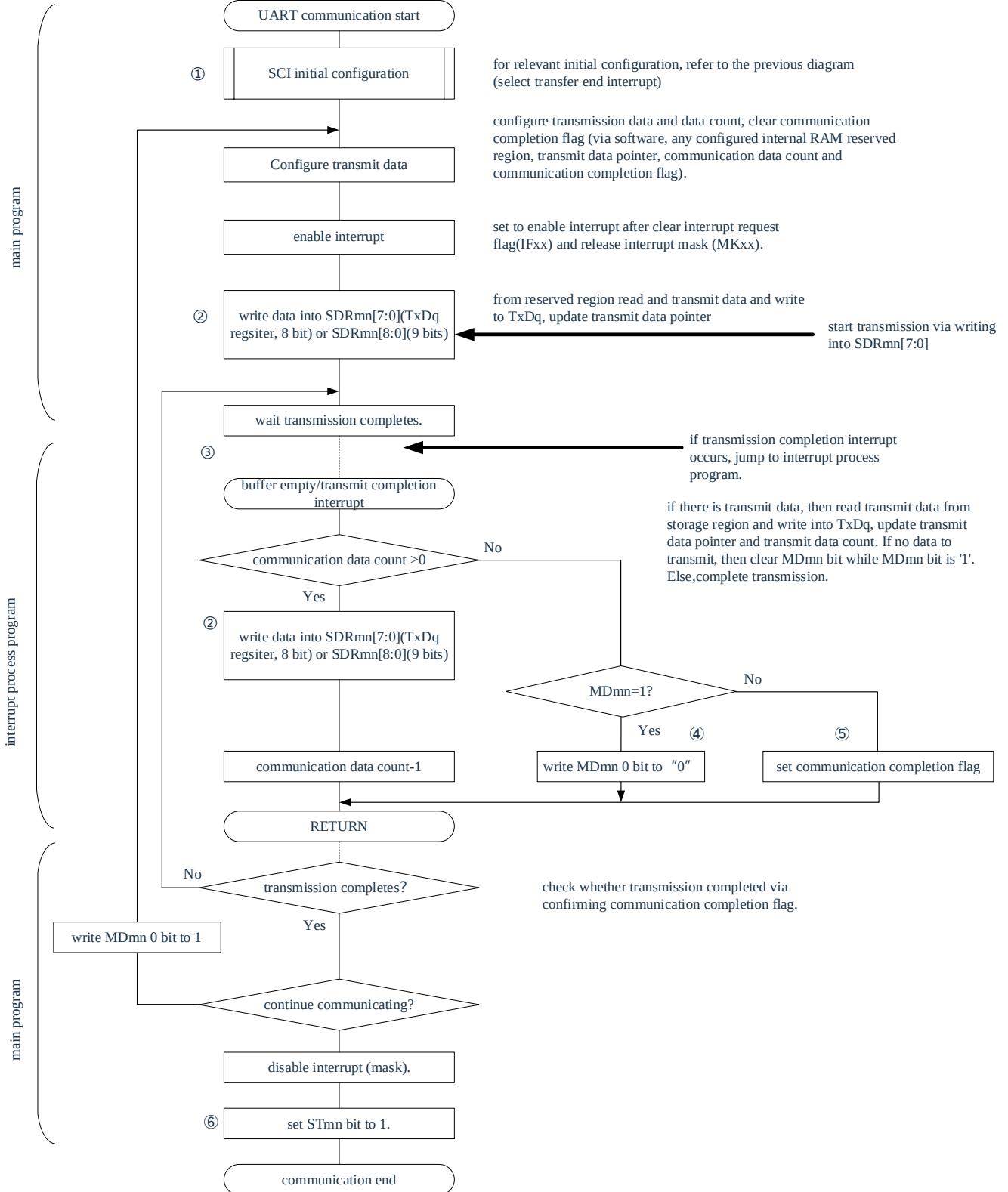


Note If the BFFmn bit of the serial status register mn (SSRmn) is “1” (when valid data is saved in the serial data register mn (SDRmn)) is given The SDRmn register writes the transmitted data and overrides the transmitted data.

Caution The MDmn0 bit of the serial mode register mn (SMRmn) can be overridden even during operation. However, 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 m: unit number (m=0, 1) n: channel number (n=0, 2) q: UART numbers (q=0~2) mn=00, 02, 10

Figure 13-103 Flowchart of UART transmission (continuous transmit mode)



Remark ① to ⑥ in the figure correspond to ① to ⑥ in “Figure 13-102 Timing diagram of UART transmission (continuous transmit mode)”.

13.7.2 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.

UART	UART0	UART1	UART2
Target channel	Channel 1 of SCI0	Channel 3 of SCI0	Channel 1 of SCI1
Used pin	RxD0	RxD1	RxD2
Interrupt	INTSR0	INTSR1	INTSR2
	Limited to transfer end interrupts (disable setting buffer empty interrupts).		
Error interrupt	INTSRE0	INTSRE1	INTSRE2
Error detection flag	• Frame error detection flag (FEFmn). • Parity error detection flag (PEFmn). • Overflow error detection flag (OVFmn).		
Transfer data length	7-bit, 8-bit or 9-bit ^{Note 1}		
Transfer rate	Max. $f_{MCK}/6[\text{bps}](\text{SDRmn}[15:9] \geq 2)$, Min. $f_{CLK}/(2 \times 2^{15} \times 128)[\text{bps}]$		
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 first or LSB first		

Notes 1. Only UART0 supports 9-bit data length.

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).

Remarks 1. f_{MCK} : The operating clock frequency of the target channel

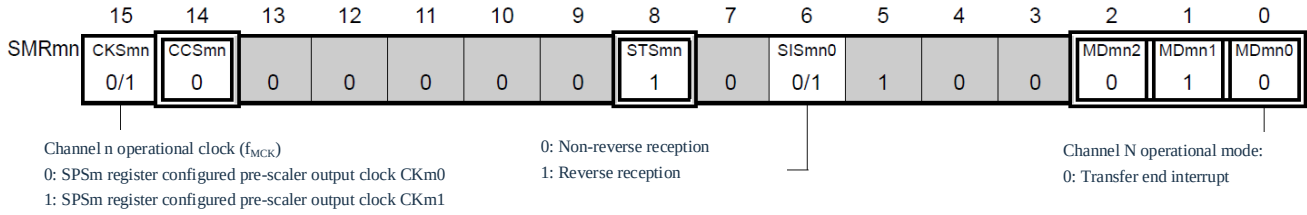
f_{CLK} : System clock frequency

2. m: unit number (m=0, 1) n: channel number (n=1, 3) mn=01, 03, 11.

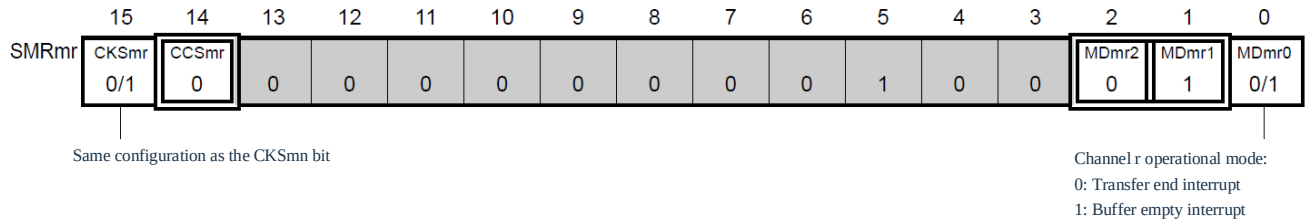
(1) Register setting

Figure 13-104 Example of register settings when UART is received by UART (UART0~UART2) (1/2)

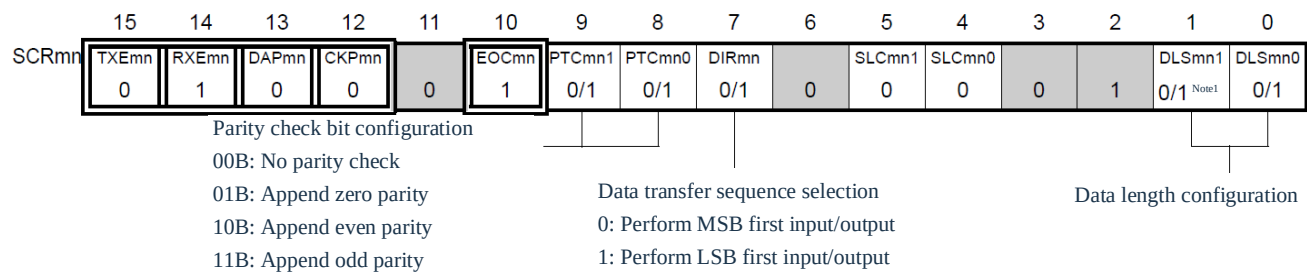
(a) Serial mode register mn (SMRmn)



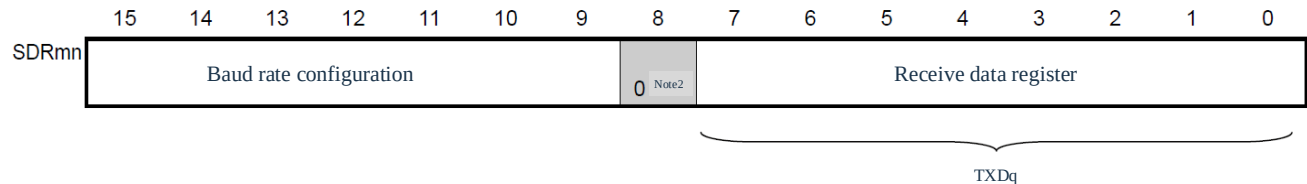
(b) Serial mode register mr(SMRmr)



(c) Serial communication operation configuration register mn (SCRmn)



(d) Serial data register mn (SDRmn) (low 8 bits: TXDq)



Notes 1. Limited to SCR01 registers, other fixed as “1”.

2. When communicating with a 9-bit data length, bits 0 to 8 of the SDRm1 register are used for setting the transmitted data.
Only UART0 can perform 9-bit data length communication.

Caution When the UART is received, the SMRmr register of channel r paired with channel n must also be set.

Remarks 1. m: unit number (m=0, 1) n: channel number (n=1, 3) mn=01, 03, 11.

r: Channel number (r=n-1) q: UART number (q=0~2)

2. ☐: Fixed in UART 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 13-104 Example of register settings when UART is received by UART (UART0~UART2) (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					CKOm3	CKOm2	CKOm1	CKOm0					SOM3	SOM2	SOM1	SOM0
	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													SOEm3	SOEm2	SOEm1	SOEm0
	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													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	0/1	×

Remarks 1. m: unit number (m=0, 1).

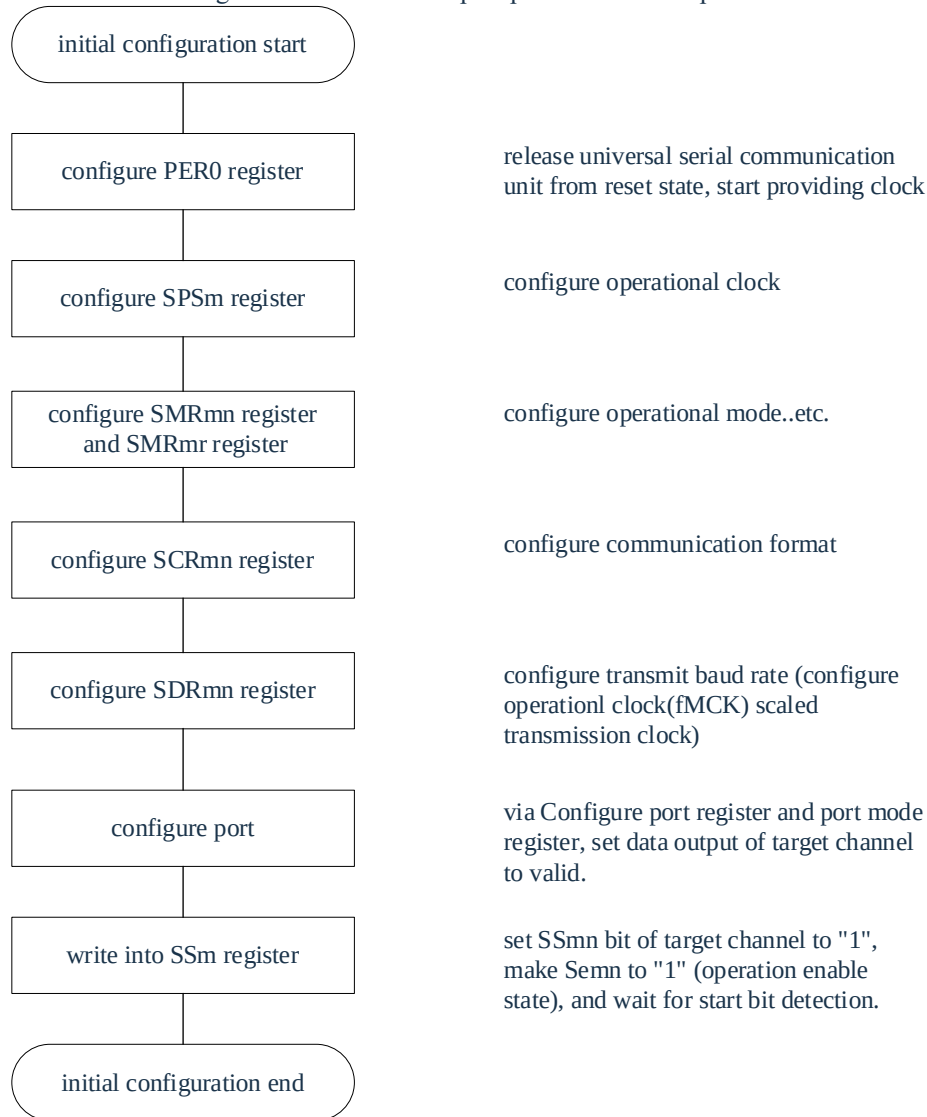
2. : Fixed in UART receive mode. : Cannot be set (initial value).

×: 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) Operation steps

Figure 13-105 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 13-106 Stop steps for UART reception

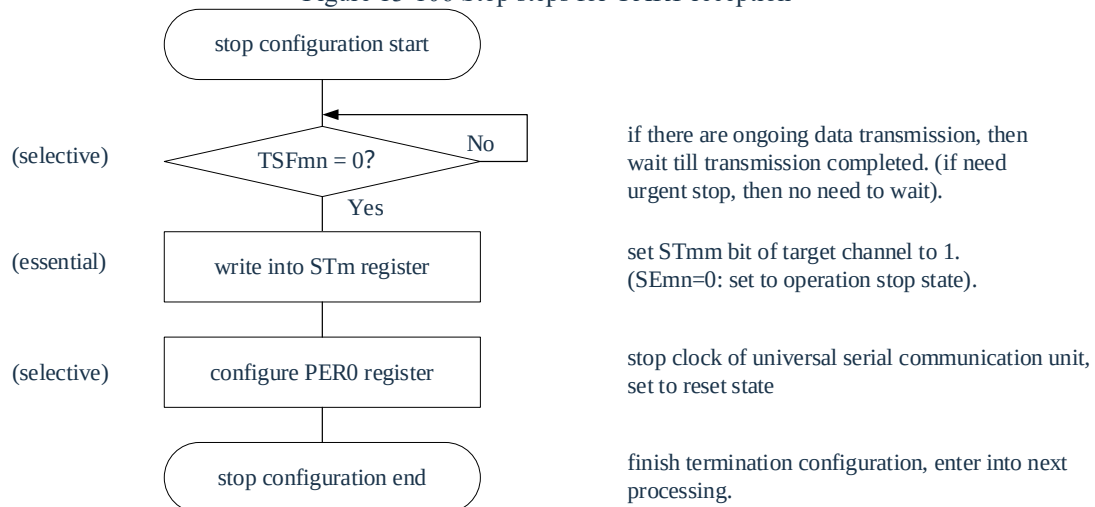
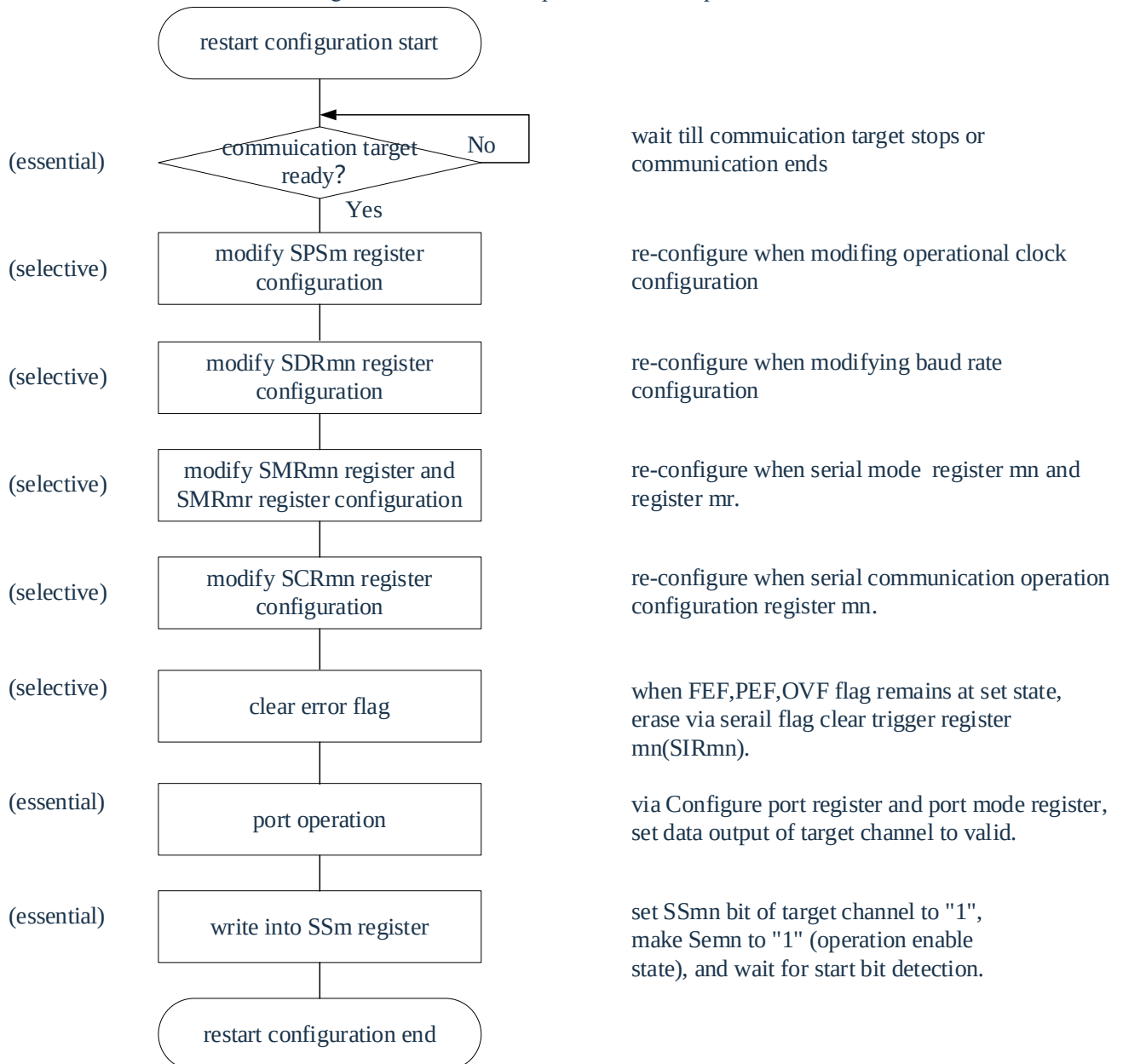


Figure 13-107 Restart steps for UART reception

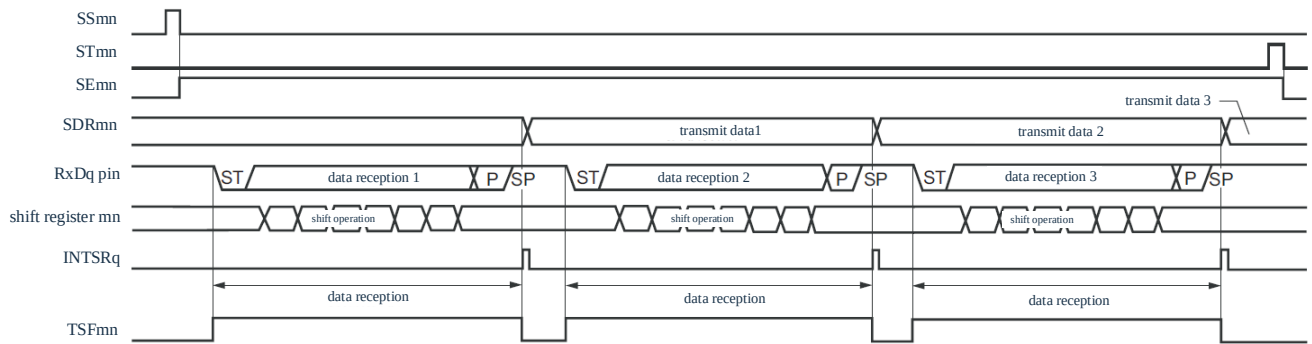


Caution At least 4 F_{MCK} clocks must be spaced after setting the RXEmn bit of the SCRmn register to "1" and then set the SSmn bit to "1".

Remark 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

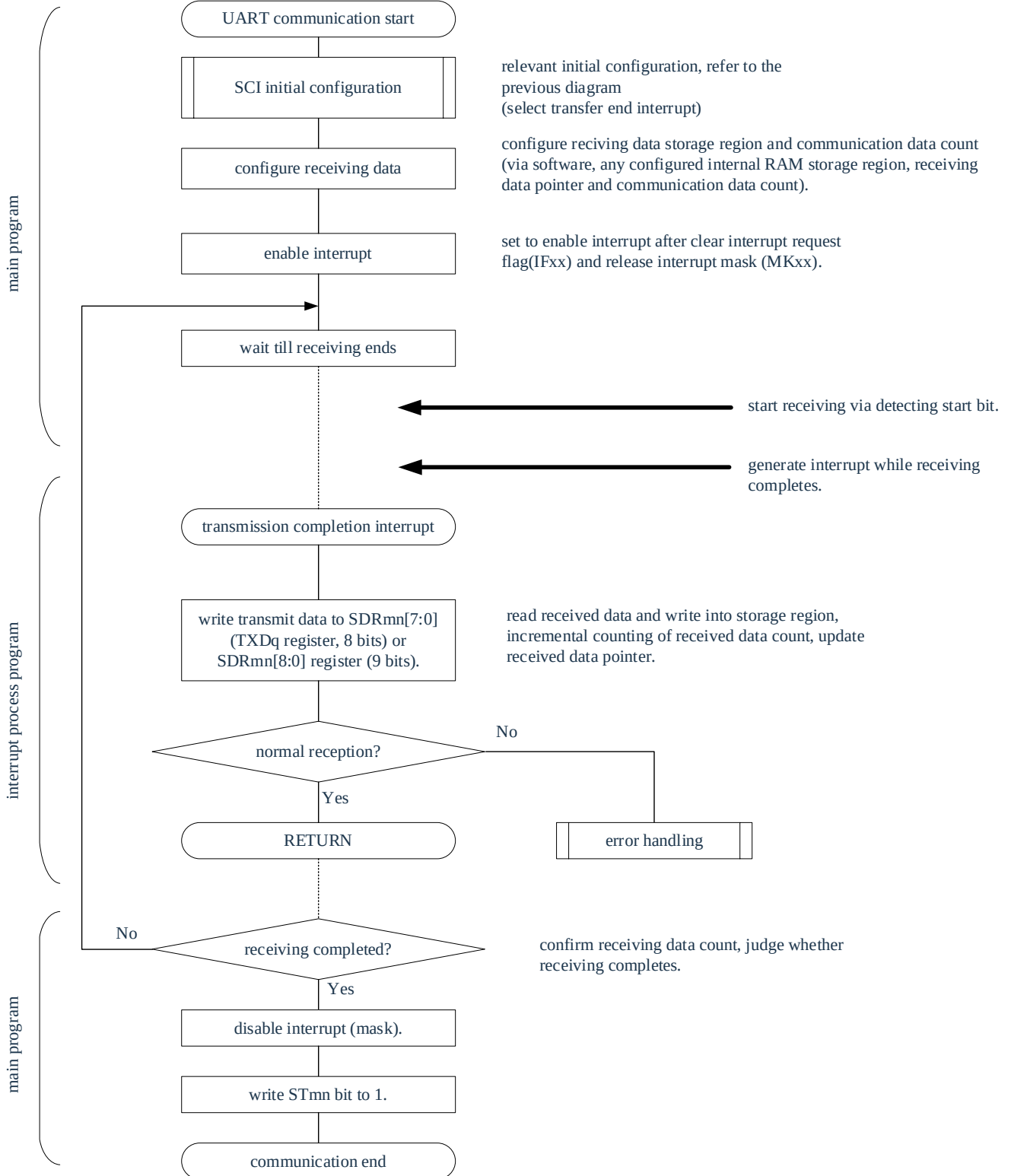
Figure 13-108 Timing diagram of UART reception



Remark m: unit number (m=0, 1) n: channel number (n=1, 3) mn=01, 03, 11.

r: Channel number (r=n-1) q: UART number (q=0~2)

Figure 13-109 Flowchart of UART reception



13.7.3 Calculation of Baud Rate

(1) Calculating the baud rate

The baud rate of UART (UART0~UART2) communication can be calculated using the following formula:

$$(\text{Baud rate}) = \{\text{Clock frequency of the target channel (f}_{\text{MCK}})\} \div (\text{SDRmn}[15:9] + 1) \div 2 [\text{bps}]$$

Caution The SDRmn [15:9] of the serial data register mn (SDRmn) is disabled from being set to “0000000B” and “0000001B”.

Remark 1. When using UART, the value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000010B to 1111111B), so it ranges from 2 to 127.

2. m: unit number (m=0, 1) n: channel number (n=0~2). mn=00~03, 10~11.

The operating clock (f_{MCK}) is determined by the serial clock select register m (SPSm) and bit 15 (CKSmn bit) of the serial mode register mn (SMRmn).

Table 13-4 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} =32MHz in operation
0	X	X	X	X	0	0	0	0	f _{CLK}	32MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	16MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	8MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	4MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	2MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	1MHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	500kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	250kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	125kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	62.5kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	31.25kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	15.63kHz
	X	X	X	X	1	1	0	0	f _{CLK} /2 ¹²	7.81kHz
	X	X	X	X	1	1	0	1	f _{CLK} /2 ¹³	3.91kHz
	X	X	X	X	1	1	1	0	f _{CLK} /2 ¹⁴	1.95kHz
	X	X	X	X	1	1	1	1	f _{CLK} /2 ¹⁵	977Hz
1	0	0	0	0	X	X	X	X	f _{CLK}	32MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	16MHz
	0	0	1	0	X	X	X	X	f _{CLK} /2 ²	8MHz
	0	0	1	1	X	X	X	X	f _{CLK} /2 ³	4MHz
	0	1	0	0	X	X	X	X	f _{CLK} /2 ⁴	2MHz
	0	1	0	1	X	X	X	X	f _{CLK} /2 ⁵	1MHz
	0	1	1	0	X	X	X	X	f _{CLK} /2 ⁶	500kHz
	0	1	1	1	X	X	X	X	f _{CLK} /2 ⁷	250kHz
	1	0	0	0	X	X	X	X	f _{CLK} /2 ⁸	125kHz
	1	0	0	1	X	X	X	X	f _{CLK} /2 ⁹	62.5kHz
	1	0	1	0	X	X	X	X	f _{CLK} /2 ¹⁰	31.25kHz
	1	0	1	1	X	X	X	X	f _{CLK} /2 ¹¹	15.63kHz
	1	1	0	0	X	X	X	X	f _{CLK} /2 ¹²	7.81kHz
	1	1	0	1	X	X	X	X	f _{CLK} /2 ¹³	3.91kHz
	1	1	1	0	X	X	X	X	f _{CLK} /2 ¹⁴	1.95kHz
	1	1	1	1	X	X	X	X	f _{CLK} /2 ¹⁵	977Hz

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.

Remarks 1. X: Ignore

2. m: unit number (m=0, 1) n: channel number (n=0~2) mn=00~03, 10~11.

(2) Baud rate error during transmission

The baud rate error when transmitting UART (UART0 to UART2) communication can be calculated using the following formula, and the baud rate of the transmitter must be set within the baud rate tolerance of the receiver.

$$(\text{Baud rate error}) = (\text{calculated value of baud rate}) \div (\text{value of target baud rate}) \times 100 - 100[\%]$$

An example of setting the UART baud rate at $f_{\text{CLK}}=32\text{MHz}$ is shown below.

UART baud rate (Target baud rate)	$f_{\text{CLK}}=32\text{MHz}$			
	Operation clock (f_{MCK})	SDRmn[15:9]	Calculated value of the baud rate	Error with target baud rate
300bps	$f_{\text{CLK}}/2^9$	103	300.48bps	+0.16%
600bps	$f_{\text{CLK}}/2^8$	103	600.96bps	+0.16%
1200bps	$f_{\text{CLK}}/2^7$	103	1201.92bps	+0.16%
2400bps	$f_{\text{CLK}}/2^6$	103	2403.85bps	+0.16%
4800bps	$f_{\text{CLK}}/2^5$	103	4807.69bps	+0.16%
9600bps	$f_{\text{CLK}}/2^4$	103	9615.38bps	+0.16%
19200bps	$f_{\text{CLK}}/2^3$	103	19230.8bps	+0.16%
31250bps	$f_{\text{CLK}}/2^3$	63	31250.0bps	±0.0%
38400bps	$f_{\text{CLK}}/2^2$	103	38461.5bps	+0.16%
76800bps	$f_{\text{CLK}}/2$	103	76923.1bps	+0.16%
153600bps	f_{CLK}	103	153846bps	+0.16%
312500bps	f_{CLK}	50	313725bps	±0.39%

Remark m: unit number (m=0, 1) n: channel number (n=0, 2) mn=00, 02, 10.

(3) Baud rate tolerance range for reception

The baud rate tolerance range for UART (UART0 to UART2) 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 13.7.4 (1) Baud rate calculation formula).

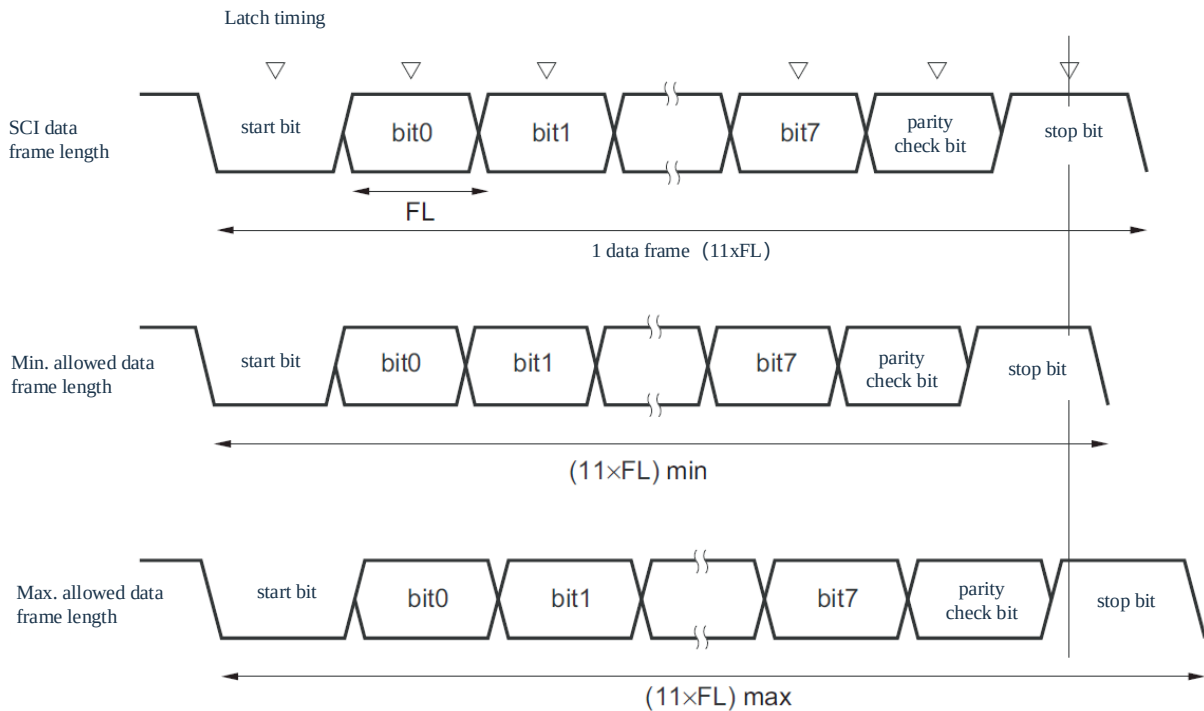
k: $\text{SDRmn}[15:9] + 1$

Nfr: 1 frame length of data [bit]

= (start bit) + (data length) + (parity bit) + (stop bit)

Remark m: unit number (m=0, 1) n: channel number (n=1, 3) mn=01, 03, 11

Figure 13-110 Baud rate tolerance range for reception (1 data frame length = 11 bits)



As shown in Figure 13-110 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.

13.7.4 Handling Steps When an Error Occurs During UART (UART0~UART2) Communication

The handling steps when an error occurs during UART (UART0~UART 2) communication are shown in Figure 13-111 and Figure 13-112.

Figure 13-111 Processing steps when a parity error or overflow 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 overflow 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.

Figure 13-112 Processing steps when a frame error occurs

Software operation	Hardware status	Remark
Read the serial data register mn(SDRMN).	The BFF m n bit of the SSRm n register is "0" and channel n is in receivable status.	This is to prevent overflow 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, 1) n: channel number (n=0~3) mn=00~03, 10~11.

13.8 Operation of LIN Communication

13.8.1 LIN Transmission

In UART transmission, UART0 supports LIN communication.

The LIN transmission uses channel 0 of unit 0.

UART	UART0	UART1	UART2
LIN communication support	Yes	No	No
Target channel	Channel 0 of SCI0	—	—
Used pin	TxD0	—	—
Interrupt	INTST0	—	—
	Selectable transfer end interrupt (single transfer mode) or buffer air interrupt (continuous transfer mode).		
Error detection flag	None		
Transfer data length	8 bits		
Transfer rate ^{Note}	Max. $f_{MCK}/6$ [bps](SDR00[15:9] ≥ 2), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]		
Data phase	Non-reverse output (default: high). Reverse output (default: low).		
Parity bit	No parity bits.		
Stop bit	Append 1 bit.		
Data direction	LSB first		

Note It must be used within the range of peripheral functional characteristics that meet this condition and meet the electrical characteristics (refer to the data sheet), and 2.4/9.6/19.2kbps are often used in LIN communication.

Remark f_{MCK} : Operation clock frequency of the target channel

f_{CLK} : System clock frequency

LIN is short for Local Interconnect Network and is a low-speed (1 to 20kbps) serial communication protocol to reduce the cost of automotive networks.

LIN communication is a single master communication, a master device can connect up to 15 slave devices.

LIN slave devices are used for the control of switches, transmissions, sensors, etc., which are connected to the master control device via LIN.

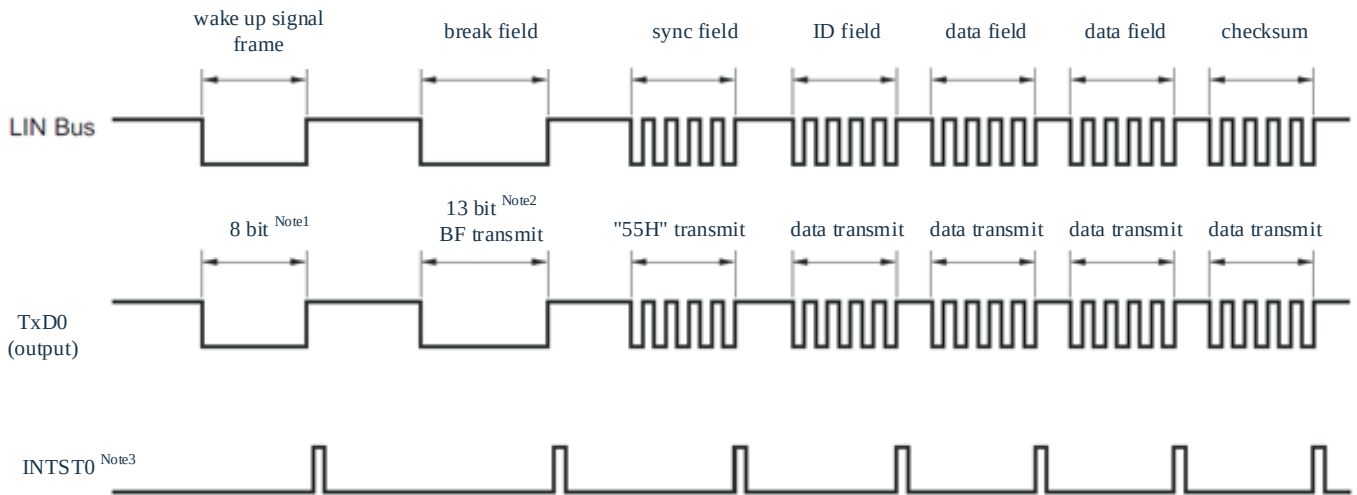
The LIN master is generally connected to networks such as the Controller Area Network.

The LIN bus is a single-wire bus that connects nodes through an ISDO9141-compliant transceiver.

According to the LIN protocol, the master device sends a frame with additional baud rate information, and the slave device receives this frame and corrects the baud rate error with the master control device. Therefore, if the baud rate error of the slave device is not greater than $\pm 15\%$, communication can be made.

A summary of the LIN's transmit operation is shown in Figure 13-113.

Figure 13-113 LIN transmission operation



Notes

1. In order to meet the requirements of the wake-up signal, the baud rate is set and the corresponding data is transmitted by "80H".
2. The break field is specified as a 13-bit wide low-level output, so the baud rate used for the main transmission is $N[\text{bps}]$. The baud rate used for the break field is as follows:

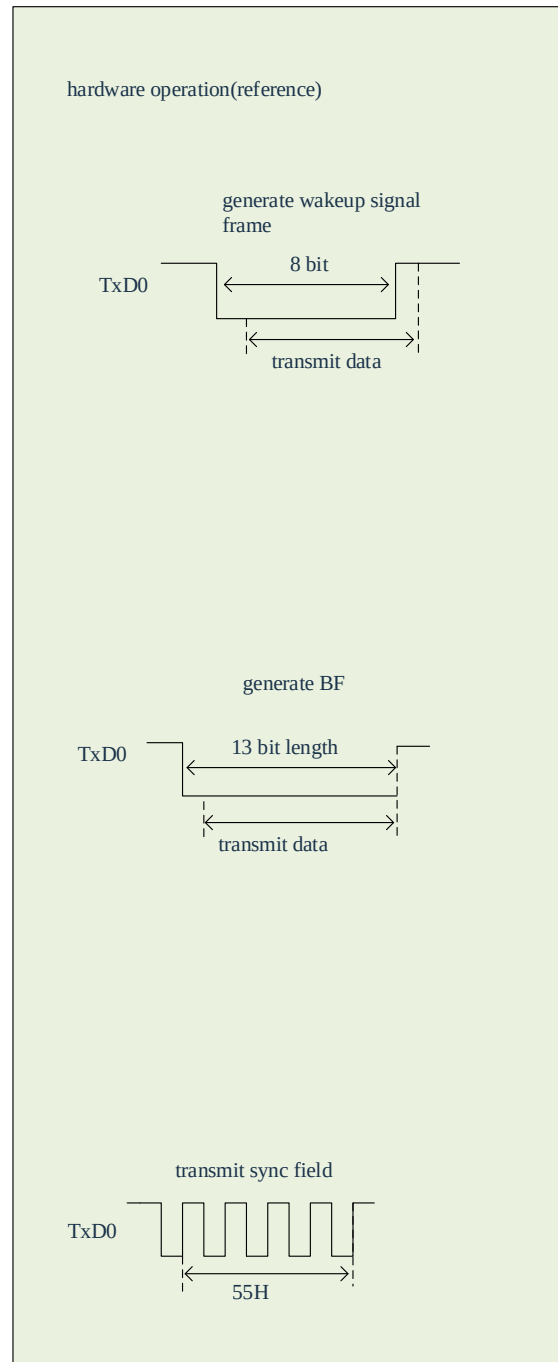
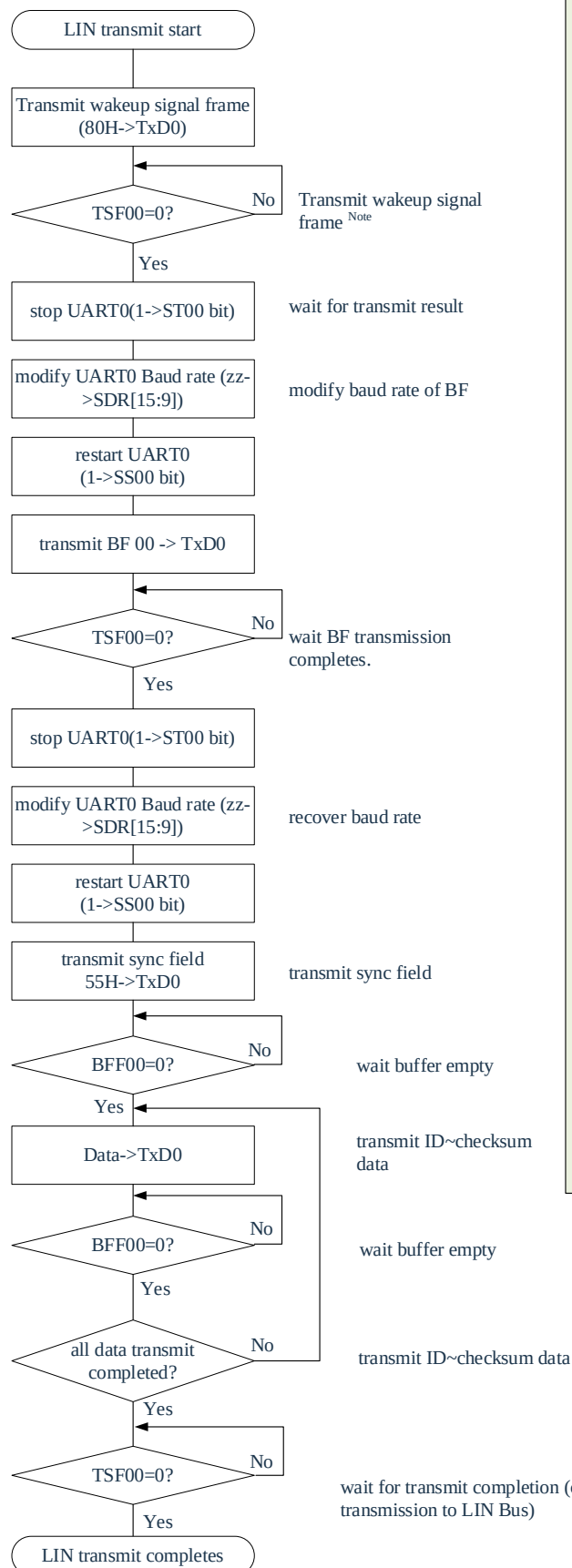
$$(\text{Baud rate for break field}) = 9/13 \times N$$

Transmit the data of "00H" through this baud rate to generate a break field.

3. Output INTST0 at the end of each data transmission, and output INTST0 at BF transmission.

Remark The software controls the break between fields.

Figure 13-114 Flowchart of LIN transmission



Note It is limit to situations starting from LIN-bus sleep.

Remark This is the process that starts by ending the initial set-up of the UART and allowing slave transmission.

13.8.2 LIN Reception

In UART reception, UART0 supports LIN communication.

The LIN reception uses channel 1 of unit 0.

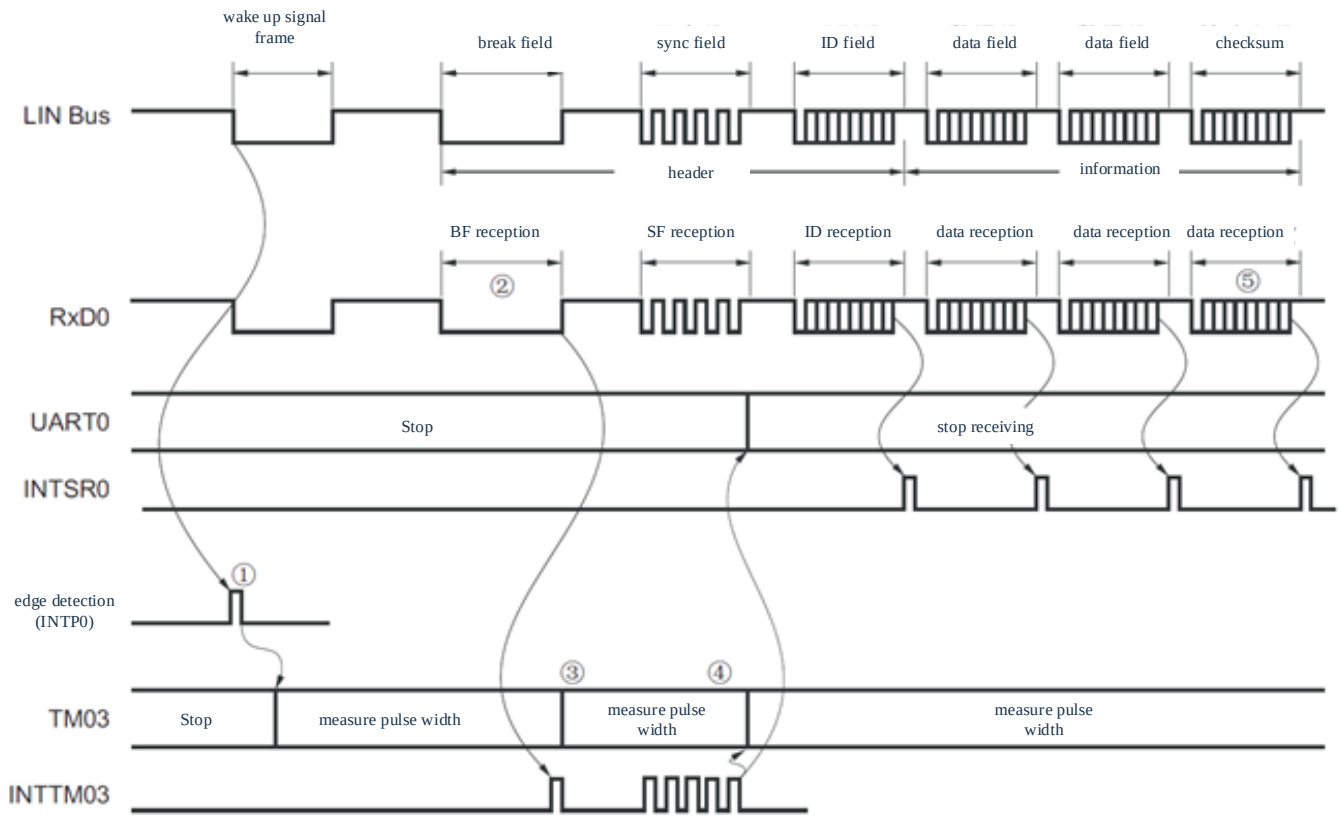
UART	UART0	UART1	UART2	UART3
LIN communication support	Yes	No	No	No
Target channel	Channel 1 of SCI0	—	—	—
Used pin	RxD0	—	—	—
Interrupt	INTSR0	—	—	—
	Limited to transmit complete interrupts (disable setting buffer empty interrupts).			
Error interrupt	INTSRE0	—	—	—
Error detection flag	• Frame error detection flag (FEF01). • Overflow error detection flag (OVF01).			
Transfer data length	8 bits			
Transfer rate ^{Note}	Max. $f_{MCK}/6[\text{bps}](\text{SDR01}[15:9] \geq 2)$, Min. $f_{CLK}/(2 \times 2^{15} \times 128)[\text{bps}]$			
Data phase	Non-reverse output (default: high). Reverse output (default: low).			
Parity bit	No parity bits (no parity).			
Stop bit	Append 1 bit.			
Data direction	LSB first			

Note 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} : Operation clock frequency of the target channel
 f_{CLK} : System clock frequency

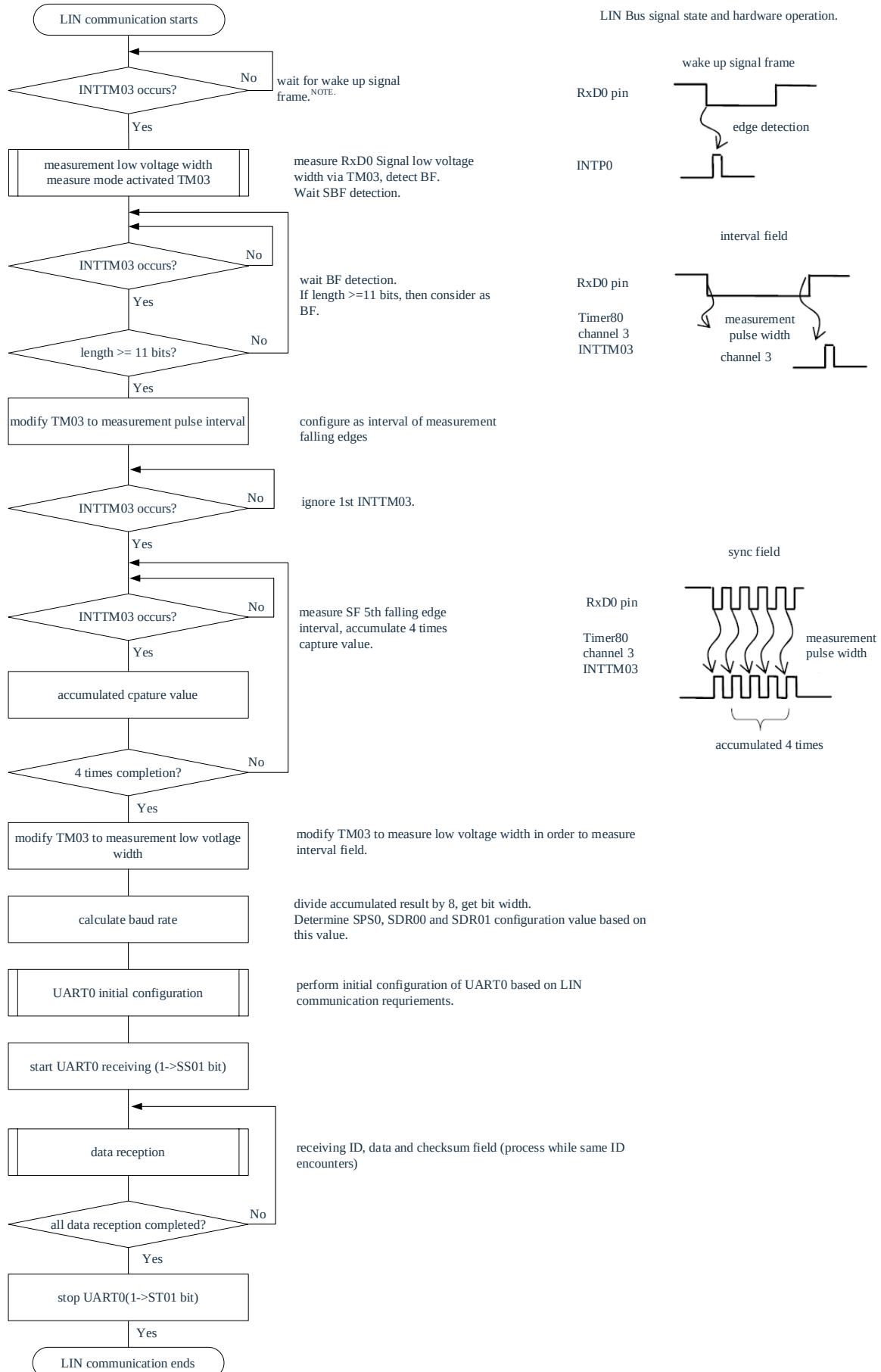
A summary of the LIN receive operation is shown in Figure 13-115.

Figure 13-115 LIN reception operation



The signal processing flow is as follows:

- ① The wake-up signal is detected by monitoring the interrupt edge of the pin (INTP0). When the wake-up signal is detected, TM03 is set to measure the low-level width of the BF signal, and then the system enters the BF reception waiting state.
- ② The wake-up signal is detected by monitoring the interrupt edge of the pin (INTP0). When the wake-up signal is detected, TM03 is set to measure the low-level width of the BF signal, and then the system enters the BF reception waiting state.
- ③ When the BF reception completes normally, TM03 must be set to measure the pulse interval and measure the interval between four consecutive falling edges of the RxD0 signal in the synchronization field (See “5.8.4 Operation as input pulse interval measurement”).
- ④ The baud rate error is calculated based on the bit interval of the synchronization field (SF). Afterward, the baud rate must be adjusted (reset) after pausing UART0 operation.
- ⑤ The checksum field must be differentiated by software. Software must also initialize UART0 and reset it to the BF reception waiting state after receiving the checksum field.

Figure 13-116 Flowchart of LIN reception


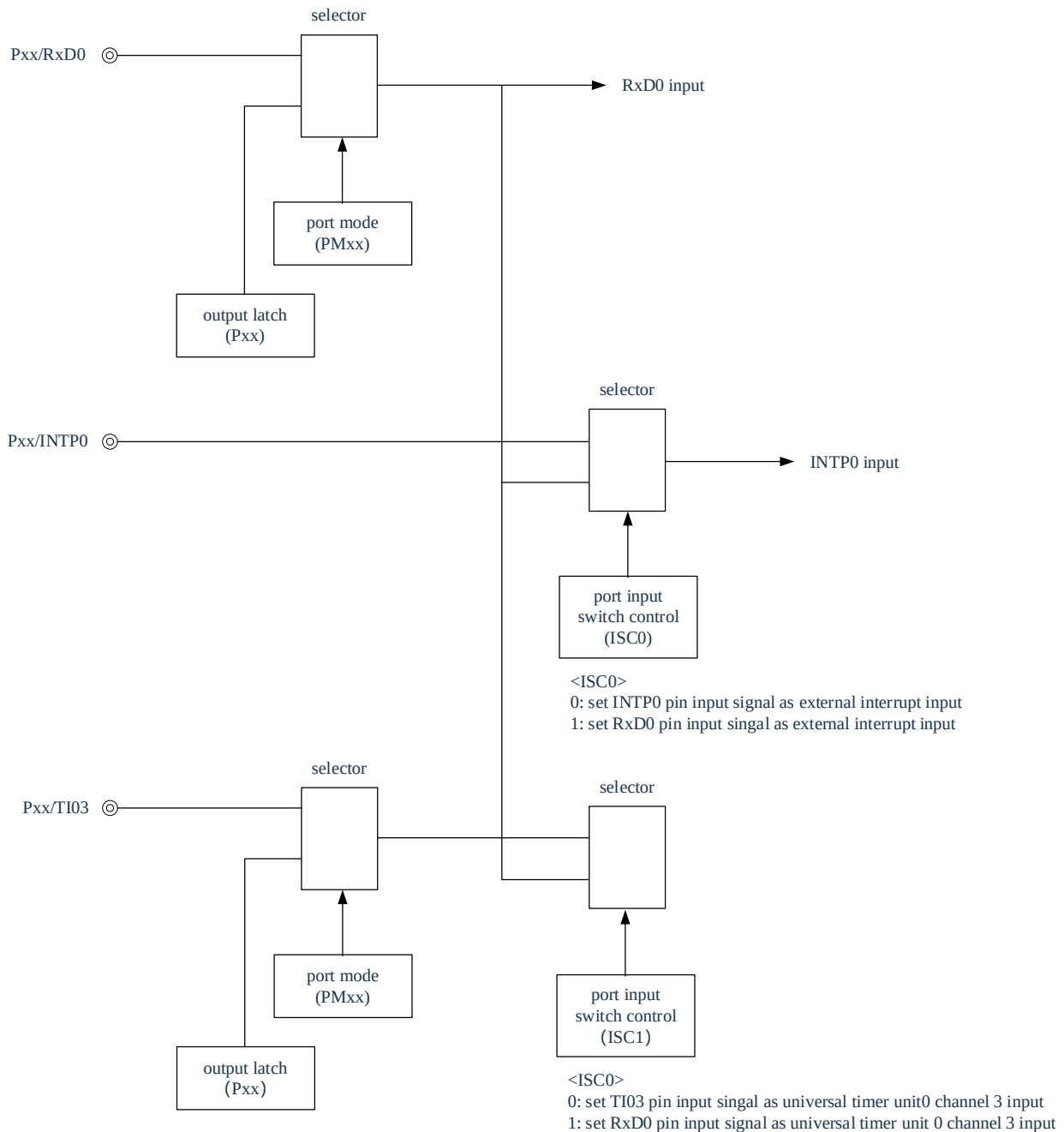
Note It is only needed in the sleeping status.

The port structure diagram for the LIN reception operation is shown in Figure 13-117.

The wake-up signal sent by the LIN master control is received through edge detection of the external interrupt (INTP0). It is possible to measure the length of the synchronization field sent by the LIN master control and calculate the baud rate error using the external event capture operation of the universal timer unit.

Through the port input switch control (ISC0/ISC1), the input source for the receiving port (RxD0) can be routed to the external interrupt (INTP0) and the timer array unit without external wiring.

Figure 13-117 Port structure diagram for LIN receive operation



Remark ISC0, ISC1: bit0 and bit1 of the Input Switching Control Register (ISC) (See Figure 13-19)

Peripheral features for LIN communication operations are summarized as follows:

<Peripheral features used>

- External interrupt (INTP0): Wake-up signal detection
Purpose: Detect the edge of the wake-up signal and the start of communication.
- Channel 3 of the universal timer unit: baud rate error detection, break field (BF) detection
Purpose: Measure the length of the synchronization field (SF) and calculate the baud rate error by dividing the length by the number of bits (by measuring the interval between the edges of the RxD0 input in capture mode). Measure the low-level width to determine whether it is a break field (BF).
- Channel 0 and Channel 1 (SCI0) of the Universal Serial Communication Unit 0 (UART0)

13.9 Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) 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.

[Transmit and receive data]

- 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

Note When receiving the last data, if you write “0” to the SDOEmn bit (SDOEm register) to stop the output of the serial communication data, the ACK is not output. For details, please refer to “13.9.3(2) Processing Flow”.

Remark m: unit number (m=0, 1) **n:** channel number (n=0~3) mn=00~03, 10~11

Channels 0 to 3 of SCI0 and channels 0 to 1 of SCI1 support Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) channels.

Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) operates in the following 4 communication modes:

- Address field transmission (see 13.9.1)
- Data transmission (see 13.9.2)
- Data reception (see 13.9.3)
- Generation of stop conditions (see 13.9.4)

13.9.1 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.

Simplified I ² C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 0 of SCI0	Channel 1 of SCI0	Channel 2 of SCI0	Channel 3 of SCI0	Channel 0 of SCI1	Channel 1 of SCI1
Used pin	SCL00, SDA00 ^{Note 1}	SCL01, SDA01 ^{Note 1}	SCL10, SDA10 ^{Note 1}	SCL11, SDA11 ^{Note 1}	SCL20, SDA20 ^{Note 1}	SCL21, SDA21 ^{Note 1}
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
Error detection flag	Limited to transmit complete interrupts (buffer empty interrupts cannot be selected).					
Transfer data length	ACK error detection flag (PEFmn).					
Transfer rate ^{Note 2}	8 bits (send the highest 7 bits as the address and the lower 1 bit as R/W control).					
Data level	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).					
Parity bits	Non-reverse output (default: high).					
Stop bit	No parity bits.					
Data direction	Append 1 bit (for ACK reception).					
	MSB first					

Notes

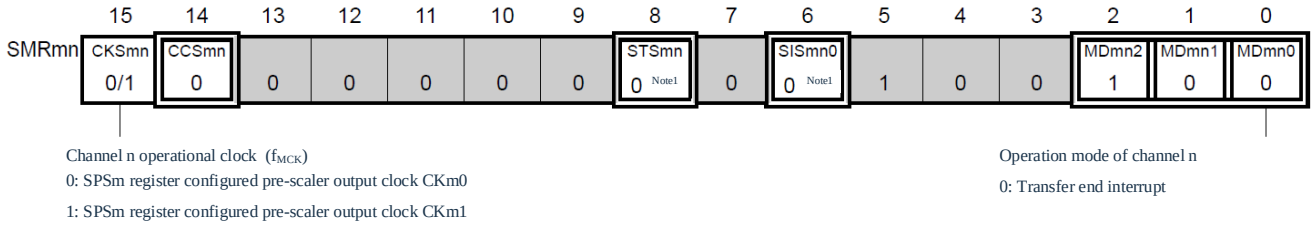
1. To communicate via Simplified I²C, N-channel open-drain output mode (POMxx=1) must be set through the port output mode register (POMxx). For details, please refer to “Chapter 2 Pin Functions”.
2. It must be used within the scope of the peripheral functional characteristics that meet this condition and meet the electrical characteristics (refer to the data sheet).

Remark m: unit number (m=0, 1) n: channel number (n=0~3) mn=00~03, 10~11.

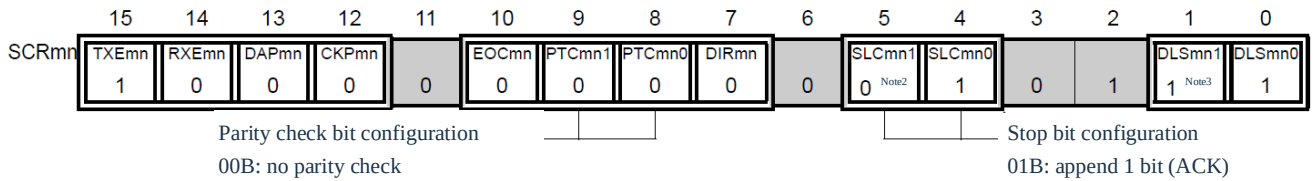
(1) Register setting

Figure 13-118 Example of register setting contents when transmitting address field of simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21)

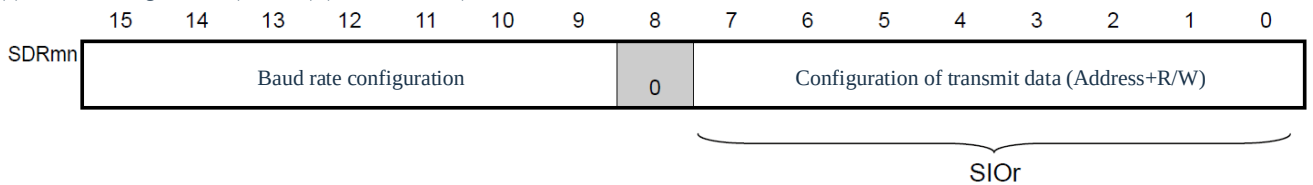
(a) Serial mode register mn (SMRmn)



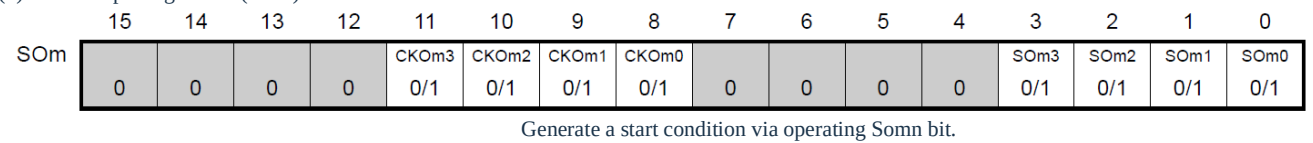
(b) Serial communication operation configuration register mn (SCRmn)



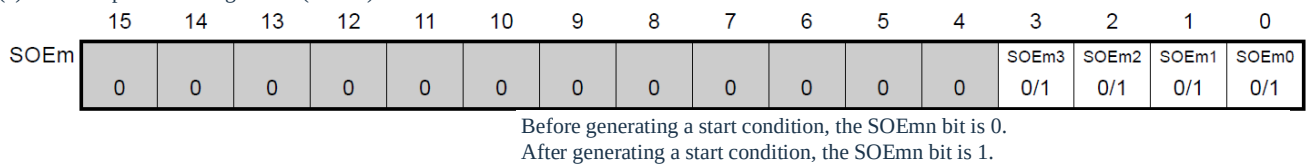
(c) serial data register mn (SDRmn) (low 8 bit: SIO_r)



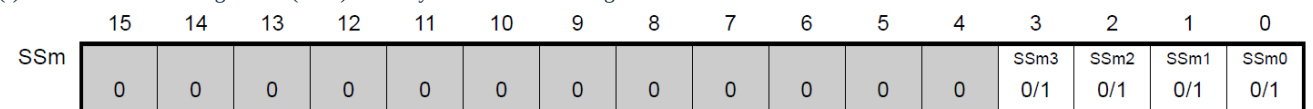
(d) Serial output register m (SOM_m)



(e) Serial output enable register m (SOEm)



(f) Serial channel start register m (SSm) Only set the bit of the target channel to 1.



Notes 1. Limited to SMR00, SMR03, SMR11.

2. Limited to SCR00, SCR02, SCR10.

3. Limited to SCR00 register and SCR01 register, other fixed as "1".

Remarks

1. m: unit number (m=0, 1) n: channel number (n=0~3) r: IIC number (r=00, 01, 10, 11, 20, 21)
 mn=00~03, 10~11

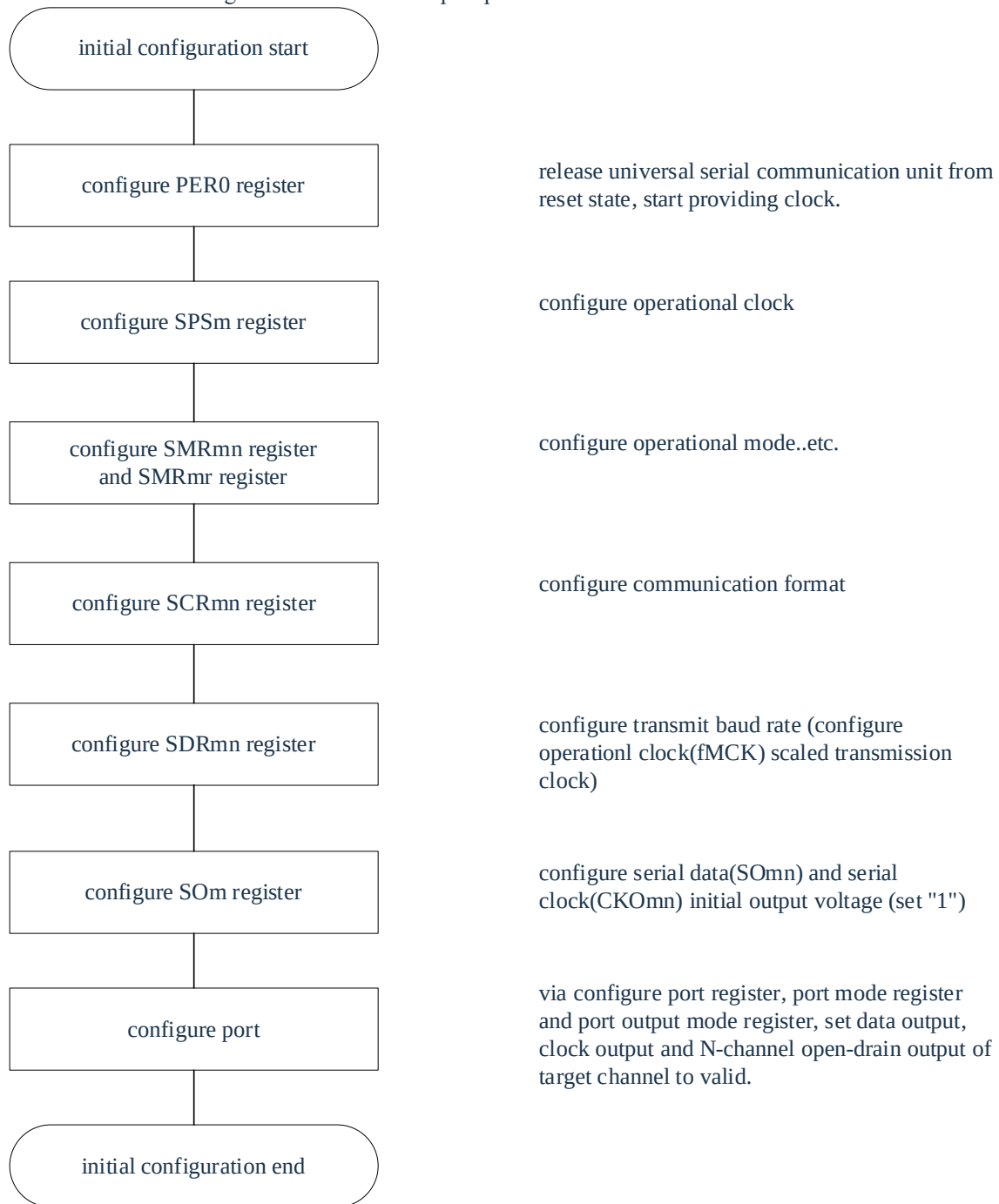
2. : Fixed in IIC mode. : Cannot be set (initial value).

×: 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) Operation steps

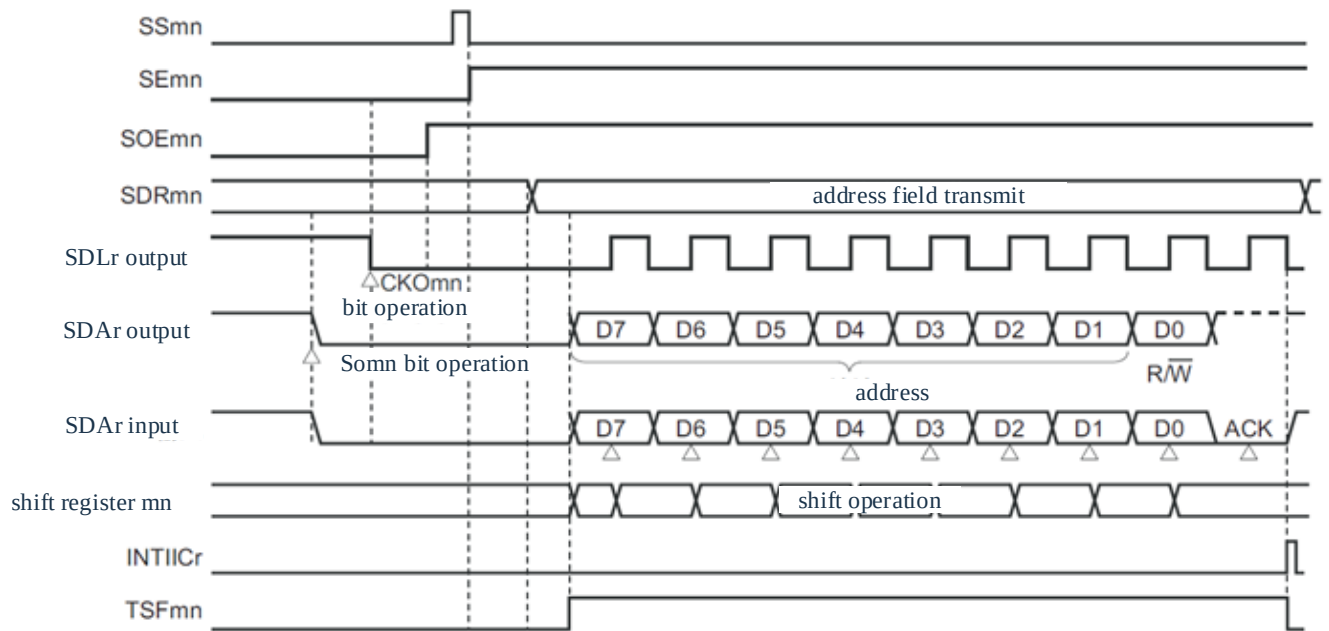
Figure 13-119 Initial setup step for the address field transmission



Remark At the end of the initial setup, Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20 IIC21) is output disabled and is in the operation stop state.

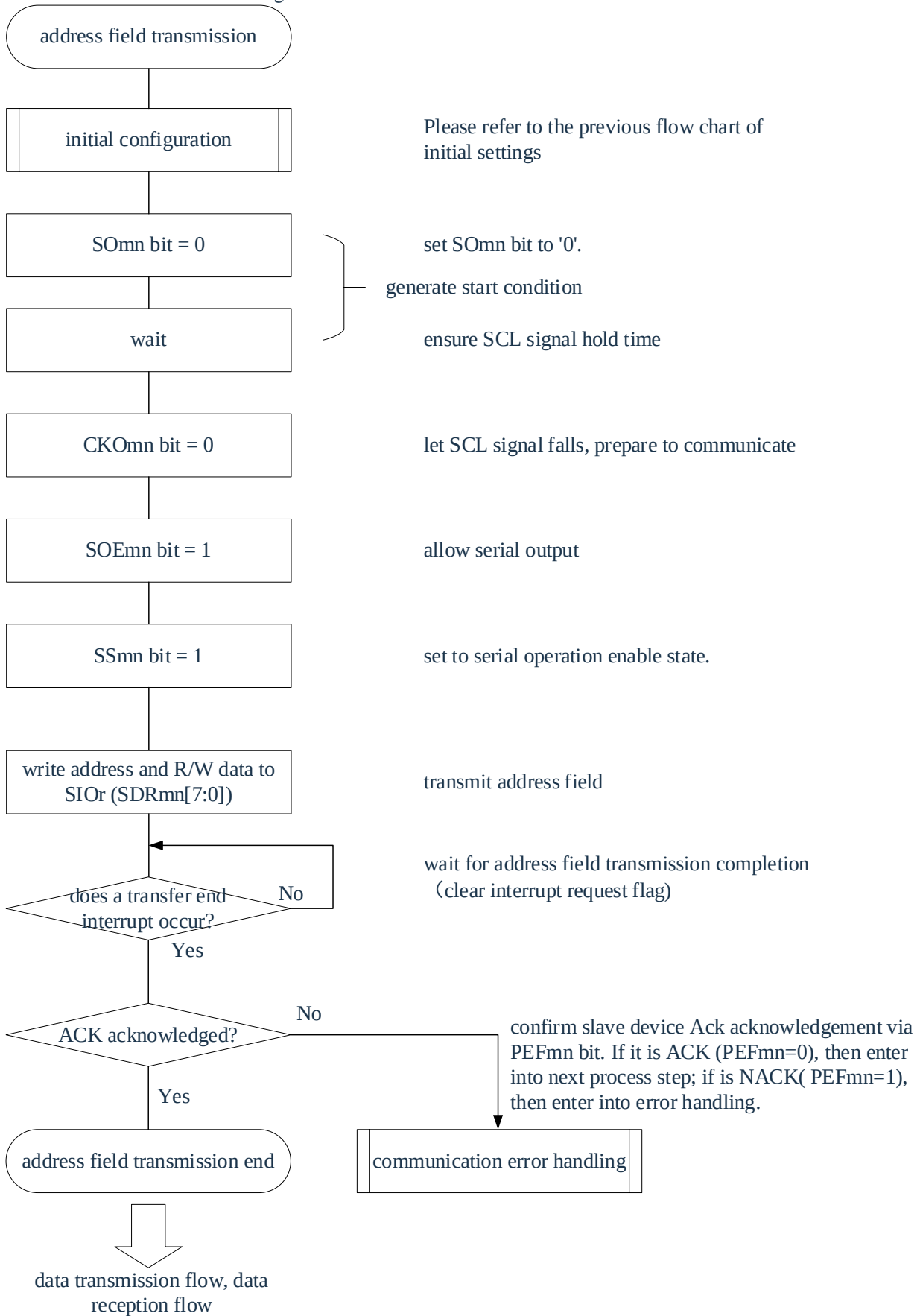
(3) Processing flow

Figure 13-120 Timing diagram of the address field transmission



Remark m: unit number (m=0, 1) n: channel number (n=0~3) r: IIC number (r=00, 01, 10, 11, 20, 21)
mn=00~03, 10~11

Figure 13-121 Flowchart of address field transmission



13.9.2 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.

Simplified I ² C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 0 of SCI0	Channel 1 of SCI0	Channel 2 of SCI0	Channel 3 of SCI0	Channel 0 of SCI1	Channel 1 of SCI1
The pin used	SCL00, SDA00 ^{Note1}	SCL01, SDA01 ^{Note1}	SCL10, SDA10 ^{Note1}	SCL11, SDA11 ^{Note1}	SCL20, SDA20 ^{Note1}	SCL21, SDA21 ^{Note1}
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	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 ^{Note 2}	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 reception).					
Data direction	MSB first					

Notes

1. To communicate via Simplified I²C, N-channel open-drain output mode (POMxx=1) must be set through the port output mode register (POMxx). For details, please refer to 2.3 Registers for Controlling Port Functions and 2.5 Special Pin Configuration.
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 m: unit number (m=0, 1) n: channel number (n=0~3) mn=00~03, 10~11

(1) Register setting

Figure 13-122 Example of register setting contents for simplified I²C data transmission (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21)

(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: SIOr) 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						
</																

(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					CKOm3	CKOm2	CKOm1	CKOm0					SOM3	SOM2	SOM1	SOM0
	0	0	0	0	0/1 Note5	0/1 Note5	0/1 Note5	0/1 Note5	0	0	0	0	0/1 Note5	0/1 Note5	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													SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	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													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

Notes 1. Limited to SMR01, SMR03, SMR11 registers.

2. Limited to SCR00, SCR02, SCR10 registers.

3. Limited to SCR00 register and SCR01 register, other fixed as “1”.

4. Because it is already set when sending the address segment, it does not need to be set.

5. During the operation of communication, the value changes due to the communication data.

Remarks 1.m: unit number (m=0, 1) n: channel number (n=0~3) r: IIC number (r=00, 01, 10, 11, 20, 21)

mn=00~03, 10~11

2. : Fixed in IIC mode. : Cannot be set (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) Processing flow

Figure 13-123 Timing diagram of data transmission

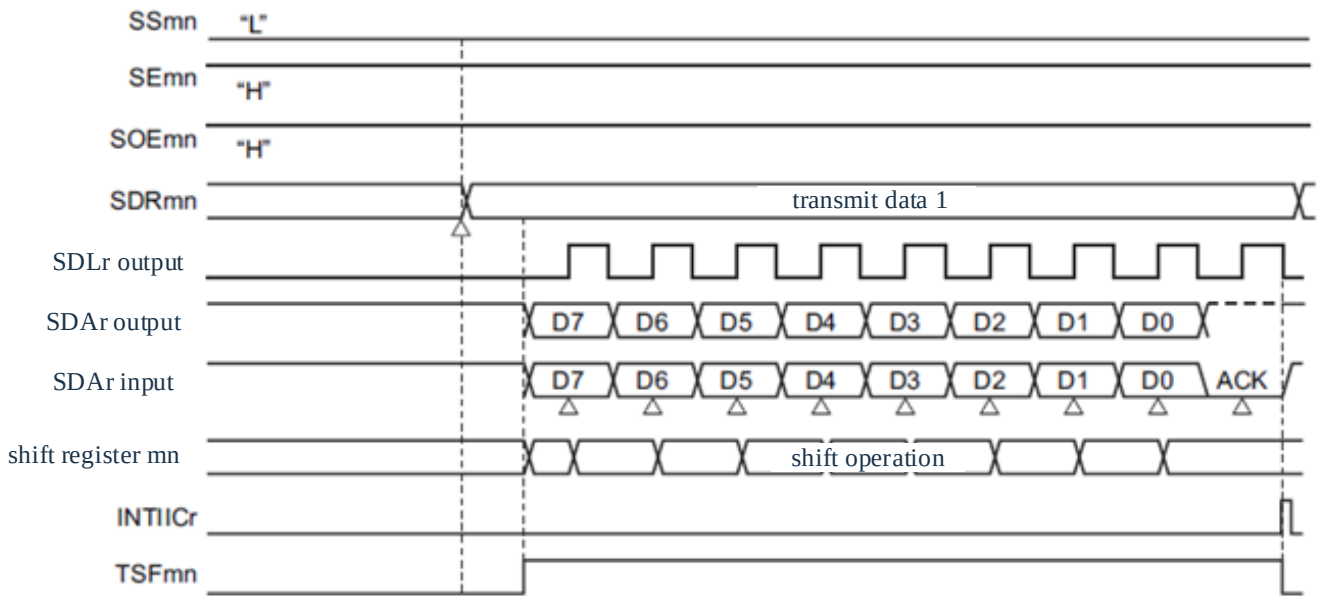
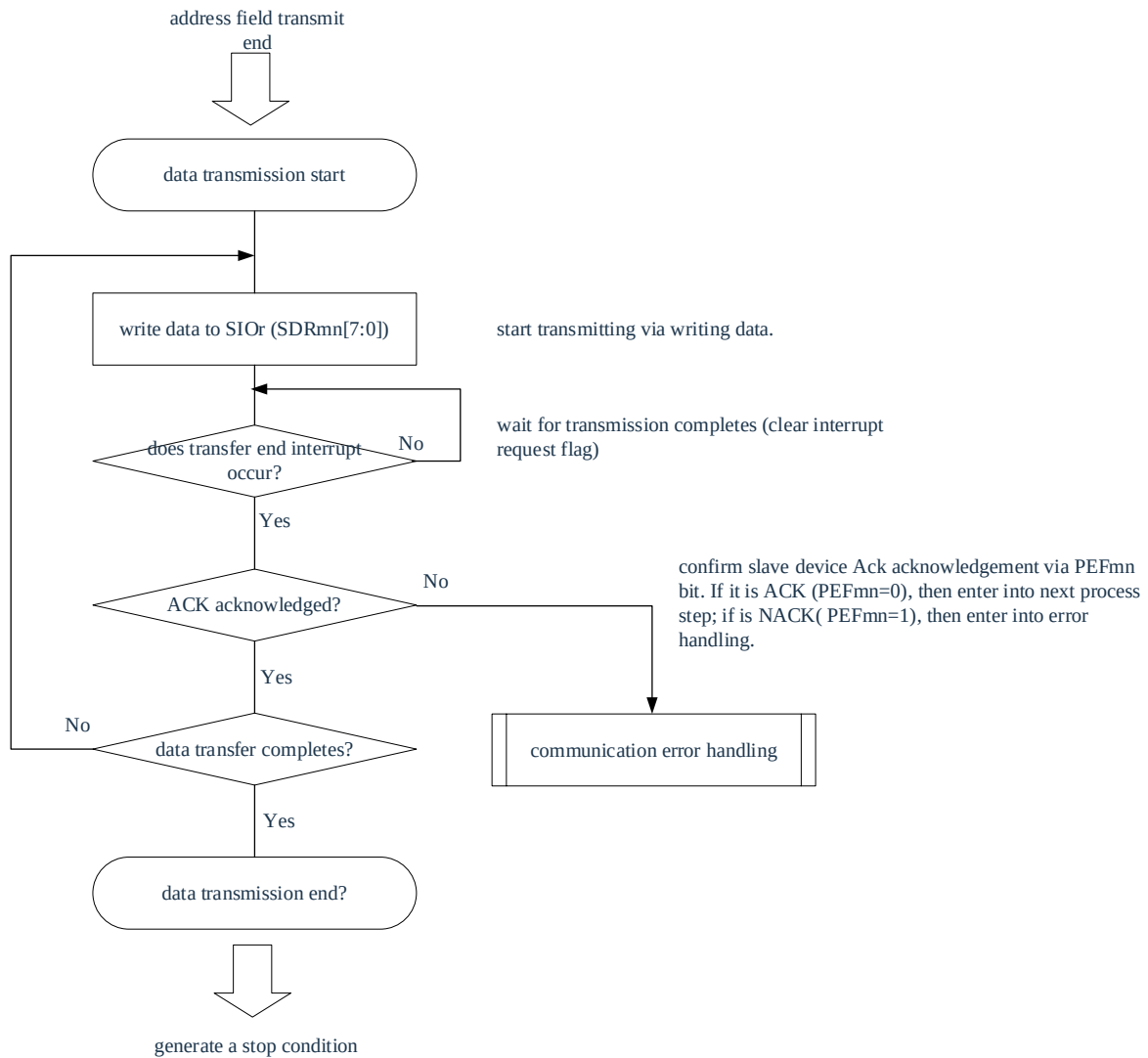


Figure 13-124 Flow chart of data transmission



13.9.3 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.

Simplified I ² C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 0 of SCI0	Channel 1 of SCI0	Channel 2 of SCI0	Channel 3 of SCI0	Channel 0 of SCI1	Channel 1 of SCI1
Used pin	SCL00, SDA00 ^{Note1}	SCL01, SDA01 ^{Note1}	SCL10, SDA10 ^{Note1}	SCL11, SDA11 ^{Note1}	SCL20, SDA20 ^{Note1}	SCL21, SDA21 ^{Note1}
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
Error detection flag	Limited to transmit complete interrupts (buffer empty interrupts cannot be selected).					
Transfer data length	Only the overflow error detection flag (OVFmn).					
Transfer rate ^{Note2}	8 bits					
	Max. $f_{MCK}/4[\text{Hz}]$ ($\text{SDRmn}[15:9] \geq 1$) 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 transmission).					
Data direction	MSB first					

Notes

1. To communicate via Simplified I²C, N-channel open-drain output mode (POMxx=1) must be set through the port output mode register (POMxx). For details, please refer to 2.3 Registers for Controlling Port Functions and 2.5 Special Pin Configuration.
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 m: unit number (m=0, 1) n: channel number (n=0~3) mn=00~03, 10~11.

(1) Register setting

Figure 13-125 Example of register setting contents for simplified I²C data reception (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21)

(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
	0	1	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)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate configuration ^{Note4}								0	Virtual 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					CKOm3	CKOm2	CKOm1	CKOm0					SOM3	SOM2	SOM1	SOM0
	0	0	0	0	0/1 Note5	0/1 Note5	0/1 Note5	0/1 Note5	0	0	0	0	0/1 Note5	0/1 Note5	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													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) 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													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

Notes 1. Limited to SMR01, SMR03, SMR11 registers.

2. Limited to SCR00, SCR02, SCR10 registers.

3. Limited to SCR00 register and SCR01 register, other fixed as “1”.

4. Because it is already set when sending the address segment, it does not need to be set.

5. During the operation of communication, the value changes due to the communication data.

Cautions 1. m: unit number (m=0, 1) n: channel number (n=0~3) r: IIC number (r=00, 01, 10, 11, 20, 21)

mn=00~03, 10~11.

2. : Fixed in IIC mode. : Cannot be set (set initial value).

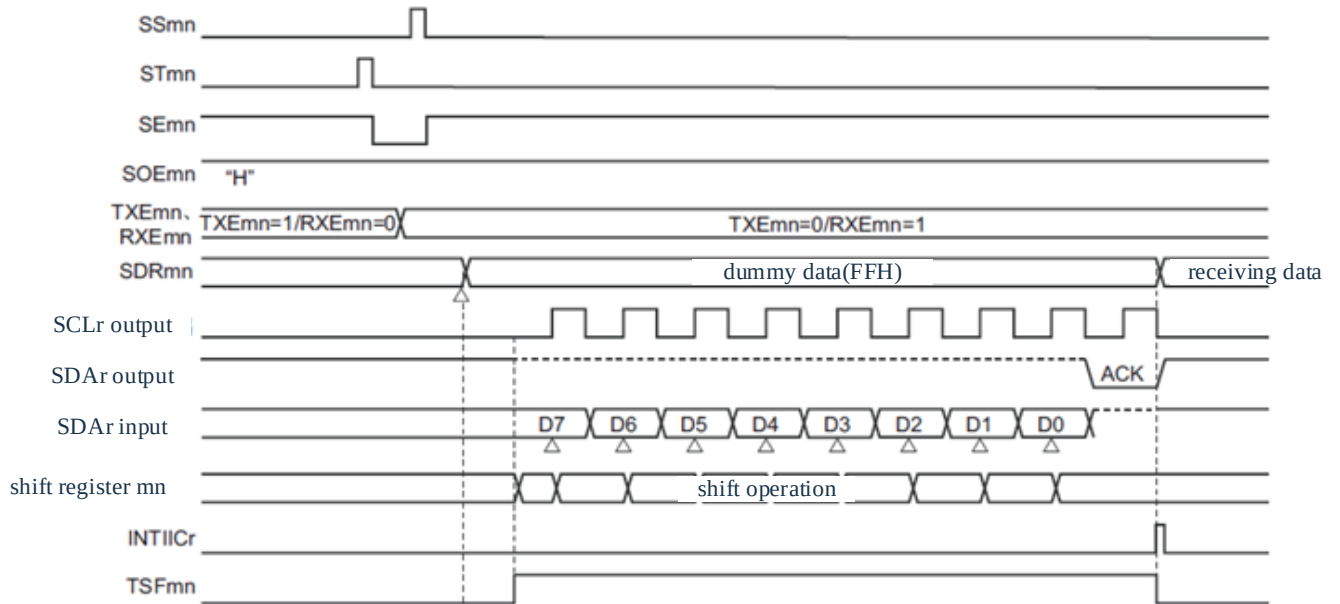
×: This is the bit that cannot be used in this mode (set the initial value if it is not used in other modes either).

0/1: Set “0” or “1” according to the user’s purpose.

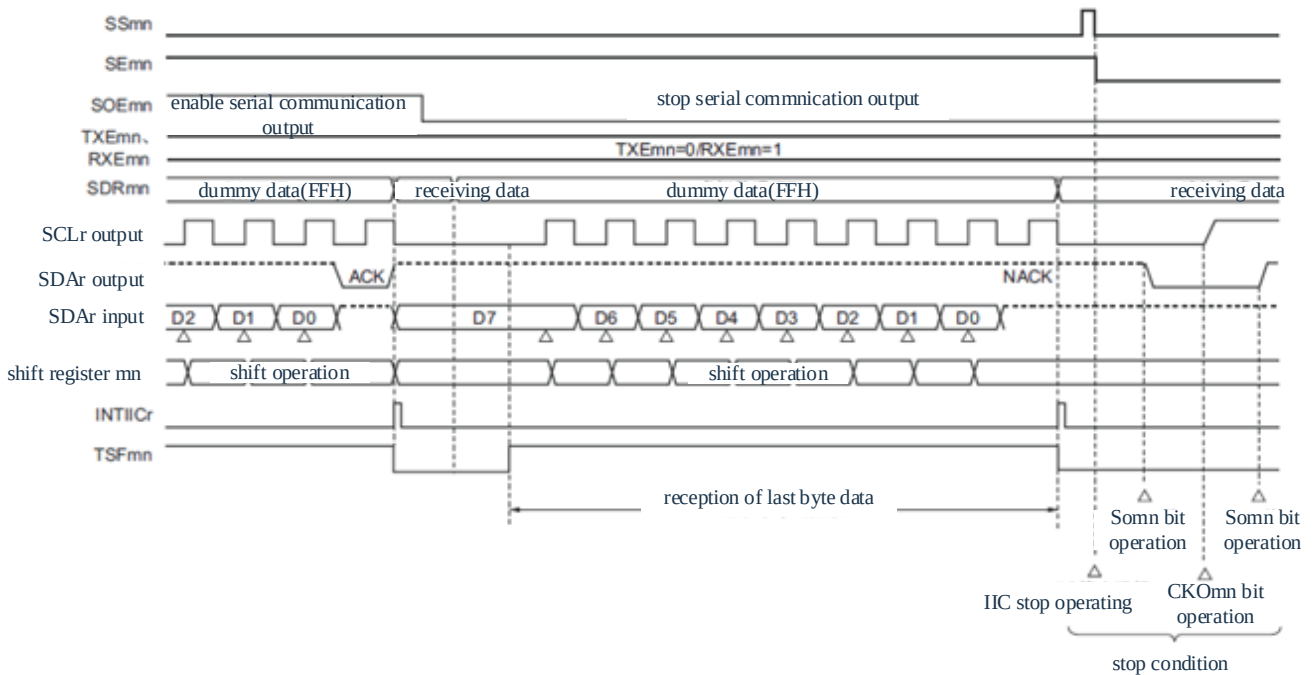
(2) Processing flow

Figure 13-126 Timing diagram of data reception

(a) Start of receiving data



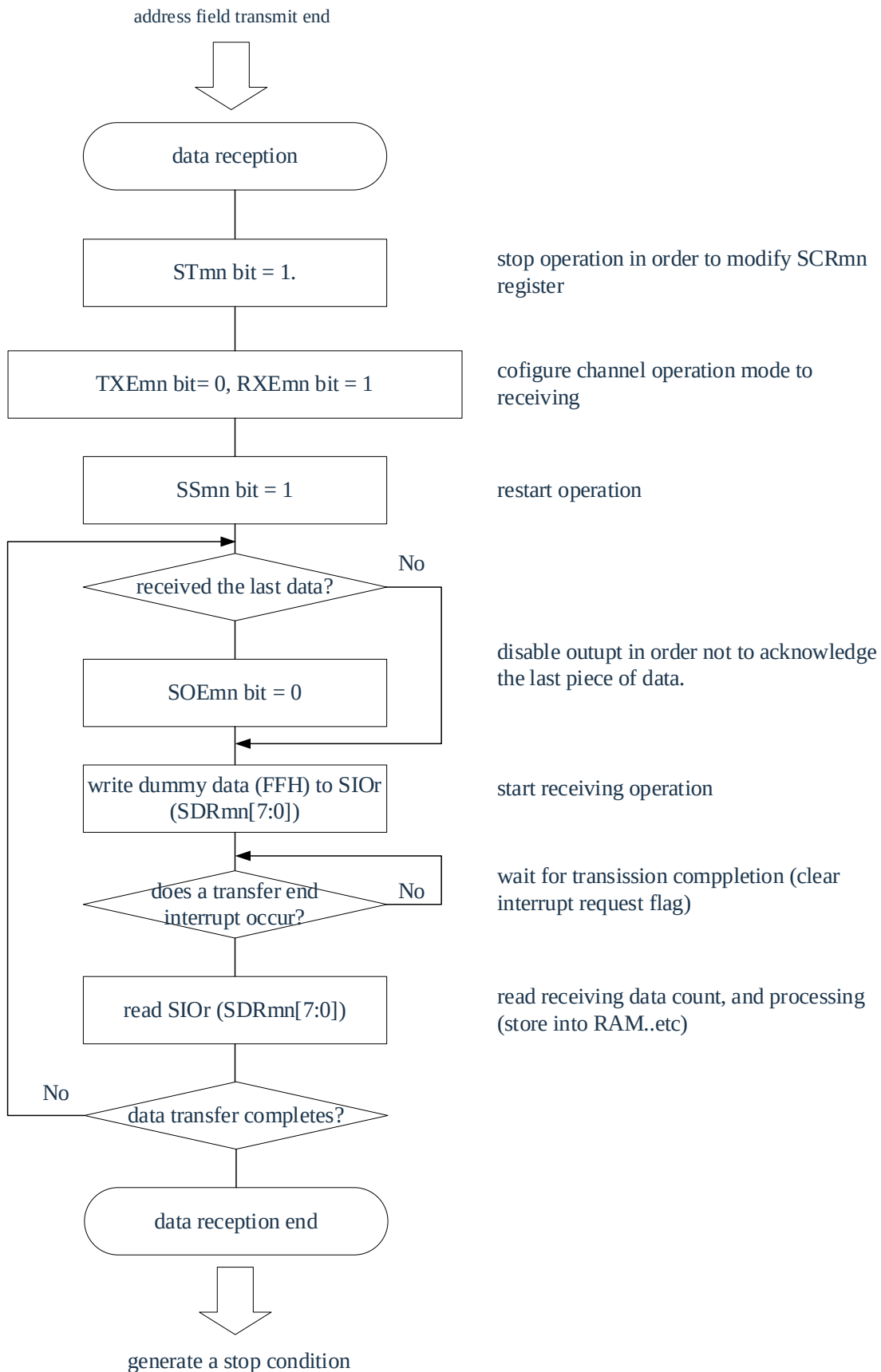
(b) Receive the last data



Remark m: unit number (m=0, 1) n: channel number (n=0~3) r: IIC number (r=00, 01, 10, 11, 20, 21)

mn=00~03, 10~11

Figure 13-127 Flowchart of data reception



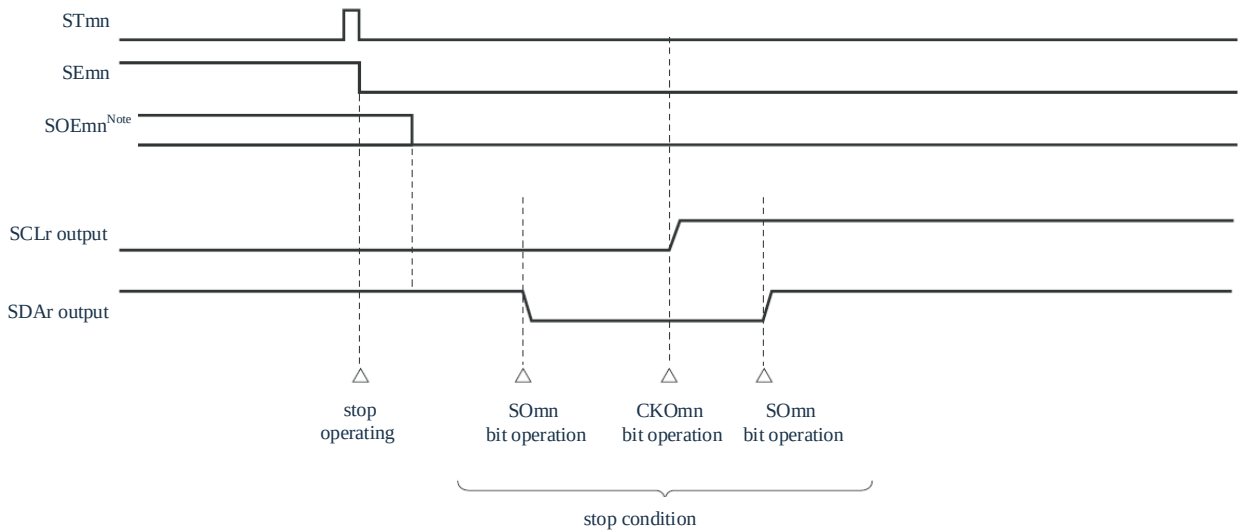
Caution No ACK(NACK) is output when the last data is received. Thereafter, the communication is terminated by stopping the STmn bit of the serial channel stop register m (STm) to 1.

13.9.4 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.

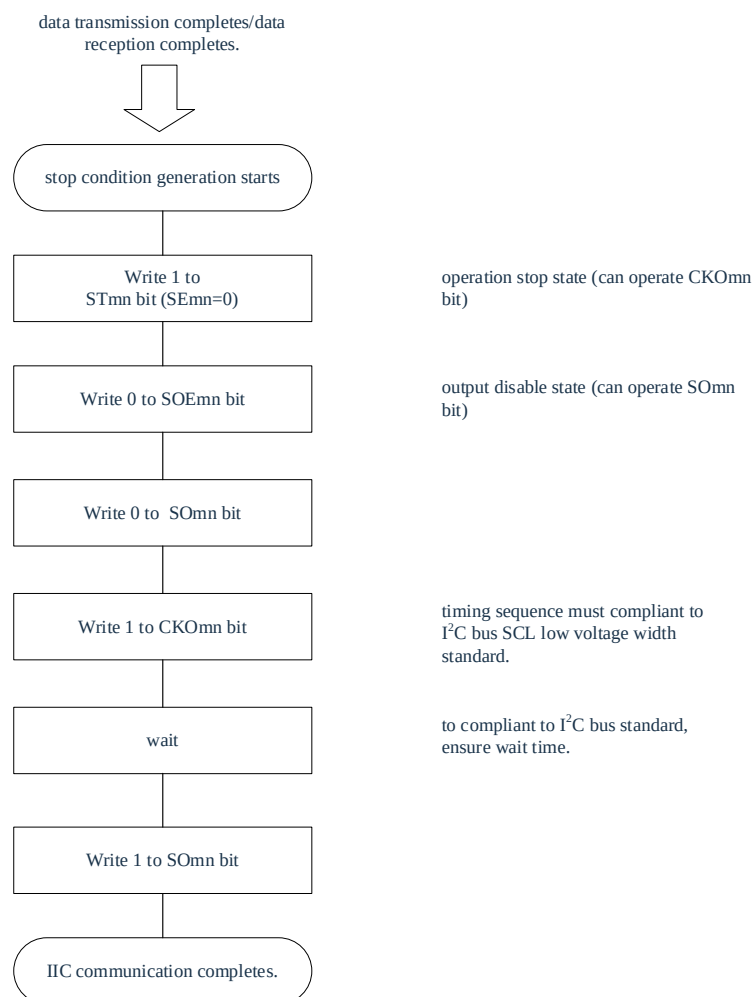
(1) Processing flow

Figure 13-128 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 13-129 Flow chart for generating a stop condition



13.9.5 Calculation of Transfer Rate

The transfer rate for simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) communication can be calculated using the following equation.

$$(\text{Transfer rate}) = \{\text{Operation clock (f}_{\text{MCK}}) \text{ frequency of the target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2$$

Caution Setting SDRmn[15:9] to “0000000B” is prohibited, and the setting value for SDRmn[15:9] must be greater than or equal to “0000001B”. The duty ratio of the SCL signal output by the Simplified I²C is 50%. In I²C bus specification, the low-level width of the SCL signal is greater than the high-level width. Therefore, if 400kbps is set as a fast mode or 1Mbps is set as an enhanced fast mode, the low-level width of the SCL signal output is less than the specification value of the I²C bus. You must set a value for SDRmn[15:9] that meets the I²C bus specification.

Remarks 1. The value of SDRmn[15:9] corresponds to bits 15 to 9 of the Serial Data Register (SDRmn), with a range from 0000001B to 1111111B, meaning the value is from 1 to 127.
2. m: unit number (m=0, 1) n: channel number (n=0~3) mn=00~03, 10~11

The operating clock (f_{MCK}) is determined by the serial clock select register m (SPSm) and bit 15 (CKSmn bit) of the serial mode register mn (SMRmn).

Table 13-5 Selection of Simplified I²C 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} =32MHz in operation
0	X	X	X	X	0	0	0	0	f _{CLK}	32MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	16MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	8MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	4MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	2MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	1MHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	500kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	250kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	125kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	62.5kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	31.25kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	15.63kHz
1	0	0	0	0	X	X	X	X	f _{CLK}	32MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	16MHz
	0	0	1	0	X	X	X	X	f _{CLK} /2 ²	8MHz
	0	0	1	1	X	X	X	X	f _{CLK} /2 ³	4MHz
	0	1	0	0	X	X	X	X	f _{CLK} /2 ⁴	2MHz
	0	1	0	1	X	X	X	X	f _{CLK} /2 ⁵	1MHz
	0	1	1	0	X	X	X	X	f _{CLK} /2 ⁶	500kHz
	0	1	1	1	X	X	X	X	f _{CLK} /2 ⁷	250kHz
	1	0	0	0	X	X	X	X	f _{CLK} /2 ⁸	125kHz
	1	0	0	1	X	X	X	X	f _{CLK} /2 ⁹	62.5kHz
	1	0	1	0	X	X	X	X	f _{CLK} /2 ¹⁰	31.25kHz
	1	0	1	1	X	X	X	X	f _{CLK} /2 ¹¹	15.63kHz
Other than the above									Disable settings.	

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 communications init (SCI) (serial channel stop register m (STm) = 000FH).

Remarks

1. X: Ignore
2. m: unit number (m=0, 1) n: channel number (n=0~3) mn=00~03, 10~11

An example of setting the I²C transfer rate at f_{MCK}=f_{CLK}=32MHz is shown below.

I ² C transfer mode (expected transfer rate)	f _{CLK} =32MHz			
	Operation clock (f _{MCK})	SDRmn[15:9]	Calculated transfer rate	Error with expected transfer rate
100kHz	f _{CLK} /2	79	100kHz	0.0%
400kHz	f _{CLK}	41	380kHz	5.0% ^{Note}
1MHz	f _{CLK}	18	0.84MHz	16.0% ^{Note}

Note The error cannot be set to 0% because the SCL signal has a 50% duty cycle.

13.9.6 Processing Steps When an Error Occurs During Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) Communication

The processing steps when an error occurs during a simple I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) communication are shown in Figure 13-130 and Figure 13-131.

Figure 13-130 Steps for handling when an overflow error occurs

Software operation	Hardware status	Remark
Read serial data register mn (SDRMN). →	The BFF m n bit of the SSRm n register is 0 and channel n is receivable.	This is to prevent overflow 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.

Figure 13-131 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, 1) n: channel number (n=0~3) r: IIC number (r=00, 01, 10, 11, 20, 21)

mn=00~03, 10~1.

Chapter 14 Serial Interface IICA

14.1 Functions of Serial Interface IICA

This Serial Interface IICA has the following three modes.

(1) Run stop mode

This is a mode for non-serial transfer, which reduces power consumption.

(2) I²C bus mode (multi-master supported)

This mode transfers 8-bit data to and from multiple devices via two lines on the serial clock (SCLAn) and serial data bus (SDAAn). Conforming to the I²C bus format, the master device can generate “start condition”, “address”, “transfer direction”, “data” and “stop condition” for slave devices on the serial data bus. The slave device automatically detects the received status and data by hardware. This function simplifies the I²C bus control of the application program.

Because the SCLAn pin and SDAAn pin of the serial interface IICA are used as open-drain outputs, the serial clock line and serial data bus require pull-up resistors.

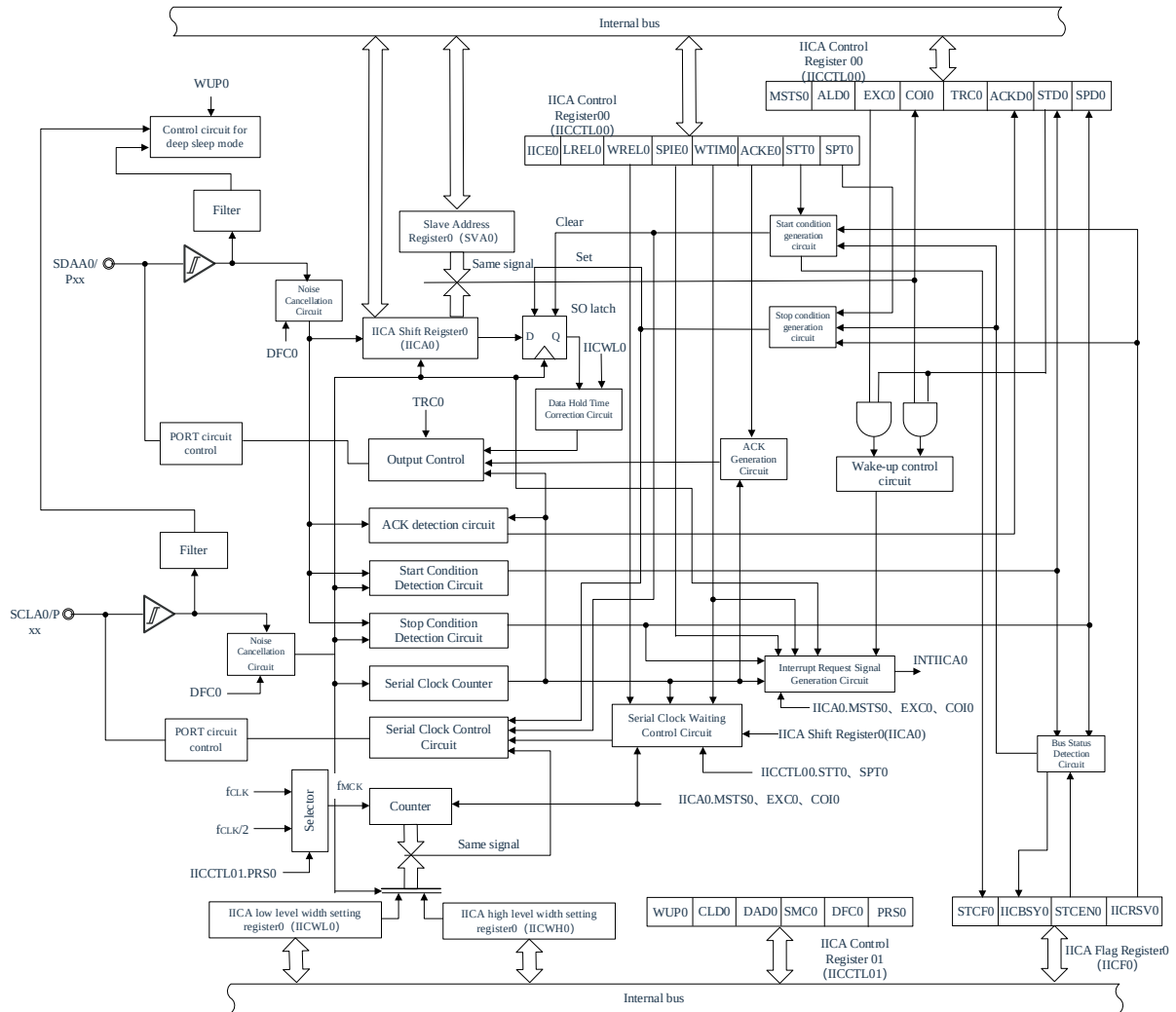
(3) Wake-up mode

In a deep sleep mode, when an extension code or a local station address from a master device is received, the deep sleep mode can be released by generating an interrupt request signal (INTIICAn). It is set by the WUPn bit of the IICA control register n1 (IICTLn1).

The block diagram of the serial interface IICA is shown in Figure 14-1.

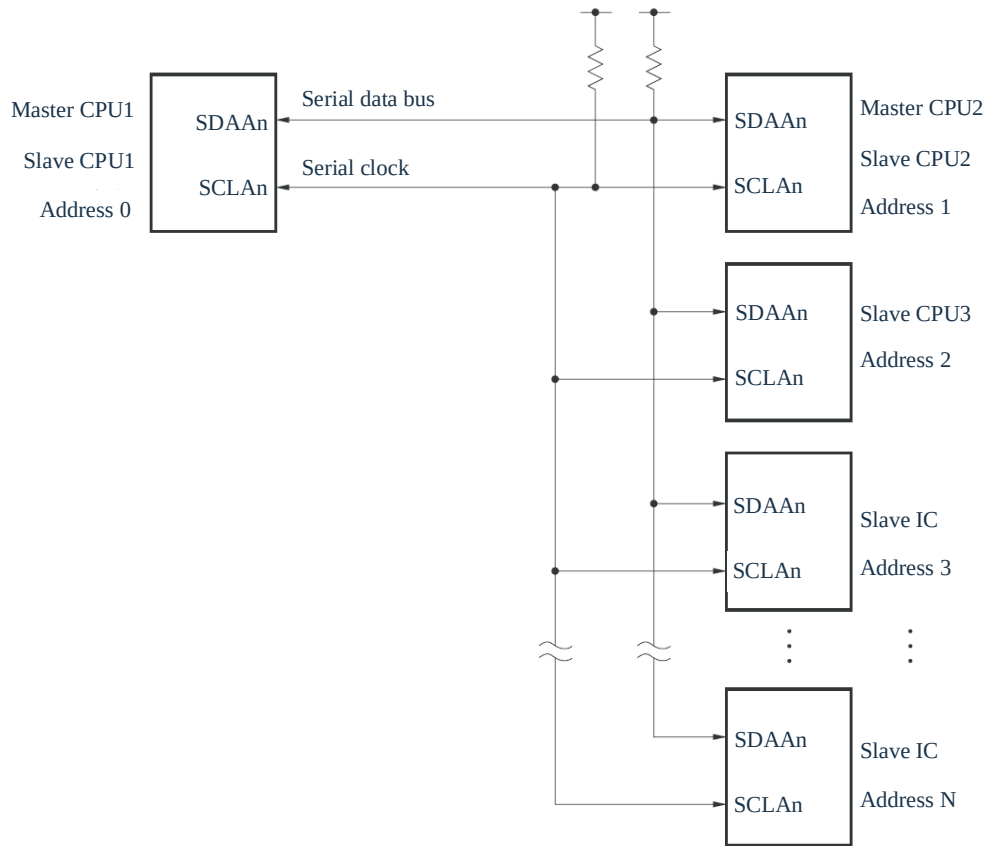
Remark: n=0.

Figure 14-1 Block diagram of serial interface IICA



An example of serial bus structure is shown in Figure 14-2.

Figure 14-2 Example of a serial bus structure for I²C bus



Remark: n=0.

14.2 Structure of Serial Interface IICA

The serial interface IICA consists of the following hardware.

Table 14-1 Structure of serial interface IICA

Item	Structure
Register	IICA shift register n (IICAn) Slave address register n (SVAn)
Control register	Peripheral enable register 0 (PER0) IICA control register n0 (IICCTLn0) IICA status register n (IICSn) IICA flag register n (ICFn) IICA control register n1 (IICCTLn1) IICA low-level width setting register n (IICWLn) IICA high-level width setting register n (IICWHn) Port mode register (PMxx) Port mode control register (PMCxx) Port alternate function configuration register (PxxCFG)

Remark 1: n=0.

Remark 2: This product can alternate the IICA input/output pin functions to multiple ports other than the P00 pin. When a port is configured to IICA pin alternate function, the N-channel open drain output (V_{DD}/EV_{DD} withstand) mode of that port is guaranteed by design to be turned on automatically, i.e., the POMxx registers do not need to be set by the user.

(1) IICA shift register n (IICAn)

The IICAn registers are registers for transmitting and receiving 8-bit serial data and 8-bit parallel data interconversion synchronously with the serial clock. The actual transmission and reception can be controlled by reading and writing the IICAn register.

During the waiting period, the waiting is released by writing the IICAn register to start transferring data. The IICAn register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register changes to “00H”.

Figure 14-3 Format of IICAn shift register n (IICAn)

Bit	Symbol	Description	Reset value
7:0	IICAn	IICAn shift register n	0x0

Note 1: During data transfer, no data can be written to IICAn registers.

Note 2: IICAn registers can only be read and written during the wait period. Access to the IICAn register is prohibited in the communication state except for the waiting period. However, in the case of a master device, the IICAn register can be written once after setting the communication trigger bit (STTn) to 1.

Note 3: When making a reservation for communication, data must be written to the IICAn register after detecting an interrupt caused by a stop condition.

Remark: n=0.

(2) Slave address register n (SVAn)

This is a register that holds the 7-bit local station address {A6, A5, A4, A3, A2, A1, A0} when used as a slave.

The SVAn register is set by an 8-bit memory manipulation instruction. However, rewriting this register is prohibited when the STDn bit is 1 (start condition detected).

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

Figure 14-4 Format of the slave address register n (SVAn)

Address: 0x40041A34	After rest: 00H			R/W			
Symbol	7	6	5	4	3	2	1 0
SVAn	A6	A5	A4	A3	A2	A1	A0 0 ^{Note}

Note: bit0 is fixed to “0”.

(3) SO latch

The SO latch holds the output level of the SDAAn pin.

(4) Wake-up control circuit

The circuit generates an interrupt request (INTIICAn) when the address value set in the slave address register n (SVAn) is the same as the received address.

(5) Serial clock counter

During transmission or reception, this counter counts the output or input serial clocks and checks whether 8-bit data transmission and reception are performed.

(6) Interrupt request signal generation circuit

This circuit controls the generation of an interrupt request signal (INTIICAn).

An I²C interrupt request is generated by the following 2 triggers.

- Falling on the 8th or 9th serial clock (set by the WTIMn bit)
- An interrupt request is generated due to the detection of a stop condition (set by the SPIEn bit).

Note 1: WTIMn bit: bit3 of IICA control register n0 (IICCTLn0)

Note 2: SPIEn bit: bit4 of IICA control register n0 (IICCTLn0)

(7) Serial clock control circuit

In the master mode, the circuit generates a clock output to the SCLAn pin from the sampling clock.

(8) Serial clock wait control circuit

This circuit controls the wait timing.

(9) Ack generation circuit, stop condition detection circuit, start condition detection circuit, Ack detection circuit

These circuits generate and detect various states.

(10) Data hold time correction circuit

This circuit generates a data hold time against a falling serial clock.

(11) Start condition generation circuit

If the STTn bit is set to 1, this circuit generates a start condition.

However, in the state where reservation communication is prohibited (IICRSVn bit = 1) and the bus is not released (IICBSYn bit = 1), the start condition request is ignored and the STCFn bit is set to 1.

(12) Stop condition generation circuit

If the SPTn bit is set to 1, this circuit generates a stop condition.

(13) Bus status detection circuit

This circuit detects whether the bus is released by detecting the start and stop conditions. However, the bus status cannot be detected immediately when it is just operated, so the initial state of the bus status detection circuit must be set by the STCENn bit.

Remarks 1. STTn bit: bit1 of IICA control register n0 (IICCTLn0)

SPTn bit: bit0 of IICA control register n0 (IICCTLn0)

IICRSVn bit: bit0 of IICA flag register n (IICFn)

IICBSYn bit: bit6 of IICA flag register n (IICFn)

STCFn bit: bit7 of IICA flag register n (IICFn)

STCENn bit: bit1 of IICA flag register n (IICFn)

2. n=0.

14.3 Registers for Controlling Serial Interface IICA

The serial interface IICA is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- IICA control register n0 (IICCTLn0)
- IICA flag register n (IICFn)
- IICA status register n (IICSn)
- IICA control register n1 (IICCTLn1)
- IICA low level width setting register n (IICWLn)
- IICA high level width setting register n (IICWHn)
- Port mode register (PMxx)
- Port mode control register (PMCxx)
- Port alternate function configuration register (PxxCFG)

Remark: n=0.

14.3.1 Peripheral Enable Register 0 (PER0)

The PER0 register is used to enable or disable the clock supplied to various peripheral hardware. By stopping the clock to unused hardware, it helps reduce power consumption and noise.

To use the serial interface IICAn, set the bit4 (IICAEN) to 1.

The PER0 register is set by the 8-bit memory manipulation instruction.

After generating a reset signal, the value of this register changes to “00H”.

Figure 14-5 Format of peripheral enable register 0 (PER0)

Address: 40020420H	After reset: 00H		R/W							
Symbol	7	6	5	4	3	2	1	0		
PER0	RTCEN	IRDAEN	ADCEN	IICAEN	SCI1EN	SCI0EN	TM41EN	TM40EN		
IICAnEN		Control of IICA input clock supply								
0		Stop to supply the input clock • The SFR used by the IICA cannot be written. • The serial interface IICA is in the reset status.								
1		Supply the input clock • The SFR used by the IICA can be written.								

Caution 1: To set the serial interface IICA, the following registers must be set when the IICAnEN bit is 1. When the IICnEN bit is 0, the control registers of the serial interface IICA are initialized and write operations are ignored (except for the port alternate function configuration register (PxxCFG), the port mode register (PMxx) and the port mode control register (PMCxx)).

- IICA control register n0 (IICCTLn0)
- IICA flag register n (IICFn)
- IICA status register n (IICSn)
- IICA control register n1 (IICCTLn1)
- IICA low-level width setting register n (IICWLn)
- IICA high-level width setting register n (IICWHn)

Remark: n=0.

14.3.2 IICA Control Register n0 (IICCTLn0)

This is the register that enables or stops I²C operation, sets wait timings, and sets other I²C operations.

The IICCTLn0 register is set by an 8-bit memory manipulation instruction. However, the SPIEn bit, the WTIMn bit, and the ACKEn bit must be set when the ICEn bit is 0 or during the wait period, and these bits can be set simultaneously when the ICEn bit is set from 0 to 1.

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

Note: n=0.

Figure 14-6 Format of IICA control register n0(IICCTLn0) (1/4)

Address: 0x40041A30				After reset: 00H		R/W		
Symbol	7	6	5	4	3	2	1	0
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	I ² C operation enable
0	Stop operation. Reset the IICA status register n (IICSn) ^{Note 1} and stop internal operation.
1	Enable operation.
This bit set to 1 in the state where the SCLAn and SDAAn lines are high.	
Clear condition (IICEn=0)	Set condition (IICEn=1)
• Cleared by instructions. • When resetting	• Set by instructions

LRELn ^{Note2,3}	Exit of communication
0	Normal operation
1	Exit the current communication and enter idle status. Automatically clear 0 after execution. Use in cases such as receiving extension codes that are not related to the local station. The SCLAn line and the SDAAn line become high impedance. The following flags in IICA control register n0 (IICCTLn0) and IICA status register n (IICSn) are cleared to 0. •STTn•SPTn•MSTSn•EXCn•COIn•TRCn•ACKDn•STDn
Enters idle state to exit the communication until the following communication participation conditions are met. • Boot as master device after detecting a stop condition. • Addresses match or extended codes are received after the start condition is detected	
Clear condition (LRELn=0)	Set condition (LRELn=1)
• Automatically clears after execution. • When resetting	• Set by instructions

WRELn ^{Note2,3}	Release waiting
0	The wait is not released.
1	Release wait. Automatically clears after the wait is released.
If the WRELn bit (release from wait) is set during the 9th clock wait in the transmit state (TRCn=1), the SDAAn line changes to a high impedance state (TRCn=0).	
Clear condition (WRELn=0)	Set condition (WRELn=1)
• Automatically cleared after execution. • When resetting	• Set by instructions

Note 1: Reset the STCFn bit and IICBSYn bit of the IICA shift register n (IICAn), the IICA flag register n (IICFn), and the CLDn bit and DADn bit of the IICA control register n1 (IICCTLn1).

Note 2: If the ICEn bit is 0, the signal of this bit is invalid.

Note 3: The LRELn bit and the WRELn bit are always read as 0.

Caution: If I²C operation is enabled (IICCTLn=1) when the SCLAn line is high, the SDAAn line is low and the digital filter is ON (DFCn=1 in the IICCTLn1 register), the start condition is detected immediately. In this case, the LRELn bit must be set to 1 consecutively by a bit memory manipulation instruction after I²C operation is enabled (ICEn=1).

Remark: n=0

Figure 14-6 Format of IICA control register n0(IICCTLn0) (2/4)

SPIEn ^{Note1}	Enable or disable interrupt requests generated by stop condition detection
0	Disable
1	Enable
When the WUPn bit of IICA control register n1 (IICCTLn1) is 1, even if the SPIEN is 1, there is also no stop condition interrupt.	
Clear condition (SPIEn=0)	
Set condition (SPIEn=1)	
• Cleared by instructions. • When resetting	• Set by instructions

WTIMn ^{Note1}	Control of wait and interrupt requests
0	An interrupt request signal is generated on the falling edge of the 8th clock. Master: After outputting 8 clocks, set the clock output low to wait. Slave: After inputting 8 clocks, set the clock low and wait for the master.
1	An interrupt request signal is generated on the falling edge of the 9th clock. Master: After outputting 9 clocks, set the clock output low to wait. Slave: After inputting 9 clocks, set the clock low and wait for the master.
An interrupt is generated on the falling edge of the 9th clock during the address transfer, regardless of the setting of this bit; the setting of this bit is valid at the end of the address transfer. The master device enters the wait state on the falling edge of the 9th clock during address transmission. A slave device that receives a local station address enters the wait state on the falling edge of the 9th clock after an acknowledge (ACK) is generated, but a slave device that receives an extension code enters the wait state on the falling edge of the 8th clock.	
Clear condition (WTIMn=0)	
Set condition (WTIMn=1)	
• Cleared by instructions. • When resetting	• Set by instructions

ACKEn ^{Note1,2}	ACK control
0	No ack.
1	Enable ACK. Set the SDAAn line low during the 9th clock.
Clear condition (ACKEn=0)	
Set condition (ACKEn=1)	
• Cleared by instructions. • When resetting	• Set by instructions

Note 1: The signal of this bit is invalid when the ICEn bit is 0. This bit must be set during this period.

Note 2: The set value is invalid when it is not an extension code during address transfer. When it is a slave device and the address matches, an ACK is generated regardless of the set value.

Remark: n=0.

Figure 14-6 Format of IICA control register n0(IICCTLn0) (3/4)

STTn ^{Note1,2}	Triggering of the start condition
0	No start conditions are generated.
1	When the bus is released (idle, IICBSYn bit is 0): If this bit is 1, a start condition (boot as the master device) is generated. When a third party is communicating: • When the communication reservation function is enabled (IICRSVn=0). Used as a start condition reservation flag. If this bit set to 1, a start condition is automatically generated just after the bus is released. • When the communication reservation function is prohibited (IICRSVn=1). Even if this bit is 1, the STTn bit is cleared and the STTn clear flag (STCFn) is set to 1 without generating a start condition. Wait status (master device): A restart condition is generated after the wait is released.
Cautions on setting timing: • Master reception: Disable setting this bit to 1 during transmission. This bit can only be set to 1 during the waiting period when the ACKEn is 0 and notifying the slave reception it has completed. • Master transmission: During the Ack, the start conditions may not be generated properly. This bit must be set to 1 during the wait period after the 9th clock is output. • It is prohibited to set 1 at the same time as the stop condition trigger (SPTn). • After setting the STTn bit to 1, it is prohibited to set this bit to 1 again before the clear condition is satisfied.	
Clear condition (STTn=0)	Set condition (STTn=1)
• Set the STTn bit to 1 in the state where communication reservation is disabled. • When arbitration fails • Master device generates start conditions. • Cleared due to the LRELn bit is 1 (Exit Communication). • When the IICEn bit is 0 (stop running). • When resetting	• Set by instructions

Notes 1: In the state where the IICEn bit is 0, the signal for this bit is invalid.

2: The STTn bit is always read as 0.

Remarks:

1. If bit1 (STTn) is read after setting the data, this bit becomes 0.
2. IICRSVn: Bit0 of IICA flag register n (IICFn)
STCFn: Bit7 of IICA flag register n (IICFn)
3. n=0

Figure 14-6 Format of IICA control register n0(IICCTLn0) (4/4)

SPTn ^{Note}	Triggering of the stop condition
0	No stop conditions are generated.
1	Generate a stop condition (end of transfer as master).
Cautions on setting timing: • Master reception: Disable setting this bit to 1 during transmission. This bit can only be set to 1 during the waiting period when ACKEn is at 0 and notifying the slave reception has completed. • Master transmission: During the Ack, the stop conditions may not be generated properly. This bit must be set to 1 during the wait period after the 9th clock is output. • Prohibit setting 1 at the same time as the start condition trigger (STTn). • When the WTIMn bit is 0, it must be noted that if the SPTn bit is set to 1 during the wait period after 8 clocks of output, a stop condition is generated during the high level of the 9th clock after the wait is released. The WTIMn bit must be set from 0 to 1 during the wait period after 8 clocks of output and the SPTn bit must be set to 1 during the wait period after the 9th clock of output. • After setting the SPTn bit to 1, it is prohibited to set this bit to 1 again until the clear condition is satisfied.	
Clear condition (SPTn=0)	Set condition (SPTn=1)
• When arbitration fails • Cleared automatically when a stop condition is detected. • Cleared due to the LRELn bit is 1 (exit communication). • When the IICEn bit is 0 (stop running). • When resetting	• Set by instructions

Note: The SPTn bit is always read as 0.

Caution: If bit3 (TRCn) of the IICA status register n (IICSn) is 1 (transmit state), the wait is released by setting bit5 (WRELn) of the IICCTLn0 register to 1 on the 9th clock, the SDAAn line is set to high impedance after clearing the TRCn bit (receive state). The wait release must be performed by writing to the IICA shift register n when the TRCn bit is 1 (transmit state).

Remark: n=0

14.3.3 IICA Status Register n (IICS_n)

This is a register that represents the I²C status.

The IICS_n register can be read by an 8-bit memory manipulation instruction only when the STT_n bit is 1 and waiting. After the reset signal is generated, the value of this register changes to “00H”.

Caution: In the enabled address matching wake function (WUP_n=1) state in deep sleep mode, reading the IICS_n register is prohibited. In the state where the WUP_n bit is 1, it has nothing to do with the INTIICAn interrupt request if you change the WUP_n bit from 1 to 0 (stop wake-up operation) reflects a change in state until the next start condition or stop condition is detected. Therefore, to use the wake-up function, it is necessary to enable (SPIEn=1) an interrupt due to the detection of a stop condition, and to read the IICS_n register after an interrupt is detected.

Remarks 1: STT_n: bit1 of IICA control register n0 (IICCTLn0), n=0.

2: WUP_n: bit7 of IICA control register n1 (IICCTLn1), n=0.

Figure 14-7 Format of IICA status register n (IICS_n) (1/3)

Symbol	7	6	5	4	3	2	1	0
IICS _n	MSTS _n	ALD _n	EXC _n	COIn	TRC _n	ACKD _n	STD _n	SPD _n

MSTS _n	Acknowledgement flag of master control status
0	Slave or communication idle status
1	Master communication status
Clear condition (MSTS _n =0)	
Set condition (MSTS _n =1)	
- When a stop condition is detected - When the ALD_n bit is 1 (arbitration fails). - Cleared due to the LREL_n bit is 1 (exit communication). - When the IICEn bit changes from 1 to 0 (stop running). - When resetting	When generating a start condition

ALD _n	Detection of arbitration failures
0	It indicates that arbitration did not occur or was won.
1	It indicates an arbitration failure. Clear the MSTS _n bit.
Clear condition (ALD _n =0)	
Set condition (ALD _n =1)	
- Automatically cleared after reading the IICS_n register Note. - When the IICEn bit changes from 1 to 0 (stop running). - When resetting	When arbitration fails

Note: This bit is cleared even if the bit memory manipulation instruction is executed on a bit other than the IICS_n register.

Therefore, when using ALD_n bits, the data for the ALD_n bits must be read before reading other bits.

Remarks:

- LREL_n: Bit6 of IICA control register n0 (IICCTLn0)
IICEn: Bit7 of IICA control register n0 (IICCTLn0)
- n=0

Figure 14-7 Format of IICA status register n (IICS_n) (2/3)

EXC _n	Reception detection of extended codes	
0	No extension code is received.	
1	An extension code is received.	
Clear condition (EXC _n =0)		Set condition (EXC _n =1)
- When a start condition is detected - When a stop condition is detected - Cleared due to the LREL_n bit is 1 (exit communication) - When the IICEn bit changes from 1 to 0 (stop running) - When resetting		When the high 4 bits of the received address data are “0000” or “1111”. (Set on the rising edge of the 8th clock)
COIn	Detection of address matching	
0	The addresses are different.	
1	The address is the same.	
Clear condition (COIn=0)		Set condition (COIn=1)
- When a start condition is detected - When a stop condition is detected - Cleared due to the LREL_n bit is 1 (exit communication). - When the IICEn bit changes from 1 to 0 (stop running). - When resetting		When the receive address and the local station address (Slave Address Register n (SVAn)) are the same (set on the rising edge of the 8th clock)
TRC _n	Transmit/receive status detection	
0	In the receive state (except in the transmit state). Set the SDAAn line to high impedance.	
1	In the transmitting state. Set to output the value of the SON latch to the SDAAn line (valid after the falling edge of the 9th clock byte of the 1st byte).	
Clear condition (TRC _n =0) <Master and slave> - When a stop condition is detected - Cleared due to the LREL_n bit is 1 (exits communication). - When the IICEn bit changes from 1 to 0 (stops operation). - Cleared because the WREL_n bit is 1 (released from wait) Note - When the ALD_n bit changes from 0 to 1 (arbitration fails) - When resetting - When not participating in communications (MSTSn, EXC_n, COIn=0) <Master> - When 1 is output in the LSB (transfer direction indication bit) of the 1st byte <Slave> - When a start condition is detected - When 0 is input to the LSB (transfer direction indication bit) of the 1st byte		Set condition (TRC _n =1) <Master> - When generating a start condition - When the LSB (transmission direction indicator bit) of byte 1 (address transmission) outputs 0 (master transmission). <Slave> - When the LSB (transmission direction indication bit) in byte 1 (address transmission) of the master device inputs 1 (slave transmission)

Note: If bit3 (TRC_n) of the IICA status register n (IICS_n) is 1 (transmit state) and bit5 (WREL_n) of the IICA control register n0 (IICCTLn0) is set to 1 on the 9th clock to release the wait, the SDAAn line is set to high impedance after clearing the TRC_n bit (receive state). The wait release must be performed by writing the IICA shift register n when the TRC_n bit is 1 (transmit state).

Remarks:

- LREL_n: Bit6 of IICA control register n0 (IICCTLn0)
IICEn: Bit7 of IICA control register n0 (IICCTLn0)
- n=0

Fiugre 14-7 Format of IICA status register n (IICSn) (3/3)

ACKDn	Detection of acknowledge	
0	No Ack is detected.	
1	An Ack is detected.	
Clear condition (ACKDn=0)		Set condition (ACKDn=1)
• When resetting • When a stop condition is detected • When the next byte of the 1st clock rises • Cleared because the LRELn bit is 1 (exit communication). • When the IICEn bit changes from 1 to 0 (stop operation). • When resetting		• When the SDAAn line is set low on the 9th clock rising edge of the SCLAn line

STDn	Detection of starting conditions	
0	No start condition detected.	
1	A start condition is detected, indicating that it is during address transfer.	
Clear condition (STDn=0)		Set condition (STDn=1)
• When a stop condition is detected • When the 1st clock rises after the next byte of the address is transmitted • Cleared because the LRELn bit is 1 (exits communication). • When the IICEn bit changes from 1 to 0 (stops operation). • When resetting		• When a start condition is detected

SPDn	Detection of stopping conditions	
0	No stop condition detected.	
1	A stop condition is detected, the master device ends communication and the bus is released.	
Clear condition (SPDn=0)		Set condition (SPDn=1)
• After setting this bit, the address transmits the byte after the start condition is detected when the clock rises • When the WUPn bit changes from 1 to 0 • When the IICEn bit changes from 1 to 0 (stop operation). • When resetting		• When a stop condition is detected

Remarks:

1. LRELn: Bit6 of IICA control register n0 (IICCTLn0)
IICEn: Bit7 of IICA control register n0 (IICCTLn0)
2. n=0

14.3.4 IICA Flag Register n (IICFn)

This is a register that sets the I²C operating mode and indicates the status of the I²C-bus.

The IICFn register is set by an 8-bit memory manipulation instruction. However, only the STTn clear flag (STCFn) and I²C bus status flag (IICBSYn) can be read.

The communication reservation function is enabled or disabled by the IICRSVn bit setting, and the initial value of the IICBSYn bit is set by the STCENn bit. The IICRSVn bit and the STCENn bit can be written only when I²C operation is disabled (bit7(IICENn)=0 of the IICA control register n0 (IICCTLn0)). Only the IICFn register can be read after operation is allowed. The value of this register changes to “00H” after the reset signal is generated.

Figure 14-8 Format of IICA flag register n(IICFn)

Address: 0x40041B52		After reset: 00H				R/W ^{Note}		
Symbol	7	6	5	4	3	2	1	0
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn
STCFn	STTn clear flag							
0	Release start conditions.							
1	The STTn flag cannot be cleared while the start condition cannot be issued.							
Clear condition (STCFn=0)				Set condition (STCFn=1)				
• Cleared due to the STTn bit is 1 • When the IICEn bit is 0 (stop operation). • When resetting				• If the STTn bit is cleared to 0 because the start condition cannot be issued in the state where communication reservation is disabled (IICRSVn=1).				
IICBSYn	I ² C bus status flag							
0	Bus release state (initial state of communication when STCENn=1)							
1	Bus communication state (initial state of communication when STCENn=0)							
Clear condition (IICBSYn=0)				Set condition (IICBSYn=1)				
• When a stop condition is detected • When the ICEn bit is 0 (stop operation) • When resetting				• When a start condition is detected • Sets the IICEn bit when the STCENn bit is 0				
STCENn	Initial start enable triggering							
0	After enabling operation (IICEn=1), a start condition is allowed to be generated by detecting a stop condition.							
1	After enabling operation (IICEn=1), the start condition is allowed to be generated without detecting a stop condition.							
Clear condition (STCENn=0)				Set condition (STCENn=1)				
• Cleared by intructions. • When a start condition is detected • When resetting				• Set by intructions				
IICRSVn	Communication reservation function disable bit							
0	Communication appointments are enabled.							
1	Communication reservations are disabled.							
Clear condition (IICRSVn=0)				Set condition (IICRSVn=1)				
• Cleared by intructions. • When resetting				• Set by intructions				

Note 1: bit6 and bit7 are read-only bits.

Caution 1: The STCENn bit can only be written when the operation is stopped (IICEn=0).

Caution 2: If the STCENn bit is 1, the bus is considered to be released (IICBSYn=0) regardless of the actual bus state, so it is necessary to confirm that there is no third party in communication in order to avoid disrupting other communications when the first start condition (STTn=1) is issued.

Caution 3: The IICRSVn can be written only when it is stopped (IICEn=0).

Remark 1: STTn: bit1 of IICA control register n0 (IICCTLn0).

Remark 2: IICEn: bit7 of IICA control register n0 (IICCTLn0).

14.3.5 IICA Control Register n1 (IICCTLn1)

This is a register used to set the I²C operation mode and detect the status of the SCLAn and SDAAn pins.

The IICCTLn1 register is set by an 8-bit memory manipulation instruction. However, only CLDn and DADn bits can be read.

In addition to the WUPn bit, the IICCTLn1 register must be set when I²C operation is disabled (bit7 (IICEn) = 0 of the IIC control register n0 (IICCTLn0)).

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

Figure 14-9 Format of IICA control register n1 (IICCTLn1) (1/2)

Address: 0x40041A31

After reset: 00H

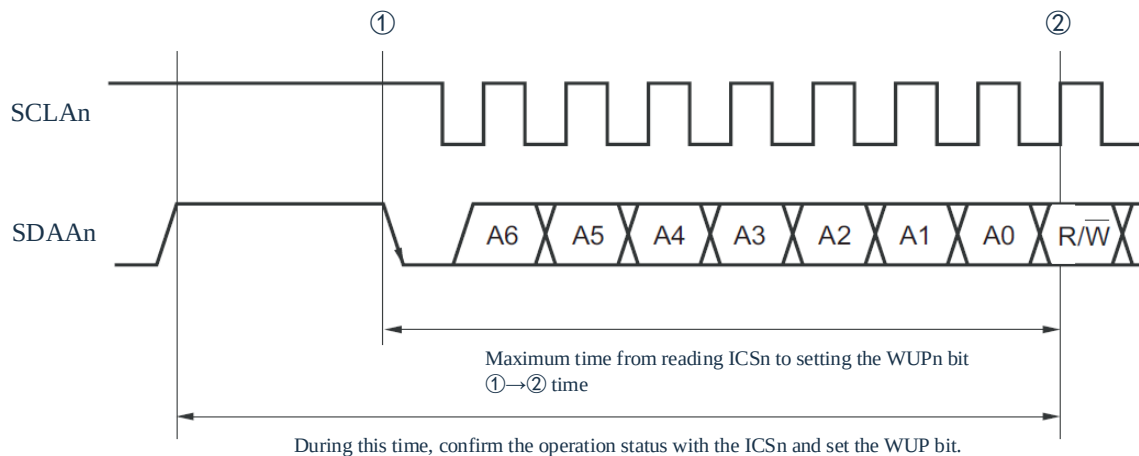
R/W^{Note1}

Symbol	7	6	5	4	3	2	1	0
IICCTLn1	WUPn	0	CLDn	DADn	SMCn	DFCn	0	PRSn

WUPn	Control of address matching wakeup
0	Stop the operation of the Address Match Wakeup function in deep sleep mode.
1	Enable the operation of the Address Match Wakeup function in deep sleep mode.
To transfer to deep sleep mode by setting the WUPn bit to 1, at least three FMCK clocks must pass after setting the WUPn bit to 1, and then the deep sleep instruction must be executed (refer to Figure 14-28 Flow when setting the WUPn bit to 1). The WUPn bit must be cleared to 0 after the address is matched or the extension code is received. It is possible to participate in subsequent communication by clearing the WUPn bit to 0 (it is necessary to release the wait and write send data after clearing the WUPn bit to 0). In the state of WUPn bit 1, the interrupt timing when the address is matched or the extension code is received is the same as the interrupt timing when WUPn bit is 0. (The delay difference of sampling error is generated according to the clock). In addition, when the WUPn bit is 1, even if the SPIEn bit is set to 1, no stop condition interrupt is generated.	
Clear condition (WUPn=0)	
Set condition (WUPn=1)	
• Cleared by instructions (after an address match or reception of an extension code). • Set by instructions (MSTSn=0, EXCn=0, COIn=0 and STDn=0 (does not participate in communication)) ^{Note 2}.	

Note 1: bit4 and bit5 are read-only bits.

Note 2: During the period shown below, it is necessary to check the status of the IIC status register n (IISn) and set it.



Remark: n=0

Figure 14-9 Format of IICA control register n1 (IICCTLn1) (2/2)

CLDn	Level detection of the SCLAn pin (valid only when the IICEn bit is 1)	
0	SCLAn pin is detected low.	
1	SCLAn pin is detected high.	
Clear condition (CLDn=0)		Set condition (CLDn=1)
• When the SCLAn pin is low • When the IICEn bit is 0 (stop operation) • When resetting		• When the SCLAn pin is high

DADn	Level detection on the SDAAn pin (valid only when the IICEn bit is 1)	
0	SDAAn pin is detected as low.	
1	SDAAn pin is detected as high.	
Clear condition (DADn=0)		Set condition (DADn=1)
• When the SDAAn pin is low • When the IICEn bit is 0 (stop operation) • When resetting		• When the SDAAn pin is high

SMCn	Switching of operating modes	
0	Operate in standard mode (maximum transmission rate: 100kbps).	
1	Operate in Fast Mode (maximum transfer rate: 400kbps) or Enhanced Fast Mode (maximum transfer rate: 1Mbps).	

DFCn	Operation control of digital filters	
0	Digital filter OFF	
1	Digital filter ON	
Digital filters must be used in fast mode or enhanced fast mode. Digital filters are used to eliminate noise. Whether the DFCn is 1 or 0, the transmission clock is unchanged.		

PRSn	Control of the operation clock (F _{MCK})	
0	Selects F _{CLK} (1MHz≤F _{CLK} ≤20MHz).	
1	Selects F _{CLK} /2 (20MHz<F _{CLK}).	

Caution 1: The maximum operating frequency of the IICA running clock (f_{MCK}) is 20MHz (Max.). Bit0 (PRSn) of the IICA control register n1 (IICCTLn1) must be set to “1” only if f_{CLK} exceeds 20MHz.

Caution 2: When setting the transfer clock, attention must be paid to the minimum operating frequency of F_{CLK}. The minimum operating frequency of the F_{CLK} of the serial interface IICA depends on the mode of operation.

Fast mode: f_{CLK}=3.5MHz(Min.)

Enhanced fast mode: f_{CLK}=10MHz(Min.)

Standard mode: f_{CLK}=1MHz(Min.)

Remark 1: IICEn: bit7 of IICA control register n0 (IICCTLn0)

Remark 2: n=0.

14.3.6 IICA Low Level Width Setting Register n (IICWLn)

This register controls the SCLAn pin signal low level width (t_{LOW}) and the SDAAn pin signal output by the serial interface IICA.

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

The IICWHn register must be set when I²C operation is disabled (bit7(IICWHn)=0 of IICA control register n0 (IICCTLn0)).

After a reset signal is generated, the value of this register changes to FFH.

For IICWLn register set-up methods, refer to 14.4.2 Setting transfer clock via IICWLn and IICWHn registers.

The data retention time is 1/4 of the time set by IICWLn.

Figure 14-10 Format of IICA low-level width setting register n (IICWLn)

Address: 0x40041A32				After reset: FFH		R/W		
Symbol	7	6	5	4	3	2	1	0
IICWLn								

14.3.7 IICA High Level Width Setting Register n (IICWHn)

This register controls the width of the SCLAn pin signal high and the SDAAn pin signal for the serial interface IICA output. The IICWHn register is set by an 8-bit memory manipulation instruction.

The IICWHn register must be set when I²C operation is disabled (bit7(IICWHn)=0 of IICA control register n0 (IICCTLn0)). After a reset signal is generated, the value of this register changes to “FFH”.

Figure 14-11 Format of high-level width setting register n(IICWHn)

Address: 0x40041A33				After reset: FFH		R/W		
Symbol	7	6	5	4	3	2	1	0
IICWHn								

Remark 1: For the method of setting the master transfer clock, please refer to 14.4.2(1); for how to set the slave IICWLn register and the IICWHn register, refer to 14.4.2(2).

Remark 2: n=0.

14.3.8 Registers for Controlling Pin Port Functions of IICA

This product allows the pin functions of IICA to be alternated to any port other than P00.

By configuring the port alternate function configuration register (PxxCFG), the SCALn pin and SDAAn pin can be assigned to two different ports.

Set the corresponding bits in the port mode control register (PMCxx) and port mode register (PMxx) to 0 for these two ports.

After these two ports are configured for IICA pin alternating, the N-channel open-drain output mode (VDD/EVDD withstand voltage) is automatically enabled by design, meaning the POMxx register does not need to be configured by the user.

For detailed configuration methods, refer to Chapter 2: Pin Functions.

14.4 Functions of I²C Bus Mode

14.4.1 Pin Structure

The serial clock pin (SCLAn) and the serial data bus pin (SDAAn) are structured as follows.

- (1) SCLAn: Input/output pins of serial clock

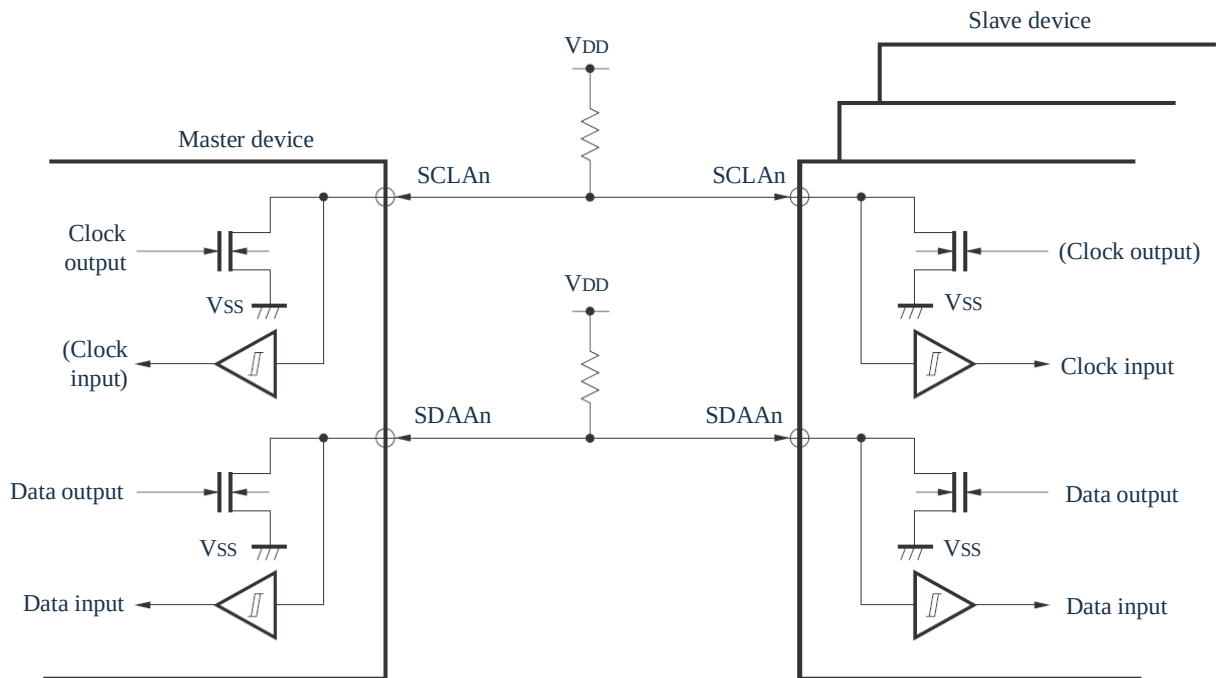
The outputs of both the master and slave devices are N-channel open-drain outputs, and the inputs are Schmitt inputs

- (2) SDAAn: Input/output pins of serial data

The outputs of both the master and slave devices are N-channel open-drain outputs, and the inputs are Schmitt inputs.

Because the outputs of the serial clock line and serial data bus are N-channel open-drain outputs, an external pull-up resistor is required.

Figure 14-12 Pin structure diagram



Remark: n=0.

14.4.2 Setting Transfer Clock via IICWL_n and IICWH_n Registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{F_{MCK}}{IICWL + IICWH + F_{MCK}(T_R + T_F)}$$

At this point, the optimal setpoints for the IICWL_n register and the IICWH_n register are as follows:
(The fractional parts of all setting values are rounded up.)

① Fast mode

$$IICWL_n = \frac{0.52}{\text{transfer clock}} \times F_{MCK}$$

$$IICWH_n = \left(\frac{0.48}{\text{transfer clock}} - t_R - t_F \right) \times F_{MCK}$$

② Standard mode

$$IICWL_n = \frac{0.47}{\text{transfer clock}} \times F_{MCK}$$

$$IICWH_n = \left(\frac{0.53}{\text{transfer clock}} - t_R - t_F \right) \times F_{MCK}$$

③ Enhanced fast mode

$$IICWL_n = \frac{0.50}{\text{transfer clock}} \times F_{MCK}$$

$$IICWH_n = \left(\frac{0.50}{\text{transfer clock}} - t_R - t_F \right) \times F_{MCK}$$

(2) Setting IICWL_n and IICWH_n registers on slave side

(The fractional parts of all setting values are rounded up.)

① Fast mode

$$IICWL_n = 1.3 \mu s \times F_{MCK}$$

$$IICWH_n = (1.2 \mu s - t_R - t_F) \times F_{MCK}$$

② Standard mode

$$IICWL_n = 4.7 \mu s \times F_{MCK}$$

$$IICWH_n = (5.3 \mu s - t_R - t_F) \times F_{MCK}$$

③ Enhanced fast mode

$$IICWL_n = 0.50 \mu s \times F_{MCK}$$

$$IICWH_n = (0.50 \mu s - t_R - t_F) \times F_{MCK}$$

Cautions:

1. The maximum operating frequency of the IICA operating clock (F_{MCK}) is 20MHz (Max.). The bit0 (PRSn) of the IICA control register n1 (IICCTLn1) must be set to 1 only when the F_{CLK} exceeds 20MHz.
2. Note the minimum F_{CLK} operation frequency when setting the transfer clock. The minimum F_{CLK} operation frequency for serial interface IICA is determined according to the mode.

Fast mode: $F_{CLK} = 3.5\text{MHz}$ (Min.)

Enhanced fast mode: $F_{CLK} = 10\text{MHz}$ (Min.)

Standard mode: $F_{CLK} = 1\text{MHz}$ (Min.)

Remarks

1. Calculate the rise time (T_R) and fall time (T_F) of the SDA_n and SCL_n signals separately, because they differ depending on the pull-up resistance and wire load.
2. IICWL_n: IICA low-level width setting register n
IICWH_n: IICA high-level width setting register n

T_F : SDAAn and SCLAn signal falling times

T_R : SDAAn and SCLAn signal rising times

F_{MCK} : IICA operation clock frequency

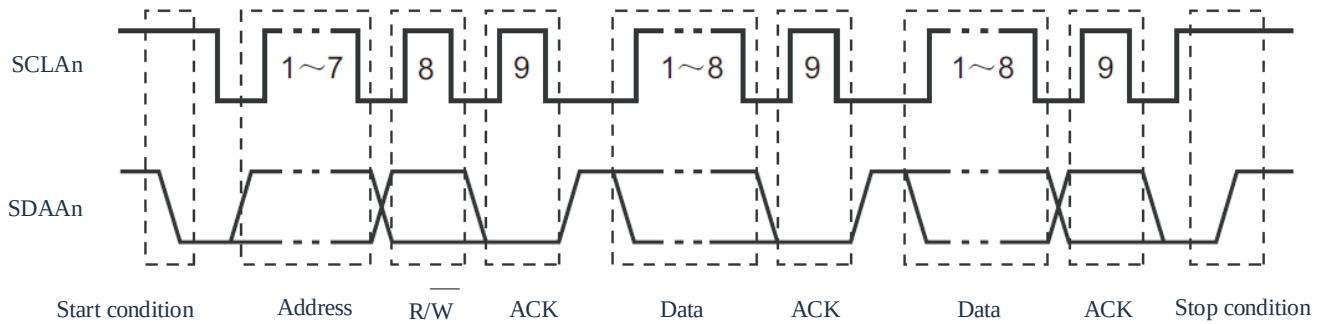
3. $n=0$

14.5 Definition and Control Method of I²C bus

The following section describes the I²C bus's serial data communication format and the signals used by the I²C bus.

The figure below shows the transfer timing for the “start condition”, “address”, “data”, and “stop condition” output via the I²C bus's serial data bus.

Figure 14-13 I²C bus serial data transfer timing



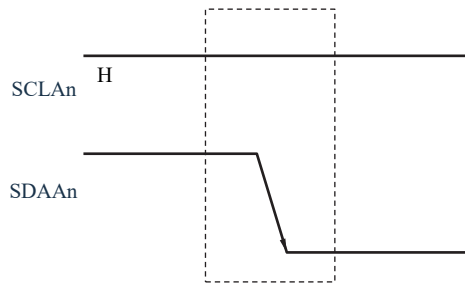
The master generates start conditions, slave addresses, and stop conditions.

Both the master and slave devices can generate an acknowledgement (ACK) (in general, the receiver outputs 8 bits of data). The master device continuously outputs a serial clock (SCLAn). However, the slave can extend the low-level period of the SCLAn pin and insert a wait.

14.5.1 Start Condition

When the SCLAn pin is high, if the SDAAn pin changes from high to low, a start condition is generated. The SCLAn pin and the SDAAn pin start conditions are the signals generated when the master device starts serial transfer to the slave device. When used as a slave, a start condition is detected.

Figure 14-14 Start condition



If bit1 (STTn) of the IICA control register n0 (IICCTLn0) is set to “1” when a stop condition (SPDn: bit0=1 of the IICA status register n (IICSn)) is detected, the start condition is output. If a start condition is detected, set bit1 (STDn) of the IICSn register to “1”.

Remark: n=0.

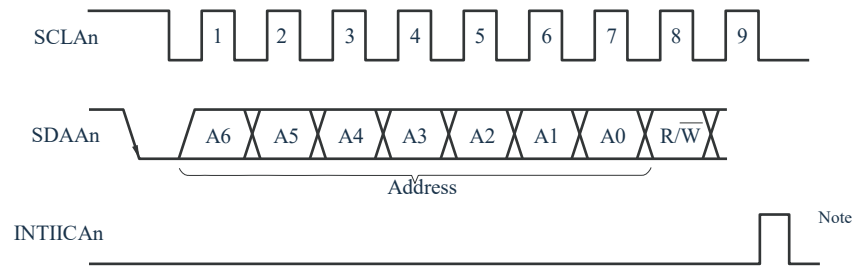
14.5.2 Address

The subsequent 7-bit data for the start condition is defined as an address.

The address is a 7-bit data output by the master device to select a specific slave device from multiple connected slave devices on the bus. Therefore, the slave device on the bus needs to set a completely different address.

The slave device detects the start condition by hardware and checks whether the 7-bit data is the same as the slave address register n (SVAn). At this time, if the 7-bit data and the SVAn register have the same value, the slave device is selected to communicate with the master device before generating a start condition or a stop condition.

Figure 14-15 Address



Note: If data other than the local station address or extension code is received while the slave is running, INTIICAn is not generated.

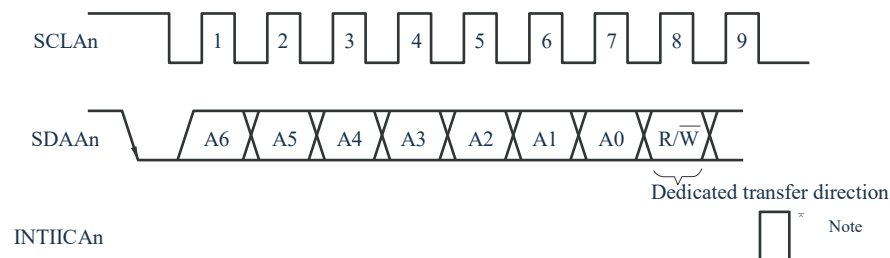
If the 8-bit data consisting of the slave address and the transfer direction described in “14.5.3 Transfer direction specification” is written to the IICA shift register n (IICAn), the address is output. The received address is written to the IICAn register. The slave address is assigned to the higher 7 bits of the IICAn register.

14.5.3 Transfer Direction Specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction.

When this transfer direction specification bit is “0”, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit is “1”, it indicates that the master device is receiving data from a slave device.

Figure 14-16 Transfer direction specification



Note: The INTIICAn is not generated if data other than a local address or extension code is received during slave device operation.

Remark: n=0.

14.5.4 Acknowledge (ACK)

The serial data status of the sender and receiver can be confirmed by an acknowledgement (ACK). The receiver returns an ACK each time it receives an 8-bit data.

Typically, the sender receives an ACK after sending an 8-bit data. When the receiver returns the ACK, it is deemed to have been received normally and continues processing. The bit2 (ACKDn) of the IICA status register n (IICSn) can confirm the detection of the ACK. When the master receives the last data for the received state, a stop condition is generated without returning an ACK. When the slave does not return an ACK after receiving the data, the master device outputs a stop condition or a restart condition to stop the transmission. The reasons why an ACK is not returned are as follows:

- ① Reception is not performed normally.
- ② The reception of the last data has been terminated.
- ③ The receiver specified by the address does not exist.

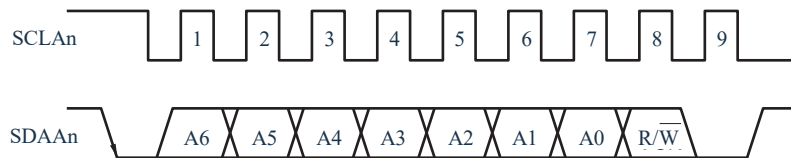
The receiver generates an ACK (normal reception) by setting the SDAAn line low on the 9th clock.

Set bit2 (ACKEn) of IICA control register n0 (IICCTLn0) to 1 to enable automatic generation of an ACK. Set bit3 (TRCn) of the IICsn register with the 8th bit of data following the 7-bit address information. In the case of reception (TRCn=0), the ACKEn bit must be set to 1 in general.

When data cannot be received during slave receive operation (TRCn=0) or the next data is not required, the ACKEn bit must be cleared to 0 to notify the master that data cannot be received.

When the master does not need the next data during reception operation (TRCn=0), the ACKEn bit must be cleared to 0 to notify the slave sender of the end of data (stop sending) in order not to generate an ACK.

Figure 14-17 ACK



When the local address is received, ACK is automatically generated, regardless of the value of the ACKEn bit. When an address other than that of the local address is received, an ACK is not generated (NACK).

When an extension code is received, an ACK is generated if the ACKEn bit is set to 1 in advance. How ACK is generated when data is received differs as follows depending on the setting of the clock stretch timing.

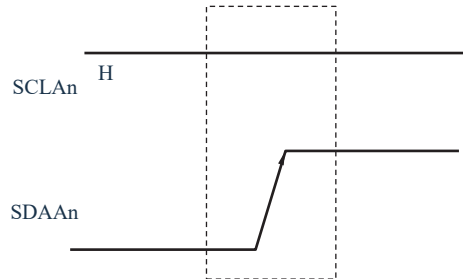
- When an 8-clock wait is selected (bit3(WTIMn)=0 in the IICCTLn0 register): an ACK is generated synchronously with the falling edge of the 8th clock on the SCLAn pin by setting the ACKEn bit to 1 before releasing the wait.
- When a 9-clock wait is selected (bit3(WTIMn)=1 in the IICCTLn0 register): an ACK is generated by setting the ACKEn bit to 1 in advance.

Remark: n=0.

14.5.5 Stop Condition

A stop condition is generated if the SDAAn pin changes from low to high while the SCLAn pin is high. The stop condition is a signal generated when the master device ends serial transfer to the slave device. The stop condition is detected when the device is used as a slave.

Figure 14-18 Stop condition



If the bit0 (SPTn) of the IICA control register n0 (IICCTLn0) is set to 1, a stop condition is generated. If a stop condition is detected, the bit0 (SPDn) of the IICA status register n (IICSn) is set to 1 and the INTIICAn is generated when the bit4 (SPIEn) of the IICCTLn0 register is 1.

Note: n=0.

14.5.6 Wait

Notify the other party that the master or slave device is preparing data for transmitting/receiving (wait state) by waiting.

Notify the other party that it is in a waiting state by setting the SCLAn pin low. If both the master and slave wait states are released, the next transfer can begin.

Figure 14-19 Wait (1/2)

- (1) When master device has a nine-clock wait and slave device has an eight-clock wait
(master transmission, slave reception, and ACKEn = 1)

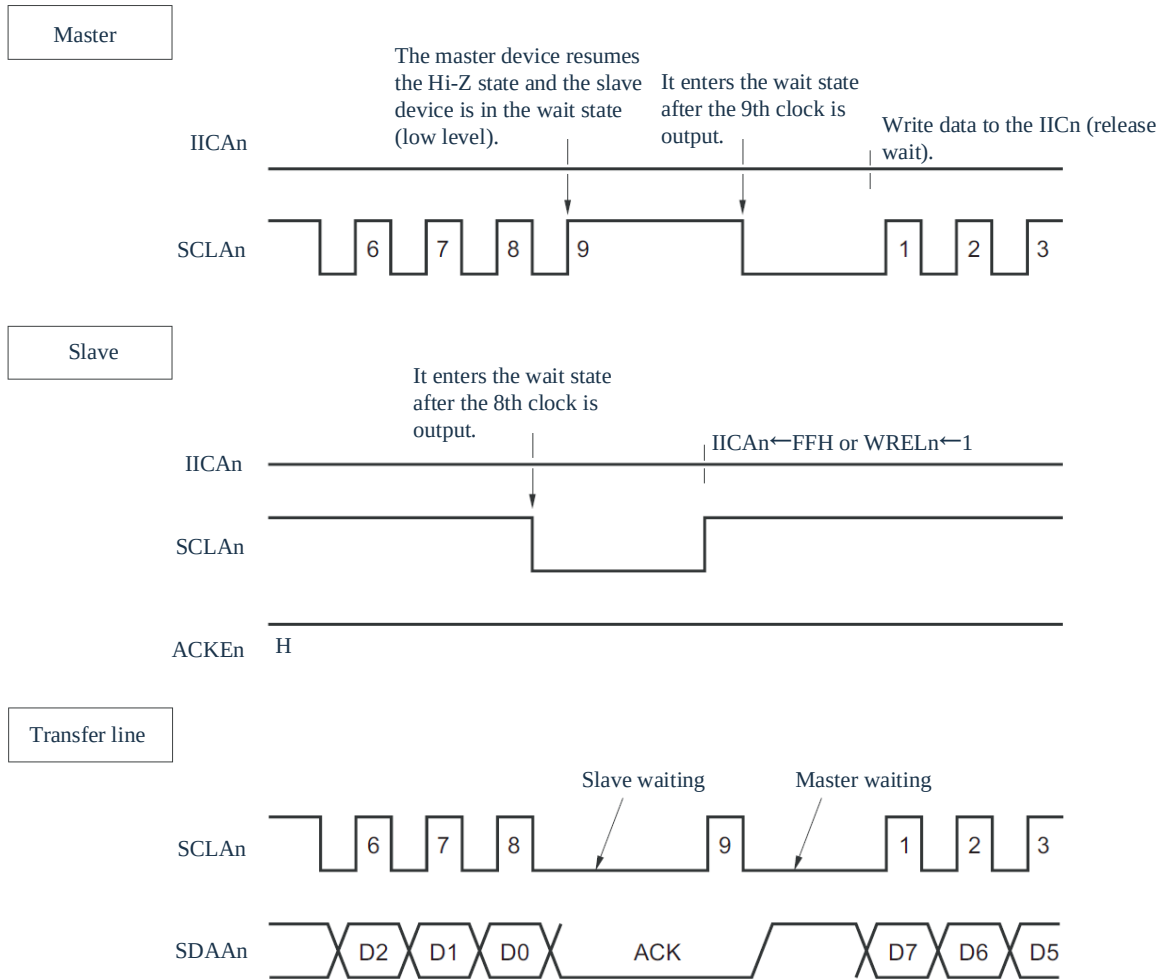
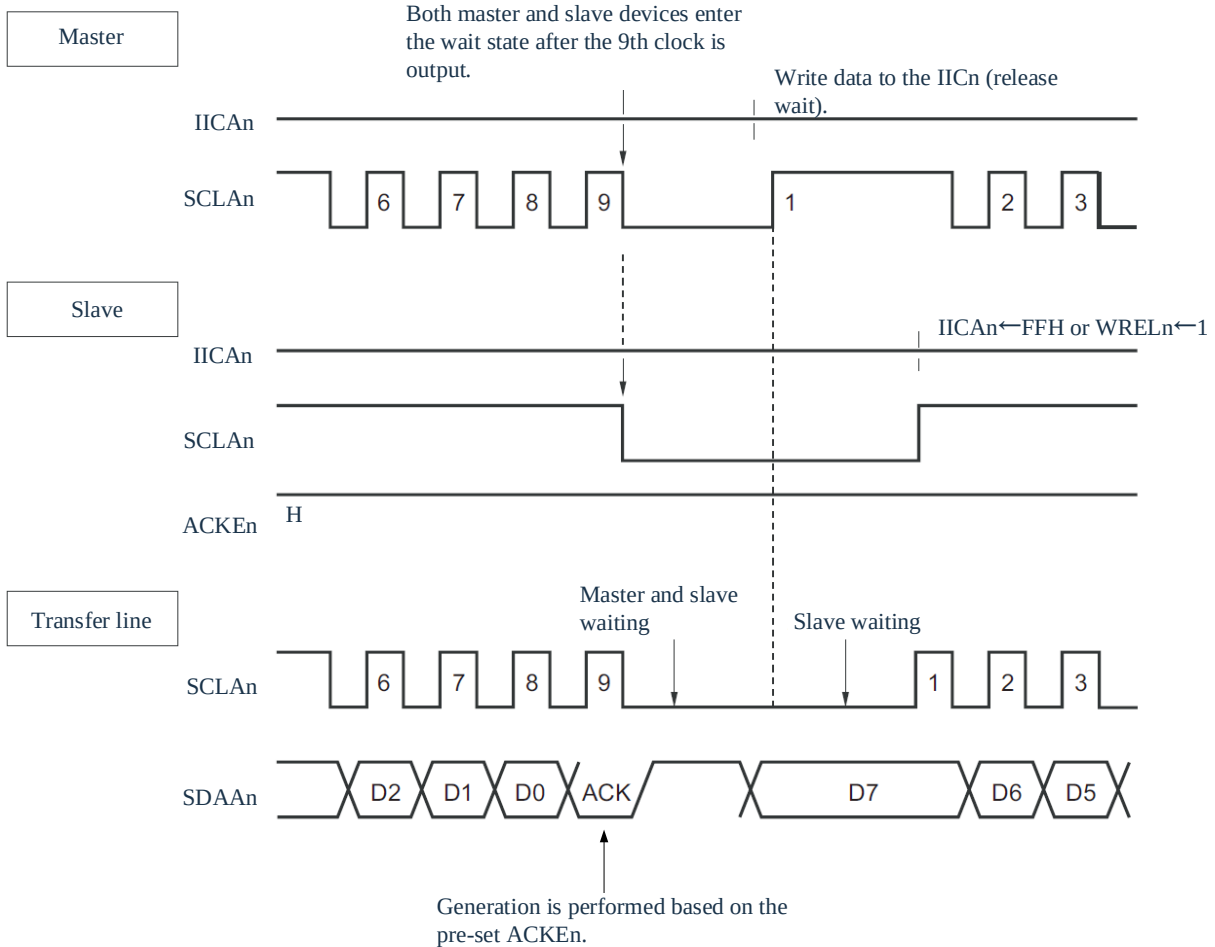


Figure 14-19 Wait (2/2)

(2) When both the master and slave devices are waiting for 9 clocks
(master transmission, slave reception, and ACKEn=1)



Remark 1: ACKEn: bit2 of IICA control register n0 (IICCTLn0)

Remark 2: WRELn: bit5 of IICA control register n0 (IICCTLn0)

A wait state is automatically generated by setting the bit3 (WTIMn) of the IICA control register n0 (IICCTLn0). In general, at the receiver, if the bit5 (WRELn) of the IICCTLn0 register is 1 or writes “FFH” to the IICA shift register n (IICAn), the wait is released. At the transmitter, if data is written to the IICAn register, the wait is released. The master device can release the wait by:

Setting the bit1 (STTn) of the IICCTLn0 register to 1.

Setting the bit0 (SPTn) of the IICCTLn0 register to 1.

Remark: n=0.

14.5.7 Wait Release Methods

In general, the I²C can release the wait by the following methods.

- Write data to the IICA shift register n(IICAn).
- Set the bit5(WRELn) of the IICA control register n0(IICCTLn0) (release wait).
- Set the bit1 (STTn) of the IICCTLn0 register (a start condition is generated) ^{Note}.
- Set the bit0 (SPTn) of the IICCTLn0 register (a stop condition is generated) ^{Note}.

Note: Limited to master devices.

If the release processing of these waits is performed, the I²C releases the waits and restarts communication. To send data (including the address) after releasing the wait, you must write data to the IICn register.

To receive data after release from waiting or to end sending data, the bit 5 (WRELn) of the IICCTLn0 register must be set to 1. To generate a restart condition after releasing the wait, the bit 1 (STTn) of the IICCTLn0 register must be set to 1. To generate a stop condition after releasing a wait, the bit 0 (SPTn) of the IICCTLn0 register must be set to 1. Only one release process can be performed for one wait.

For example, if data is written to the IICAn register after the wait is released by setting the WRELn bit to 1, the change timing of the SDAAn line may conflict with the write timing of the IICAn register, resulting in the wrong value being output to the SDAAn line. In addition to these processes, if the IICEn bit is cleared to 0 when stopping communications in the middle of the communication, communication is stopped, so that waiting can be released. If the I²C bus state is deadlocked due to noise, if the bit 6 (LRELn) of the IICCTLn0 register is set to 1, communication is exited, and thus waiting is released.

Caution: If the wait release process is performed when the WUPn bit is 1, the wait will not be released.

Remark: n=0.

14.5.8 Generation Timing and Waiting Control of Interrupt Requests (INTIICAn)

By setting the bit3 (WTIMn) of the IICA control register n0 (IICCTLn0), the INTIICAn is generated at the timing shown in Table 14-2 and wait control is performed.

Table 14-2 Generation timing and waiting control of INTIICAn

WTIMn	Slave operation			Master operation		
	Address	Data reception	Data transmission	Address	Data reception	Data transmission
0	9 ^{Note1, 2}	8 ^{Note2}	8 ^{Note2}	9	8	8
1	9 ^{Note1, 2}	9 ^{Note2}	9 ^{Note2}	9	9	9

Note 1: Only when the received address and the set address of the slave address register n(SVAn) are the same, the slave generates an INTIICAn signal on the falling edge of the 9th clock and enters a waiting state. At this point, regardless of the bit2 (ACKEn) setting of the IICCTLn0 register, an ACK is generated. The slave that receives the extension code generates INTIICAn on the falling edge of the 8th clock. If the addresses are different after restarting, the INTIICAn is generated on the falling edge of the 9th clock, but does not enter the waiting state.

Note 2: If the contents of the received address and the slave address register n(SVAn) are different and the extension code is not received, the INTIICAn is not generated and does not enter the waiting state.

Remark: The numbers in the table represent the number of clocks for a serial clock. Both interrupt request and wait control are synchronized with the falling edge of the serial clock.

(1) Address transmission and reception

- Slave operation: Regardless of the WTIMn bit, the timing of interrupts and waits is determined according to the conditions in Note 1 and Note 2 above.
- Master operation: Regardless of the WTIMn bit, the timing of interrupts and waits is generated on the falling edge of the 9th clock.

(2) Data reception

- Master/slave operation: Determines the timing of interrupts and waits via the WTIMn bit.

(3) Data transmission

- Master/slave operation: Determines the timing of interrupts and waits via the WTIMn bit.

Remark: n=0.

(4) Release waiting

There are 4 ways to release from waiting:

- Writes data to IICA shift register n (IICAn).
- Sets the bit5 (WRELn) of the IICA control register n0 (IICCTLn0) (wait released).
- Sets the bit1 (STTn) of the IICCTLn0 register (a start condition is generated) ^{Note}.
- Sets the bit0 (SPTn) of the IICCTLn0 register (a stop condition is generated) ^{Note}.

Note 1: Limited to master devices.

When you select a wait for 8 clocks (WTIMn=0), you need to decide whether to generate an ACK before you release the wait.

(5) Detection of stop condition

If a stop condition is detected, the INTIICAn is generated (only when SPIEn=1).

14.5.9 Detection for Address Matching

In I²C-bus mode, the master device can select a specific slave by sending a slave address. Address matching can be automatically detected by hardware. When the slave address sent by the master device and the set address of the slave address register n(SVAn) are the same or only the extension code is received, an INTIICAn interrupt request is generated.

14.5.10 Error Detection

In I²C bus mode, because the state of the serial data bus (SDAAn) is captured into the transmitter's IICA shift register n (IICAn) during the transmission process, it is possible to detect transmission errors by comparing the data in the IICA before the transmission starts and after it ends. If the two sets of data are different, it is considered that a transmission error has occurred.

Remark: n=0.

14.5.11 Extension Code

- (1) When the high 4 bits of the receiving address are “0000” or “1111”, as the received extension code, the extended code receive flag (EXCn) is set to 1, and in the 8th The falling edge of the clock generates an interrupt request (INTIICAn).

Does not affect local station addresses stored in slave address register n (SVAn).

- (2) When the SVAn register is set to “11110xx0”, if “11110xx0” is sent from the master device via a 10-bit address, the following assertion occurs. However, an interrupt request (INTIICAn) is generated on the falling edge of the 8th clock.
 - High 4 bits data are the same: EXCn=1
 - 7 bits of data are the same: COIn=1

Remarks: EXCn: Bit5 of the IICA status register n

COIn: Bit4 of IICA status register n (IICSn)

- (3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software. If the extension code is received while a slave device is operating, then the slave device is participating in communication even if its address does not match. For example, after the extension code is received, if you do not wish to operate the target device as a slave device, set bit 6 (LRELn) of IICA control register n0 (IICCTLn0) to 1 to set the standby mode for the next communication operation.

Table 14-3 Bit definitions of major extension codes

Slave address	R/W bit	Definition
0000000	0	Full call address
11110xx	0	10-bit slave address specification (during address authentication)
11110xx	1	10-bit slave address specification (after address match, when read command is issued)

Remark 1: See the I²C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

Remark 2: n=0.

14.5.12 Arbitration

When multiple master devices generate start conditions at the same time (Set STTn bit to 1 before the STDn bit becomes 1), the communication of the master device is carried out while adjusting the clock until the data is different. This operation is called arbitration.

When arbitration fails, the master device that has failed arbitration sets the arbitration failure flag (ALDn) in the IICA status register n (IICSn) to 1 and sets both the SCLAn line and the SDAAn line to a high impedance state, releasing the bus.

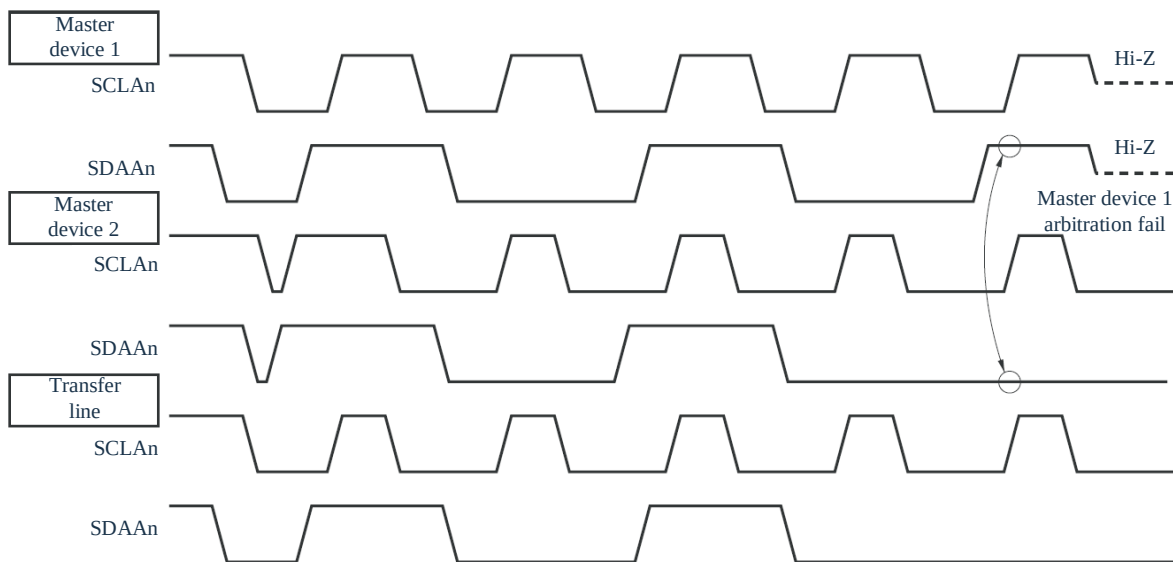
When the next interrupt request occurs (e.g., a stop condition is detected on the 8th or 9th clock), the failure of arbitration is detected by setting the ALDn bit to 1 in software.

For the timing of interrupt requests, please refer to “14.5.8 Generation timing and waiting control of interrupt requests (INTIICAn)”.

Remark 1: STDn: bit1 of IICA status register n(IICSn)

Remark 2: STTn: bit1 of IICA control register n0 (IICCTLn0)

Figure 14-20 Timing example of arbitration



Remark: n=0.

Table 14-4 Status at the time of arbitration and timing of generation of interrupt requests

The state in which the arbitration occurred	Timing of the generation of interrupt requests
During address transmission	On the falling edge of the 8th or 9th clock after the byte is transmitted ^{Note1}
Reads and writes information after address transmission	
During address extension codes transmission	
Reads and writes information after extension codes transmission	
During data transmission	
During ACK transmission after transmitted data	
Restart condition detected during data transmission	
Stop condition detected during data transmission	When generating a stop condition (SPIEn=1) ^{Note2}
Trying to generate a restart condition, but the data is low.	On the falling edge of the 8th or 9th clock after the byte is transmitted ^{Note1}
Trying to generate a restart condition, but a stop condition was detected.	When generating a stop condition (SPIEn=1) ^{Note2}
Trying to generate a stop condition, but the data is low.	On the falling edge of the 8th or 9th clock after the byte is transmitted ^{Note1}
Trying to generate a restart condition, but SCLAn is low.	

Note 1: The interrupt request is generated on the falling edge of the 9th clock when the WTIMn bit (bit3 of the IICA control register n0 (IICCTLn0)) is 1, and on the falling edge of the 8th clock when the WTIMn bit is 0 and the slave address of the extended code is received.

Note 2: When there is a possibility of arbitration, the SPIEn bit must be 1 when the master is operating.

Remark 1: SPIEn: bit4 of IICA control register n0 (IICCTLn0)

Remark 2: n=0.

14.5.13 Wake-Up Function

This is a slave function of I²C, which is the function of generating an interrupt request signal (INTIICAn) when the local station address and extension code are received. The processing efficiency is improved by not generating unwanted INTIICAn signals under different addresses. If a start condition is detected, it enters wake-up standby. Because the master device (where a start condition has already been generated) may also become a slave due to an arbitration failure, it enters wake-up standby at the same time as the address is sent.

To use the wake function in deep sleep mode, you must place the WUPn at 1. The address can be received independent of the operating clock. Even in this case, an interrupt request signal (INTIICAn) is generated when the local station address and extension code are received. After this interrupt is generated, the WUPn bit is cleared to 0 by the instruction and returned to the normal operation.

The flow when the WUPn bit is set to 1 is shown in Figure 14-21, and the flow when the WUPn bit is set to 0 by address matching is shown in Figure 14-22.

Figure 14-21 Setting the WUPn bit to 1

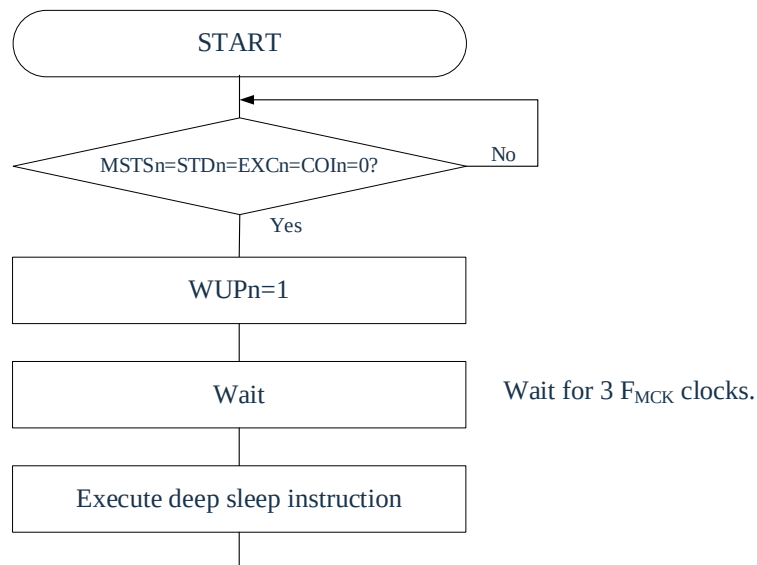
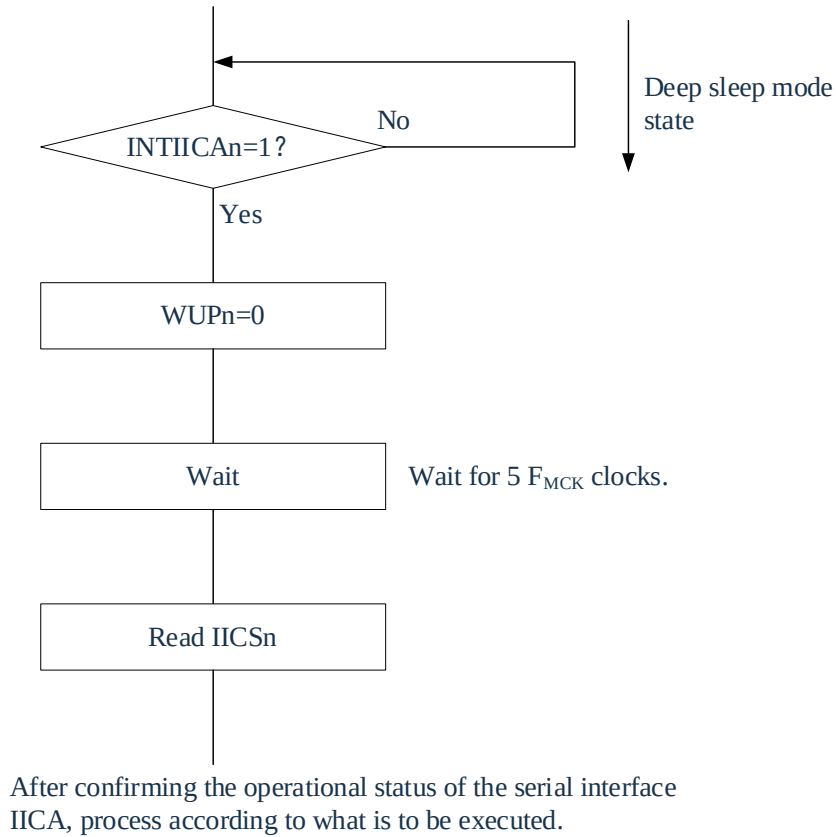


Figure 14-22 Flow when the WUPn bit is set to 0 by address matching (including receiving extension codes)



In addition to the interrupt request (INTIICAn) generated by the serial interface IICA, the deep sleep mode must be removed through the following procedure.

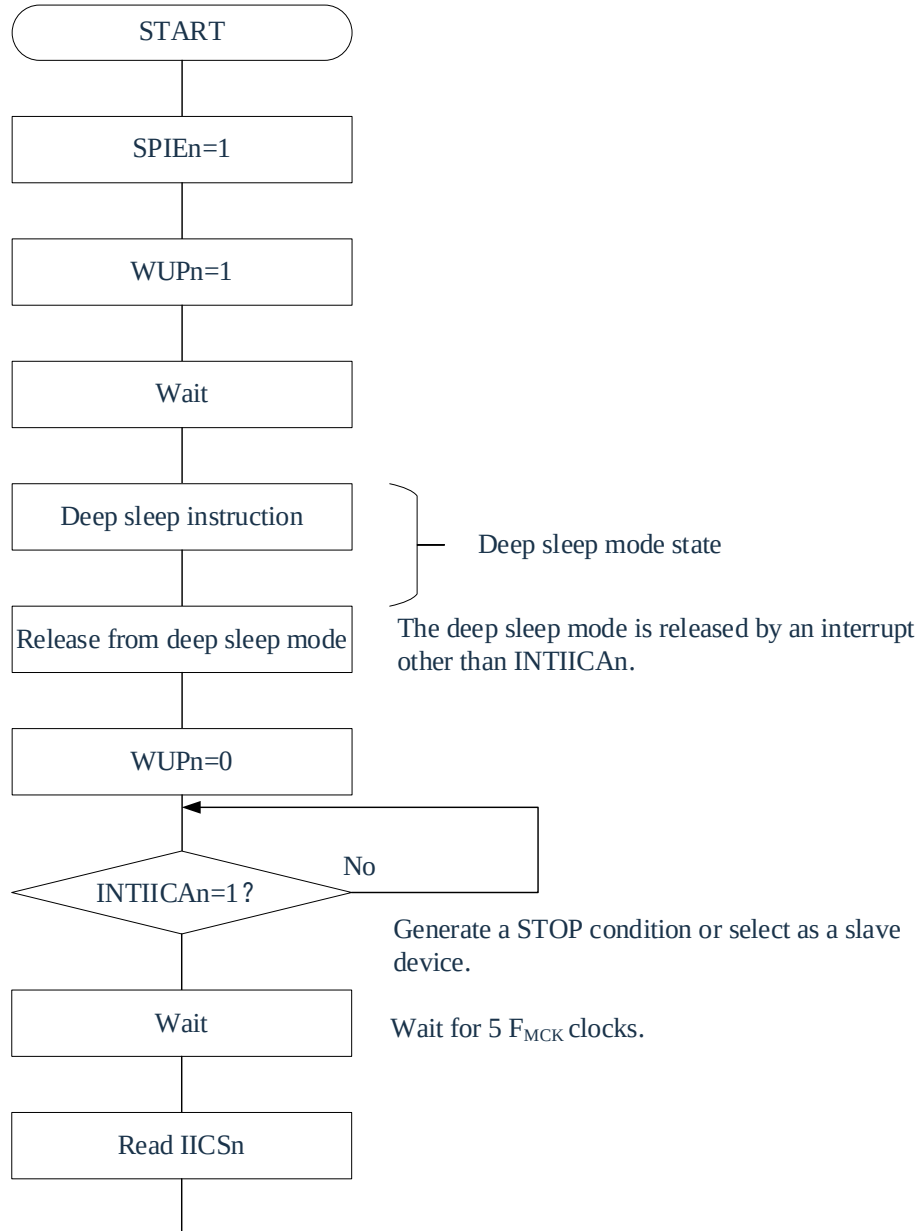
- Next IIC communication for the operation of the master device: the flow of Figure 14-23
- Next IIC communication for slave device operation:

Return via INTIICAn interrupt: same process as Figure 14-22.

Return from an interrupt other than the INTIICAn interrupt: operation must be continued with the WUPn bit set to 1 before the INTIICAn interrupt is generated.

Remark: n=0.

Figure 14-23 Operation as master device after being released from deep sleep mode by an interrupt other than INTIICAn



After confirming the operational status of the serial interface IICA, process according to what is to be executed.

Remark: n=0.

14.5.14 Communication Reservation

- (1) When communication reservation function is enabled (bit0 (IICRSVn)=0 of the IIC flag register n (IICFn))

To perform the next master communication without using the bus, you can send a start condition when the bus is released through a communication reservation. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When the extended code is received but the device does not operate as a slave (by not returning an ack and setting bit 6 (LRELn) of the IICA control register n0 (IICCTLn0) to 1), the bus is released after exiting communication.

If bit 1 (STTn) of the IICCTLn0 register is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.

If an address is written to the IICA shift register n (IICAn) after bit 4 (SPIEn) of the IICCTLn0 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICAn) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICAn register before the stop condition is detected is invalid.

When the STTn bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- Generate a start condition when the bus is released.
- Communication reservation when the bus is not released (standby state)

After the STTn bit is set to “1” and a wait time has elapsed, operation as a communication reservation is confirmed by the MSTSn bit (bit 7 of the IICA status register n (IICSn)).

Waiting times must be ensured by software for the following formula calculations.

Wait time from setting STTn bit to 1 until the MSTSn flag is confirmed:

$$(IICWL_n \text{ set value} + IICWH_n \text{ set value} + 4) / F_{MCK} + T_F \times 2$$

Remarks 1: IICWLn: IICA low-level width setting register n

IICWHn: IICA high-level width setting register n

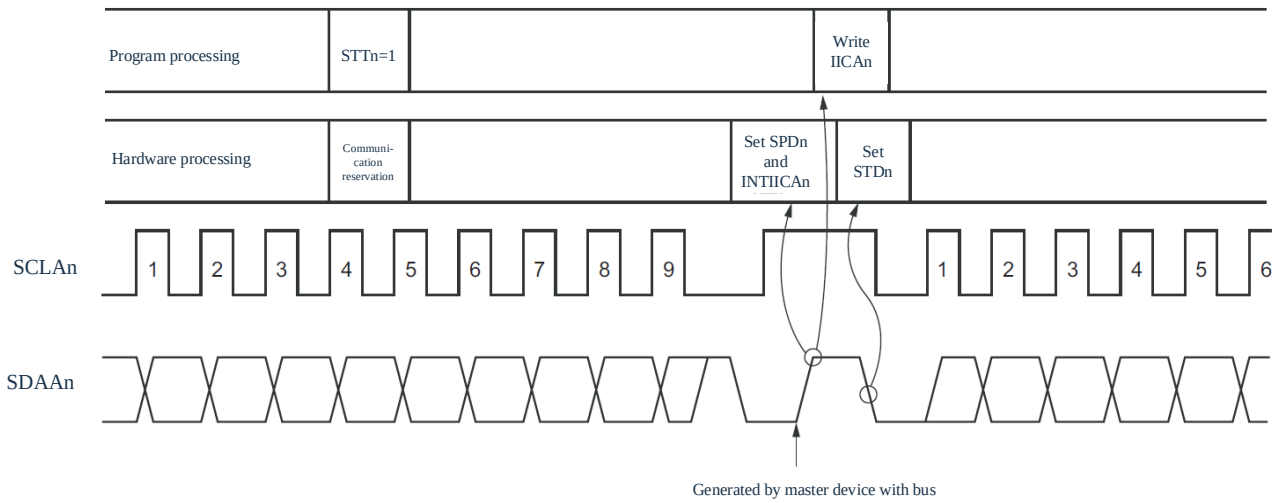
T_F: SDAAn and SCLAn signal falling times

F_{MCK}: IICA operation clock frequency

2: n=0

The timing of the communication reservation is shown in the figure below.

Figure 14-24 Timing of communication reservation



Remark: IICAn: IICA shift register n.

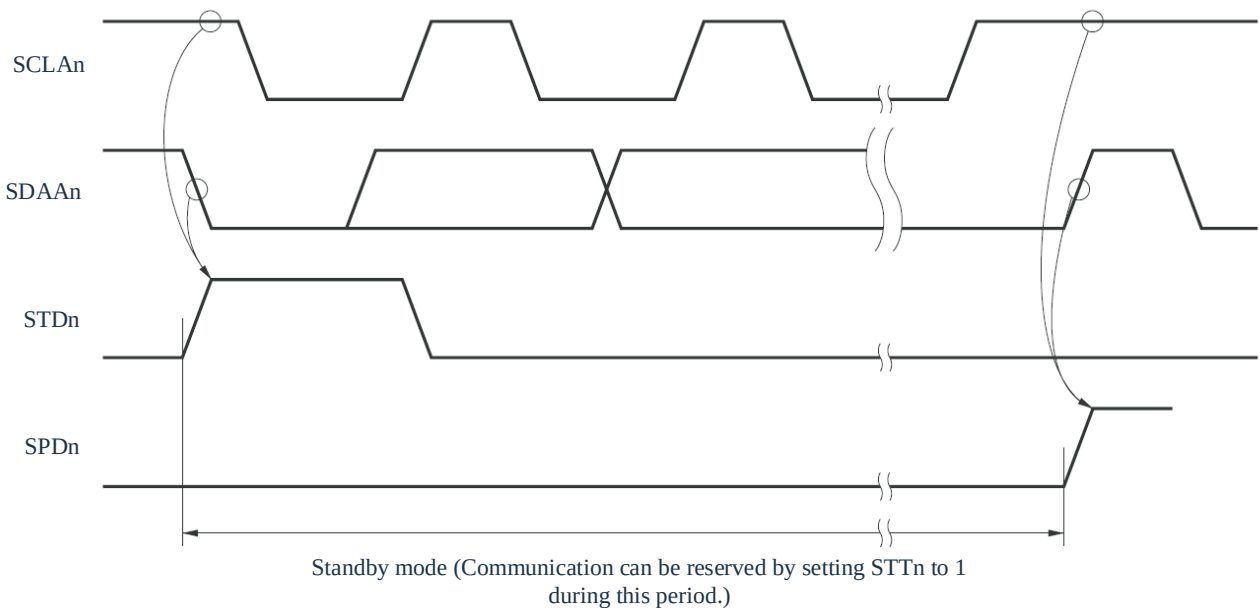
STTn: bit1 of IICA control register n0 (IICCTLn0)

STDn: bit1 of IICA status register n (IICSn)

SPDn: bit0 of IICA status register n (IICSn)

The communication reservation is accepted by the timing sequence shown in Figure 14-25. After the bit1 (STDn) of the IICA status register n (IICSn) becomes 1 and before a stop condition is detected, the bit1 (STTn) of the IICA control register n0 (IICCTLn0) is set to 1 to make a communication reservation.

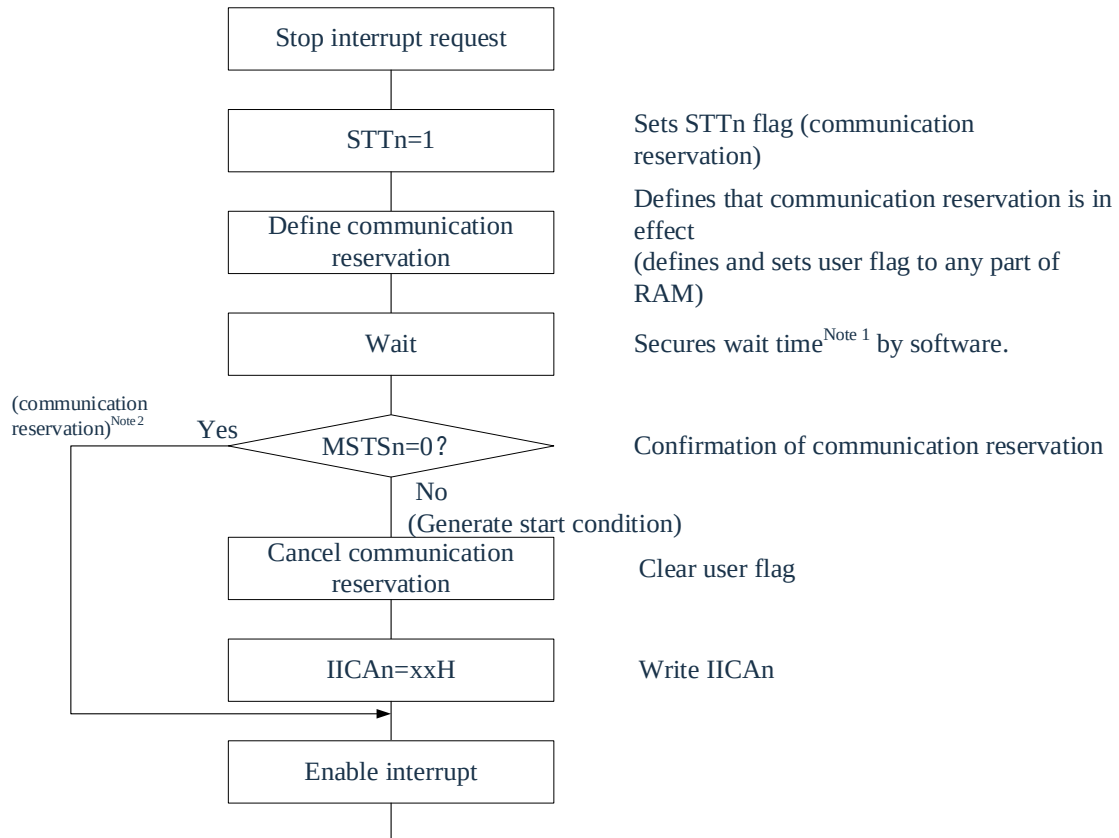
Figure 14-25 Timing of accepting communication reservations



Note: n=0

The steps for communication reservations are shown in Figure 14-26.

Figure 14-26 Steps for communication reservation



Note 1: The wait time is calculated as follows. $(IICWLn \text{ set value} + IICWHn \text{ set value} + 4) / F_{MCK} + T_F \times 2$.

Note 2: The communication reservation operation executes a write to the IICA shift register n (IICAn) when a stop condition interrupt request occurs.

Remark 1. STT_n: Bit1 of IICA control register n0 (IICCTLn0).

MSTS_n: Bit 7 of IICA status register n (IICS_n).

IICAn: IICA shift register n.

IICWL_n: IICA low-level width setting register n.

IICWH_n: IICA high-level width setting register n.

T_F: SDAAn and SCLAn signal falling times.

F_{MCK}: IICA operation clock frequency.

2. n=0.

- (2) When communication reservation function is disabled (bit 0 (IICRSVn) of IICA flag register n (IICFn) = 1)

When bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LRELn) of the IICCTLn0 register to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCFn (bit 7 of the IICFn register).

It takes up to 5 cycles of F_{MCK} until the STCFn bit is set to 1 after setting $STTn = 1$. Therefore, secure the time by software.

Remark: n=0.

14.5.15 Cautions

(1) When $STCENn = 0$

Immediately after I²C operation is enabled ($IICEn = 1$), the bus communication status ($IICBSYn = 1$) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication. When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected). Use the following sequence for generating a stop condition.

- ① Set IICA control register n1 ($IICCTLn1$).
- ② Set bit 7 ($IICEn$) of IICA control register n0 ($IICCTLn0$) to 1.
- ③ Set bit 0 ($SPTn$) of the $IICCTLn0$ register to 1.

(2) When $STCENn = 1$

Immediately after I²C operation is enabled ($IICEn = 1$), the bus released status ($IICBSYn = 0$) is recognized regardless of the actual bus status. To generate the first start condition ($STTn = 1$), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I²C communications are already in progress

If I²C operation is enabled and the device participates in communication already in progress when the $SDAAn$ pin is low and the $SCLAn$ pin is high, the macro of I²C recognizes that the $SDAAn$ pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, ACK is returned, but this interferes with other I²C communications. To avoid this, start I²C in the following sequence.

- ① Clear bit 4 ($SPIEn$) of the $IICCTLn0$ register to 0 to disable generation of an interrupt request signal ($INTIICAn$) when the stop condition is detected.
 - ② Set bit 7 ($IICEn$) of the $IICCTLn0$ register to 1 to enable the operation of I²C.
 - ③ Wait for detection of the start condition.
 - ④ Set bit 6 ($LRELn$) of the $IICCTLn0$ register to 1 before ACK is returned (4 to 72 cycles of F_{MCK} after setting the $IICEn$ bit to 1), to forcibly disable detection.
- (4) Setting the $STTn$ and $SPTn$ bits (bits 1 and 0 of the $IICCTLn0$ register) again after they are set and before they are cleared to 0 is prohibited.
- (5) When transmission is reserved, set the $SPIEn$ bit (bit 4 of the $IICCTLn0$ register) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to the IICA shift register n ($IICAn$) after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set the $SPIEn$ bit to 1 when the $MSTSn$ bit (bit 7 of the IICA status register n ($IICSn$)) is detected by software.

Remark: $n=0$.

14.5.16 Communication Operation

The following shows three operation procedures with flowcharts.

(1) Master operation in single master system

The flowchart when using as the master in a single master system is shown below.

This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

(2) Master operation in multimaster system

In a multi-master system of the I²C bus, it is not possible to judge whether the bus is in the release state or in use during the stage of participating in the communication based on the specifications of the I²C bus. Here, if the data and clock are high for a certain amount of time (1 frame), the bus participates in communication as a release state. This process is roughly divided into “initial settings”, “communication waiting” and “communication processing”. The processing designated as a slave due to the failure of the arbitration is omitted here, and only the processing used as the master device is omitted. Join the bus after performing the Initial Setup section at startup, and then wait for a communication request from the master device or a designation of the slave device via Communication Wait. The actual communication is the “Communication Processing” section, which supports arbitration with other master devices in addition to data sending and receiving with the slave.

(3) Slave operation

An example of an I²C bus slave is shown below.

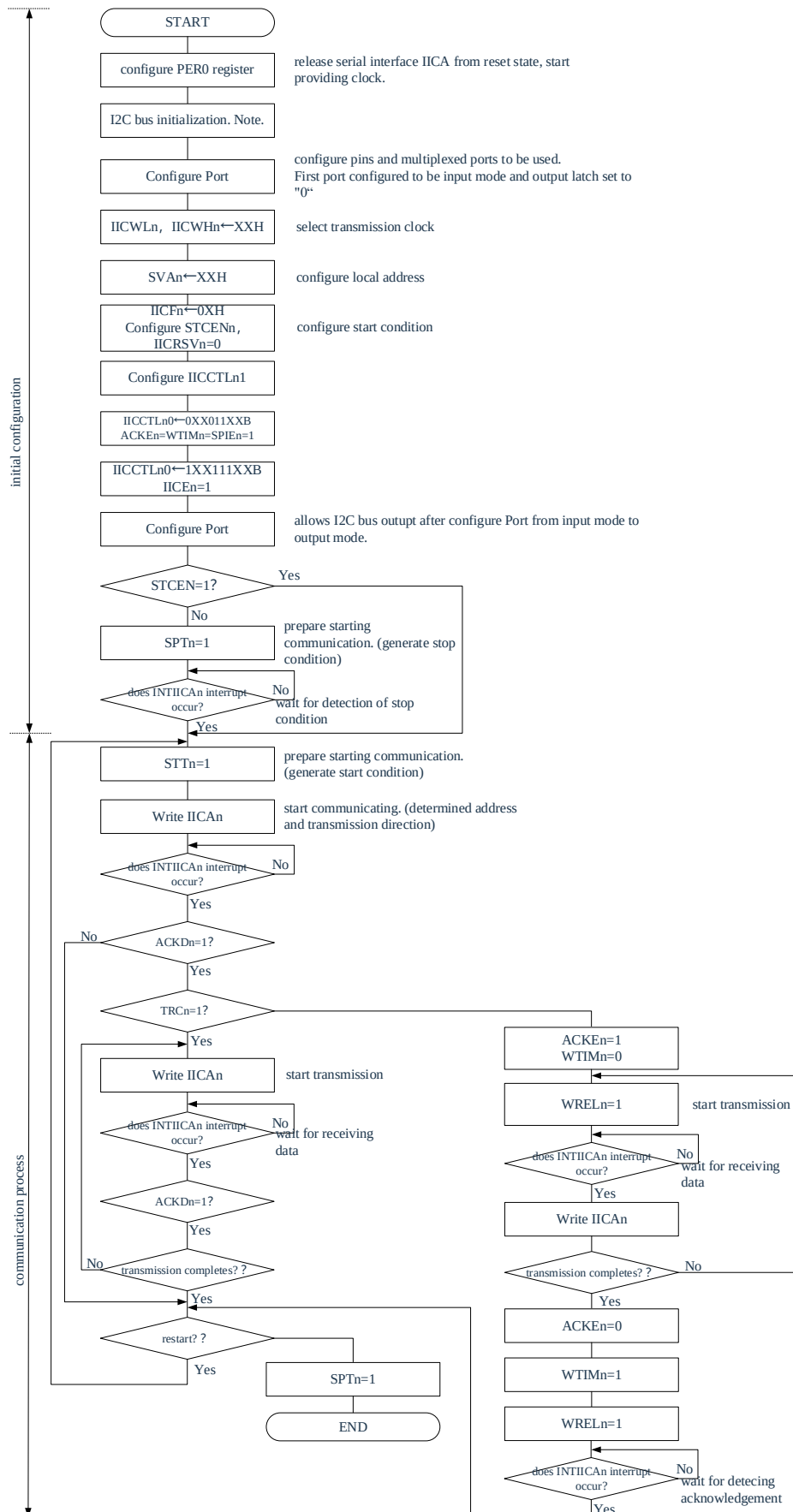
When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICAn interrupt occurrence (communication waiting). When an INTIICAn interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

Remark: n=0.

(1) Master operation in single-master system

Figure 14-27 Master operation of single-master system



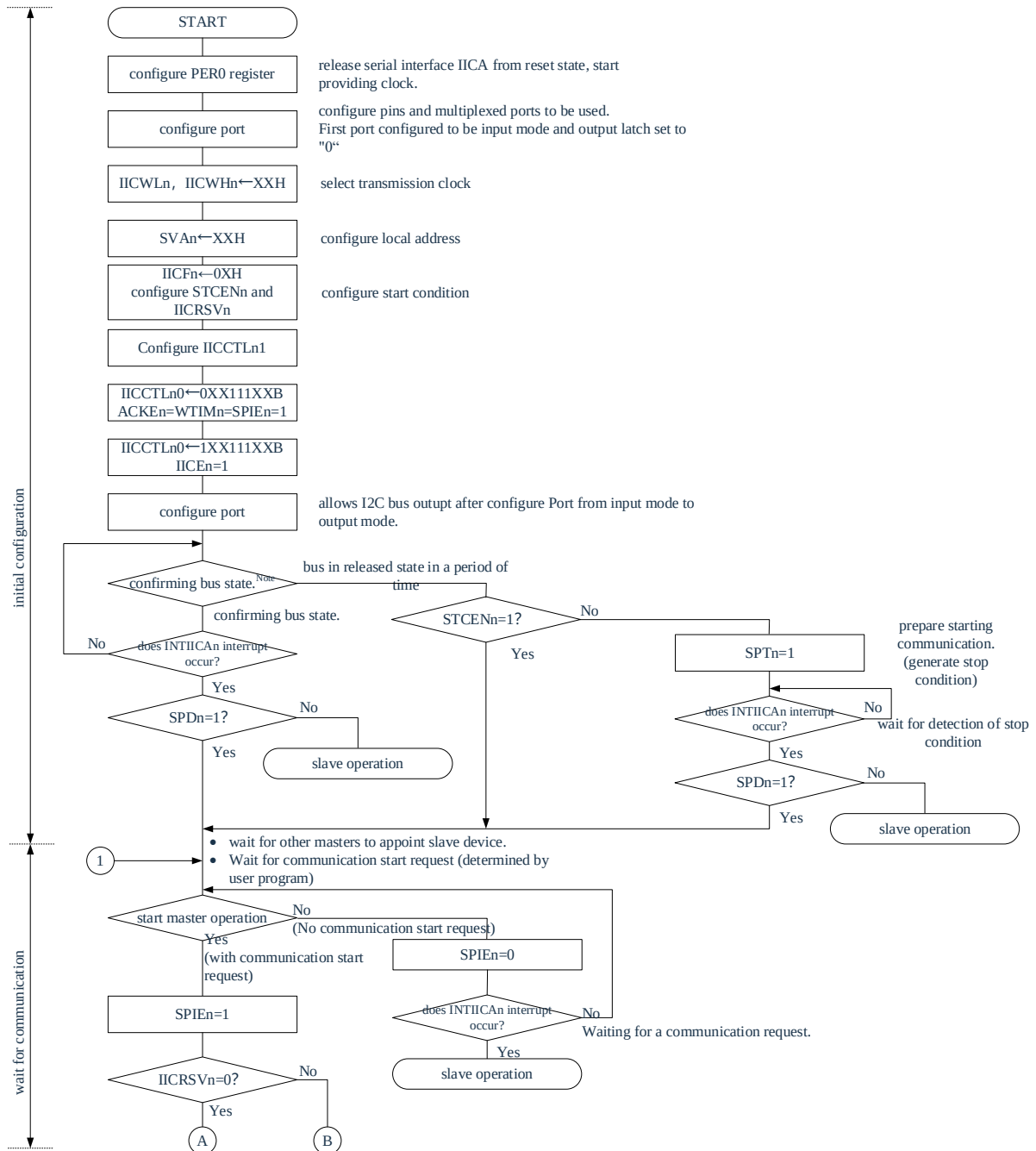
Note 1: I²C-bus must be released (SCLAn pins and SDAAn pins are high) depending on the specifications of the product in communication. For example, if the EEPROM is in a state that outputs a low level to the SDAAn pin, the SCLAn pin must be set to the output port and a clock pulse must be output from the output port before the SDAAn pin is fixed high.

Note 2: The format of transmission and reception must conform to the specifications of the product in communication.

Note 3: n=0.

(2) Master operation in multi-master system

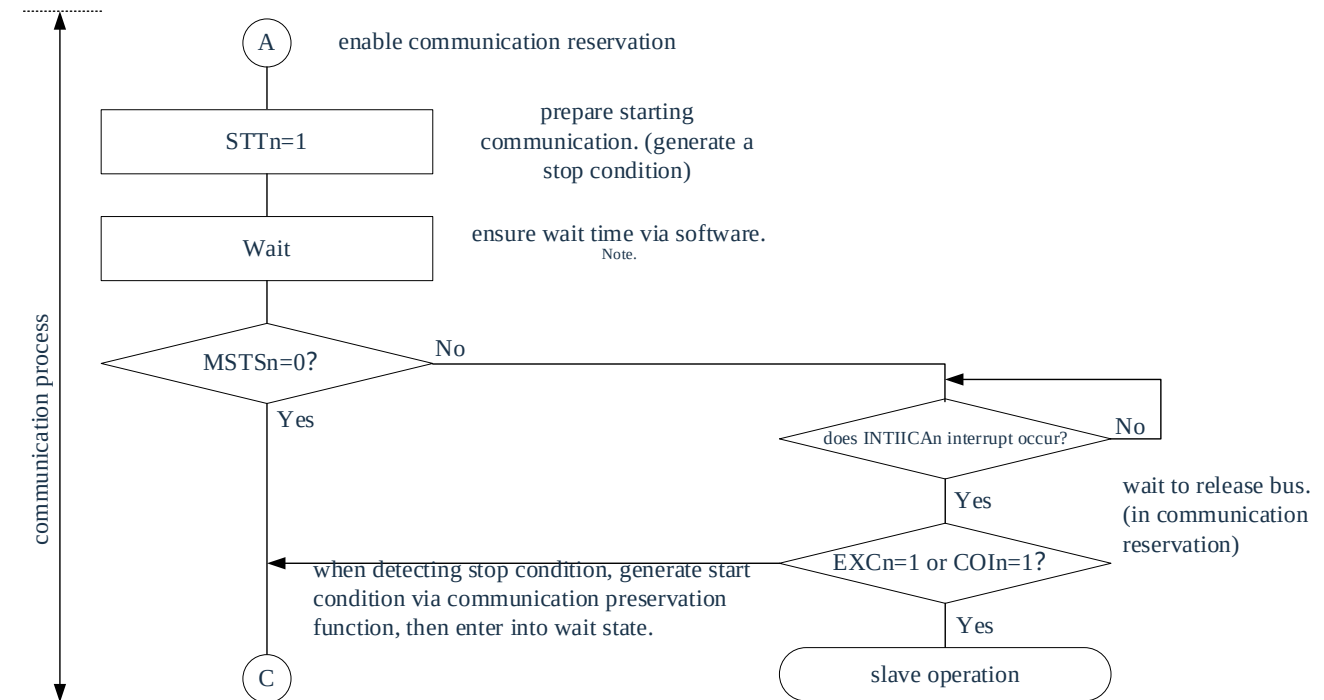
Figure 14-28 Master operation in multi-master system (1/3)



Note: Confirm that the bus is released (CLDn bit = 1, DADn bit = 1) for a specific period (for example, for a period of one frame). If the SDAAn pin is constantly at low level, decide whether to release the I²C bus (SCLAn and SDAAn pins = high level) in conformance with the specifications of the product that is communicating.

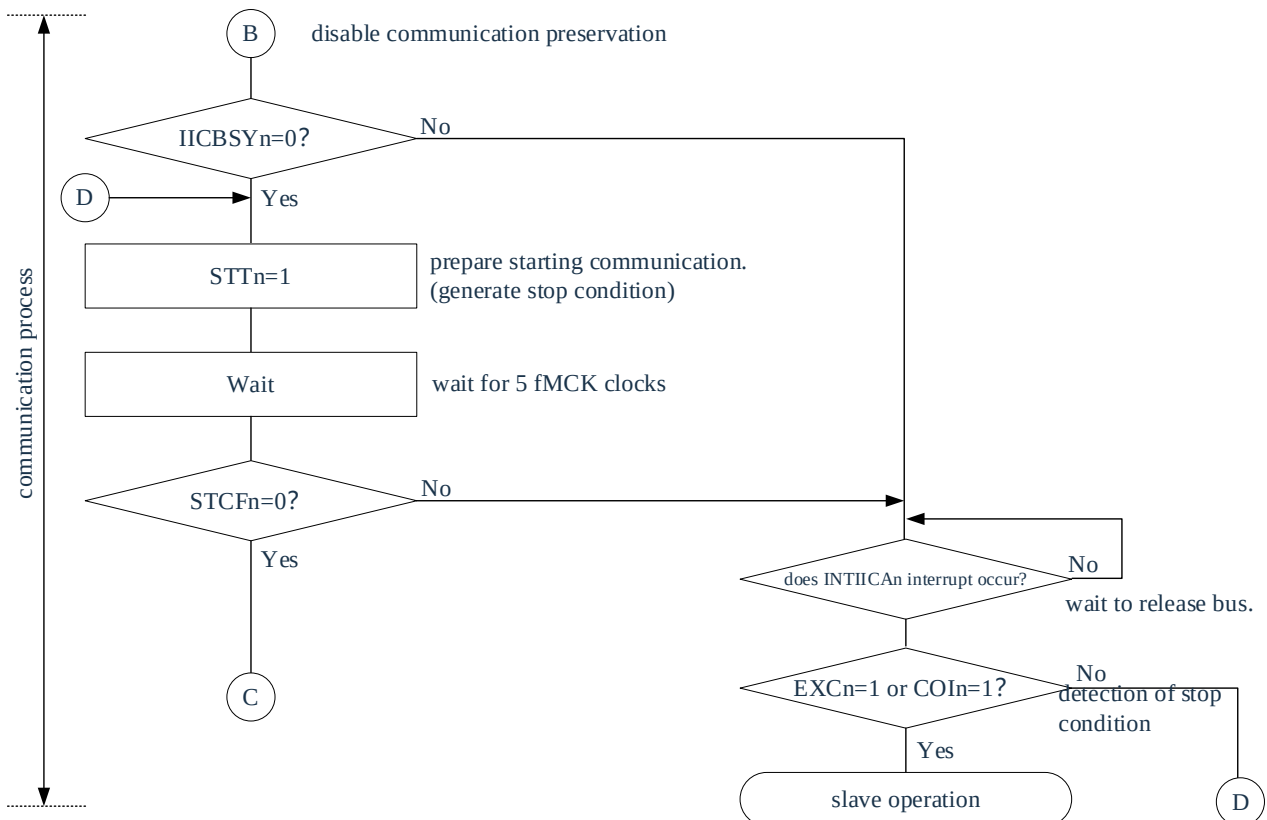
Remark: n=0.

Figure 14-28 Master operation in multi-master system (2/3)



Note wait time as following:

$$(IICWLn \text{ configured value} + IICWHn \text{ configured value} + 4) / f_{MCK} + t_F \times 2$$



Remarks 1. IICWLn: IICA low-level width setting register n

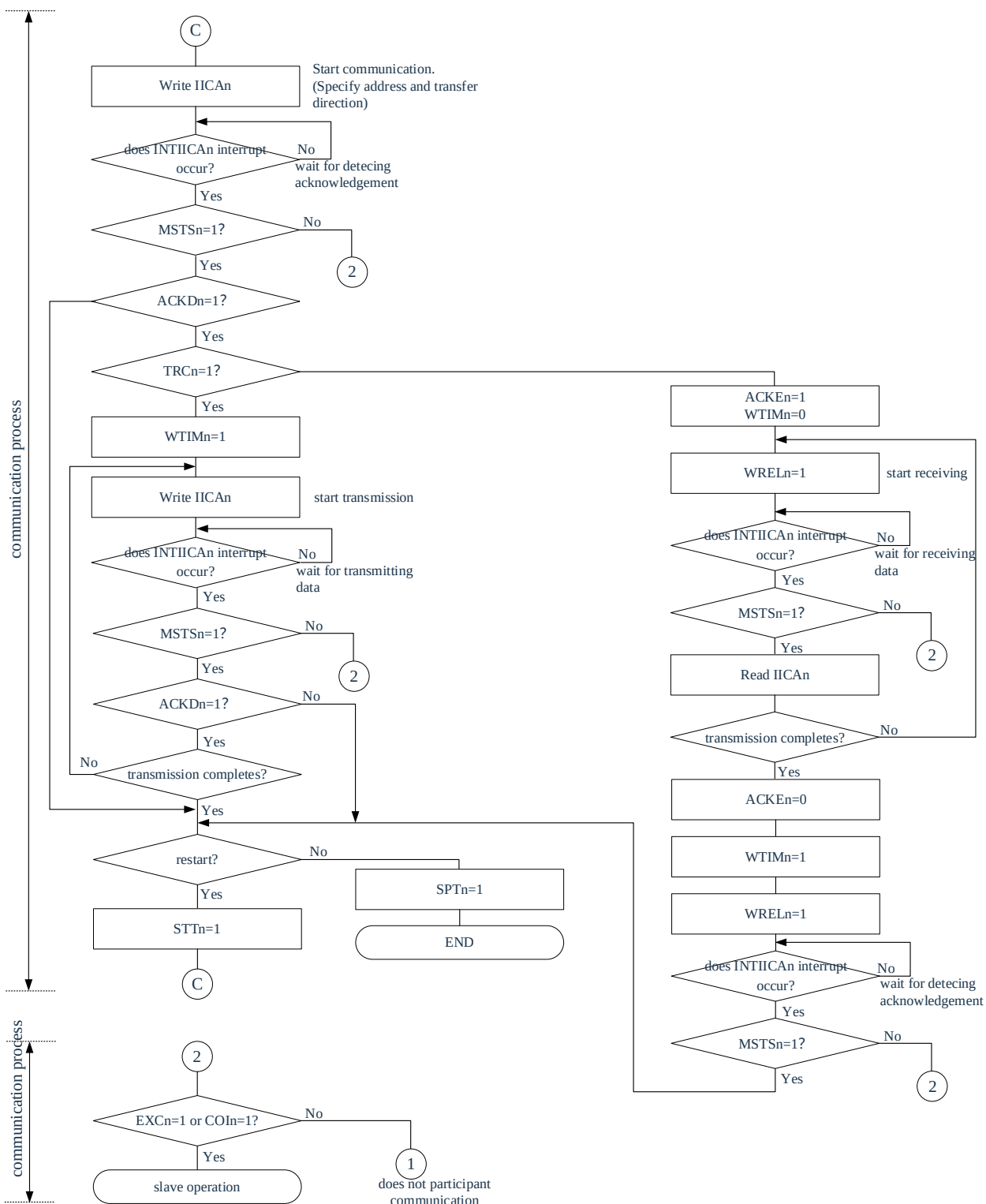
IICWHn: IICA high-level width setting register n

T_F: SDAAn and SCLAn signal falling times

f_{MCK}: IICA operation clock frequency

2. n=0.

Figure 14-28 Master operation in multi-master system (3/3)



Remarks 1. The format of transmission and reception must conform to the specifications of the product in communication.

2. When used as a master device in a multi-master system, the MSTSn bit must be read each time an INTIICAn interrupt occurs to acknowledge the arbitration result.

3. When used as a slave device in a multi-master system, the status must be confirmed each time an INTIICAn interrupt occurs by means of the IICA status register n (IICS_n) and the IICA flag register n (IICF_n) to determine subsequent processing.

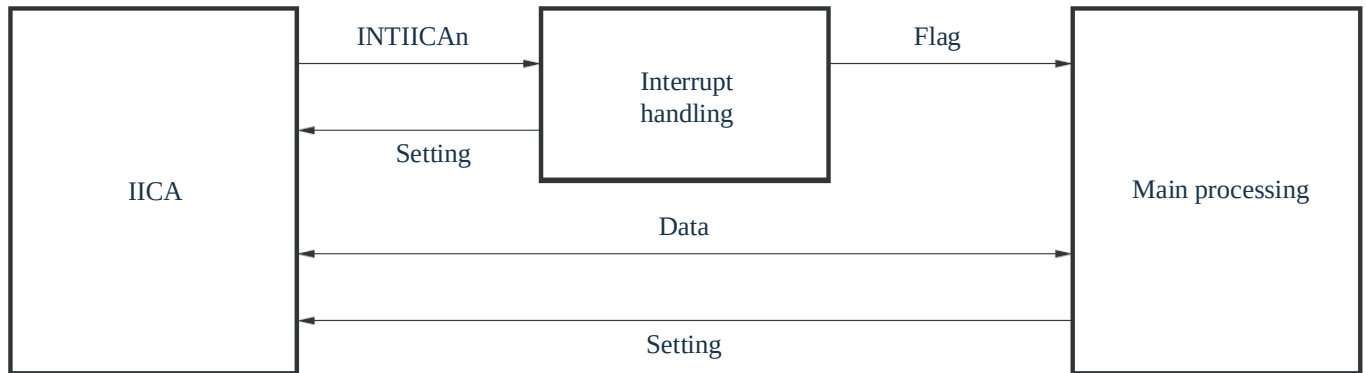
4: n=0.

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICAn interrupt (processing that must substantially change the operation status such as detection of a stop condition during communication) is necessary.

In the following explanation, it is assumed that the extension code is not supported for data communication. It is also assumed that the INTIICAn interrupt servicing only performs status transition processing, and that actual data communication is performed by the main processing.



Therefore, the following three flags are prepared and pass the flags to the main processing department instead of INTRAICAn for data communication processing.

① Communication mode flag

This flag indicates the following 2 communication states:

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of ACK from master, address mismatch)

② Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIICAn interrupt for ordinary data communication. This flag is set by interrupt handling and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

③ Communication direction flag

This flag indicates the direction of communication. Its value is the same as the TRCn bit.

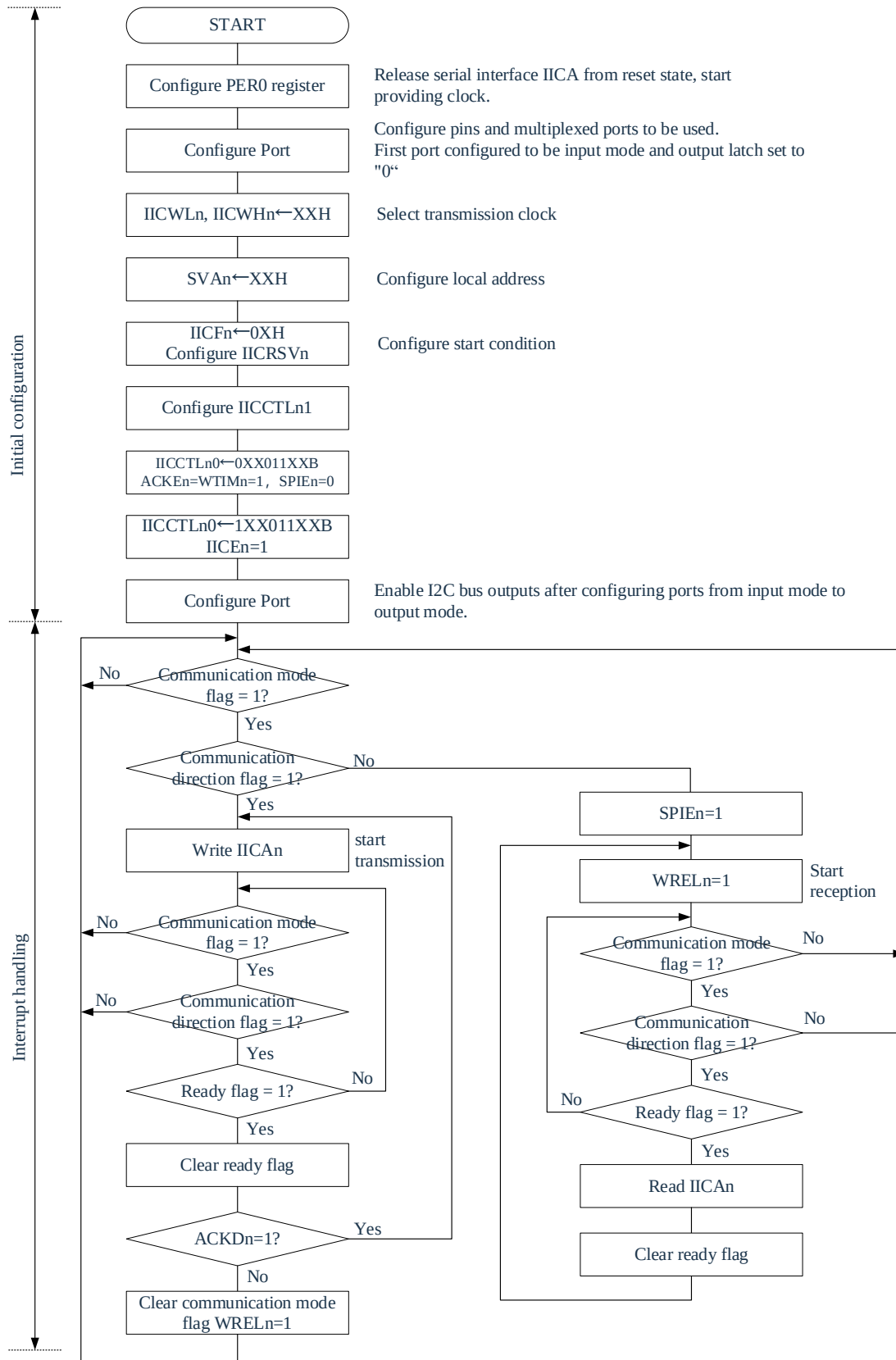
Remark: n=0.

The main processing of the slave operation is explained next.

Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, checks the status by using the flags).

The transmission operation is repeated until the master no longer returns ACK. If ACK is not returned from the master, communication is completed. For reception, the necessary amount of data is received. When communication is completed, ACK is not returned as the next data. After that, the master device generates a stop condition or a restart condition to exit the communication state.

Figure 14-29 Slave operation flowchart (1)



Remarks 1: The formats of transmission and reception must conform to the specifications of the product in communication.

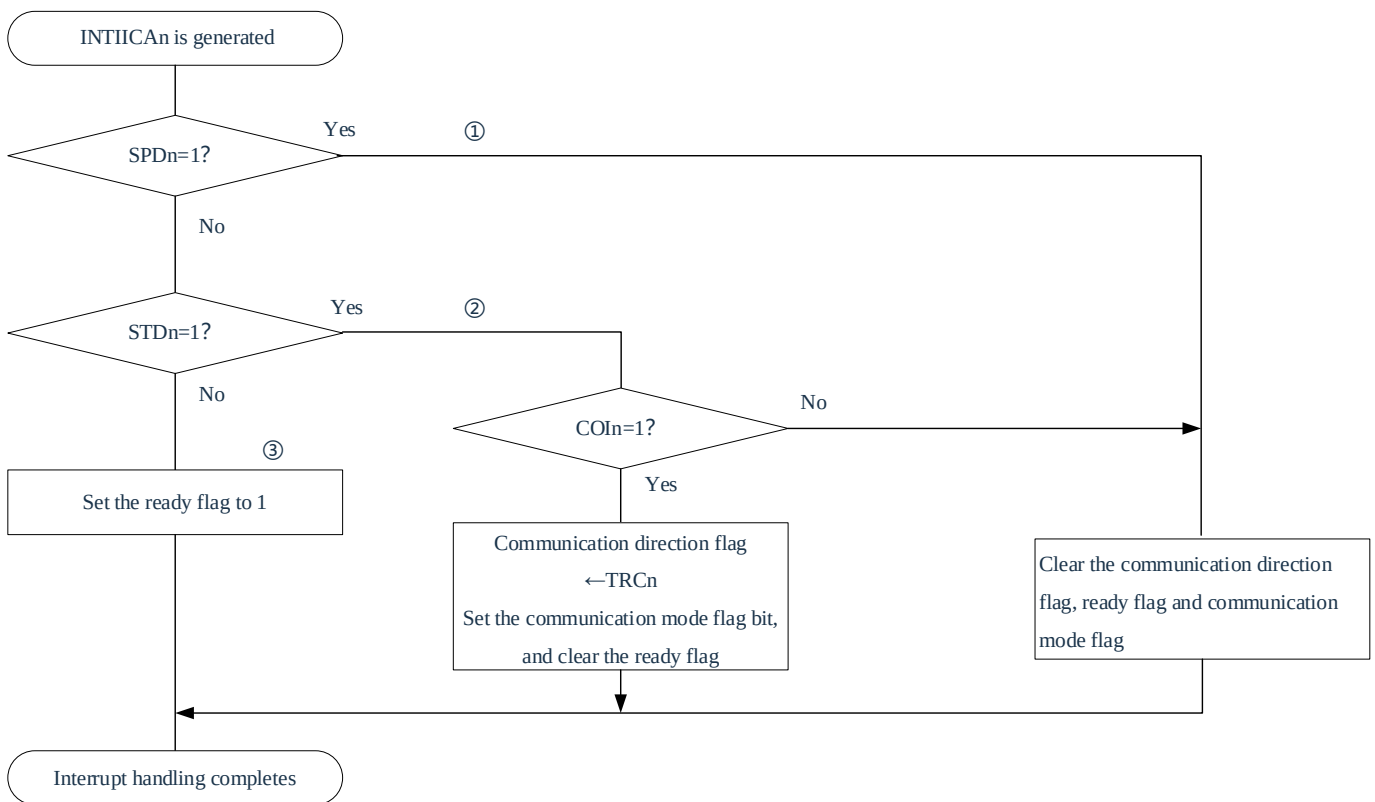
2: n=0.

An example of the steps for slave device processing via the INTIICAn interrupt is shown below (it is assumed that there is no processing with the extension code here). Confirm the status via the INTIICAn interrupt and perform the following processing.

- ① Communication is stopped if the stop condition is issued.
- ② If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is cancelled, and processing returns from the interrupt (the ready flag is cleared).
- ③ For data transmission/reception, only the ready flag is set. Processing returns from the interrupt with the I²C bus remaining in the wait state.

Remark: ① to ③ above correspond to ① to ③ in Figure 14-30.

Figure 14-30 Slave operation flowchart (2)



Remark: n=0.

14.5.17 Timing of I²C Interrupt Request (INTIICAn) Generation

The transmission and reception timing of data, the timing for the generation of the INTIICAn interrupt request signal, and the value of the IICA status register n (IICSn) when the INTIICAn signal is generated are as follows.

Remarks 1: ST: Start condition

AD6~AD0: Address

R/ $\overline{W}$: Transfer direction specification

ACK: Acknowledge

D7~D0: Data

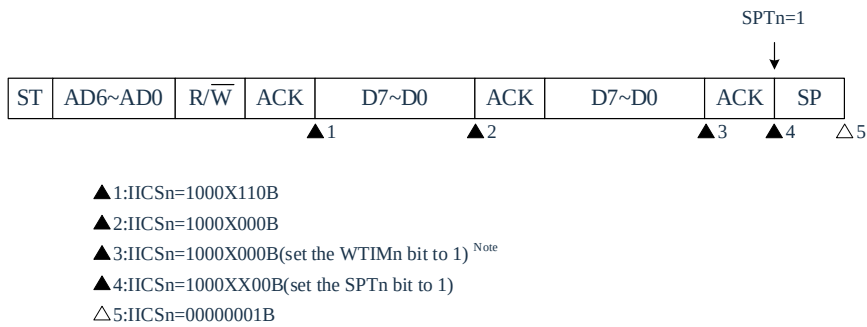
SP: Stop condition

2. n=0.

(1) Master operation

(a) Start~Address~Data~Data~Stop (transmit and receive)

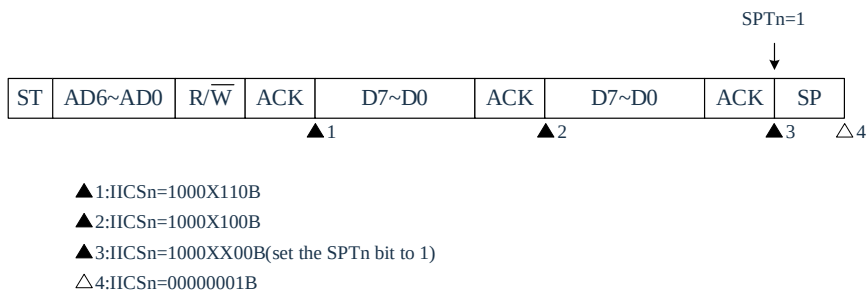
(i) When WTIMn=0



Note: To generate a stop condition, set the WTIM_n bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲ Always generated
 △ Generated only when SPIEn = 1
 X Any

(ii) When WTIMn=1

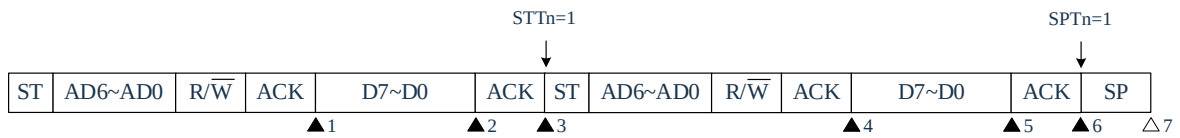


Remark ▲ Always generated
 △ Generated only when SPIEn = 1
 X Any

Remark: n=0.

(b) Start~Address~Data~Start~Address~Data~Stop (restart)

(i) When $WTIM_n=0$



- ▲1: ICSn=1000X110B
 ▲2: ICSn=1000X000B(set the WTIMn bit to 1) ^{Note1}
 ▲3: ICSn=1000XX00B(set the WTIMn bit to 0) ^{Note2} and set the STTn bit to 1)
 ▲4: ICSn=1000X110B
 ▲5: ICSn=1000X000B(set the WTIMn bit to 1) ^{Note3}
 ▲6: ICSn=1000XX00B(set the SPTn bit to 1)
 △7: ICSn=00000001B

Note1. To generate a start condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

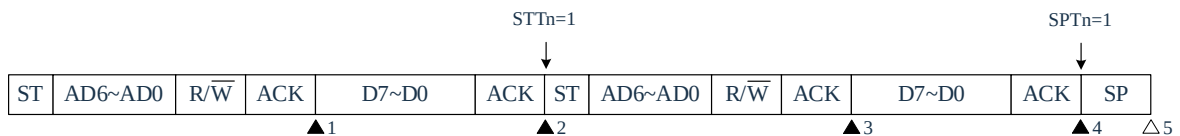
2. Clear the WTIMn bit to 0 to restore the original setting.
3. To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲ Always generated

△ Generated when SPIE =1

X Any

(ii) When $WTIM_n=1$



- ▲1:IICSn=1000X110B
- ▲2:IICSn=1000XX00B(set STTn bit to "1")
- ▲3:IICSn=1000X110B
- ▲4:IICSn=1000XX00B(set SPTn bit to "1")
- △5:IICSn=00000001B

Remark ▲ Always generated

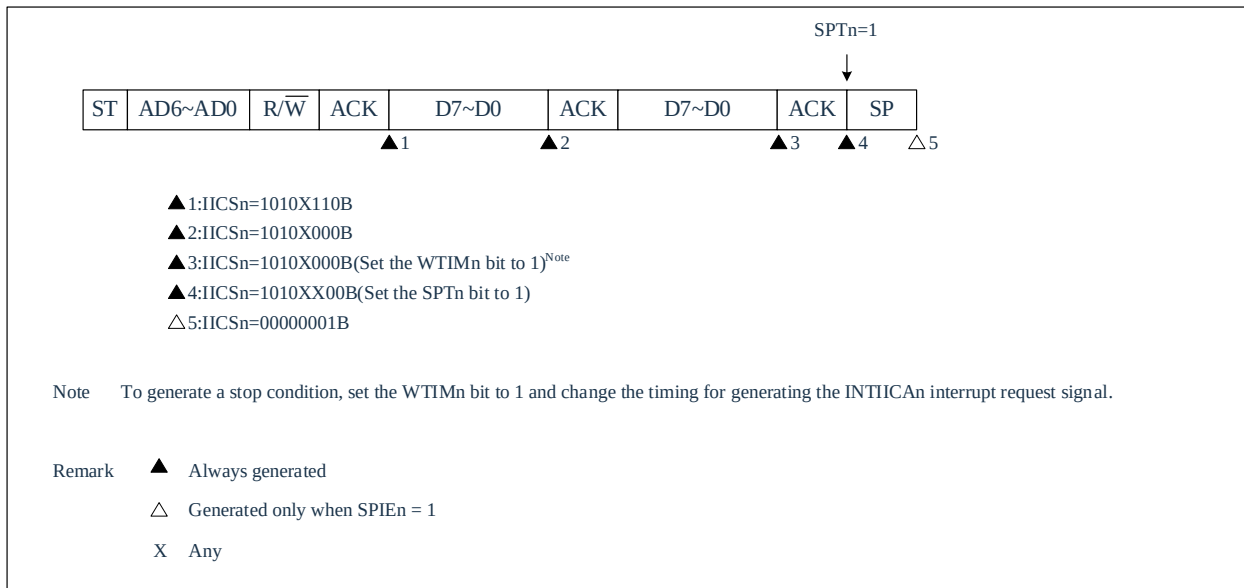
△ Generated only when SPIEn = 1

X Any

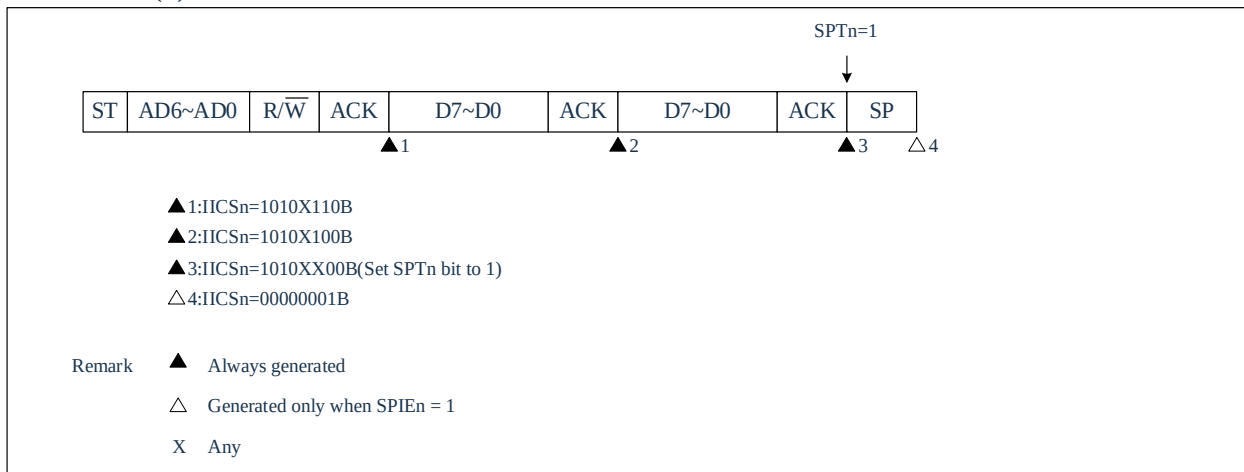
Remark: $n=0$.

(c) Start~Code~Data~Data~Stop (extension code transmission)

(i) When WTIMn=0



(ii) When WTIMn=1



Remark: n=0.

(2) Slave operation (when receiving a slave address)

(a) Start~Address~Data~Data~Stop

(i) When WTIMn=0



▲1:IICS_n=0001X110B

▲2:IICS_n=0001X000B

▲3:IICS_n=0001X000B

△4:IICS_n=00000001B

Remark ▲ Always generated
 △ Generated only when SPIEn = 1
 X Any

(ii) When WTIMn=1



▲1:IICS_n=0001X110B

▲2:IICS_n=0001X100B

▲3:IICS_n=0001XX00B

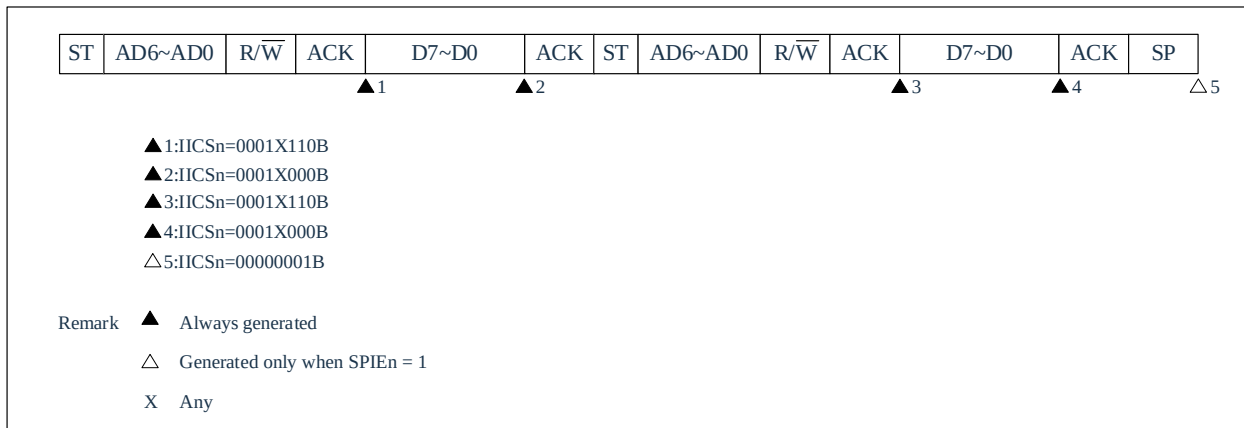
△4:IICS_n=00000001B

Remark ▲ Always generated
 △ Generated only when SPIEn = 1
 X Any

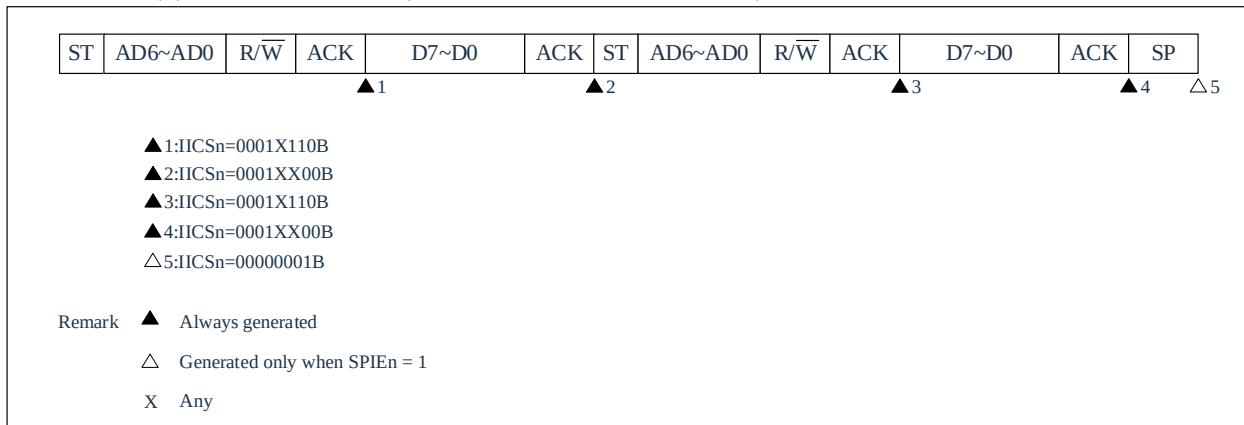
Remark: n=0.

(b) Start~Address~Data~Start~Address~Data~Stop

(i) When WTIMn=0 (after restart, matches with SVAn)



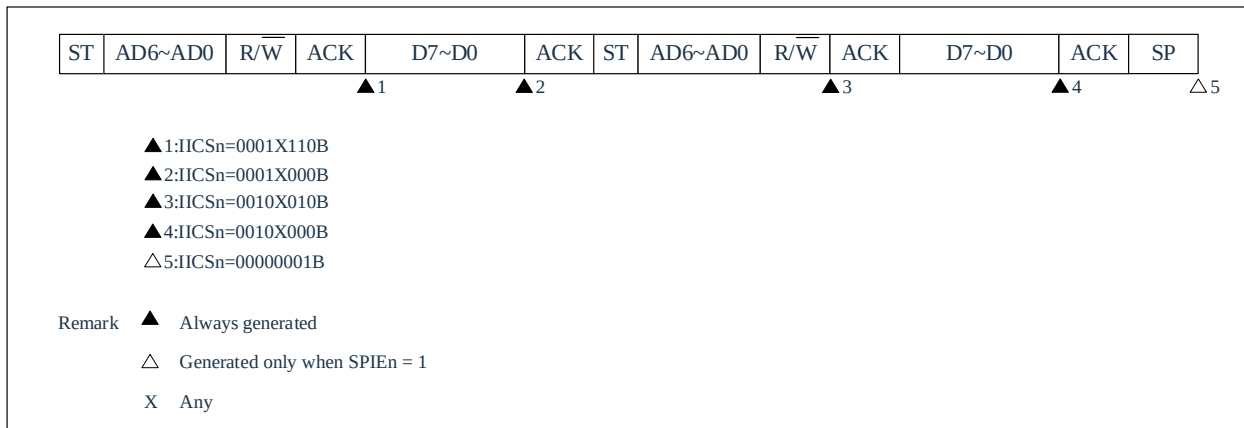
(ii) When WTIMn=1 (after restart, matches with SVAn)



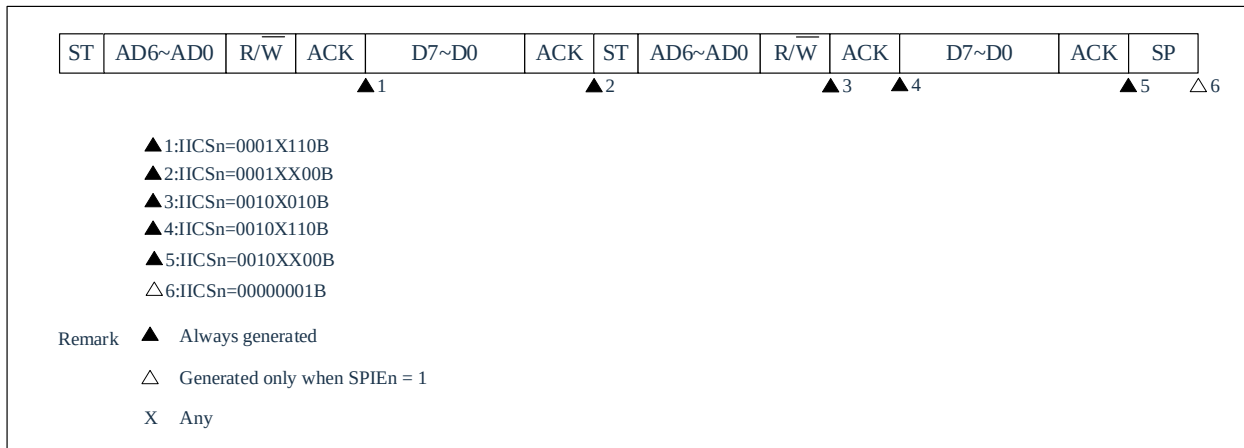
Remark: n=0.

(c) Start~Address~Data~Start~Code~Data~Stop

(i) When WTIMn=0 (after restart, does not match address (= extension code))



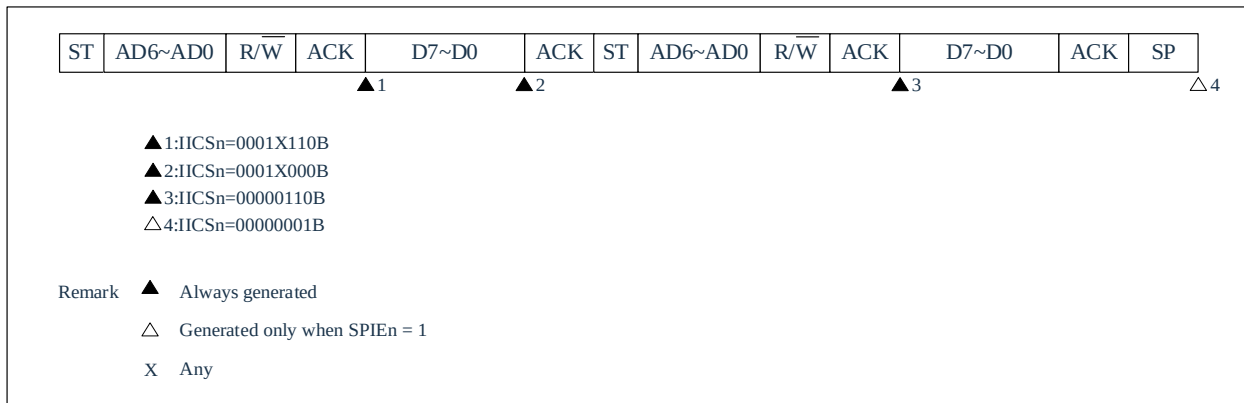
(ii) When WTIMn=1 (after restart, does not match address (= extension code))



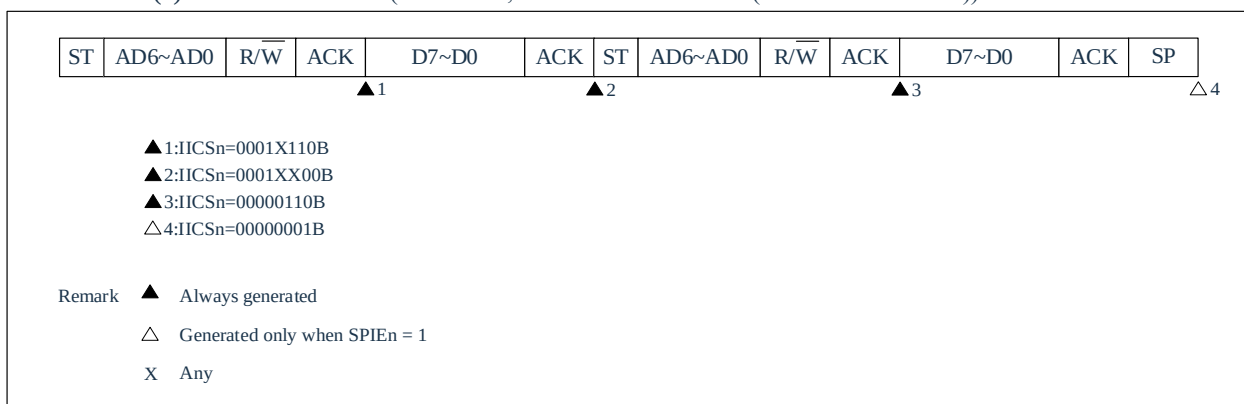
Remark: n=0.

(d) Start~Address~Data~Start~Address~Data~Stop

(i) When WTIMn=0 (after restart, does not match address (= not extension code))



(ii) When WTIMn=1 (after restart, does not match address (= not extension code))



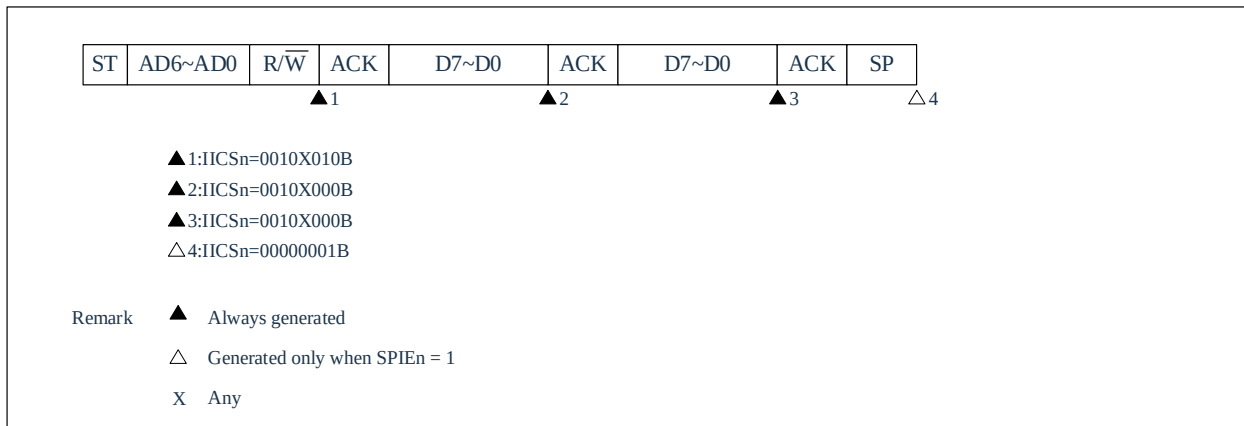
Remark: n=0.

(3) Slave operation (when receiving extension code)

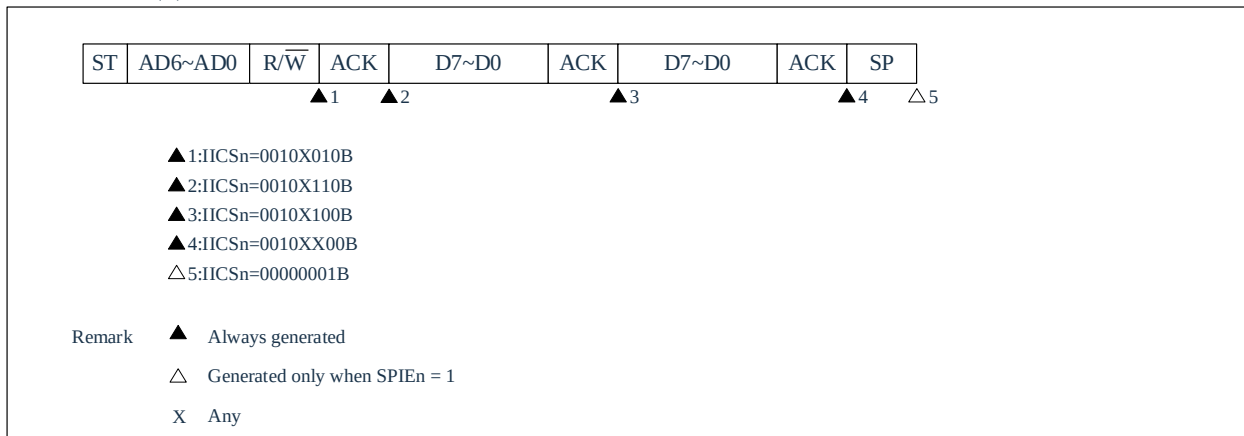
The device is always participating in communication when it receives an extension code.

(a) Start~Code~Data~Data~Stop

(i) When WTIMn=0



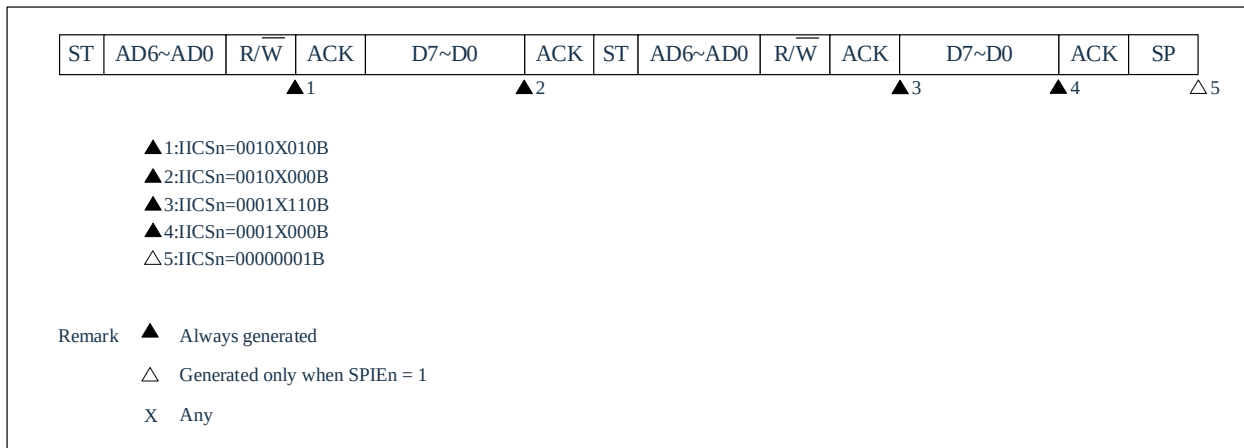
(ii) When WTIMn=1



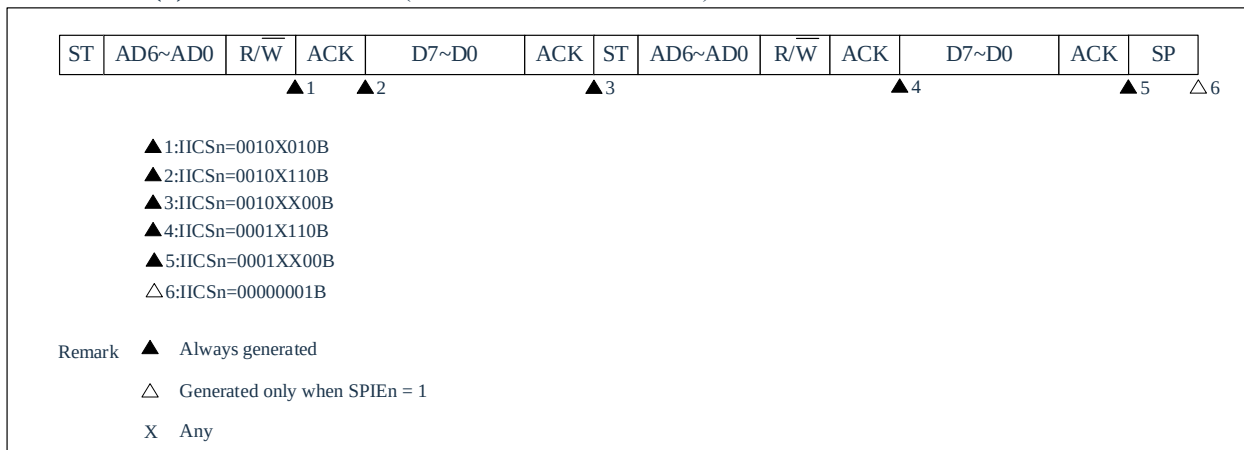
Remark: n=0.

(b) Start~Code~Data~Start~Address~Data~Stop

(i) When WTIMn=0 (after restart, matches SVAn)



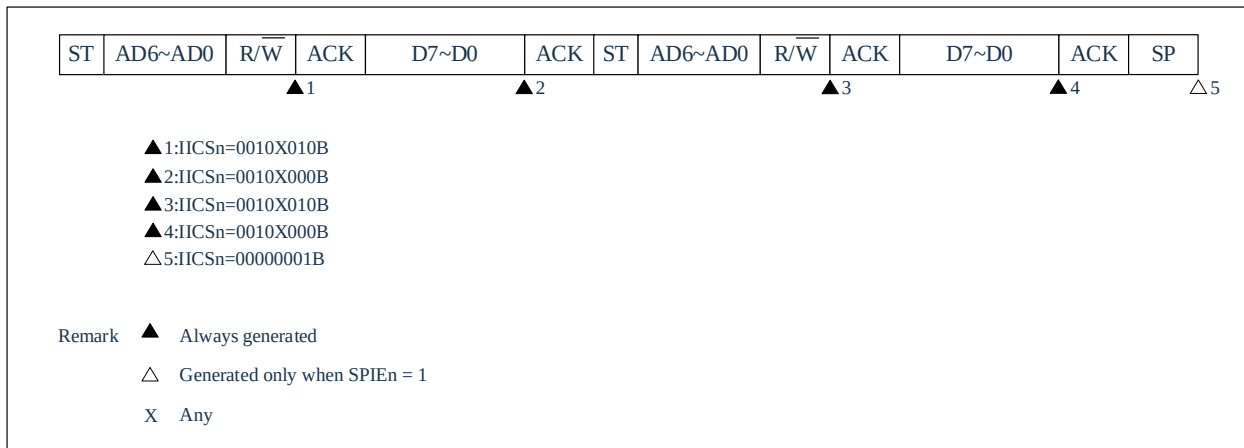
(ii) When WTIMn=1 (after restart, matches SVAn)



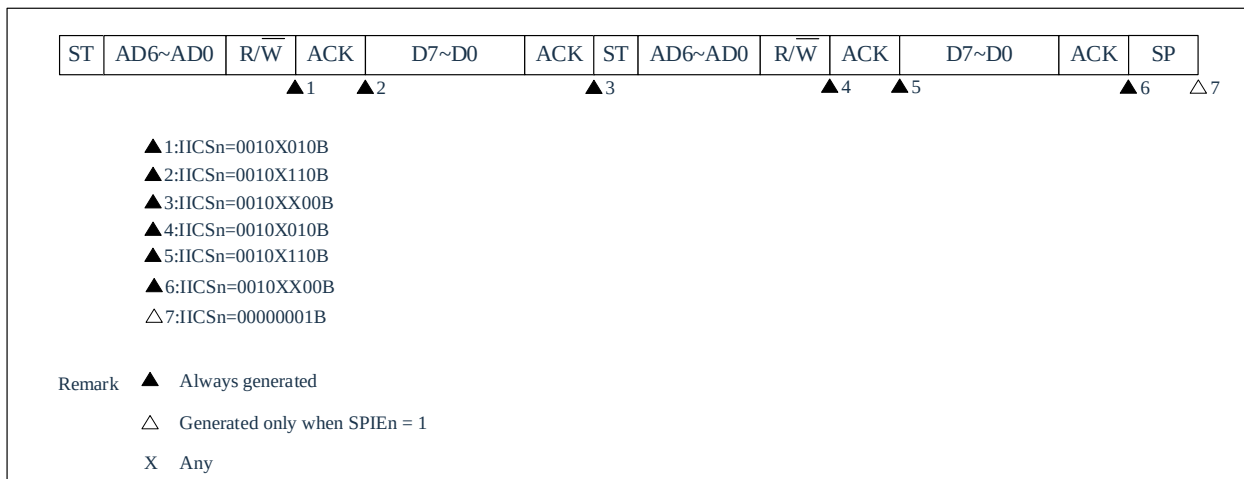
Remark: n=0.

(c) Start~Code~Data~Start~Code~Data~Stop

(i) When WTIMn=0 (after restart, extension code reception)



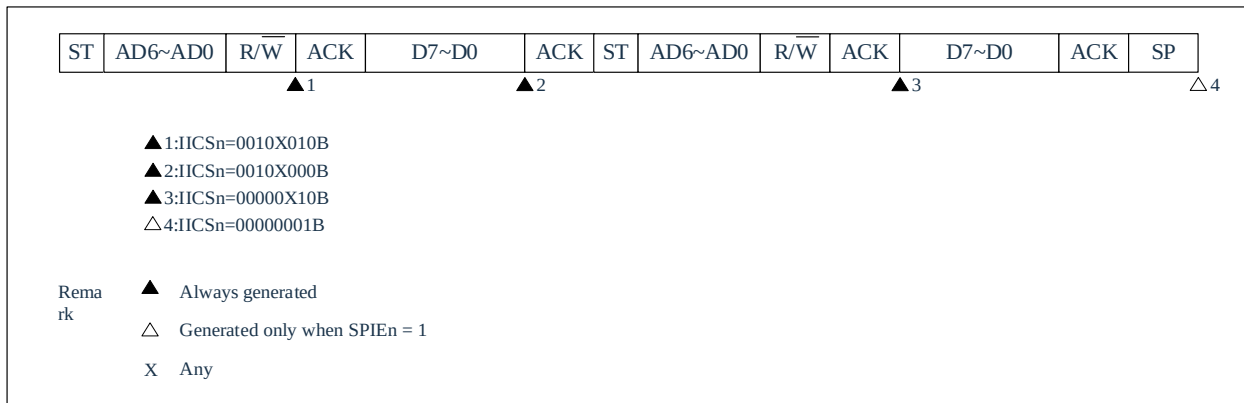
(ii) When WTIMn=1 (after restart, extension code reception)



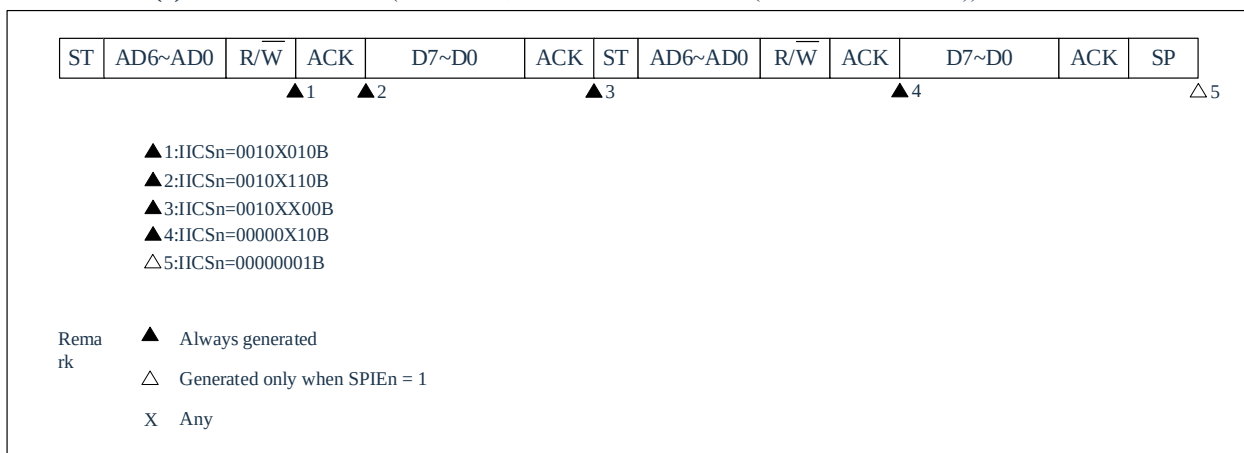
Remark: n=0.

(d) Start~Code~Data~Start~Address~Data~Stop

(i) When WTIMn=0 (after restart, does not match address (= not extension code))



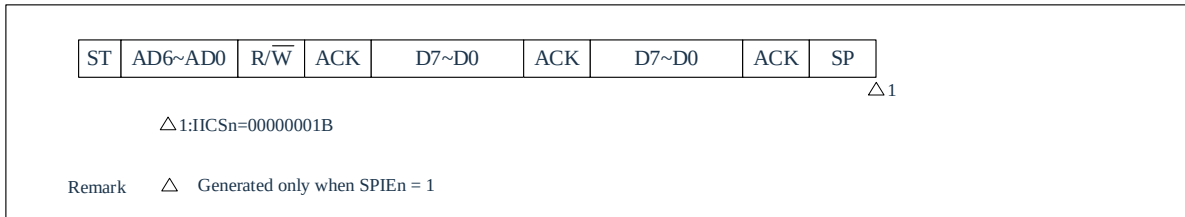
(ii) When WTIMn=1 (after restart, does not match address (= not extension code))



Remark: n=0.

(4) Operation without communication

(a) Start~Code~Data~Data~Stop

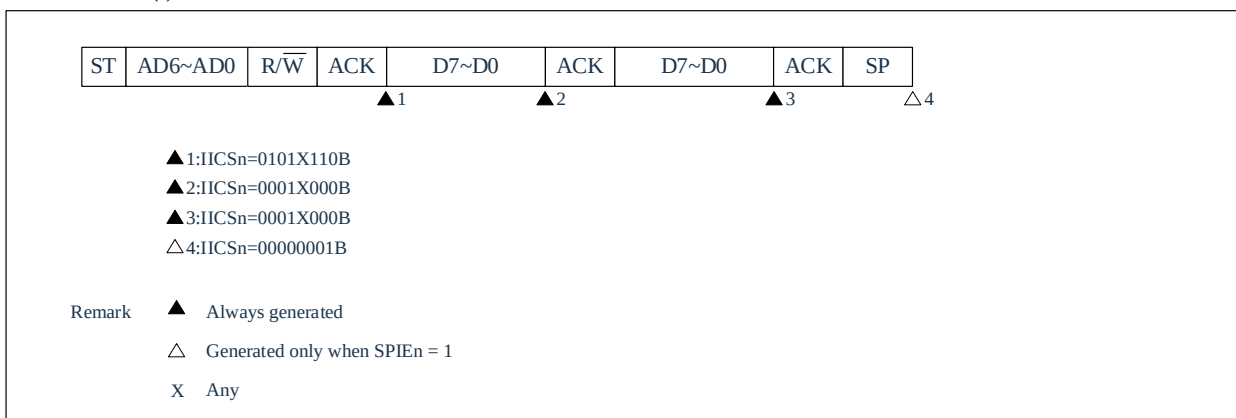


(5) Arbitration loss operation (operation as slave after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTSn bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.

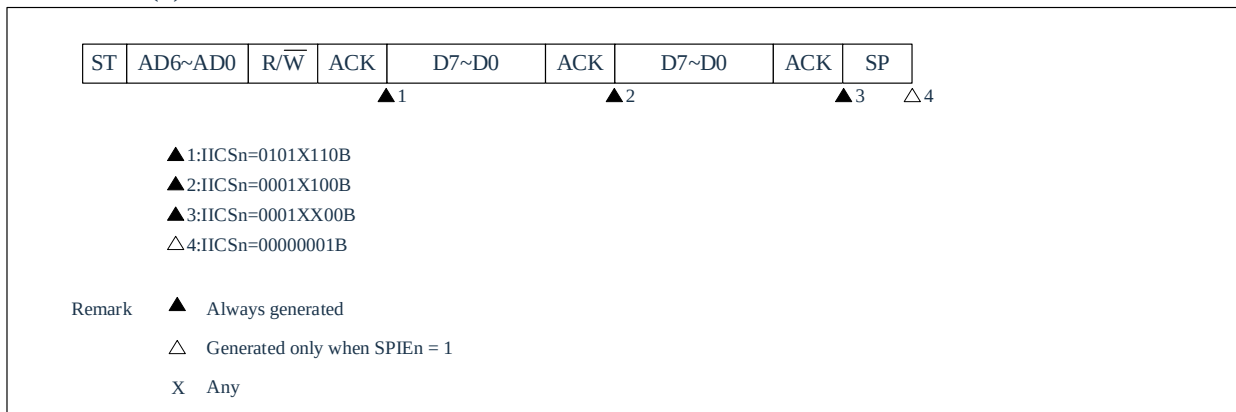
(a) When arbitration loss occurs during transmission of slave address data

(i) When WTIMn=0



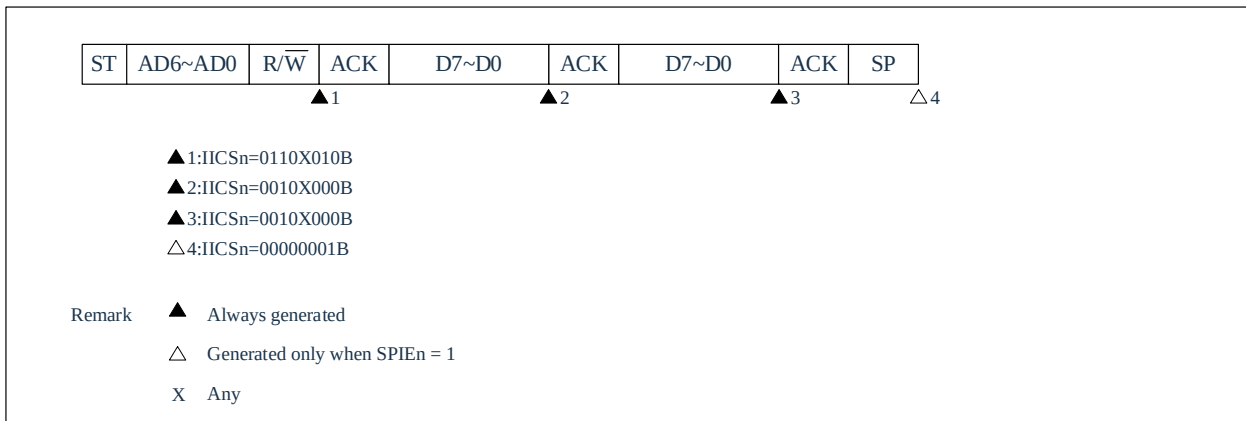
Remark: n=0.

(ii) When WTIMn=1



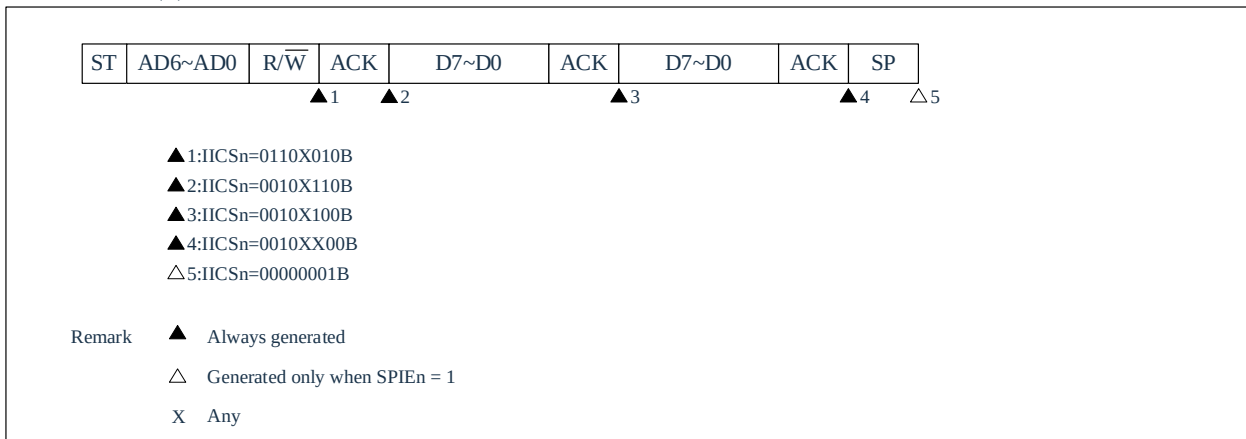
(b) When arbitration loss occurs during transmission of extension code

(i) When WTIMn=0



Remark: n=0.

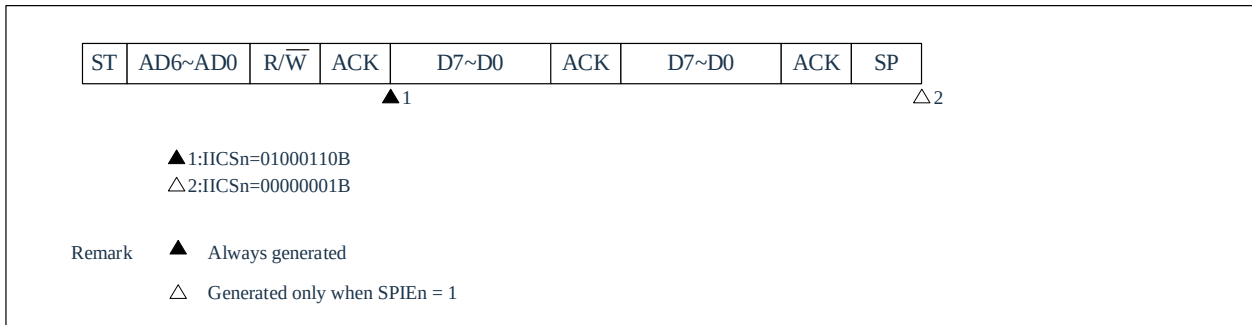
(ii) When WTIMn=1



(6) Operation when arbitration loss occurs (no communication after arbitration loss)

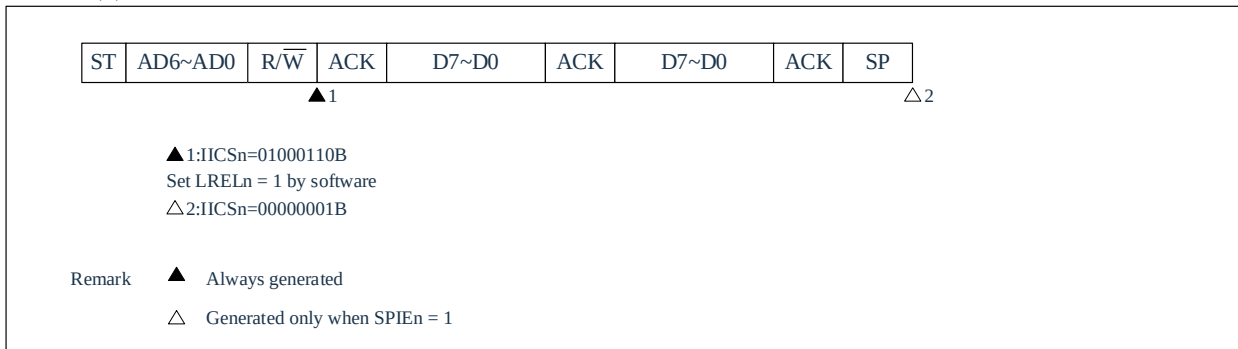
When the device is used as a master in a multi-master system, read the MSTSn bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data (when WTIMn = 1)



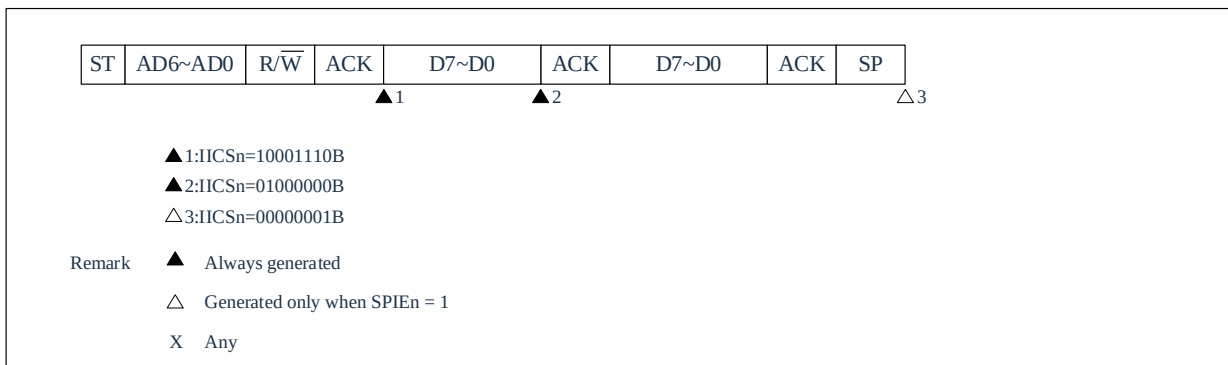
Remark: n=0.

(b) When arbitration loss occurs during transmission of extension code



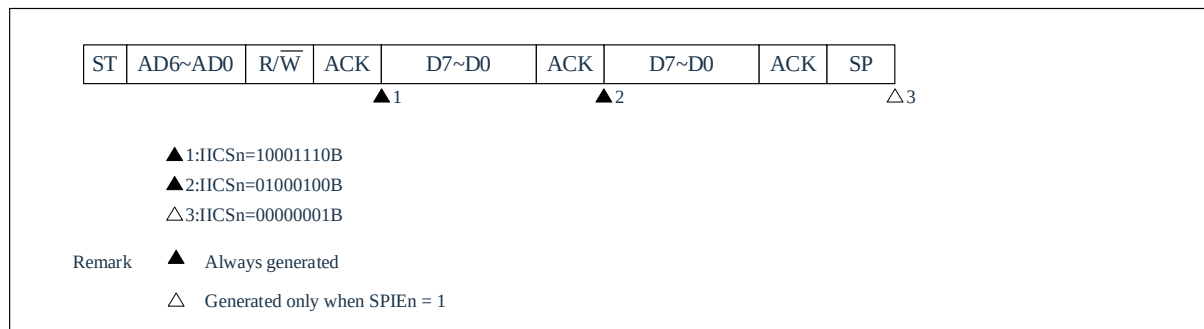
(c) When arbitration loss occurs during transmission of data

(i) When WTIMn=0



Remark: n=0.

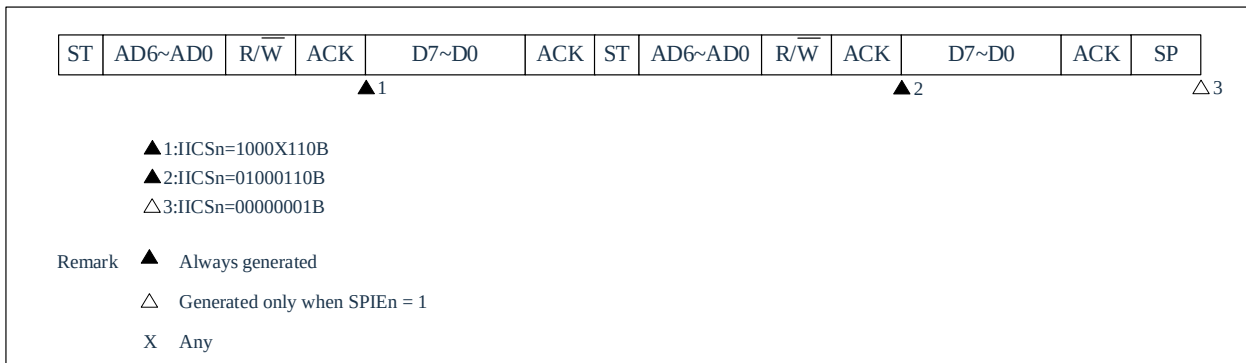
(ii) When WTIMn=1



Remark: n=0.

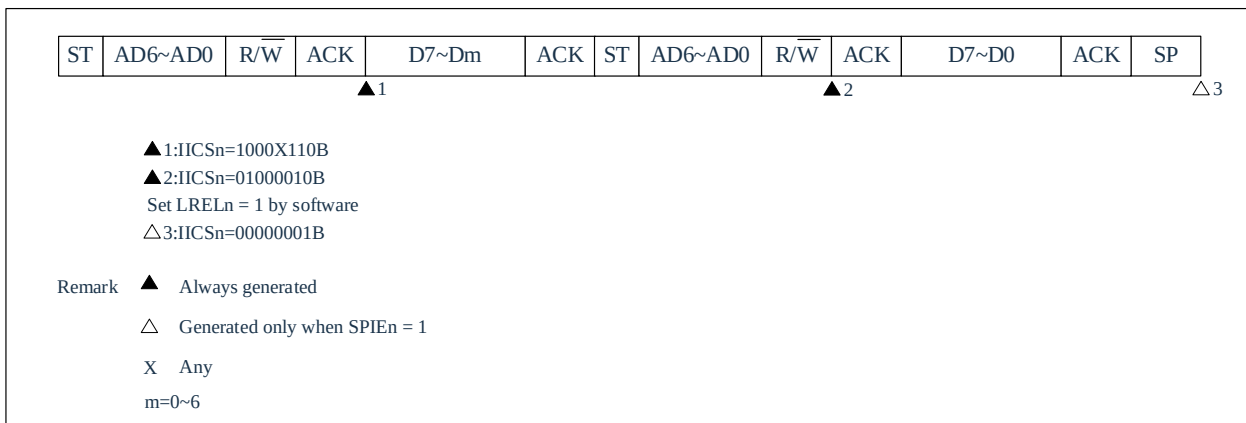
(d) When loss occurs due to restart condition during data transfer

(i) Not extension code (Example: unmatched with SVAn)



Remark: n=0.

(ii) Extension code



(e) When loss occurs due to stop condition during data transfer



▲ 1: ICSn=10000110B

△ 2: ICSn=01000001B

Remark ▲ Always generated

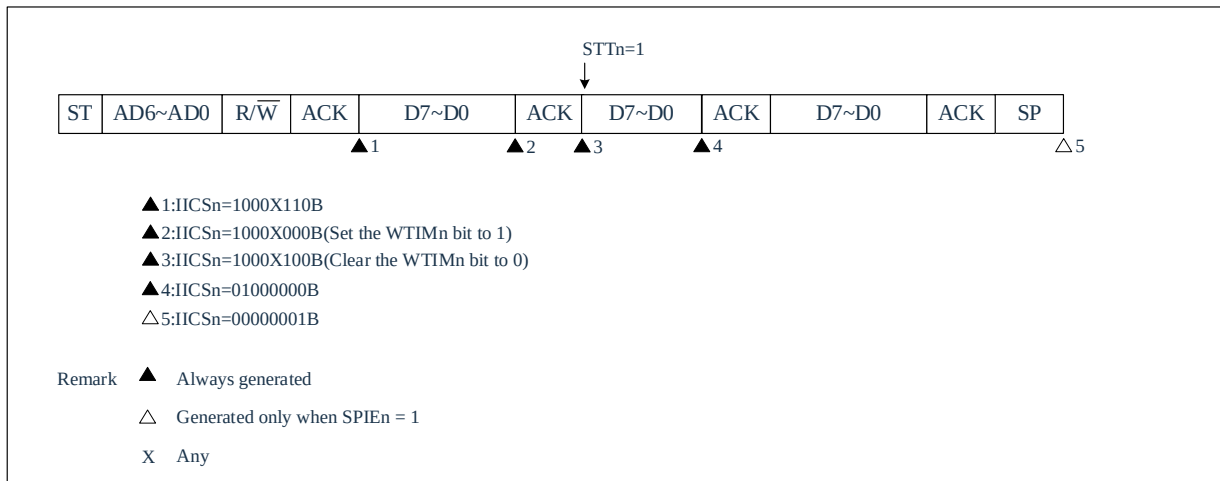
△ Generated only when SPIEn = 1

m=0~6

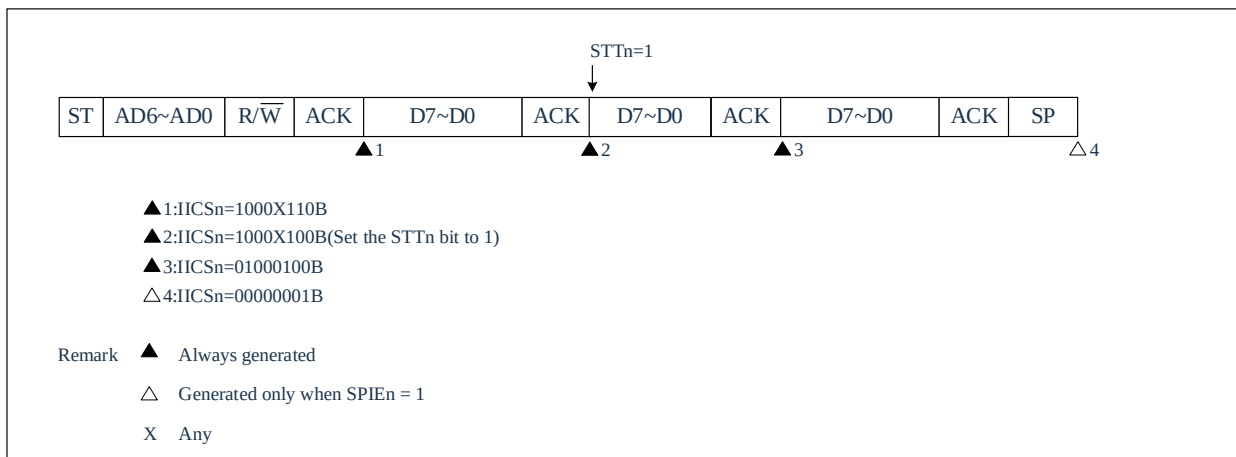
Remark: n=0.

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

(i) When WTIMn=0



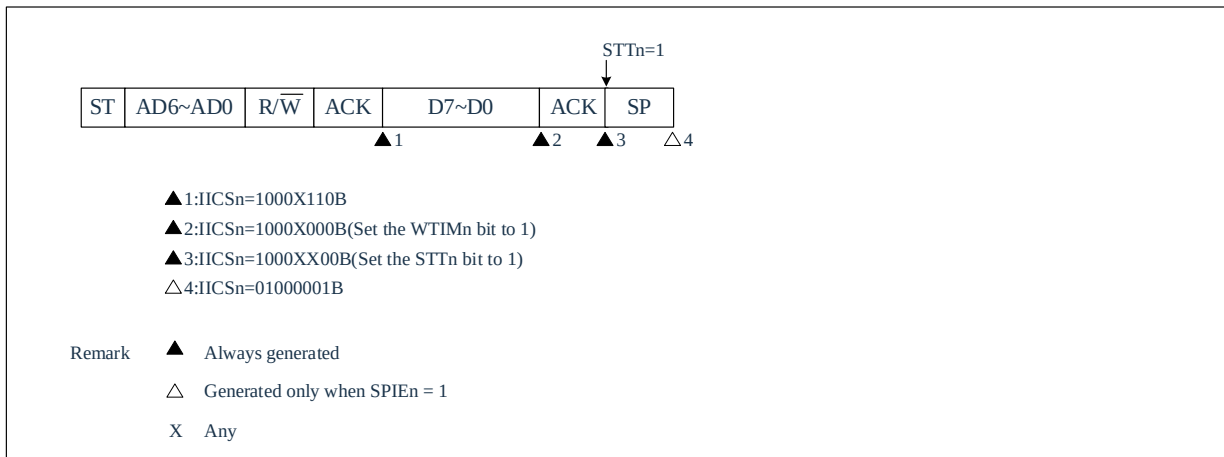
(ii) When WTIMn=1



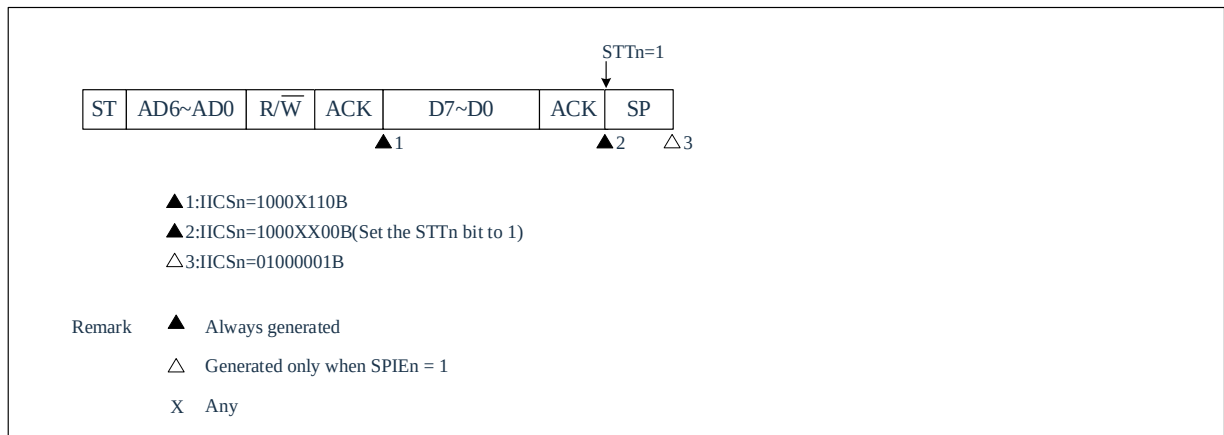
Remark: n=0.

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

(i) When WTIMn=0



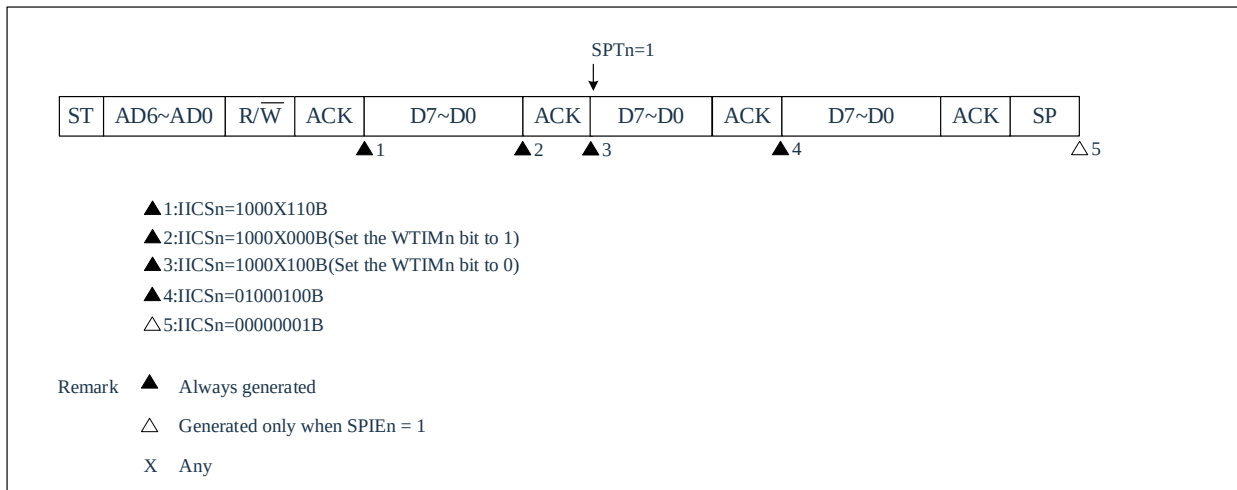
(ii) When WTIMn=1



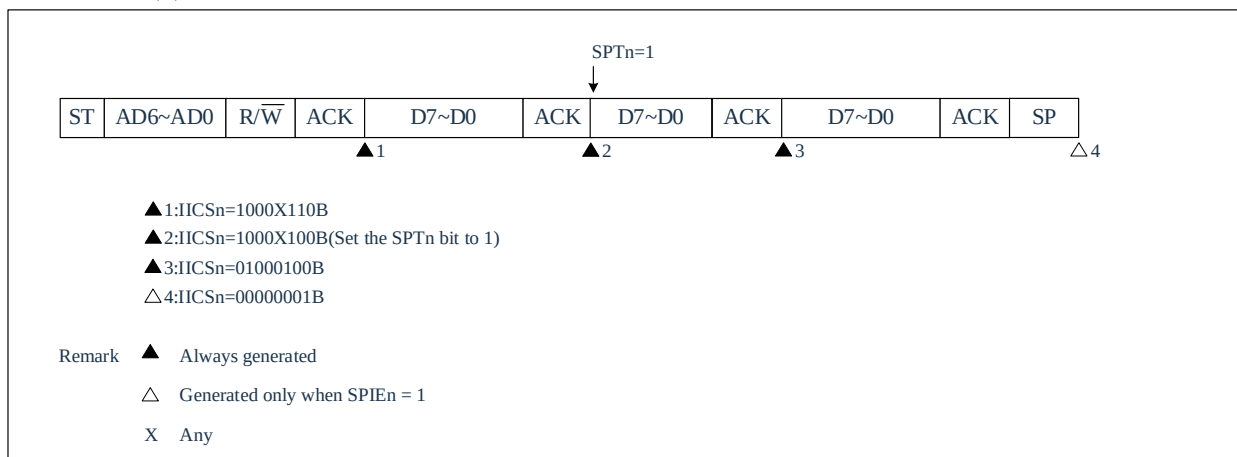
Remark: n=0.

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

(i) When WTIMn=0



(ii) When WTIMn=1



Remark: n=0.

14.6 Timing Diagrams

When using the I²C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner. After outputting the slave address, the master device transmits the TRCn bit (the bit 3 of the IICA status register n (IICSn)), which specifies the data transfer direction, and then starts serial communication with the slave device. Timing diagrams of the data communication are shown in Figure 14-31.

The IICA shift register n (IICAn)'s shift operation is synchronized with the falling edge of the serial clock (SCLAn). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAAn pin.

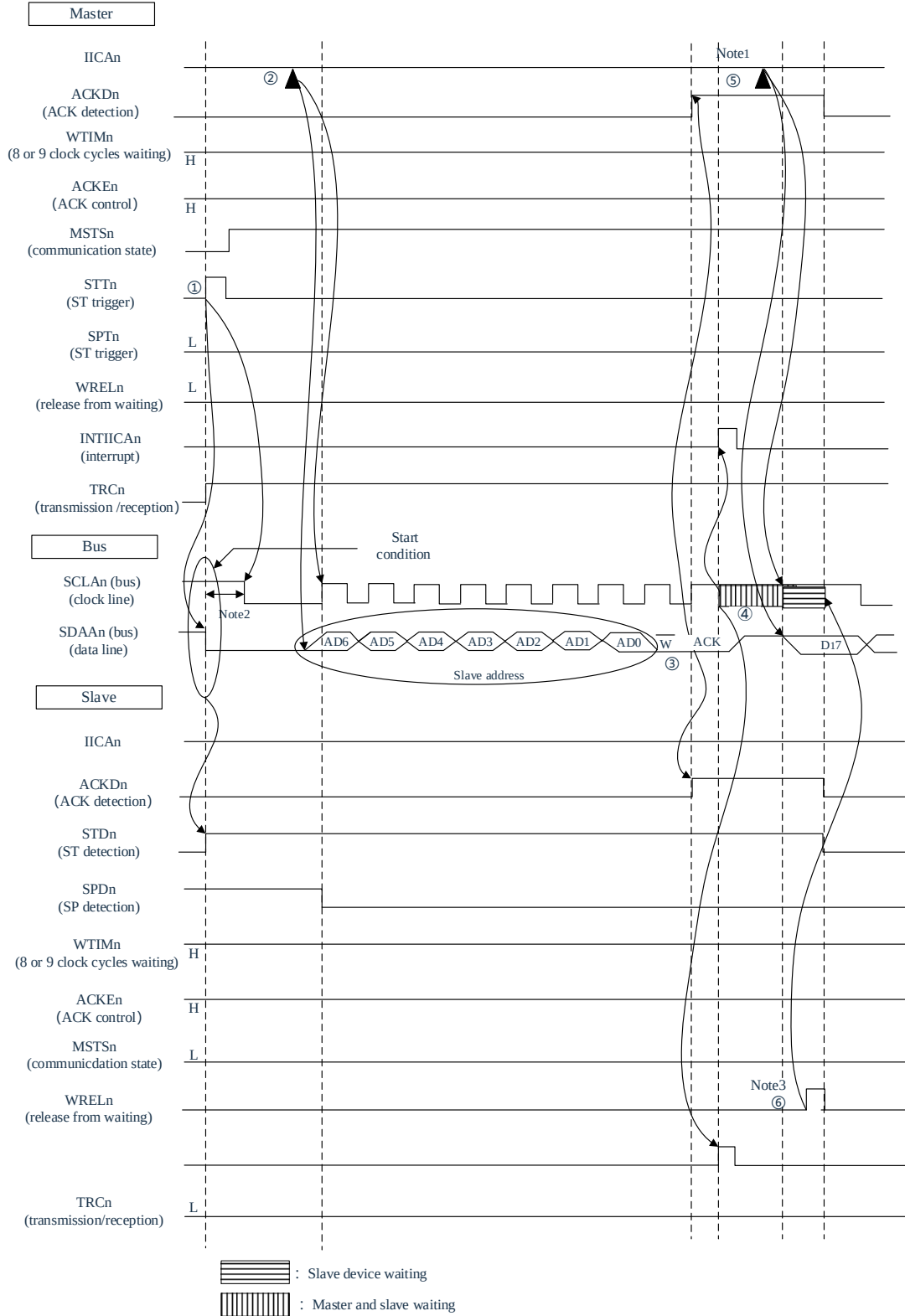
Data input via the SDAAn pin is captured into IICAn at the rising edge of SCLAn.

Remark: n=0.

Figure 14-31 Example of master to slave communication

(Master: selects 9 clocks to wait, slave: selects 9 clocks to wait) (1/4)

(1) Start Condition ~ Address ~ Data



Note 1: To release the master from waiting during transmission, data must be written to IICAn instead of setting the WRELn bit.

Note 2: Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 μ s when specifying standard mode and at least 0.6 μ s when specifying fast mode.

Note 3: For releasing wait state during reception of a slave device, write "FFH" to IICAn or set the WRELn bit.

Figure 14-31 shows the descriptions of ① to ⑥ of “(1) Start condition ~ Address ~ Data”

- ① If the master sets the start condition trigger set (STTn=1), the bus data line (SDAAn) drops and the start condition is generated (SDAAn is changed from 1 to 0 by SCLAn=1). Thereafter, if a start condition is detected, the master enters the master communication state (MSTS_n=1) and after the hold time elapses the bus clock line drops (SCLAn=0), ending the communication preparation.
- ② If the master writes address +W (transmit) to IICAn shift register n (IICAn), the slave address is sent.
- ③ On the slave, if the receiving address and the local station address (the value of the SVAn) are the same ^{Note}, an ACK is sent to the master through hardware. The master detects ACK on the rising edge of the 9th clock (ACKDn=1).
- ④ The master generates an interrupt on the falling edge of the 9th clock (INTIICAn: address send end interrupt). Slaves with the same address enter a waiting state (SCLAn=0) and an interrupt (INTIICAn: address matching interrupt) ^{Note}.
- ⑤ The master writes and transmits data to the IICAn registers, releasing the master of waiting.
- ⑥ If the slave releases the wait (WRELn=1), the master begins to transmit data to the slave.

Note: If the transmitted address and the slave address are different, the slave does not return an ACK (NACK: SDAAn=1) to the master, and does not generate an INTIICAn interrupt (address matching interrupt) or enter a waiting state. However, the master generates ANTIICAn interrupts (address transmit end interrupts) for both ACK and NAK.

Remarks 1. ①~⑮ of Figure 14-31 shows a series of operational steps for data communication via the I²C bus.

“(1) Address~Data~Data” of Figure 14-31 illustrates steps ①~⑥.

“(2) Address~Data~Data” of Figure 14-31 illustrates steps ③~⑩.

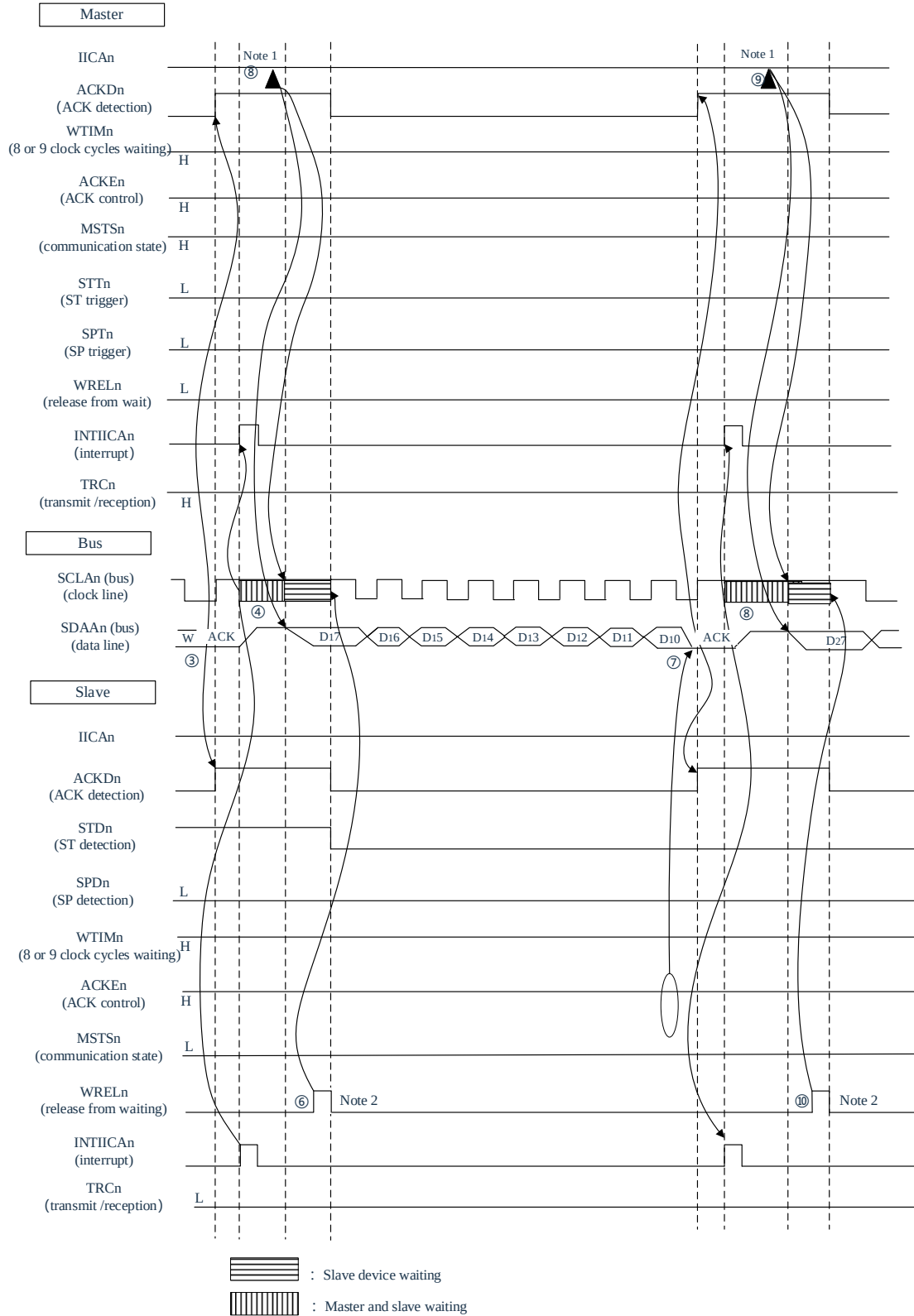
“(3) Data~Data~Stop Condition” of Figure 14-31 illustrates steps ⑦~⑮.

2. n=0.

Figure 14-31 Example of master to slave communication

(Master: selects 9 clocks to wait, slave: selects 9 clocks to wait) (2/4)

(2) Address ~ Data ~ Data


Note 1: To release the master from waiting during transmission, data must be written to IICAn instead of setting the WREL_n bit.

Note 2: For releasing wait state during reception of a slave device, write "FFH" to IICAn or set the WREL_n bit.

Figure 14-31 shows the descriptions of ③ to ⑩ of “(2) Address~Data~Data”:

③ On the slave, if the receiving address and the local station address (the value of the SVAn) are the same ^{Note}, the ACK is sent to the master through the hardware. The master detects ACK on the rising edge of the 9th clock (ACKDn=1).

④ The master generates an interrupt on the falling edge of the 9th clock (INTIICAn: address send end interrupt). Slaves with the same address enter a waiting state (SCLAn=0) and an interrupt (INTIICAn: address matching interrupt)

Note .

⑤ The master writes and sends data to the IICA shift register n (IICAn) to remove the wait of the main controller.

⑥ If the slave releases the wait (WRELn=1), the master party begins to transmit data to the slave.

⑦ After the data transfer is completed, because the ACKEn bit of the slave is 1, the ACK is sent to the master control through hardware. The master detects ACK on the rising edge of the 9th clock (ACKDn=1).

⑧ Both the master and the slave enter a waiting state (SCLAn= 0) on the falling edge of the 9th clock, and both produce interrupts (INTIICAn: Transmit End Interrupt).

⑨ The master controller writes and sends data to the IICAn register to remove the waiting of the main controller.

⑩ If the slave reads and receives the data and cancels the wait (WRELn=1), the master party begins to transmit data to the slave.

Note: If the transmitted address and the slave address are different, the slave does not return an ACK (NACK: SDAAn=1) to the master and does not generate an INTIICAn interrupt (address matching interrupt) or enter a waiting state. However, the master generates ANTIICAn interrupts (address send end interrupts) for both ACK and NAK.

Remarks 1. ①~⑮ of Figure 14-31 shows a series of operational steps for data communication via the I²C bus.

“(1) Start Condition~Address~Data” of Figure 14-31 illustrates steps ①~⑥.

“(2) Address~Data~Data” of Figure 14-31 illustrates steps③~⑩.

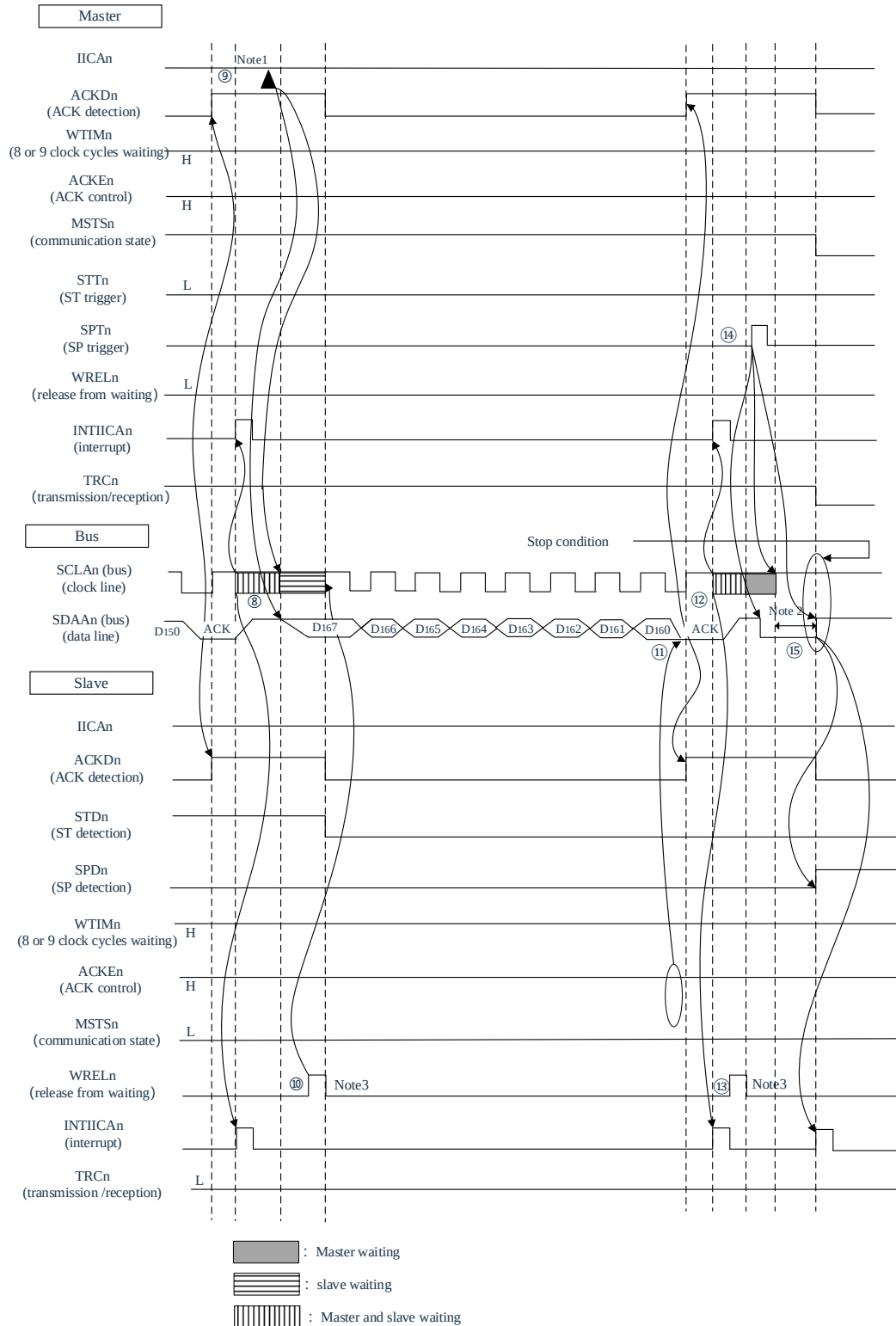
“(3) Data~Data~Stop Condition” of Figure 14-31 illustrates steps ⑦~⑮.

2. n=0.

Figure 14-31 Example of master to slave communication

(Master: selects 9 clocks to wait, slave: selects 9 clocks to wait) (3/4)

(3) Data~Data~Stop Condition



Note 1: To release the master from waiting during transmission, data must be written to IICAn instead of setting the WREL_n bit.

Note 2: After the stop condition is issued, the time from the SCLAn pin signal to generate the stop condition is at least 4.0μs when set to standard mode and at least 0.6μs when set to fast mode.

Note 3: For releasing wait state during reception of a slave device, write "FFH" to IICAn or set the WREL_n bit.

Figure 14-31 shows the descriptions of ⑦~⑮ of “(3) Data ~ Data ~ Stop condition”:

- ⑦ At the end of the data transfer, because the ACKEn bit of the slave is 1, the ACK is sent to the master through the hardware. The master detects ACK on the rising edge of the 9th clock (ACKDn=1)
- ⑧ Both the master and slave enter a waiting state (SCLAn=0) on the falling edge of the 9th clock, and both produce an interrupt (INTIICAn: transfer end interrupt).
- ⑨ The master writes and transmits data to the IICA shift register n (IICAn), releasing the master of waiting.
- ⑩ If the slave reads the received data and releases the wait (WRELn=1), the master starts transmitting data to the slave.
- ⑪ At the end of the data transfer, the slave (ACKEn=1) sends an ACK to the master through the hardware. The master detects ACK on the rising edge of the 9th clock (ACKDn=1).
- ⑫ Both the master and slave enter a waiting state (SCLAn=0) on the falling edge of the 9th clock, and both produce an interrupt (INTIICAn: transfer end interrupt).
- ⑬ The slave reads the received data and releases the wait (WRELn=1).
- ⑭ If the master sets the stop condition trigger bit (SPTn=1), the bus data line (SDAAn=0) is cleared and the bus clock line is set (SCLAn=1), and the bus data line is set after the preparation time for the stop condition is elapsed (SDAAn=1), a stop condition is generated (changes SDAAn from “0” to “1” by SCLAn=1).
- ⑮ If a stop condition is generated, the slave detects the stop condition and generates an interrupt (INTIICAn: stop condition interrupt).

Remarks 1. ①~⑮ of Figure 14-31 shows a series of operational steps for data communication via the I²C bus.

“(1) Start Condition~Address~Data” of Figure 14-31 illustrates steps ①~⑥.

“(2) Address~Data~Data” of Figure 14-31 illustrates steps ③~⑩.

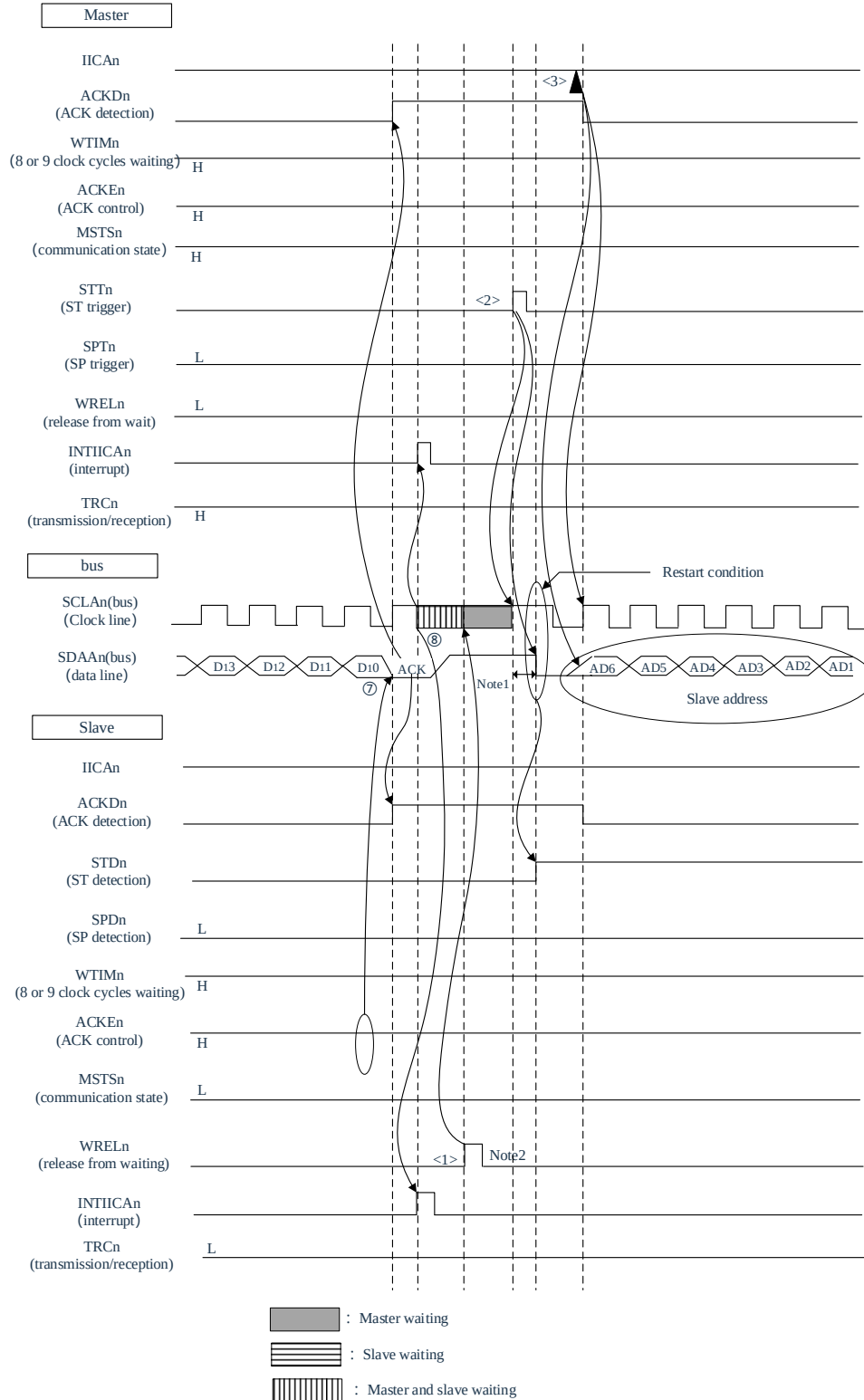
“(3) Data~Data~Stop Condition” of Figure 14-31 illustrates steps ⑦~⑮.

2. n=0.

Figure 14-31 Example of master to slave communication

(Master: selects 9 clocks to wait, slave: selects 9 clocks to wait) (4/4)

(4) Data ~ Restart Condition ~ Address



Note 1: Make sure that the time between the rise of the SCLAn pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

Note 2: For releasing wait state during reception of a slave device, write “FFH” to IICAn or set the WREL_n bit.

Figure 14-31 shows the operation of “(4) Data ~ Restart Condition ~ Address” as follows. After executing steps ⑦ and ⑧, execute <1> to <3> to return to the step ③: data transmit step.

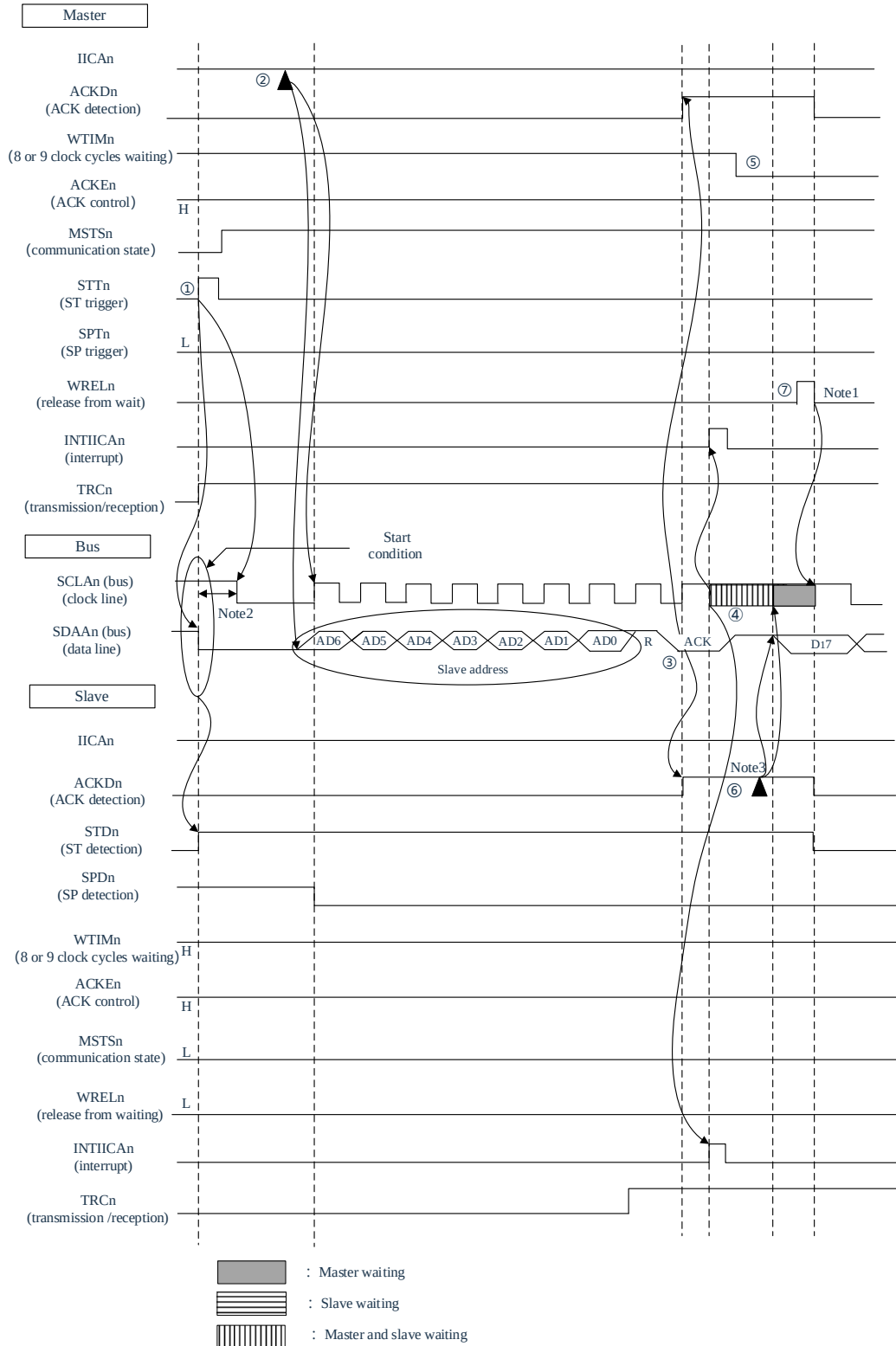
- ⑦ After the data transfer is completed, because the ACKEn bit of the slave is 1, the ACK is sent to the master through hardware. The master detects ACK on the rising edge of the 9th clock (ACKDn=1).
- ⑧ Both the master and the slave enter a waiting state (SCLAn=0) on the falling edge of the 9th clock, and both produce interrupts (INTIICAn: transfer end interrupt).
 - <1> The slave reads and receives the data, and the wait is released (WRELn=1).
 - <2> If the master device triggers the start condition again (STTn=1), the bus clock line rises (SCLAn=1). After the preparation time for the restart condition, the bus data line falls (SDAAn=0), generating the start condition (by causing SDAAn to change from 1 to 0 through SCLAn=1). Then, if the start condition is detected, the bus clock line falls (SCLAn=0) after the hold time, ending the communication preparation.
 - <3> The master device writing the address + R/W (transmission) to the IICA shift register (IICAn) enables the slave address to be transmitted.

Remark: n=0

Figure 14-32 Example of slave to master communication

(Master: selects 8 clocks to wait, slave: selects 9 clocks to wait) (1/3)

(1) Start Condition ~ Address ~ Data



Note 1: To release the master from waiting during transmission, the IICn must be set to "FFH" or the WRELn bit must be set.

Note 2: Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 μ s when specifying standard mode and at least 0.6 μ s when specifying fast mode.

Note 3: To release the slave from waiting during transmission write data to the IICn instead of setting the WRELn bit.

Figure 14-32 shows ① to ⑦ of “(1) Start condition~Address~Data” as follows.

① If the master device sets the start condition (STTn=1), the bus data line (SDAAn) falls, generating the start condition (by causing SDAAn to change from 1 to 0 through SCLAn=1). Afterward, if the start condition is detected, the master device enters the master communication state (MSTSn=1), and after the hold time, the bus clock line falls (SCLAn=0), ending the communication preparation.

② If the master writes address +R (receive) to the IICA shift register n (IICAn), the slave address is sent.

③ On the slave, if the receiving address and the local station address (the value of the SVAn) are the same ^{Note}, an ACK is sent to the master through hardware. The master detects ACK on the rising edge of the 9th clock (ACKDn=1).

④ The master generates an interrupt on the falling edge of the 9th clock (INTIICAn: address transmit end interrupt). Slaves with the same address enter a waiting state (SCLAn=0) and an interrupt (INTIICAn: address matching interrupt)

Note

⑤ The master changes the wait sequence to the 8th clock (WTIMn=0).

⑥ The slave writes and sends data to the IICAn register, releasing the slave of the wait.

⑦ The master releases the wait (WRELn=1) and begins data transfer from the slave.

Note: If the transmitted address and the slave address are different, the slave does not return an ACK (NACK: SDAAn=1) to the master and does not generate an INTIICAn interrupt (address matching interrupt) or enter a waiting state. However, the master generates ANTIICAn interrupts (address transmit end interrupt) for both ACK and NAK.

Remarks 1. ①~⑱ of Figure 14-32 shows a series of operational steps for data communication via the I²C bus.

“(1) Start Condition ~ Address ~ Data” of Figure 14-32 illustrates steps ①~⑦.

“(2) Address~Data~Data” of Figure 14-32 illustrates steps ③~⑫.

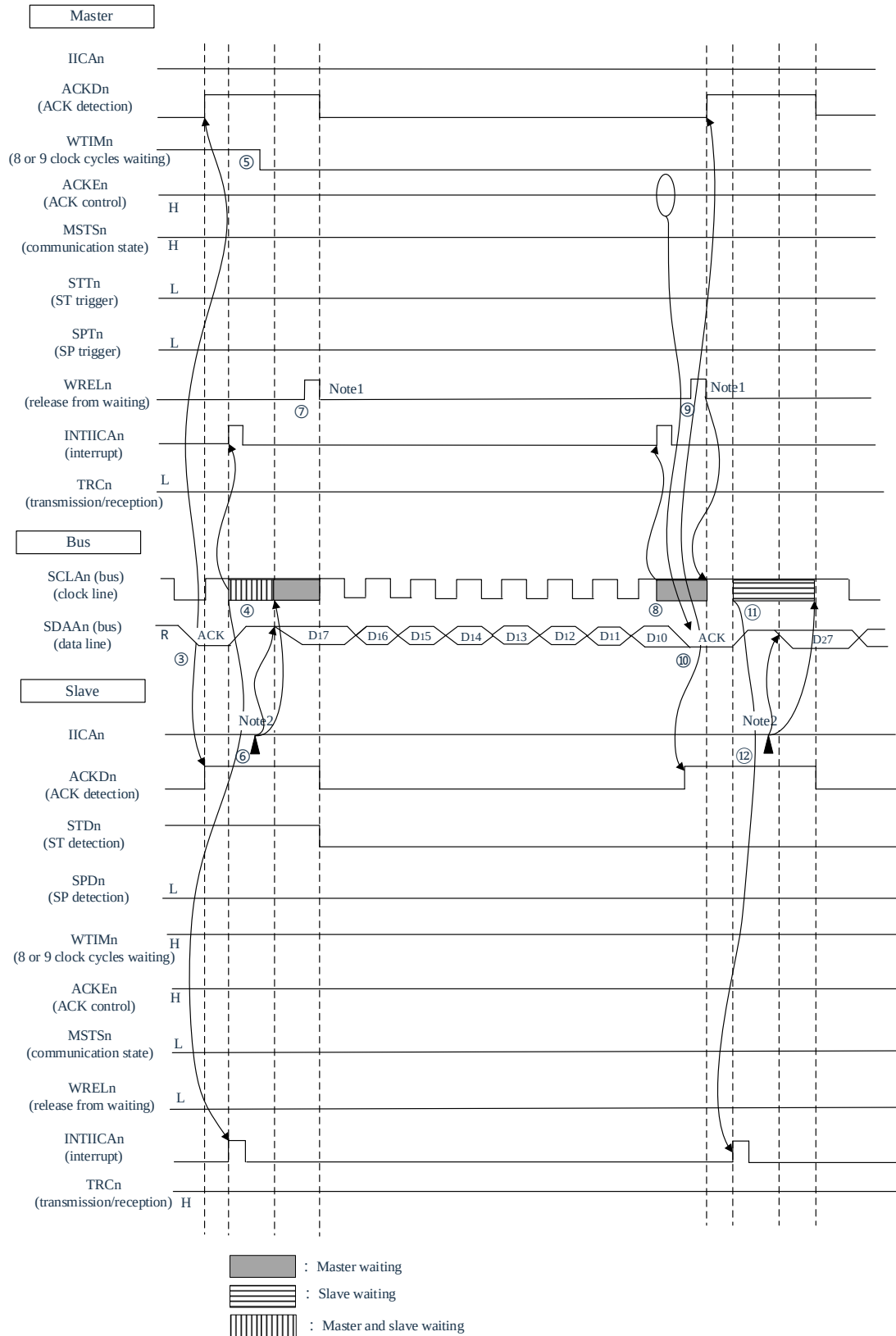
“(3) Data~Data~Stop Condition” of Figure 14-32 illustrates steps ⑧~⑱.

2. n=0.

Figure 14-32 Example of slave to master communication

(Master: selects 8 clocks to wait, slave: selects 9 clocks to wait) (2/3)

(2) Address $\sim$ Data $\sim$ Data



Note 1: To release the master from waiting during reception, the IICn must be set to “FFH” or the WRELn bit must be set.

Note 2: To release the slave from waiting during transmission, write data to the IICn instead of setting the WRELn bit.

Figure 14-32 shows ③~⑫ of “(2) Address~Data~Data” as follows.

③ On the slave, if the receiving address and the local station address (the value of the SVAn) are the same ^{Note}, the ACK is sent to the master through the hardware. The master detects ACK on the rising edge of the 9th clock (ACKDn=1).

④ The master generates an interrupt on the falling edge of the 9th clock (INTIICAn: address transmit end interrupt). Slaves with the same address enter a waiting state (SCLAn=0) and an interrupt (INTIICAn: address matching interrupt)

Note

⑤ The master changes the wait sequence to the 8th clock (WTIMn= 0).

⑥ The slave writes and sends data to the IICA shift register n (IICAn) to release the slave’s wait.

⑦ The master releases the wait (WRELn=1) and begins data transfer from the slave.

⑧ The master enters a waiting state (SCLAn= 0) on the falling edge of the 8th clock and produces an interrupt (INTIICAn: Transmit End Interrupt). Because the ACKEn bit of the master is 1, the ACK is sent to the slave through the hardware.

⑨ The master reads the received data and releases the wait (WRELn=1).

⑩ The slave detects ACK (ACKDn=1) on the rising edge of the 9th clock.

⑪ The slave enters a waiting state on the falling edge of the 9th clock (SCLAn = 0) and produces an interrupt (INTIICAn: transfer end interrupt)

⑫ If the slave writes and transmits data to the IICAn register, the slave’s wait is released and the data transfer from the slave to the master is started.

Note: If the transmitted address and the slave address are different, the slave does not return an ACK (NACK: SDAAn=1) to the master and does not generate an INTIICAn interrupt (address matching interrupt) or enter a waiting state. However, the master generates ANTIICAn interrupts (address transmit end interrupt) for both ACK and NAK.

Remarks 1. ①~⑲ of Figure 14-32 shows a series of operational steps for data communication via the I²C bus.

“(1) Start Condition ~ Address ~ Data” of Figure 14-32 illustrates steps ①~⑦.

“(2) Address~Data~Data” of Figure 14-32 illustrates steps ③~⑫.

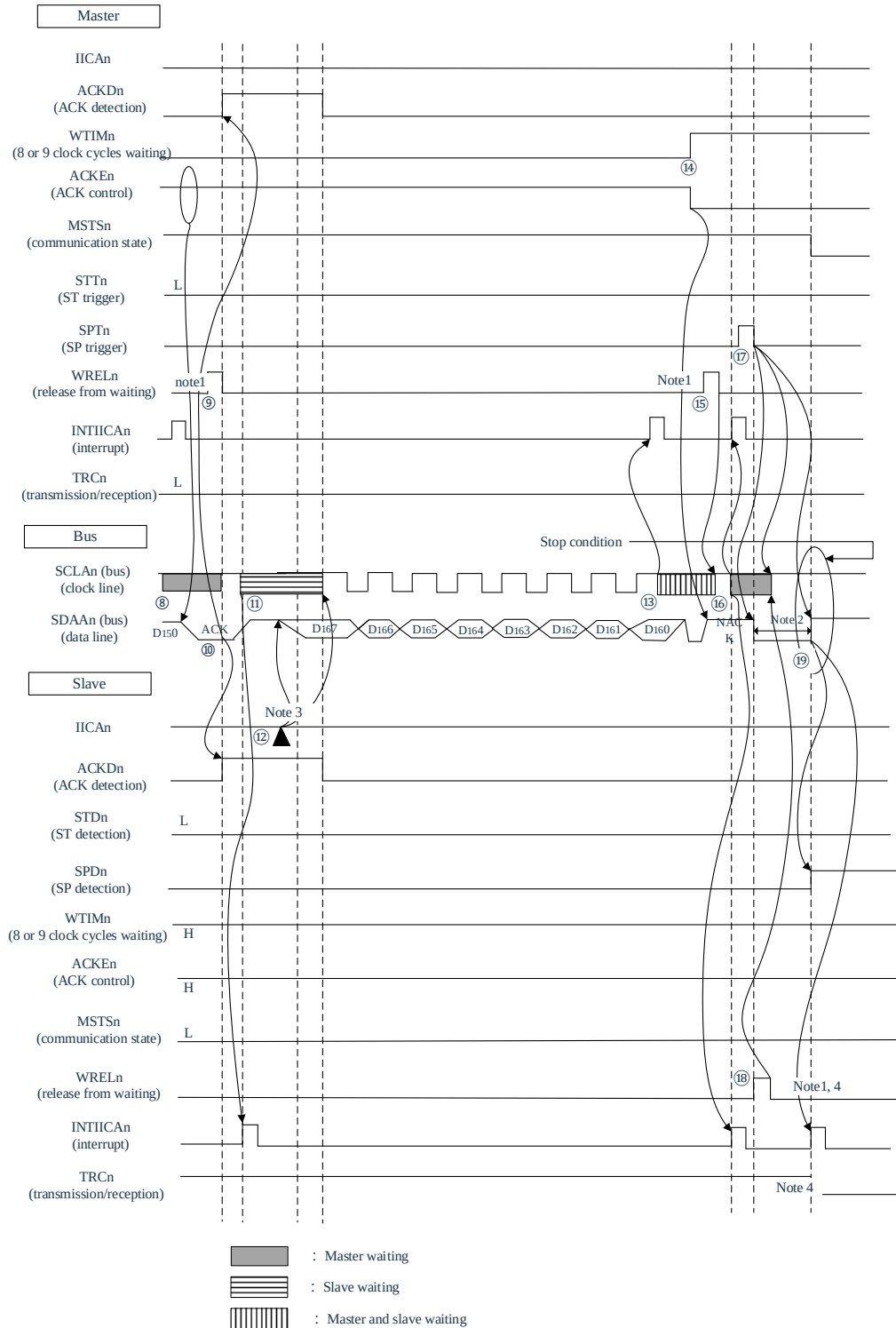
“(3) Data~Data~Stop Condition” of Figure 14-32 illustrates steps ⑧~⑲.

2. n=0.

Figure 14-32 Example of slave to master communication

(Master: selects 8 to 9 clocks to wait, slave: selects 9 clocks to wait) (3/3)

(3) Data ~ Data ~ Stop Condition



Note 1: To release the wait, the IICAn must be set to "FFH" or the WREL_n bit must be set.

Note 2: After the stop condition is issued, the time from the SCLAn pin signal to generate the stop condition is at least 4.0μs when set to standard mode and at least 0.6μs when set to fast mode.

Note 3: To release the slave from waiting during transmission, you must write data to the IICn instead of setting the WREL_n bit.

Note 4: The TRC_n bit is cleared if the wait is released by setting the WREL_n bit during the slave transmission.

Figure 14-32 shows ⑧~⑲ of “(3) Data~Data~Stop Condition” as follows.

- ⑧ The master enters a waiting state ($SCLAn = 0$) on the falling edge of the 8th clock and generates an interrupt (INTIICAn: transfer end interrupt). Because the ACKEn bit of the master is 0, the ACK is sent to the slave through the hardware
- ⑨ The master reads the received data and releases the wait ($WRELn=1$).
- ⑩ The slave detects ACK ($ACKDn=1$) on the rising edge of the 9th clock.
- ⑪ The slave enters a waiting state on the falling edge of the 9th clock ($SCLAn= 0$) and generates an interrupt (INTIICAn: transmit end interrupt).
- ⑫ If the slave writes and transmits data to the IICA shift register n (IICAn), the slave's wait is released and the transfer of data from the slave to the master begins.
- ⑬ The master generates an interrupt (INTIICAn: transmit end interrupt) on the falling edge of the 8th clock and enters a waiting state ($SCLAn=0$). Because ACK control ($ACKEn=1$) occurs, the bus data line at this stage becomes low ($SDAAn=0$)
- ⑭ The master sets the NACK Acknowledge ($ACKEn=0$) and changes the wait sequence to the 9th clock ($WTIMn=1$). If the master releases the wait ($WRELn=1$), the slave detects NACK ($ACKDn=0$) on the rising edge of the 9th clock.
- ⑮ Both the master and slave enter a waiting state ($SCLAn=0$) on the falling edge of the 9th clock, and both produce an interrupt (INTIICAn: transfer end interrupt).
- ⑯ If the master issues a stop condition ($SPTn=1$), the bus data cable ($SDAAn=0$) is cleared and the master's wait is released. After that, the master is on standby until the bus clock line is set ($SCLAn=1$).
- ⑰ The slave stops sending after confirming the NACK, in order to end the communication, the wait is released ($WRELn=1$). If the slave wait is released, the bus clock line is set ($SCLAn=1$).
- ⑱ If the master confirms that the bus clock line is set ($SCLAn=1$), the bus data line is set after the stop condition preparation time has elapsed.
- ⑲ ($SDAAn=1$), and then issue a stop condition ($SDAAn$ is changed from 0 to 1 by $SCLAn=1$). If a stop condition is generated, the slave detects the stop condition and generates an interrupt (INTIICAn: stop condition interrupt).

Chapter 15 IrDA

IrDA (Infrared Data Association) functionality is implemented through collaboration with the General Serial Communication Unit (SCI), enabling the transmission and reception of signals that comply with the IrDA 1.0 protocol.

15.1 Functions of IrDA

When the IrDA function is enabled by setting the IRE bit in the IRCR register, the TxD2 and RxD2 signals of the SCI can encode or decode waveforms that conform to the IrDA 1.0 protocol (via the IrTxD/IrRxD pins). These signals are then transmitted or received by an infrared transmitter or receiver, enabling infrared communication that supports the IrDA 1.0 protocol.

In systems that support the IrDA 1.0 protocol, communication begins at a rate of 9600 bps, and the transfer rate can be adjusted as needed. IrDA does not have an inherent mechanism for automatic rate adjustment, so changes in transfer rate must be made through software configuration.

When selecting a high-speed internal oscillator ($f_{IH} = 24, 12, 6, 3$ MHz), the following baud rates can be configured:

- 115.2kbps, 57.6kbps, 38.4kbps, 19.2kbps, 9600bps, 2400bps

A block diagram illustrating the collaboration between IrDA and SCI is shown in Figure 15-1.

Figure 15-1 Block diagram of IrDA and SCI

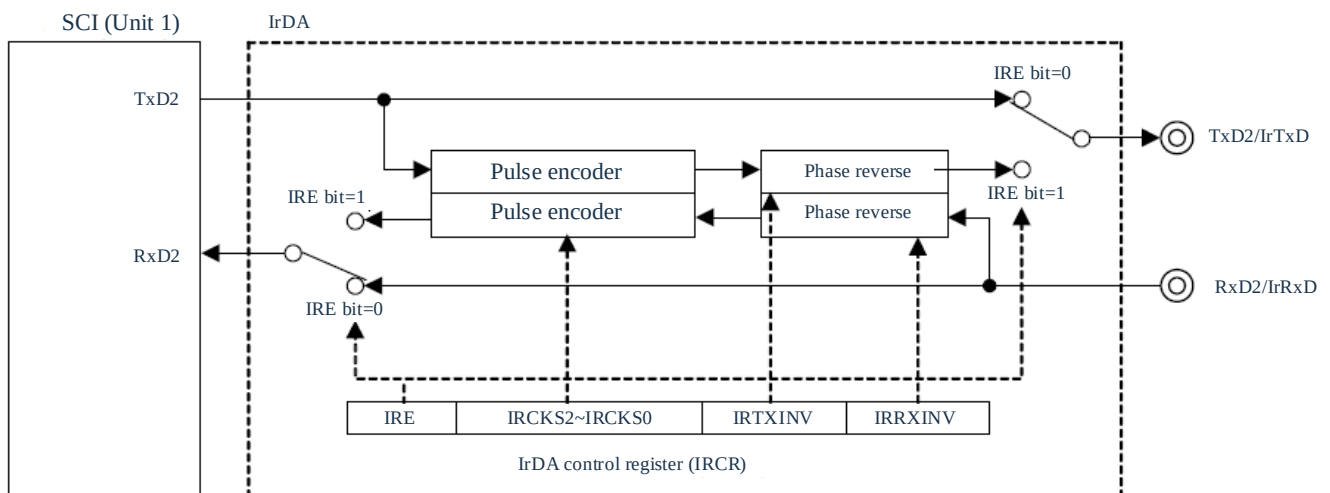


Table 15-1 IrDA pin structure

Pin name	Input/output	Function
IrTxD	Output	The output pin that transmits the data
IrRxD	Input	The input pin that receives the data

15.2 Registers for Controlling IrDA

The IrDA functionality is controlled through the following registers:

- Peripheral Enable Register 0 (PER0)
- IrDA Control Register (IRCR)

15.2.1 Peripheral Enable Register 0 (PER0)

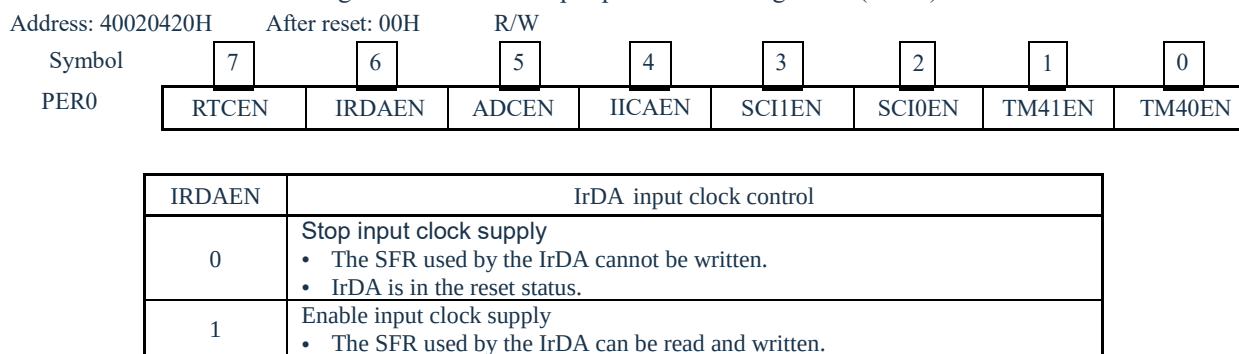
The PER0 register is used to enable or disable the clock to various peripheral hardware units. By stopping the clock to unused hardware, power consumption and noise can be reduced.

To enable IrDA functionality, bit 6 (IRDAEN) must be set to “1”.

The PER0 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”.

Figure 15-2 Format of peripheral enable register 0 (PER0)



Caution 1. When setting up IrDA, the IRDAEN bit must be set to “1” first. When IRDAEN is “0”, write operations to the IrDA control registers are ignored, and all read values will be the initial values.

15.2.2 IrDA Control Register (IRCR)

This is the register that controls the IrDA functionality. It is used for switching the polarity of received and transmitted data, selecting the IrDA clock, and toggling between the standard serial function and IrDA function for the serial input/output pins. The IRCR register is set through 8-bit memory manipulation instructions. After a reset signal is generated, the value of this register is set to “00H”.

Figure 15-3 Format of IrDA control register (IRCR)

Address: 40044000H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IRCR	IRE	IRCKS2	IRCKS1	IRCKS0	IRTXINV	IRRXINV	0	0

IRE	IrDA enable
0	Serial input/output pins are used for the standard serial function.
1	Serial input/output pins are used for the IrDA function.

IRCKS2	IRCKS1	IRCKS0	IrDA clock selection
0	0	0	B3/16 (B=bit rate)
0	0	1	$f_{CLK}/2$
0	1	0	$f_{CLK}/4$
0	1	1	$f_{CLK}/8$
1	0	0	$f_{CLK}/16$
1	0	1	$f_{CLK}/32$
1	1	0	$f_{CLK}/64$
1	1	1	Disable settings

IRTXINV	IrTxD data polarity inversion
0	Output the transmitted data as IrTxD.
1	Invert the transmitted data before outputting it as IrTxD.

IRRXINV	IrRxD data polarity inversion
0	Use the input data from the IrRxD pin as the received data.
1	Use the inverted input data from the IrRxD pin as the received data.

Cautions 1. Bit1 and bit0 must be set to “0”.

2. The IRCKS[2:0] bits, IRTXINV bit, and IRRXINV bit can only be set when the IRE bit is “0”.

15.3 IrDA Operation

15.3.1 IrDA Communication Operation Steps

(1) Initial setup process for IrDA communication

Follow these steps to initialize IrDA communication:

1. Set the IRDAEN bit in the PER0 register to “1”.
2. Configure the IRCR register.
3. Set the relevant SCI registers (refer to the setup steps for UART mode).

(2) IrDA communication stop process

1. Set the IrTxD pin state after stopping IrDA communication by configuring the port register and port mode register.

Remark When resetting IrDA through step 3, the IrTxD pin might change its output state due to switching to the regular UART data output.

- Output a low level from the IrTxD pin
Set the port register to “0”. After this setting, the IrTxD pin will immediately be fixed to a low level.
 - Output a high level from the IrTxD pin
Set the port register to “1”. After this setting, the IrTxD pin will immediately be fixed to a high level after IrDA reset in step 3.
 - Set the IrTxD pin to Hi-Z state. Set the port mode register to “1”. After this setting, the IrTxD pin will immediately go into Hi-Z state.
2. Set the STm register (related SCI registers) STm0 and STm1 bits to “1” (stop operation for SCI channels 0 and 1).
 3. Set the IRDAEN bit in the PER0 register to “0” to reset IrDA.

The STm register’s STm0 and STm1 bits, or the IRE bit for IrDA, cannot be set to 1 or 0, respectively, except in the above steps.

(3) Steps for handling IrDA frame errors

In case of a frame error during IrDA communication, the following steps must be taken to return to a state where subsequent data can be received:

1. Set the STm1 bit in the SCI’s STm register to 1 (stop operation for SCI channel 1).
2. Set the SSm1 bit in the SCI’s SSm register to 1 (start operation for SCI channel 1).

Remark m: Unit number (m=0)

For SCI frame error handling, refer to Chapter 13 Universal Serial Communication Unit.

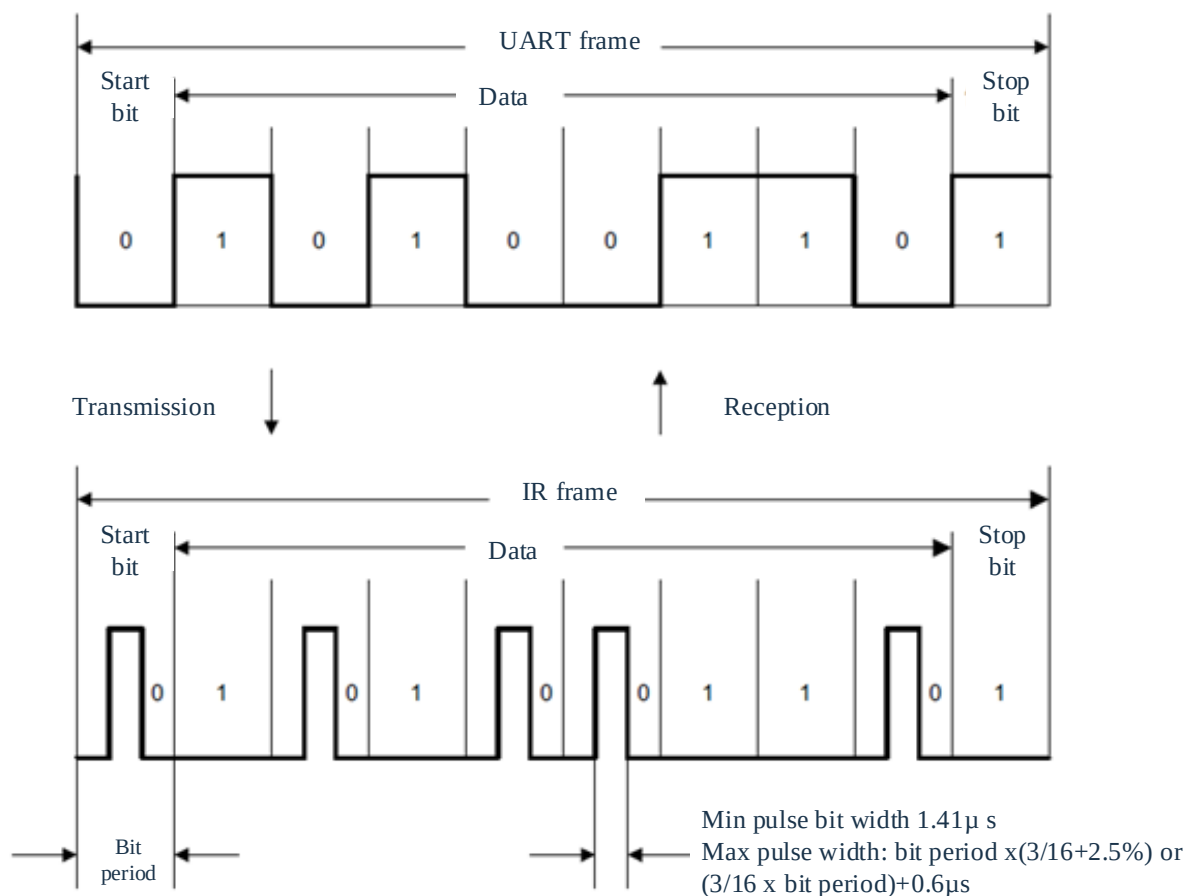
15.3.2 Transmission

During transmission, the output signal from the SCI (UART frame) is converted into an IR frame via IrDA (refer to Figure 15-4).

When the IRTXINV bit is 0 and the serial data is 0, a high-level pulse is output during 3/16 of the bit period (initial value). Additionally, the width of the high-level pulse can be adjusted based on the settings of the IRCKS2 to IRCKS0 bits. According to the standard, the minimum high-level pulse width is 1.41 μ s, and the maximum is $(3/16 + 2.5\%) \times \text{bit period}$, or $(3/16 \times \text{bit period}) + 0.6 \mu$ s.

When the CPU or peripheral hardware clock (f_{CLK}) is 24 MHz, the minimum high-level pulse width that can be set is 1.5 μ s, which satisfies the condition that the high-level pulse width should be no less than 1.41 μ s. Also, when the serial data is 1, no pulse is output.

Figure 15-4 IrDA transmit/receive timing diagram



15.3.3 Reception

During reception, the IR frame data is converted into a UART frame via IrDA and then input to the SCI. When the IRRXINV bit is 0 and a high-level pulse is detected, a low-level data bit is output. If no pulse is detected within the 1-bit period, a high-level data bit is output. It is important to note that pulses smaller than the minimum pulse width of 1.41 μ s cannot be recognized.

15.3.4 Selection of High-Level Pulse Width

If the pulse width during transmission is smaller than the bit rate $\times 3/16$, the settings of IRCKS2~IRCKS0 bits (minimum pulse width) and the corresponding high-level pulse width during setup are shown in Table 15-2.

Table 15-2 IRCKS2~IRCKS0 bit settings

f _{CLK} [MHz]	Item	<Upper> Bit rate [kbps] <Lower> Bit rate $\times 3/16$ [μ s]					
		2.4	9.6	19.2	38.4	57.6	115.2
		78.13	19.53	9.77	4.87	3.26	1.63
1	IRCKS2~IRCKS0	001	001	001	_Note 1	_Note 1	_Note 1
	High-level pulse width[μ s]	2.00	2.00	2.00	_Note 1	_Note 1	_Note 1
2	IRCKS2~IRCKS0	010	010	010	010	010	_Note 1
	High-level pulse width[μ s]	2.00	2.00	2.00	2.00	2.00	_Note 1
3	IRCKS2~IRCKS0	011	011	011	011	011	_Note 1
	High-level pulse width[μ s]	2.67	2.67	2.67	2.67	2.67	_Note 1
4	IRCKS2~IRCKS0	011	011	011	011	011	000 ^{Note 2}
	High-level pulse width[μ s]	2.00	2.00	2.00	2.00	2.00	1.50
6	IRCKS2~IRCKS0	100	100	100	100	100	000 ^{Note 2}
	High-level pulse width[μ s]	2.67	2.67	2.67	2.67	2.67	1.50
8	IRCKS2~IRCKS0	100	100	100	100	100	000 ^{Note 2}
	High-level pulse width[μ s]	2.00	2.00	2.00	2.00	2.00	1.50
12	IRCKS2~IRCKS0	101	101	101	101	101	000 ^{Note 2}
	High-level pulse width[μ s]	2.67	2.67	2.67	2.67	2.67	1.50
16	IRCKS2~IRCKS0	101	101	101	101	101	000 ^{Note 2}
	High-level pulse width[μ s]	2.00	2.00	2.00	2.00	2.00	1.50
24	IRCKS2~IRCKS0	110	110	110	110	110	000 ^{Note 2}
	High-level pulse width [μ s]	2.67	2.67	2.67	2.67	2.67	1.50

Notes: 1. “-” indicates that the communication standard is not met.

2. The pulse width cannot be smaller than bit rate $\times 3/16$.

15.4 Notes on Using IrDA

1. The peripheral enable register can be used to allow or prohibit the IrDA runtime clock supply. By default, the clock supply is disabled, so the registers cannot be accessed. Before setting the registers, the peripheral enable register must be configured to allow the IrDA runtime clock.
2. The IrDA functionality continues to operate in sleep mode.
3. During IrDA communication, the SCI initialization function (SS bit = 1) must not be used.
4. The IRCR register's IRRXINV bit, IRTXINV bit, and IRCKS[2:0] bits can only be set when the IRE bit is 0.

Chapter 16 Enhanced DMA

16.1 Functions of DMA

DMA is a function that performs data transfer between memory units without using the CPU. DMA is initiated through peripheral interrupts to carry out data transfers. When both DMA and the CPU access the same unit in Flash, SRAM0, SRAM1, or a peripheral module, DMA has higher bus access priority over the CPU. When DMA and the CPU access different units in Flash, SRAM0, SRAM1, or peripheral modules, they do not interfere with each other and can execute in parallel.

The specifications of DMA are shown in Table 16-1.

Table 16-1 DMA specifications (1/2)

Item		Specification
Boot source		Up to 24 boot sources
Assignable control data		24 groups
The address space that can be transmitted	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 16-5.

Table 16-1 DMA specifications (2/2)

Item		Specification
Interrupt request	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 mode	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 DMAENi0~DMAENi7 bits of the DMAENi register is 1 (boot allowed), the transfer of data begins each time the DMA boot source occurs.
Transfer stop	Normal mode	- Set DMAENi0 to DMAENi7 to 0(disable startup). - At the end of the data transfer when the DMACTj register changes from 1 to 0.
	Repeat mode	- Set DMAENi0~DMAENi7 bits 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.

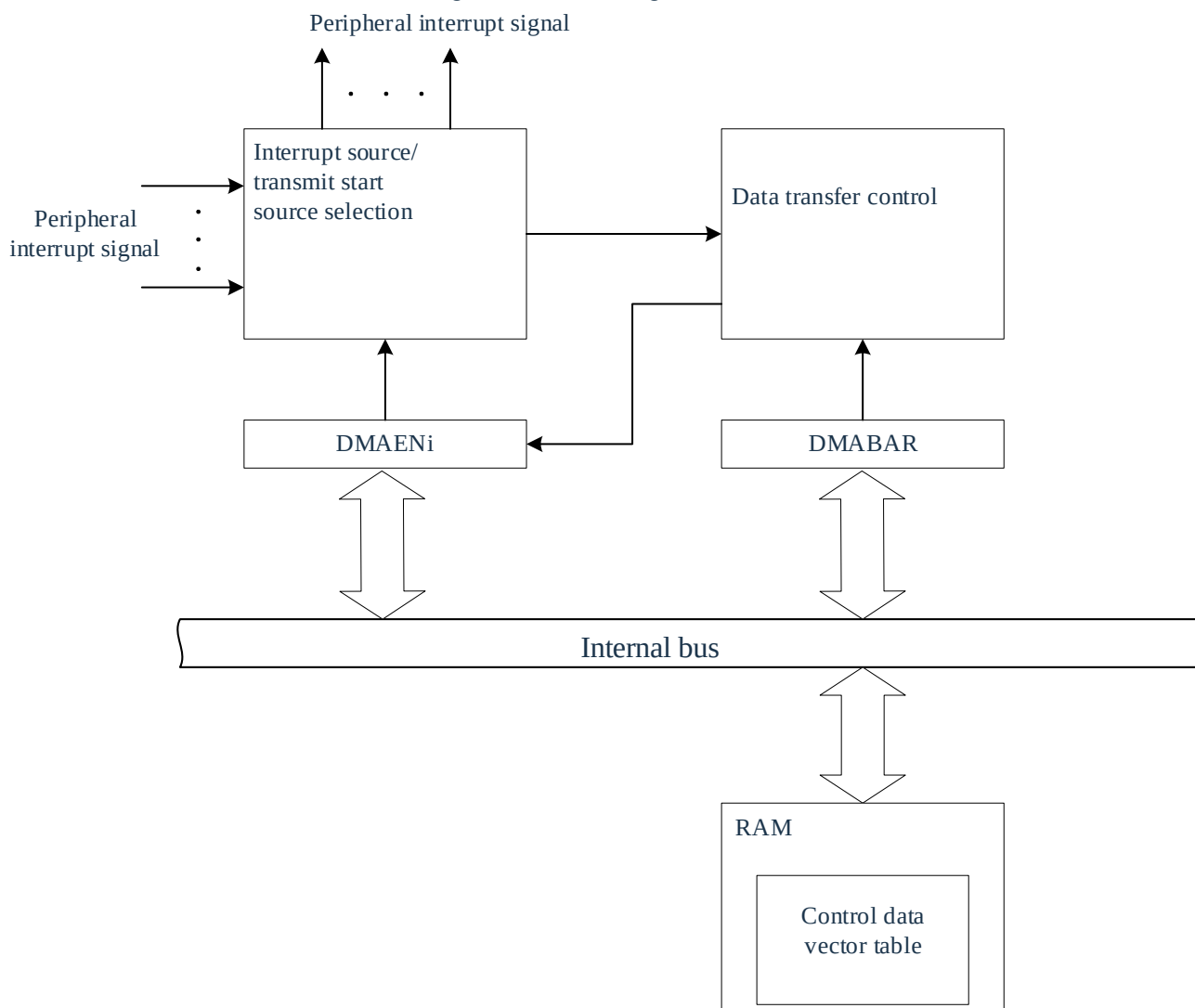
Note: In deep sleep mode, the flash memory cannot be used as a DMA transfer source because it is stopped.

Remark: i=0~2, j=0~23

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 Registers for Controlling DMA

The registers that control the DMA are shown in Table 16-2.

Table 16-2 Registers for controlling DMA

Register name	Symbol
Peripheral enable register 1	PER1
DMA start enable register 0	DMAEN0
DMA start enable register 1	DMAEN1
DMA start enable register 2	DMAEN2
DMA base address register	DMABAR

The control data of the DMA is shown in Table 16-3.

The control data for DMA is allocated in the DMA control data area of RAM. The DMA control data area and the 416-byte area containing the DMA vector table area (the starting address where the control data is stored) are set via the DMABAR register.

Table 16-3 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

Remark j=0~23

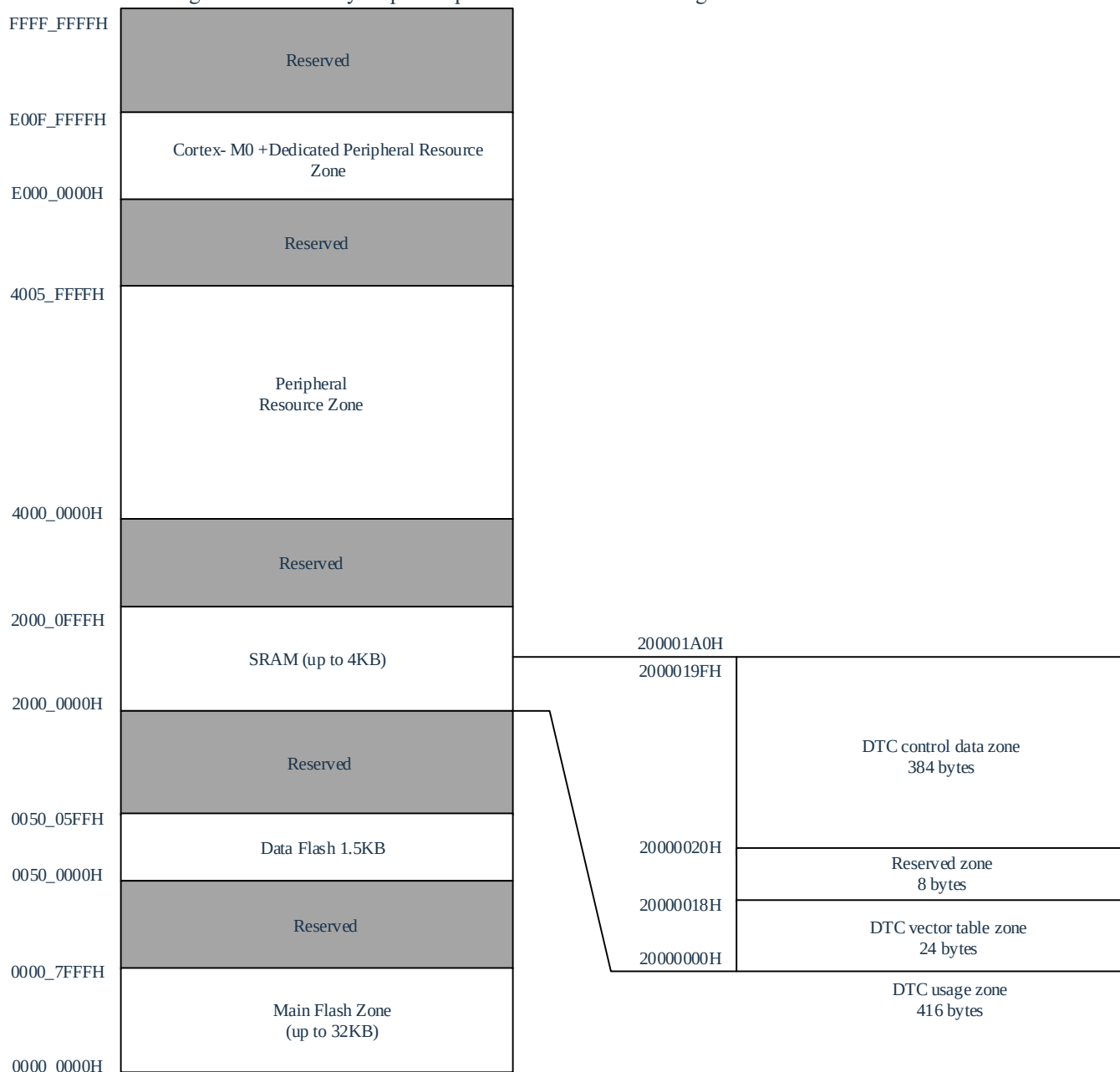
16.3.1 Allocation of DMA Control Data Zone and DMA Vector Table Zone

The DMABAR register allocates the control data for DMA and the 416-byte vector table to the RAM area.

When the value of the DMABAR register is set to “20000000H”, the memory map example is shown in Figure 16-2.

In the 384-byte DMA control data zone, the space not used by DMA can be used as RAM.

Figure 16-2 Memory map example when the DMABAR register is set to “20000000H”



16.3.2 Control Data Allocation

Starting from the initial address, the control data is allocated according to the sequence of the registers DMACR_j, DMBLS_j, DMACT_j, DMRLD_j, DMSAR_j, DMDAR_j (where $j = 0 \sim 23$).

The starting address is set by the DMABAR register, and the lower 10 bits are allocated by the vector tables set by each trigger source.

The control data allocation is shown in Figure 16-3.

- Cautions
1. The corresponding DMAEN_i ($i = 0 \sim 2$) DMAEN_{i0}~DMAEN_{i7} bits must be set to 0 (disable the trigger) when modifying the DMACR_j, DMBLS_j, DMACT_j, DMRLD_j, DMSAR_j, and DMDAR_j registers.
 2. It is not allowed to access DMACR_j, DMBLS_j, DMACT_j, DMRLD_j, DMSAR_j, and DMDAR_j through DMA transfer.

Figure16-3 Control data allocation (DMABAR set to 2000000H)

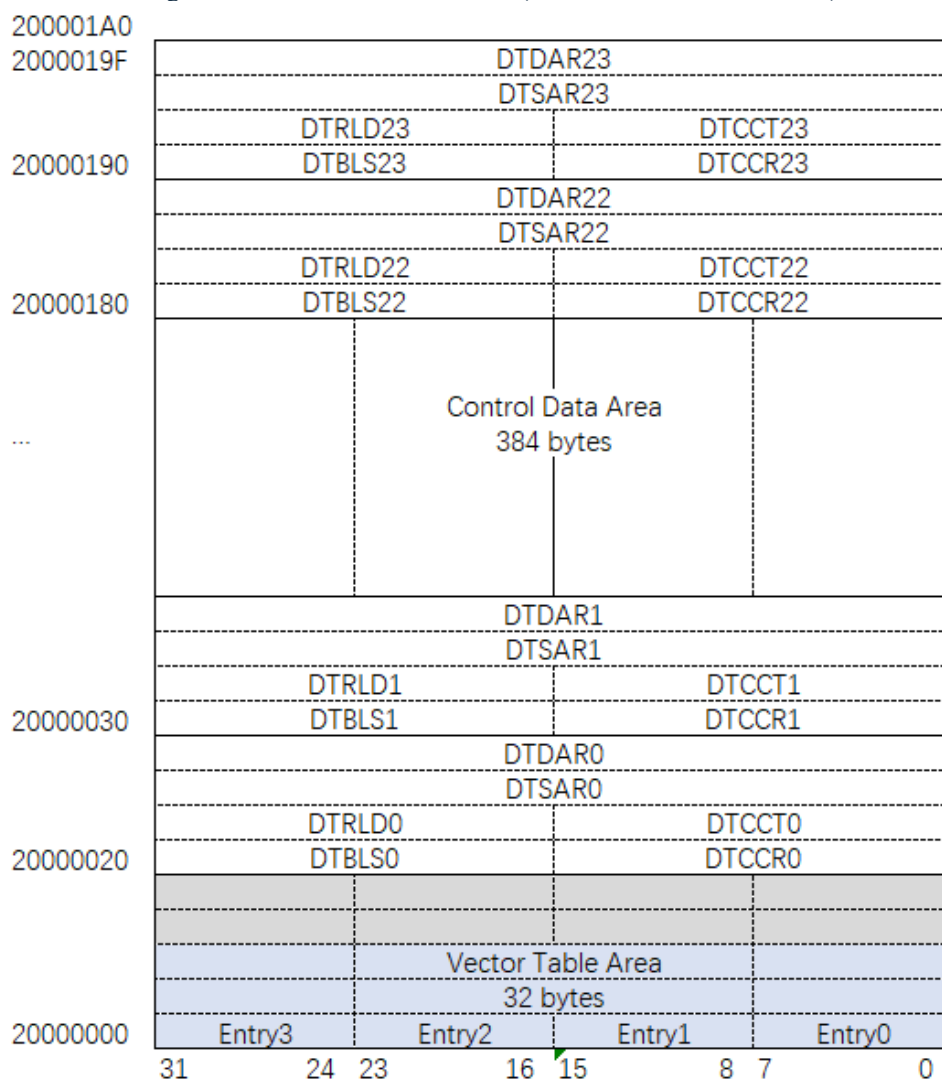


Table 16-4 Starting addresses for control data

j	Address	j	Address
11	baseaddr+D0H	23	baseaddr+190H
10	baseaddr+C0H	22	baseaddr+180H
9	baseaddr+B0H	21	baseaddr+170H
8	baseaddr+A0H	20	baseaddr+160H
7	baseaddr+90H	19	baseaddr+150H
6	baseaddr+80H	18	baseaddr+140H
5	baseaddr+70H	17	baseaddr+130H
4	baseaddr+60H	16	baseaddr+120H
3	baseaddr+50H	15	baseaddr+110H
2	baseaddr+40H	14	baseaddr+100H
1	baseaddr+30H	13	baseaddr+F0H
0	baseaddr+20H	12	baseaddr+E0H

Remark: baseaddr: The value set in the DMABAR register.

16.3.3 Vector Table

Once DMA is initiated, it determines the control data by reading from the vector table allocated from each start source. The control data is read from the DMA control data area.

The DMA start source and vector addresses are shown in Table 16-5. Each start source has a vector table of 1 byte, containing values from “00H” to “17H”. These values are used to select one set of control data from the 24 sets. The high 22 bits of the vector address are set by the DMABAR register, while the low 10 bits are allocated to the corresponding start source’s “00H” to “17H”.

Caution: The DMAENi (i = 0~2) register’s DMAENi0~DMAENi7 bits must be 0 (to disable the start) when modifying the start address of the DMA control data area in the vector table.

Figure 16-4 Starting addresses for control data and vector table
Example with DMABAR register value set to “2000000H”

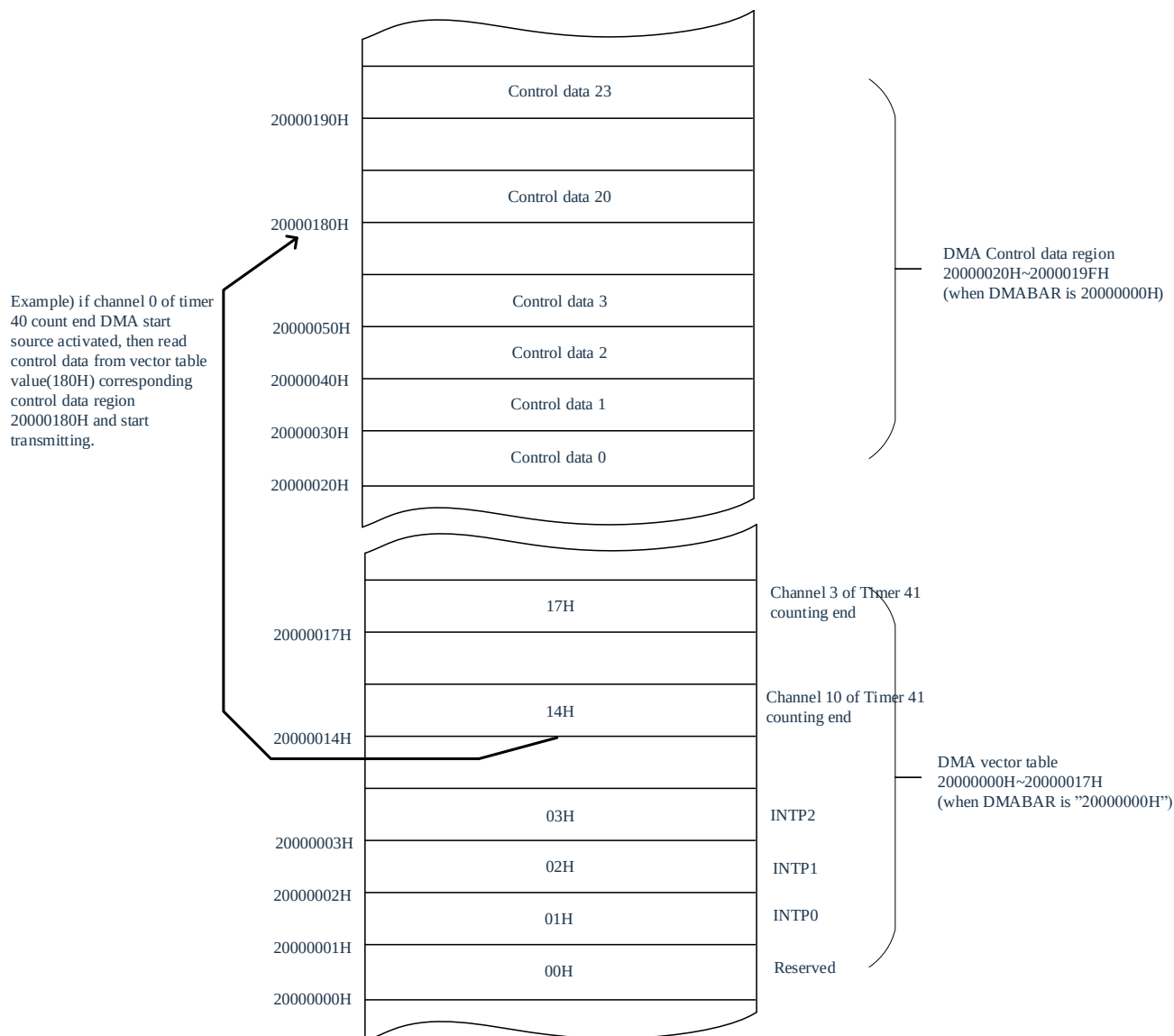


Table 16-5 DMA start source and vector address

DMA start source (interrupt request resource)	No.	Vector address	Priority
Flash read/write erase end	0	DMABAR register set address+00H	High ↑ ↓ Low
INTP0	1	DMABAR register set address+01H	
INTP1	2	DMABAR register set address+02H	
INTP2	3	DMABAR register set address+03H	
INTP3	4	DMABAR register set address+04H	
A/D conversion end	5	DMABAR register set address+05H	
Comparatoe detect 0	6	DMABAR register set address+06H	
Comparatoe detect 1	7	DMABAR register set address+07H	
UART0 receive transfer end/SSPI01 transfer end or buffer empty /IIC01 transfer end	8	DMABAR register set address+08H	
UART0 transmit transfer end/SSPI00 transfer end or buffer empty /IIC00 transfer end	9	DMABAR register set address+09H	
UART1 receive transfer end/SSPI11 transfer end or buffer empty /IIC11 transfer end	10	DMABAR register set address+0AH	
UART1 transmit transfer end/SSPI10 transfer end or buffer empty /IIC10 transfer end	11	DMABAR register set address+0BH	
UART2 receive transfer end/SSPI21 transfer end or buffer empty /IIC21 transfer end	12	DMABAR register set address+0CH	
UART2 transmit transfer end/SSPI20 transfer end or buffer empty /IIC20 transfer end	13	DMABAR register set address+0DH	
IICA0 communication end	14	DMABAR register set address+0EH	
A 15 -bit interval timer generates a count interrupt	15	DMABAR register set address+0FH	
Timer40 channel 0 count/capture end	16	DMABAR register set address+10H	
Timer40 channel 1 count/capture end	17	DMABAR register set address+11H	
Timer40 channel 2 count/capture end	18	DMABAR register set address+12H	
Timer40 channel 3 count/capture end	19	DMABAR register set address+13H	
Timer41 channel 0 count/capture end	20	DMABAR register set address+14H	
Timer41 channel 1 count/capture end	21	DMABAR register set address+15H	
Timer41 channel 2 count/capture end	22	DMABAR register set address+16H	
Timer41 channel 3 count/capture end	23	DMABAR register set address+17H	

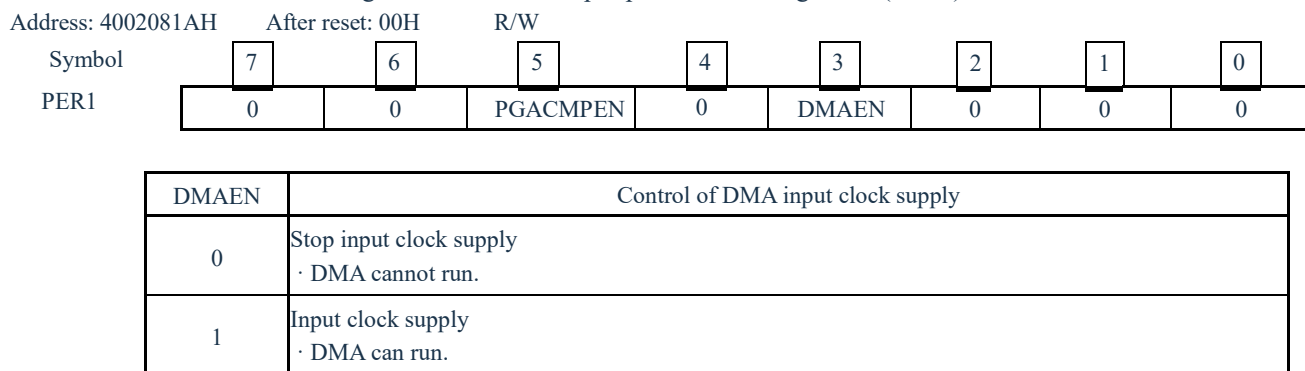
16.3.4 Peripheral enable register 1 (PER1)

The PER1 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 PER1 register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register changes to “00H”.

Figure 16-5 Format of peripheral enable register 1 (PER1)



16.3.5 DMA Control Register j (DMACRj) (j=0~23)

The DMACRj register controls the mode of operation of the DMA.

Figure 16-6 Format of DMA control register j (DMACRj)

Address: refer to 16.3.2

After reset: undefine R/W

Symbol:	15	14	13	12	11	10	9	8
DMACRj	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	SZ		RPTINT	CHNE	DAMOD	SAMOD	RPTSEL	MODE

SZ	Selection of transfer data length
00	8-bit
01	16-bit
10	32-bit
11	Settings are disabled

RPTINT	Enable/disable repeat mode interrupt
0	Interrupts are disabled
1	Interrupts are enabled
When the MODE bit is 0 (normal mode), the setting of the RPTINT bit is invalid.	

CHNE	Chain transfer enabled/disabled
0	Chain transfer is disabled.
1	Chain transfer is enabled.
The CHNE bit of the DMACR23 register must be set to 0 (chain transfer disabled).	

DAMOD	Control of the transfer target address
0	Fixed
1	Incremental
The setting of the DAMOD bit is invalid when the MODE bit is 1 (Repeat mode) and the RPTSEL bit is 0 (the transfer target is a repeat area).	

SAMOD	Control of transfer source address
0	Fixed
1	Incremental
The setting of the SAMOD bit is invalid when the MODE bit is 1 (Repeat mode) and the RPTSEL bit is 1 (the transfer source is a repeat area).	

RPTSEL	Selection of repeat area
0	The transfer target is a repeat area.
1	The transfer source is a repeat area.
The setting for the RPTSEL bit is not valid when the MODE bit is 0 (normal mode).	

MODE	Selection of transfer modes
0	Normal mode
1	Repeat mode

Caution: Access to the DMACRj register cannot be performed via DMA transfer.

16.3.6 DMA Block Size Register j (DMBLSj) (j=0~23)

This register sets the block size for 1-time startup transfer data.

Figure 16-7 Format of DMA block size register j(DMBLSj)

Address: refer to 16.3.2

After reset: undefine R/W

Symbol:	15	14	13	12	11	10	9	8
DMBLSj	DMBLSj15	DMBLSj14	DMBLSj13	DMBLSj12	DMBLSj11	DMBLSj10	DMBLSj9	DMBLSj8
	7	6	5	4	3	2	1	0
	DMBLSj7	DMBLSj6	DMBLSj5	DMBLSj4	DMBLSj3	DMBLSj2	DMBLSj1	DMBLSj0

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

Caution: Access to the DMBLSj register cannot be performed via DMA transfer.

16.3.7 DMA Transfer Count Register j (DMACTj) (j=0~23)

This register sets the number of DMA data transfers. It is decremented by 1 whenever 1 DMA transfer is initiated.

Figure 16-8 Format of DMA transfer count register j (DMACTj)

Symbol:	15	14	13	12	11	10	9	8
DMACTj	DMACTj15	DMACTj14	DMACTj13	DMACTj12	DMACTj11	DMACTj10	DMACTj9	DMACTj8
	7	6	5	4	3	2	1	0
	DMACTj7	DMACTj6	DMACTj5	DMACTj4	DMACTj3	DMACTj2	DMACTj1	DMACTj0

Address: refer to 16.3.2

After reset: undefine R/W

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

Caution: Access to the DMACTj register cannot be performed via DMA transfer.

16.3.8 DMA Transfer Count Reload Register j (DMRLDj) (j=0~23)

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.

Figure 16-9 Format of DMA transfer count reload register j(DMRLDj)

Address: refer to 16.3.2	After reset: undefine							R/W
Symbol:	15	14	13	12	11	10	9	8
DMRLDj	DMRLDj15	DMRLDj14	DMRLDj13	DMRLDj12	DMRLDj11	DMRLDj10	DMRLDj9	DMRLDj8
	7	6	5	4	3	2	1	0
	DMRLDj7	DMRLDj6	DMRLDj5	DMRLDj4	DMRLDj3	DMRLDj2	DMRLDj1	DMRLDj0

Caution: Access to the DMRLDj register cannot be performed via DMA transfer.

16.3.9 DMA Source Address Register j (DMSARj) (j=0~23)

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.

Figure 16-10 Format of DMA source address register j (DMSARj)

Address: refer to 16.3.2		After reset: undefine							R/W
Symbol	31	30	29	28	27	26	25	24	
DMSARj	DMSARj3	DMSARj3	DMSARj2	DMSARj2	DMSARj2	DMSARj2	DMSARj2	DMSARj2	
	1	0	9	8	7	6	5	4	
	23	22	21	20	19	18	17	16	
	DMSARj2	DMSARj2	DMSARj2	DMSARj2	DMSARj1	DMSARj1	DMSARj1	DMSARj1	
	3	2	1	0	9	8	7	6	
	15	14	13	12	11	10	9	8	
	DMSARj1	DMSARj1	DMSARj1	DMSARj1	DMSARj1	DMSARj1	DMSARj9	DMSARj8	
	5	4	3	2	1	0			
	7	6	5	4	3	2	1	0	
	DMSARj7	DMSARj6	DMSARj5	DMSARj4	DMSARj3	DMSARj2	DMSARj1	DMSARj0	

Caution: Access to the DMSARj register cannot be performed via DMA transfer.

16.3.10 DMA Target Address Register j (DMDARj) (j=0~23)

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.

Figure 16-11 Format of DMA target address register j (DMDARj)

Address: refer to 16.3.2		After reset: undefine R/W						
Symbol	31	30	29	28	27	26	25	24
DMDARj	DMDARj3	DMDARj3	DMDARj2	DMDARj2	DMDARj2	DMDARj2	DMDARj2	DMDARj2
	1	0	9	8	7	6	5	4
	23		22		21		20	
	19		18		17		16	
	DMDARj2	DMDARj2	DMDARj2	DMDARj2	DMDARj1	DMDARj1	DMDARj1	DMDARj1
	3	2	1	0	9	8	7	6
	15		14		13		12	
	11		10		9		8	
	DMDARj1	DMDARj1	DMDARj1	DMDARj1	DMDARj1	DMDARj1	DMDARj9	DMDARj8
	5	4	3	2	1	0		
7		6		5		4		
3		2		1		0		
DMDARj7	DMDARj6	DMDARj5	DMDARj4	DMDARj3	DMDARj2	DMDARj1	DMDARj0	

Caution: Access to the DMDARj register cannot be made via DMA transfer.

16.3.11 DMA Start Enable Register i (DMAENi) (i=0~2)

This is an 8-bit register that controls whether or not starting DMA through each interrupt source. The correspondence between the interrupt sources and the DMAENi0 to DMAENi7 bits is shown in Table 16-6.

The DMAENi register can be set by an 8-bit memory manipulation instruction.

Cautions:

1. The DMAENi0 to DMAENi7 bits must be changed at a location that does not generate a startup source corresponding to this bit.
2. Access to the DMAENi register cannot be performed via DMA transfer.
3. The assigned function varies from product to product and must be set to 0 if no function is assigned.

Figure 16-12 Format of DMA start enable register i(DMAENi) (i=0~2)

Address: 40005000H (DMAEN0), 40005001H (DMAEN1),

40005002H (DMAEN2)

After reset: 00H

R/W

Symbol	7	6	5	4	3	2	1	0
DMAENi	DMAENj7	DMAENj6	DMAENj5	DMAENj4	DMAENj3	DMAENj2	DMAENj1	DMAENj0

DMAENi7	DMA start enable i7
0	Start disabled
1	Start enabled
The DMAENi7 bit changes to 0 (start disabled) according to the condition of the transfer end interrupt.	

DMAENi6	DMA start enable i6
0	Start disabled
1	Start enabled
The DMAENi6 bit changes to 0 (start disabled) according to the condition of the transfer end interrupt.	

DMAENi5	DMA start enable i5
0	Start disabled
1	Start enabled
The DMAENi5 bit changes to 0 (start disabled) according to the condition of the transfer end interrupt.	

DMAENi4	DMA start enable i4
0	Start disabled
1	Start enabled
The DMAENi4 bit changes to 0 (start disabled) according to the condition of the transfer end interrupt.	

DMAENi3	DMA start enable i3
0	Start disabled
1	Start enabled
The DMAENi3 bit changes to 0 (start disabled) according to the condition of the transfer end interrupt.	

DMAENi2	DMA start enable i2
0	Start disabled
1	Start enabled
The DMAENi2 bit changes to 0 (start disabled) according to the condition of the transfer end interrupt.	

DMAENi1	DMA start enable i1
0	Start disabled
1	Start enabled
The DMAENi1 bit changes to 0 (start disabled) according to the condition of the transfer end interrupt.	

DMAENi0	DMA start enable i0
0	Start disabled
1	Start enabled
The DMAENi0 bit changes to 0 (start disabled) according to the condition of the transfer end interrupt.	

Table 16-6 Correspondence of interrupt sources and DMAENi0~DMAENi7 bits

Register	DMAENi7 bit	DMAENi6 bit	DMAENi5 bit	DMAENi4 bit	DMAENi3 bit	DMAENi2 bit	DMAENi1 bit	DMAENi0 bit
DMAEN0	Comparator detect 1	Comparator detect 0	A/D conversion end	INTP3	INTP2	INTP1	INTP0	Flash erase/write end
DMAEN1	15-bit interval timer interrupt	IICA0 communication end	UART2 transmit transfer end/SSPI20 transfer end or buffer empty /IIC20 transfer end	UART2 receive transfer end/SSPI21 transfer end or buffer empty /IIC21 transfer end	UART1 transmit transfer end/SSPI10 transfer end or buffer empty /IIC10 transfer end	UART1 receive transfer end/SSPI11 transfer end or buffer empty /IIC11 transfer end	UART0 transmit transfer end/SSPI00 transfer end or buffer empty /IIC00 transfer end	UART0 receive transfer end/SSPI01 transfer end or buffer empty/IIC01 transfer end
DMAEN2	Timer array unit 1 channel 3 count end or capture end	Timer array unit 1 channel 2 count end or capture end	Timer array unit 1 channel 1 count end or capture end	Timer array unit 1 channel 0 count end or capture end	Timer array unit 0 channel 3 count end or capture end	Timer array unit 0 channel 2 count end or capture end	Timer array unit 0 channel 1 count end or capture end	Timer array unit 0 channel 0 count end or capture end

Caution: Bits that are not assigned a function must be set to 0.

Remark: i=0~2

16.3.12 DMAENi Set Register (DMSETi)

This is the DMA start enable register, DMAENi. Setting the corresponding bit to 1 allows the respective bit of DMAENi to be set to 1.

Figure 16-13 Format of DMAENi set register (DMSETi) (i=0~2)

Address: 40005018H (DMSET0), 40005019H (DMSET1),
4000501AH (DMSET2)

					After reset: 00H		R/W	
Symbol	7	6	5	4	3	2	1	0
DMSETi	DMSETi7	DMSETi6	DMSETi5	DMSETi4	DMSETi3	DMSETi2	DMSETi1	DMSETi0

DMSETin	DMAENin set register
0	No set operation occurs.
1	Set the bit n of DMCENi to 1.

Note: i=0~2, n=0~7

16.3.13 DMAENi Reset Register (DMCLRi)

This is the reset register for the DMA start enable register, DMAENi. Setting the corresponding bit to 1 will reset the respective bit of DMAENi to 0.

Figure 16-14 Format of DMAENi reset register (DMCLRi) (i=0~2)

Address: 40005020H (DMCLR0), 40005021H (DMCLR1),
40005022H (DMCLR2)

					After reset: 00H		R/W	
Symbol	7	6	5	4	3	2	1	0
DMCLRi	DMCLRi7	DMCLRi6	DMCLRi5	DMCLRi4	DMCLRi3	DMCLRi2	DMCLRi1	DMCLRi0

DMSETin	DMAENin reset register
0	No reset occurs.
1	Resets the bit n of DMCENi to 0.

Note: i=0~2, n=0~7

16.3.14 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 area.

Cautions:

1. The DMABAR register must be changed with all DMA startup sources set to disable startup.
2. The DMABAR register can be rewritten only 1 time.
3. Access to the DMABAR register cannot be performed via DMA transfer.
4. Refer to 16.3.1 for the DMA control data area and DMA vector table area assignment.
5. To set this register, keep it 512-byte aligned, that is, the lower 8 bits are set to zero. The DMA hardware ignores the lower 8 bits.
6. The register can only be accessed by WORD, BYTE and HALFWORD access are ignored.

Figure 16-15 Format of DMA base address register (DMABAR)

Address: 40005008H		After reset: 00000000H		R/W			
Symbol	31	30	29	28	27	26	25
DMABARj	DMABARj	DMABARj	DMABARj	DMABARj	DMABARj	DMABARj	DMABARj
	31	30	29	28	27	26	25
	23	22	21	20	19	18	17
	DMABARj	DMABARj	DMABARj	DMABARj	DMABARj	DMABARj	DMABARj
	23	22	21	20	19	18	17
	15	14	13	12	11	10	9
	DMABARj	DMABARj	DMABARj	DMABARj	DMABARj	DMABARj	DMABARj
	15	14	13	12	11	10	9
	7	6	5	4	3	2	1
	0	0	0	0	0	0	0

16.4 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 24 sets of control data to a DMA control data area and to perform transfer of 24 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=0~23) register is 1 (allowing chain transfer), the continuous data transfer (chain transfer) is read by a start source.

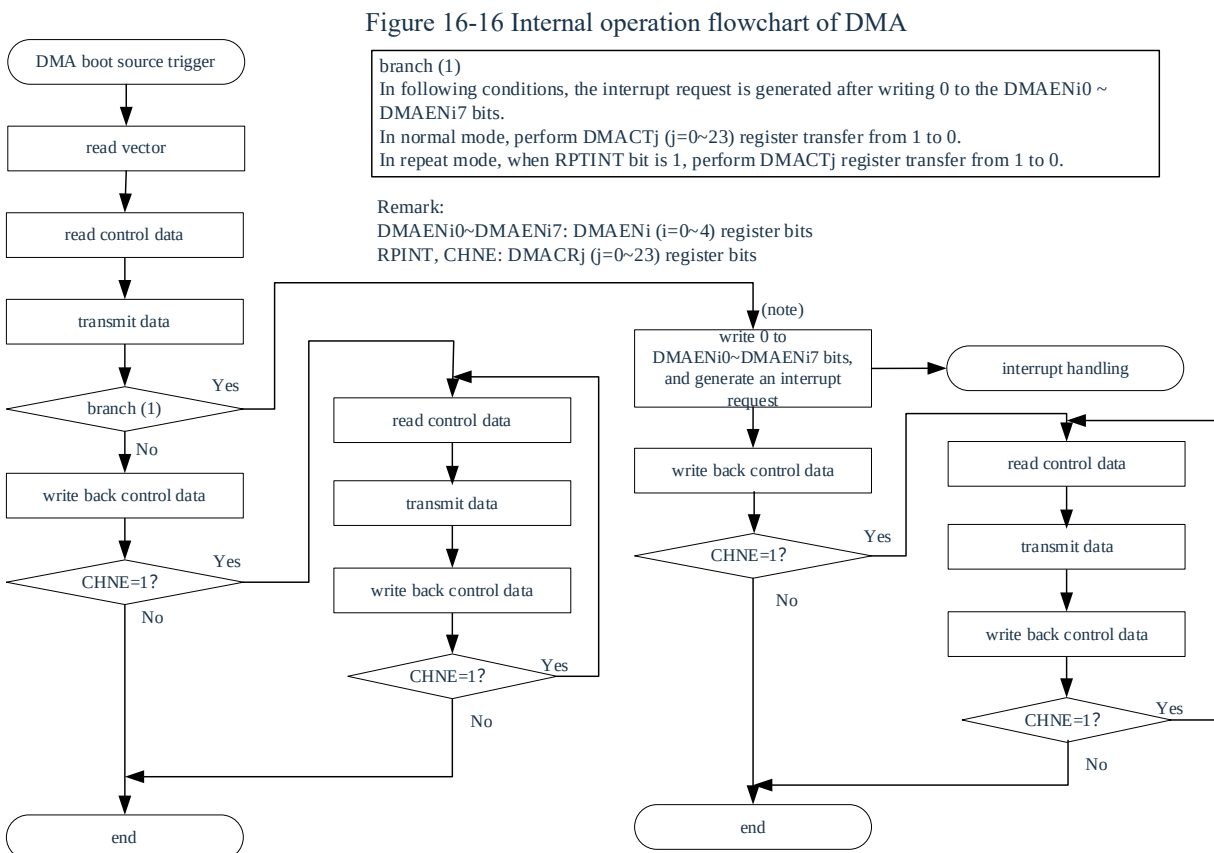
The transfer source address and the transfer target address are specified through the 32-bit DMSARj register and the 32-bit DMDARj register, respectively. After the data is transferred, the values of the DMSARj register and the DMDARj register are incremented or fixed according to the control data.

16.4.1 Boot Source

The DMA is started by a peripheral function interrupt signal and the interrupt signal for starting the DMA is selected by a DMAENi (i=0~2) register. When the data transfer (the initial transfer is continued during chain transfer) is set to the following two conditions, the corresponding DMAENi0-DMAENi7 bits of the DMAENi register are set to 0 (start disabled).

- In normal mode, the DMACTj (j=0~23) register is transferred to 0.
- In repeat mode, the RPTINT bit of the DMACRj register is 1 (allowing an interrupt to occur) and a transfer in which the DMACTj register becomes 0 is performed.

The internal operation flowchart of the DMA is shown in Figure 16-16.



Note: In a data transfer initiated by a setting enable chain transfer (CHNE=1), the DMAENi0~DMAENi7 bits are not written to 0 and no interrupt request is generated.

16.4.2 Normal Mode

At 8-bit transfer, the transfer data of 1 start-up is 1~65535 bytes. In 16-bit transfer, the transfer data of 1 start-up is 2~131070 bytes; At 32-bit transfer, the transfer data of 1 start-up is 4 to 262140 bytes. The number of transfers is 1 to 65535. If a data transfer in which the DMACTj (j=0~23) register becomes 0 is performed, an interrupt request corresponding to the start source is generated to the interrupt controller during DMA operation, and the DMAENi0~DMAENi7 bits of the corresponding DMAENi (i=0~2) register are set to 0 (startup disabled).

The register functions and data transfers for normal mode are shown in Table 16-7 and Figure 16-17, respectively.

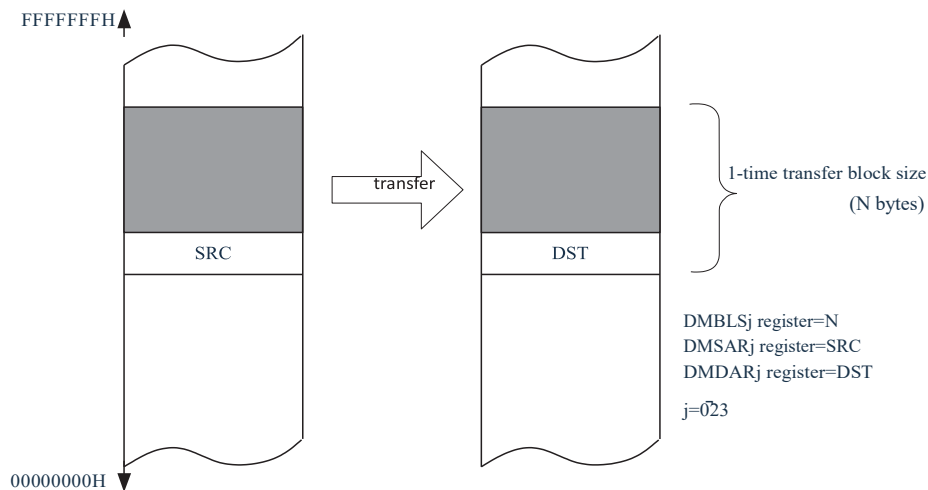
Table 16-7 Register functions in normal mode

Register name	Symbol	Function
DMA block size register j	DMBLSj	1-start-up 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: When a parity error reset (RPERDIS=0) is allowed to be generated by the RAM parity error detection function, it must be initialized (00H).

Remark: j=0~23

Figure 16-17 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

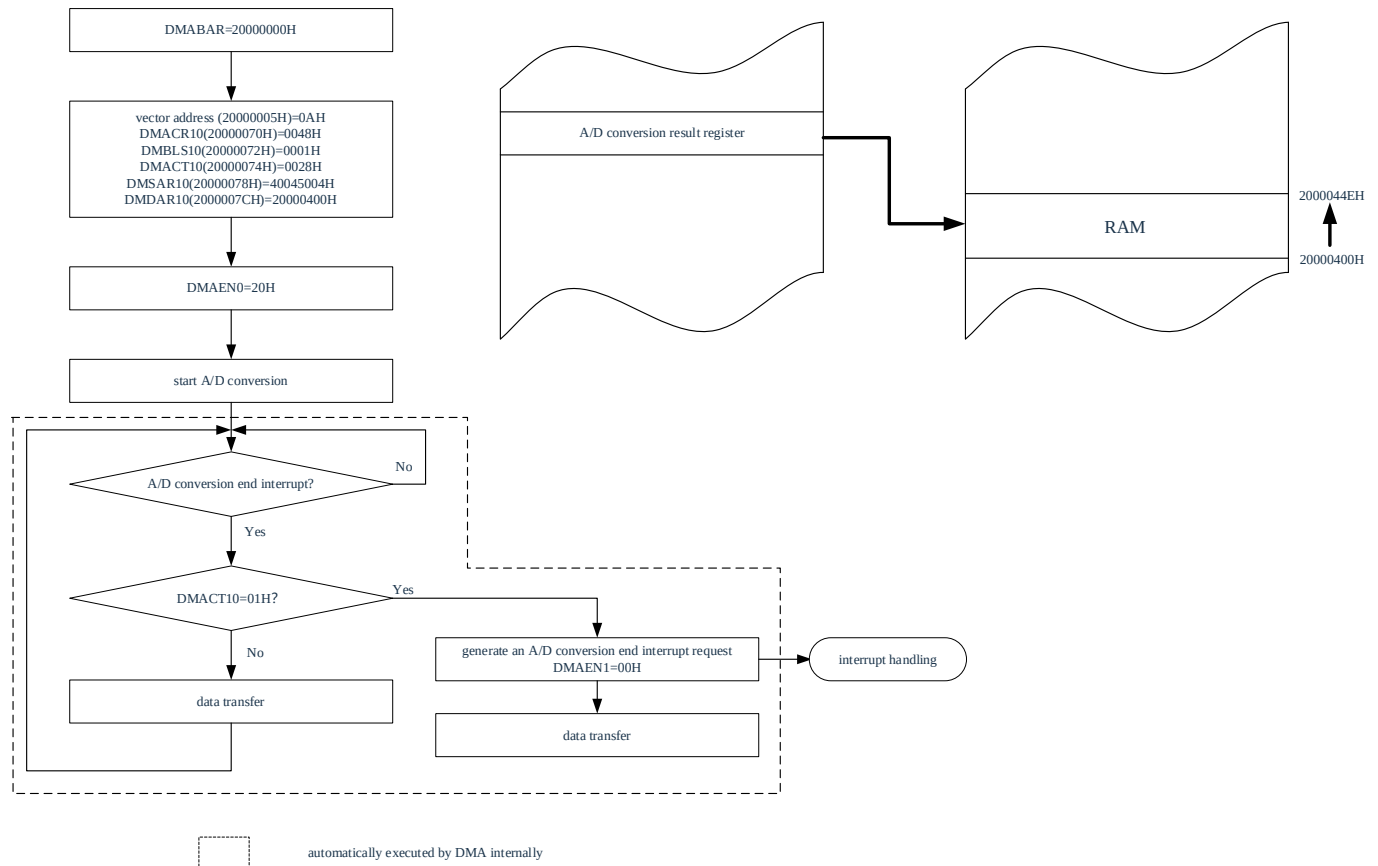
X: "0" or "1".

(1) Example 1 of normal mode usage: 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 20000005H and the control data is assigned at 20000070H~2000007FH.
- Transfer 2 bytes of data from A/D conversion result registers (40045004H, 40045005H) 40 times to 80 bytes of RAM 20000400H~2000044FH.

Figure 16-18 Example 1 of normal mode usage: Continuous A/D conversion results



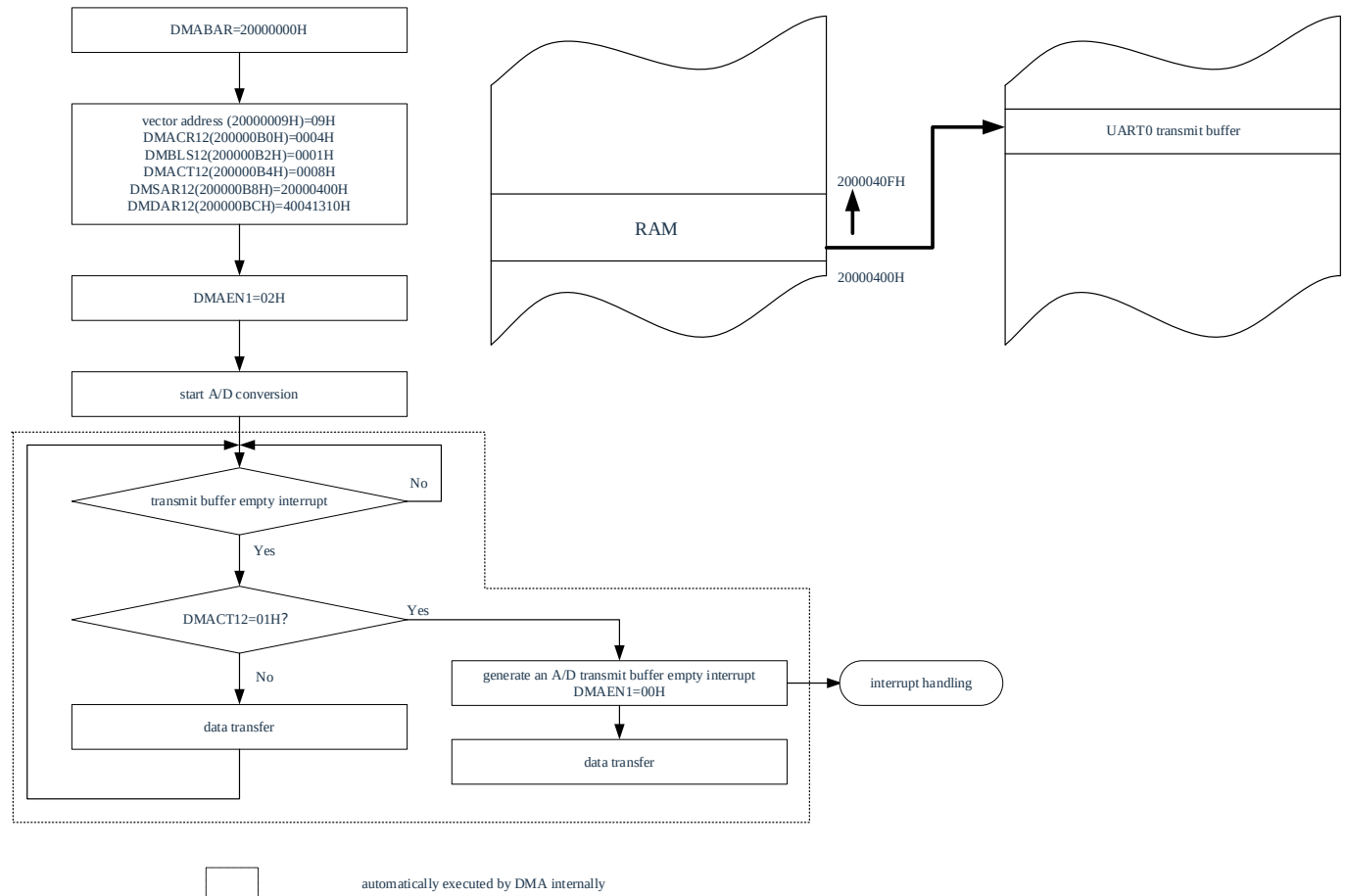
The value of the DMRLD10 register is not used because it is in normal mode. However, the DMRLD10 register must be initialized (0000H) when parity error reset (RPERDIS=0) is permitted via RAM parity.

(2) Example 2 of normal mode usage: UART0 continuous transmission

The DMA is initiated via the transmit buffer empty interrupt of the UART0 and the value of the RAM is transferred to the transmit buffer of the UART0.

- The vector address is assigned in 2000009CH, and the control data is assigned in 20000B00H~200001BFH.
- Transfer 8 bytes from 20000400H~20000407H of RAM to the UART0 transmit buffer (40041310H).

Figure 16-19 Example 2 of normal mode usage: UART0 continuous transmission



The value of the DMRLD12 register is not used because it is in normal mode. However, the DMRLD12 register must be initialized (0000H) when parity error reset (RPERDIS=0) is permitted via RAM parity.

The first UART0 transmission must be started via software. The DMA is started via a transmit buffer empty interrupt and then the second later transmission is automatically performed.

16.4.3 Repeat Mode

The transfer data for 1 startup 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 23) 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 DMAENi7 bits of the corresponding DMAENi (i=0 to 2) register are set to 0 (startup 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 DMAENi7 bits remain unchanged at 0.

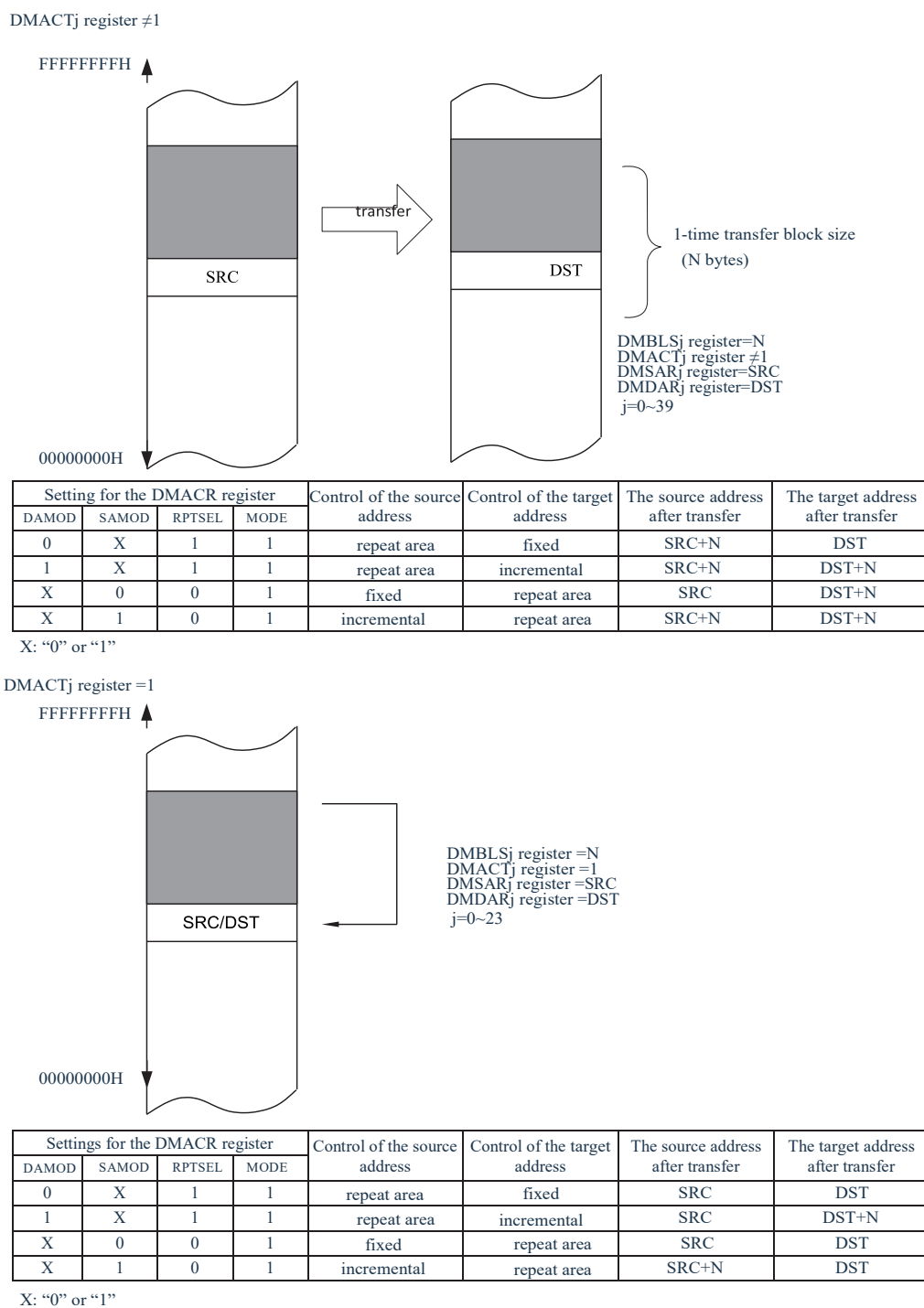
The register functions and data transfers for Repeat mode are shown in Table 16-8 and Figure 16-20, respectively.

Table 16-8 Register function 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 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	Data transfer source address
DMA target address register j	DMDARj	Data transfer target address

Remark: j=0~23

Figure 16-20 Repeat mode data transfer



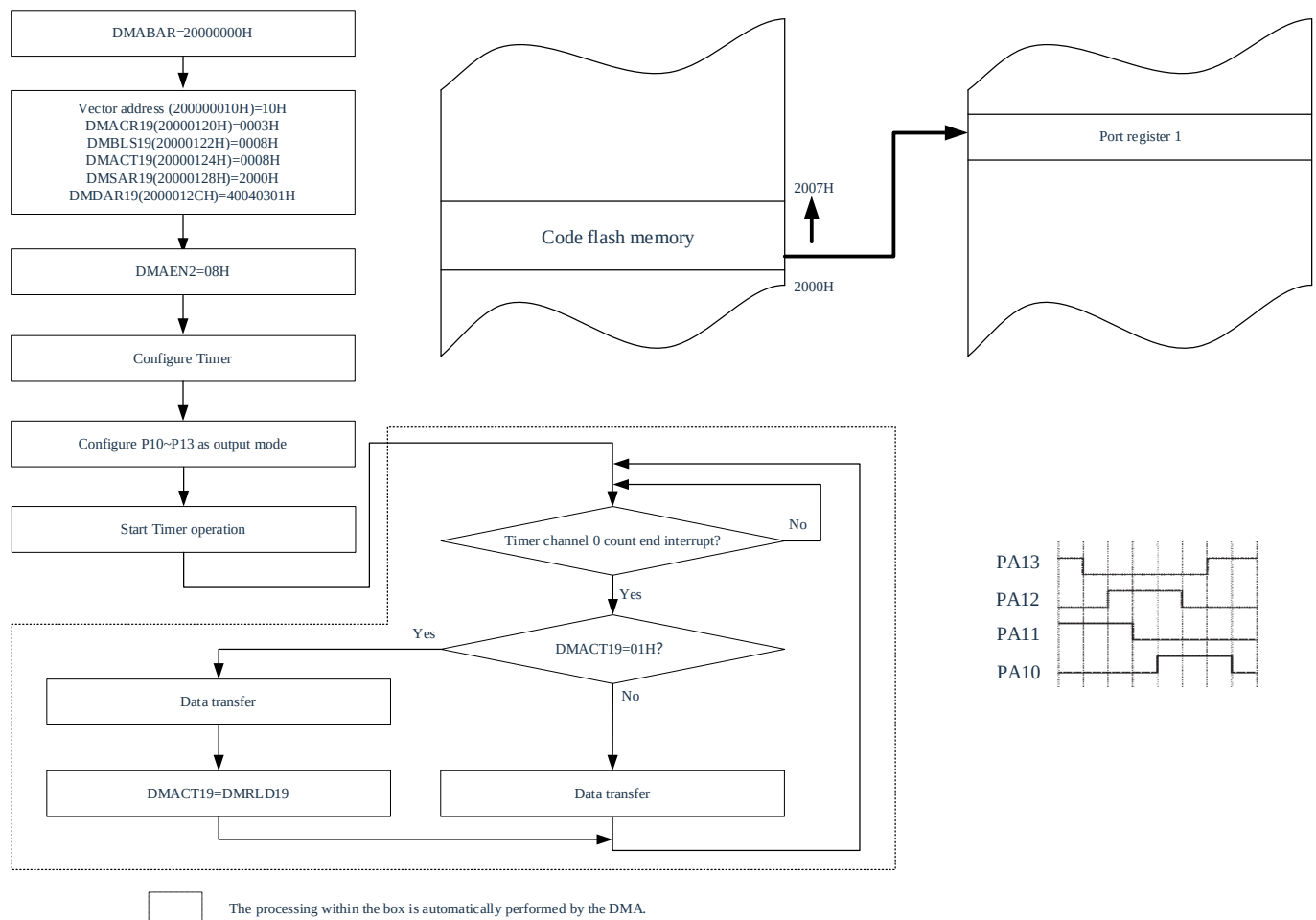
Caution: When using repeat mode, the data length of the repeat area must be set within 65535 bytes.

(1) Example 1 of repeat mode usage: Use the stepper motor of the port to control the pulse output

A channel 0 interval timer function of Timer4 is used to start DMA and transfer the mode of the motor control pulse saved in the code flash memory to the universal port.

- The vector address is assigned in 20000015H, and the control data is assigned at 20000190H~2000019FH.
- Transfer 8 bytes from 02000H to 2007H of the code flash to port register 1 (40040001H).
- Disable repeat mode interrupt.

Figure 16-21 Example 1 of repeat mode usage: use the stepper motor of the port to control the pulse output



To stop the output, the bit 0 of the `DMAEN2` must be cleared after stopping the timer.

16.4.4 Chain Transfer

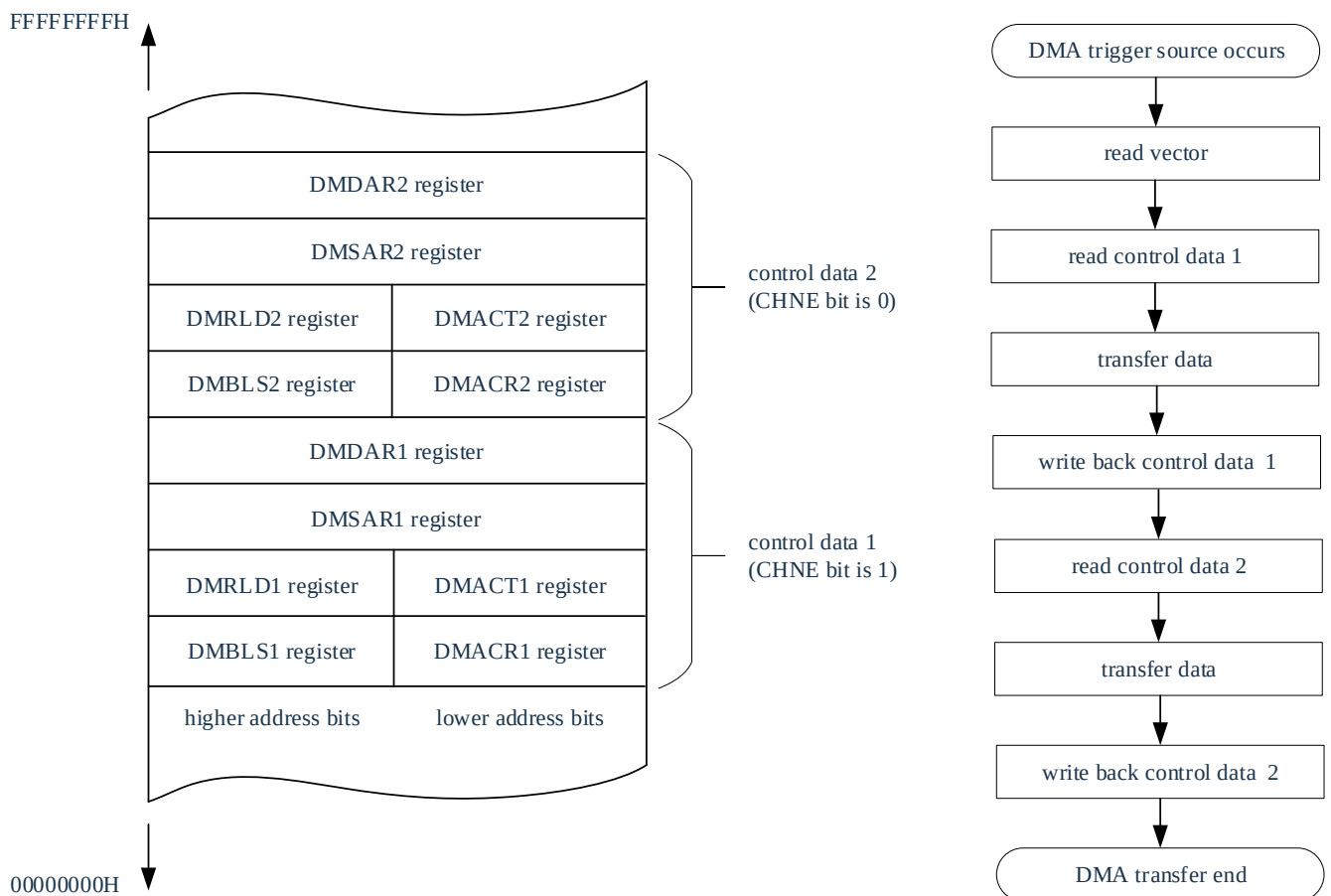
When the CHNE bit of the DMACRj (j=0~23) register is 1 (chain transfer is enabled), multiple data transfers can be performed consecutively through 1 boot source.

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 startup 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. Repeat this operation until the transfer of control data with a CHNE bit at 0 (chain transfer disabled) is completed.

When the plurality of control data is used for chain transfer, the transfer times of the first control data setting are valid, and the transfer times of the second subsequent processing are invalid.

The flow chart for chain transfer is shown in Figure 16-22.

Figure 16-22 Flow chart for chain transfer



Cautions:

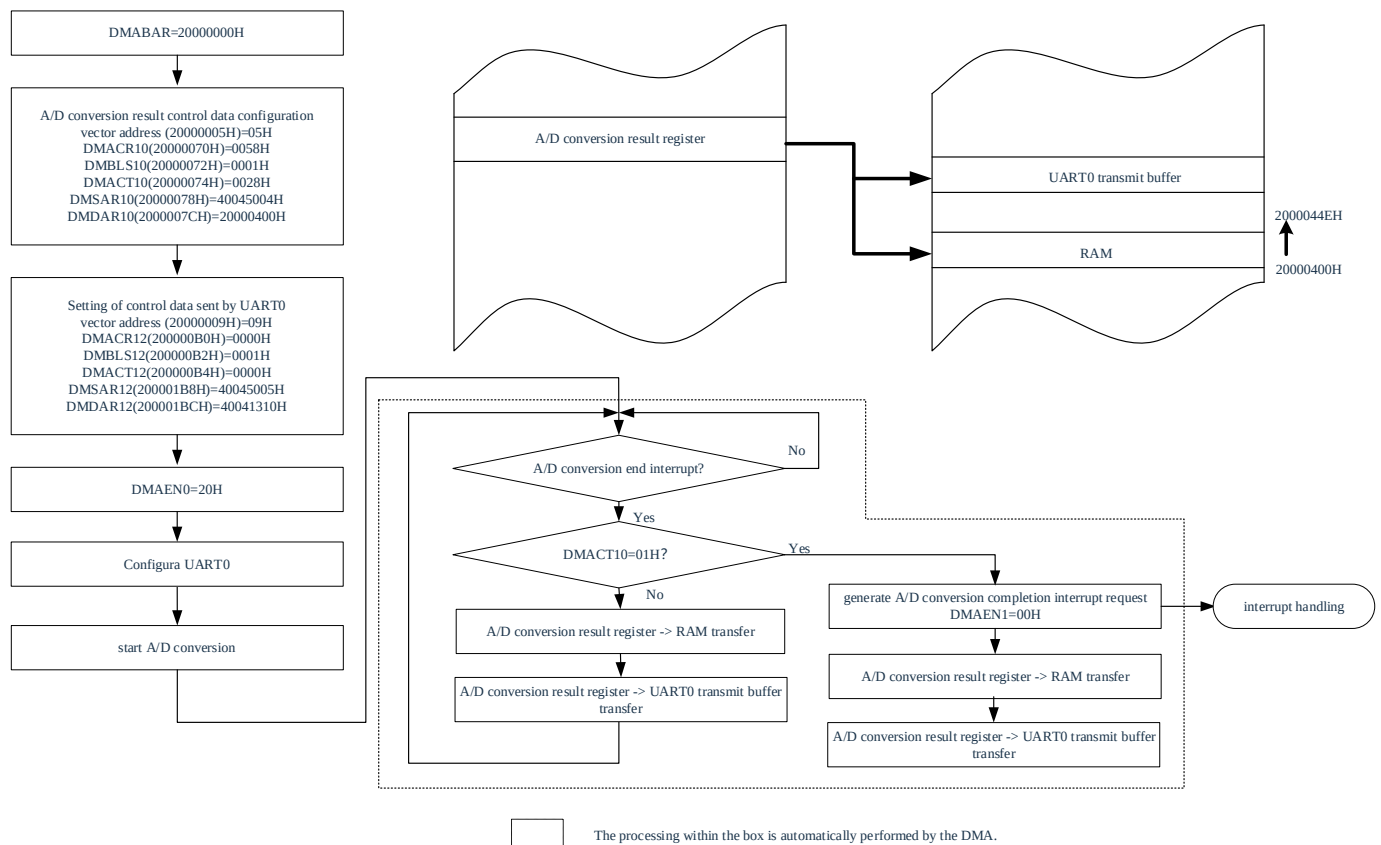
1. The CHNE bit of the DMACR23 register must be set to 0 (chain transfer disabled).
2. For data transfer after the second chain transfer, the DMAENi0 to DMAENi7 bits of the DMAENi (i=0~2) register are unchanged to 0 (DMA boot is disabled), and no interrupt request is generated.

(1) Examples of chain transfer: Continuously taking A/D conversion result for UART0 transmission

The DMA is started by the A/D conversion end interrupt and the A/D conversion result is transferred to RAM for UART0 transmission.

- The vector addresses are 20000005H and 20000009H respectively.
- The control data of the A/D conversion result is assigned at 20000070H to 2000007FH.
- The control data of UART0 transmission is assigned at 200000B00H to 200000BFH.
- The two-byte data of 40045004H and 40045005H is transferred to 20000400H to 2000044FH of RAM, and the high 1-byte (40045005H) of A/D is transferred to UART0 transmission buffer (40041310H).

Figure 16-23 Examples of chain transfer: Continuously taking A/D conversion result for UART0 transmission



16.5 Cautions When Using DMA

16.5.1 DMA Control Data and Vector Table Settings

- The DMA base address register (DMABAR) must be changed with all DMA start sources set to disabled.
- The DMA base address register (DMABAR) can only be rewritten once.
- The data in the DMACRj, DMBLSj, DMACTj, DMRLDj, DMSARj, and DMDARj registers must be changed when the DMAENi0 to DMAENi7 bits of the corresponding DMAENi (i=0~2) registers are 0 (DMA boot disabled).
- 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=0~2) register are 0 (DMA boot disabled).

16.5.2 Assignment of DMA Control Data Area and DMA Vector Table Area

The areas that can be assigned DMA control data and vector tables vary by product and usage conditions.

- The stack area, DMA control data area, and DMA vector table area cannot overlap.
- When a parity error reset is allowed to be generated by the RAM parity error detection function (RPERDIS=0), the DMRLD register must be initialized (0000H) even when using the normal mode.

16.5.3 Number of Execution Clocks for DMA

The execution of the DMA at start-up and the number of clocks required are shown in Table 16-9.

Table 16-9 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 16-10.

Note 2: For the number of clocks required to read and write data, please refer to Table 16-11.

Table 16-10 Number of clocks required to write back control data

Settings for the DMACR register				Address setting		Control writeback of register				Clock count
DAMOD	SAMOD	RPTSEL	MODE	Source	Objective	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 area	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 area	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

Remark: j=0~23, X: “0” or “1”

Table 16-11 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 number ^{Note}
Write data	1	—	—	1	1	1+ wait number ^{Note}

16.5.4 Response Time for DMA

The DMA response time is shown in Table 16-12. The DMA response time is the time from the detection of the DMA start source to the start of the DMA transfer, excluding the number of DMA execution clocks.

Table 16-12 Response time for DMA

	Minimum time	Maximum time
Response time	3 clocks	23 clocks

However, the DMA response may also be delayed in the following cases. The number of delayed clocks varies depending on the condition.

- Maximum response time to execute instructions from internal RAM: 20 clocks

Remark: 1 clock: $1/F_{CLK}$ (F_{CLK} : CPU/peripheral hardware clock)

16.5.5 Start Source for DMA

- It is not possible to input the same start source during the period from the input of the DMA start source to the end of the DMA transfer.
- The DMA startup allow bit corresponding to the startup source cannot be manipulated at the location where the DMA startup source is generated.
- If the DMA startup source sends a contention, it judges the priority when the CPU accepts the DMA transfer and decides to start the startup source. Refer to 16.3.3 Vector Table for the priority of the startup source.
- If DMA start is allowed in one of the following states, a DMA transfer starts and an interrupt is generated at the end of the transfer. Therefore, it must be set to allow DMA start after the monitor flag (CnMON) of the comparator is recognized as necessary.
 - Set to generate an interrupt request by single edge detection of the comparator note (CnEDG=0) and to generate an interrupt request by the rising edge of the comparator (CnEPO=0) and IVCMP>IVREF (or internal reference voltage 1.45V).
 - Set to generate an interrupt request by single edge detection of the comparator (CnEDG=0) and an interrupt request by falling edge of the comparator (CnEPO=1) and IVCMP < IVREF (or internal reference voltage 1.45V).

(n=0, 1)

16.5.6 Operation in Standby Mode

Status	DMA operation
Sleep mode	Can operate (disable operation in low-power RTC mode).
Deep sleep mode	Can accept DMA start source, and perform DMA transfer ^{Note 1}

Note 1: In deep sleep mode, it is possible to perform a DMA transfer after detecting a DMA start source and return to deep sleep mode after the transfer is completed. However, since the code flash and data flash stop operating in the deep sleep mode, the flash cannot be set as the transfer source.

Chapter 17 Linkage Controller (EVENTC)

17.1 Functions of EVENTC

The EVENTC mutually connects events output from each peripheral function. By linking events, it becomes possible to coordinate operation between peripheral functions directly without going through the CPU.

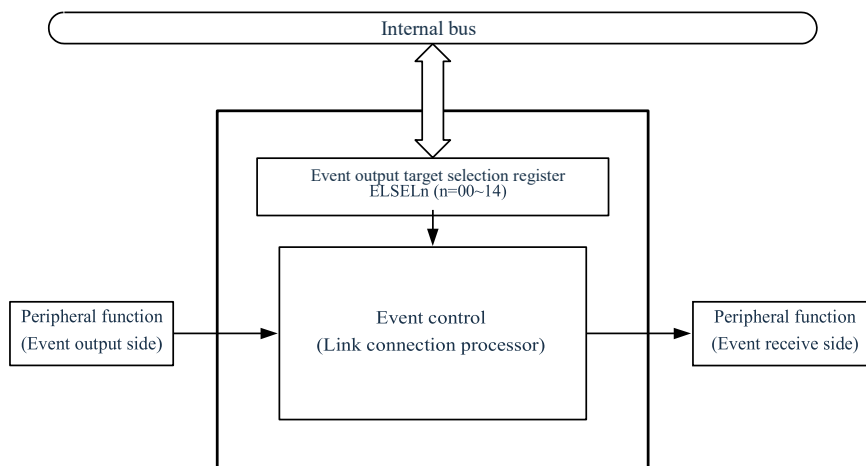
The EVENTC has the following features:

- Capable of directly linking event signals from 15 types of peripheral functions to specified peripheral functions.
- Event signals can be used as activation sources for operating any one of 3 types of peripheral functions.

17.2 Structure of EVENTC

Figure 17-1 shows the Event Link Controller Block Diagram.

Figure 17-1 Block diagram of EVENTC



17.3 Control Registers

Table 17-1 lists the control registers.

Table 17-1 Registers controlling EVENTC

Register name	Symbol
Event output target selection register 00	ELSELR00
Event output target selection register 01	ELSELR01
Event output target selection register 02	ELSELR02
Event output target selection register 03	ELSELR03
Event output target selection register 04	ELSELR04
Event output target selection register 05	ELSELR05
Event output target selection register 06	ELSELR06
Event output target selection register 07	ELSELR07
Event output target selection register 08	ELSELR08
Event output target selection register 09	ELSELR09
Event output target selection register 10	ELSELR10
Event output target selection register 11	ELSELR11
Event output target selection register 12	ELSELR12
Event output target selection register 13	ELSELR13
Event output target selection register 14	ELSELR14

17.3.1 Output Target Selection Register n (ELSELRn) (n=00~14)

The ELSELRn register links each event signal to an operation of an event-receiving peripheral function (link target peripheral function) after reception. Do not set multiple event inputs to the same event output target (event receive side). The operation of the event receiving peripheral function will become undefined, and event signals may not be received correctly. In addition, do not set the event link generation source and the event link output target to the same function.

Set an ELSELRn register during a period when no event output peripheral functions are generating event signals.

Table 17-2 lists the correspondence between ELSELRn (n = 00 to 14) registers and peripheral functions, and Table 17-3 lists the correspondence between values set to ELSELRn (n = 00 to 14) registers and operation of link target peripheral functions at reception.

Figure 17-2 Format of event output target select register n (ELSELRn)

Address: 40043400H (ELSELR00)~4004340EH (ELSELR14) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	0	0	0	ELSELn1	ELSELn0

ELSELn1	ELSELn0	Selection of event links
0	0	Disable event links.
0	1	Select operation of peripheral function 1 to link ^{Note 1}
1	0	Select operation of peripheral function 2 to link ^{Note 1}
1	1	Select operation of peripheral function 3 to link ^{Note 1}

Note 1: Refer to Table 17-3 Correspondence between values set to ELSELRn (n = 00 to 14) registers and operation of link target peripheral functions at reception.

Table 17-2 Correspondence between ELSELRn register (n=00~14) and peripheral functions

Register name	Event generation source (output source for event input n)	Event content
ELSELR00	External interrupt edge detection 0	INTP0
ELSELR01	External interrupt edge detection 1	INTP1
ELSELR02	External interrupt edge detection 2	INTP2
ELSELR03	External interrupt edge detection 3	INTP3
ELSELR04	RTC fixed cycle/alarm clock consistency detection	INTRTC
ELSELR05	Timer40 channel 00 count/capture end	INTTM00
ELSELR06	Timer40 channel 01 count/capture end	INTTM01
ELSELR07	Timer40 channel 02 count/capture end	INTTM02
ELSELR08	Timer40 channel 03 count/capture end	INTTM03
ELSELR09	Timer41 channel 00 count/capture end	INTTM10
ELSELR10	Timer41 channel 01 count/capture end	INTTM11
ELSELR11	Timer41 channel 02 count/capture end	INTTM12
ELSELR12	Timer41 channel 03 count/capture end	INTTM13
ELSELR13	Comparator detection 0	INTCMP0
ELSELR14	Comparator detection 1	INTCMP1

Table 17-3 Correspondence between values set to ELSELRn (n = 00 to 14) registers and operation of link target peripheral functions at reception

ELSELRn3~ELSELRn0 bits of ELSELRn register	Link target number	Link target peripheral function	Operation during reception
01B	1	A/D converter	Start A/D conversion
10B	2	Timer input for Timer40 channel 0 ^{Note 1}	Delay counter, measurement of input pulse interval, external event counter
11B	3	Timer input for Timer40 channel 1 ^{Note 2}	Delay counter, measurement of input pulse interval, external event counter

Note 1. To select the timer input of Timer40 channel 0 as the link target peripheral function, first set the operation clock of channel 0 to F_{CLK} via Timer Clock Select Register 0 (TPS0), set the noise filter of the TI00 pin to OFF via Noise Filter Enable Register 1 (NFEN1) (TNFEN00=0), and set the timer input used by channel 0 to the event input signal of the link controller via Timer Input Selection Register 0 (TIS0).

Note 2. To select the timer input of Timer40 channel 1 as the link target peripheral function, first set the operation clock of channel 1 to F_{CLK} via Timer Clock Select Register 0 (TPS0), set the noise filter of the TI01 pin to OFF via Noise Filter Enable Register 1 (NFEN1) (TNFEN01=0), and set the timer input used by channel 1 to the event input signal of the link controller via Timer Input Selection Register 0 (TIS0).

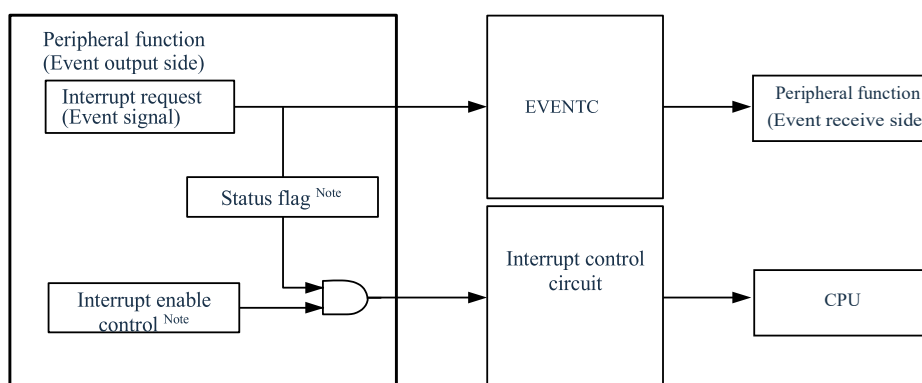
17.4 Operation of EVENTC

The path for using an event signal generated by a peripheral function as an interrupt request to the interrupt control circuit is independent from the path for using it as an EVENTC event. Therefore, each event signal can be used as an event signal for operation of an event-receiving peripheral function, regardless of interrupt control.

Figure 17-3 shows the relationship between interrupt handling and EVENTC. The figure shows an example of an interrupt request status flag and a peripheral function possessing the enable bits that control enabling/disabling of such interrupts.

A peripheral function which receives an event from the ELC will perform the operation corresponding to the event receiving peripheral function after reception of an event (see Table 17-3 Correspondence Between Values Set to ELSEL_{Rn} (n = 00 to 14) Registers and Operation of Link Destination Peripheral Functions at Reception).

Figure 17-3 Relationship between interrupt handling and EVENTC



Note: Some peripheral features do not have this feature.

The response of the peripheral function that accepts the event is shown in Figure 17-4.

Table 17-4 Response of peripheral functions to receive events

Event receive target number	Function of event link target	Operation after event reception	Response
1	A/D converter	A/D conversion	The EVENTC event changes directly to the hardware trigger of the A/D conversion.
2	Timer input of Timer40 channel 0	External event counter for measuring input pulse width of delay counter	The detection of an edge is performed after 3 or 4 F_{CLK} cycles have elapsed from the occurrence of an EVENTC event.
3	Timer input of Timer40 channel 1	External event counter for measuring input pulse width of delay counter	The detection of an edge is performed after 3 or 4 F_{CLK} cycles have elapsed from the occurrence of an EVENTC event.

Chapter 18 Interrupt Function

The Cortex-M0+ processor has a built-in nested vector interrupt controller (NVIC), which supports up to 32 interrupt request (IRQ) inputs and one non-maskable interrupt (NMI) input, in addition to multiple internal exceptions.

In this system, the interrupt sources for 32 interrupt request (IRQ) inputs and one non-maskable interrupt (NMI) input are handled in this system. This user's manual only explains the processing in this system. Please refer to the user's manual of Cortex-M0+ processor for the functions of the built-in NVIC of Cortex-M0+ processor.

18.1 Types of Interrupt Function

There are two types of interrupts.

(1) Maskable interrupt

This is the interrupt that receives mask control. If the interrupt mask flag register is not turned on, the interrupt request will not be responded to even if it is generated.

A standby release signal can be generated to release deep sleep mode and sleep mode.

Maskable interrupts are categorized into external interrupt requests and internal interrupt requests.

(2) Unmaskable interrupt

This is an interrupt that does not accept mask control, and the CPU must respond to the interrupt request once it is generated.

18.2 Interrupt Source and Structure

The interrupt source list can be referenced in Table 18-1.

Table 18-1 List of interrupt sources (1/3)

Interrupt handling	Interrupt source number	Interrupt source		Internal/external	Basic structure type ^{Note 1}	24-pin	20-pin	16-pin
		Name	Trigger					
Maskable	0	INTLVI	Voltage detection ^{Note 2}	Internal	(A)	☐	☐	☐
	1	INTP0	Detection of pin input edges	External	(B)	☐	☐	☐
	2	INTP1	Detection of pin input edges			☐	☐	☐
	3	INTP2	Detection of pin input edges			☐	☐	☐
	4	INTP3	Detection of pin input edges			☐	☐	☐
	5	INTTM01H	Timer channel 01 count end or capture end (High 8-bit timer operation)	Internal	(A)	☐	☐	☐
	6	INTTM03H	Timer channel 03 count end or capture end (High 8-bit timer operation)			☐	☐	☐
	7	INTST2/ INTSSPI20	UART2 transmit transfer end or buffer empty end/SSPI20 transfer end or buffer empty interrupt			☐	☐	☐
	8	INTSR2/ INTSSPI21	UART2 receive transfer end/SSPI21 transfer end or buffer empty interrupt			☐	☐	☐
	9	INTSRE2	UART2 receive communication error			☐	☐	☐
	10	INTST0/ INTSSPI00	UART0 transmit transfer end or buffer empty end/SSPI00 transfer end or buffer empty interrupt			☐	☐	☐
	11	INTSR0/ INTSSPI01	UART0 receive transfer end/SSPI01 transfer end or buffer empty interrupt			☐	☐	☐
	12	INTSRE0	UART0 receive communication error			☐	☐	☐

Notes: 1. The basic structure types (A)~(C) correspond to the (A)~(C) in Figure 18-1.

2. This is the case when the bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 0.

Table 18-1 List of interrupt sources (2/3)

Interrupt handling	Interrupt source number	Interrupt source		Internal/external	Basic structure type ^{Note 1}	24-pin	20-pin	16-pin
		Name	Trigger					
Maskable	13	INTST1/ INTSSPI10	UART1 transmit transfer end or buffer empty end/SSPI10 transfer end or buffer empty interrupt	Internal	(A)	☐	☐	☐
	14	INTSR1/ INTSSPI11	UART1 receive transfer end/SSPI11 transfer end or buffer empty interrupt			☐	☐	☐
	15	INTSRE1	UART1 receive communication error			☐	☐	☐
	16	INTIICA0	IICA0 communication end			☐	☐	☐
	17	INTTM00	Timer channel00 count end or capture end			☐	☐	☐
	18	INTTM01	Timer channel01 count end or capture end			☐	☐	☐
	19	INTTM02	Timer channel 02 count end or capture end			☐	☐	☐
	20	INTTM03	Timer channel 03 count end or capture end			☐	☐	☐
	21	INTAD	A/D conversion end			☐	☐	☐
	22	INTRTC	Fixed period/alarm consistency detection for real time clocks			☐	☐	☐
	23	INTIT	Interval signal detection			☐	☐	☐
	24	INTCMP0	Comparator detection 0			☐	☐	☐
	25	INTCMP1	Comparator detection 1			☐	☐	☐
	26	Reserved				☐	☐	☐
	27	INTTM10	Timer channel 10 count end or capture end			☐	☐	☐
	28	INTTM11	Timer channel 11 count end or capture end			☐	☐	☐
	29	INTTM12	Timer channel 12 count end or capture end			☐	☐	☐
	30	INTTM13	Timer channel 13 count end or capture end			☐	☐	☐
	31	INTFL	Flash programming end			☐	☐	☐

Note: 1. The basic structure types (A)~(C) correspond to the (A)~(C) in Figure 18-1.

Table 18-1 List of interrupt sources (3/3)

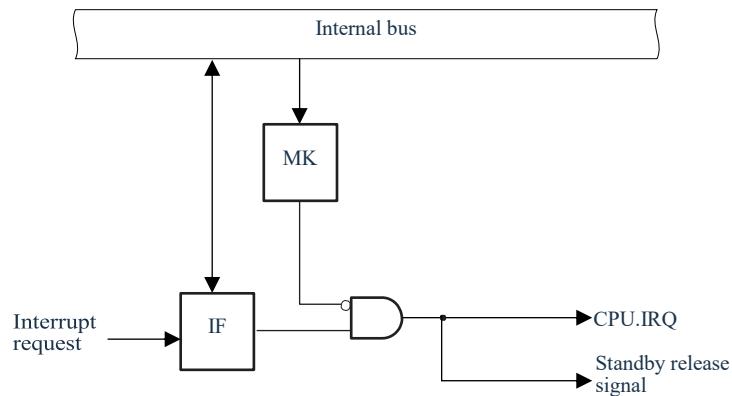
Interrupt handling	Interrupt source number	Interrupt source		Internal/external	Basic structure type ^{Note 1}
		Name	Trigger		
Non-maskable	—	INTWDT	Watchdog timer interval interrupt ^{Note 2}	Internal	(C)

Notes: 1. The basic structure types (A)~(C) correspond to the (A)~(C) in Figure 18-1.

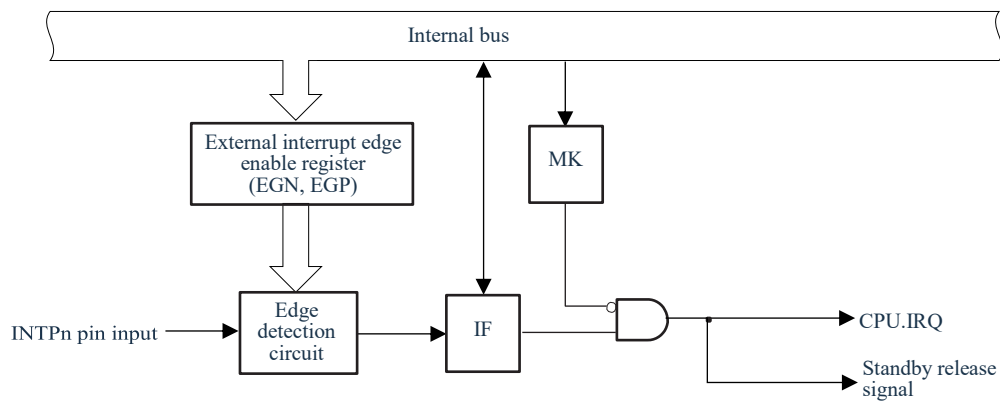
2. This is the case when the bit7 (WDTINT) of the option byte (000C0H) is set to 1.

Figure 18-1 Basic structure of interrupt function

(A) Internal maskable interrupt

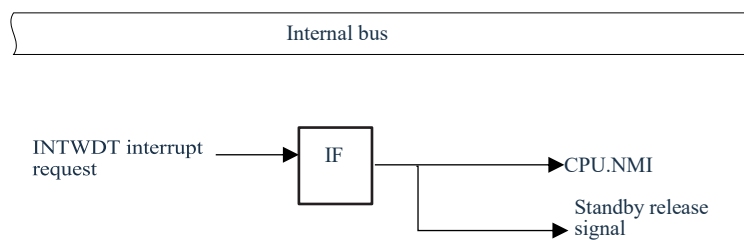


(B) External maskable interrupt (INTPn)



Note: n=0~3

(C) Non-maskable interrupt



Note: The interrupt request flag IF of non-maskable interrupts does not have a physical register and cannot be used to generate interrupt requests by reading or writing registers on the bus.

18.3 Registers for Controlling Interrupt Function

The interrupt function is controlled by the following registers.

- Interrupt request flag register (IF00~IF31)
- Interrupt mask flag register (MK00~MK31)
- External interrupt rising edge enable register (EGP0)
- External interrupt falling edge enable register (EGN0)

18.3.1 Interrupt Request Flag Register (IF00~IF31)

The interrupt request flag is set to 1 by generating a corresponding interrupt request or executing an instruction.

The interrupt request flag is cleared by generating a reset signal or by executing an instruction.

Set IF00L~IF31L, IF00H~IF31H registers by an 8-bit memory manipulation instruction.

Or set IF00~IF31 registers by a 32-bit memory manipulation instruction.

After a reset signal is generated, the value of these registers becomes 0000_0000H.

Figure 18-2 Format of interrupt request flag register (IFm) (m=0~31)

Address: IF00: 40006000H, IF01: 40006004H, IF02: 40006008H, IF03: 4000600CH
 IF04: 40006010H, IF05: 40006014H, IF06: 40006018H, IF07: 4000601CH
 IF08: 40006020H, IF09: 40006024H, IF10: 40006028H, IF11: 4000602CH
 IF12: 40006030H, IF13: 40006034H, IF14: 40006038H, IF15: 4000603CH
 IF16: 40006040H, IF17: 40006044H, IF18: 40006048H, IF19: 4000604CH
 IF20: 40006050H, IF21: 40006054H, IF22: 40006058H, IF23: 4000605CH
 IF24: 40006060H, IF25: 40006064H, IF26: 40006068H, IF27: 4000606CH
 IF28: 40006070H, IF29: 40006074H, IF30: 40006078H, IF31: 4000607CH
 Reset value: 0000_0000H R/W

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							IF

IFmL

IFmL	Interrupt request flag for interrupt source numbered 0 to 31
0	No interrupt request signal is generated.
1	An interrupt request is generated in the interrupt request state.

Note 1: The correspondence between the interrupt source and the interrupt request flag register is shown in Table 18-2.

Note 2: The correspondence between the interrupt request flag register and CPU.IRQ is shown in Figure 18-4.

18.3.2 Interrupt Mask Flag Register (MK00~MK31)

The interrupt mask flag is set to enable or disable the corresponding maskable interrupt processing.

Set MK00L~MK31L, MK00H~MK31H registers by an 8-bit memory manipulation instruction or set MK00~MK31 registers by a 32-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “FFFF_FFFF”.

Figure 18-3 Format of interrupt request mask register (MKm) (m=0~31)

Address: MK00: 40006100H, MK01: 40006104H, MK02: 40006108H, MK03: 4000610CH
 MK04: 40006110H, MK05: 40006114H, MK06: 40006118H, MK07: 4000611CH
 MK08: 40006120H, MK09: 40006124H, MK10: 40006128H, MK11: 4000612CH
 MK12: 40006130H, MK13: 40006134H, MK14: 40006138H, MK15: 4000613CH
 MK16: 40006140H, MK17: 40006144H, MK18: 40006148H, MK19: 4000614CH
 MK20: 40006150H, MK21: 40006154H, MK22: 40006158H, MK23: 4000615CH
 MK24: 40006160H, MK25: 40006164H, MK26: 40006168H, MK27: 4000616CH
 MK28: 40006170H, MK29: 40006174H, MK30: 40006178H, MK31: 4000617CH
 Reset value: FFFF_FFFFH R/W

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							MKL

MKmL	Interrupt handling control for interrupt sources numbered 0 to 31 ^{Note 1}
0	Enable interrupt handling.
1	Disable interrupt handling.

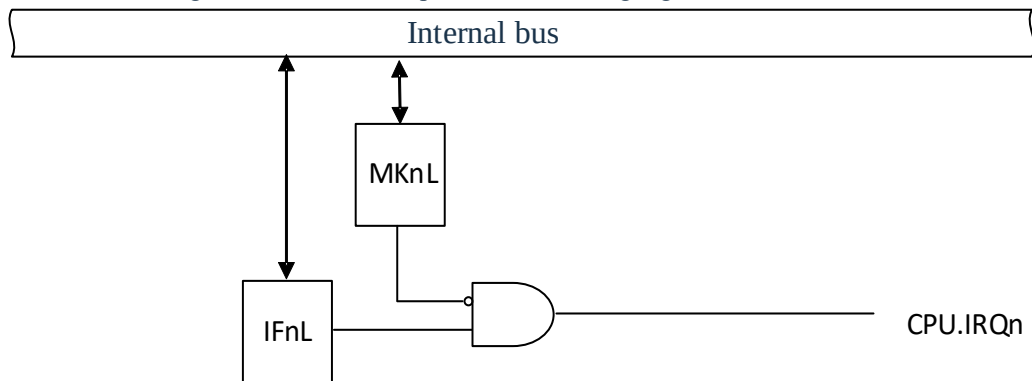
Note 1: The corresponding relationship between interrupt source and interrupt request mask register is shown in Table 18-2.

Note 2: The corresponding relationship between interrupt request mask register and CPU.IRQ is shown in Figure 18-4.

Table 18-2 Correspondence between interrupt sources and flag registers

No.	Interrupt source	Interrupt request flag register	Interrupt mask flag register
0	INTLVI	IF00.IFL	MK00.MKL
1	INTP0	IF01.IFL	MK01.MKL
2	INTP1	IF02.IFL	MK02.MKL
3	INTP2	IF03.IFL	MK03.MKL
4	INTP3	IF04.IFL	MK04.MKL
5	INTTM01H	IF05.IFL	MK05.MKL
6	INTTM03H	IF06.IFL	MK06.MKL
7	INTST2/INTSSPI20	IF07.IFL	MK07.MKL
8	INTSR2/INTSSPI21	IF08.IFL	MK08.MKL
9	INTSRE2	IF09.IFL	MK09.MKL
10	INTST0/INTSSPI00	IF10.IFL	MK10.MKL
11	INTSR0/INTSSPI01	IF11.IFL	MK11.MKL
12	INTSRE0	IF12.IFL	MK12.MKL
13	INTST1/INTSSPI10	IF13.IFL	MK13.MKL
14	INTSR1/INTSSPI11	IF14.IFL	MK14.MKL
15	INTSRE1	IF15.IFL	MK15.MKL
16	INTIICA0	IF16.IFL	MK16.MKL
17	INTTM00	IF17.IFL	MK17.MKL
18	INTTM01	IF18.IFL	MK18.MKL
19	INTTM02	IF19.IFL	MK19.MKL
20	INTTM03	IF20.IFL	MK20.MKL
21	INTAD	IF21.IFL	MK21.MKL
22	INTRTC	IF22.IFL	MK22.MKL
23	INTIT	IF23.IFL	MK23.MKL
24	INTCMP0	IF24.IFL	MK24.MKL
25	INTCMP1	IF25.IFL	MK25.MKL
26	INTRAMPRTERR	IF26.IFL	MK26.MKL
27	INTTM10	IF27.IFL	MK27.MKL
28	INTTM11	IF28.IFL	MK28.MKL
29	INTTM12	IF29.IFL	MK29.MKL
30	INTTM13	IF30.IFL	MK30.MKL
31	INTFL	IF31.IFL	MK31.MKL

Figure 18-4 Relationship between each flag register and CPU.IRQ



18.3.3 External Interrupt Rising Edge Enable Register (EGP0), External Interrupt Falling Edge Enable Register (EGN0)

These registers set the active edges of INTP0 to 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 change to “00H”.

Figure 18-5 Format of External interrupt rising edge enable register (EGP0), External interrupt falling edge enable register (EGN0)

Address: 40045B38H		After reset: 00H		R/W				
Symbol	7	6	5	4	3	2	1	0
EGP0	0	0	0	0	EGP3	EGP2	EGP1	EGP0

Address: 40045B39H		After reset: 00H		R/W				
Symbol	7	6	5	4	3	2	1	0
EGN0	0	0	0	0	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	Effective edge selection for INTPn pin (n=0 to 11)
0	0	Disable edge detection.
0	1	Falling edge
1	0	Rising edge
1	1	Rising and falling edges

The ports corresponding to the EGPn and EGNn bits are shown in Table 18-3.

Table 18-3 Interrupt request signals corresponding to the EGPn and EGNn bits

Detection enable bit		Interrupt request signal	24-pin	20-pin	16-pin
EGP0	EGN0	INTP0	○	○	○
EGP1	EGN1	INTP1	○	○	○
EGP2	EGN2	INTP2	○	○	○
EGP3	EGN3	INTP3	○	○	○

Caution: If you switch the input port used by the external interrupt function to output mode, an INTPn interrupt may be detected. When switching to output mode, the port mode register (PMxx) must be set to 0 after disabling the detection edges (EGPn, EGNn=0, 0).

Remarks: 1. For port configuration related to edge detection, please refer to 2.3.7 Port Alternate Function Configuration Register (PxxCFG).

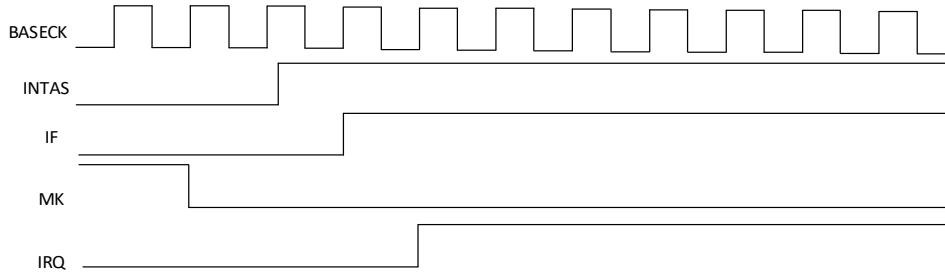
2. n=0~3

18.4 Operation of Interrupt Handling

18.4.1 Acceptance of Maskable Interrupt Requests

If the interrupt request flag is set to 1 and the mask (MK) flag for the interrupt request is cleared to 0, the interrupt request is accepted and can be passed to the NVIC.

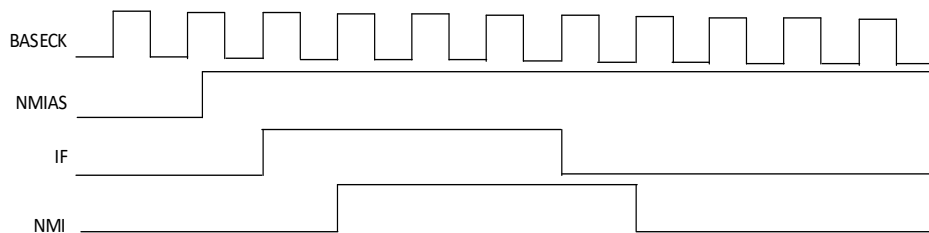
From the time the interrupt request flag is set to 1, to the time the CPU's IRQ is set to 1, it takes only 1 clock.



18.4.2 Acceptance of Non-Maskable Interrupt Requests

If a non-maskable interrupt request is generated, the interrupt request flag will be set to 1 and passed directly to the NVIC.

From the time the interrupt request flag is set to 1, to the time the CPU's NMI is set to 1, it takes only 1 clock.



Chapter 19 Standby Function

19.1 Standby Function

The standby function is a function that further reduces the operating current of the system, and there are two modes as follows.

(1) Sleep mode

Sleep mode is the mode in which the CPU is stopped from running the clock. If the high-speed system clock oscillation circuit, high-speed on-chip oscillator, or subsystem clock oscillation circuit is oscillating before the sleep mode is set, the clocks continue to oscillate. Although this mode does not reduce the operating current to the level of deep sleep mode, it is an effective mode for wanting to restart processing immediately through interrupt requests or if you want to run frequently in intermittent operations.

(2) Deep sleep mode

Deep sleep mode is a mode that stops the oscillation of the high-speed system clock oscillation circuit and the high-speed on-chip oscillator and stops the entire system. The operating current of the CPU can be greatly reduced.

Because deep sleep mode can be dismissed by interrupt requests, intermittent operations can also be performed. However, in the case of the X1 clock, because the wait time to ensure oscillation stability is required when decommissioning the deep sleep mode, it is necessary to select the sleep mode if you need to start processing immediately through the interrupt request.

In either mode, registers, flags, and data memory are all left set to before standby mode, and the output latches and output buffers of the input/output ports are also maintained.

Cautions:

1. Deep sleep mode is only available when the CPU is running on the main system clock. When the CPU is running on the subsystem clock, it cannot be set to deep sleep mode. Sleep mode can be used regardless of whether the CPU is running on the main system clock or the subsystem clock.
2. When moving to deep sleep mode, WFI instructions must be executed after stopping peripheral hardware running in the master system clock.
3. To reduce the operating current of the A/D converter, the bit7 (ADCS) and bit0 (ADCE) of the A/D converter mode register 0 (ADM0) must be set to 0, and execute the WFI instruction after stopping the A/D conversion operation.
4. The option byte selects whether to continue or stop oscillation of the low-speed on-chip oscillator in sleep mode or deep sleep mode. For details, please refer to Chapter 25 Option Bytes.

19.2 Sleep Mode

19.2.1 Sleep Mode Configuration

When the SLEEPDEEP bit of the SCR register is 0, execute the WFI instruction and enter sleep mode. In sleep mode, the CPU stops operating, but the values of the internal registers are still maintained, and the peripheral modules remain in the state they were in before they entered sleep mode. The status of peripheral modules, vibrators, etc. in sleep mode is shown in Table 19-1.

Sleep mode can be set regardless of whether the CPU clock before setup is a high-speed system clock, a high-speed on-chip oscillator clock, or a sub-system clock.

Caution: When the interrupt mask flag is 0 (enable interrupt processing) and the interrupt request flag is 1 (generate an interrupt request signal), the interrupt request signal is used to decommission sleep mode. Therefore, even if the WFI command is executed in this case, it does not shift to sleep mode.

Table 19-1 Operation status in sleep mode (1/2)

Sleep mode setting Item			Execution of WFI instructions while the CPU is running at the main system clock		
			CPU runs at high-speed on-chip oscillator clock (f _{IH})	CPU runs at X1 clock (f _X)	CPU runs at external main system clock (f _{EX})
System clock			Stop to supply clocks to the CPU.		
	Main system clock	f _{IH}	Continues running (cannot be stopped).	Disable running.	
		f _X	Disable running.	Continues running (cannot be stopped).	Cannot run.
		f _{EX}		Cannot run.	Continues running (cannot be stopped).
	Subsystem clock	f _{XT}	Remain in the state before sleep mode.		
		f _{EXS}			
	Low-speed on-chip oscillator clock	f _{IL}	It is set by bit0 (WDSTBYON) and bit4 (WDTON) of the option byte (000C0H) and the WUTMMCK0 bit of the Subsystem Clock Supply Mode Control Register (OSMC). WUTMMCK0=1: oscillate WUTMMCK0=0 and WDTON=0: stop WUTMMCK0=0, WDTON=1 and WDSTBYON=1: oscillate WUTMMCK0=0, WDTON=1 and WDSTBYON=0: stop		
CPU			Stop running.		
Code Flash					
RAM			Stop running (can run when DMA is executed).		
Port (latch)			Remain in the state before sleep mode.		
Universal timer unit			Can run.		
Real time clock (RTC)					
15-bit interval timer					
Watchdog timer			Refer to Chapter 9 Watch Dog Timer		
Clock output/buzzer output			Can run.		
AD converter					
CMP					
Universal serial communication unit (SCI)					
Serial interface (IICA)					
Data transfer controller (DMA)			Links between runnable function blocks.		
Linkage controller					
Power-on reset function			Can run.		
Voltage detection function					
External interrupts					
CRC operation function	High-speed CRC	It can be run when DMA is performed in the operation of the RAM area.			
	General CRC				
RAM parity check function			It can be run when the DMA is performed.		
SFR protection function					

Remark: Stop running: Automatically stops running when shifting to sleep mode.

Disable running: Stops running before shifting to sleep mode.

f_{IH} : High-speed on-chip oscillator clock f_{IL} : Low-speed on-chip oscillator clock

f_X : X1 clock f_{EX} : External main system clock

f_{XT} : XT1 clock f_{EXS} : External subsystem clock

Table 19-1 Operation status in sleep mode (2/2)

Sleep mode setting			Execution of WFI instructions while the CPU is running at the subsystem clock	
Item			CPU running at XT1 clock (F _{XT})	CPU running on external subsystem clock (F _{EXS})
System clock			Stop to supply clocks to the CPU.	
	Main system Clock	f _{IH}	Disable running.	
		f _X		
		f _{EX}		
	Subsystem clock	f _{XT}	Continues running (cannot be stopped).	Cannot run.
		f _{EXS}	Cannot run.	Continues running (cannot be stopped).
Low-speed on-chip oscillator clock	f _{IL}	It is set by bit0 (WDSTBYON) and bit4 (WDTON) of the option byte (000C0H) and the WUTMMCK0 bit of the Subsystem Clock Supply Mode Control Register (OSMC). - WUTMMCK0=1: oscillate - WUTMMCK0=0 and WDTON=0: stop - WUTMMCK0=0, WDTON=1 and WDSTBYON=1: oscillate - WUTMMCK0=0, WDTON=1 and WDSTBYON=0: stop		
CPU			Stop running.	
Code Flash				
RAM			Stop running (can run when DMA is executed).	
Port (latch)			Remain in the state before sleep mode.	
Universal timer unit			When RTCLPC=0, it can run (otherwise it is prohibited).	
Real time clock (RTC)			Can run.	
15-bit interval timer				
Watchdog timer			Refer to Chapter 9 Watchdog Timer	
Clock output/buzzer output			When RTCLPC=0, it can run (otherwise it is prohibited).	
AD converter			Disable running.	
CMP			Can operate.	
Universal communication unit (SCI)			When RTCLPC=0, it can run (otherwise it is prohibited).	
Serial interface (IICA)			Disable running.	
Data transfer controller (DMA)			When RTCLPC=0, it can run (otherwise it is prohibited).	
Linakge controller			Links can be made between runnable function blocks.	
Power-on reset function			Can run.	
Voltage detection function				
External interrupts				
CRC operation function	High-speed CRC		Disable running.	
	General CRC		It can be run when DMA is performed in the operation of the RAM area.	
RAM parity check function			It can be run when the DMA is performed.	
SFR protection function				

Remark: Stop running: Automatically stops running when shifting to sleep mode.

Disable running: Stops running before shifting to sleep mode.

f_{IH} : High-speed on-chip oscillator clock f_{IL} : Low-speed on-chip oscillator clock

f_X : X1 clock f_{EX} : External main system clock

f_{XT} : XT1 clock f_{EXS} : External subsystem clock

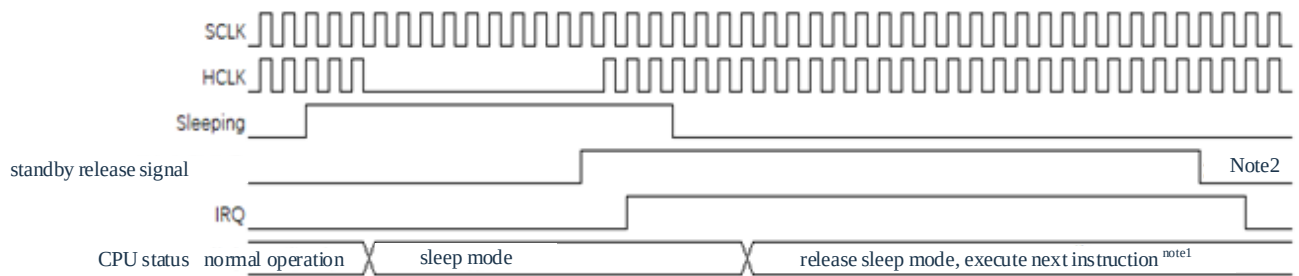
19.2.2 Release of Sleep Mode

Sleep mode can be released by any interrupt and external reset terminal, POR reset, low voltage detect reset, RAM parity error reset, WDT reset, software reset.

(1) Released by interrupts

When an unmasked interrupt is generated and is in a state that allows interrupts to be accepted, the sleep mode is released and the CPU begins to process the interrupt service program.

Figure 19-1 Release sleep mode by interrupt requests



Notes:

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.
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 program.

Caution: Before entering sleep mode, only the mask bit corresponding to the interrupt expected to be used to release sleep mode should be cleared to zero.

(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 19-2 Release sleep mode by resetting



Note 1: For the reset processing, please refer to “Chapter 20 Reset Function”. For reset processing for power-on reset (POR) circuits and voltage detection (LVD) circuits, refer to “Chapter 21 Power-on Reset Circuit”.

19.3 Deep Sleep Mode

19.3.1 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 19-2.

The deep sleep mode can only be set if the CPU clock before setting is the main system clock.

Caution: When the interrupt mask flag is 0 (allows interrupt processing) and 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 19-2 Operation status in deep sleep mode

Deep sleep mode setting Item			Execution of WFI instructions while the CPU is running at the main system clock		
			CPU runs at high-speed on-chip oscillator clock (f _{IH})	CPU runs at X1 clock (f _X)	CPU runs at external main system clock (f _{EX})
System clock			Stop to supply clocks to the CPU.		
	Main system clock	f _{IH}	Stop		
		f _X			
		f _{EX}			
	Subsystem clock	f _{XT}	Remain the state before it was set to deep sleep mode.		
		f _{EXS}			
f _{IL}		Set by bit0 (WDSTBYON) and bit4 (WDTON) of the option byte (000C0H) and the WUTMMCK0 bit of the Subsystem Clock Supply Mode Control Register (OSMC). WUTMMCK0=1: oscillate WUTMMCK0=0 and WDTON=0: stop WUTMMCK0=0, WDTON=1 and WDSTBYON=1: oscillate WUTMMCK0=0, WDTON=1 and WDSTBYON=0: stop			
CPU			Stop running.		
Code Flash					
RAM					
Port (latch)			Remain the state before it was set to deep sleep mode.		
Universal timer unit			Disable running.		
Real time clock (RTC)			Can run.		
15-bit interval timer					
Watchdog timer			Refer to Chapter 9 Watchdog Timer.		
Clock output/buzzer output			Operation is enabled when the subsystem clock is selected as the count clock and the RTCLPC bit is 0 (otherwise, operation is disabled).		
AD converter			Can perform a wake-up call.		
CMP			Can run (only without digital filters).		
Universal serial communication unit (SCI)			Only SSPIp and UARTq can wake up. All other operations are prohibited except for SSPIp and UARTq.		
Serial interface (IICA)			Can wake up by address matching.		
Data transfer controller (DMA)			Can accept DMA boot sources.		
Linkage controller			Links can be made between runnable function blocks.		
Power-on reset function			Can run.		
Voltage detection function					
External Interrupts					
CRC operation function	High-speed CRC		Stop running.		
	General CRC				
RAM parity check function					
SFR protection function					

Remark: Stop running: Automatically stops running when shifting to sleep mode.

Disable running: Stops running before shifting to sleep mode.

f_{IH} : High-speed on-chip oscillator clock f_{IL} : Low-speed on-chip oscillator clock

f_X : X1 clock f_{EX} : External main system clock

f_{XT} : XT1 clock f_{EXS} : External subsystem clock

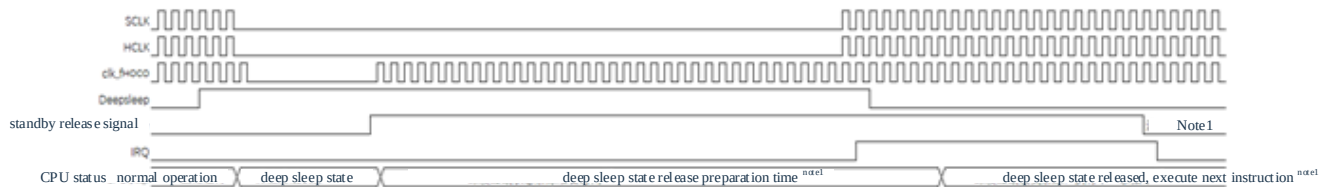
19.3.2 Release of Deep Sleep Mode

Deep sleep mode can be released in 2 ways.

(a) Released by unmasked interrupt request

If an unmasked interrupt request occurs, deep sleep mode is released. After the oscillation stabilization time, if the interrupt is allowed to be accepted, the vector interrupt is processed. If the interrupt acceptance is disabled, the next address is executed.

Figure 19-3 Release deep sleep mode by interrupt requests



Notes:

1. Standby release signal: For details of the standby release signal, please refer to “Figure 18-1 Basic structure of interrupt function”.
2. Deep sleep release preparation time:
When the CPU clock is a high-speed internal oscillation clock or an external clock input before entering deep sleep mode: 20μs~40μs
When entering deep sleep mode before the CPU clock is a high-speed system clock (X1 oscillation): 20μs~40μs and a longer time in the oscillation settling time (set by OSTs).
3. Wait: 14 clocks are required from the time CPU.IRQ is valid to the interrupt service program start.

Cautions:

1. Before entering sleep mode, only the mask bits corresponding to the interrupts expected to be used to release sleep mode should be cleared to zero.
2. When the CPU is running at high-speed system clock (X1 oscillation) and to shorten the oscillation stabilization time after the deep sleep mode is released, the CPU clock must be temporarily switched to the high-speed on-chip oscillator clock before the WFI instruction is executed.

Remark: The oscillation accuracy of the high-speed on-chip oscillator clock varies steadily depending on temperature conditions during deep sleep mode.

(b) 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 19-4 Release deep sleep mode by resetting



Note: For reset processing, see “Chapter 20 Reset Function”. For reset processing for power-on reset (POR) circuits and voltage detection (LVD) circuits, see “Chapter 21 Power-on Reset Circuit”.

Chapter 20 Reset Function

The following 7 methods generate a reset signal.

- (1) An external reset input via the RESETB pin.
- (2) An internal reset is generated by program runaway detection of the watchdog timer.
- (3) An internal reset is generated by comparing the power supply voltage and the detection voltage of the POR circuit.
- (4) An internal reset is generated by comparing the power supply voltage of the voltage detection circuit (LVD) with the detection voltage.
- (5) An internal reset is generated by setting the system reset request register bit (AIRC.R.SYSRESETREQ) to 1.
- (6) An internal reset is generated by RAM parity errors.
- (7) An internal reset is generated by accessing the illegal memory.

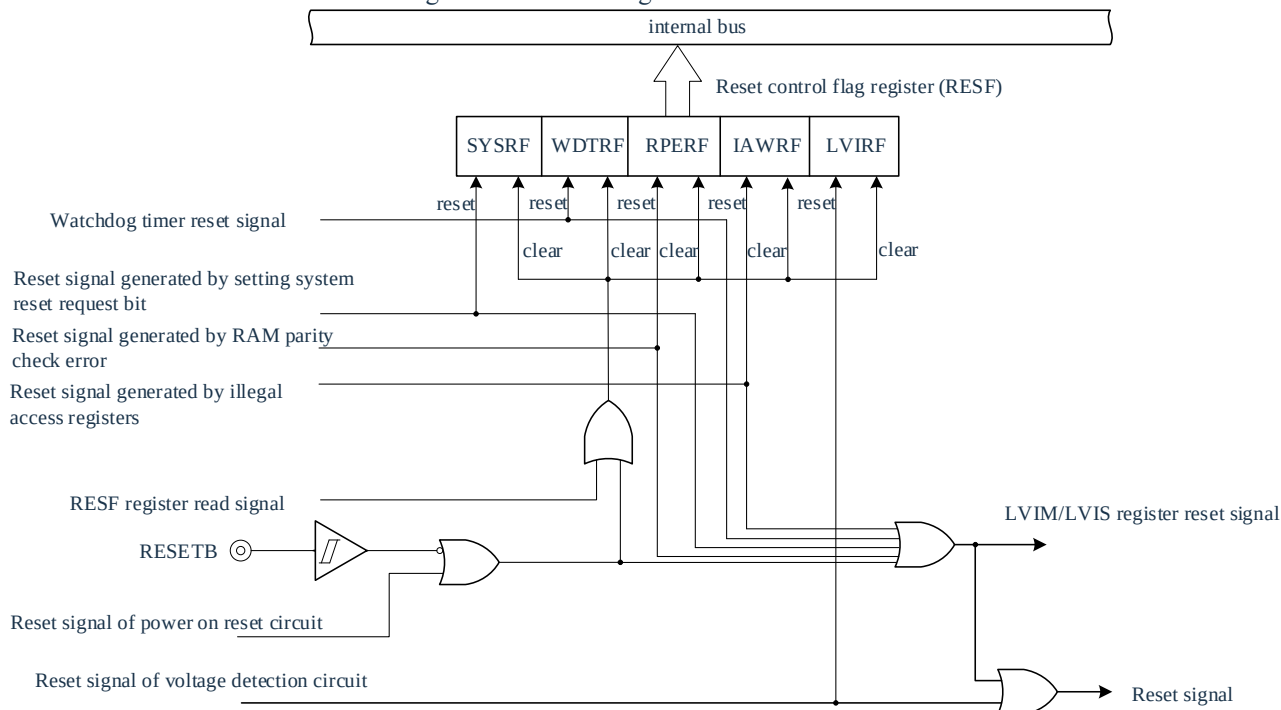
The internal reset and the external reset are the same, and after a reset signal is generated, the program is executed from the user-defined program start address.

When a low level is applied to the RESETB pin, or a program runaway is detected by the watchdog timer, or voltages are detected in the POR and LVD circuits, or the system reset request bit is set, or a RAM parity check error occurs, or an illegal memory is accessed, a reset is generated and each hardware changes to the state shown in Table 20-1.

Notes:

1. When an external reset is performed, 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 User's Manual, and then input a high level.
2. The oscillation of the X1 clock, XT1 clock, high-speed on-chip oscillator clock and low-speed on-chip oscillator clock is stopped during reset signal generation. The input of the external master clock and external sub system clock is invalid.
3. If a reset occurs, each SFR is initialized, so the port pins change to the following state:
 - P20, P21, P35, P36, P37: High impedance during external reset or POR reset. High during other resets and after accepting a reset (connect internal pull-up resistor).
 - Ports other than P20, P21, P35, P36, P37: High impedance during reset and after accepting reset.

Figure 20-1 Block diagram for reset function



Caution: The internal reset of the LVD circuit does not reset the LVD circuit.

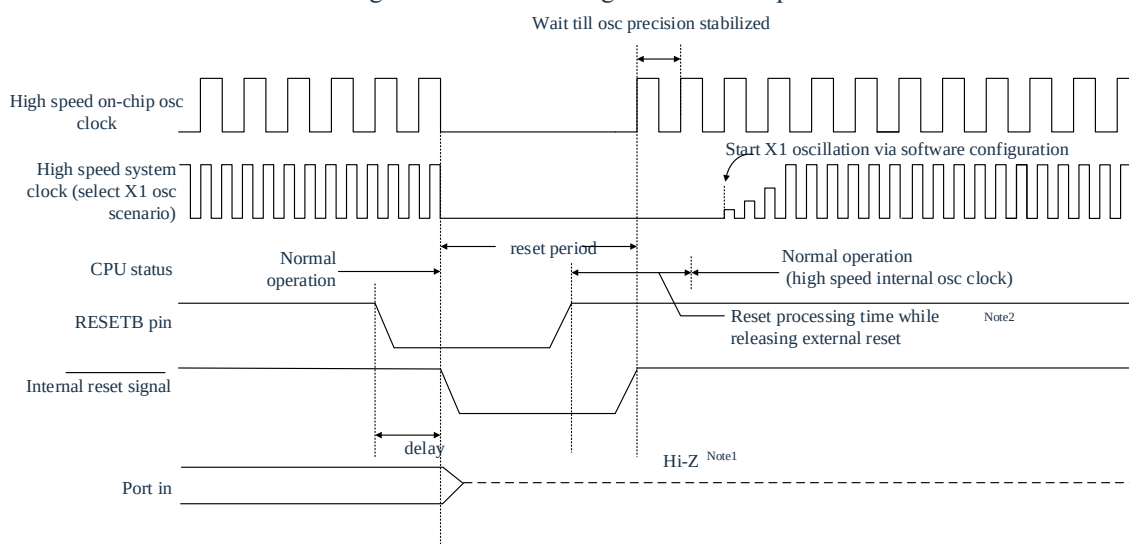
Remarks:

1. LVIM: Voltage detection register
2. LVIS: Voltage detection level register

Reset timing

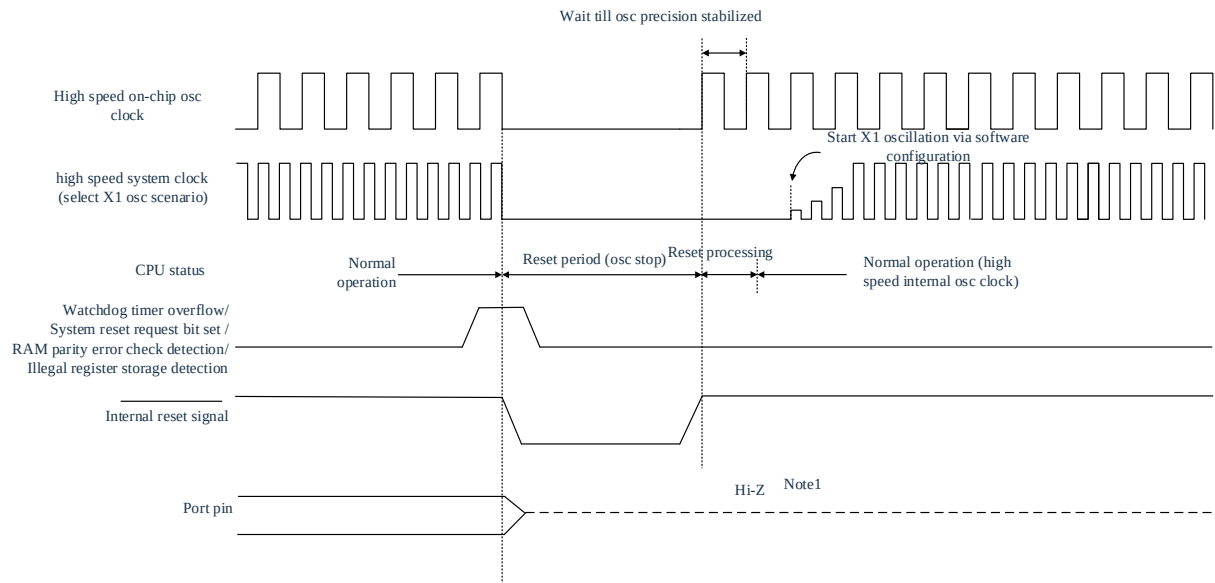
When the RESETB pin is input low, a reset is generated. The reset state is then released if the RESETB pin is entered high and the program begins with a high-speed on-chip oscillator clock after the reset process is completed.

Figure 20-2 Reset timing of RESETB input



The reset state is automatically released for the reset caused by the overflow of the watchdog timer, the setting of the system reset request bit, the detection of the RAM parity error or the detection of the illegal memory access, and the program is started by the high-speed on-chip oscillator clock after the reset process is finished.

Figure 20-3 Reset timing due to overflow of watchdog timer, set of system reset request bits, detection of RAM parity errors, or detection of illegal memory access



Note: 1. Ports P20, P21, P35, P36, P37 become the following states:

- High impedance during external reset or POR reset.
- High during other resets and after accepting a reset (connect internal pull-up resistor)

Caution: The watchdog timer is no exception and is reset when an internal reset occurs.

If $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD}$ is satisfied after the reset for the reset generated by the voltage detection of the POR and LVD circuits, the reset state is released and the program execution starts with the high-speed on-chip oscillator clock after the reset is processed. For details, refer to Chapter 21 Power-on Reset Circuit and Chapter 22 Voltage Detection Circuit.

Remark: V_{POR} : POR supply voltage rise detection voltage

V_{LVD} : LVD detection voltage

Table 20-1 Operation status during reset

Item			During resetting
System clock			Stop to supply clocks to CPU.
	Main system clock	f _{IH}	Stop running.
		f _X	Stop running (X1 pin and X2 pin are in input port mode).
		f _{EX}	Invalid clock input (pin is in input port mode).
	Sub system clock	f _{XT}	Can run.
		f _{EXS}	Invalid clock input (pin is in input port mode).
f _{IL}			Stop running.
CPU			Stop running.
Code Flash			Stop running.
RAM			Stop running.
Port (Latch)			High impedance ^{Note1}
Universal timer unit			Stop running.
RTC			
15-bit interval timer			
Watchdog timer			
Clock output/Buzzer output			
A/D converter			
CMP ^{Note1}			
Universal serial communications unit (SCI)			
Serial interface (IICA)			
Data transfer controller (DMA)			
Power-on reset function			Can perform detection runs.
Voltage detection function			Can run on LVD reset. On other resets, stops running.
External interrupt			Stop running.
Key interrupt function			
CRC operation function	High speed CRC		
	General CRC		
RAM parity check function			
SFR protection function			

Note: 1. Ports P20, P21, P35, P36, P37 become the following states:

High impedance during external reset or POR reset. High during other resets (connect internal pull-up resistor).

Remark	f_{IH}	: High-speed on-chip oscillator clock	f_X	: X1 oscillator clock
	f_{EX}	: External main system clock	f_{XT}	: XT1 oscillator clock
	f_{EXS}	: External subsystem clock	f_{IL}	: Low-speed on-chip oscillator clock

20.1 Registers for Confirming the Reset Source

20.1.1 Reset Control Flag Register (RESF)

The BAT32G133B microcontroller has multiple internal reset generation sources. The Reset Control Flag register (RESF) holds the reset source where the reset request occurs. The RESF register can be read by an 8-bit memory manipulation instruction.

The SYSRF, WDTRF, RPERF, IAWRF and LVIRF flags are cleared by inputting RESETB, resetting the power-on reset (POR) circuit, and reading the RESF register. To determine the reset source, the value of the RESF register must be saved to any RAM and then determined by its RAM value.

Figure 20-4 Format of reset control flag register (RESF)

Address: 40020440H

After reset: undefined^{Note 1} R

Symbol	7	6	5	4	3	2	1	0
RESF	SYSRF	0	0	WDTRF	0	RPERF	IAWRF	LVIRF

SYSRF	Internal reset request resulting from the system reset request bit being set
0	No internal reset request is generated or the RESF register is cleared.
1	Generate an internal reset request.

WDTRF	Internal reset request generated by the watchdog timer (WDT)
0	No internal reset request is generated or the RESF register is cleared.
1	Generate an internal reset request.

RPERF	Internal reset request generated by RAM parity error
0	No internal reset request is generated or the RESF register is cleared.
1	Generate an internal reset request.

IAWRF	Access to internal reset requests generated by illegal memory
0	No internal reset request is generated or the RESF register is cleared.
1	Generate an internal reset request.

LVIRF	Internal reset request generated by the voltage detection circuit (LVD)
0	No internal reset request is generated or the RESF register is cleared.
1	Generate an internal reset request.

Note 1: Different due to reset source. Refer to Table 20-2.

Caution: In the case of allowing RAM parity error reset (RPERDIS=0), when accessing data, the “RAM area used” must be initialized; When executing an instruction from a RAM region, the area of “RAM area +10 bytes used” must be initialized. By generating a reset, enter a state that allows the generation of a RAM parity error reset (RPERDIS=0). For details, please refer to “23.3.3 RAM Parity Error Detection Function”.

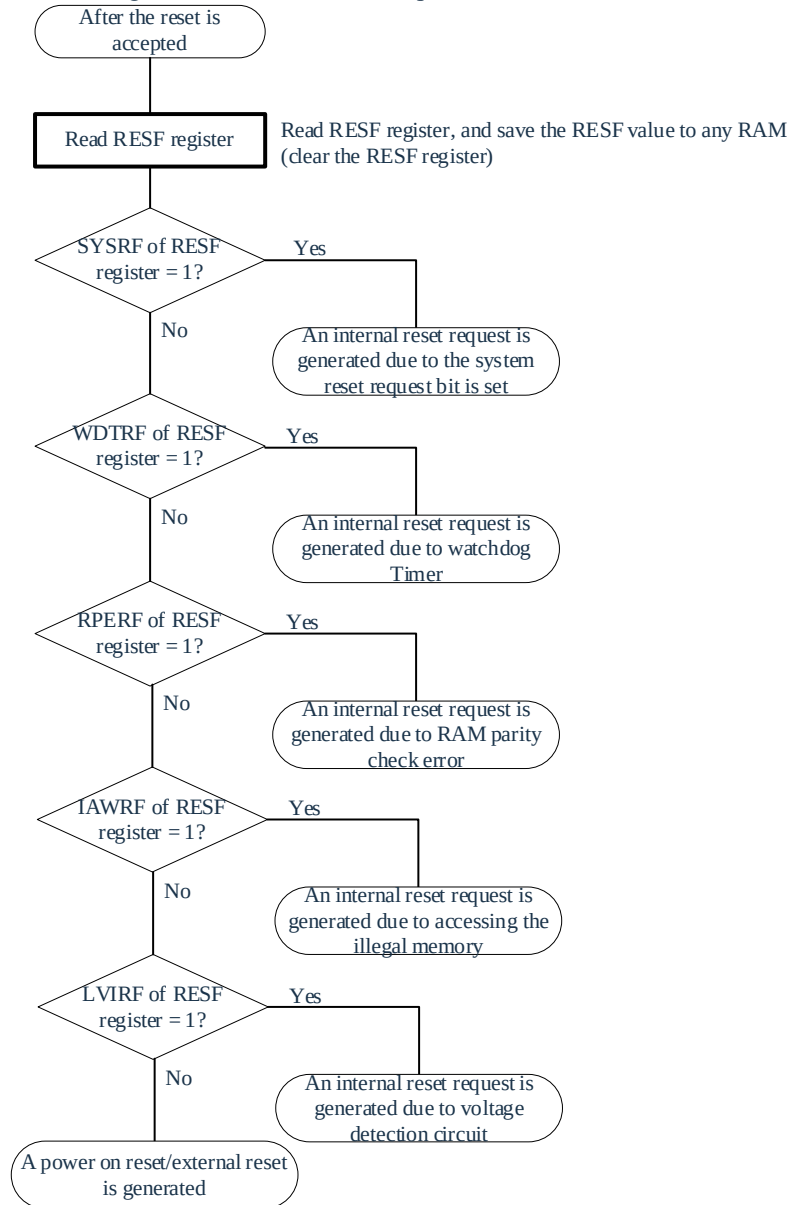
The RESF register status at the time of the reset request is shown in Table 20-2.

Table 20-2 RESF register state when reset request occurs

Reset source Flag	RESETB input	Reset generated by POR	Reset generated by system reset request bit set	Reset generated by WDT	Reset generated by RAM parity error	Reset generated by accessing illegal memory	Reset generated by LVD
SYSRF	Set "0"	Set "0"	Set "1"	Held	Held	Held	Held
WDTRF			Held	Set "1"	Set "1"		
RPERF				Held	Held	Set "1"	
IAWRF					Held	Held	
LVIRF						Held	Set "1"

The confirmation steps for the reset source are shown in Figure 20-5.

Figure 20-5 Confirmation steps for reset sources



Caution: The above procedure is an example of confirmation steps.

Chapter 21 Power-On Reset Circuit

21.1 Functions of Power-On Reset Circuit

The power-on reset circuit (POR) has the following functions.

- Generate an internal reset signal when power is turned on.

If the supply voltage (V_{DD}) exceeds the sense voltage (V_{POR}), the reset is dismissed. However, the reset state must be maintained by voltage detection circuitry or an external reset until the supply voltage reaches the operating voltage range shown in AC characteristics of the data sheet.

- Drag the supply voltage (V_{DD}) and the detection voltage (V_{PDR}) to compare. While $V_{DD} < V_{PDR}$, an internal reset signal is generated. However, when the supply voltage drops, the supply voltage must be lower than AC characteristics of the data sheet before the operating voltage range shown. It is reset by means of transfer in deep sleep mode, voltage detection circuitry, or external reset. When restarting operation, you must confirm that the supply voltage has returned to the operating voltage range.

Caution: When the power-on reset circuit generates an internal reset signal, the reset control flag register (RESF) is cleared to “00H”.

Remarks:

1. The BAT32G133 has built-in hardware to generate several internal reset signals. When the internal reset signal is generated by the watchdog timer (WDT), voltage detection (LVD) circuit, system reset request bit, RAM parity error or illegal memory access, the flag to indicate the reset source is assigned in the RESF register; when the internal reset signal is generated by the WDT, LVD, system reset request bit setting, RAM parity error or illegal memory access, the flag is set to 1 instead of clearing to 00H in the RESF register. For details of the RESF register, please refer to “Chapter 20 Reset function”.
2. V_{POR} : POR supply voltage rise detection voltage

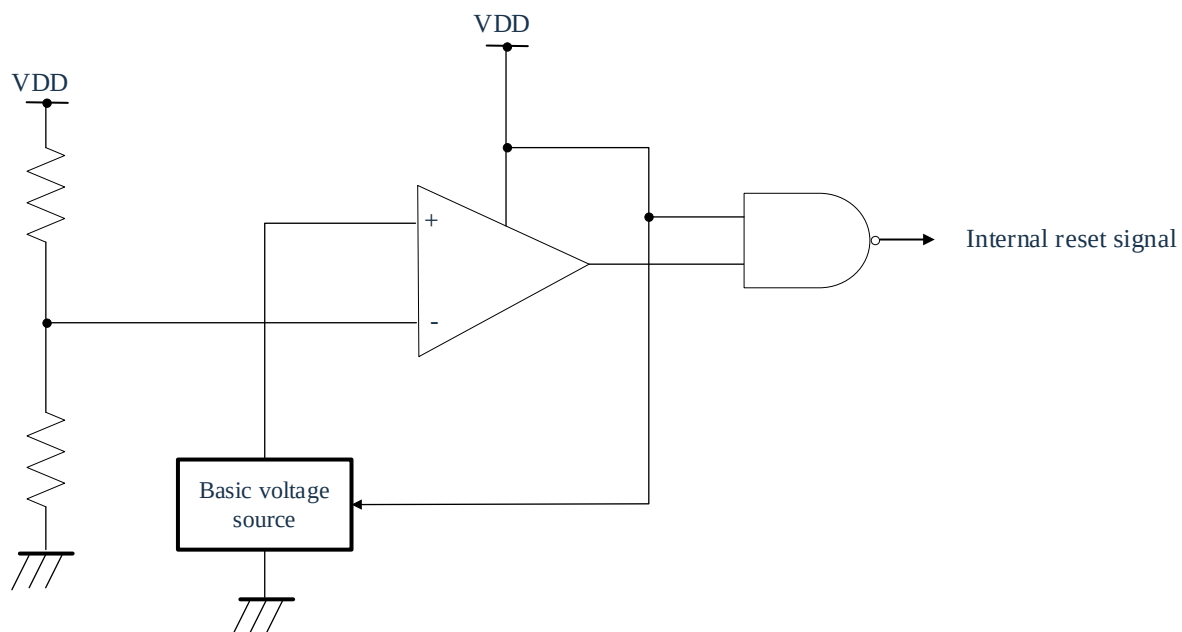
V_{PDR} : POR supply voltage drop detection voltage

Please refer to the POR circuit characteristics in the data sheet for details.

21.2 Structure of Power-On Reset Circuit

A block diagram of the power-on reset circuit is shown in Figure 21-1.

Figure 21-1 Block diagram of power-on reset circuit

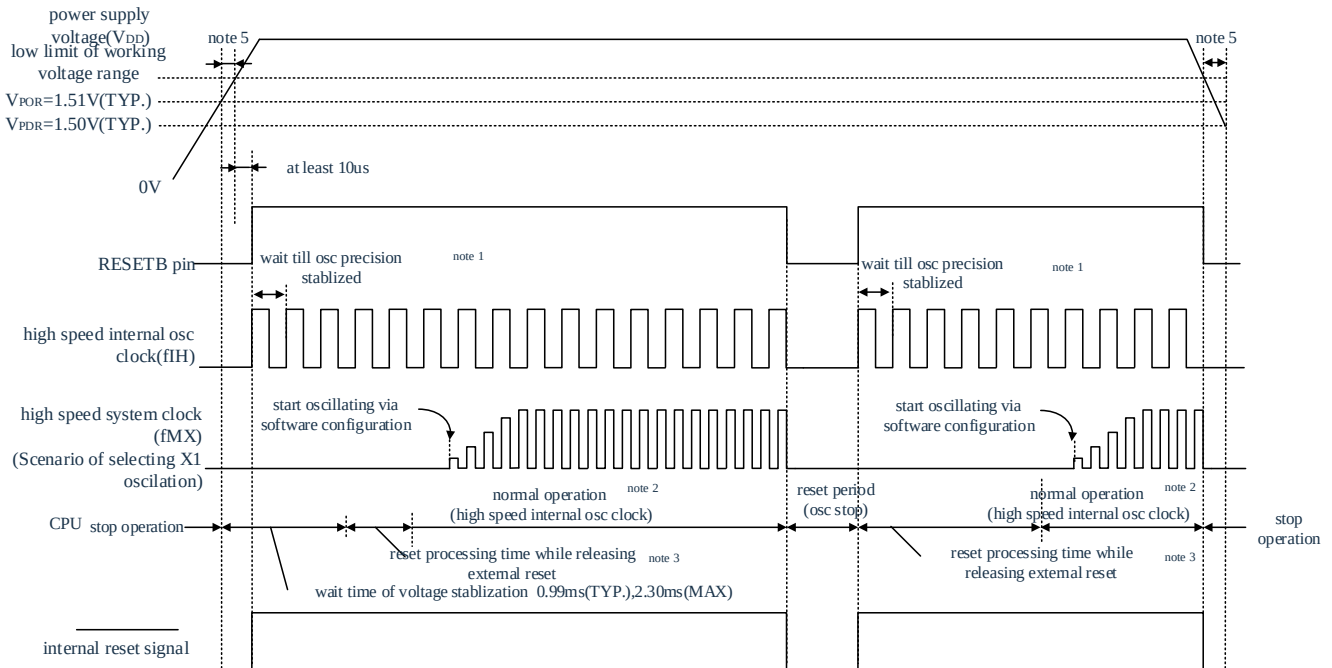


21.3 Operation of Power-On Reset Circuit

The generation timing of the internal reset signal of the power-on reset circuit and the voltage detection circuit is as follows.

Figure 21-2 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



Notes:

1. The internal reset processing time includes the oscillation accuracy stable waiting time of the high-speed on-chip oscillator clock.
2. The CPU clock can be switched from a high-speed on-chip oscillator clock to a high-speed system clock or a sub-system clock. In the case of using the X1 clock, switching must be performed after confirming the oscillation stabilization time by the OSTC of the oscillation stabilization time counter. In the case of XT1 clock, the switch must be made after checking the oscillation stability time by the timer function, etc.
3. The time until normal operation begins, in addition to the voltage stabilization wait time after reaching V_{POR} (1.51V(TYP.)), also requires the following reset processing time after releasing the external reset (first time after releasing POR) once the RESETB signal is set to a high level ("1").
4. When the power supply voltage rises, the reset state must be maintained through external reset before the power supply voltage reaches the operation voltage range indicated by the AC characteristic of the data manual; When the power supply voltage drops, it must be reset before the power supply voltage falls below the operation voltage range by the deep sleep mode transfer, voltage detection circuit or external reset. When you restart the operation, you must verify that the supply voltage is back in the operation voltage range.

Remark:

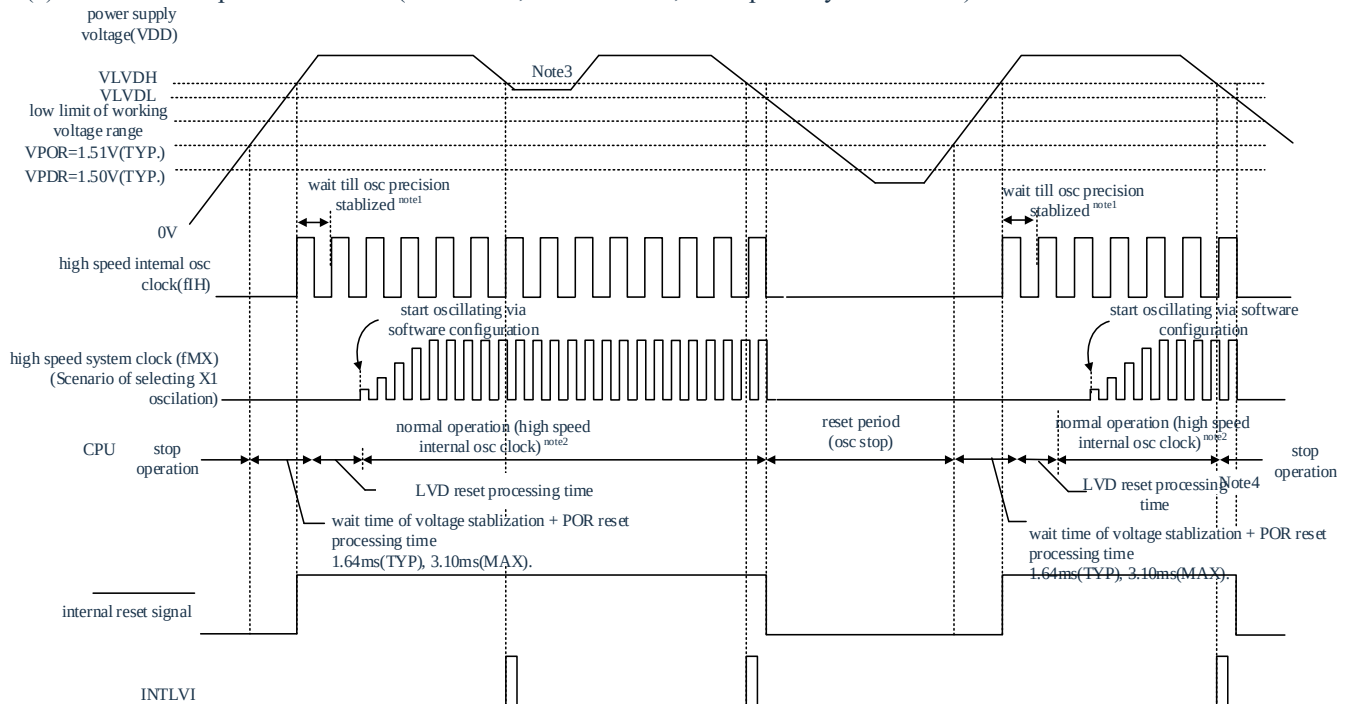
V_{POR}: POR supply voltage rise detection voltage

V_{PDR}: POR supply voltage drop detection voltage

Note: When LVD is OFF, you must use an external reset of the RESETB pin. For details, refer to Chapter 22 Voltage Detection Circuit.

Figure 21-2 Timing of internal reset signal generation for power-on reset circuit and voltage detection circuit (2/3)

(2) LVD in interrupt & reset mode (LVIMDS1, LVIMDS0=1, 0 of option bytes 000C1H)



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 CPU clock can be switched from the high-speed on-chip oscillator clock to the high-speed system clock or the subsystem clock. In the case of X1 clock, the switch must be made after checking the oscillation stability time by the status register of the oscillation stability time counter (OSTC); in the case of XT1 clock, the switch must be made after checking the oscillation stability time by the timer function, etc.

Note 3: After generating the interrupt request signal (INTLVI), the LVILV bit and the LVIMD bit of the voltage detection level register (LVIS) are automatically set to 1. Therefore, it must be considered that the supply voltage may return to the high voltage detection voltage (VLVDDH) or higher without falling below the low voltage detection voltage (VLVDDL), and after generating INTLVI, follow the steps in Figure 21-8 Setting procedure for confirmation/reset of operating voltage and Figure 21-9: Initial setting procedure for interrupt & reset mode.

Note 4: In addition to the “Voltage stabilization wait time + POR reset processing time” after reaching VPOR (1.51V(TYP.)), the following “LVD reset processing time” is required after reaching the LVD detection level (VLVDDH).

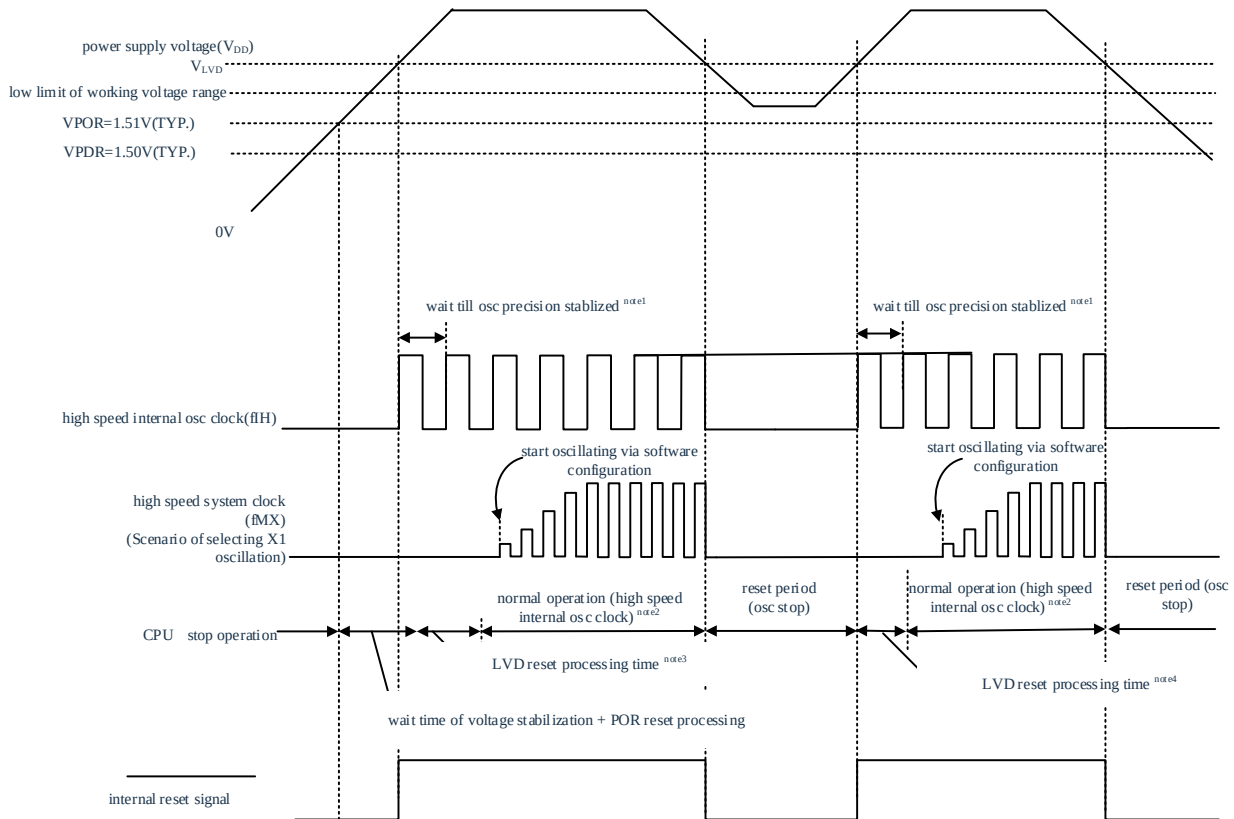
Remark: VLVDDH, VLVDDL: LVD detection voltage

VPOR: POR supply voltage rise detection voltage

VPDR: POR supply voltage drop detection voltage

Figure 21-2 Timing of internal reset signal generation for power-on reset circuit and voltage detection circuit (3/3)

(3) LVD reset mode (LVIMDS1, LVIMDS0=1, 1 for option byte 000C1H)



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 CPU clock can be switched from the high-speed on-chip oscillator clock to the high-speed system clock or the subsystem clock. In the case of X1 clock, the switch must be made after checking the oscillation stability time by the status register of the oscillation stability time counter (OSTC); in the case of XT1 clock, the switch must be made after checking the oscillation stability time by the timer function, etc.

Note 3: The time until the start of normal operation except for reaching V_{POR} (1.51V (TYP.)). In addition to the “Voltage Stabilization Wait Time + POR Reset Processing Time”, the following is required after the LVD detection level (V_{LVD}) is reached LVD Reset Processing Time.

Note 4: When the supply voltage drops, if the supply voltage is only restored after the internal reset of the voltage detection circuit (LVD) occurs, the following “LVD Reset Processing Time” is required after reaching the LVD detection level (V_{LVD}).

Remarks:

1. V_{LVDH} , V_{LVDL} : LVD detection voltage
 V_{POR} : POR supply voltage rise detection voltage
 V_{PDR} : POR supply voltage drop detection voltage
2. When the LVD interrupt mode is selected (LVIMD1, LVIMD0=0, 1 for option byte 000C1H), the time from power-on to the start of normal operation is the same as that in “Note 3” of “Figure 21-4(3/3) LVD Reset Mode”.

Chapter 22 Voltage Detection Circuit

22.1 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 25 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 (000C2H/010C2H).

a) 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.

b) 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.

c) 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.

Interrupt & Reset mode (LVIMDS1, LVIMDS0=1, 0)	Reset mode (LVIMDS1, LVIMDS0=1, 1)	Interrupt mode (LVIMDS1, LVIMDS0=0, 1)
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.

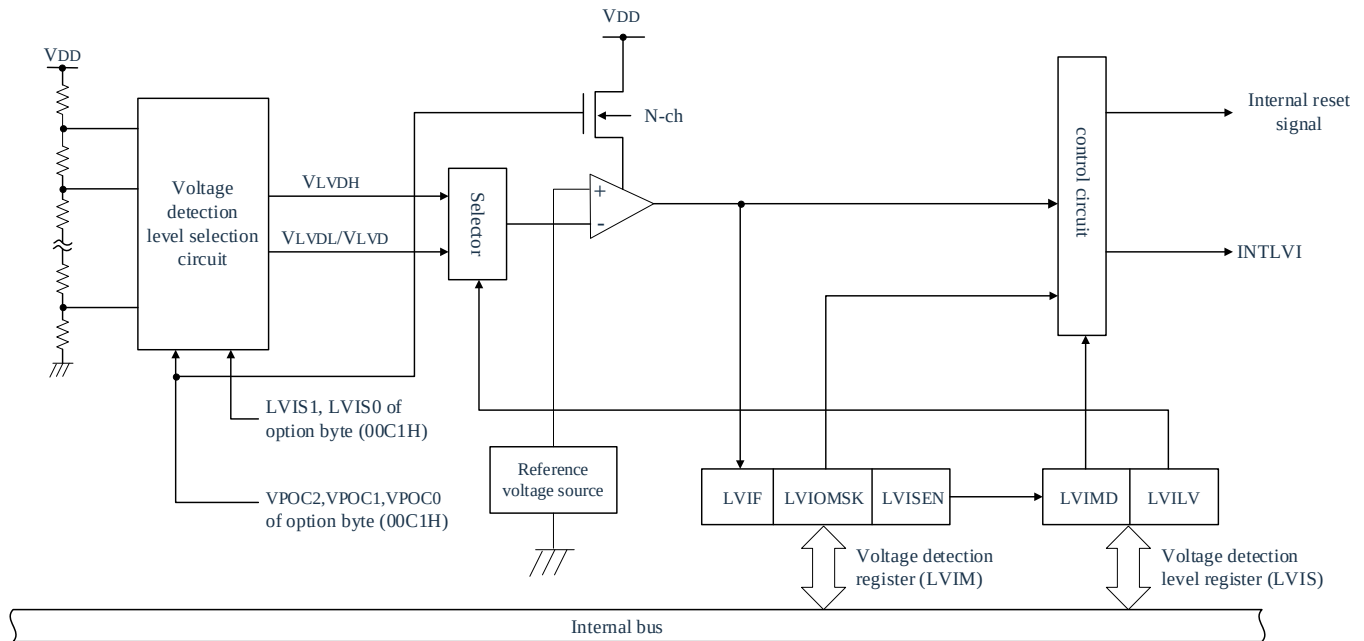
When the voltage detection circuit is operating, it can confirm whether the power supply voltage is greater than or equal to the detection voltage or less than the detection voltage by reading the voltage detection flag (LVIF: bit0 of the voltage detection register (LVIM)).

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 20 Reset Function.

22.2 Structure of Voltage Detection Circuit

The block diagram of the voltage detection circuit is shown in Figure 22-1.

Figure 22-1 Block diagram of voltage detection circuit



22.3 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)

22.3.1 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”.

Figure 22-2 Format of voltage detection register (LVIM)

Address: 40020441H After reset: 00H^{Note 1} R/W^{Note 2}

Symbol	7	6	5	4	3	2	1	0
LVIM	LVISEN ^{Note 3}	0	0	0	0	0	LVIOMSK	LVIF

LVISEN ^{Note 3}	Setting of the voltage detection level register (LVIS)
0	Disable rewriting LVIS register (LVIOMSK=0 (LVD output mask is invalid)).
1	Enable rewriting LVIS register (LVIOMSK=1 (LVD output mask valid)).

LVIOMSK	Mask status flag for LVD output
0	LVD output masking is invalid.
1	LVD output masking is valid ^{Note 4} .

LVIF	Voltage detection mark
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 reset value varies depending on the reset source.

When the LVD is reset, the value of the LVIM register is not reset and the original value is maintained; During other resets, clear the LVISEN to 0.

Note 2: Bit0 and bit1 are read-only bits.

Note 3: It can only be set when the interrupt & reset mode is selected (LVIMDS1 bit and LVIMDS0 bit of the option bytes are 1 and 0 respectively), the initial value cannot be changed in other modes.

Note 4: Only when the interrupt & reset mode is selected (LVIMDS1 bit and LVIMDS0 bit of the option byte are 1 and 0 respectively).

The LVIOMSK bit automatically changes to 1 during the following periods, masking the reset or interrupt generated by the LVD.

- When LVISEN=1
- Waiting time from the occurrence of LVD interrupt to the stabilization of LVD detection voltage
- Wait time from changing the value of the LVILV bit until the LVD detection voltage stabilizes

22.3.2 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 ^{Note 1}.

Figure 22-3 Format of voltage detection level register (LVIS)

Address: 40020442H

After reset: 00H/01H/81H^{Note 1}

R/W

Symbol	7	6	5	4	3	2	1	0
LVIS	LVIMD ^{Note 2}	0	0	0	0	0	0	LVILV ^{Note 2}

LVIMD ^{Note 2}	Operation mode of voltage detection
0	Interrupt mode
1	Reset mode

LVILV ^{Note 2}	LVD detection level
0	High voltage detection level (V _{LVDH})
1	Low voltage detection level (V _{LVDL} or V _{LVD})

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 22-7 Setting procedure for confirmation/reset of operating voltage and Figure 22-8: Initial setting procedure for interrupt & reset mode.
2. Select the operating mode of LVD and the detection voltage of each mode (V_{LVDH} , V_{LVDL} , V_{LVD}) by option byte 000C1H). The format of the user option bytes (000C1H/010C1H) is shown in Table 22-1. For details on option bytes, refer to Chapter 25 Option Byte.

Table 22-1 Format of user option bytes (000C1H/010C1H) (1/2)

Address: 000C1H/010C1H ^{Note}

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- 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
1.77V	1.73V	1.63V	0	0	0	1	0	1	0
1.88V	1.84V					0	1		
2.92V	2.86V					0	0		
1.98V	1.94V	1.84V		0	1	1	0		
2.09V	2.04V					0	1		
3.13V	3.06V					0	0		
2.61V	2.55V	2.45V		1	0	1	0		
2.71V	2.65V					0	1		
3.75V	3.67V					0	0		
2.92V	2.86V	2.75V		1	1	1	0		
3.02V	2.96V					0	1		
4.06V	3.98V					0	0		
—			Setting values other than those mentioned above is prohibited.						

- LVD setting (reset mode)

Detection voltage		Setting value of option byte						
V _{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
rise	drop						LVIMDS1	LVIMDS0
1.67V	1.63V	0	0	0	1	1	1	1
1.77V	1.73V		0	0	1	0		
1.88V	1.84V		0	1	1	1		
1.98V	1.94V		0	1	1	0		
2.09V	2.04V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.61V	2.55V		1	0	1	0		
2.71V	2.65V		1	0	0	1		
2.81V	2.75V		1	1	1	1		
2.92V	2.86V		1	1	1	0		
3.02V	2.96V		1	1	0	1		
3.13V	3.06V		0	1	0	0		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—			Setting values other than those mentioned above is prohibited.					

Remark: 1. The detection voltage is the TYP. value. For details, refer to LVD Circuit Characteristics in the datasheet.

Table 22-1 Format of user option bytes (000C1H) (2/2)

Address: 000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt mode)

Detection voltage		Setting value of option byte						
V _{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
rise	drop						LVIMDS1	LVIMDS0
1.67V	1.63V	0	0	0	1	1	0	1
1.77V	1.73V		0	0	1	0		
1.88V	1.84V		0	1	1	1		
1.98V	1.94V		0	1	1	0		
2.09V	2.04V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.61V	2.55V		1	0	1	0		
2.71V	2.65V		1	0	0	1		
2.81V	2.75V		1	1	1	1		
2.92V	2.86V		1	1	1	0		
3.02V	2.96V		1	1	0	1		
3.13V	3.06V		0	1	0	0		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—		Setting values other than those mentioned above is prohibited.						

- LVD is OFF (external reset using RESETB pin)

Detection voltage		Setting value of option byte						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
rise	drop						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		Setting values other than those mentioned above is prohibited.						

Notes:

- Write 1 to bit4.
- When the power supply voltage rises, the reset state must be maintained through the voltage detection circuit or external reset before the power supply voltage reaches the working voltage range indicated by the AC characteristic of the data sheet. When the power supply voltage drops, it must be reset before the power supply voltage falls below the operation voltage range by the deep sleep mode transfer, voltage detection circuit or external reset. The operation voltage range depends on the setting of the user option byte (000C2H).

Remarks:

- 1.×: Ignore
- The detection voltage is TYP. value. Refer to the LVD circuit characteristics in the datasheet for details.

22.4 Operation of Voltage Detection Circuit

22.4.1 Settings When Used as Reset Mode

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}).

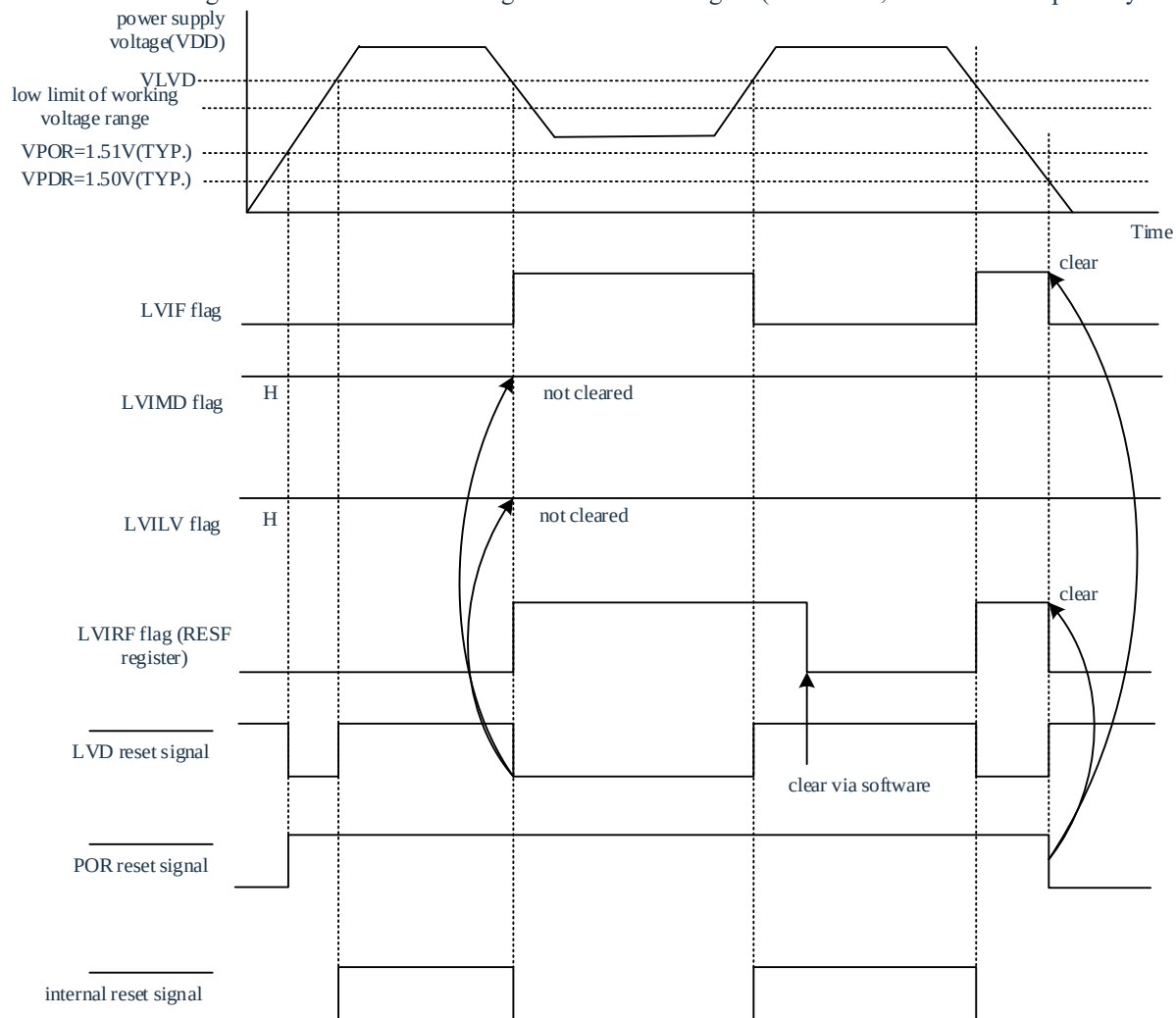
- Operation of LVD reset mode

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 22-4.

Figure 22-4 Generation timing of internal reset signal (LVIMDS1, LVIMDS0 of option bytes =1, 1)



Remark V_{POR} : POR supply voltage rise detection voltage

V_{PDR} : POR supply voltage drop detection voltage

22.4.2 Settings When Used as Interrupt Mode

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}).

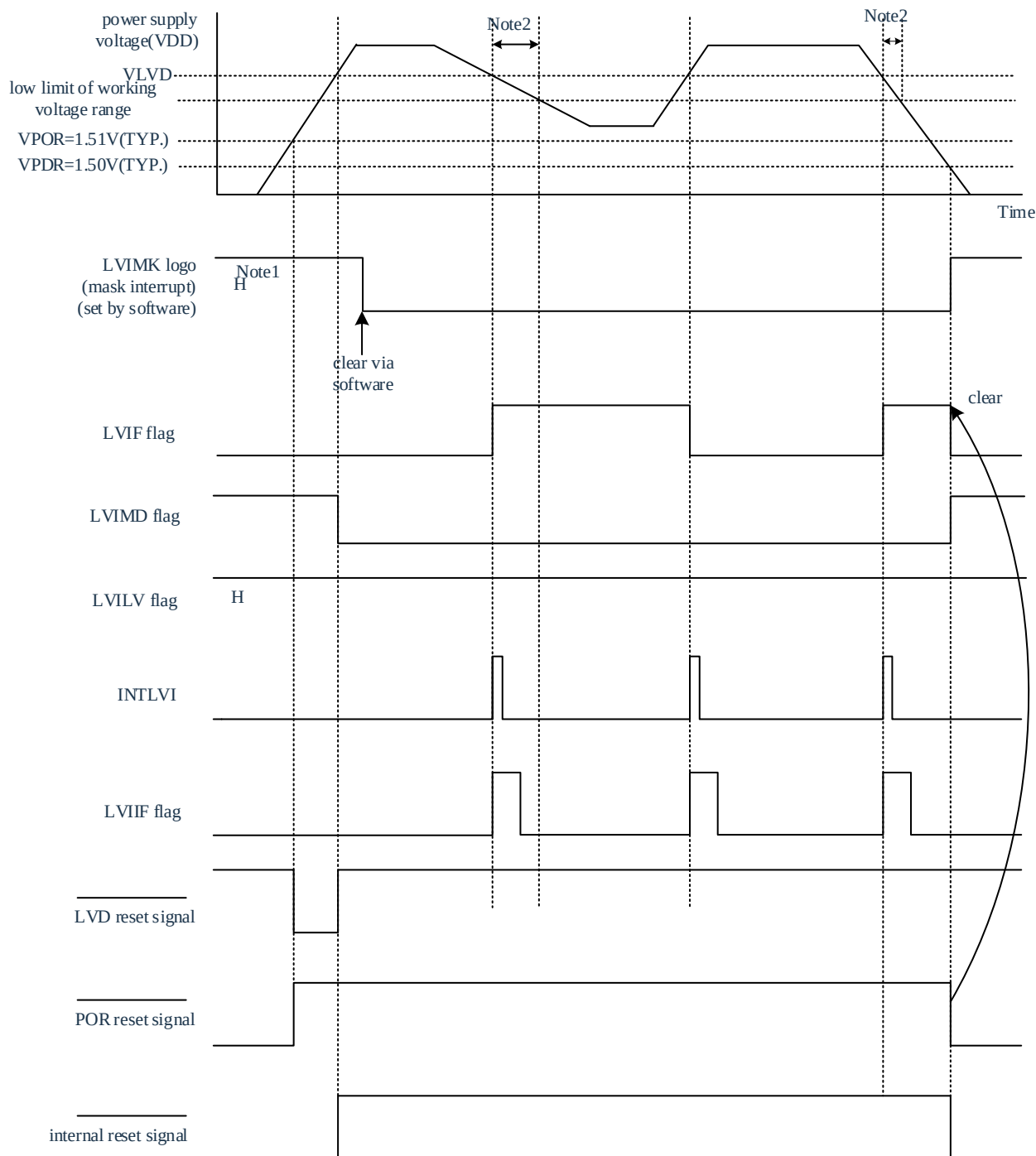
- Operation of LVD interrupt mode

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 (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 of the LVD is released.

The interrupt request signal (INTLVI) for the LVD is generated if the supply voltage (V_{DD}) 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 22-5.

Figure 22-5 Generation timing of interrupt signal (LVIMDS1, LVIMDS0 of option bytes =0, 1)



Notes 1. After a reset signal is generated, the LVIMK flag becomes 1.

2. 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. When restarting operation, it must be verified that the supply voltage returns to the operating voltage range.

Remark: V_{POR} : POR power supply voltage rise detection voltage

V_{PDR} : POR power supply voltage drop detection voltage

22.4.3 Settings When Used as Interrupt & Reset Mode

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 (V_{DD}) exceeds the high voltage detection level (V_{LVDH}).

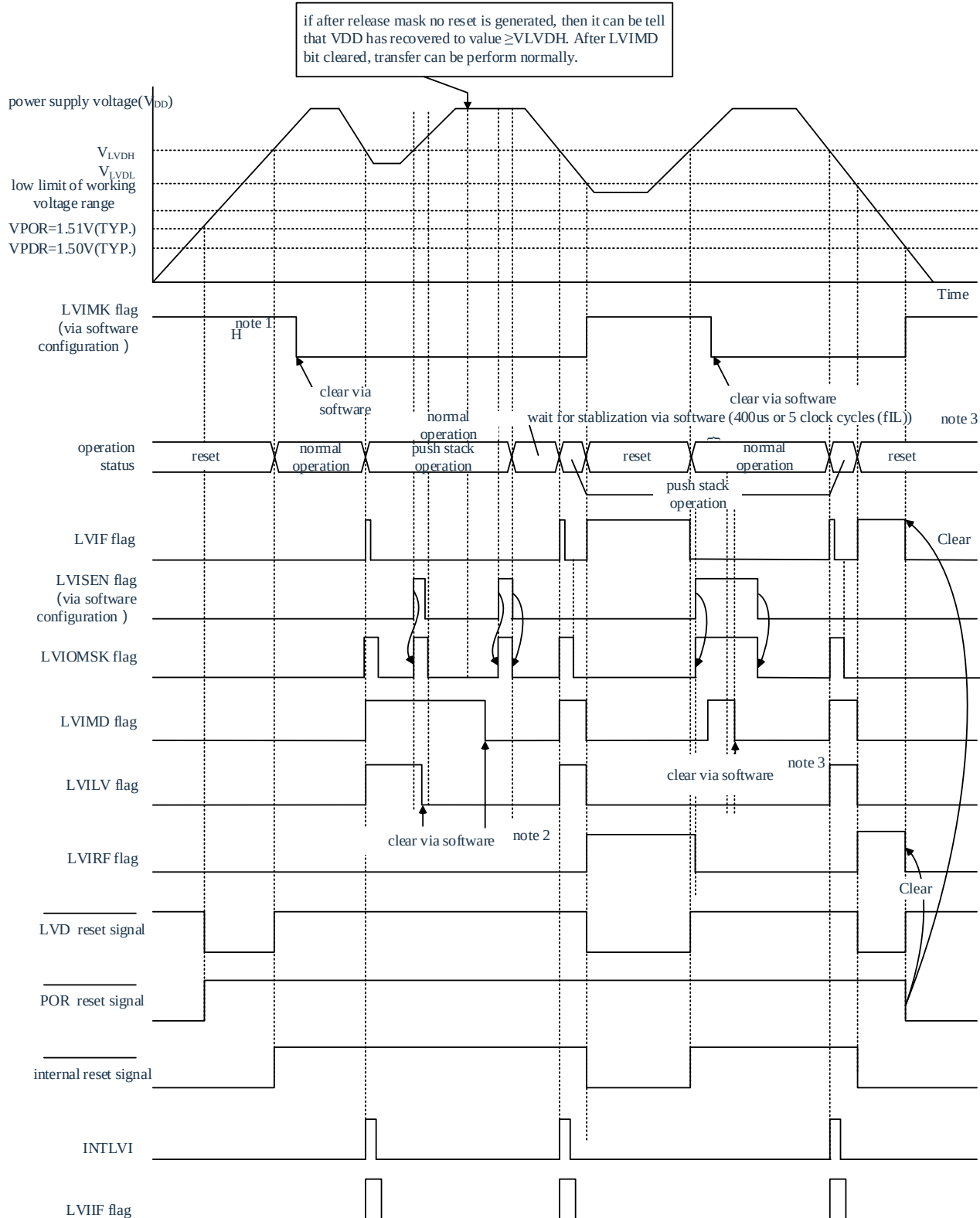
If the supply voltage (V_{DD}) exceeds the high voltage detection level (V_{LVDH}), the internal reset is released.

When the operating voltage drops, if the supply voltage (V_{DD}) 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 (V_{DD}) 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 (V_{DD}) 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 30-7: Setting procedure for confirmation/reset of operating voltage and Figure 22-8: 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 22-6.

Figure 22-6 Timing of reset & interrupt signal generation (LVIMDS1, LVIMDS0 of option bytes =1, 0) (1/2)

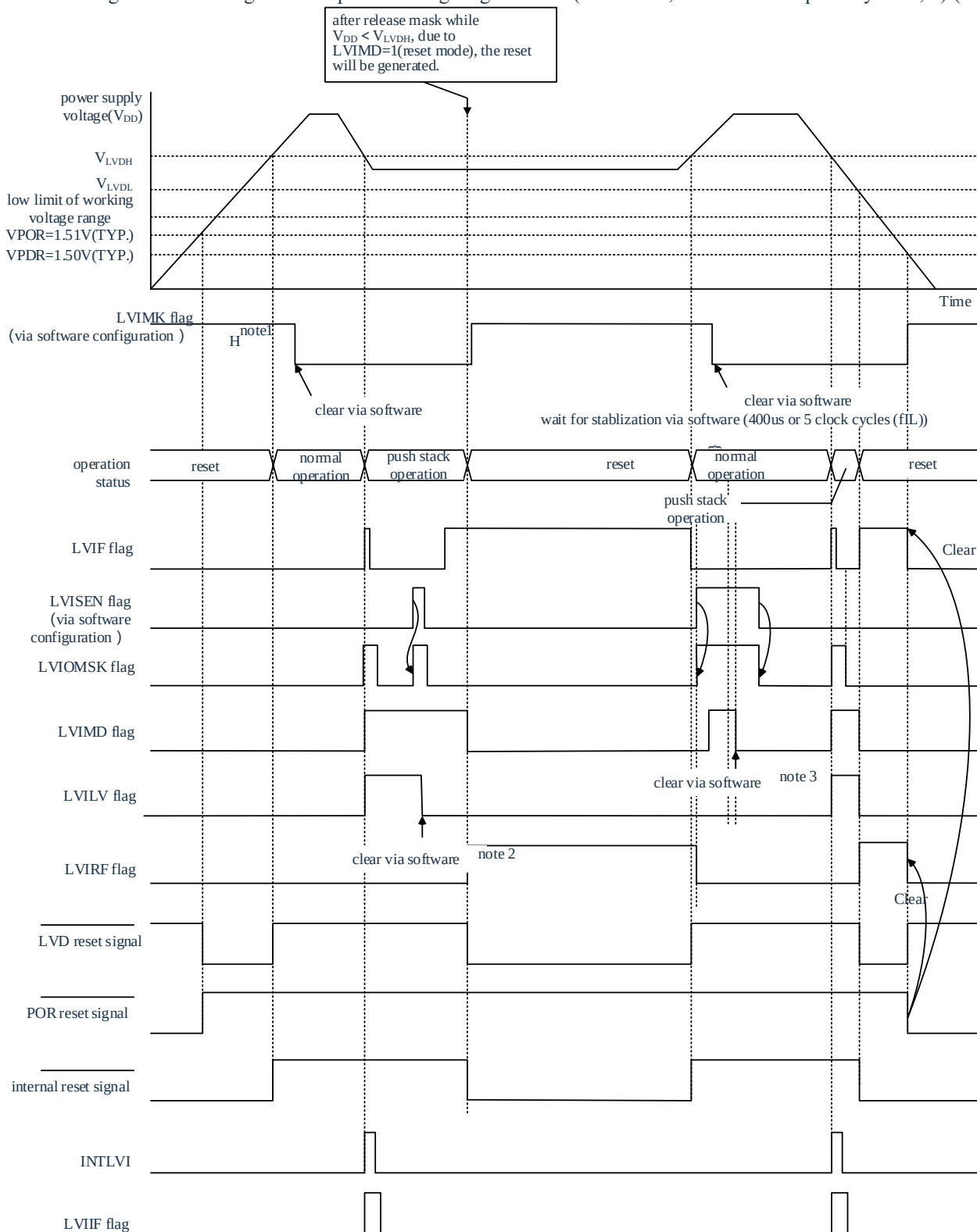


Notes:

1. After a reset signal is generated, the LVIMK flag becomes 1.
2. When using interrupt & reset mode, the settings must be made after an interrupt occurs according to Figure 22-7.
3. When in use & reset mode, the settings must be made according to Figure 22-8 after releasing the reset.

Remark V_{POR} : POR supply voltage rise detection voltage
 V_{PDR} : POR supply voltage drop detection voltage

Figure 22-6 Timing of interrupt & reset signal generation (LVIMDS1, LVIMDS0 of option bytes=1, 0) (2/2)



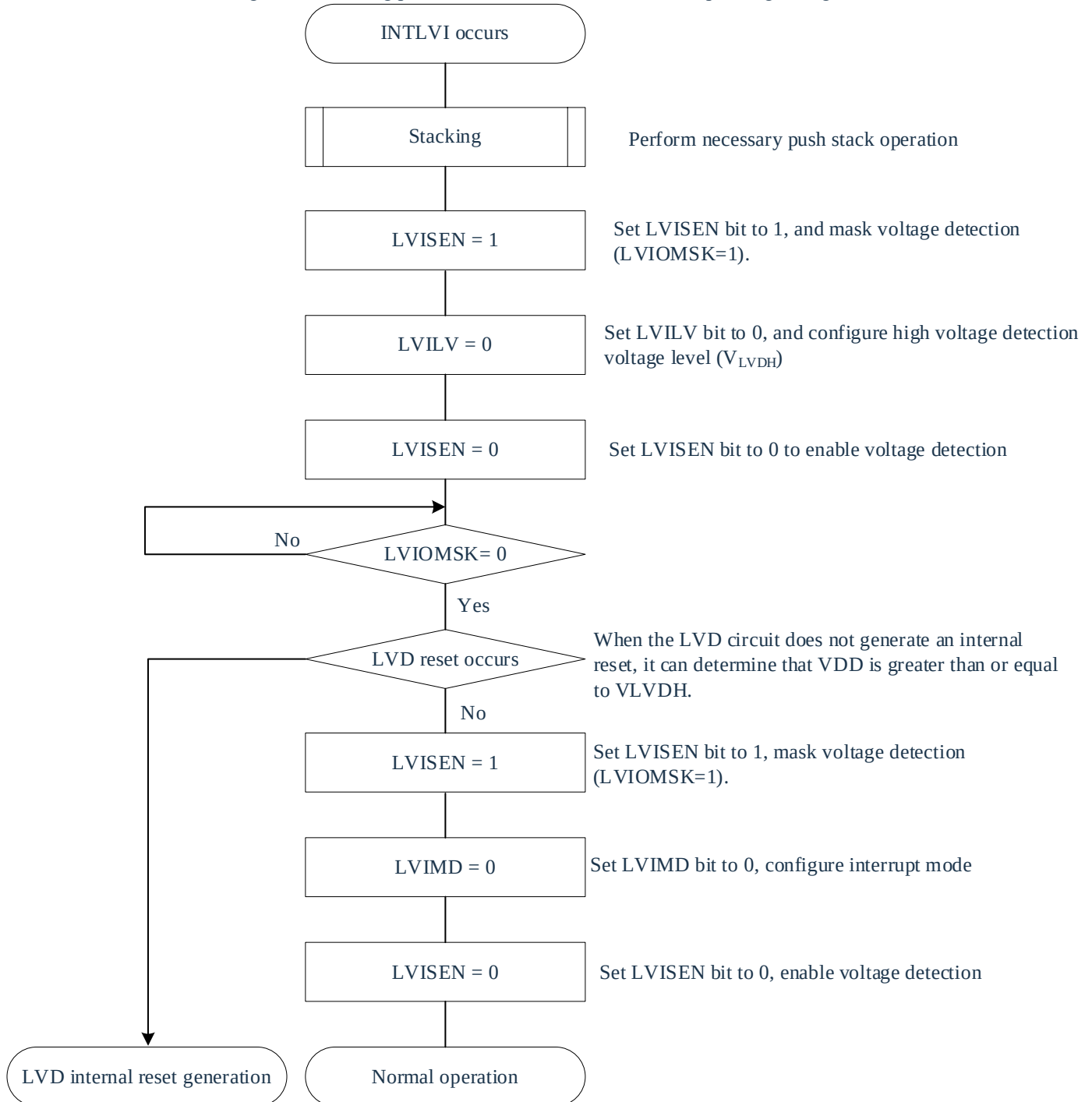
Notes:

1. After a reset signal is generated, the LVIMK flag becomes 1.
2. When using interrupt & reset mode, follow the steps in Figure 22-7 Setting procedure for confirmation/reset of operating voltage.
3. When using interrupt & reset mode, follow the steps in Figure 22-8: Initial setting procedure for interrupt & reset mode.

Remark: V_{POR} : POR power supply voltage rise detection voltage

V_{PDR} : POR power supply voltage drop detection voltage

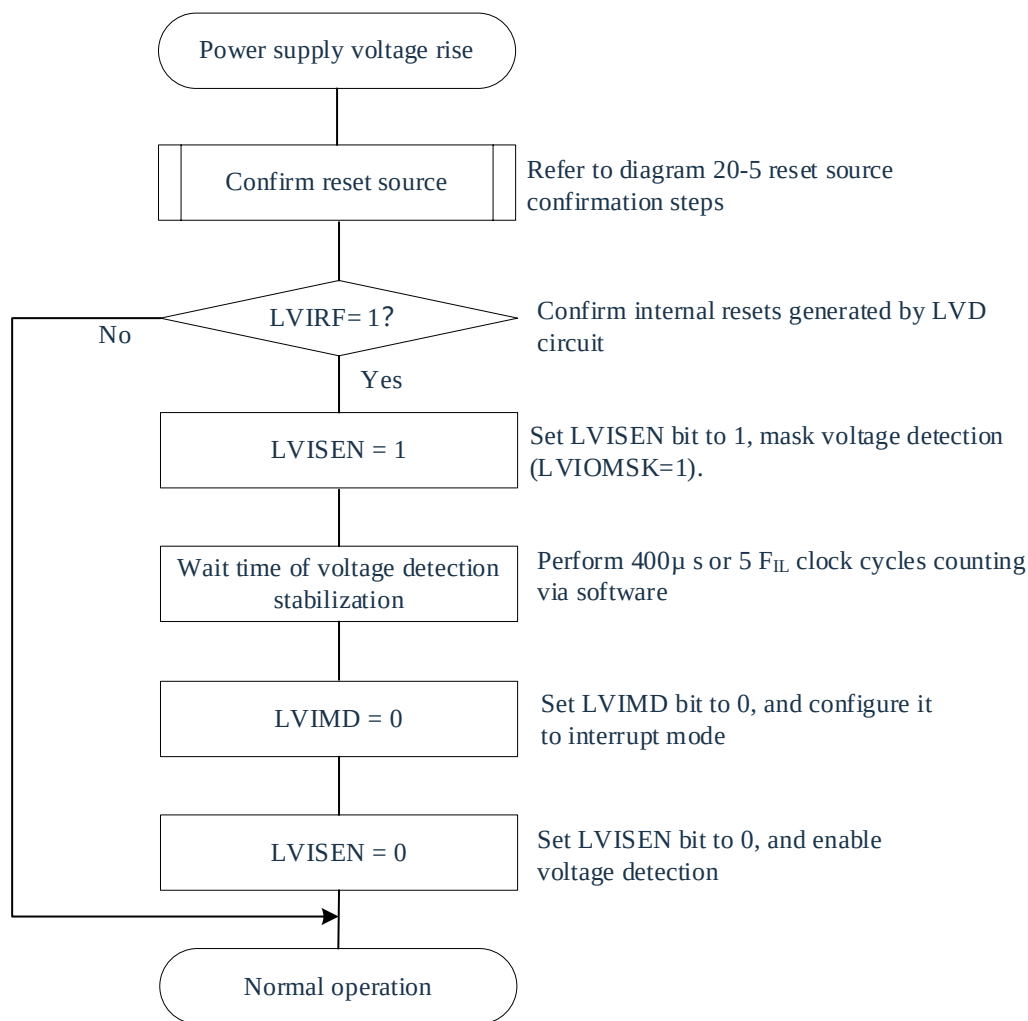
Figure 22-7 Setting procedure for confirmation/reset of operating voltage



If the interrupt & reset mode is set (LVIMDS1, LVIMDS0=1, 0), a voltage detection stabilization wait time of 400μs or 5 F_{IL} clocks is required after the LVD reset (LVIRF=1) is released. The LVIMD bit must be cleared to 0 for initialization after waiting for voltage detection to stabilize. During the counting of the voltage detection stabilization wait time and when rewriting the LVIMD bit, the LVISEN bit must be set to 1 to block the reset or interrupt generated by the LVD.

The initial setup procedure for the interrupt & reset mode is shown in Figure 22-8.

Figure 22-8 Initial setting procedure for interrupt & reset mode



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

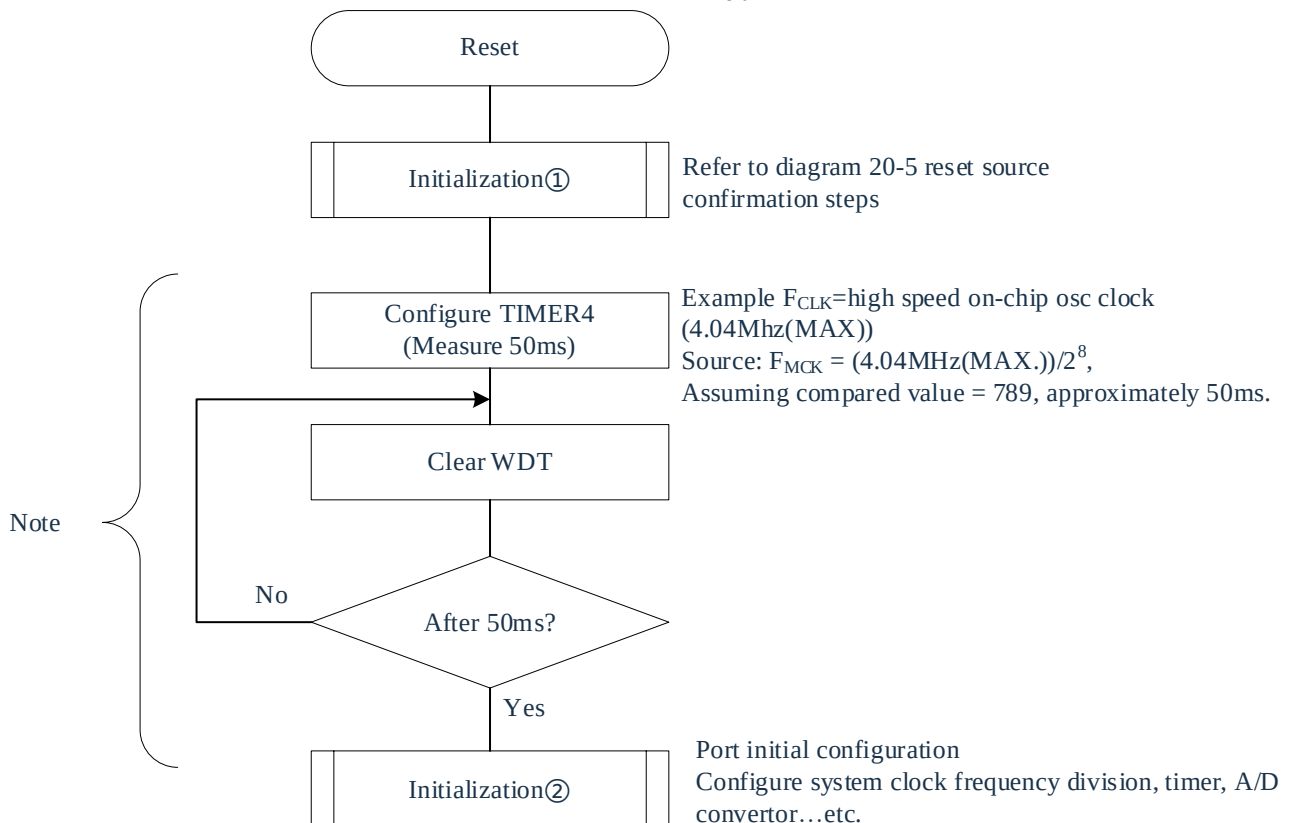
22.5 Cautions for Voltage Detection Circuit

(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.

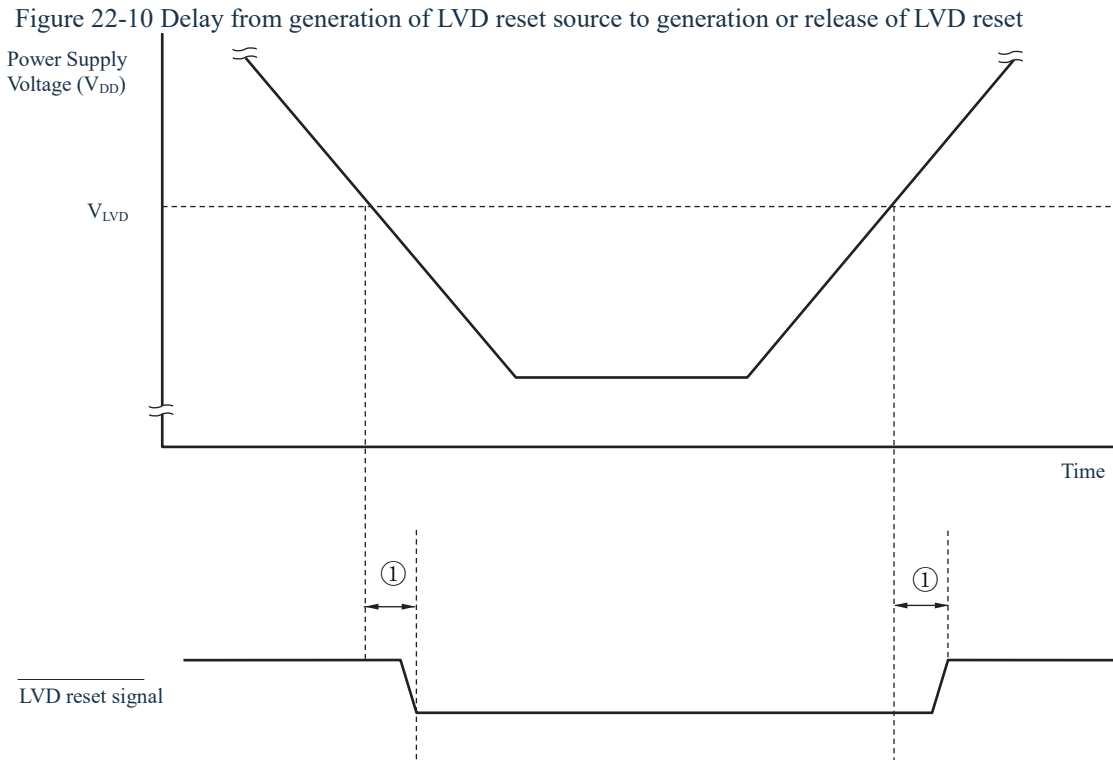
Figure 22-9 Example of software processing when the power supply voltage near the LVD detection voltage does not fluctuate for more than 50ms



Note 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 22-10).



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 23 Safety Functions

23.1 Overview of Safety Functions

In order to comply with IEC60730 and EC61508 safety standards, the BAT32G133B has the following built-in security features.

The purpose of this function is to safely stop the operation when a fault is detected through the self-diagnosis of the microcontroller.

(1) Flash CRC function (high-speed CRC, universal CRC)

The data error of flash memory is detected by CRC operation. The following two CRCs can be used depending on the application and usage conditions.

- “High-speed CRC”...During the initialization program, it is possible to stop the CPU and check the entire code flash area at high speed.
- “General-purpose CRC”...can be used for multi-purpose checks during CPU operation, not limited to the code flash area.

(2) RAM parity error detection function

Detects parity errors when reading RAM data.

(3) SFR guard function

Prevents rewriting of SFR due to uncontrolled CPU.

(4) Frequency detection function

Self-testing of CPU/peripheral hardware clock frequency using a general-purpose timer unit.

(5) A/D test function

A/D converter self-test by A/D conversion of positive (+) reference voltage, negative (-) reference voltage, analog input channel (ANI), temperature sensor output and internal reference voltage output of A/D converter.

(6) 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.

(7) Product unique ID register (128-bit)

23.2 Registers Used by Safety Functions

The following registers are used for the safety functions.

Register name	Function
• Flash CRC control register (CRC0CTL) • Flash CRC operation result register (PGCRCL)	Flash CRC operation function (High-speed CRC)
• CRC input register (CRCIN) • CRC data register (CRCD)	CRC arithmetic function (Universal CRC)
RAM parity error control register (RPECTL)	RAM parity error detection function
Special SFR guard control register (SFRGD)	SFR guard function
Timer input selection register 0 (TIS0)	Frequency detection 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

For the contents of each register, see the section “23.3 Operation of safety functions”.

23.3 Operation of Safety Functions

23.3.1 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 (e.g., 64KB flash: 512μs@32MHz).

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.

Remark: The operational result is different due to the universal CRC is LSB-preferred.

23.3.1.1 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”.

Figure 23-1 Format of flash CRC control register (CRC0CTL)

Address: 40021810H	After reset: 00H	R/W						
Symbol	7	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	0	0	0	0	FEA1	FEA0

CRC0EN	Control of high-speed CRC operation
0	Stop the operation.
1	Start the operation by executing the WFE instruction.

FEA1	FEA0	High-speed CRC operation range
0	0	00000H ~ 1FFBH(8K-4byte)
0	1	00000H ~ 3FFBH(16K-4byte)
1	0	00000H ~ 5FFBH(24K-4byte)
1	1	00000H ~ 7FFBH(32K-4byte)

Note: bit2-6 should be set to 0.

Remark: 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.

23.3.1.2 Flash CRC Operation Result Register (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”.

Figure 23-2 Format of flash memory CRC operation result register (PGCRCL)

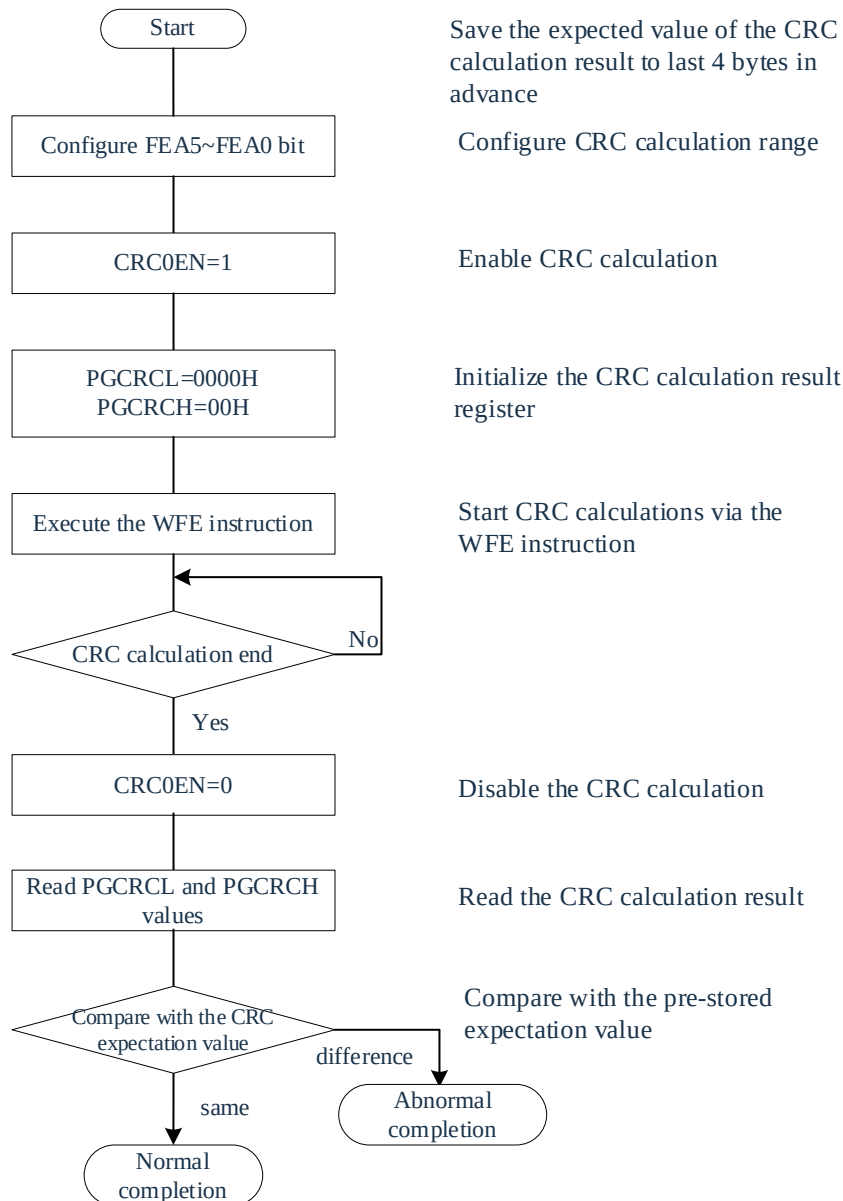
Address:	0x40021812		After reset:		0000H		R/W									
Symbol	15		14		13		12		11		10		9		8	
PGCRCL	PGCRC15		PGCRC14		PGCRC13		PGCRC12		PGCRC11		PGCRC10		PGCRC9		PGCRC8	
	7		6		5		4		3		2		1		0	
	PGCRC7		PGCRC6		PGCRC5		PGCRC4		PGCRC3		PGCRC2		PGCRC1		PGCRC0	
	PGCRC15~0				High-speed CRC operation results											
	0000H~FFFFH				Store the high-speed CRC operation results.											

Caution: 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 23-3.

<Operation flow>

Figure 23-3 Flowchart for flash memory CRC operation function (high-speed CRC)



Cautions: 1. Only code flash is the object of CRC operation.

2. The expected value of CRC operation must be stored in the area after the operation range in the code flash memory.

3. CRC cannot be calculated when the debugger is connected.

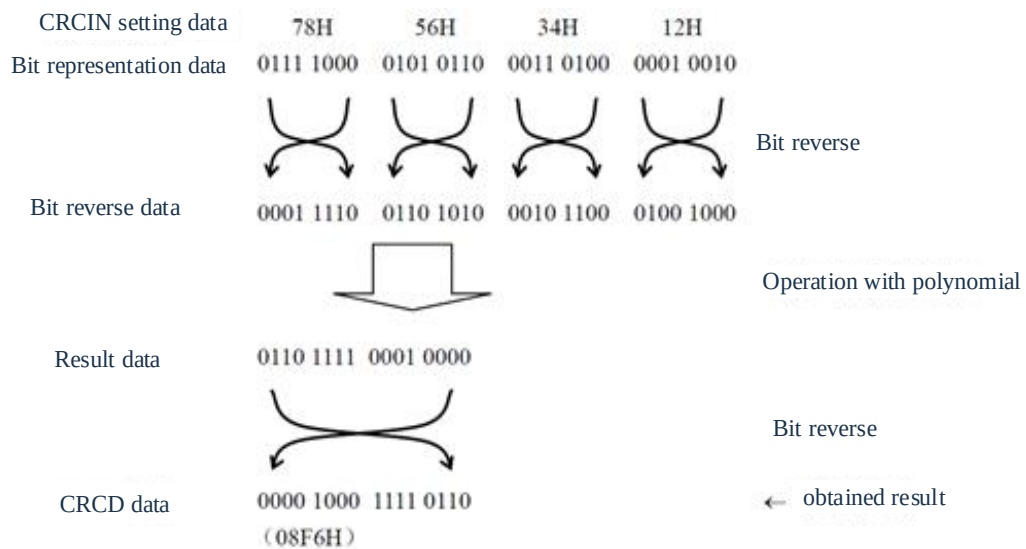
23.3.2 CRC Operation Function (General-Purpose CRC)

In order to guarantee safety during operation, the IEC61508 standard mandates the checking of data even while the CPU is operating.

A general CRC operation can be executed as a peripheral function while the CPU is operating. The general CRC can be used for checking various data in addition to the code flash memory area. The data to be checked can be specified by using software (a user-created program). CRC calculation function in the sleep mode can be used only during the DMA transmission.

The general CRC operation can be executed in the main system clock operation mode as well as the subsystem clock operation mode.

The CRC generator polynomial used is “ $X^{16} + X^{12} + X^5 + 1$ ” of CRC-16-CCITT. The data to be input is inverted in bit order and then calculated to allow for LSB-first communication. For example, if the data 12345678H is sent from the LSB, values are written to the CRCIN register in the order of 78H, 56H, 34H, and 12H, enabling a value of 08F6H to be obtained from the CRCD register. This is the result obtained by executing a CRC operation on the bit rows shown below, which consist of the data 12345678H inverted in bit order.



Caution: Because the debugger rewrites the software break setting line to a break instruction during program execution, the CRC operation result differs if a software break is set in the CRC operation target area.

23.3.2.1 CRC Input Register (CRCIN)

The CRCIN register is an 8-bit register that is used to set the CRC operation data of general-purpose CRC. The available set range is 00H~ FFh.

The CRCIN register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register changes to “00H”.

Figure 23-4 Format of CRC input register (CRCIN)

Figure 25-4 Format of CRC input register (CRCIN)

Address: 400433ACH	After reset: 00H		R/W					
Symbol	7	6	5	4	3	2	1	0
CRCIN								
Bit7~0		Function						
00H~FFH		Data input						

23.3.2.2 CRC Data Register (CRCD)

This is a register that holds the results of the general-purpose CRC operation. The range that can be set is “0000H~FFFFH”.

After writing the CRCIN register, it goes through a CPU/peripheral hardware clock (F_{CLK}) to save the CRC operation result to the CRCD register.

The CRCD register is set by a 16-bit memory manipulation instruction.

After a reset signal is generated, the value of this register becomes “0000H”.

Figure 23-5 Format of CRC data register (CRCD)

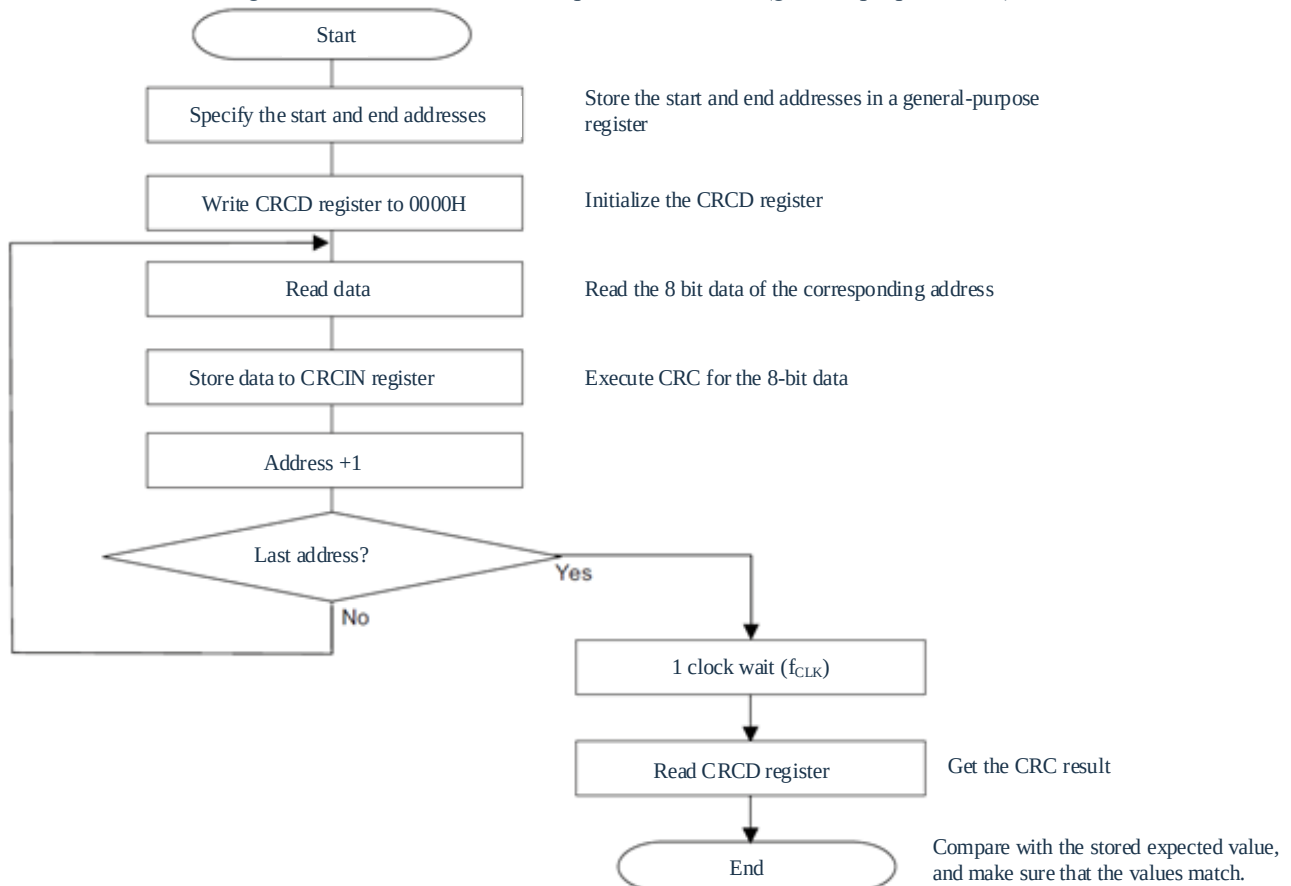
Address: 400432FAH		After reset: 0000H		R/W												
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

Cautions:

1. Read the value written to CRCD register before writing to CRCIN register.
2. If conflict between writing and storing operation result to CRCD register occurs, the writing is ignored.

<Operation flow>

Figure 23-6 Flowchart of CRC operation function (general-purpose CRC)



23.3.3 RAM Parity Error Detection Function

The IEC60730 standard mandates the checking of RAM data. A single-bit parity bit is therefore added to all 8-bit data in the BAT32G133B's RAM. By using this RAM parity error detection function, the parity bit is appended when data is written, and the parity is checked when the data is read. This function can also be used to trigger a reset when a parity error occurs.

23.3.3.1 RAM Parity Error Control Register (RPECTL)

This register is used to control parity error generation check bit and reset generation due to parity errors.

The RPECTL 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".

Figure 23-7 Format of RAM parity error control register (RPECTL)

Address: 40020425H	After reset: 00H	R/W						
Symbol	7	6	5	4	3	2	1	0
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

RPERDIS	Parity error reset mask flag
0	Enable parity error resets.
1	Disable parity error resets.

RPEF	Parity error status flag
0	No parity errors occurred.
1	A parity error occurred.

Caution: The parity bit is appended when the data is written and checked when the data is read.

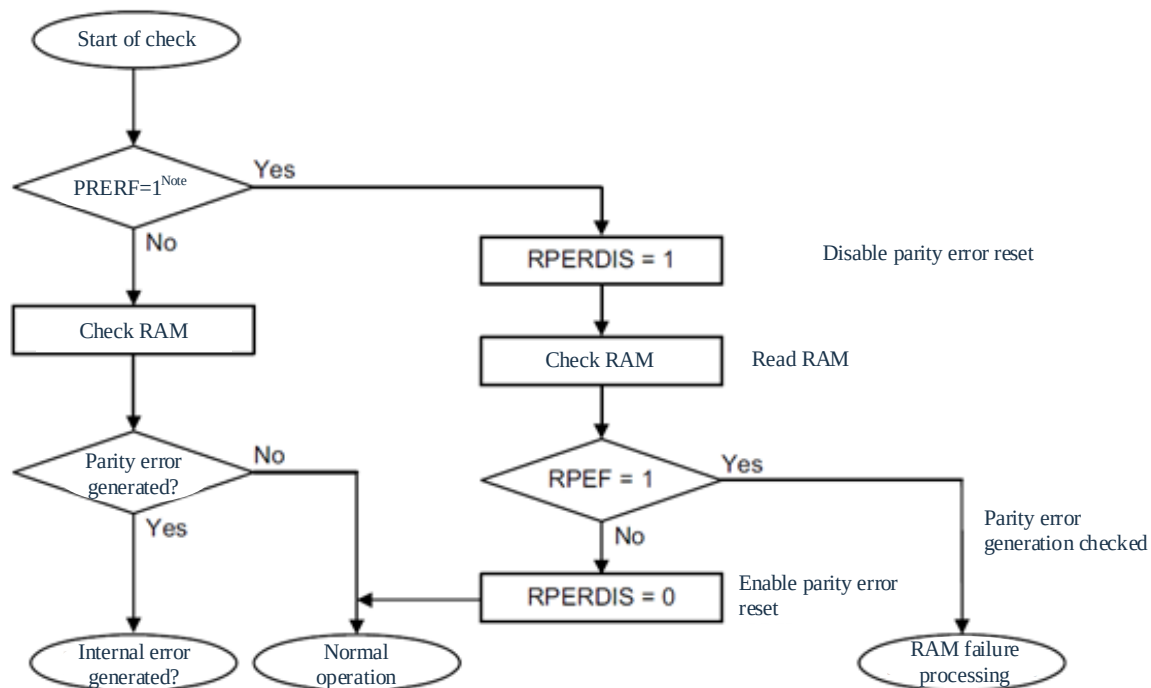
Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed before reading data.

The CPU executes look-ahead due to the pipeline operation, the CPU might read an uninitialized RAM area that is allocated beyond the RAM used, which causes a RAM parity error. Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the RAM area + 10 bytes when instructions are fetched from RAM areas.

Remarks:

1. The parity error reset is enabled by default (RPERDIS = 0).
2. Even if it is set to prohibit generation of parity check error reset (RPERDIS=1), the RPEF flag is set to 1 when a parity check error occurs. If the RPEF bit is set to allow generation of parity check error reset (RPERDIS=0) in the state where the RPEF bit is 1, parity check error reset is generated when RPERDIS is cleared to 0.
3. Even if the parity error reset is disabled (RPERDIS = 1), the RPEF flag will be set (1) if a parity error occurs. If parity error resets are enabled (RPERDIS = 0) with RPEF set to 1, a parity error reset is generated when the RPERDIS bit is cleared to 0.
4. The general registers are not included for RAM parity error detection.

Figure 23-8 Flow of RAM parity check



Note To check internal reset status using a RAM parity error, see Chapter 20 Reset Function.

23.3.4 SFR Guard Function

In order to guarantee safety during operation, the IEC61508 standard mandates that important data stored in the SFR be protected, even if the CPU freezes. The SFR protection function is used to protect data of control registers of comparator function, port function, interrupt function, clock control function, voltage detection circuit and RAM parity error detection function.

If the SFR guard function is specified, writing to the specified SFRs is disabled, but reading from the SFRs can be carried out as usual.

23.3.4.1 SFR Guard Control Register (SFRGD)

This register is used to control the detection of invalid memory access and SFR guard function.

GCOMP bits, GPORT bits, GINT bits, and GCSC bits 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”.

Figure 23-9 Format of SFR protection control register (SFRGD)

Address: 40040478H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SFRGD	0	0	0	0	GCOMP	GPORT	GINT	GCSC
GCOMP		Control registers of comparator function						
0		Disabled. Control registers of comparator function can be read and written to.						
1		Enabled. Writing to control registers of comparator function is disabled. Reading is enabled. [Guarded SFR] COMPPMDR, COMPFIR, COMPOCR, CVRCTL, CxRVM, PGAxCTL, PGASHMD, CMPSELx						
GPORT		Control registers of port function						
0		Disabled. Control registers of port function can be read and written to.						
1		Enabled. Writing to control registers of port function is disabled. Reading is enabled. [Guarded SFR]PMxx, PUxx, PDxx, POMxx, PMCxx, PxxCFG, PIORx ^{Note}						
GINT		Control registers of interrupt function						
0		Disabled. Control registers of interrupt function can be read and written to.						
1		Enabled. Writing to control registers of interrupt function is disabled. Reading is enabled. [Guarded SFR]JFxx, MKxx, PRxx, EGPx, EGNx						
GCSC		Control registers of clock control function, voltage detection circuit and RAM parity error detection function						
0		Disabled. Control registers of clock control function, voltage detection circuit and RAM parity error detection function can be read and written to.						
1		Enabled. Writing to control registers of clock control function, voltage detection circuit and RAM parity error detection function is disabled. Reading is enabled. [Guarded SFR]CMC, CSC, OSTS, CKC, PERx, OSMC, LVIM, LVIS, RPECTL.						

Note: Pxx (port register) is not guarded.

23.3.5 Frequency Detection Function

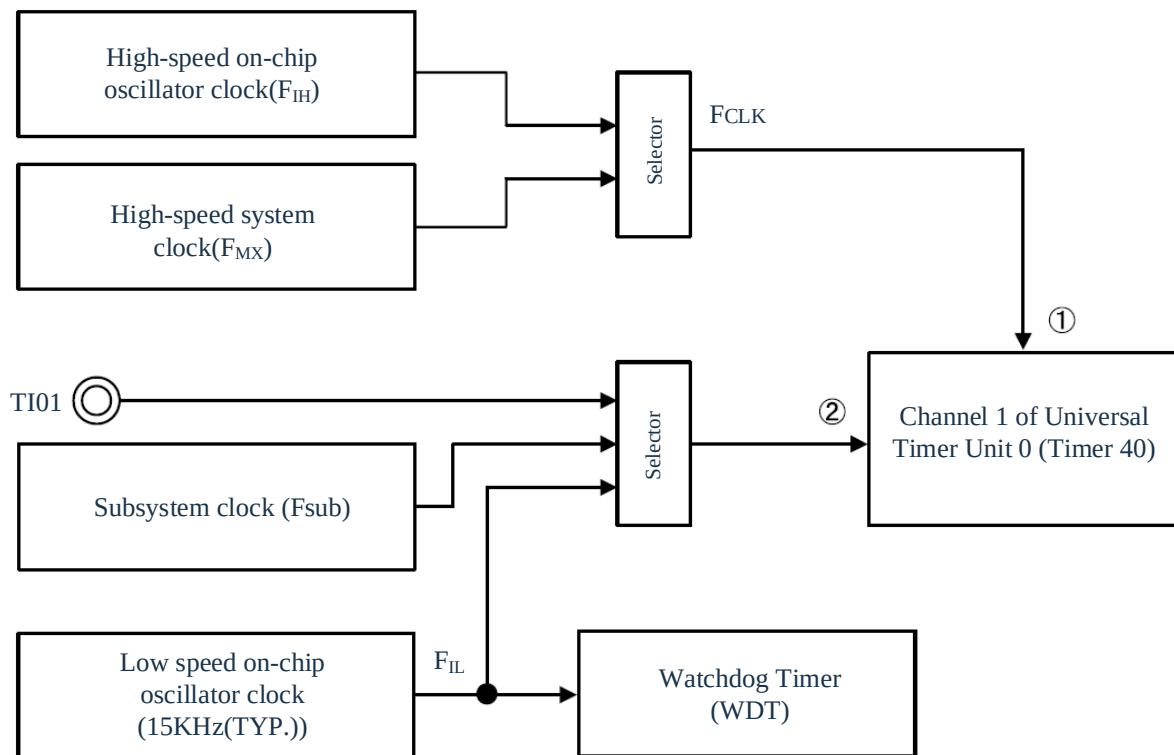
The IEC60730 standard mandates checking that the oscillation frequency is correct.

By using the CPU/peripheral hardware clock frequency (f_{CLK}) and measuring the pulse width of the input signal to channel 1 of the timer40, whether the proportional relationship between the two clock frequencies is correct can be determined. Note that, however, if one or both clock operations are completely stopped, the proportional relationship between the clocks cannot be determined.

<Clocks to compare>

- (1) CPU/peripheral hardware clock frequency (f_{CLK}):
 - High-speed on-chip oscillator clock (f_{IH})
 - High-speed system clock (f_{MX})
- (2) Timer40 channel 1 input:
 - Channel 1 Timer input (TI01)
 - Low-speed on-chip oscillator clock (f_{IL} :15kHz (TYP.))
 - Subsystem clock (f_{SUB})^{Note}

Figure 23-10 Structure of frequency detection function



If input pulse interval measurement results in an abnormal value, it can be concluded that the clock frequency is abnormal. For how to execute input pulse interval measurement, see 5.8.4 Operation as input pulse interval measurement.

Note: Can only be selected in the products incorporating the subsystem clock.

23.3.5.1 Timer Input Select Register 0 (TIS0)

For the register description, refer to Section 5.3.8.

23.3.6 A/D Test Function

The IEC60730 standard mandates testing the A/D converter. The A/D test function checks whether or not the A/D converter is operating normally by executing A/D conversions of the A/D converter's positive and negative reference voltages, analog input channel (ANI), temperature sensor output voltage, and the internal reference voltage.

The analog multiplexer can be checked using the following procedure.

- ① Select the ANIx pin for A/D conversion using the ADTES register (ADTES2 = 0, ADTES1 = 0, ADTES0 = 0).
- ② Perform A/D conversion for the ANIx pin (conversion result 1-1).
- ③ Select the A/D converter's negative reference voltage for A/D conversion using the ADTES register (ADTES2 = 0, ADTES1 = 0, ADTES0 = 1)
- ④ Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).
- ⑤ Select the ANIx pin for A/D conversion using the ADTES register (ADTES2 = 0, ADTES1 = 0, ADTES0 = 0).
- ⑥ Perform A/D conversion for the ANIx pin (conversion result 1-2).
- ⑦ Select the A/D converter's positive reference voltage for A/D conversion using the ADTES register (ADTES2 = 1, ADTES1 = 0, ADTES0 = 1)
- ⑧ Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).
- ⑨ Select the ANIx pin for A/D conversion using the ADTES register (ADTES2 = 0, ADTES1 = 0, ADTES0 = 0).
- ⑩ Perform A/D conversion for the ANIx pin (conversion result 1-3).
- ⑪ Check that the conversion results 1-1, 1-2, and 1-3 are equal.
- ⑫ Check that the A/D conversion result 2-1 is all zero and conversion result 2-2 is all one. Using the procedure above can confirm that the analog multiplexer is selected and all wiring is connected.

Remarks:

1. If the analog input voltage is variable during A/D conversion in steps ① to ⑩ above, use another method to check the analog multiplexer.
2. The conversion results might contain an error. Consider an appropriate level of error when comparing the conversion results.

23.3.6.1 A/D Test Register (ADTES)

This register is used to select the A/D converter's positive reference voltage, A/D converter's negative reference voltage, analog input channel (ANLxx), temperature sensor output voltage, or internal reference voltage (1.45 V) as the target of A/D conversion.

When using the A/D test function, specify the following settings:

- Select negative reference voltage as the target of A/D conversion for zero-scale measurement.
- Select positive reference voltage as the target of A/D conversion for full-scale measurement.

For the register description, refer to 10.2.10.

23.3.6.2 Analog Input Channel Specification Register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

Set A/D test register (ADTES) to 00H when measuring the ANLxx/temperature sensor output voltage/internal reference voltage (1.45 V).

For the register description, refer to 10.2.13.

23.3.7 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.

23.3.7.1 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".

Figure 23-11 Format of port mode selection register (PMS)

Address: 4004087BH

After reset: 00H

R/W

Symbol	7	6	5	4	3	2	1	0
PMS	0	0	0	0	0	0	0	PMS0

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.

Caution: For pins that are forced into a high-impedance state due to the pulse output forced truncation function of timer M, if the digital output level of the pin is read, the read value will be 0.

Remark: m=0~3

n=0~7

23.3.8 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.

Base address: 0x0050_0848

Address offset: 0x00

Read-only, its value is written at factory.

U_ID[31:0]

Address offset: 0x04

Read-only, its value is written at factory.

U_ID[63:32]

Address offset: 0x08

Read-only, its value is written at factory.

U_ID[95:64]

Address offset: 0x0C

Read-only, its value is written at factory.

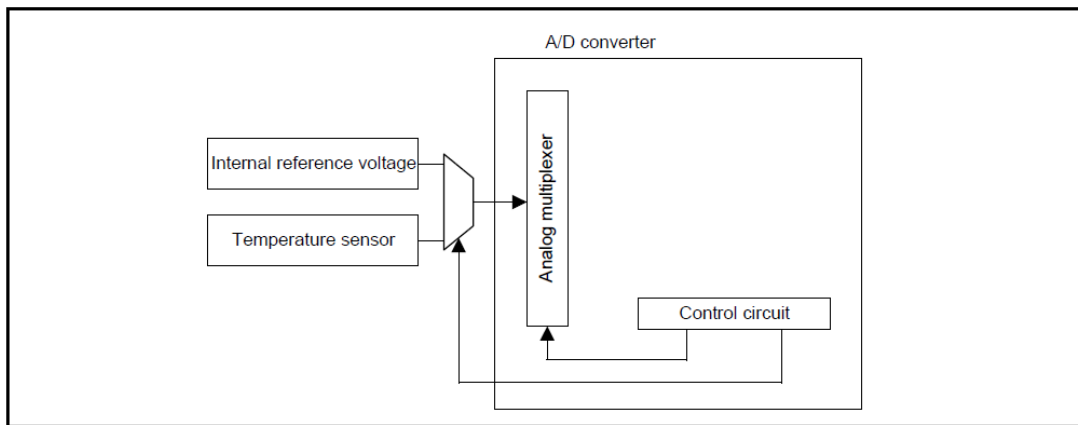
Chapter 24 Temperature Sensor

24.1 Functions of Temperature Sensors

The on-chip temperature sensor measures and monitors the core temperature of the product, ensuring its reliable operation. The output voltage of the temperature sensor is proportional to the core temperature, and there is a linear relationship between the voltage and temperature. The output voltage is provided to the ADC for conversion.

Figure 24-1 shows the block diagram of the temperature sensor.

Figure 24-1 Block diagram of temperature sensor



24.2 Registers Used by Temperature Sensors

24.2.1 Temperature Sensor Calibration Data Register TSCDR1

Address: 0x4002_1C38

Symbol	7	6	5	4	3	2	1	0	After reset	R/W
TSCDR1	TSCDR1[7:0]								-	R

A read-only register for recording the calibration data 1 of the temperature sensor, automatically loaded when the power is turned on or reset is started, each chip having its own calibration data.

24.2.2 Temperature Sensor Calibration Data Register TSCDR2

Address: 0x4002_1C3C

Symbol	7	6	5	4	3	2	1	0	After reset	R/W
TSCDR2	TSCDR2[7:0]								-	R

A read-only register for recording calibration data 2 of the temperature sensor, automatically loaded when the power supply is turned on or reset is started, each chip having its own calibration data.

24.3 Instructions for Using Temperature Sensors

24.3.1 How to Use the Temperature Sensor

The temperature (T) is proportional to the sensor voltage output (Vs), so the temperature is calculated as follows:

$$T = (V_s - V_1) / \text{slope} + T_1$$

T: Measured temperature (°C)

Vs: Output voltage (V) of temperature sensor in temperature measurement

T1: Temperature measured experimentally at the first point (°C)

V1: Voltage output (V) when the temperature sensor measures T1

T2: Temperature measured experimentally at the second point (°C)

V2: Voltage output (V) when the temperature sensor measures T2

Slope: Temperature slope of the temperature sensor (V/°C), $\text{slope} = (V_2 - V_1) / (T_2 - T_1)$.

The characteristics of different sensors are different, so we suggest to measure two different sample temperatures:

1. The voltage V1 output by the temperature sensor at temperature T1 is measured using an A/D converter.
2. The voltage V2 output by the temperature sensor at the second temperature T2 is measured using an A/D converter.
3. The slope of temperature was calculated from both results ($\text{slope} = (V_2 - V_1) / (T_2 - T_1)$)
4. Subsequently, the temperature ($T = (V_s - V_1) / \text{slope} + T_1$) is obtained by substituting the slope into the formula of the temperature characteristic.

24.3.2 Methods for Using the Temperature Sensor

Method 1: In this product, the TSCDR1 register stores a voltage conversion value (CAL25) of the temperature sensor measured at $T_a=T_j=25^{\circ}\text{C}$ and $V_{DD}=3.0\text{v}$. The TSCDR2 register stores the voltage conversion value (CAL85) of the temperature sensor measured at $T_a=T_j=85^{\circ}\text{C}$ and $V_{DD}=3.0\text{v}$. Using these two sets of values, the temperature slope can be calculated:

$$\text{slope} = (V_2 - V_1) / (85 - 25).$$

$$V_1 = 3.0 \times \text{CAL25} / 256 \text{ [V]}$$

$$V_2 = 3.0 \times \text{CAL85} / 256 \text{ [V]}$$

Using the above results, the temperature can be calculated according to the following formula:

$$T = (V_s - V_1) / \text{slope} + 25 \text{ [}^{\circ}\text{C]}$$

T: Measured temperature ($^{\circ}\text{C}$)

V_s : Output voltage (V) at T temperature of a temperature sensor obtained using an A/D converter

Method 2: If you use the temperature slope given in Electrical Characteristics, you can calculate the measured temperature directly using the following formula:

$$T = (V_s - V_1) / \text{slope} + 25 \text{ [}^{\circ}\text{C]}$$

Note: The temperature generated by this method is lower than the accuracy measured by the method 1.

Chapter 25 Option Byte

25.1 Functions of Option Bytes

Addresses 000C0H~000C3H, 500004H~50000BH of the flash memory of the BAT32G133B form an option byte area.

The option byte consists of the user option byte (000C0H~000C2H) and the flash memory data protection option byte (000C3H, 500004H~50000BH). Upon power-on or reset startup, the option byte is automatically referenced to configure the specified functions. When using this product, the following functions must be configured through the option byte. For bits that do not have a configured function, the initial values cannot be changed. To use the boot swap function during self-programming, since 000C0H~000C3H is replaced by 010C0H~010C3H, the values of 010C0H~010C3H must be set to the same as those of 000C0H~000C3H.

Caution: The option bytes should always be set regardless of whether each function is used.

25.1.1 User Option Byte (000C0H~000C2H/010C0H~010C2H)

(1) 000C0H/010C0H

- Setting of watchdog timer operation
 - Enabling or disabling of counter operation
 - Enabling or disabling of counter operation in the sleep/deep sleep mode.
- Watchdog timer overflow time setting
- Watchdog timer window open period setting
- Watchdog timer interval interrupt setting
 - Use or do not use interval interrupts.

Caution: During boot swap, 000C0H is replaced by 010C0H, so the value of 010C0H must be set to the same as 000C0H.

(2) 000C1H/010C1H

- Setting of LVD operation mode
 - Interrupt & reset mode
 - Reset mode
 - Interrupt mode
 - LVD off (by controlling the externally input reset signal on the RESETB pin)
- Setting of LVD detection level (V_{LVDH} , V_{LVDL} , V_{LVD})

Cautions: 1. When the power supply voltage rises, the reset state must be maintained through a voltage detection circuit or external reset before the power supply voltage reaches the operating voltage range specified in the data sheet's AC characteristics. When the power supply voltage drops, the reset state must be set through a deep sleep mode transition, voltage detection circuit, or external reset before the power supply voltage falls below the operating voltage range.

The operating voltage range depends on the settings of the user option byte (000C2H/010C2H).

2. During boot swap, 000C1H is replaced by 010C1H, so the value of 010C1H must be set to the same as 000C1H.

(3) 000C2H/010C2H

- Setting of high-speed on-chip oscillator frequency
 - Select from 1MHz~32MHz, 48MHz, and 64MHz.

Caution: During boot swap, 000C2H is replaced by 010C2H, so the value of 010C2H must be set to the same as 000C2H.

25.1.2 Flash Memory Data Protection Option Byte (000C3H/010C3H, 500004H~50000BH)

- Control of on-chip debug operation
 - Level0: Enabling read/write/erase operations on flash 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.
- Control of boot swap function
 - Enable or disable the use of the boot swap function.

Caution: During boot swap, 000C3H is replaced by 010C3H, so the value of 010C3H must be set to the same as 000C3H.

25.2 Format of User Option Byte

Figure 25-1 Format of user option byte (000C0H/010C0H)

Address: 000C0H/010C0H ^{Note 1}

Symbol	7	6	5	4	3	2	1	0
	WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	Use of interval interrupt of watchdog timer	
0	Interval interrupt is not used.	
1	Interval interrupt is generated when 75% of the overflow time + 1/2 f_{IL} is reached.	

WINDOW1	WINDOW0	Watchdog timer window open period ^{Note2}
0	-	Setting prohibited
1	0	75%
1	1	100%

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time ($f_{IL}=40\text{kHz}(\text{MAX.})$)
0	0	0	$2^6/f_{IL}$ (1.6ms)
0	0	1	$2^7/f_{IL}$ (3.2ms)
0	1	0	$2^8/f_{IL}$ (6.4ms)
0	1	1	$2^9/f_{IL}$ (12.8ms)
1	0	0	$2^{11}/f_{IL}$ (51.2ms)
1	0	1	$2^{13}/f_{IL}$ (204.8ms)
1	1	0	$2^{14}/f_{IL}$ (409.6ms)
1	1	1	$2^{16}/f_{IL}$ (1638.4ms)

WDSTBYON	Operation control for watchdog timer counter (sleep mode)
0	Counter operation stopped in sleep mode ^{Note 2} .
1	Counter operation enabled in sleep mode.

Notes: 1. During boot swap, 000C0H is replaced by 010C0H, so the value of 010C0H must be set to the same as 000C0H.

2: When the WDSTBYON bit is 0, regardless of the values of WINDOW1 and WINDOW0 bits, the window is 100% open during the window period.

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

Figure 25-2 Format of user option byte (000C1H/010C1H) (1/4)

Address: 000C1H/010C1H ^{Note}

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt & reset mode)

Detection voltage			Setting value of option byte						
VLVDH		VLVDL	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
rise	drop	drop						LVIMDS1	LVIMDS0
1.98V	1.94V	1.84V	0	0	1	1	0	1	0
2.09V	2.04V					0	1		
3.13V	3.06V					0	0		
2.61V	2.55V	2.45V		1	0	1	0		
2.71V	2.65V					0	1		
3.75V	3.67V					0	0		
2.92V	2.86V	2.75V		1	1	1	0		
3.02V	2.96V					0	1		
4.06V	3.98V					0	0		
—			Setting values other than those mentioned above is prohibited.						

Note: During boot swap, 000C1H is replaced by 010C1H, so the value of 010C1H must be set to the same as 000C1H.

Caution: Set the bit4 to 1.

Remarks: 1. For details of the LVD circuit, refer to “Chapter 22 Voltage Detection Circuit”.

2. The detection voltage is the TYP. value. For details, refer to LVD Circuit Characteristics in the datasheet.

Figure 25-3 Format of user option byte (000C1H/010C1H) (2/4)

Address: 000C1H/010C1H^{Note}

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD setting (reset mode)

Detection voltage		Setting value of option byte						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
rise	drop						LVIMDS1	LVIMDS0
1.88V	1.84V	0	0	1	1	1	1	1
1.98V	1.94V		0	1	1	0		
2.09V	2.04V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.61V	2.55V		1	0	1	0		
2.71V	2.65V		1	0	0	1		
2.81V	2.75V		1	1	1	1		
2.92V	2.86V		1	1	1	0		
3.02V	2.96V		1	1	0	1		
3.13V	3.06V		0	1	0	0		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—		Setting values other than those mentioned above is prohibited.						

Note: During boot swap, 000C1H is replaced by 010C1H, so the value of 010C1H must be set to the same as 000C1H.

Caution: Set the bit4 to 1.

Remark: 1. For details of the LVD circuit, refer to “Chapter 22 Voltage Detection Circuit”

2. The detection voltage is the TYP. value. For details, refer to LVD Circuit Characteristics in the datasheet.

Figure 25-4 Format of user option byte (000C1H/010C1H) (3/4)

Address: 000C1H/010C1H^{Note}

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD setting (interrupt mode)

Detection voltage		Setting value of option byte						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
rise	drop						LVIMDS1	LVIMDS0
1.88V	1.84V	0	0	1	1	1	0	1
1.98V	1.94V		0	1	1	0		
2.09V	2.04V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.61V	2.55V		1	0	1	0		
2.71V	2.65V		1	0	0	1		
2.81V	2.75V		1	1	1	1		
2.92V	2.86V		1	1	1	0		
3.02V	2.96V		1	1	0	1		
3.13V	3.06V		0	1	0	0		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—		Setting values other than those mentioned above is prohibited.						

Note: During boot swap, 000C1H is replaced by 010C1H, so the value of 010C1H must be set to the same as 000C1H.

Caution: Set the bit4 to 1.

Remarks: 1. For details of the LVD circuit, refer to “Chapter 22 Voltage Detection Circuit”.

2. The detection voltage is the TYP. value. For details, refer to LVD Circuit Characteristics in the datasheet.

Figure 25-5 Format of user option byte (000C1H/010C1H) (4/4)

Address: 000C1H/010C1H ^{Note}

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- 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.						

Note: During boot swap, 000C1H is replaced by 010C1H, so the value of 010C1H must be set to the same as 000C1H.

Cautions:

- Set the bit4 to 1.
- 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/010C2H).

Remarks:

- 1.×: Ignore
- For details of the LVD circuit, refer to “Chapter 22 Voltage Detection Circuit”.
- The detection voltage is the TYP. value. For details, refer to LVD Circuit Characteristics in the datasheet.

Figure 25-6 Format of user option byte (000C2H/010C2H)

Address: 000C2H/010C2H^{Note}

7	6	5	4	3	2	1	0
1	1	1	FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	High-speed on-chip oscillator clock frequency	
					f_{HOCO}	f_{IH}
1	0	0	0	0	64MHz	64MHz
0	1	0	0	0	48MHz	48MHz
1	0	0	0	1	64MHz	32MHz
0	1	0	0	1	48MHz	24MHz
1	0	0	1	0	64MHz	16MHz
0	1	0	1	0	48MHz	12MHz
1	0	0	1	1	64MHz	8MHz
0	1	0	1	1	48MHz	6MHz
1	0	1	0	0	64MHz	4MHz
0	1	1	0	0	48MHz	3MHz
1	0	1	0	1	64MHz	2MHz
Others					Setting prohibited	

Note: During boot swap, 000C2H is replaced by 010C2H, so the value of 010C2H must be set to the same as 000C2H.

Cautions:

1. Set the bits 7 to 5 to 1.
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.

25.3 Format of Flash Memory Data Protection Option Byte

The format of the flash data protection option bytes is shown below.

Figure 25-7 Format of flash memory data protection option byte (000C3H/010C3H)

Address: 000C3H/010C3H^{Note}

Symbol	7	6	5	4	3	2	1	0
OCDEN[7:0]								

Address: 500004H

Symbol	7	6	5	4	3	2	1	0
OCDM[7:0]								

OCDM	OCDEN	Control of flash data protection
3C	C3	Manipulation of flash data via debugger is not allowed.
Values other than 3C	C3	Chip full erase operations on flash data via debugger are allowed, read and write operations are not allowed.
Others		Read/write/erase operations on flash data via debugger are allowed.

Note: During boot swap, 000C3H is replaced by 010C3H, so the value of 010C3H must be set to the same as 000C3H.

Caution: The addresses 50_0004H and 50_0008H belong to the data flash memory area. If these addresses are used for data storage, it is necessary to ensure that the values will not cause incorrect setting of protection options.

Address: 500008H

Symbol	31~0
BTEN	

BTEN	Control of boot swap function
0x53574150	Use the boot swap function to swap the contents of 0000H~0FFFH and 1000H~1FFFH.
Other than the above	Disable the swap function.

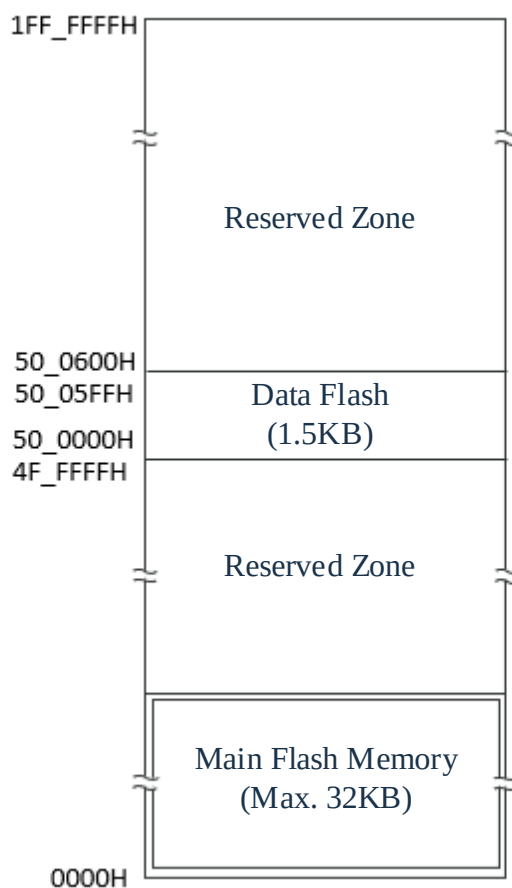
Caution: The addresses 50_0004H and 50_0008H belong to the data flash memory area. If these addresses are used for data storage, it is necessary to ensure that the values will not cause incorrect setting of protection options.

Chapter 26 Flash Control

26.1 Description of Flash Control

This product contains a 32KB Flash memory, which is divided into 64 sectors, each with a capacity of 512 bytes. It can be used as both program memory and data memory. This module supports erase, programming, and read operations on this memory.

26.2 Structure of Flash Memory



26.3 Registers for Controlling Flash

The registers that control the Flash are shown below:

- Flash write protect register (FLPROT)
- Flash operation control register (FLOPMD1, FLOPMD2)
- Flash erase mode control register (FLERMD)
- Flash status register (FLSTS)
- Flash chip erase time control register (FLCERCNT)
- Flash sector erase time control register (FLSERCNT)
- Flash write time control register (FLPROCNT)
- Flash mode time control register (FLNVSCNT/FLPRVCNT/FLERVCNT)

26.3.1 Flash Write Protect Register (FLPROT)

The Flash protect register is used to protect the Flash operation control registers.

Address: 0x40020020 After reset: 00000000H R/W

Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLPROT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PRKEY[7:1]							WRP
WRP		Operation register (FLOPMD1/FLOPMD2) write protection														
0		Rewriting of FLOPMD1/ FLOPMD2 is not allowed.														
1		Rewriting of FLOPMD1/ FLOPMD2 is allowed.														
PRKEY[7:1]		WRP write protection														
78h		Rewriting of WRP is allowed.														
Others		Rewriting of WRP is not allowed.														

26.3.2 Flash Operation Control Register (FLOPMD1, FLOPMD2)

The Flash Operation Control Register is used to set the Flash erase and write operations.

Address: 0x40020004 After reset: 00000000H R/W

Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLOPMD1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	FLOPMD1[7:0]						

Address: 0x40020008 After reset: 00H R/W

Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLOPMD2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	FLOPMD2[7:0]						

FLOPMD1	FLOPMD2	OPERATION
55	AA	Erase
AA	55	Write
00	00	Read
Others		Settings are prohibited

26.3.3 Flash Erase Control Register (FLERMD)

The Flash Erase Control Register is used to set the type of Flash erase operation.

Address: 0x4002000C After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLERMD	0	0	0	ERMD1	ERMD0	0	0	0

ERMD1	ERMD0	OPERATION
0	0	Sector erase, no hardware verification after erase
1	0	Sector erase, hardware verification after erase
0	1	Chip erase ^{Note}
1	1	Settings are prohibited

Note: Chip erase only erases the code flash area, not the data flash area. And chip erase does not support hardware verification.

26.3.4 Flash Status Register (FLSTS)

The status of the Flash controller can be queried through the status register.

Address: 0x40020000 After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLSTS	0	0	0	0	0	EVF ^{Note}	0	OVF ^{Note}

OVF	Flash erase operation complete flag
0	The Flash erase operation is not completed
1	The Flash erase operation is completed

Note: The OVF needs to be cleared by writing to 1 via software. If it is not cleared, the next erase/write operation cannot be performed.

EVF	Flash erase hardware check error flag
0	No error occurred in hardware verification after Flash erase
1	A hardware check error has occurred after Flash erase

Note: The EVF needs to be cleared by writing to 1 via software.

26.3.5 Flash Chip Erase Time Control Register (FLCERCNT)

The Flash full erase time can be set through the FLSERCNT register.

Address: 0x40020010 After reset: undefined R/W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
load	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	FLCERCNT[9:0]									

Load	Selection of erase time setting ^{Note}
0	Use hardware to set erase time
1	Use software to set erase time (FLCERCNT[9:0])

Note: When the main clock is an on-chip high-speed OCO or an external input clock ≤ 20M, you can use hardware to set the time without setting the FLCERCNT.

FLCERCNT[9:0]	Software erase time setting
Chip erase time = (CERCNT*2048*Tfclk), need to meet >20ms hardware requirement	

26.3.6 Flash Sector Erase Time Control Register (FLSERCNT)

The Flash sector erase time can be set through the FLSERCNT register.

Address: 0x40020014 After reset: undefined R/W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Load	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	FLSERCNT[9:0]									

Load	Selection of erase time setting ^{Note}
0	Use hardware to set erase time
1	Use software to set erase time (FLSERCNT[9:0])

Note: When the main clock is an on-chip high-speed OCO or an external input clock $\leq 20\text{M}$, you can use hardware to set the time without setting the FLSERCNT.

FLSERCNT[9:0]	Software erase time setting
Sector erase time = (SERCNT*256*Tfclk) need to meet >4ms hardware requirement	

26.3.7 Flash Write Time Control Register (FLPROCNT)

The Flash word write time can be set through the FLPROCNT register.

Address: 0x4002001C After reset: undefined R/W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Load1	-	-	-	-	-	-	FLPGSCNT[8:0]								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Load0	-	-	-	-	-	-	FLPROCNT[8:0]								

Load0	Write time (T_{prog}) setting ^{Note}
0	Use hardware to set write times
1	Use hardware to set erase times (FLPROCNT[9:0])

Note: When the main clock is an on-chip high-speed OCO or an external input clock $\leq 20\text{M}$, you can use hardware to set the time without setting the FLPROCNT.

FLPROCNT[8:0]	Software erase time setting
Write time = (PROCNT*4*T _{fclk}), need to meet >24us hardware requirements	

Load1	Write action setup time (T_{pgs}) setting ^{Note}
0	Write action setup time using hardware
1	Erase time set using software (FLPGSCNT8:0])

Note: When the main clock is an on-chip high-speed OCO or an external input clock $\leq 20\text{M}$, you can use hardware to set the time without setting the FLPGSCNT.

FLPGSCNT[8:0]	Software erase time setting
Write action setup time = (PGSCNT*T _{fclk}), need to meet >5us hardware requirement	

26.3.8 Flash Mode Time Control Register (FLNVSCNT/FLPRVCNT/FLERVCNT)

When writing/erasing to Flash, the related signals can be configured through the FLNVSCNT register to set the setup time (T_{nv}s) for the Flash address.

Address: 0x40020018 After reset: undefined R/W

Symbol

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
load	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	FLNVSCNT[8:0]							

FLNVSCNT

Load	Setup time selection ^{Note}
0	Use the time set by hardware.
1	Use the software-set erase time (FLNVSCNT[8:0])

Note: When the main clock is the on-chip high-speed OCO or the external input clock is $\leq 20\text{M}$, hardware time can be used, and FLNVSCNT does not need to be set.

FLNVSCNT[8:0]	Software erase time setting
Setup erase time = (NVSCNT * T _{flck}), which needs to meet the hardware requirement of $>6\mu\text{s}$.	

Through the FLPRVCNT register, the Flash write recovery time (T_{rcv}) can be set.

Address: 0x40020038 After reset: undefined R/W

Symbol

FLPRVCNT

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
load	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	FLPRVCNT[8:0]							

Load	Program recovery time setting ^{Note}
0	Use the time set by hardware.
1	Use the software-set erase time (FLPRVCNT[8:0])

Note: When the main clock is the on-chip high-speed OCO or the external input clock is $\leq 20\text{M}$, hardware time can be used, and FLPRVCNT does not need to be set.

FLPRVCNT[8:0]	Software erase time setting
Recovery time = (PRVCNT * T _{flck}), which needs to meet the hardware requirement of $>5\mu\text{s}$.	

Through the FLERVCNT register, the Flash erase recovery time (T_{rcv}) can be set.

Address: 0x4002003C After reset: undefined R/W

Symbol

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Load1	-	-	-	-	-	FLCRVCNT[25:16]									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Load0	-	-	-	-	-	-	FLSRVCNT[8:0]								

Load1	Recovery time setting for chip erase ^{Note}
0	Use the time set by hardware.
1	Use the software-set erase time (FLCRVCNT[25:16])

Note: When the main clock is the on-chip high-speed OCO or the external input clock is $\leq 20\text{M}$, hardware time can be used, and FLCRVcnt does not need to be set.

FLCRVCNT[8:0]	Software erase time setting
Recovery time = (CRVCNT * 16 * T _{clk}), which needs to meet the hardware requirement of $>200\mu\text{s}$.	

Load0	Recovery time setting for sector erase ^{Note}
0	Use the time set by hardware.
1	Use the software-set erase time (FLSRVCNT[8:0])

Note: When the main clock is the on-chip high-speed OCO or the external input clock is $\leq 20\text{M}$, hardware time can be used, and FLSRVCNT does not need to be set.

FLSRVCNT[8:0]	Software erase time setting
Recovery time = (SRVCNT * 8 * T _{clk}), which needs to meet the hardware requirement of $>50\mu\text{s}$.	

26.4 How to Operate the Flash

26.4.1 Sector Erase

Sector erase, and the erase time are implemented by hardware or can be configured by FLSERCNT. The operation flow is as follows:

1) Set FLERMD.ERMD0 to 1'b0, select the sector erase mode, and choose to set the value of ERMD1 according to whether or not hardware check is required.

2) Set FLPROT to 0xF1 to unprotect FLOPMD. Then set FLOPMD1 to 0x55 and FLOPMD2 to 0xAA.

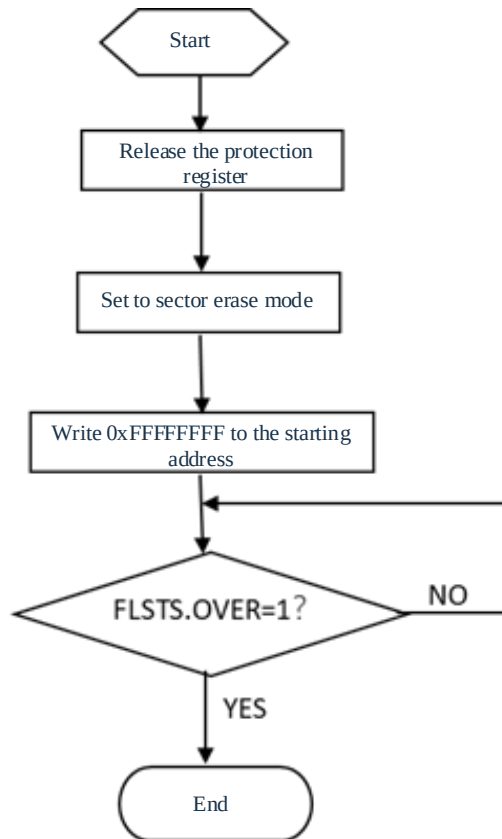
3) Write arbitrary data to the first address of the erase target sector.

Example: *((unsigned long *) 0x00000200)=0xffffffff.

4) Software query status register FLSTS.OVF,OVF=1, indicates that the erase operation is completed.

5) If hardware check after erase is set (ERMD1=1), you can determine whether the verification is correct by software for FLSTS.EVF.

6) Before the next operation, set the software to 1 to clear the FLSTS.



26.4.2 Chip Erase

Chip erase, and the erase time are implemented by hardware and can also be configured via FLCERCNT. The operation process is as follows:

- 1) Set FLERMD. ERMD0 to 1'b1, and select chip erase mode.
- 2) Set FLPROT to 0xF1 to unprotect FLOPMD. Then set FLOPMD1 to 0x55 and FLOPMD2 to 0xAA.
- 3) Write arbitrary data to any address in the flash area of the code.
- 4) Query the status register FLSTS.OVF through software, when OVF=1, it means the erase operation is completed.
- 5) Before the next operation, set the software to 1 to clear the FLSTS.

26.4.3 Word Program

Word programming and write time are implemented by hardware and can also be configured via PROCNT. The operation process is as follows:

- 1) Set FLPROT to 0xF1, unprotect FLOPMD. Then set FLOPMD1 to 0xAA, FLOPMD2 to 0x55.
- 2) Write the corresponding data to the target address.
- 3) Query the status register FLSTS.OVF through software, when OVF=1, it means the write operation is completed.
- 4) Before the next operation, set the software to 1 to clear the FLSTS.

26.5 Flash Read

The fastest fetch frequency supported by the built-in Flash is 32 MHz. when the HCLK frequency exceeds 32 MHz, the hardware will insert 1 wait cycle when the CPU accesses the Flash.

26.6 Cautions for Flash Operation

- The Flash memory has strict timing requirements for erase and program operations. If the timing of the control signals does not meet the specifications, the erase and program operations may fail. The erase and program parameters can be set either by hardware or modified via parameter registers through software. When using the internal high-speed OCO, MAINOSC/external input clock = 20M, it is recommended to use the hardware-set erase/program parameters without modifying the parameter registers.
- If the erase/program operation is executed from Flash, the CPU will stop fetching instructions, and the hardware will automatically wait until the operation is completed before continuing with the next instruction. If the operation is executed from RAM, the CPU will not stop fetching instructions, and it can continue with the next instruction.
- While the Flash is performing a program operation, if the CPU executes a deep sleep instruction, the system will wait until the programming operation is complete before entering deep sleep.

Appendix Version History

Version #	Date	Description of changes
V1.0	Oct. 2019	Initial release
V1.01	Nov. 2019	Full text typo corrections
V1.02	Apr. 2020	Full text typo corrections
V1.03	Jul. 2020	13.1.1 The maximum transfer rate of SSPI master control was modified to $f_{CLK}/2$.
		Added Chapter 24 Temperature Sensor
V1.04	Aug. 2021	19.3 The standby function comparator operating status was modified. 23.3 “RAM operation only” was deleted.
V1.05	Nov. 2022	Descriptions of the high-speed internal oscillator frequency selection register in 4.4, 4.3.8, 4.4.3, 4.6.1, 25.2 were modified.
V1.0.6	Sep. 2023	Modified errors in Figure 22-1 and Figure 22-6.
	Mar. 2024	Corrected the error in section 5.1.2: The number of multiple PWM in Timer4 was modified. Corrected the error in section 5.1.26.3.4: The description of the RWAIT bit in the register section was modified.
V1.0.7	Jul. 2024	- Modified 1.4: DBGSTOPCR register added the RESMSK bit. - Modified 2.3.7, 2.3.8, 2.3.9: Added PORT's PSET, PCLR, PREAD registers and compatibility with simplified IIC. - Modified 4.3.8: HOCODIV2-0 was set to 6 for a 64-clock division. - Modified Timer4 TDR register address decoding: Made TDR addresses contiguous. - Modified SCI SDR register address decoding: Made SDR addresses contiguous. - Modified 9.3.2 and 9.3.3: Added LOCKCTL, PRCR, and WDTCFG0/1/2/3 registers. - Modified 16.3.12 and 16.3.13: Added DMA set and reset registers. - Modified 25.3: BOOT enable was changed from 1 bit to 32-bit control.
Version B V0.5.0	Nov. 2024	- Chapter 2 Section 2.3.20: Added compatibility configuration for simplified IIC. - Chapter 13: Added relevant information about simplified IIC. 23.3.8: The base address of the ID register was changed from 50_084C to 50_0848.
V0.5.1	June 2025	Correct the incorrect content