BAT32A237 User Manual

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

Rev1.00

Documentation Instructions

This manual is the technical reference manual for the BAT32A237 microcontroller product. The technical reference manual is the application instruction material on how to use this series of products, including the structure, function description, working mode and register configuration of each functional module.

The technical reference manual is a description of all functional modules of this series of products. If you want to know the feature description of the product (that is, the functional configuration), you can refer to the respective data sheet.

The data sheet information is as follows：

BAT32A237xx：BAT32A237_datasheet_vx.x. pdf

Usually in the early stage of chip selection, you shall first check the data sheet to evaluate whether the product can meet the functional requirements of the design; after basically selecting the required product, you need to check the technical reference manual to determine whether the working mode of each functional module does meet the requirement；When determining the selection and entering the programming design stage, you need to read the technical reference manual in detail to understand the specific implementation and register configuration of each function. Refer to the data sheet for information on voltages, currents, drive capabilities, and pin assignments when designing hardware。

For a detailed description of the Cortex-M0+ core, SysTick timer and NVIC, please refer to the respective ARM documents.

Chapter 1 CPU

1.1 verview

This Chapter provides a brief introduction to the features and debugging features of the ARM Cortex-M0+ kernel on which this product is built. Please refer to the ARM documentation for details.

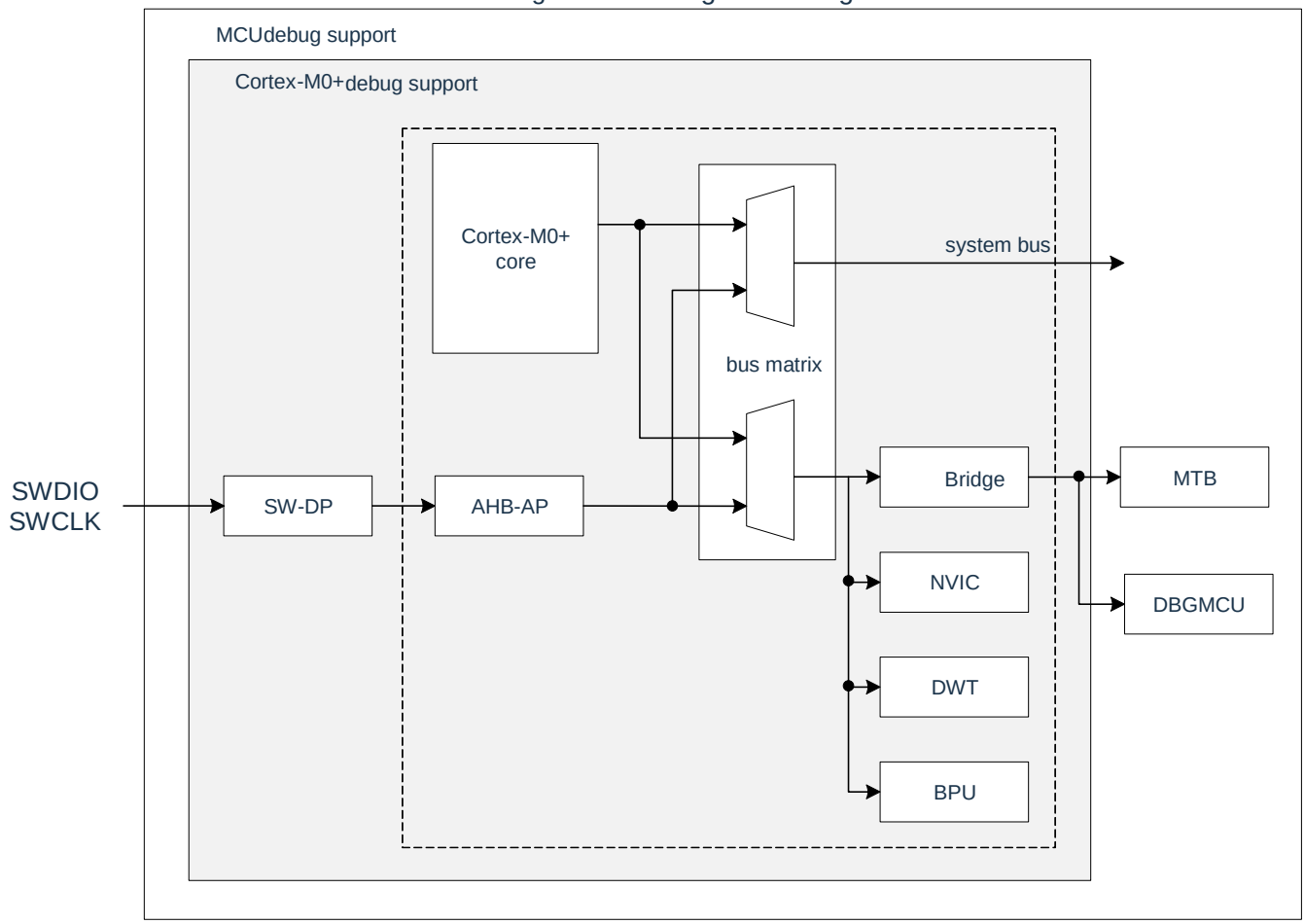
1.2 Cortex-M0+ core features

- ARM Cortex-M0+ processors are 32-bit RISC cores with a 2-stage pipeline that supports privileged and user modes
- Memory Protection Units (MPUs) support 8 separate Zone (region) protection
- single cycle hardware multiplier
- Nested Vector Interrupt Controller (NVIC)
  - 1 Unshielded Interrupt (NMI)
  - Supports 32 Masking Interrupt Requests (IRQs)
  - 4 interrupt priority
- The system timer SysTick is a 24-bit countdown timer that can be selected for fCLK or fIL 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, with low 8-bit write ignore, read to zero, that is, offset 256 bytes aligned

1.3 Debug features

- 2-wire SWD debug interface
- Support for pausing, resuming, and single-step execution procedures
- Access processor's kernel register and special function register
- 4 Hardware Breakpoints (BPUs)
- Unlimited software breakpoints (BKPT instruction)
- 2 data observation points (DWT)
- Access memory during kernel execution
- Micro Trace Cache (MTB) provides a simple instruction execution trace scheme
  - MTB shared with user SRAM (0x2000_0000-0x2000_2 FFF zone)
  - The base address of the MTB control register is 0x4001_9000

Figure 1-1 Debug Block Diagram for Cortex-M0+



Note: SWD does not work in Deep Sleep mode, please debug in active and sleep mode.

1.4 SWD interface pin

2 GPIO of the product can be used as SWD interface pins that are present in all packages.

Table 1-1 SWD Debug Port Pin

SWD port name	Debug Function	pin assignment
SWCLK	serial clock	P137
SWDIO	Serial Data Input/Output	P40

When you do not use the SWD feature, you can disable the SWD by setting the debug stop control register (DBGSTOPCR).

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

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

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

Bit no.	7	6	5	4	3	2	1	0
DBGSTOPCR	-	-	-	-	-	-	FRZEN1	FRZEN0
Default	0	0	0	0	0	0	0	0

SWDIS	SWD Debug Interface Disable
0	SWD debug interface enable. P40 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. P40 can be used as GPIO

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 action
1	peripheral stop

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 action
1	peripheral stop

Note 1: The timer system peripheral module of the product includes: Universal timer unit Timer4, timer A, timer B, timer C, and timer M

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

1.5 ARM reference document

The built-in debugging features in the Cortex®-M0+ kernel are part of the ARM® CoreSight design suite. For documentation, 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 function

2.1 port function

Refer to datasheet for each product family.

2.2 port multiplex

Refer to datasheet for each product family.

2.3 Register for controlling port function

Control the port through the following registers.

- Port Mode Registers (PMxx)
- Port Registers (Pxx)
- Pull-up resistor selection registers (PUxx)
- Port Input Mode Registers (PIMx)
- Port Output Mode Registers (POMx)
- Port Mode Control Registers (PMCxx)
- Peripheral I/O Redirection Registers (PIORx)
- Global Digital Input Disable Register (GDIDIS)

Note: The assigned registers and bits differ depending on the product. Refer to for registers and bits allocated by each productTable 2-1. You must set an initial value for unassigned bits.

Table 2-1 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and bits (1/2)

Port		bit name						64 pin	48 pin	44 pin	40 pin	32 pin	24 pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register						
Port 0	0	PM00	P00	PU00	—	POM00	PMC00 Note 1	○	○	○	○	○	○
	1	PM01	P01	PU01	PIM01	—	PMC01 Note 1	○	○	○	○	○	○
	2	PM02	P02	PU02	—	POM02	PMC02	○	—	—	—	—	—
	3	PM03	P03	PU03	PIM03	POM03	PMC03	○	—	—	—	—	—
	4	PM04	P04	PU04	PIM04	POM04	PMC04	○	—	—	—	—	—
	5	PM05	P05	PU05	—	—	—	○	—	—	—	—	—
	6	PM06	P06	PU06	—	—	—	○	—	—	—	—	—
Port 1	0	PM10	P10	PU10	PIM10	POM10	PMC10	○	○	○	○	○	○
	1	PM11	P11	PU11	—	POM11	PMC11	○	○	○	○	○	○
	2	PM12	P12	PU12	—	—	PMC12 Note 2	○	○	○	○	○	○
	3	PM13	P13	PU13	—	POM13	PMC13 Note 2	○	○	○	○	○	○
	4	PM14	P14	PU14	PIM14	POM14	PMC14 Note 2	○	○	○	○	○	○
	5	PM15	P15	PU15	PIM15	POM15	—	○	○	○	○	○	○
	6	PM16	P16	PU16	PIM16	—	—	○	○	○	○	○	—
	7	PM17	P17	PU17	PIM17	POM17	—	○	○	○	○	○	—
Port 2	0	PM20	P20	—	—	—	PMC20	○	○	○	○	○	○
	1	PM21	P21	—	—	—	PMC21	○	○	○	○	○	○
	2	PM22	P22	—	—	—	PMC22	○	○	○	○	○	○
	3	PM23	P23	—	—	—	PMC23	○	○	○	○	○	—
	4	PM24	P24	—	—	—	PMC24	○	○	○	○	—	—
	5	PM25	P25	—	—	—	PMC25	○	○	○	○	—	—
	6	PM26	P26	—	—	—	PMC26	○	○	○	—	—	—
	7	PM27	P27	—	—	—	PMC27	○	○	○	—	—	—

Note: 1. Products with 24-48 pins only.

2. Products with 24-64 pins only.

Table 2-2: PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and bits (2/2)

Port		bit name						64 pin	48 pin	44 pin	40 pin	32 pin	24 pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register						
Port 3	0	PM30	P30	PU30	PIM30	POM30	—	○	○	○	○	○	—
	1	PM31	P31	PU31	PIM30 Note 1	POM31 Note 1	—	○	○	○	○	○	—
Port 4	0	PM40	P40	PU40	—	—	—	○	○	○	○	○	○
	1	PM41	P41	PU41	—	—	—	○	○	○	—	—	—
	2	PM42	P42	PU42	—	—	—	○	—	—	—	—	—
	3	PM43	P43	PU43	—	—	—	○	—	—	—	—	—
Port 5	0	PM50	P50	PU50	PIM50	POM50	—	○	○	○	○	○	○
	1	PM51	P51	PU51	—	POM51	—	○	○	○	○	○	○
	2	PM52	P52	PU52	—	—	—	○	—	—	—	—	—
	3	PM53	P53	PU53	—	—	—	○	—	—	—	—	—
	4	PM54	P54	PU54	—	—	—	○	—	—	—	—	—
	5	PM55	P55	PU55	PIM55	POM55	—	○	—	—	—	—	—
Port 6	0	PM60	P60	—	—	—	—	○	○	○	○	—	—
	1	PM61	P61	—	—	—	—	○	○	○	○	—	—
	2	PM62	P62	—	—	—	—	○	○	○	—	—	—
	3	PM63	P63	—	—	—	—	○	○	○	—	—	—
Port 7	0	PM70	P70	PU70	—	—	—	○	○	○	○	○	—
	1	PM71	P71	PU71	—	POM71	—	○	○	○	—	—	—
	2	PM72	P72	PU72	—	—	—	○	○	○	○	○	○
	3	PM73	P73	PU73	—	—	—	○	○	○	○	○	○
	4	PM74	P74	PU74	PIM74 Note 1	POM74	—	○	○	—	○	○	—
	5	PM75	P75	PU75	—	—	—	○	○	—	○	—	—
	6	PM76	P76	PU76	—	—	—	○	—	—	—	—	—
	7	PM77	P77	PU77	—	—	—	○	—	—	—	—	—
Port 12	0	PM120	P120	PU120	—	—	PMC120	○	○	○	○	○	—
	1	—	P121	—	—	—	—	○	○	○	○	○	○
	2	—	P122	—	—	—	—	○	○	○	○	○	○
	3	—	P123	—	—	—	—	○	○	○	○	—	—
	4	—	P124	—	—	—	—	○	○	○	○	—	—
Port 13	0	—	P130	—	—	—	—	○	○	—	—	—	—
	6	PM136	P136	PU136	—	—	—	○	○	—	○	○	○
	7	PM137	P137	PU137	—	—	—	○	○	○	○	○	○
Port 14	0	PM140	P140	PU140	—	—	—	○	○	—	○	—	—
	1	PM141	P141	PU141	—	—	—	○	—	—	—	—	—
	6	PM146	P146	PU146	—	—	PMC146	○	○	○	—	—	—
	7	PM147	P147	PU147	—	—	PMC147	○	○	○	○	○	○

Note: 1. Products with 32 pins only.

2.3.1 Port Mode Register (PMxx)

This is a register that sets the port input/output in bits. After the reset signal is generated, the values of these registers become "FFH". When using a port pin as a pin for a multiplexing function, it must be configured according to reference to "2.5 Register settings when using multiplexing function".

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

Figure 2-1 Port mode register format

symbol	7	6	5	4	3	2	1	0	offset	After Reset	R/W
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00	0x320	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM1 0	0x321	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	0 x322	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM3 0	0x323	FFH	R/W
PM4	1	1	1	1	PM43	PM42	PM 41	PM4 0	0x324	FFH	R/W
PM5	1	1	PM55	PM54	PM53	PM52	PM 51	PM50	0x325	FFH	R/W
PM6	1	1	1	1	PM63	PM62	PM61	PM60	0x326	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM 71	PM70	0x327	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	0x32C	FFH	R/W
PM13	PM137	PM136	1	1	1	1	1	PM130	0x32D	FEH	R/W
PM14	PM147	PM146	1	1	1	1	PM141	PM140	0x32E	FFH	R/W

PMmn	Selection of Input/Output Modes of Pmn Pins (m=0~7,12~14, n=0~7)
0	Output mode (used as output port (output buffer ON))
1	Input mode (used as input port (output buffer OFF))

Note: You must set an initial value for unassigned bits.

2.3.2 Port Register (Pxx)

This is a register that sets the value of the port output latch in bits. The pin level is read in the input mode and the output latch value of the port is read in the output mode. After the reset signal is generated, the values of these registers change to '00H'.

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

Figure 2-2 Port register format

symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
P0	0	P06	P05	P04	P03	P02	P01	P00	0x300	00H (Output Latch)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	0x301	00H (Output Latch)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	0x302	00H (Output Latch)	R/W
P3	0	0	0	0	0	0	P31	P30	0x303	00H (Output Latch)	R/W
P4	0	0	0	0	P43	P42	P41	P40	0x304	00H (Output Latch)	R/W
P5	0	0	P55	P54	P53	P52	P51	P50	0x305	00H (Output Latch)	R/W
P6	0	0	0	0	P63	P62	P61	P60	0x306	00H (Output Latch)	R/W
P7	P77	P76	P75	P74	P73	P72	P71	P70	0x307	00H (Output Latch)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	0x30C	indefinite value	R/W Note 1
P13	P137	P136	0	0	0	0	0	P130	0x30D	00H (Output Latch)	R/W
P14	P147	P146	0	0	0	0	P141	P140	0x30E	00H (Output Latch)	R/W

Pmn	m=0~7,12~14, n=0~7	
	Control of output data (output mode)	Reading of input data (input mode)
0	Output "0".	Input low level.
1	Output "1".	Input high level.

Note: 1. P121~P124 is a read only bit.
2. You must set an initial value for unassigned bits.

2.3.3 Pull-up resistance selection register (PUxx)

Selection register for internal pull-up resistance. Internal pull-up resistors can only be used in bit units if the POMmn bit is '0' and set to the input mode (PMmn=1). For the bit set to the output mode, the internal pull-up resistor is not connected regardless of the setting of the pull-up resistor selection register. The same is true when the output pin used as the multiplexing function or set as the analog function.

After the reset signal is generated, the values of these registers become '00H' (only PU4 is '01H' and PU13 is '80H').

register address = base address + offset address; The PU register has a base address of 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	offset	After Reset	R/W
PU0	0	PU06	PU05	PU04	PU03	PU02	PU01	PU00	0x030	00H	R/W
PU1	PU17	PU1 6	PU15	PU14	PU13	PU12	PU11	PU1 0	0x031	00H	R/W
PU3	0	0	0	0	0	0	PU31	PU30	0x033	00H	R/W
PU4	0	0	0	0	PU43	PU42	PU41	PU4 0	0x034	01H	R/W
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50	0x035	00H	R/W
PU7	PU77	PU7-6	PU7-5	PU7-4	PU7-3	PU7-2	PU7-1	PU70	0x037	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	0x03C	00H	R/W
PU13	PU137	PU136	0	0	0	0	0	0	0x03D	80H	R/W
PU14	PU147	PU146	0	0	0	0	PU141	PU140	0x03E	00H	R/W

PUMN	Selection of internal pull-up resistance of Pmn pin (m=0,1,3~5,7,12~14,n=0~7)
0	The internal pull-up resistor is not connected.
1	Connect the internal pull-up resistor.

Note: You must set an initial value for unassigned bits.

2.3.4 Port input mode register (PIMxx)

This is a register that sets the input buffer in bits. The TTL input buffer can be selected in serial communication with external devices of different potentials. After the reset signal is generated, the values of these registers change to '00H'.

register address = base address + offset address; The PIM register has a base address of 0x40040000 and the offset address is shown in the figure below.

Figure 2-4: Format of port input mode register

symbol	7	6	5	4	3	2	1	0	offset	After Reset	R/W
PIM0	0	0	0	PIM04	PIM03	0	PIM01	0	0x040	00H	R/W
PIM1	PIM17	PIM16	PIM15	PIM14	0	0	0	PIM1 0	0x041	00H	R/W
PIM3	0	0	0	0	0	0	PIM31 Note 1	PIM3 0	0x043	00H	R/W
PIM5	0	0	PIM55	0	0	0	0	PIM5 0	0x045	00H	R/W
PIM7	0	0	0	PIM74 Note 1	0	0	0	0	0x047	00H	R/W

PIMmn	Selection of input buffer for Pmn pin (m=0,1,3,5,7, n=0~7)
0	Schmidt input buffer
1	TTL input buffer

Note: 1. Products with 32 pins only.

2. You must set an initial value for unassigned bits.

2.3.5 Port output mode register (POMxx)

This is a register that sets the output mode in bits. When serial communication is performed with external devices of different potentials and simple I2C communication is performed with external devices of same potential, N channel drain open output mode is selected for SDAxx pin.

After the reset signal is generated, the values of these registers change to '00H'.

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

Note: The internal pull-up resistor is not connected to the bit where the N-channel drain open-circuit output mode (POMmn=1) is set.

Figure2-5 Port output mode register format

symbol	7	6	5	4	3	2	1	0	offset	After Reset	R/W
POM0	0	0	0	POM04	POM03	POM02	0	POM00	0 x050	00H	R/W
POM1	POM17	0	POM15	POM14	POM13	0	POM11	POM1 0	0 x051	00H	R/W
POM3	0	0	0	0	0	0	POM31 Note 1	POM3 0	0 x053	00H	R/W
POM5	0	0	POM55	0	0	0	POM51	POM5 0	0 x055	00H	R/W
POM7	0	0	0	POM74	0	0	POM71	0	0 x057	00H	R/W

POMmn	Selection of Output Mode of Pmn Pin (m=0,1,3,5,7, n=0~7)
0	Normal output mode
1	N-channel drain open-circuit output mode

Note: 1. Products with 32 pins only.

2. You must set an initial value for unassigned bits.

2.3.6 Port mode control register (PMCxx)

A digital input/output or analog input that is set in bits by the PMC register.

After the reset signal is generated, the values of these registers become "FFH".

register address = base address + offset address; The PMC register has a base address of 0x40040000 and the offset address is shown in the figure below.

Figure2-6: Port mode control register format

symbol	7	6	5	4	3	2	1	0	offset	After Reset	R/W
PMC0	1	1	1	PMC04	PMC03	PMC02	PMC01 Note 1	PMC00 Note 1	0 x06 0	FFH	R/W
PMC1	1	1	1	1	1	1	PMC11	PMC10	0 x06 1	FFH	R/W
PMC2	PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20	0 x062	FFH	R/W
PMC12	1	1	1	1	1	1	1	PMC120	0x06C	FFH	R/W
PMC14	PMC147	PMC146	1	1	1	1	1	1	0x06E	FFH	R/W

PMCmn	Digital input/output of Pmn pin or selection of analog input (m=0,1,2,12,14, n=0~7)
0	Digital Input/Output (multiplexing function other than analog input)
1	analog input

NOTE: 1. Products with 32-48 pins only.

2. You must set an initial value for unassigned bits.

2.3.7 Peripheral I/O Redirection Register 0 (PIOR0)

This is register 0 that sets the ability to allow or prohibit peripheral I/O redirection. The peripheral I/O redirection function switches ports to which multiplexing is assigned.

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

Figure 2-7: Format of Peripheral I/O Redirection Register 0 (PIOR0)

Address: 0x40040877 After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PIOR0	PIOR07	PIOR06	PIOR05	PIOR04	PIOR03	PIOR02	PIOR01	PIOR00

PIOR0	Features	64pin		48pin		44pin		40pin		32pin		24pin	
		Set Values		Set Values		Set Values		Set Values		Set Values		Set Values	
		0	1	0	1	0	1	0	1	0	1	0	1
bit7(PIOR07)	INTP8	P74/P42	P00	P74	P00	must be set to 0 (initial value)							
	INTP10	P76/P05	P01										
	INTP11	P77/P06	P20										
bit6(PIOR06)	RxD2	must be set to 0 (initial value)		P14	P14	P14	P14	P14	P14	P14	P14	P14	P14
	TxD2			P13	P10	P13	P10	P13	P10	P13	P10	P13	P10
	SCL20			P15	-	P15	-	P15	-	P15	-	P15	-
	SDA20			P14	-	P14	-	P14	-	P14	-	P14	-
	SDI20			P14	-	P14	-	P14	-	P14	-	P14	-
	SDO20			P13	-	P13	-	P13	-	P13	-	P13	-
	SCLK20			P15	-	P15	-	P15	-	P15	-	P15	-
bit5(PIOR05)	RXD1			P01	P73	P01	P73	P01	P73	P01	P73	P01	P73
	TXD1			P00	P72	P00	P72	P00	P72	P00	P72	P00	P72
bit4(PIOR04)	CLKBUZ1	P141	P55	must be set to 0 (initial value)									
	INTP5	P16	P12										
bit3(PIOR03)	CLKBUZ0	P140	P31	P140	P31	-	P31	P140	P31				
bit2(PIOR02)	SCLA0	P60	P14	P60	P14	P60	P14	P60	P14	P31	P14	P01	P14
	SDAA0	P61	P15	P61	P15	P61	P15	P61	P15	P74	P15	P00	P15
bit1(PIOR01)	INTP10	P76	P05	P01	P01	P01	P01	P01	P01	P01	P01	must be set to 0 (initial value)	
	INTP11	P77	P06	P20	P20	P20	P20	P20	P20	P20	P20		
	RXD2	P14	P76	must be set to 0 (initial value)									
	TXD2	P13	P77										
	SCL20	P15	-										
	SDA20	P14	-										
	SDI20	P14	-										
	SDO20	P13	-										
	SCLK20	P15	-										
	TXD0	P51	P17	P51	P17	P51	P17	P51	P17	P51	P17		
	RXD0	P50	P16	P50	P16	P50	P16	P50	P16	P50	P16		
	SCL00	P30	-	P30	-	P30	-	P30	-	P30	-		
	SDA00	P50	-	P50	-	P50	-	P50	-	P50	-		
	SDO00	P51	P17	P51	-	P51	-	P51	-	P51	-		
	SDI00	P50	P16	P50	-	P50	-	P50	-	P50	-		
	SCLK00	P30	P55	P30	-	P30	-	P30	-	P30	-		
bit0(PIOR00)	INTP1	P50	P52	must be set to 0 (initial value)									

	INTP2	P51	P53		
	INTP3	P30	P54		
	INTP4	P31	P55		
	INTP8	P74	P42		
	INTP9	P75	P43		

2.3.8 Peripheral I/O Redirection Register 1 (PIOR1)

This is setting to allow or disable peripheral I/O redirection feature register 1. The peripheral I/O redirection function switches ports to which multiplexing is assigned.

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

Figure 2-8: Format of Peripheral I/O Redirection Register 1 (PIOR1)

Address: 0x40040879H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PIOR1	0	0	0	0	PIOR13	PIOR12	PIOR11	PIOR10

PIOR13	PIOR12	Selection of Timer ATAO Pin
0	0	Multiplexed with P30 (32-64 pin products).
0	1	Multiplexed with P50.
1	0	Multiplexed with P00.
1	1	Setting is prohibited.

PIOR11	PIOR10	Selection of Timer ATAIO Pin
0	0	Multiplexed with P01.
0	1	Multiplexed with P31 (32-64 pin products).
1	0	Multiplexed with P41 (limited to 48, 64-pin products). Other pin products are prohibited from setting.
1	1	Multiplexed with P06 (limited to 64-pin products). Other pin products are prohibited from setting.

2.3.9 Peripheral I/O Redirection Register 2 (PIOR2)

This is setting to enable or disable peripheral I/O redirection feature register 2. The peripheral I/O redirection function switches ports to which multiplexing is assigned. After the reset signal is generated, the value of this register changes to "00H".

Figure 2-9: Format of Peripheral I/O Redirection Register 2 (PIOR2)

Address: 0x40040875 After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PIOR2	PIOR27	PIOR26	PIOR25	PIOR24	PIOR 23	PIOR22	PIOR21	PIOR20

PIOR2	Features	64pin		48pin		44pin		40pin		32pin		24pin	
		Set Values		Set Values		Set Values		Set Values		Set Values		Set Values	
		0	1	0	1	0	1	0	1	0	1	0	1
bit7(PIOR27)	TMIOC0	x	x	x	x	x	x	x	x	x	x	P16	P13
bit6(PIOR26)	TMIOD0	P15	P17	P15	P17	P15	P17	P15	P17	P15	P17	-	-
bit5(PIOR25)	TMIOD1	P11	P51	P11	P51	P11	P51	P11	P51	P11	P51	P11	P51
bit4(PIOR24)	TMIOC1	P13	P50	P13	P50	P13	P50	P13	P50	P13	P50	P13	P50
bit3(PIOR23)	TMIOB1	P10	P30	P10	P30	P10	P30	P10	P30	P10	P30	-	-
bit2(PIOR22)	TMIOA1	P12	P16	P12	P16	P12	P16	P12	P16	P12	P16	-	-
bit1(PIOR21)	VCOUT1	P31	P70	P31	P70	P31	P70	P31	P70	P31	P70	P147	P72
bit0(PIOR20)	VCOUT0	P120	P71	P120	P71	P120	P71	P120	P70	P120	P73	P12	P73

Note: x indicates that the bit must be set to the initial value of 0.

2.3.10 Peripheral I/O Redirection Register 3 (PIOR3)

This is setting to enable or disable peripheral I/O redirection feature register 3. The peripheral I/O redirection function switches ports to which multiplexing is assigned. After the reset signal is generated, the value of this register changes to "00H".

Figure 2-10: Format of Peripheral I/O Redirection Register 3 (PIOR3)

Address: 0x4004087C After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PIOR3	PIOR37	PIOR36	PIOR35	PIOR34	PIOR33	PIOR32	PIOR31	0

PIOR3	Features	64pin		48pin		44pin	
		Set Values		Set Values		Set Values	
		0	1	0	1	0	1
bit5(PIOR35)	TXD0	PIOR34 Control	P12	PIOR34 Control	P12	PIOR34 Control	P12
	RXD0	PIOR34 Control	P11	PIOR34 Control	P11	PIOR34 Control	P11
bit4(PIOR34)	TXD0	PIOR01 Control	P40	PIOR01 Control	P40	PIOR01 Control	P40
	RXD0	PIOR01 Control	P137	PIOR01 Control	P137	PIOR01 Control	P137
bit3(PIOR33)	CRXD0	P03	P50	P01	P50	P01	P50
	CTXD0	P02	P51	P00	P51	P00	P51
bit2(PIOR32)	VCOUT1	Output L	PIOR21 Control	Output L	PIOR21 Control	Output L	PIOR21 Control
bit1(PIOR31)	VCOUT0	Output L	PIOR20 Control	Output L	PIOR20 Control	Output L	PIOR20 Control
bit0(PIOR30)	IrRXD	×	×	×	×	×	×
	IrTXD	×	×	×	×	×	×

PIOR3	Features	40pin		32pin		24pin	
		Set Values		Set Values		Set Values	
		0	1	0	1	0	1
bit5(PIOR35)	TXD0	PIOR34 Control	P12	PIOR34 Control	P12	PIOR34 Control	P12
	RXD0	PIOR34 Control	P11	PIOR34 Control	P11	PIOR34 Control	P11
bit4(PIOR34)	TXD0	PIOR01 Control	P40	PIOR01 Control	P40	PIOR01 Control	P40
	RXD0	PIOR01 Control	P137	PIOR01 Control	P137	PIOR01 Control	P137
bit3(PIOR33)	CRXD0	P01	P50	P01	P50	P01	P50
	CTXD0	P00	P51	P00	P51	P00	P51
bit2(PIOR32)	VCOUT1	Output L	PIOR21 Control	Output L	PIOR21 Control	Output L	PIOR21 Control
bit1(PIOR31)	VCOUT0	Output L	PIOR20 Control	Output L	PIOR20 Control	Output L	PIOR20 Control
bit0(PIOR30)	IrRXD	×	×	×	×	P14	P01
	IrTXD	×	×	×	×	P13	P00

PIOR37	PIOR36	TMIA0	TMIOB0	TMIOC0	TMIOD0	TMIA1	TMIOB1	TMIOC1	TMIOD1
0	0	P17	P14	P16	P15	P12	P1 0	P1 3	P1 1
0	1	P17	P12	P16	P15	P1 1	P1 0	P14	P1 3
1	0	P17	P15	P16	P14	P1 3	P12	P1 1	P1 0
1	1	Disable setting							

Note: x indicates that the bit must be set to the initial value of 0.

2.4 Handling of unused pins

The handling of each unused pin is as following Table2-3.

Table2-3 Handling of each unused pin

name of the pin	Input/Output	Recommended connection method when not in use
P00~P06	Input/Output	Input: The EVDD or EVSS are connected by resistance alone· Output: Open Circuit.
P10~P17		
P20~P27		Input: The VDD or VSS are connected by resistance alone· Output: Open Circuit.
P30, P31		Input: The EVDD or EVSS are connected by resistance alone· Output: Open Circuit.
P40		Input: The EVDD is connected by resistance alone or set as an open circuit. Output: Open Circuit.
P41~P43		Input: The EVDD or EVSS are connected by resistance alone· Output: Open Circuit.
P50~P55		
P60~P63		Input: The EVDD or EVSS are connected by resistance alone· Output: Either place the port output latch "0" and open, or place the port output latch "1" separately The EVDD or the EVSS are connected by a resistor·
P70~P77		Input: The EVDD or EVSS are connected by resistance alone· Output: Open Circuit.
P120		
P121~P124	Input	The VDD or VSS are connected by resistance alone·
P130	output	Open Circuit.
P136	Input/Output	Input: The EVDD or EVSS are connected by resistance alone· Output: Open Circuit.
P137		Input: The VDD is connected by resistance alone or set as an open circuit. Output: Open Circuit.
P140~P147		Input: The EVDD or EVSS are connected by resistance alone· Output: Open Circuit.
RESETB	Input	The VDD is connected directly or through resistance·

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

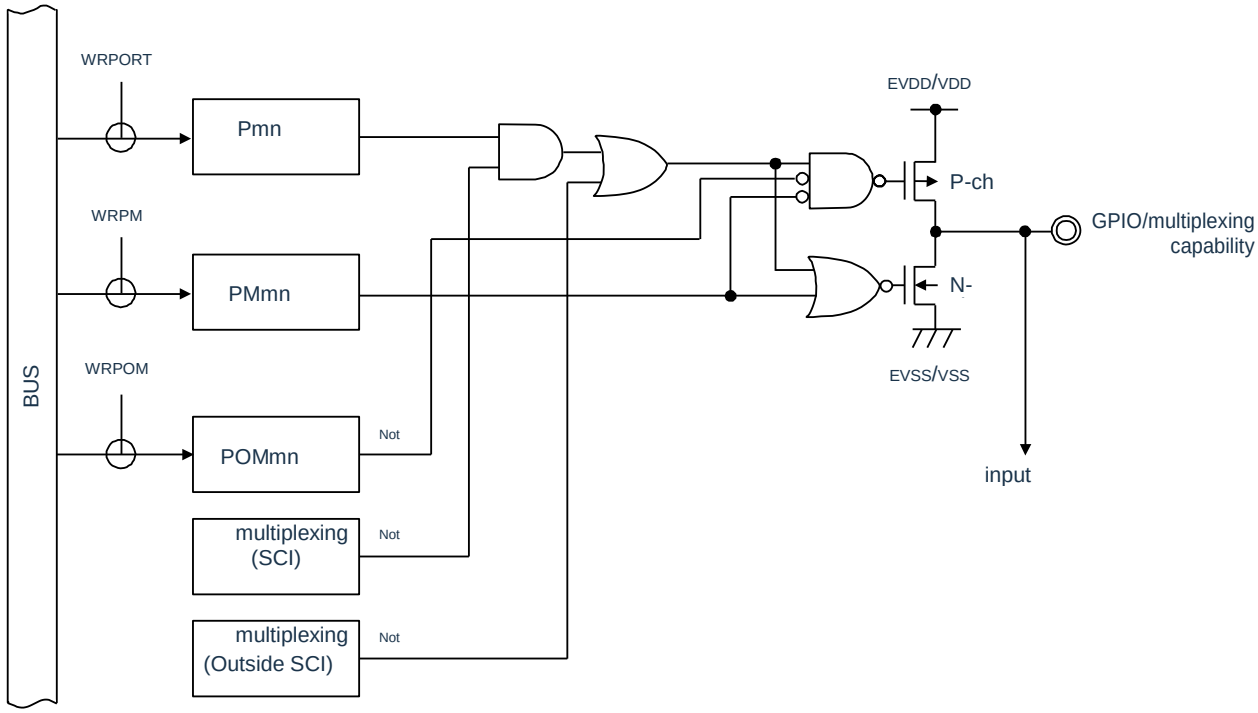
2.5 Register Settings When Using Multiplexing

2.5.1 The Basic principal of Using Multiplexing Function

First, you must set the pins that can be multiplexed with the analog function to either been used as the analog function or as the digital input/output through port mode control register (PMCxx).

The basic structure of the output circuit used as the digital input/output is as shown in Figure2-11. The output of the SCI function multiplexed with the output latch of the port is input to the AND gate, the output of the AND gate is input to the OR gate, and the other input of the OR gate is connected with the output of the multiplexed non-SCI function (timer, RTC, clock/buzzer, IICA etc.). When such a pin is used as a port function or a multiplexing function, the unused multiplexing function cannot affect the output of the function to be used. The basic idea of the setting is as shown in Table2-4.

Figure2-11 Port output basic structure



Note: 1. When there is no POM register, this signal is Low level (0).

2. When the multiplexing function is absent, the signal is High level (1).

3. The signal is Low level (0) when the multiplexing function is not available.

Table2-4 Basic Principal for Establishment

Pin Output Used	Output settings for unused multiplexing		
	port function	SCI output function	Output capabilities other than SCI
Port Output	—	High level output (1)	Low level output (0)
SCI output function	High(1)	—	Low level output (0)
Output capabilities other than SCI	Low(0)	High level output (1)	Low level output (0) note

Note: Since one pin may be reused as output functions other than SCI, it is necessary to set the output of unused multiplexing functions to a Low level (0). For specific set-up methods, refer to "2.5.2 Examples of Register Settings for Port and Multiplexing Functions Used".

2.5.2 Examples of Register Settings for Port and Multiplexing Functions Used

Examples of register settings (64 pin products) using port functions and multiplexing functions are shown in Tables 2-5 through 2-10.

Table 2-5: Example of Register Setting When Using the P00~P06 Pin Function

Pin Name	Functions Used		PIORx	POMxx	PMCxx	PMxx	Pxx	Output of multiplexing functions	
	Feature Name	Input/Output						Output Function of SCI/CAN	Beyond SCI/CAN
P00	P00	Input	—	×	0	1	×	×	—
		output	×	0	0	0	0/1	—	(TAO)=0
		N-channel drain open circuit output		1	0	0	0/1		
	TI00	Input	—	—	0	1	×	—	—
	TBCLK0	Input	—	×	0	1	×	×	—
	(TAO)	output	PIOR13, PIOR12=10B	0	0	0	0	—	—
	(INTP8)	Input	PIOR07=1	—	0	1	×	—	—
P01	P01	Input	—	—	0	1	×	—	—
		output	×	—	0	0	0/1	—	TEO=0, TO00=0
	TO00	output	—	—	0	0	0	—	TEO=0
	TBCLK1	Input	—	—	0	1	×	—	—
	TIAO	Input	PIOR13, PIOR12=00B	—	0	1	×	—	—
		output	—	—	0	0	0	—	TO00=0
	(INTP10)	Input	PIOR07=1	—	0	1	×	—	—
P02	P02	Input	—	×	0	1	×	×	—
		output	×	0	0	0	0/1	TxD1/SDO10=1 CTxD0=1	—
		N-channel drain open circuit output		1	0	0	0/1		
	ANI11	analog input	—	×	1	1	×	×	—
	TxD1	output	—	0/1	0	0	1	CTxD0=1	—
	SDO10	output	—	0/1	0	0	1	CTxD0=1	—
	VCIN10	analog input	—	×	1	1	×	×	—
	CTxD0	output	—	0/1	0	0	1	TxD1/SDO10=1	—
P03	P03	Input	—	×	0	1	×	×	—
		output	×	0	0	0	0/1	SDA10=1	—
		N-channel drain open circuit output		1	0	0	0/1		
	ANI10	analog input	—	×	1	1	×	×	—
	STI10	Input	—	×	0	1	×	×	—
	RxD1	Input	—	×	0	1	×	×	—
	SDA10	Input/Output	—	1	0	0	1	—	—
	VCIN11	analog input	—	×	1	1	×	×	—
	CRxD0	Input	—	×	0	1	×	×	—
P04	P04	Input	—	×	0	1	×	—	—
		output	×	0	0	0	0/1	SCLK10/SCL10 =1	—
		N-channel drain open circuit output		1	0	0	0/1		
	ANI1 3	analog input	—	×	1	1	×	×	—
	SLK10	Input	—	×	0	1	×	—	—
		output		0/1	0	0	1	—	—

	SCL10	output	—	0/1	0	0	1	—	—
P05	P05	Input	—	—	—	1	×	—	—
		output	—	—	—	0	0/1	—	—
	(INTP10)	Input	PIOR01=1	—	—	1	×	—	—
P06	P06	Input	—	—	—	1	×	—	—
		output	×	—	—	0	0/1	—	(TAIO)=0
	(TAIO)	Input	PIOR13, PIOR12=11B	—	—	1	×	—	—
		output		—	—	0	0	—	—
	(INTP11)	Input	PIOR01=1	—	—	1	×	—	—

Table 2-6: Example of Register Setting When Using P10~P17 Pin Functions (1/3)

name of the feet	Functions Used		PIORx	POMxx	PMCxx	PMxx	Pxx	Output of multiplexing functions	
	Feature Name	Input/Output						SCI's output function	Outside SCI
P10	P10	Input	—	×	0	1	×	—	
		output	—	0	0	0	0/1	SCLK11/SCL11 =1	TMIOB1=0
		N-channel drain open circuit output		1	0	0	0/1		
	ANI9	analog input	—	×	1	1	×	×	—
	SLK11	Input	—	×	0	1	×	—	—
		output	—	0/1	0	0	1	—	TMIOB1=0
	SCL11	output	—	0/1	0	0	1	—	TMIOB1=0
	TMIOB1	Input	—	×	0	1	×	—	—
		output	—	0	0	0	0	SCLK11/SCL11 =1	—
	TMIOD1	Input	PIOR37=1, PIOR36=0	×	0	1	×	—	—
		output		0	0	0	0	SCLK11/SCL11 =1	—
P11	P11	Input	—	×	0	1	×	—	
		output	—	0	0	0	0/1	SDA11=1	TMIOD1=0
		N-channel drain open circuit output		1	0	0	0/1		
	ANI8	analog input	—	×	1	1	×	×	—
	SI11	Input	—	×	0	1	×	—	—
	SDA11	Input/Output	—	1	0	0	1	—	TMIOD1=0
	RxD0	Input	PIOR35=1	×	0	1	1	—	TMIOD1=0
	TMIOD1	Input	—	×	0	1	×	—	—
		output	—	0	0	0	0	SDA11=1	—
	TMIA1	Input	PIOR37=0, PIOR36=1	×	0	1	×	—	—
		output		0	0	0	0	SDA11=1	—
	TMIOC1	Input	PIOR37=1, PIOR36=0	×	0	1	×	—	—
		output		0	0	0	0	SDA11=1	—
P12	P12	Input	—	—	—	1	×	—	—
		output	—	—	—	0	0/1	SDO11=0	TMIOA1=0
	SDO11	output	—	—	—	0	1	—	TMIOA1=0
	TxD0	output	PIOR35=1	0/1	—	0	1	—	TMIOA1=0
	TMIA1	Input	—	×	—	1	×	—	—
		output		0	—	0	0	SDO11=0	—
	TMIOB0	Input	PIOR37=0, PIOR36=1	—	—	1	×	—	—

		output		—	—	0	0	SDO11=0	—
	TMIOB1	Input	PIOR37=1, PIOR36=0	—	—	1	×	—	—
		output		—	—	0	0	SDO11=0	—
	(INTP5)	Input	PIOR04=1	—	—	1	×	—	—

Table 2-6: Example of Register Setting When Using the P10~P17 Pin Function (2/3)

name of the feet	Functions Used		PIORx	POMxx	PMCxx	PMxx	Pxx	Output of multiplexing functions	
	Feature Name	Input/Output						SCI's output function	Outside SCI
P13	P13	Input	—	×	—	1	×	—	—
		output	×	0	—	0	0/1	TxD2/SDO20=1	TMIOC1=0
		N-channel drain open circuit output		1	—	0	0/1		
	TxD2	output	PIOR01=0	0/1	—	0	1	—	TMIOC1=0
	SDO20	output	PIOR01=0	0/1	—	0	1	—	TMIOC1=0
	TMIOC1	Input	—	×	—	1	×	—	—
		output	—	0	—	0	0	TxD2/SDO20=1	—
	TMIOD1	Input	PIOR37=0, PIOR36=1	×	—	1	×	—	—
		output		0	—	0	0	TxD2/SDO20=1	—
	TMIA1	Input	PIOR37=1, PIOR36=0	×	—	1	×	—	—
		output		0	—	0	0	TxD2/SDO20=1	—
P14	P14	Input	—	×	—	1	×	—	—
		output	×	0	—	0	0/1	SDA20=1	TMIOB 0=0, (SCLA0)=0
		N-channel drain open circuit output		1	—	0	0/1		
	RxD2	Input	PIOR01=0	×	—	1	×	—	—
	STI20	Input	PIOR01=0	×	—	1	×	—	—
	SDA20	Input/Output	PIOR01=0	1	—	0	1	—	TMIOB 0=0, (SCLA0)=0
	TMIOB0	Input	—	×	—	1	×	—	—
		output	—	1	—	0	0	SDA20=1	(SCLA0)=0
	TMIOC1	Input	PIOR37=0, PIOR36=1	×	—	1	×	—	—
		output		1	—	0	0	SDA20=1	(SCLA0)=0
	TMIOB0	Input	PIOR37=1, PIOR36=0	×	—	1	×	—	—
		output		1	—	0	0	SDA20=1	(SCLA0)=0
	(SCLA0)	Input/Output	PIOR02=1	1	—	0	0	SDA20=1	TMIOB0=0

Table 2-6: Example of Register Setting When Using the P10~P17 Pin Function (3/3)

name of the feet	Functions Used		PIORx	POMxx	PMCxx	PMxx	Pxx	Output of multiplexing functions	
	Feature Name	Input/Output						SCI's output function	Outside SCI
P15	P15	Input	—	×	—	1	×	—	—
		output	×	0	—	0	0/1	SCLK20/SCL20 =1	TMIOD 0=0, (SDAA0)=0, CLKBUZ1=0
		N-channel drain open-circuit output		1	—	0	0/1		
	SLK20	Input	PIOR01=0	×	—	1	×	—	—
		output		0/1	—	0	1	—	TMIOD 0=0, (SDAA0)=0, CLKBUZ1=0
	SCL20	output	PIOR01=0	0/1	—	0	1	—	
	TMIOD0	Input	—	×	—	1	×	—	—
		output	—	0	—	0	0	SCLK20/SCL20 =1	(SDAA0)=0, CLKBUZ1=0
	TMIOB0	Input	PIOR37=1, PIOR36=0	×	—	1	×	—	—
		output		0	—	0	0	SCLK20/SCL20 =1	(SDAA0)=0, CLKBUZ1=0
	(SDAA0)	Input/Output	PIOR02=1	1	—	0	0	SCLK20/SCL20 =1	TMIOD 0=0, CLKBUZ1=0
P16	P16	Input	—	—	—	1	×	—	—
		output	—	—	—	0	0/1	—	TO01=0, TMIOC0=0, (TMIOA1)=0
	TI01	Input	—	—	—	1	×	—	—
	TO01	output	—	—	—	0	0	—	TMIOC0=0, (TMIOA1)=0
	INTP5	Input	PIOR04=0	—	—	1	×	—	—
	TMIOC0	Input	—	—	—	1	×	—	—
		output	—	—	—	0	0	—	TO01=0, (TMIOA1)=0
	(SDI00)	Input	PIOR01=1	—	—	1	×	—	—
	(RxD0)	Input	PIOR01=1, PIOR06=0	—	—	1	×	—	—
	(TMIOA1)	Input	PIOR22=1	×	—	1	×	—	—
		output		—	—	0	0	—	TO01=0, TMIOC0=0
P17	P17	Input	—	×	—	1	×	—	—
		output	×	0	—	0	0/1	(TxD0)/(SDO00) =1	TO02=0, TMIOA0=0, (TMIOD0)=0
		N-channel drain open-circuit output		1	—	0	0/1		
	TI02	Input	—	×	—	1	×	—	—
	TO02	output	×	0	—	0	0	(TxD0)/(SDO00) =1	TMIOA0=0, (TMIOD0)=0
	TMIA0	Input	—	×	—	1	×	—	—
		output	×	0	—	0	0	(TxD0)/(SDO00) =1	TO02=0, (TMIOD0)=0
	TMCLK	Input	—	×	—	1	×	—	—
	(SDO00)	output	PIOR01=1	0/1	—	0	1	—	TO02=0, TMIOA0=0, (TMIOD0)=0
	(TxD0)	output	PIOR01=1, PIOR06=0	0/1	—	0	1	—	TO02=0, TMIOA0=0, (TMIOD0)=0
	(TMIOD0)	Input	PIOR26=1	×	—	1	×	—	—
		output		0	—	0	0	(TxD0)/(SDO00) =1	TO02=0, TMIOA0=0

Table 2-7: Example of Register Setting When Using the P20~P27 Pin Function

name of the feet	Functions Used		PIORx	PMCxx	PMxx	Pxx
	Feature Name	Input/Output				
P20	P20	Input	—	0	1	×
		output	—	0	0	0/1
	ANI0	analog input	—	1	1	×
	AVREFP	reference voltage input	—	1	1	×
	VCIN12	analog input	—	1	1	×
	(INTP11)	Input	PIOR07=1	0	1	1
P21	P21	Input	—	0	1	×
		output	—	0	0	0/1
	ANI1	analog input	—	1	1	×
	AVREFM	reference voltage input	—	1	1	×
	VCIN13	analog input	—	1	1	×
P22	P22	Input	—	0	1	×
		output	—	0	0	0/1
	ANI2	analog input	—	1	1	×
	ANO0	analog output	—	1	1	×
	VCIN0	analog input	—	1	1	×
	PGA0IN	analog input	—	1	1	×
P23	P23	Input	—	0	1	×
		output	—	0	0	0/1
	ANI3	analog input	—	1	1	×
	ANO1	analog output	—	1	1	×
	PGA0GND	analog input	—	1	1	×
P24	P24	Input	—	0	1	×
		output	—	0	0	0/1
	ANI4	analog input	—	1	1	×
	PGA1IN	analog input	—	1	1	×
P25	P25	Input	—	0	1	×
		output	—	0	0	0/1
	ANI5	analog input	—	1	1	×
	PGA1GND	analog input	—	1	1	×
P26	P26	Input	—	0	1	×
		output	—	0	0	0/1
	ANI6	analog input	—	1	1	×
P27	P27	Input	—	0	1	×
		output	—	0	0	0/1
	ANI7	analog input	—	1	1	×

Table 2-8: Example of Register Setting When Using the P30~P43 Pin Function

name of the feet	Functions Used		PIORx	POMxx	PMCxx	PMxx	Pxx	Output of multiplexing functions	
	Feature Name	Input/Output						SCI's output function	Outside SCI
P30	P30	Input	—	×	—	1	×	—	—
		output	×	0	—	0	0/1	SLK00/SCL00=1	RTC1HZ=0, TAO=0, (TMIOB1)=0
		N-channel drain open circuit output		1	—	0	0/1		
	INTP3	Input	PIOR00=0, PIOR05=0	×	—	1	×	—	—
	RTC1HZ	output	×	0	—	0	0	SLK00/SCL00=1	TAO=0, (TMIOB1)=0
	SCLK00	Input	PIOR01=0	×	—	1	×	—	—
		output		0/1	—	0	1	—	RTC1HZ=0, TAO=0, (TMIOB1)=0
	SCL00	output	PIOR01=0	0/1	—	0	0	—	RTC1HZ=0, TAO=0, (TMIOB1)=0
	TAO	output	PIOR13, PIOR12=00B	0	—	0	0	SCLK00/SCL00 =1	RTC1HZ=0, (TMIOB1)=0
	(TMIOB1)	Input	PIOR23=1	×	—	1	×	—	—
		output		0	—	0	0	SLK00/SCL00=1	RTC1HZ=0, TAO=0
P31	P31	Input	—	—	—	1	×	—	—
		output	×	—	—	0	0/1	—	TO03=0, VCOUT1=0, (CLKBUZ0)=0, (TAIO)=0
	TI03	Input	—	—	—	1	×	—	—
	TO03	output	×	—	—	0	0	—	VCOUT1=0, (CLKBUZ0)=0, (TEIO)=0
	INTP4	Input	PIOR00=0 note 13, PIOR05=0 note 14	—	—	1	×	—	—
	(TAIO)	Input	PIOR11, PIOR10=01B	—	—	1	×	—	—
		output		—	—	0	0	—	TO03=0, VCOUT1=0, (CLKBUZ0)=0
	(CLKBUZ0)	output	PIOR03=1	—	—	0	0	—	TO03=0,(TEIO)=0,VCOUT1=0
	VCOUT1	output	PIOR21=0	—	—	0	0	—	TO03=0, (TAIO)=0, (CLKBUZ0)=0
P40	P40	Input	—	×	—	1	×	—	—
		output	×	—	—	0	0/1	—	—
	SWDIO	Input	—	×	—	1	×	—	—
		output	×	—	—	0	×	—	—
P41	P41	Input	—	—	—	1	×	—	—
		output	×	—	—	0	0/1	—	(TAIO)=0
	(TAIO)	Input	PIOR11, PIOR10=10B	—	—	1	×	—	—
		output		—	—	0	0	—	—
P42	P42	Input	—	—	—	1	×	—	—
		output	×	—	—	0	0/1	—	—
	(INTP8)	Input	PIOR00=1, PIOR07=0	—	—	1	×	—	—
P43	P43	Input	—	—	—	1	×	—	—
		output	×	—	—	0	0/1	—	—
	(INTP9)	Input	PIOR00=1	—	—	1	×	—	—

Table 2-9: Example of Register Setting When Using the P50~P55 Pin Function

name of the feet	Functions Used		PIORx	POMxx	PMCxx	PMxx	Pxx	Output of multiplexing functions	
	Feature Name	Input/Output						Output Function of SCI/CAN	Beyond SCI/CAN
P50	P50	Input	—	×	—	1	×	—	—
		output	×	0	—	0	0/1	SDA00=1	TBIO0=0, (TAO)=0, (TMIOC1)=0
		N-channel drain open circuit output		1	—	0	0/1		
	INTP1	Input	PIOR00=0	×	—	1	×	—	—
	STI00	Input	PIOR01=0	×	—	1	×	—	—
	RxD0	Input	PIOR01=0	×	—	1	×	—	—
	SDA00	Input/Output	PIOR01=0	1	—	0	1	—	TBIO0=0, (TAO)=0, (TMIOC1)=0
	(CRxD0)	Input	PIOR33=1	×	—	1	×	—	—
	TBIO0	Input	—	×	—	1	×	—	—
		output	—	0	—	0	0	SDA00=1	(TAO)=0, (TMIOC1)=0
	(TAO)	output	PIOR13, PIOR12=01B	0	—	0	0	SDA00=1	TBIO0=0, (TMIOC1)=0
	(TMIOC1)	Input	PIOR24=1	×	—	1	×	—	—
		output		0	—	0	0	SDA00=1	TBIO0=0, (TAO)=0
P51	P51	Input	—	×	—	1	×	—	—
		output	—	0	—	0	0/1	TxD0/SDO00=1	TBIO1=0, (TMIOD1)=0
		N-channel drain open circuit output		1	—	0	0/1		
	INTP2	Input	PIOR01=0	×	—	1	×	—	—
	SDO00	output	PIOR01=0	0/1	—	0	1	—	TBIO1=0, (TMIOD1)=0
	TxD0	output	PIOR01=0	0/1	—	0	1	—	TBIO1=0, (TMIOD1)=0
	(CTxD0)	output	PIOR33=1	0/1	—	0	1	TxD0/SDO00=1	TBIO1=0, (TMIOD1)=0
	TBIO1	Input	—	×	—	1	×	—	—
		output		0	—	0	0	TxD0/SDO00=1	(TMIOD1)=0
	(TMIOD1)	Input	PIOR25=1	×	—	1	×	—	—
		output		0	—	0	0	TxD0/SDO00=1	TBIO1=0
P52	P52	Input	—	×	—	1	×	—	—
		output	—	0	—	0	0/1	—	—
		N-channel drain open circuit output		1	—	0	0/1		
	(INTP1)	Input	PIOR00=1	×	—	1	×	—	—
P53	P53	Input	—	×	—	1	×	—	—
		output	×	0	—	0	0/1	—	—
		N-channel drain open circuit output		1	—	0	0/1		
	(INTP2)	Input	PIOR00=1	×	—	1	×	—	—
P54	P54	Input	—	×	—	1	×	—	—
		output	—	0	—	0	0/1	—	—
		N-channel drain open circuit output		1	—	0	0/1		
	(INTP3)	Input	PIOR00=1	×	—	1	×	—	—
P55	P55	Input	—	×	—	1	×	—	—
		output	—	0	—	0	0/1	(SCLK00)=1	(CLKBUZ1)=0
		N-channel drain open circuit output		1	—	0	0/1		
	(INTP4)	Input	PIOR00=1	×	—	1	×	—	—
	(CLKBUZ1)	output	PIOR04=1	0	—	0	0	(SCLK00)=1	—
	(SCLK00)	Input	PIOR01=1	×	—	1	×	—	—
		output		0/1	—	0	1	—	(CLKBUZ1)=0

Table 2-10: Example of Register Setting When Using the P60~P63 Pin Function

name of the feet	Functions Used		PIORx	POMxx	PMCxx	PMxx	Pxx	Output of multiplexing functions	
	Feature Name	Input/Output						SCI's output function	Outside SCI
P60	P60	Input	—	—	—	1	×	—	—
		N-channel drain open circuit output (6V Withstand Voltage)	×	—	—	0	0/1	—	SCLA0=0
	SCLA0	Input/Output	PIOR02=0	—	—	0	0	—	—
P61	P61	Input	—	—	—	1	×	—	—
		N-channel drain open circuit output (6V Withstand Voltage)	×	—	—	0	0/1	—	SDAA0=0
	SDAA0	Input/Output	PIOR02=0	—	—	0	0	—	—
P62	P62	Input	—	—	—	1	×	—	—
		N-channel drain open circuit output (6V Withstand Voltage)	×	—	—	0	0/1	—	—
	SS00	Input	—	—	—	1	×	—	—
P63	P63	Input	—	—	—	1	×	—	—
		N-channel drain open circuit output (6V Withstand Voltage)	×	—	—	0	0/1	—	—

Table 2-11: Example of Register Setting When Using the P70~P77 Pin Function

name of the feet	Functions Used		PIORx	POMxx	PMCxx	PMxx	Pxx	Output of multiplexing functions	
	Feature Name	Input/Output						SCI's output function	Outside SCI
P70	P70	Input	—	—	—	1	×	—	—
		output	—	—	—	0	0/1	SCLK21/SCL21=1	(VCOUT1)=0
	KR0	Input	—	—	—	1	×	—	—
	SLK21	Input	—	—	—	1	×	—	—
		output	—	—	—	0	1	—	(VCOUT1)=0
	SCL21	output	—	—	—	0	1	—	(VCOUT1)=0
	(VCOUT1)	output	PIOR21=1	—	—	0	0	SCLK21/SCL21=1	—
P71	P71	Input	—	×	—	1	×	—	—
		output	—	0	—	0	0/1	SDA21=1	(VCOUT0)=0
		N-channel drain open circuit output		1	—	0	0/1		
	KR1	Input	—	×	—	1	×	—	—
	STI21	Input	—	×	—	1	×	—	—
	SDA21	Input/Output	—	0/1	—	0	1	—	(VCOUT0)=0
	(VCOUT0)	output	PIOR20=1	—	—	0	0	SDA21=1	—
P72	P72	Input	—	—	—	1	×	—	—
		output	—	—	—	0	0/1	SDO21=1	—
	KR2	Input	—	—	—	1	×	—	—
	SDO21	output	—	—	—	0	1	—	—
P73	P73	Input	—	—	—	1	×	—	—
		output	—	—	—	0	0/1	SDO01=1	—
	KR3	Input	—	—	—	1	×	—	—
	SDO01	output	—	—	—	0	1	—	—
P74	P74	Input	—	×	—	1	×	—	—
		output	—	0	—	0	0/1	SDA01=1	—
		N-channel drain open circuit output		1	—	0	0/1		
	KR4	Input	—	—	—	1	×	—	—
	INTP8	Input	PIOR00=0, PIOR07=0	—	—	1	×	—	—
	STI01	Input	—	—	—	1	×	—	—
	SDA01	Input/Output	—	0/1	—	0	1	—	—
P75	P75	Input	—	—	—	1	×	—	—
		output	—	—	—	0	0/1	SLK01/SCL01=1	—
	KR5	Input	—	—	—	1	×	—	—
	INTP9	Input	PIOR00=0	—	—	1	×	—	—
	SLK01	Input	—	—	—	1	×	—	—
		output	—	—	—	0	1	—	—
	SCL01	output	—	—	—	0	1	—	—
P76	P76	Input	—	—	—	1	×	—	—
		output	—	—	—	0	0/1	—	—
	KR6	Input	—	—	—	1	×	—	—
	INTP10	Input	PIOR01=0, PIOR07=0	—	—	1	×	—	—
	(RxD2)	Input	PIOR01=1	—	—	1	×	—	—
P77	P77	Input	—	—	—	1	×	—	—
		output	×	—	—	0	0/1	(TxD2)=1	—

	KR7	Input	—	—	—	1	×	—	—
	INTP11	Input	PIOR01=0, PIOR07=0	—	—	1	×	—	—
	(TxD2)	output	PIOR01=1	—	—	0	1	—	—

Figure 2-12: Example of Register Setting When Using the P120 Pin Feature

name of the feet	Functions Used		PIORx	POMxx	PMCxx	PMxx	Pxx	Output of multiplexing functions	
	Feature Name	Input/Output						SCI output function	Outside SCI
P120	P120	Input	—	—	0	1	×	—	—
		output	—	—	0	0	0/1	—	—
	ANI14	analog input	—	—	1	1	×	—	—
	VCOUT0	output	PIOR20=0	—	0	0	0	—	—

Figure 2-13: Example of Register Setting When Using the P121~P124 Pin Function

name of the feet	Functions Used		CMC	Pxx
	Feature Name	Input/Output	(EXCLK, OSCSEL, EXCLKS, OSCELS)	
P121	P121	Input	00xx/10xx/11xx	×
	X1	—	01xx	—
P122	P122	Input	00xx/10xx/11xx	×
	X2	—	01xx	—
	EXCLK	Input	11xx	—
P123	P123	Input	xx00/xx10/xx11	×
	XT1	—	xx01	—
P124	P124	Input	xx00/xx10/xx11	×
	XT2	—	xx01	—
	EXCLKS	Input	xx11	—

Figure 2-14: Example of Register Setting When Using the P130~P147 Pin Function

name of the feet	Functions Used		PIORx	POMxx	PMCxx	PMxx	Pxx	Output of multiplexing functions	
	Feature Name	Input/Output						SCI output function	Outside SCI
P130	P130	output	—	—	—	—	0/1	—	—
P136	P136	Input	—	—	—	1	×	—	—
		output	—	—	—	0	0/1	—	—
	INTP0	Input	—	—	—	1	×	—	—
P137	P137	Input	—	—	—	1	×	—	—
		output	—	—	—	0	0/1	—	—
	SWCLK	Input	—	—	—	1	×	—	—
P140	P140	Input	—	—	—	1	×	—	—
		output	—	—	—	0	0/1	—	CLKBUZ0=0
	CLKBUZ0	output	PIOR03=0	—	—	0	0	—	—
	INTP6	Input	—	—	—	1	×	—	—
P141	P141	Input	—	—	—	1	×	—	—
		output	—	—	—	0	0/1	—	CLKBUZ1=0
	CLKBUZ1	output	PIOR04=0	—	—	0	0	—	—
	INTP7	Input	—	—	—	1	×	—	—
P146	P146	Input	—	—	0	1	×	—	—
		output	—	—	0	0	0/1	—	—
	ANI15	analog input	—	—	1	1	×	—	—
P147	P147	Input	—	—	0	1	×	—	—
		output	—	—	0	0	0/1	—	—
	ANI12	analog input	—	—	1	1	×	—	—
	VREF0	analog input	—	—	1	1	×	—	—

Chapter 3 system structure

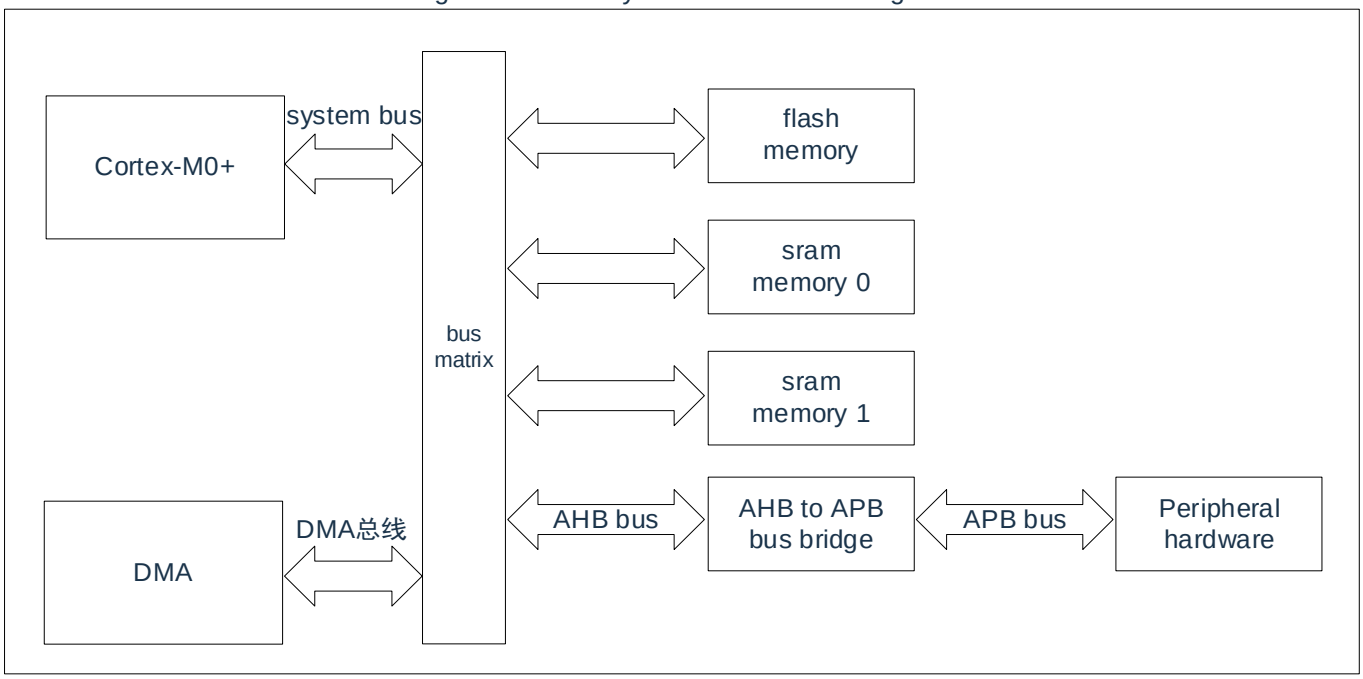
3.1 Overview

This product system consists of the following components:

- 2 AHB buses Master:
  - Cortex-M0+
  - Enhanced DMA

- 4 AHB buses Slaves:
  - FLASH Storage
  - SRAM Memory 0
  - SRAM Memory 1
  - AHB to APB Bridge, contains all APB interface peripherals

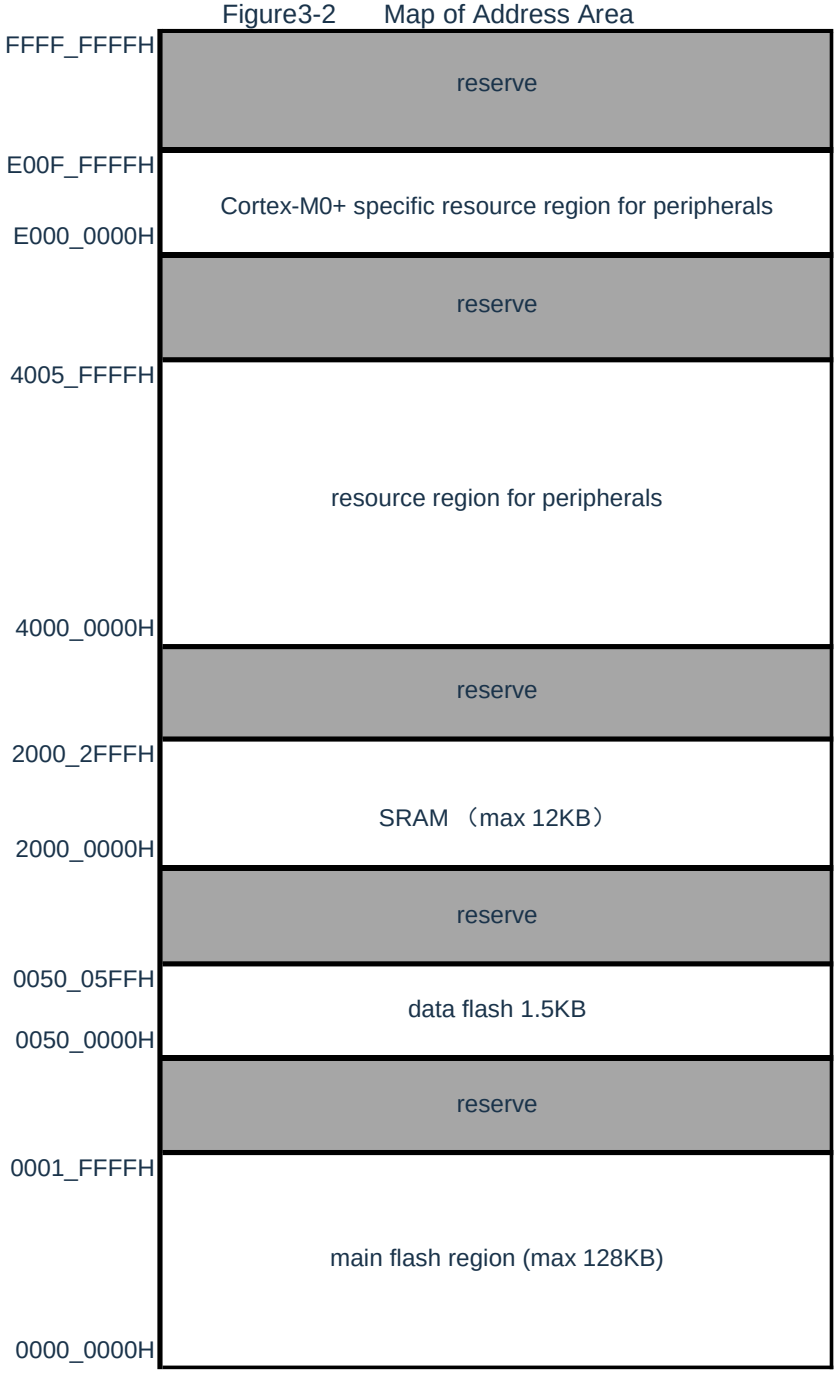
Figure3-1 System Architecture Diagram



- System bus: This bus connects the system bus (peripheral bus) of the Cortex-M0+ kernel to the bus matrix, which coordinates access between the kernel 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 bridgeTable3-1.

3.2 system address partitioning

Figure3-2 Map of Address Area



peripheral address assignment

Table3-1 Register group start address for peripheral

Start Address	peripheral	Remark
0x4000_000 - 0x4000_4FFF	Reserve	
0x4000_5000 - 0x4000_5FFF	DMA	
0x4000_6000 - 0x4000_6FFF	interrupt control	
0x4000_7000 - 0x4001_8FFF	Reserve	
0x4001_9000 - 0x4001_9FFF	MTB Control	See Chapter 1 Debug Support for more
0x4001_A000 - 0x4001_FFFF	Reserve	
0x4002_0000 - 0x4002_03FF	FLASH control	
0x4002_0400 - 0x4002_0FFF	clock control	
0x4002_1000 - 0x4002_1001	watchdog timer	
0x4002_1002 - 0x4002_1800	Reserve	
0x4002_1800 - 0x4002_1BFF	high speed CRC	See Chapter 31 for Safety Functions
0x4002_1C00 - 0x4002_1FFF	clock control	
0x4002_2000 - 0x4003_FFFF	Reserve	
0x4004_000 - 0x4004_0FFF	GPIO	
0x4004_1100 - 0x4004_19FF	serial communication unit	
0x4004_1A00 - 0x4004_1CFF	serial interface IICA	
0x4004_1D00 - 0x4004_1FFF	timer array	
0x4004_2000 - 0x4004_21FF	Reserve	
0x4004_2200 - 0x4004_25FF	Timer A	
0x4004_2600 - 0x4004_29FF	Timer B	
0x4004_2A00 - 0x4004_2BFF	Timer M	
0x4004_2C00 - 0x4004_2FFF	timer C	
0x4004_3000 - 0x4004_31FF	Reserve	
0x4004_3200 - 0x4004_32FF	universal CRC	See Chapter 31 for Safety Functions
0x4004_3300 - 0x4004_33FF	Reserve	
0x4004_3400 - 0x4004_37FF	linkage controller	
0x4004_3800 - 0x4004_3BFF	comparator	
0x4004_3C00 - 0x4004_3FFF	Timer M	
0x4004_4000 - 0x4004_46FF	IrDA	
0x4004_4700 - 0x4004_4AFF	DA converter	
0x4004_4B00 - 0x4004_4EFF	key break	
0x4004_4F00 - 0x4004_4FFF	real-time clock	
0x4004_5000 - 0x4004_53FF	AD converter	
0x4004_5400 - 0x4004_5AFF	CAN controller	
0x4004_5B00 - 0x4004_5BFF	interrupt control	
0x4008_000 - 0x4008_01FF	hardware divider	
0x4008_0200 - 0xDFFF_FFFF	Reserve	

Chapter 4 clock generator

The presence of resonator connection pin/external clock input pin for the main system clock and the resonator connection pin/external clock input pin for the secondary system clock are different among products.

	24. 32 pin products	36,40, 48,52,64 pin products
X1, X2 pins	○	○
EXCLK Pin	○	○
XT1, XT2 Pin	—	○
EXCLKS Pin	—	○

4.1 Function 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 oscillating circuit

The X1 pin and X2 pin can oscillate a clock with fX=1 to 20MHz by connecting the X1 pin and the X2 pin to the resonator.

② high speed internal oscillator (high-speed OCO)

Can oscillate from fHOCO=64MHz, 48MHz, 32MHz, 24MHz, 16MHz, 12MHz, 8MHz, 6MHz, 4MHz, 3MHz, 2MHz and 1MHz (TYP.) via option bytes (000C2H). When 64MHz is selected as fHOCO, the frequency division of fIH is 32MHz. When 48 MHz or a lower frequency is selected as the fHOCO, the fIH is not divided and is the same frequency as the fHOCO. After the reset is removed, the CPU must start running with this high speed internal oscillator clock. The oscillation can be stopped by Inputing a deep sleep mode or by setting a HIOSTOP bit (bit0 of a CSC register). The frequency at which option byte settings can be changed through a HOCODIV (Frequency Selection Register) of a high speed internal oscillator. For frequency settings, refer to "Figure4-10 Format of Frequency Selective Register (HOCODIV) for High-speed Internal Oscillator". The oscillation frequencies that can be set by the high speed internal oscillator are shown below (types that can be selected by option bytes and the high speed internal oscillator frequency selection register (HOCODIV)).

In addition, the external main system clock (fEX=1~20MHz) can be provided by the EXCLK/X2/P122 pin, and the input of the external main system clock can be set invalid by entering deep sleep mode or setting MSTOP bit.

A high-speed system clock (X1 clock or an external main system clock) and a high speed internal oscillator clock can be switched by setting a MCM0 bit (bit4 of a system clock control register (CKC)).

(2) sub-system clock

· XT1 oscillating circuit

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

In addition, the external sub-system clock (f_{EXS}=32.768kHz) can be provided by the EXCLKS/XT2/P124 pin, and the input of the external sub-system clock can be set invalid by setting the XTSTOP bit.

(3) Low speed internal oscillator clock (low speed OCO)

It can make the clock oscillate at fIL=15 kHz (TYP.).

A low speed internal oscillator clock cannot be used as a CPU clock.

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

Only the following peripheral hardware can run through a low-speed internal oscillator clock:

- watchdog timer
- real-time clock
- 15-bit interval timer
- Timer A

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

However, the low-speed internal 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.

Note: The low-speed internal 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.

Notes: f_X : X1 clock oscillation frequency

f_{HOCO} : Clock frequency of high speed internal oscillator (up to 64MHz)

f_{IH} : Clock frequency (maximum 48 MHz) Note for high-speed internal oscillator

f_{EX} : External master clock frequency

f_{XT} :XT1 clock oscillation frequency

f_{EXS} : external sub-system clock frequency

f_{IL} : Clock frequency of low speed internal oscillator

Note: Controlled by hardware, when fHOCO is set to 64MHz, the clock frequency is the same as fHOCO divided by 2; when it is set to 48MHz or lower, the clock frequency is the same as fHOCO. To supply 64MHz to Timer M, fCLK must be set to fIH.

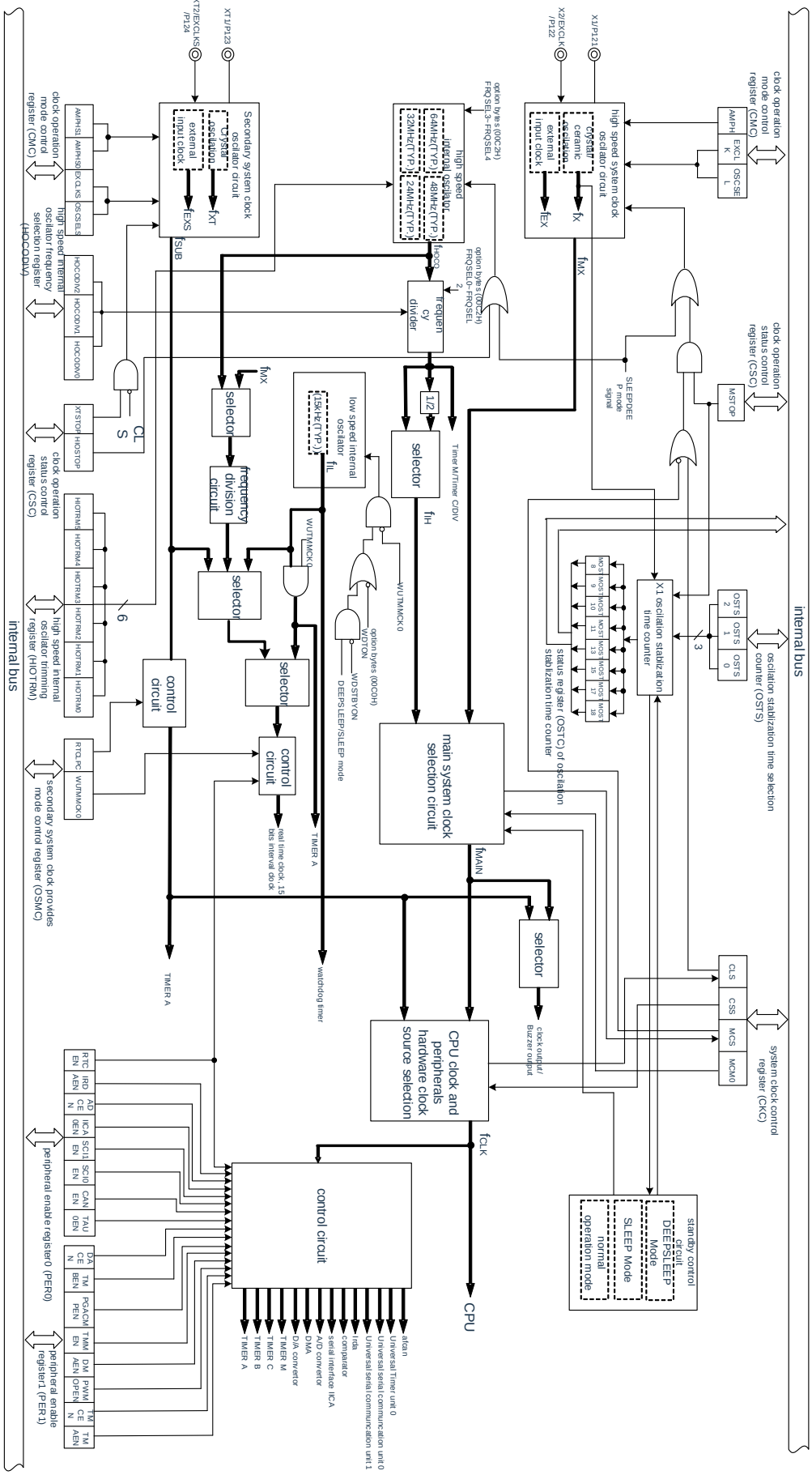
4.2 Structure of clock generating circuit

The clock generating circuit is composed of the following hardware.

Table4-1 Structure of clock generating circuit

Project	structure
control register	Clock Run 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 internal oscillator frequency selection register (HOCODIV) High-speed internal oscillator trim register (HIOTRM)
oscillating circuit	X1 oscillating circuit XT1 oscillating circuit high speed internal oscillator low-speed internal oscillator

Figure4-1 Block diagram of clock generating circuit



Remark
fX : X1 clock oscillation frequency
fHOCO : Clock frequency of high speed internal oscillator (maximum 64MHz)
fIH : Clock frequency (maximum 48 MHz) of high-speed internal oscillator
fEX : External master clock frequency
fMX : high speed system clock frequency
fMAIN : main system clock frequency
fXT :XT1 clock oscillation frequency
fEXS : external sub-system clock frequency
fSUB : sub-system clock frequency
fCLK : Clock frequency of the CPU/peripheral hardware
fIL : Clock frequency of low speed internal oscillator

Note: Controlled by hardware, when f_{HOCO} is set to 64MHz, the clock frequency is the same as f_{HOCO} divided by 2; when it is set to 48MHz or lower, the clock frequency is the same as f_{HOCO}. To supply 64MHz to Timer M, f_{CLK} must be set to f_{IH}.

4.3 Register 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 internal oscillator frequency selection register (HOCODIV)
- High-speed internal oscillator trim register (HIOTRM)

Note: The assigned registers and bits differ depending on the product. You must set an initial value for unassigned bits.

4.3.1 Clock Run Mode Control Register (CMC)

This is a register that sets the operation mode of the X1/P121, X2/EXCLK/P122, XT1/P123, XT2/EXCLKS/P124 pin and selects the gain of the oscillating circuit.

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

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

Figure4-2 Format of Clock Run 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/P121 Pin	X2/EXCLK/P122 Pin
0	0	input port mode	input port	
0	1	X1 oscillation mode	Connect a crystal or ceramic resonator.	
1	0	port mode	input port	
1	1	external clock input mode	input port	external clock input

EXCLKS	OSCELS	Secondary system clock pin operation mode	XT1/P123 Pin	XT2/EXCLKS/P124 pin
0	0	input port mode	input port	
0	1	XT1 oscillation mode	Connect a crystal resonator.	
1	0	input port mode	input port	
1	1	external clock input mode	input 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	$1MHz \leqslant f_X \leqslant 10MHz$
1	$10MHz < f_X \leqslant 20MHz$

Note: The EXCLKS bit, OSCSELS bit, AMPHS1 bit, and AMPHS0 bit are initialized only when the power is reset, and remain unchanged when the other reset is reset.

Note 1. After the reset is removed, the CMC register can only be written 1 time through the 8-bit memory instruction. When using CMC registers with an initial value (00H), to prevent malfunctions when programs are out of control (if values other than "00H" cannot be recovered)

2. The CMC register must be set after the reset is removed and before X1 or XT1 oscillations are started by setting clock run status control registers.

3. The AMPH position "1" must be set when the X1 clock oscillates above 10 MHz.

4. AMPH bits, AMPHS1 bits, and AMPHS0 bits must be set after the reset is removed and under fIH as fCLK (switch fCLK to fMX or prefSUB).

5. The stability time of fXT must be counted by software.

6. The upper limit of the system clock is 48MHz, but the upper limit of the X1 oscillator circuit is 20MHz.

Note: $_{fX}$: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 operation instruction.

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

Figure4-3 Format of the System Clock Control Register (CKC)

Address: 40020404H After reset: 00H R/W Note 1

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	Secondary System Clock (f_{SUB})

CSS Note 2	CPU/Peripheral Hardware Clock (f_{CLK}) Selection
0	Main System Clock (f_{MAIN})
1	Secondary System Clock (f_{SUB})

MCS	Status of the main system clock (fMAIN)
0	high speed internal oscillator clock (f_{IH})
1	High speed system clock (f_{MX})

MCM0 Note 2	Main System Clock (f_{MAIN}) Operation Control
0	A high speed internal oscillator clock (f_{IH}) is selected as the main system clock (f_{MAIN}).
1	Select the high speed system clock (f_{MX}) as the primary system clock (f_{MAIN}).

Note: 1.bit7 and bit5 are read-only bits.

2. Disable changing the value of the MCM0 bit in the state with CSS position "1".

Note: f_{HOCO} : Clock frequency of high speed internal oscillator (maximum 64MHz)

f_{IH} : Clock frequency (maximum 48 MHz) of high-speed internal oscillator

f_{MX} : high speed system clock frequency

f_{MAIN} : main system clock frequency

f_{SUB} : sub-system clock frequency

Note: Hardware controls the clock frequency to be the same as the 2th of f_{HOCO} when f_{HOCO} is set to 64 MHz. Set to 48MHz or lower, to have the same clock frequency as f_{HOCO}. f_{CLK} must be set to f_{IH} to supply timer M with 64 MHz.

Note: 1. bit0~3 must be set to 0.

2. Provides CSS bit setting clocks for the CPU and peripheral hardware. If you change the CPU clock, change the peripheral hardware clock at the same time (except for real-time clocks, 15-bit interval timers, clock output/buzzer output, and watchdog timer). Therefore, if you want to change the clock on the CPU/peripheral hardware, you must stop the peripheral functions.

3. If the secondary system clock is used as the peripheral hardware clock, the A/D converter and IICA cannot be

guaranteed. For operation characteristics of the peripheral hardware, refer to the electrical characteristics of the sections and datasheet for each peripheral hardware.

4. To select fHOCO as the counter source for timer M, you must set fCLK to fIH before setting bit4(TMMEN) of peripheral enable register 1 (PER1. If you want to change fCLK to a clock other than fIH, you must change it after clearing the bit4(TMMEN) of Peripheral enable register 1 (PER).

4.3.3 Clock Operational Status Control Register (CSC)

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

After the reset signal is generated, the value of this register changes to "C0H".

Figure4-4 Format of Clock Operational 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	input port mode
0	Operation of X1 oscillator circuit	External clock valid for EXCLK pin	input port
1	The X1 oscillation circuit stops	The external clock for the EXCLK pin is invalid	

XTSTOP	Running Control of Sub-system Clock		
	XT1 oscillation mode	external clock input mode	input port mode
0	XT1 Oscillation Circuit Operation	External clock valid for EXCLKS pin	input port
1	XT1 oscillation circuit stop	The external clock for the EXCLKS pin is invalid	

HIOSTOP	Operation control of high speed internal oscillator clock
0	high speed internal oscillator operation
1	high speed internal oscillator stop

Note:

1. After the reset is removed, the CSC register must be set after setting the Clock Run Mode Control Register (CMC).
2. After the reset is removed and before the MSTOP position "0", an oscillatory stabilization time selection register (OSTS) must be set. However, when using the OSTS register at the initial value, you do not need to set the OSTS register.
3. When X1 oscillation is started by setting the MSTOP bit, the oscillation stability time of the X1 clock must be confirmed by OSTC.
4. When you want to start XT1 oscillation by setting the XSTOP bit, you must wait for the oscillation stabilization time required by the secondary system clock through software.
5. The clock selected as the CPU/peripheral hardware clock ($_{fCLK}$) cannot be stopped $_{through}$ the CSC register.
6. Refer to Table 4-2 for register flag settings for stopping clock oscillation (invalid external clock input) and conditions before stopping.

Table4-2 Clock stopping method

clock	Condition before clock stops (invalid external clock input)	Set the flag for the CSC register
X1 clock external main system 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
XT1 clock external sub-system clock	The CPU/peripheral hardware clock runs at a clock other than the secondary system clock. (CLS=0)	XTSTOP=1
high speed internal oscillator clock	The CPU/peripheral hardware clock operates at a clock other than the high speed internal oscillator clock. (CLS=0 and MCS=1, or CLS=1)	HIOSTOP=1

4.3.4 State register of the oscillation stabilization time counter (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 internal oscillator clock or a sub-system clock and the oscillation of the X1 clock is started

When the CPU clock is a high speed internal 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 operation instruction.

By generating a reset signal, entering deep sleep mode or MSTOP bit (bit7 of clock running status control register (CSC) is 1.

Note: The oscillation steady-time counter starts counting when:

· When the X1 clock starts to oscillate (EXCLK, OSCSEL=0, 1→MSTOP=0)

· when deep sleep mode is removed

Figure4-5 The Format of State Register (OSTC) for Oscillatory Stable Time Counters

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	oscillatory steady-time state		
									f_X=10MHz	f_X=20MHz
0	0	0	0	0	0	0	0	Less than $28/f_X$	less than 25.6μs	less than 12.8μs
1	0	0	0	0	0	0	0	At least 28/fX	At least 25.6μs	At least 12.8μs
1	1	0	0	0	0	0	0	At least 29/fX	At least 51.2μs	At least 25.6μs
1	1	1	0	0	0	0	0	At least 210/fX	At least 102μs	At least 51.2μs
1	1	1	1	0	0	0	0	At least 211/fX	At least 204μs	At least 102μs
1	1	1	1	1	0	0	0	At least 213/fX	At least 819μs	At least 409μs
1	1	1	1	1	1	0	0	At least 215/fX	At least 3.27ms	At least 1.63ms
1	1	1	1	1	1	1	0	At least 217/fX	At least 13.1ms	At least 6.55ms
1	1	1	1	1	1	1	1	At least 218/fX	At least 26.2ms	At least 13.1ms

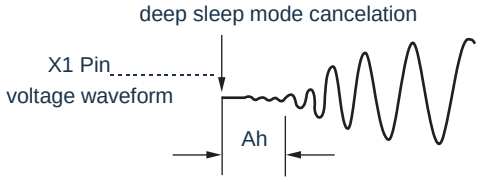
Note 1. After the time mentioned above, you change from MOST8 to "1" and remain in "1".

2. The oscillation stable time counter counts only within the oscillation stable time set by the OSTS. 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.

When the CPU clock is a high speed internal oscillator clock or a sub-system clock and the X1 clock is to start oscillating

When the CPU clock is a high-speed internal oscillator clock and is transferred to deep sleep mode and then released from deep sleep mode in the X1 clock oscillation state

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



Note: $_{fX}$:X1 clock oscillation frequency.

4.3.5 Oscillating Stable 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 internal oscillator clock or a sub-system clock to an X1 clock, or if the CPU clock is a high speed internal 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 operation instruction. After the reset signal is generated, the value of this register changes to "07H".

Figure4-6 Format of Oscillatory Stable 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 Stability Time		
				f_X=10MHz	f_X=20MHz
0	0	0	$28/f_X$	25.6μs	12.8μs
0	0	1	$29/f_X$	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

Note:1. To change the setting of the OSTS register, you must make the change before the MSTOP position of the Clock Run Status Control Register (CSC).

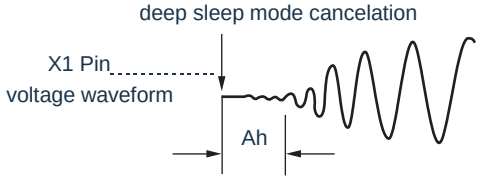
2. The oscillation stable time counter is counted only in that oscillation stable time set in the OSTS register.

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 internal oscillator clock or a sub-system clock and the X1 clock is to start oscillating

When the CPU clock is a high-speed internal oscillator clock and is transferred to deep sleep mode and then released from deep sleep mode in the X1 clock oscillation state

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



Note: 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 position '1' must be made 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
- afcan
- Timer4 0
- D/A converter
- Timer B
- comparator
- Timer M
- Enhanced DMA
- PWMOP
- timer C
- Timer A

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

After the reset signal is generated, the values of these registers change to '00H'.

Figure4-7 Format of Peripheral Enable Register 0 (PER0) (1/3)

Location: 40020420H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDEN	ADCEN	ICAHN	SCI1EN	SCI0EN	CAN0	TM40EN

RTCEN	Control of an input clock of a real-time clock (RTC) and a 15-bit interval timer
0	Stop provide an input clock. · Cannot write real time clock(RTC) and the SFR used by the and 15-bit interval timers. · The real-time clock (RTC) and 15-bit interval timer are reset.
1	Provides an input clock. · Read and write to the SFR used by the real-time clock (RTC) and 15-bit interval timer.

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

Figure4-7 Format of Peripheral Enable Register 0 (PER0) (2/3)

Location: 40020420H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDEN	ADCEN	ICAHN	SCI1EN	SCI0EN	CAN0	TM40E

IRDEN	Provides control of input clock for serial interface IICA1
0	Stop provide an input clock. · Cannot write the SFR used by IRDA. · IRDA is in a reset state.
1	Provides an input clock. · Can read and write SFRs used by IRDA.

ADCEN	Control of an input clock of an A/D converter is provided
0	Stop provide an input clock. · Cannot write the SFR used by the A/D converter. · The A/D converter is in a reset state.
1	Provides an input clock. · Read and write SFRs used by A/D converters.

IICA0EN	Provides control of input clock for serial interface IICA0
0	Stop provide an input clock. · Cannot write the SFR used by the serial interface IICA0. · Serial interface IICA0 is in a reset state.
1	Provides an input clock. · SFRs that can read and write to the serial interface IICA0.

SCI1EN	Control of an input clock of a universal serial communication unit 1 is provided
0	Stop provide an input clock. · Cannot write the SFR used by Universal Serial Communication Unit 1. · The universal serial communication unit 1 is in a reset state.
1	Provides an input clock. · Read and write the SFR used by the Universal Serial Communication Unit 1.

SCI0EN	Control of an input clock of a universal serial communication unit 0 is provided
0	Stop provide an input clock. · Cannot write the SFR used by the Universal Serial Communication Unit 0. · Universal serial communication unit 0 is in reset state.
1	Provides an input clock. · Read and write the SFR used by the Universal Serial Communication Unit 0.

Figure4-7 Format of Peripheral Enable Register 0 (PER0) (3/3)

Location: 40020420H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDEN	ADCEN	ICAHN	SCI1EN	SCI0EN	CAN0	TM40EN

CAN0	Control of input clock of CAN module
0	Stop provide an input clock. · Cannot write the SFR used by CAN0. ·CAN0 is in a reset state.
1	Provides an input clock. · Read and write SFRs used by CAN0.

TM40EN	Provides control of the input clock of the Timer4
0	Stop provide an input clock. · Cannot write the SFR used by the universal timer unit 0. · The universal timer unit 0 is in a reset state.
1	Provides an input clock. · Read and write the SFR used by the universal timer unit 0.

Figure4-8 Format (1/2) of the Peripheral Admission Register 1 (PER1)

Address: 4002081AH After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN	TMMEN Note 1	DMAEN	PWMPEN	TMCEN Note 2	TMAEN

DACEN	Provides control of the input clock of a D/A converter
0	Stop provide an input clock. · You cannot write the SFR used by the D/A converter. · The D/A converter is in a reset state.
1	Provides an input clock. · Read and write SFRs used by D/A converters.

TMBEN	Control of input clock of timer B is provided
0	Stop provide an input clock. · SFR used by timer B cannot be written. · Timer B is in the reset state.
1	Provides an input clock. · Read and write the SFR used by timer B.

PGACMPEN	Control of an input clock of an amplifier and a comparator is provided
0	Stop provide an input clock. · SFRs used by amplifiers and comparators cannot be written. · The amplifier comparator is in a reset state.
1	Provides an input clock. · SFRs that can read and write to the amplifiers and comparators.

TMMEN Note	Control of an input clock of a timer M is provided
0	Stop provide an input clock. · SFR used by timer M cannot be written. · The timer M is in a reset state.
1	Provides an input clock. · Read and write the SFR used by the timer M.

Note 1: When user option byte (000C2H) has FRQSEL4 bit '1', fCLK must be set to fIH before setting the bit4 (TMMEN) of Peripheral Enable Register 1 (PER1).

If you want to change fCLK to a clock other than fIH, you must change it after clearing the bit4(TMMEN) of Peripheral enable register 1 (PER).

Note 2: If you select fHOCO as the count source for TMC, you must set fCLK to fIH before setting bit1 (TMCEN) of Peripheral Enable Register 1 (PER1.

If you want to change fCLK to a clock other than fIH, you must change it after clearing bit1 (PER1) (TMCEN).

Figure4-8 Format of Peripheral Enable Register 1 (PER1) (2/2)

Address: 4002081AH After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN	TMMEN Note 1	DMAEN	PWMPEN	TMCEN Note 2	TMAEN

DMAEN	Providing control of an input clock of a DMA
0	Stop provide an input clock. · DMA cannot run.
1	Provides an input clock. · DMA can run.

PWMPEN	Control of input clock of PWM cut-off circuit
0	Stop provide an input clock. · Cannot write the SFR used by the PWM cutoff circuit. · The PWM cut-off circuit is in the reset state.
1	Provides an input clock. · Can read and write SFR used by PWM cut-off circuit.

TMCEN	Control of an input clock of a timer C is provided
0	Stop provide an input clock. · SFR used by timer C cannot be written. · Timer C is in a reset state.
1	Provides an input clock. · Read and write the SFR used by timer C.

TMAEN	Control of an input clock of timer A is provided
0	Stop provide an input clock. · SFR used by timer A cannot be written. · Timer A is in a reset state.
1	Provides an input clock. · Read and write the SFR used by timer A.

Note 1: When user option byte (000C2H) has FRQSEL4 bit '1', fCLK must be set to fIH before setting the bit4 (TMMEN) of Peripheral Enable Register 1 (PER1)·

If you want to change fCLK to a clock other than fIH, you must change it after clearing the bit4(TMMEN) of Peripheral enable register 1 (PER).

Note 2: If you select fHOCO as the count source for TMC, you must set fCLK to fIH before setting bit1 (TMCEN) of Peripheral Enable Register 1 (PER1.

If you want to change fCLK to a clock other than fIH, you must change it after clearing bit1 (PER1) (TMCEN).

4.3.7 Secondary system clock provides mode control register (OSMC)

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

If that RTCLPC position" 1", the clock is stop supplied to peripheral functions other than the real-time clock and the 15-bit interval timer in deep sleep mode or sleep mode of the CPU running with a secondary system clock.

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 operation instruction.

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

Figure4-9 The secondary system clock provides the format of the mode control register (OSMC)

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

symbol	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	UTMCK0	0	0	0	0

RTCLPC	Settings in Deep Sleep Mode and Sleep Mode in which the CPU is running at a sub-system clock
0	Allow sub-system clock to be provided for peripheral functions (Refer to Table 27-1~Table 27-3 for allowable peripherals.
1	Stop providing that sub-system clock to peripheral function other than the real time clock and 15 bit interval timer.

UTMCK0	Selection of a real-time clock, a 15-bit interval timer and an operation clock of timer A
0	· The secondary system clock is the real-time clock and the runtime clock of the 15-bit interval timer. · A low-speed internal oscillator cannot be selected as the count source for timer A.
1	· The low-speed internal oscillator clock is the real-time clock and the runtime clock of the 15-bit interval timer. · A low-speed internal oscillator or sub-system clock can be selected as the count source of timer A.

4.3.8 Frequency selection register (HOCODIV) for high speed internal oscillator.

This is a register that changes the high-speed internal 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 operation instruction.

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

Figure4-10 Format of Frequency Selective Register (HOCODIV) for High-speed Internal Oscillator

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 Clock Frequency of High Speed Internal Oscillator			
			FRQSEL4=0		FRQSEL4=1	
			FRQSEL3=0	FRQSEL3=1	FRQSEL3=0	FRQSEL3=1
0	0	0	f_{IH}=24MHz	f_{IH}=32MHz	f_{IH}=48$_{MHzfHOCO}$=48MHz	f_{IH}=32$_{MHzHOCO}$=64MHz
0	0	1	f_{IH}=12MHz	f_{IH}=16MHz	f_{IH}=24$_{MHzHOCO}$=48MHz	f_{IH}=16$_{MHzHOCO}$=64MHz
0	1	0	f_{IH}=6MHz	f_{IH}=8MHz	f_{IH}=12$_{MHzHOCO}$=48MHz	f_{IH}=8$_{MHzHOCO}$=64MHz
0	1	1	f_{IH}=3MHz	f_{IH}=4MHz	f_{IH}=3$_{MHzHOCO}$=48MHz	f_{IH}=4$_{MHzHOCO}$=64MHz
1	0	0	Disable setting.	f_{IH}=2MHz	Disable setting.	f_{IH}=2$_{MHzHOCO}$=64MHz
1	0	1	Disable setting.	f_{IH}=1MHz	Disable setting.	f_{IH}=1$_{MHzHOCO}$=64MHz
above			Disable setting.			

Note 1. The HOCODIV register must be set in a state where the high speed internal oscillator clock (f_{IH}) is selected as the CPU/peripheral hardware clock (fCLK).

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 High speed internal oscillator fine-tuned register (HIOTRM)

This is a register that corrects the accuracy of the high speed internal oscillator. Self-measurement of the frequency of the high speed internal oscillator and accuracy correction can be performed using a timer or the like with a high precision external clock input. The HIOTRM register is set by an 8-bit memory operation instruction.

Note: If the temperature and the voltage of the VDD pin change after the correction accuracy, the frequency changes. In the case where the temperature and the voltage of the VDD pin are varied, it is necessary to perform the correction before or periodically before the required frequency accuracy.

Figure4-11 The Scheme of Fine-Tuning Register (HIOTRM) for High Speed Internal Oscillator

Address: 40021C00H After reset: Note R/W

symbol	7	6	5	4	3	2	1	0
HIOTRM	0	0	HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0

HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0	high speed internal oscillator
0	0	0	0	0	0	minimum speed
0	0	0	0	0	1	▲
0	0	0	0	1	0	
0	0	0	0	1	1	
0	0	0	1	0	0	
· The · The · The						
1	1	1	1	1	0	▼
1	1	1	1	1	1	maximum speed

Note: The reset value is the adjustment value at shipment.

Note: 1. Every 1 bit of the HIOTRM register can correct the clock accuracy of the high-speed internal oscillator by about 0.05%.

4.4 System clock oscillating circuit

4.4.1 X1 oscillating 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, 1,1

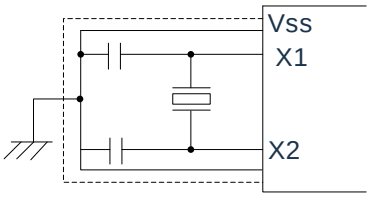
· external clock input :EXCLK, OSCSEL=1,1

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 2-4 when not used as an input port. Treatment of each unused pin."

Examples of the external circuit of the X1 oscillating circuit are as followsFigure4-12in the

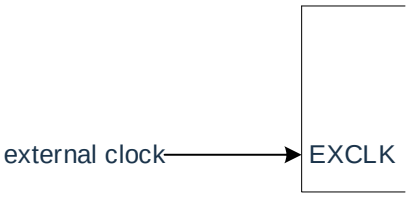
Figure4-12Example of an external circuit of an X1 oscillating circuit

(a) Crystal or Ceramic oscilator



Crystal oscilator or ceramic oscillator

(b) external clock



The notes are shown on the following page.

4.4.2 XT1 oscillating circuit

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

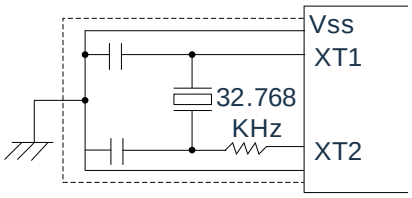
When using the XT1 oscillator circuit, the bit5 and bit4 (EXCLKS, OSCSELS) of the clock mode control register must be set:

- crystal oscillation :EXCLKS, OSCILS=0,1
- external clock input :EXCLKS, OSCILS=1,1

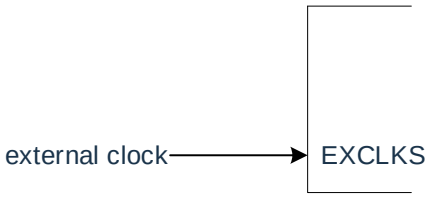
When the XT1 oscillator circuit is not used, it must be set to the input port mode (EXCLKS, OSCSELS=0, 0). Also, when not used as an input port, refer to Table 2-4 for Handling Without Pins." Examples of the external circuit of the XT1 oscillating circuit are as followsFigure4-13in the

Figure4-13 Example of External Circuit of XT1 Oscillation Circuit

(a) cyrstal oscilation



(b) external clock



Note: To avoid the effect of wiring capacitance or the like when using the X1 oscillator circuit and the XT1 oscillator circuitFigure4-12andFigure4-13The dotted section in routes:

· Cabling must be minimized.

· Cannot cross with other signal lines and cannot approach wiring through which a variable high current flows.

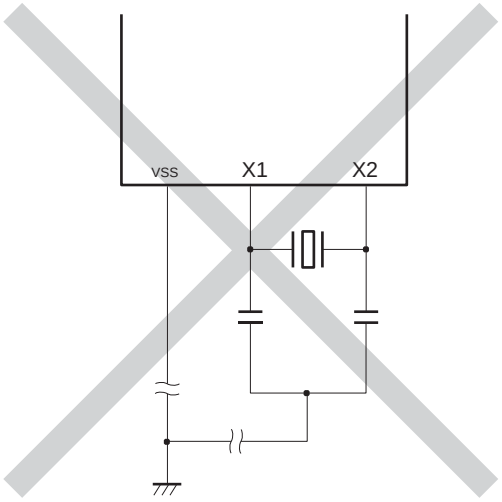
· The capacitor and VSS contacts of the oscillating circuit must always be kept at the same potential, and the grounding pattern through which a large current flows must not be grounded.

· The signal cannot be removed from the oscillator circuit.

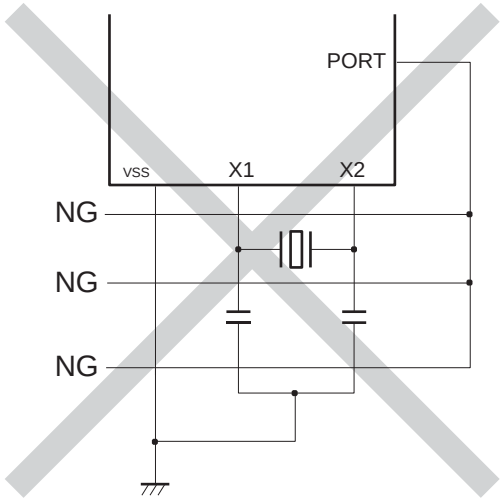
incorrect resonator connection examples such asFigure4-14in the

Figure4-14 Examples of incorrect resonator connections (1/2)

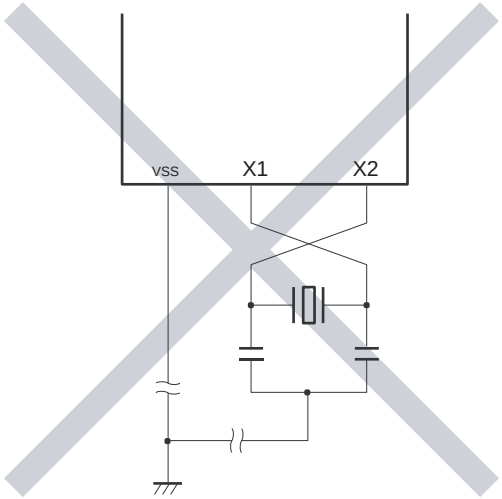
(a) The wiring of the connection circuit is too long



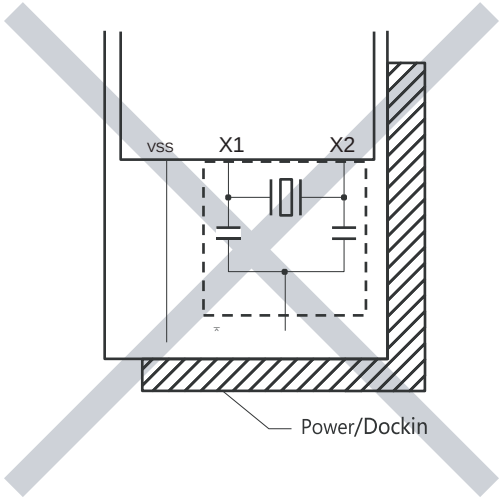
(b) Signal line crossing



(c) Cross-wiring of signal lines for X1 and X2 the wiring



(d) The X1 and X2 are powered or mapped under

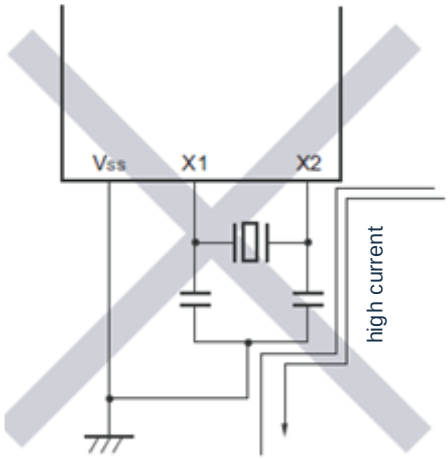


NOTE: In a multi-layered or dual-faceplate, you cannot configure power or map shapes below the X1 pin, X2 pin, and resonator cabling areas (dashed lines in the figure). The wiring may not produce a capacitive component and affect the oscillation characteristics.

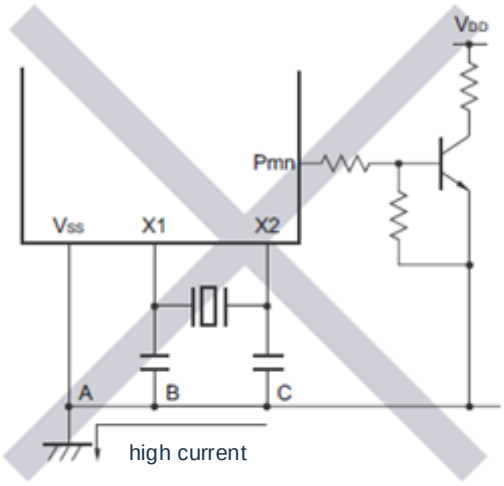
Note: Using the secondary system clock, replace X1 and X2 with XT1 and XT2 respectively.

Figure4-14 Examples of incorrect resonator connections (2/2)

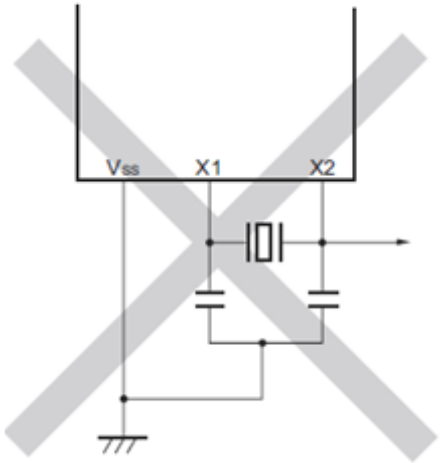
(e) varying high current source close to singal lines



(f) Current flows along grounding of oscilation circuit
(Point A, B, C has difference in electric potential)



(g) extracted signal



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

Note: Using the secondary system clock, replace X1 and X2 with XT1 and XT2 respectively.

4.4.3 high speed internal oscillator

The BAT32A237 has a built-in high speed internal oscillator. Frequency can be selected from 64MHz, 48MHz, 32MHz, 24MHz, 16MHz, 12MHz, 8MHz, 6MHz, 4MHz, 3MHz, 2MHz, 1MHz and via option bytes (000C2H). The CPU clock is 2-division clock when 64MHz is selected. The oscillation can be controlled by the bit0 (HIOSTOP) of the clock running state control register (CSC).

After the reset is removed, the high speed internal oscillator automatically starts to oscillate.

4.4.4 low-speed internal oscillator

The BAT32A237 has a built-in low-speed internal oscillator.

The low-speed internal oscillator clock is used as the watchdog timer, real-time clock, 15-bit interval timer clock and timer A clock, and an external reference clock of SysTick timer.

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

The low-speed internal 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 internal oscillator stops oscillating when the WDSTBYON bit is 0. When the watchdog timer runs, the low-speed internal oscillator clock does not stop running even if the program is out of control.

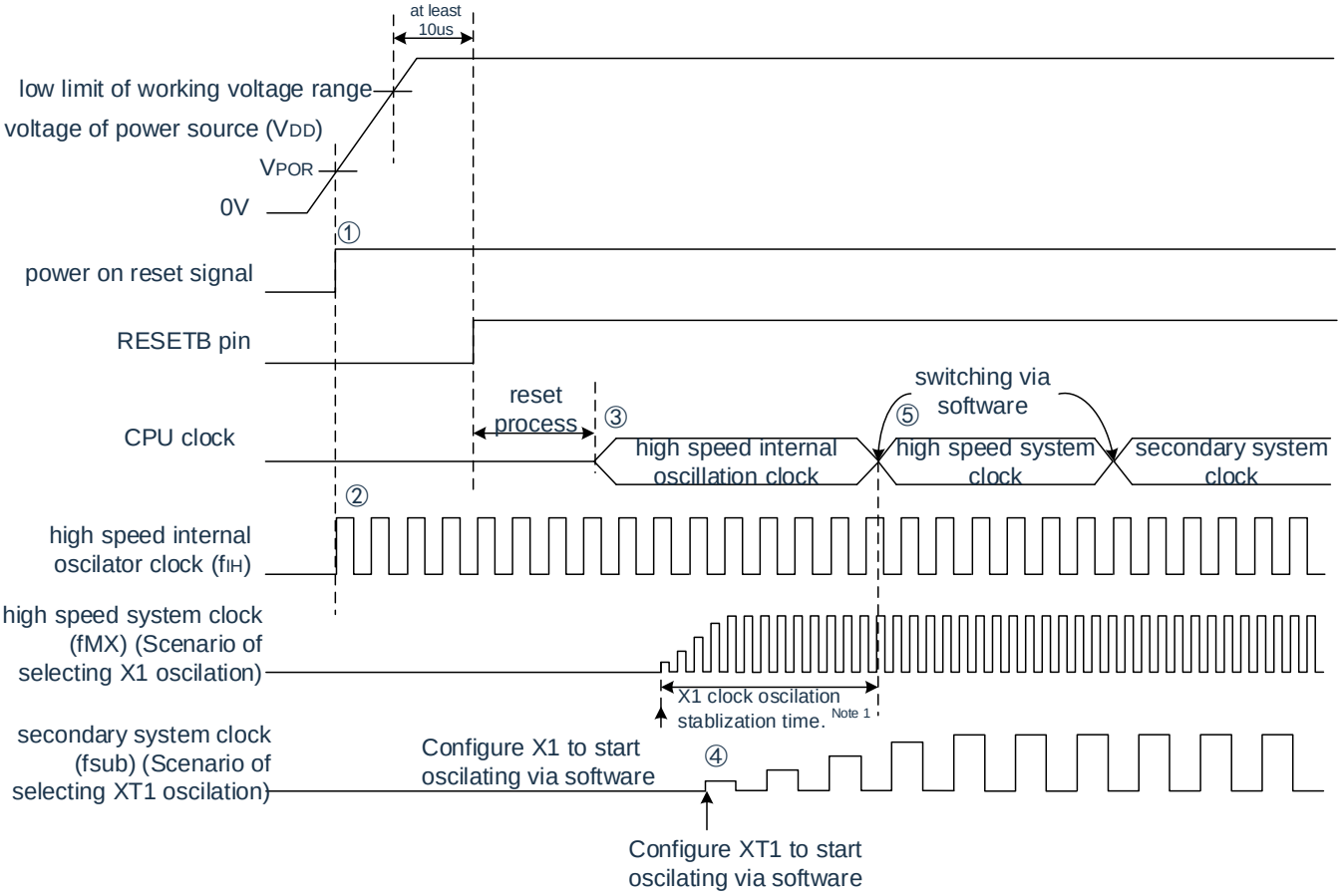
4.5 Operation of clock generation circuit

The clock generation circuit generates various clocks as shown below and controls the operation mode of the CPU such as the standby mode (referenceFigure4-1).

- main system clock fMAIN
  - high speed system clock fMX
    - X1 clock fX
    - external main system clock fEX
  - high speed internal oscillator clock fIH
- sub-system clock fSUB
  - XT1 clock fXT
  - external sub-system clock fEXS
- low-speed internal oscillator clock fIL
- CPU/peripheral hardware clock fCLK

The CPU starts to run through the output of the high speed internal oscillator after the BAT32A237 is reset-free. The operation of the clock generating circuit when the power is turned on is as shown in Figure4-15.

Figure4-15 Operation of clock generating circuit when power is turned on



① An internal reset signal is generated by the power-on reset (POR) circuit after the power is turned on. However, before reaching the operation voltage range shown by the AC characteristic of the data manual, the reset state is maintained by a voltage detection circuit or an external reset (the above figure is an example when an external reset is used).

② If that reset is release, the high speed internal oscillator will automatically start to oscillate.

③ After the reset is removed, a voltage stable waiting and reset process are performed, and then the CPU starts running with a high speed internal oscillator clock.

④ You must set the X1 or XT1 clock start oscillation by software (reference "4.6.2 An example of X1 oscillation circuit"and"4.6.3 Setting example of XT1 oscillation circuit").

⑤ If you want to switch the CPU clock to an X1 or XT1 clock, you must switch by software settings after waiting for the clock to oscillate.4.6.2 An example of X1 oscillation circuit"and"4.6.3 Setting example of XT1 oscillation circuit").

NOTE:1. When the reset is removed, the X1 clock's oscillation stability time must be confirmed through OSTC of the oscillation stability time counter.

Note: If you use an external clock input by the EXCLK pin, you do not need an oscillatory steady-state wait time.

4.6 clock control

4.6.1 Example of high speed internal oscillator set-up

The CPU/peripheral hardware clock (f_{CLK}) must run at high internal oscillator clock. High speed internal 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 a frequency selection register (HOCODIV) of a high speed internal oscillator.

[Settings for option bytes]
Address: 000C2H

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

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of high speed internal oscillator	
					fHOCO	fIH
1	1	0	0	0	64MHz	32MHz
1	0	0	0	0	48MHz	48MHz
0	1	0	0	0	32MHz	32MHz
0	0	0	0	0	24MHz	24MHz
0	1	0	0	1	32MHz	16MHz
0	0	0	0	1	24MHz	12MHz
0	1	0	1	0	32MHz	8MHz
0	0	0	1	0	24MHz	6MHz
0	1	0	1	1	32MHz	4MHz
0	0	0	1	1	24MHz	3MHz
0	1	1	0	0	32MHz	2MHz
0	1	1	0	1	32MHz	1MHz
above					Disable setting.	

[Setting of HOCODIV for high speed internal 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 of High Speed Internal Oscillator			
			FRQSEL4=0		FRQSEL4=1	
			FRQSEL3=0	FRQSEL3=1	FRQSEL3=0	FRQSEL3=1
0	0	0	f_{IH}=24MHz	f_{IH}=32MHz	f_{IH}=48$_{MHzfHOCO}$ =48MHz	f_{IH}=32 $_{MHzHOCO}$=64 MHz
0	0	1	f_{IH}=12MHz	f_{IH}=16MHz	f_{IH}=24$_{MHzHOCO}$= 48MHz	f_{IH}=16$_{MHzHOCO}$ =64MHz
0	1	0	f_{IH}=6MHz	f_{IH}=8MHz	f_{IH}=12$_{MHzHOCO}$ =48MHz	f_{IH}=8 $_{MHzHOCO}$=64 MHz
0	1	1	f_{IH}=3MHz	f_{IH}=4MHz	f_{IH}=6$_{MHzfHOCO}$= 48MHz	f_{IH}=4$_{MHzfHOCO}$ =64MHz
1	0	0	Disable setting.	f_{IH}=2MHz	f_{IH}=3$_{MHzfHOCO}$= 48MH	f_{IH}=2 $_{MHzHOCO}$=64 MHz
1	0	1	Disable setting.	f_{IH}=1MHz	Disable setting.	f_{IH}=1 $_{MHzfHOCO}$=64 MHz
above			Disable setting.			

4.6.2 An example of X1 oscillation circuit

The CPU/peripheral hardware clock (f_{CLK}) must run at high internal oscillator clock. Thereafter, if the X1 oscillating clock is changed, setting of the oscillating circuit and controlling the oscillation start are performed by the OSTS, the clock operation mode control register (CMC) and the clock operation state control register (CSC), and the oscillation stabilization is waited by the state register (OSTC) of the oscillation stabilization time counter. The $_X$1 oscillation clock is set to $_{fCLK}$ through the system clock control register (CKC) after waiting for oscillation stabilization·

[Register Settings] Registers must be set in the order (1) to (5).

① The OSCSEL position of the CMC register is "1", and the AMPH position is "1" when fX is 10 MHz.

	7	6	5	4	3	2	1	0
CMC	EXCLK0	OSCSEL1	EXCLKS0	OSCILS0	0	AMPHS10	AMPHS00	AMPH0/1

② The oscillation stability time of the X1 oscillation circuit when the deep sleep mode is released is selected by the OSTS register.

Example) To wait at least 102 μs through a 10 MHz resonator, you must set the following values.

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS20	OSTS11	OSTS00

③ The MSTOP bit of the CSC register is cleared "0" so that the X1 oscillation circuit starts oscillation.

	7	6	5	4	3	2	1	0
CSC	MSTOP0	XTSTOP1	0	0	0	0	0	HIOSTOP0

④ The OSTC register is used to wait for the oscillation stabilization of the X1 oscillation circuit.

Example) To wait at least 102 μs through a 10 MHz resonator, you must wait until you become the following values.

	7	6	5	4	3	2	1	0
OSTC	MOST81	MOST91	MOST101	MOST110	MOST130	MOST150	MOST170	MOST180

⑤ The X1 oscillation clock is set to the CPU/peripheral hardware clock via the MCM0 bit of the CKC register.

	7	6	5	4	3	2	1	0
CKC	CLS0	CSS0	MCS0	MCM01	0	0	0	0

4.6.3 Setting example of XT1 oscillation circuit

The CPU/peripheral hardware clock (f_{CLK}) must run at high internal oscillator clock. After that, the setting of the oscillation circuit and the control of the oscillation start are performed by the sub-system clock supply mode control register OSMC, CMC and CSC.

Register Settings must be set in the order (1) to (5).

① In a deep sleep mode or a sleep mode in which the CPU runs at a sub-system clock, the RTCLPC position"1".

	7	6	5	4	3	2	1	0
OSMC	RTCLPC0/1	0	0	WUTMCK00	0	0	0	0

② The OSCSELS position of the CMC register is "1" so that the XT1 oscillation circuit operates.

	7	6	5	4	3	2	1	0
CMC	EXCLK0	OSCSEL0	EXCLKS0	OSCILS1	0	AMPHS10/1	AMPHS00/1	AMPH0

AMPHS0 and AMPHS1 bits: An oscillation mode of the XT1 oscillation circuit is set.

③ The XTSTOP bit of the CSC register is cleared "0" so that the XT1 oscillation circuit starts to oscillate.

	7	6	5	4	3	2	1	0
CSC	MSTOP1	XTSTOP0	0	0	0	0	0	HIOSTOP0

④ It is necessary to wait for that oscillation steady time required by the secondary system clock through software, timer function, etc.

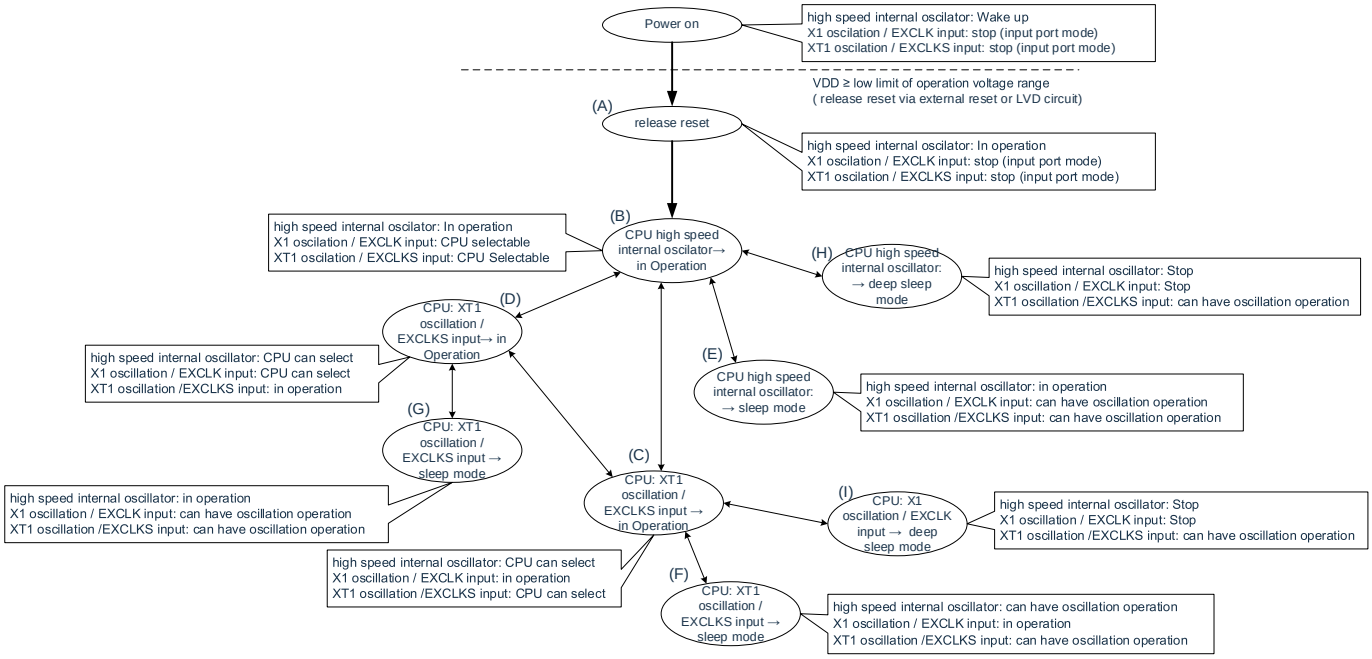
⑤ The XT1 oscillation clock is set to the CPU/peripheral hardware clock by the CSS bit of the CKC register.

	7	6	5	4	3	2	1	0
CKC	CLS0	CSS0	MCS0	MCM01	0	0	0	0

4.6.4 CPU Clock State Transition Diagram

The CPU clock state transition diagram of this product is as shown in Figure4-16.

Figure4-16 CPU Clock State Transition Diagram



Examples of CPU clock transfer and SFR register setting areTable4-3in the

Table4-3Examples of CPU clock transfer and SFR register set-up (1/5)

(1). After the reset (A) is released, the CPU is transferred to the high speed internal oscillator clock operation (B).

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

(2). After the reset (A) is released, the CPU moves to the high speed system clock operation (C).

(The CPU operates with a high speed internal oscillator clock immediately after the reset is released (B)

(Order in which SFR registers are set) →

Settings flag for SFR register / state transition	CMC Register Note 1			OSTS register	CSC register	OSTC register	CKC register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(A) → (B) → (C) (X1 clock: 1MHz≤f_X≤10MHz)	0	1	0	Note 2	0	Confirmation Required	1
(A) → (B) → (C) (X1 clock: 10MHz<f_X≤20MHz)	0	1	1	Note 2	0	Confirmation Required	1
(A)→(B)→ (C) (external main clock)	1	1	×	Note 2	0	No confirmation required	1

NOTE:1. After the reset is removed, the clock run mode control register (CMC) can only be written 1 time via the 8-bit memory operation instruction.

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

· The expected oscillation stability time of the state register (OSTC) of the oscillation stability time counter≤the oscillation stability time set by the OSTS register

Attention The clock must be set after the supply voltage has reached the set clock operational voltage (reference to the data manual).

(3). After the reset (A) is released, the CPU transfers to the sub-system clock operation (D).

(The CPU operates with a high speed internal oscillator clock immediately after the reset is released (B)

(Order in which SFR registers are set) →

Settings flag for SFR register / state transition	CMC register note				CSC register	oscillatory waiting	CKC register
	EXCLKS	OSCELS	AMPHS1	AMPHS0	XTSTOP		CSS
(A) → (B) → (D) (XT1 clock)	0	1	0/1	0/1	0	Required	1
(A) → (B) → (D) (external sub-clock)	1	1	×	×	0	Required	1

Note: After the reset is removed, the clock run mode control register (CMC) can only be written once through the 8-bit memory operation instruction.

Remarks: 1.×: Ignore

2. Table 4-3 (A)-I corresponding to (A)-(I) FIG.

Table4-3 Examples of CPU clock transfer and SFR register set-up (2/5)

(4). The CPU is transferred from a high speed internal oscillator clock operation (B) to a high speed system clock operation (C).

(Order in which SFR registers are set) →

Settings flag for SFR register / state transition	CMC Register Note 1			OSTS register	CSC register	OSTC register	CKC register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(B) → (C) (X1 clock: 1MHz≤f_X≤10MHz)	0	1	0	Note 2	0	Confirmation Required	1
(B) → (C) (X1 clock: 10MHz<f_X≤20MHz)	0	1	1	Note 2	0	Confirmation Required	1
(B) → (C) (external main clock)	1	1	×	Note 2	0	No confir matio	1

Not required if set. operation. Not required in high speed system clock

Note:1. Only the Clock Run Mode Control Register (CMC) can be set 1 time after the reset is removed. Not required if set.

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

· The expected oscillation stability time of the state register (OSTC) of the oscillation stability time counter≤the oscillation stability time set by the OSTS register

Note: The clock must be set after the supply voltage has reached the set clock operational voltage (reference to the data manual).

(5). The CPU is transferred from a high speed internal oscillator clock operation (B) to a sub-system clock operation (D).

(Order in which SFR registers are set) →

Settings flag for SFR register / state transition	CMC register note			CSC register	oscillatory waiting	CKC register
	EXCLKS	OSCELS	AMPHS1,0	XTSTOP		CSS
(B)(D) (XT1 clock)	0	1	00: low power oscillation 01: normal oscillation	0	Required	1
(B)(D) (external sub-clock)	1	1	×	0	Required	1

Not required if set. clock operation. Not required in sub-system

Note: After the reset is removed, the clock run mode control register (CMC) can only be written once through the 8-bit memory operation instruction. Not required if set.

Remarks: 1.×: Ignore

2. Table 4-3 (A)-I corresponding to (A)-(I) FIG.

Table4-3 Examples of CPU clock transfer and SFR register set-up (3/5)

(6). The CPU is transferred from a high-speed system clock operation (C) to a high speed internal oscillator clock operation (B).

(Order in which SFR registers are set) →

Settings flag for SFR register / state transition	CSC register	stable oscillation precision waiting	CKC register
	HIOSTOP		MCM0
(C) (B)	0	Note	0

It is not require in high speed internal oscillator clock operation.

Note: At FRQSEL4=0: 45μs~ 65μs

at FRQSEL4=1: 45μs ~135μs

Note: The oscillation accuracy of the high speed internal oscillator clock is stably waiting to change due to temperature conditions and deep sleep mode.

Table4-3 Examples of CPU clock transfer and SFR register set-up (4/4)

(7). The CPU is transfer from a high speed system clock run (C) to a sub system clock run (D).

(Order in which SFR registers are set) →

Settings flag for SFR register / state transition	CSC register	stable oscillation precision waiting	CKC register
	XTSTOP		CSS
(C) → (D)	0	Required	1

Not required in sub-system clock operation.

(8). The CPU is transferred from the sub-system clock operation (D) to the high speed internal oscillator clock operation (B).

(Order in which SFR registers are set) →

Settings flag for SFR register / state transition	CSC register	stable oscillation precision waiting	CKC register
	HIOSTOP		CSS
(D)(B)	0	Note	0

It is not require in high speed internal oscillator clock operation.

Note: At FRQSEL4=0: 45μs~ 65μs

at FRQSEL4=1: 45μs ~135μs

Note: 1. Table 4-3 (A)~(I) corresponds to (A)~(I).

2. The oscillation accuracy of the high speed internal oscillator clock is stably waiting to change due to temperature conditions and deep sleep mode.

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

(9) The CPU is moved from the sub-system clock run (D) to the high speed system clock run (C).

(Order in which SFR registers are set) ⟶

Settings flag for SFR register / state transition	OSTS register	CSC register	OSTC register	CKC register
		MSTOP		CSS
(D) → (C) (X1 clock: 1MHz≤$_{fX}$≤10MHz)	Note	0	Confirmation Required	0
(D) → (C) (X1 clock: 10MHz<$_{fX}$≤20MHz)	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 stability time of the oscillation stability time selection register (OSTS) must be set as follows:

· The expected oscillation stability time of the state register (OSTC) of the oscillation stability time counter≤the oscillation stability time set by the OSTS register

Note: The clock must be set after the supply voltage has reached the set clock operational voltage (reference to the data manual).

(10)· The CPU is transferred to a sleep mode (E) in high-speed internal oscillator clock operation.

· The CPU is transferred to a sleep mode (F) in a high speed system clock operation (C).

· The CPU is transferred to a sleep mode (G) in a sub-system clock operation (D).

state transition	Setting Content
(B) → (E) (C) → (F) (D) → (G)	Execute WFI instructions.

Note: Table 4-3 (A) to (I) corresponds to (A) to (I).

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

(11)· The CPU is transferred to a deep sleep mode (H) in high speed internal oscillator clock operation.

· The CPU is transferred to a deep sleep mode (I) in a high speed system clock operation.

(Set-up Order) ⟶

state transition		Setting Content		
(B)(H)		Stop Peripheral functions that cannot be run in deep sleep mode.	—	The SCR register bit2 (SLEEPDEEP) is set to 1 and executes the WFI instruction.
(C)(I)	X1 oscillation		Sets the OSTS register.	
	external clock		—	

Note: Table 4-3 (A) to (I) corresponds to (A) to (I) of FIG.

4.6.5 Conditions before CPU clock transfer and post-transfer processing

The conditions before and after the CPU clock transfer are as follows.

Table4-4 Transfer of CPU clock (1/3)

CPU Clock		Conditions before transfer	Post-transfer processing
Before Transfer	After Transfer		
high speed internal oscillator clock	X1 clock	The X1 oscillation is stable. • OSCSEL=1, EXCLK=0, MSTOP=0 · After an oscillatory stabilization time	If the oscillation of the high speed internal oscillator is stopped (HIOSTOP=1), the operation current can be reduced.
	external main system clock	Set the external clock entered by the EXCLK pin to be valid. • OSCSEL=1, EXCLK=1, MSTOP=0	
	XT1 clock	The XT1 oscillation is stable. • OSCILS=1, EXCLKS=0, XTSTOP=0 · After an oscillatory stabilization time	
	external sub-system clock	Set the external clock entered by the EXCLKS pin to be valid. • OSCILS=1, EXCLKS=1, XTSTOP=0	
X1 clock	high speed internal oscillator clock	High speed internal oscillator oscillation is allow. • HIOSTOP=0 · After an oscillatory stabilization time	The X1 oscillation (MSTOP=1).
	external main system clock	Can't move.	—
	XT1 clock	The XT1 oscillation is stable. • OSCILS=1, EXCLKS=0, XTSTOP=0 · After an oscillatory stabilization time	It can stop the X1 oscillation (MSTOP=1).
	external sub-system clock	Set the external clock entered by the EXCLKS pin to be valid. • OSCILS=1, EXCLKS=1, XTSTOP=0	It can stop the X1 oscillation (MSTOP=1).
external main system clock	high speed internal oscillator clock	High speed internal oscillator oscillation is allow. • HIOSTOP=0 · After an oscillatory stabilization time	Can invalidate external master clock inputs (MSTOP=1).
	X1 clock	Can't move.	—
	XT1 clock	The XT1 oscillation is stable. • OSCILS=1, EXCLKS=0, XTSTOP=0 · After an oscillatory stabilization time	Can invalidate external master clock inputs (MSTOP=1).
	external sub-system clock	Set the external clock entered by the EXCLKS pin to be valid. • OSCILS=1, EXCLKS=1, XTSTOP=0	Can invalidate external master clock inputs (MSTOP=1).

Table4-4 Transfer of CPU clock (2/2)

CPU Clock		Conditions before transfer	Post-transfer processing
Before Transfer	After Transfer		
XT1 clock	high speed internal oscillator clock	A high speed internal oscillator is oscillating and a high speed internal is selected The oscillator clock acts as the main system clock. • HIOSTOP=0, MCS=0	The oscillation of XT1 can be stopped (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 an oscillatory stabilization time • MCS=1	
	external main system clock	The external clock input by the EXCLK pin is set to be valid and the high speed system clock is selected as the main system clock. • OSCSEL=1, EXCLK=1, MSTOP=0 • MCS=1	
	external sub-system clock	Can't move.	—
external sub-system clock	high speed internal oscillator clock	A high speed internal oscillator is oscillating and a high speed internal is selected The oscillator clock acts as the main system clock. • HIOSTOP=0, MCS=0	Can set external secondary system clock input invalid (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 an oscillatory stabilization time • MCS=1	
	external main system clock	The external clock input by the EXCLK pin is set to be valid and the high speed system clock is selected as the main system clock. • OSCSEL=1, EXCLK=1, MSTOP=0 • MCS=1	
	XT1 clock	Can't move.	—

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 internal 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 (seeTable4-5~Table4-7).

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 internal oscillator clock.

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

Table4-5 Maximum time required to switch master system clock

Clock A	Switch direction	Clock B	Remark
fIH	⟷	fMX	Refer to Table 4-6.
fMAIN	⟷	fSUB	Refer to Table 4-7.

Table4-6 Maximum number of clocks required for fIH↔fMX

Set value before switching		Set value after switch	
MCM0		MCM0	
		0 ($f_{MAIN}=f_{IH}$)	1 ($f_{MAIN}=f_{MX}$)
0 ($f_{MAIN}=f_{IH}$)	$f_{MX}\geq f_{IH}$		2 Clock
	$f_{MX}<f_{IH}$		2 f_{IH}/fMX clocks
1 ($f_{MAIN}=f_{MX}$)	$f_{MX}\geq f_{IH}$	2 f_{MX}/fIH clocks	
	$f_{MX}<f_{IH}$	2 Clock	

Table4-7 Maximum number of clocks required for fMAIN↔fSUB

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

Remarks: 1.Table4-6andTable4-7The number of clocks in is the number of CPU clocks before the switch.

2.Table4-6andTable4-7The number of clocks in is the number of clocks rounded to the decimal portion.

Example of a master system clock switching from a high speed system clock to a high speed internal oscillator clock (f_{IH}=8MHz, fMX=10 MHz)

$2_{fMX}/_{fIH}$=2 (10/8)=2.5➞3 clocks

4.6.7 Conditions before clock oscillation stops

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

Table4-8 Condition and flag setting before clock oscillation stops

clock	Condition before clock stops (invalid external clock input)	Flag setting for SFR register
high speed internal oscillator clock	MCS=1 or CLS=1 (CPU operates at a clock other than the high speed internal oscillator clock)	HIOSTOP=1
X1 clock external main system clock	MCS=0 or CLS=1 (CPU runs at a clock other than the high speed system clock)	MSTOP=1
XT1 clock external sub-system clock	CLS=0 (CPU runs at a clock other than the secondary system clock)	XTSTOP=1

Chapter 5 hardware divider

The hardware divider is dedicated hardware that supports high-performance computing. The hardware divider is a 32-bit signed integer divider that outputs a 32-bit signed quotient and remainder result.

5.1 characteristics

- 32-bit signed (2 complement) integer division calculation
- 32 bit signed divisor, 32 bit signed divisor
- 32-bit signed quotient and 32-bit signed remainder output
- Write Division Register Automatic Trigger Division Calculation
- Except 0 warning flag
- Indicates the BUSY flag in the operation
- Interrupt request with calculation end
- 4 or 8 CPU clock cycles per calculation
  - Spend 4 CPU clock cycles at double speed
  - Spend 8 CPU clock cycles in non-speed state

5.2 Feature Description

When using hardware dividers, you need to set the Division Register (DIVIDEND) and then the Division Register (DIVISOR), because writing to the division register automatically trigger division calculations. You can know when the calculation is done by querying the BUSY bit of the STATUS or by using the interrupt at the end of the calculation. The results can be read out through the QUOTIENT and the remainder (REMAINDER) registers.

Note: Do not write divisor or divisor registers, nor read quotient or remainder registers during calculation, otherwise the results are unpredictable.

5.3 Register for hardware divider

The register for the hardware divider is as follows:

Register Base Address: DIV_BASE = 4008_0000H;

register name	register description	R/W	Reset Value	register address
DIVIDEND	division register	R/W	0000_000H	DIV_BASE+00H
DIVISOR	divisor register	R/W	0000_000H	DIV_BASE+04H
QUOTIENT	quotient register	R	0000_000H	DIV_BASE+08H
REMAINDER	remainder register	R	0000_000H	DIV_BASE+0CH
STATUS	state register	R	0000_000H	DIV_BASE+10H

R: read only, W: write only, R/W: booth read and write

5.3.1 division register (DIVIDEND)

The divisor register is a register that holds the divisor and its value participates in the division operation as a 32-bit signed integer.

31	30	29	28	27	26	25	24
DIVIDEN [31:24]							
23	22	21	20	19	18	17	16
DIVIDEN [23:16]							
15	14	13	12	11	10	9	8
DIVIDEN [15:8]							
7	6	5	4	3	2	1	0
DIVIDEN [7:0]							

5.3.2 Divisor register (DIVISOR)

The divisor register is a register for storing divisors, whose value is a 32-bit signed integer participating in the division operation. A write to this register automatically triggers a division calculation.

31	30	29	28	27	26	25	24
DIVISOR [31:24]							
23	22	21	20	19	18	17	16
DIVISOR [23:16]							
15	14	13	12	11	10	9	8
DIVISOR [15:8]							
7	6	5	4	3	2	1	0
DIVISOR [7:0]							

5.3.3 quotient register (QUOTIENT)

The register stores the quotient of the division calculation result after the division calculation is completed, and the value is taken as a 32-bit signed integer.

31	30	29	28	27	26	25	24
QUOTIENT [31:24]							
23	22	21	20	19	18	17	16
QUOTIENT [23:16]							
15	14	13	12	11	10	9	8
QUOTIENT [15:8]							
7	6	5	4	3	2	1	0
QUOTIENT [7:0]							

5.3.4 Remainder register (REMAINDER)

The register stores the remainder of the division result after division calculation, and the value is taken as a 32-bit signed integer.

31	30	29	28	27	26	25	24
REMAINDER [31:24]							
23	22	21	20	19	18	17	16
REMAINDER [23:16]							
15	14	13	12	11	10	9	8
REMAINDER [15:8]							
7	6	5	4	3	2	1	0
REMAINDER [7:0]							

5.3.5 status register (STATUS)

The status of the hardware divider can be queried through the status register, including the zero-division flag and the BUSY flag.

31	30	29	28	27	26	25	24
Reserve							
23	22	21	20	19	18	17	16
Reserve							
15	14	13	12	11	10	9	8
Reserve						DIVBYZERO	BUSY
7	6	5	4	3	2	1	0
Reserve							

DIVBYZERO	Used to indicate the case of a division, updated each time the division register is written.
0	The divisor is not 0.
1	Divisor is 0

BUSY	Used to indicate the status of the division operation.
0	Division operation complete
1	Divisor in progress

Chapter 6 Universal Timer Unit Timer4

The product is carried with a universal timer unit and contains four channels. The number of channels of the universal timer unit varies depending on the product. See table below:

Table 6-1 The products have timer channels

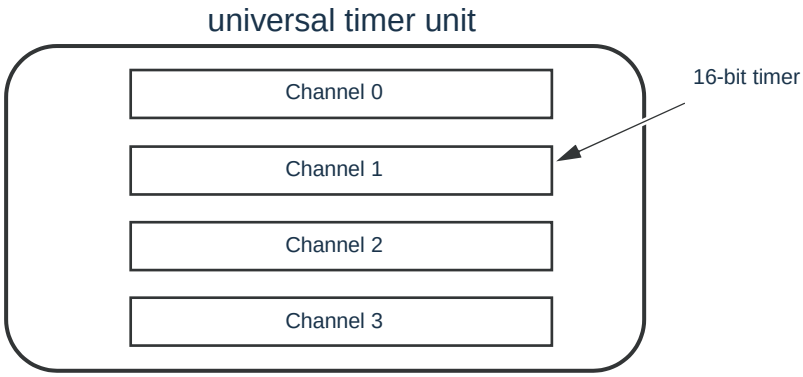
unit	channel	24 pins	32,36,40,48,52,64 pins
Unit 0	Channel 0	○	○
	Channel 1	-	○
	Channel 2	-	○
	Channel 3	○	○

Description:

1. In this chapter, the label "m" is the unit number. When carrying multiple Timer4 units, it is distinguished. This product only carries a general timer Timer4, and fixes m=0.
2. In this chapter, the label "n" below represents the channel number (n=0~3), and whether the channel 0~3 has/output pins different by product. Refer to "Table 6-2 Products with Timer Input / Output Pins"for details.
3. The following sections in this chapter are intended for the 64 pin products.

The universal timer unit has four 16-bit timers.

Each 16-bit timer is called a "channel" and can be used as a separate timer or a combination of multiple channels for advanced timer functions.



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

Stand-alone channel operation	multi-channel linkage function
· Interval timer (→Ref. 6.8.1) · Square wave output (→Ref. 6.8.1) · External event counter (➞Ref. 6.8.2) · Frequency divider note (→reference 6.8.3) · Measurement of the input pulse interval (→Ref. · Measurement of the high and low level width of the · Delay counter (➞Ref. 6.8.6)	· Single trigger pulse output (➞ Ref. 6.9.1) · PWM output (→Ref. 6.9.2) · Multiple PWM Output (→Ref. 6.9.3) The

Note: Only channel 0 of universal timer unit 0 is.

The 16-bit timers of channel 1 and channel 3 of cell 0 can be used as two 8-bit timers (high and low). The functions of channel 1 and channel 3 can be used as 8-bit timers are:

- Interval timers (high 8-bit and low 8-bit timers)/square wave output (low 8-bit timers only)
- External event counters (low 8-bit timers only)
- Delay Counter (Low 8-bit timers only)

The LIN-bus communication can be realized through the coordination of the channel 3 of the unit 0 and the UART0 of the universal serial communication unit.

6.1 Function of universal timer unit

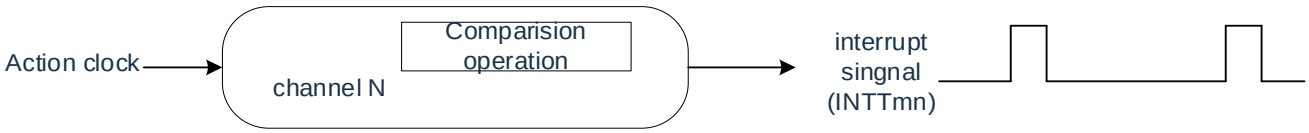
The universal timer unit has the following functions:

6.1.1 Stand-alone channel operation

Independent channel operation function is independent of the other channel operation mode to use any channel function.

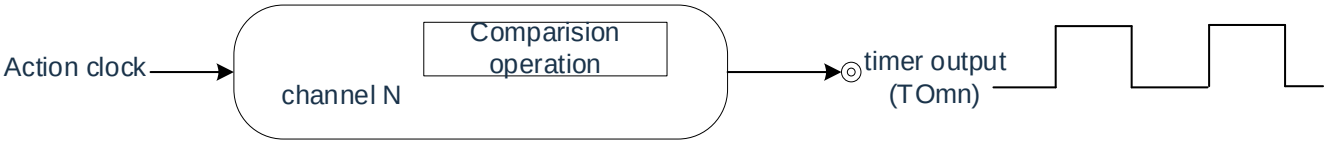
(1) interval timer

Can be used as a reference timer to generate an interrupt (INTTMmn) at a fixed interval.



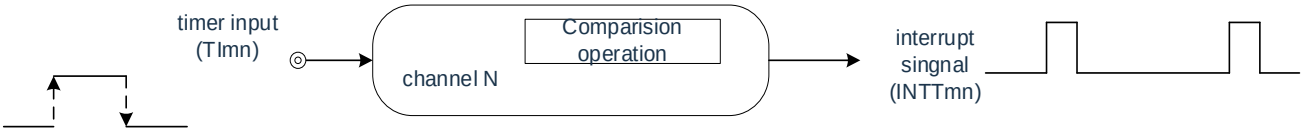
(2) square wave output

When a INTTMmn interrupt is generated, an alternating operation is performed and a 50% duty cycle square wave is output from an output pin (TOmn) of the timer.



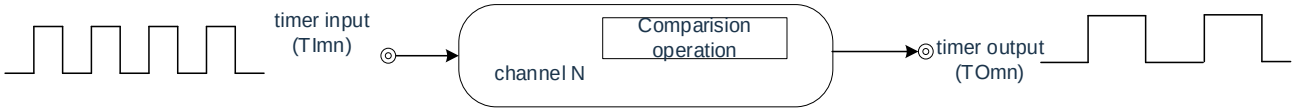
(3) External event counters

An effective edge of the input signal of the timer input pin (TImn) is counted, and if a prescribed number of times is reached, an event counter generating an interrupt can be used.



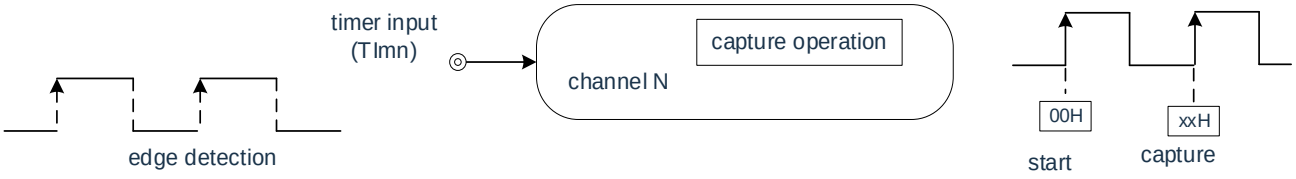
(4) Divider Function (Cell 0-only channel 0)

An input clock of a timer input pin (TI00) is frequency-divided and then output from an output pin (TO00).



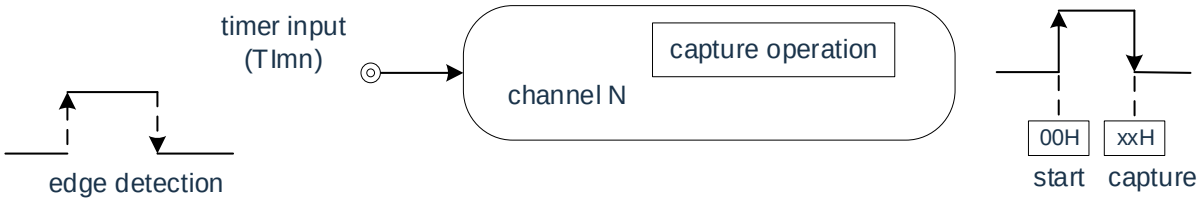
(5) Measurement of input pulse interval

An effective edge of the input pulse signal of the input pin (TImn) starts counting at a timer and captures the count value at the effective edge of the next pulse, thereby measuring the interval of the input pulse.



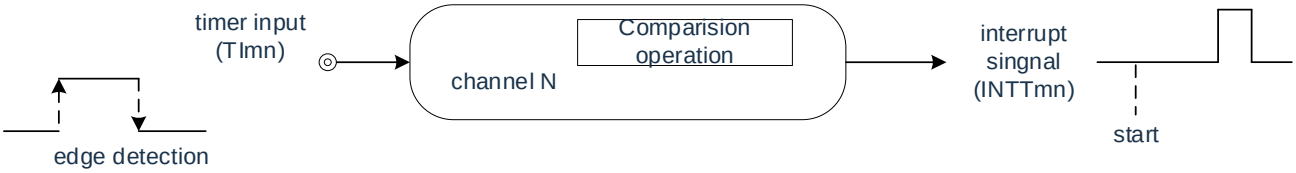
(6) Measurement of High and Low Level Width of Input Signal

The input signal of the TImn is counted at one edge of the input pin at the timer and the count value is captured at the other edge, thereby measuring the high and low level width of the input signal.



(7) delay counter

An effective edge of the input signal of the timer input pin (TImn) starts counting and an interrupt is generated after an arbitrary delay period.



Remarks: 1.m: Cell number (m=0)n: Channel number (n=0~3)

2. Whether the timer input/output pins of channels 0 to 3 are different or not depends on the product. Refer to "Table 6-2, Product Has Timer Input/Output Pins" for details.

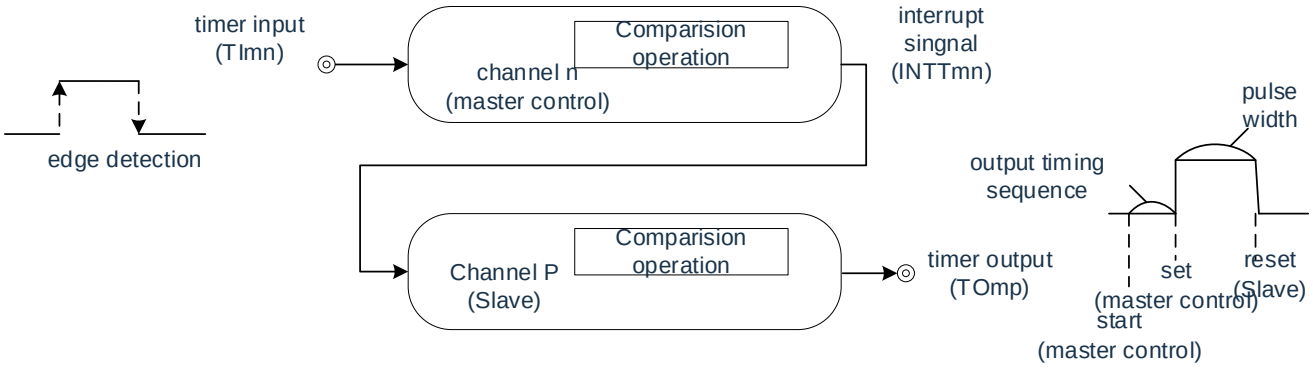
6.1.2 multi-channel coordinated operation function

The multi-channel coordinated operation function is a function which combines the main control channel (the reference timer of the main control period) and the subordinate channel (the timer which follows the main control channel).

The multi-channel coordinated operation function can be used in the following modes.

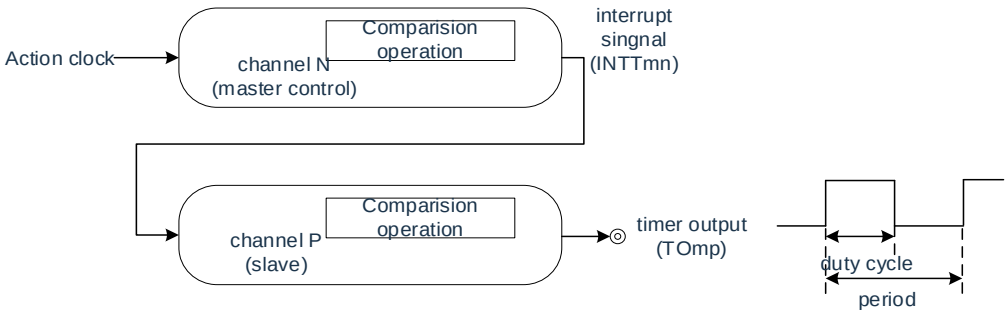
(1) single trigger pulse output

The two channels are used in pairs, and a single trigger pulse with arbitrary output timing and pulse width is generated.



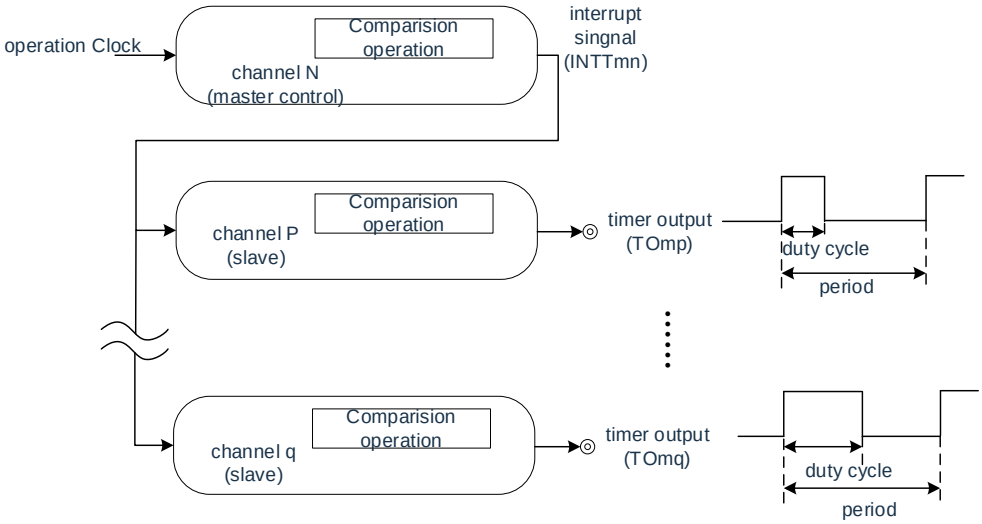
(2) PWM (Pulse Width Modulation) output

The two channels are used in pairs to generate pulses with arbitrary period and duty cycle.



(3) Multiple PWM (Pulse Width Modulation) output

The PWM signal can be generated at most 3 arbitrary duty cycles by extending the PWM function and using 1 master channel and multiple slave channels.



Note: For details on the Multi-Channel Coordinated Operation Functional Rules, refer to "6.4.1 Basic Rules for Multi-Channel Coordinated Operation Functions.

Note: m: Cell number (m=0,1)n: Channel number (n=0~3)p,q: Slave channel number (n<p<q≤ 3)

6.1.3 8-bit timer operation function (only for channel 1 and channel 3 of unit 0).

The 8-bit timer run function is the function of using the 16-bit timer channel as the 2 8-bit timer channels. Only channel 1 and channel 3 of unit 0 can be used.

Note: There are several rules when you use an 8-bit timer to run functions.

Refer to the "Basic Rules for 6.4.2 8-bit Timer Operation Functions (Channel 1 and Channel 3 only)" for details.

6.1.4 LIN-bus support (Cell 0-only channel 3)

A universal timer unit is used to check whether a received signal in the LIN-bus communication is suitable for the LIN-bus communication table.

(1) Detection of wake-up signal

A low level width is measured by counting at the beginning of the falling edge of the input signal of the UART0 serial data input pin (RxD0) and capturing the counting at the rising edge. If that low level width is great than or equal to a fixed value, it is consider a wake-up signal.

(2) Detection of interval segments

After detecting the wake-up signal, the low level width is measured by counting the falling edge of the input signal of the UART0 serial data input pin (RxD0). If that low level width is great than or equal to a fixed value, it is consider as an interval segment.

(3) Measurement of the Pulse Width of Synchronous Section

After detecting the interval segment, the low level width and the high level width of the input signal of the UART0 serial data input pin (RxD0) are measured. The baud rate is calculated based on the bit intervals of the synchronization segments measured in this manner.

Note: Refer to "6.3.13 Input Switch Control Register (ISC)" and "6.8.5 Operation as Input Signal Level Width Measurement".

6.2 Structure of universal timer unit

The universal timer unit consists of the following hardware.

Table 6-1 Structure of universal timer unit

Project	structure
counter	timer count register mn (TCRmn).
register	timer data register mn (TDRmn)
Timer input	TI00~TI03 note 1, RxD0 pin (for LIN-bus)
output of timer	TO00~TO03 injection 1, output control circuit
control register	<Register for Cell Set-up> · Peripheral Enable Register 0 (PER0) · Timer clock select register m (TPSm). · The timer channel allows the status register m (TEm) · Timer Channel Start Register m (TSm) TSm. · Timer channel stop register m (TTm) TTm. · Timer input selection register 0 (TIS0) · Timer output allows register m (TOEm) to register · Timer output register m (TOm) · Timer output level register m (TOLm) · Timer output mode register m(TOMm)
	<register per channel> · Timer mode register mn(TMRmn) · Timer state register mn(TSRmn) · Input Switch Control Register (ISC) · Noise filter allows registers 1,2 (NFEN1, NFEN2) · Port mode control register (PMCxx) Note 2 · Port mode register (PMxx) Note 2 · Port register (Pxx) Note 2

Note: 1. Whether the timer input / output pin of channel 0~3 is different by product. Refer to "Table 6-2, Product Has Timer Input/Output Pins" for details.

2. The set port mode control registers (PMCxx), port mode registers (PMxx), and port registers (Pxx) differ by product. Please refer to the "2.5 Register Settings When Using Multiplexing ".

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

Whether the timer input/output pins of each channel of the universal timer unit are different depends on the product.

Table 6-2 The products have timer input/output pins.

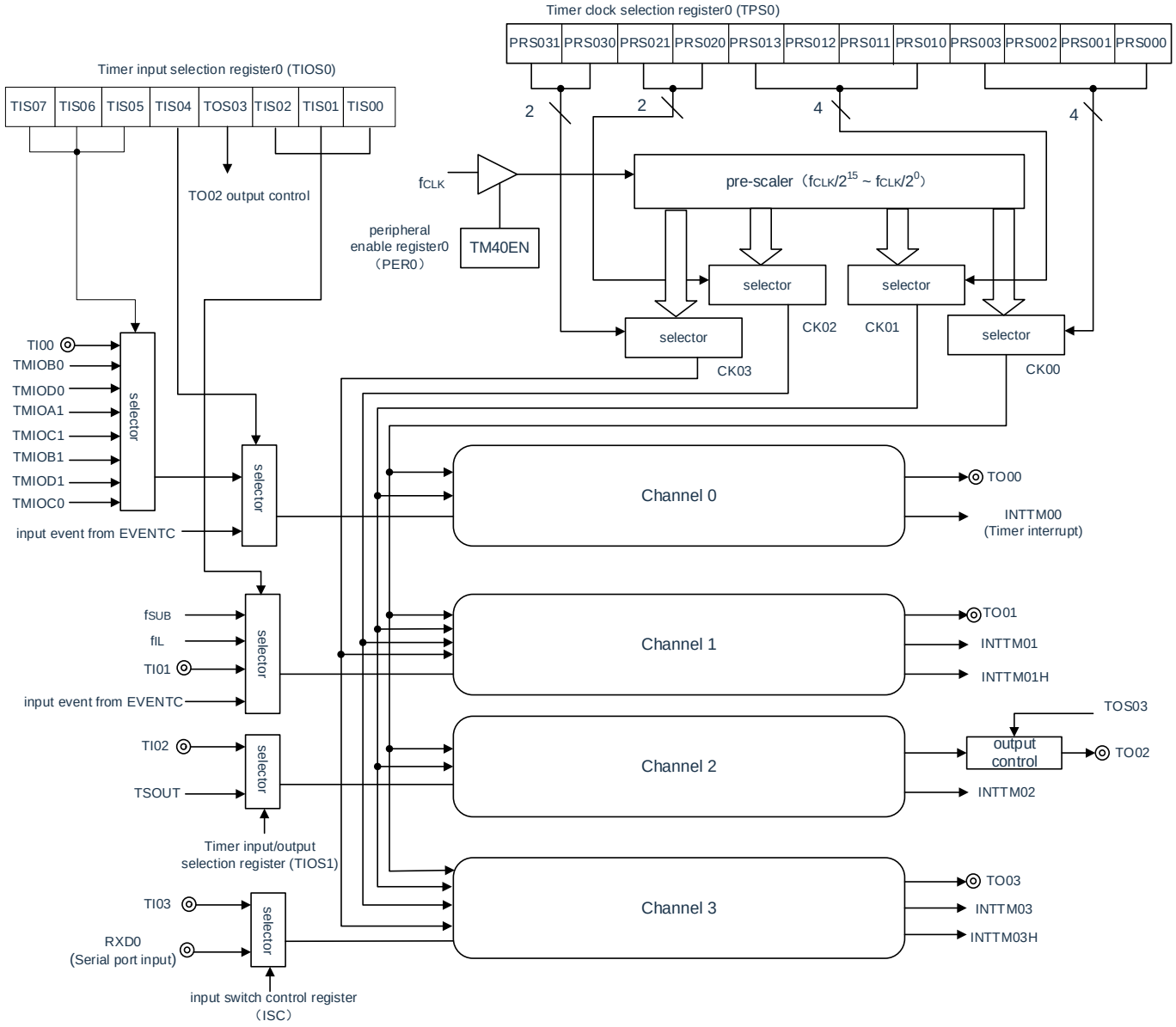
Channel of timer array unit		Whether the Input/Output Pins of Each Product Have
		32,36,40,48,52,64 pins
Unit 0	Channel 0	TI00/TO00
	Channel 1	TI01/TO01
	Channel 2	TI02/TO02
	Channel 3	TI03/TO03

Note:1. When the input of the timer and the output of the timer are multiplexed by the same pin, they can only be used as the input of the timer or the output of the timer.

2. -: No built-in channels.

The block diagram of the universal timer unit is shown in Figure 6-1.

Figure6-1 Overall block diagram of universal timer unit 0



Note: f_{SUB} : sub-system clock frequency

f_{IL} : low-speed internal oscillator clock frequency

6.2.1 Universal timer unit register list

Register Base Address: 0x40041C00

offset address	register name	Read and Write	bit width	Reset Value
0x180	TCR00	R	16	FFFFFH
0x182	TCR01	R	16	FFFFFH
0x184	TCR02	R	16	FFFFFH
0x186	TCR03	R	16	FFFFFH
0x190	TMR.00	R/W	16	0000H
0x192	TMR.01	R/W	16	0000H
0x194	TMR.02	R/W	16	0000H
0x196	TMR.03	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
0x318	TDR.00	R/W	16	0000H
0x31A	TDR.01	R/W	16	0000H
0x31A	TDR.01L	R/W	8	00H
0x31B	TDR.01H	R/W	8	00H
0x364	TDR.02	R/W	16	0000H
0x366	TDR.03	R/W	16	0000H
0x366	TDR.03L	R/W	8	00H
0x367	TDR.03H	R/W	8	00H

6.2.2 Timer count register mn (TCRmn).

The TCRmn register is a 16-bit read-only register that counts the count clock. Count is increased or decreased in synchronization with the rising edge of the count clock.

The operation mode is selected by the MDmn3~MDmn0 bit of the timer mode register mn (TMRmn) to switch the increment and decrement count (reference 6.3.3 timer mode register mn (TMRmn)).

Figure6-2 Table for timer count register mn (TCRmn)

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

Note: m: Cell number (m=0)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 TM4 mEN bit of the Peripheral Enable Register 0 (PER0)
- The count of the slave channels in the PWM output mode ends
- The count of the dependent channels ends in a delayed count mode
- The count of the main/slave channels in the single-trigger pulse output mode ends
- Count end of slave channels in multiple PWM output mode

The count value becomes "0000H" in the following cases.

- When you enter the start trigger in capture mode
- At the end of the capture in capture mode

Note: 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 value of the TCRmn register varies depending on the mode and state of operation.

Table 6-3: Read value of timer count register mn (TCRmn) in each running mode

operation mode	Count Method	Read value note for timer counter register mn (TCRmn)			
		The value when the Run Mode is changed after the reset is removed	Count Paused (TTmn=1)	Count Paused (TTmn=1) Changes the value when the run mode is changed	The value after a single count when waiting for the start of the trigger
interval timer mode	decremental count	FFFFFH	Value at Stop	indefinite value	—
capture mode	incremental count	0000H	Value at Stop	indefinite value	—
Event Counter Mode	decremental count	FFFFFH	Value at Stop	indefinite value	—
single count mode	decremental count	FFFFFH	Value at Stop	indefinite value	FFFFFH
Capture & Single Count Mode	incremental count	0000H	Value at Stop	indefinite value	Snap value for TDRmn register +1

Note: Represents the read value of the TCRmn register when channel n is in the timer idle state (TEmn=0) and the count allow state (TSmn=1). Keep this value in the TCRmn register until the count starts.

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

6.2.3 timer data register mn (TDRmn)

This is a 16-bit register that can be used for switching between capture and comparison functions. The operation mode is selected by the MDmn3~MDmn0 bit of the timer mode register mn(TMRmn) to switch the capture function and comparison function.

Can rewrite TDRmn register at any time.

This register can be read and written in units of 16 bits.

In 8-bit timer mode (SPLIT bits of timer mode registers m1, m3 (TMRm1, TMRm3), TDRm1 registers and TDRm3 registers can be read and written in units of 8 bits, where TDRm1H and TDRm3H are used as high 8 bits and TDRm1L and TDRm3L are used as low 8 bits.

After the reset signal is generated, the value of the TDRmn register changes to "0000H".

Figure6-3 Table of timer data register mn(TDRmn) (n=0,2,4,5,6,7)

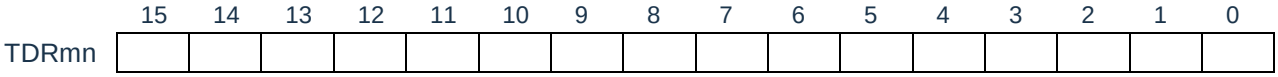
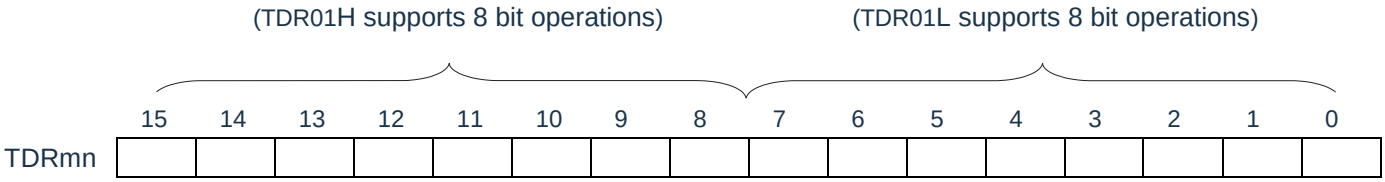


Figure6-4 Table of timer data register mn(TDRmn) (n=1,3)



(i) The case where the timer data register mn (TDRmn) is used as a comparison register

The count is decremented from the set value of the TDRmn register, and an interrupt signal (INTTMmn) is generated when the count value becomes '0000H'. Holds the value of the TDRmn register until it is overwritten.

Note: Even if you enter a capture trigger, the TDRmn register set to the comparison function does not capture the run.

(ii) the use of the timer data register mn(TDRmn) as a capture register

The counter value of timer count register mn (TCRmn) is captured to the TDRmn register by input capture trigger.

You can select the valid edge of the TImn pin as the capture trigger. A selection of capture triggers is set by a timer mode register mn (TMRmn).

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

6.3 Register for controlling universal timer unit

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

- Peripheral Enable Register 0 (PER0).
- Timer clock selection register m (TPSm)
- timer mode register mn (TMRmn)
- timer status register mn (TSRmn)
- Timer channel allows state register m (TEm).
- Timer Channel Start Register m (TSm)
- Timer channel stop register m (TTm).
- Timer Input-Output Selection Register (TIOS0, TIOS1)
- timer output allow register m (TOEm)
- Timer output register m (TOm)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Input switch control register (ISC)
- Noise Filter Allow Register 1 (NFEN1)
- Port mode control register (PMCxx)
- Port Mode Register (PMxx)
- Port Register (Pxx)

Note: The assigned registers and bits differ depending on the product. You must set an initial value for unassigned bits.

Note: m: Cell Number (m= 0)n: Channel Number (n=0~3)

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.

You must set bit0 (TM40EN) to "1" when you want to use universal timer unit 0. The PER0 register is set by the 8-bit memory operation instruction. After the reset signal is generated, the value of the PER0 register changes to "00H".

Figure 6-5 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	IRDEN	ADCEN	ICAHN	SCI1EN	SCI0EN	CAN0	TM40EN

TM4EN	Control of an input clock of a universal timer unit 0
0	Stop provide an input clock. · Cannot write the SFR used by the universal timer unit. · The universal timer unit 0 is in a reset state.
1	Provides an input clock. · The SFR used by the universal timer unit 0 can be read and written

Note 1. To set a universal timer unit, you must first set the following register in the state with the TM4mEN bit "1". When the TM4mEN bit is "0", the control register value of the timer array unit is the initial value, neglecting the write operation (except for timer input-output selection register 0 (TIOS0), input switching control register (ISC), noise filter permit register 1 (NFEN1), port mode control register PMCx, port mode registerPMx and port register Px).

· Timer state register mn(TSRmn)

· The timer channel allows the state register m (TEm) to

· Timer channel start register m (TSm).

· Timer channel stop register m (TTm).

· Timer output allows register m (TOEm).

· Timer output register m (TOm).

· Timer output level register m (TOLm).

· Timer output mode register m (TOMm).

6.3.2 Timer Clock Select Register m (TPSm).

The TPSm register is a 16-bit register that selects two or four common runtime clocks (CKm0, CKm1, CKm2, CKm3). CKm0 is selected by bit3~0 of the TPSm register and CKm1 is selected by bit7~4 of the TPSm register. In addition, only channel 1 and channel 3 can select CKm2 and CKm3, select CKm2 by bit9~8 of TPSm register, and select CKm3 by bit13 and bit12 of TPSm register.

The TPSm register in the timer run can only be overridden in the following cases.

The case of PRSm00~PRSm03 bit can be rewritten (n=0~3):

Select CKm0 as the channel for the runtime clock (CKSmn1, CKSmn0=0, 0) all in the stopped state (TEmn=0).

The case of PRSm10~PRSm13 bit can be rewritten (n=0~3):

Select CKm2 as the channel for the runtime clock (CKSmn1, CKSmn0=0, 1) all in the stopped state (TEmn=0).

Can override PRSm20 and PRSm21 bits (n=1, 3):

Select CKm1 as the channel for the runtime clock (CKSmn1, CKSmn0=1, 0) all in the stopped state (TEmn=0).

Can override PRSm30 and PRSm31 bits (n=1, 3):

Select CKm3 as the channel for the runtime clock (CKSmn1, CKSmn0=1, 1) all in the stopped state (TEmn=0).

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

Figure 6-6 Table (1/2) of timer clock selection register m (TPSm)

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

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection note for runtime clock ($CKmk$) (k=0,1)					
					f_{CLK}=2MHz	f_{CLK}=4MHz	f_{CLK}=8MHz	f_{CLK}=20MHz	f_{CLK}=32MHz
0	0	0	0	f_{CLK}	2MHz	4MHz	8MHz	20MHz	32MHz
0	0	0	1	$f_{CLK}/2$	1MHz	2MHz	4MHz	10MHz	16MHz
0	0	1	0	$f_{CLK}/2^2$	500kHz	1MHz	2MHz	5MHz	8MHz
0	0	1	1	$f_{CLK}/2^3$	250kHz	500kHz	1MHz	2.5MHz	4MHz
0	1	0	0	$f_{CLK}/2^4$	125kHz	250kHz	500kHz	1.25MHz	2MHz
0	1	0	1	$f_{CLK}/2^5$	62.5kHz	125kHz	250kHz	625kHz	1MHz
0	1	1	0	$f_{CLK}/2^6$	31.3kHz	62.5kHz	125kHz	313kHz	500kHz
0	1	1	1	$f_{CLK}/2^7$	15.6kHz	31.3kHz	62.5kHz	156kHz	250kHz
1	0	0	0	$f_{CLK}/2^8$	7.81kHz	15.6kHz	31.3kHz	78.1kHz	125kHz
1	0	0	1	$f_{CLK}/2^9$	3.91kHz	7.81kHz	15.6kHz	39.1kHz	62.5kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95kHz	3.91kHz	7.81kHz	19.5kHz	31.25kHz
1	0	1	1	$f_{CLK}/2^{11}$	977Hz	1.95kHz	3.91kHz	9.77kHz	15.6kHz
1	1	0	0	$f_{CLK}/2^{12}$	488Hz	977Hz	1.95kHz	4.88kHz	7.81kHz
1	1	0	1	$f_{CLK}/2^{13}$	244Hz	488Hz	977Hz	2.44kHz	3.91kHz
1	1	1	0	$f_{CLK}/2^{14}$	122Hz	244Hz	488Hz	1.22kHz	1.95kHz
1	1	1	1	$f_{CLK}/2^{15}$	61.0Hz	122Hz	244Hz	610Hz	977Hz

Note: In case of changing the clock selected as fCLK (changing the value of the System Clock Control Register (CKC)), the universal timer unit (TTm=000FH) must be stopped. It is necessary to stop that $_{universal}$ timer unit even when selecting an effective edge of the run-time clock (fMCK) or TImn pin input signal.

Note: 1. bit15, 14,11,10 must be placed "0.

2. If you select fCLK as the runtime clock (CKmk) and set TDRnm to 0000H (n=0~3), you cannot use the universal timer unit.

Note: 1.$_{fCLK}$: Clock frequency for CPU/peripheral hardware

2. The clock waveform selected by the TPSm register is high (m=1~15) with only 1 fCLK period. Refer to the "6.5.1 Count Clock ($_{fTCLK}$)".

Figure 6-7 Table (2/2) of timer clock selection register m (TPSm)

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 Note for Runtime Clock (CKm2)					
			f_{CLK}=2MHz	f_{CLK}=4MHz	f_{CLK}=8MHz	f_{CLK}=20MHz	f_{CLK}=32MHz
0	0	$f_{CLK}/2$	1MHz	2MHz	4MHz	10MHz	16MHz
0	1	$f_{CLK}/2^2$	500kHz	1MHz	2MHz	5MHz	8MHz
1	0	$f_{CLK}/2^4$	125kHz	250kHz	500kHz	1.25MHz	2MHz
1	1	$f_{CLK}/2^6$	31.3kHz	62.5kHz	125kHz	313kHz	500kHz

PRSm31	PRSm30	Selection Note for Runtime Clock (CKm3)					
			f_{CLK}=2MHz	f_{CLK}=4MHz	f_{CLK}=8MHz	f_{CLK}=20MHz	f_{CLK}=32MHz
0	0	$f_{CLK}/2^8$	7.81kHz	15.6kHz	31.3kHz	78.1kHz	125kHz
0	1	$f_{CLK}/2^{10}$	1.95kHz	3.91kHz	7.81kHz	19.5kHz	31.3kHz
1	0	$f_{CLK}/2^{12}$	488Hz	977Hz	1.95kHz	4.88kHz	7.81kHz
1	1	$f_{CLK}/2^{14}$	122Hz	244Hz	488Hz	1.22kHz	1.95kHz

Note: In case of changing the clock selected as fCLK (changing the value of the System Clock Control Register (CKC)), the universal timer unit (TTm=000FH) must be stopped. It is necessary to stop that universal timer unit even when selecting an effective edge of the run-time clock (fMCK) or TImn pin input signal.

Note: bit15, 14,11,10 must be set to 0.

The interval time shown in Table 6-4 can be achieved by the interval timer function if channels 1 and 3 are used in 8-bit timer mode.

Table 6-4 The interval that the runtime clocks CKSm2 and CKSm3 can set

clock		Interval Time Note (f_{CLK}=32MHz)			
		10μs	100μs	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: ○ Contains errors within 5%.

Note: 1. f_{CLK}: Clock frequency for CPU/peripheral hardware

2. Refer to the "6.5.1 Count Clock (fTCLK)" for details on the $f_{CLK}/2$ r waveform selected by TPSm registers.

6.3.3 timer mode register mn (TMRmn)

The TMRmn register is a register for setting channel n running mode, selecting fMCK, counting clock, controlling/dependent,16bit/8 bit timer (only for channel 1 and channel 3), triggering and capturing, selecting the effective edge of timer input and running mode (interval, capture, event counter, single count, capture & single count).

Prevents the TMRmn register from being overwritten in the run (TEmn=1). However, bit7 and bit6 (CISmn1, CISmn0) can be rewritten in part of the functional operation (TEmn=1) (see "Independent Channel Operation Function of 6.8 Universal Timer Unit" and "Multi-channel Operation Function of 6.9 Timer Array Unit" for details).

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

Note: The bit11 for the TMRmn register varies by channel.

TMRm2:MASTERmn bit (n=2)

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

TMRm0: Fixed as "0".

Figure 6-8 Table (1/4) of timer mode register mn (TMRmn)

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=1,3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=0)	CKSmn1	CKSmn0	0	CCSmn	0 Note 1	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CKSmn1	CKSmn0	Channel n Runtime Clock (f_{MCK}) Selection
0	0	Runtime clock CKm0 set by timer clock selection register m (TPSm)
0	1	Runtime clock CKm2 set by timer clock selection register m (TPSm)
1	0	Runtime clock CKm1 set by timer clock selection register m (TPSm)
1	1	Runtime clock CKm3 set by timer clock selection register m (TPSm)
The runtime clock (f_{MCK}) is used for edge detection circuits. The sampling clock and the counting clock (f_{TCLK}) are generated by setting the CCSmn $_{bit}$. Only channel 1 and channel 3 can select the runtime clocks CKm2 and CKm3.		

CCSmn	Selection of channel n count clock (f_{TCLK})
0	Runtime clocks specified by CKSmn0 and CKSmn1 bits (f_{MCK})
1	Effective Edge of TImn Pin Input Signal · The case of Unit 0: Channel 0: Valid edge of input signal selected by TIS0 Channel 1: Valid Edges of the Input Signal Selected by TIS0
A count clock (f_{TCLK}) is used for the counter, the output control circuit, and the interrupt control circuit.	

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

Note: 1. bit13, 5, 4 must be set to 0.

2. To change the clock selected as fCLK (change the value of the System Clock Control Register (CKC), the timer array unit (TTm=00FFH) must be stopped even if the runtime clock (fMCK) specified by the CKSmn0 bit and the CKSmn1 bit is selected or the valid edge of the TImn pin input signal is $_{fTCLK}$.

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

Figure 6-9 Table (2/4) of timer mode register mn (TMRmn)

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=1,3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=0)	CKSmn1	CKSmn0	0	CCSmn	0 Note 1	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

(bit11 for TMRmn (n=2)

MASTERmn	Selection of independent channel operation/multi-channel simultaneous operation (slave or master) for channel n
0	A slave channel used as an independent channel operation function or a multi-channel coordinated operation function.
1	The main control channel is used as the multi-channel coordinated operation function.
Only channel 2 can be set as the master channel (MASTERmn=1). Channel 0 is fixed as '0' (because channel 0 is the highest bit channel, it is not related to this bit setting and is used as the master channel). For a channel that is used as a standalone channel operation function, the MASTERmn position "0".	

(TMRmn(n=1,3) bit11)

SPLITmn	Operation selection of 8-bit/16-bit timer for channel 1 and 3
0	Used as a 16-bit timer. (Slave channels used as independent channel operation functions or multi-channel coordinated operation functions)
1	Used as an 8-bit timer.

STSmn2	STSmn1	STSmn0	Settings for the start trigger and capture trigger of channel n
0	0	0	Only software triggers are active (no other trigger source is selected).
0	0	1	Use the valid edges entered by the TImn pin for the start trigger and capture trigger.
0	1	0	Use the two-sided edges of the TImn pin entries for the start trigger and capture trigger, respectively.
1	0	0	Use the interrupt signal of the main control channel (the case of a slave channel of the multi-channel coordinated operation function).
Other than above			Disable setting.

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

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

Figure 6-10: Table (3/4) of timer mode register mn (TMRmn)

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=1,3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=0)	CKSmn1	CKSmn0	0	CCSmn	0 Note 1	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CISmn1	CISmn0	Effective Edge Selection for TImn Pins
0	0	descending edge
0	1	rising edge
1	0	Double Edge (when measuring low level width) Start trigger: Down Edge, Capture Trigger: rising edge
1	1	Double Edge (when measuring high level width) Start trigger: Rising edge, capture trigger: descending edge
When STSmn2~STSmn0 bit is not '010B' and double edge is specified, CISmn1~CISmn0 position '10B'.		

Note: 1. bit11 is read-only, fixed to "0", ignoring write operations.

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

Figure 6-11Table (4/4) of timer mode register mn (TMRmn)

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=1,3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=0)	CKSmn1	CKSmn0	0	CCSmn	0 Note 1	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

MD mn3	MD mn2	MD mn1	Settings for Channel n Running Mode	corresponding function	Count run of TCR
0	0	0	interval timer mode	Interval Timer/Square Wave Output/ Divider Function/PWM Output (Master)	decremental count
0	1	0	capture mode	Measurement of input pulse interval	incremental count
0	1	1	Event Counter Mode	External event counters	decremental count
1	0	0	single count mode	Delay Counter/Single Trigger Pulse Output/PWM Output (Subordinates)	decremental count
1	1	0	Capture & Single Count Mode	Measurement of High and Low Level Width of Input Signal	incremental count
Other than above			Disable setting.		
The operation of the modes varies with the MDmn0 bit (see the following table).					

Run mode (settings for MDmn3~MDmn1 bits (see above table))	MD mn0	Settings to start counting and interrupts
· Interval timer mode (0, 0,0) · capture mode (0,1,0)	0	The timer interrupt does not occur at the start of the count (the output of the timer does not change).
	1	A timer interrupt is generated at the start of the count (the output of the timer also changes).
· Event counter mode (0,1,1)	0	The timer interrupt does not occur at the start of the count (the output of the timer does not change).
· Single Count Mode Note 2 (1,0,0)	0	Invalid start trigger in count run. No interruption occurs at this time.
	1	The start of the count run valid triggers note 3. No interruption occurs at this time.
· Capture& Single Count Mode (1,1,0)	0	The timer interrupt does not occur at the start of the count (the output of the timer does not change). Invalid start trigger in count run. No interruption occurs at this time.

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

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

3. If a start trigger (TSmn=1) is generated in the run, the counter is initialized and counts are restarted (no interrupt requests are generated). Note: m: Cell number (m=0)n: Channel Number (n=0~3)

6.3.4 timer status register mn (TSRmn)

The TSRmn register is a register that represents 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 6-5 for variations and placement/removal conditions of OVF bits in each mode of operation.

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

The lower 8 bits of the TSRmn register can be read with the TSRmnL and through an 8-bit memory operation instruction. After the reset signal is generated, the value of the TSRmn register changes to "0000H".

Figure 6-12: Table for 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 overspill occurred.
1	Overspill occurred.
If the OVF bit is "1", clear this flag (OVF=0) when the next count does not overrun and the count value is captured.	

Note: m: Cell number (m=0,1)n: Channel number (n=0~3 when m=0, n=0~7 when m=1)

Table 6-5 OVF bit variation and placement/removal conditions in each mode of operation

timer operation mode	OVF bit	Placement/Purge Criteria
· Capture Mode · Capture & Single Count Mode	Clear	No overflow condition occurred during capture
	position	Overspill on capture
· Interval timer mode · Event counter mode · Single count mode	Clear	— (not available)
	position	

Note: Even if the counter overflows, the OVF bit does not change immediately, and changes occur in subsequent captures.

6.3.5 Timer channel allows state register m (TEm) to be

The TEm register is a register that represents the permitted or stopped state of operation of each channel timer.

Each of the TEm registers corresponds to the bits of the timer channel start register m (TSm) and the timer channel stop register m (TTm). If each position of the TSm register is "1", the corresponding position of the TEm register is "1". If each position of the TTm register is "1", its corresponding bit is "0".

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

A lower 8-bit of a TEm register can be read with a TEmL and through an 8-bit memory operation instruction. After the reset signal is generated, the value of the TEm register changes to "0000H".

Figure 6-13: Table of timer channel allowed 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	TEHm3	0	TEHm1	0	0	0	0	0	TEm3	TEm2	TEm1	TEm0

m=0

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEm	0	0	0	0	0	0	0	0	TEm7	TEm6	TEm5	TEm4	TEm3	TEm2	TEm1	TEm0

m=1

TEHm3	A representation of the operational permit or stop state of a high 8-bit timer when channel 3 is in 8-bit timer mode
0	Idle Status
1	Run Allowed Status

TEHm1	A representation of the operational permit or stop state of a high 8-bit timer when channel 1 is in 8-bit timer mode
0	Idle Status
1	Run Allowed Status

TEmn	A representation of the running permit or stop state of channel n
0	Idle Status
1	Run Allowed Status
When channels 1 and 3 are in 8-bit timer mode, TEm1 and TEm3 indicate the allowed or stopped states of low 8-bit timers.	

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

6.3.6 Timer channel start register m (TSm).

The TSm register initializes the timer count register mn (TCRmn) and sets the trigger register when each channel count operation starts. If each position is "1", the counter bit of the timer channel allows state register m (TEm) to be "1. Because TSmn bits, TSHm1 bits, and TSHm3 bits are trigger bits, clear TSmn bits, TSHm1 bits, and TSHm3 bits immediately if it becomes run-enabled (TEmn, TEHm1, TEHm3=1).

The TSm register is set by a 16-bit memory operation instruction.

The low 8 bits of the TSm register can be set with TSmL and through 8-bit memory. After the reset signal is generated, the value of the TSm register becomes "0000H".

Figure 6-14 Table for 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	Channel 3 is triggered when the operation of the high 8-bit timer in the 8-bit timer mode allows (start)
0	No trigger.
1	Enter the TEHm3 position "1" into the count allowed state. If the count of the TCRm3 register is started in the count allowed state, the interval timer mode is entered (see tables 6-6 of 6).

TSHm1	Channel 1 is triggered when the operation of the high 8-bit timer in the 8-bit timer mode allows (start)
0	No trigger.
1	Enter the TEHm1 position "1" into the count allowed state. If the count of the TCRm1 register is started in the count allowed state, the interval timer mode is entered (see tables 6-6 of 6).

TSmn	Channel n Run Permit (Start) Trigger
0	No trigger.
1	Enter the TEmn position "1" into the count allowed state. The count start of the TCRmn register in the count allowed state varies for each mode of operation (referto Table 6-6 for 6.5.2 Start Order). When channel 1 and 3 are in 8-bit timer mode, TSm1 and TSm3 allow the operation of the low 8-bit timer.

Note:

1) bit15 ~ 12, 10, 8 ~ 4 must be set.

2) When switching from a function that never uses TImn pin input to a function that uses TImn pin input, the wait period from setting timer mode register mn (TMRmn) to TSmn (TSHm1, TSHm3) position '1' is required:

TImn Pin Noise Filter Valid Time (TNFENmn=1): 4 Runtime Clocks ($_{fMCK}$)

TImn Pin Noise Filter Invalid Time (TNFENmn=0): 2 Runtime Clocks ($_{fMCK}$)

Note: 1. The read value for the TSm register is always "0".

2. m: Cell Number (m= 0)n: Channel Number (n=0 ~3)

6.3.7 Timer channel stop register m (TTm).

The TTm register is a trigger register that sets that count stop for each channel.

If each position is" 1", the counter bits of the timer channel allow state register m (TEm) are cleared. Because the TTmn bit, TTHm1 bit, and TTHm3 bit are trigger bits, the TTmn bit, TTHm1 bit, and TTHm3 bit are cleared immediately if the idle state (TEmn, TEHm1, TEHm3=0) occurs.

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

A lower 8-bit of the TTm register can be set with the TTmL and through 8-bit memory operation instructions. After the reset signal is generated, the value of the TTm register changes to "0000H".

Figure 6-15 Table for 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	The High 8-bit Timer Stop Trigger When Channel 3 is in 8-bit Timer Mode
0	No trigger.
1	Clear TEHm3 bit '0' to count stop state.

TTHm1	The High 8-bit Timer Stop Trigger When Channel 1 is in 8-bit Timer Mode
0	No trigger.
1	Clear TEHm1 bit '0' to count stop state.

TTmn	Idle trigger for channel n
0	No trigger.
1	Clear the TEmn bit "0" and enter the count stop state. When channel 1 and 3 are in 8-bit timer mode, TTm1 and TTm3 are triggered to stop operation of low 8-bit timers.

Note: The bit15~12,10,8~4 must be 0.

Note: 1.TTm register always reads "0".

2.m: Cell number (m=0)n: Channel Number (n=0~3)

6.3.8 Timer Input-Output Selection Register (TIOS0, TIOS1)

A channel 0 and a timer input of the channel 1 of the TIOS0 register selection unit 0 and a timer output of the channel 2. The TIOS0 register is set by the 8-bit memory operation instruction. After the reset signal is generated, the value of the TIOS0 register changes to "00H".

Figure 6-16 Timer Input Select Register 0 (TIOS0) table of Tables

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 by channel 0
0	0	0	Timer input pin (TI0) input signal
0	0	1	Output signal TMIOB0 from TimerM (without PWMOP)
0	1	0	Output signal TMIOD0 from TimerM (without PWMOP)
0	1	1	Output signal TMIOA1 from TimerM (without PWMOP)
1	0	0	Output signal TMIOC1 from TimerM (without PWMOP)
1	0	1	Output signal TMIOB1 from TimerM (without PWMOP)
1	1	0	Output signal TMIOD1 from TimerM (without PWMOP)
1	1	1	Output signal TMIOC0 from TimerM (without PWMOP)

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

TOS03	Enable of timer output for channel 2
0	Allow Output
1	Suppress output (output fixed to 0)

TIS02	TIS01	TIS00	Selection of timer input used by 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 internal oscillator clock (f_{IL})
1	0	1	Secondary System Clock (f_{SUB})
Other than above			Disable setting.

Note: 1. The selected timer input must have a high level width greater than or equal to $1/f_{MCK}$+10ns. Therefore, when you select fSUB as fCLK (CSS=1 of the CKC register), you cannot position "1".

2. When selecting the event input signal of ELC through timer input selection register 0 (TIOS0), fCLK must be selected through timer clock selection register 0 (TPS0).

A timer input of a channel 2 of the TIOS1 register selection unit 0. The TIOS1 register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of the TIOS1 register changes to "00H".

Figure 6-17 Timer Input Table for Selection Register 1 (TIOS1)

Address: 0x40020475 After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
TIOS1	0	0	0	0	0	0	0	TIS10

TIS10	Selection of timer inputs used by channel 2
0	Timer input pin (TI02) input signal
1	TSOUT output from CAN Controller

6.3.9 timer output allow register m (TOEm)

The TOEm register is a register that sets the output of each channel timer to be allowed or disabled.

For the channel n, the value of the TOmn bit of the timer output register m (TOm) cannot be rewritten by software.

The TOEm register is set by the 16-bit memory operation instruction.

A lower 8-bit of the TOEm register can be set with the TOEmL and through an 8-bit memory operation instruction. After the reset signal is generated, the value of the TOEm register changes to "0000H".

Figure 6-18 The timer outputs a table that allows the 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	TOEm3	TOEm2	TOEm1	TOEm0

TOEmn	Allow/Disable of timer output for channel n
0	Disables timer output. The operation of the timer is not reflected to the TOmn bit, and the output is fixed. Can write TOmn bits and output the level of the TOmn bit setting from the TOmn pin.
1	Allow timer output. The operation of the timer is reflected to the TOmn bit, and the output waveform is generated. Writes of TOmn bits are ignored.

Note: bit15~4 must be placed at 0.

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

6.3.10 Timer output register m (TOm)

The TOm register is a buffer register output by each channel timer.

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

The TOmn bit of this register can only be overwritten by software if timer output (TOEmn=0) is prohibited. When timer output is allowed (TOEmn=1), the override operation through the software is ignored and the value is changed only through the timer running.

To use the TI00, TO00, TI01/TO01, TI02/TO02, TI03/TO03 pins as port functions, you must use the TOmn location "0".

The TOm register is set by a 16-bit memory operation instruction.

The low 8 bits of the TOm register can be set with TOmL and through 8-bit operation instructions. After the reset signal is generated, the value of the TOm register becomes "0000H".

Figure 6-19 Table for 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 for channel n
0	The output value of the timer is "0".
1	The output value of the timer is "1".

Note: bit15~4 must be placed at 0.

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

6.3.11 Timer output level register m (TOLm)

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

When a timer output (TOEmn=1) is allowed and a multi-channel coordinated operation function (TOMmn=1) is used, the setting and reset timing of the output signal of the timer reflect the reversed setting of each channel n by this register. This register has an invalid setting in the TOMmn=0 (Master Channel Output Mode).

The TOLm register is set by the 16-bit memory operation instruction.

A low 8-bit of the TOLm register can be set with TOLmL and through 8-bit memory instructions. After the reset signal is generated, the value of the TOLm register changes to "0000H".

Figure 6-20 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 Logical Output (High Level Valid)
1	Invert output (low level active)

Note: bit15~4 and bit0 must be set to'0'.

Note:1. If you override the value of this register during a timer run, invert the output logic of the timer when the next timer output signal changes, not immediately after the override.

2.m: Cell number (m=0)n: Channel Number (n=0~3)

6.3.12 Timer output mode register m (TOMm)

The TOMm register is a register that controls the output mode of each channel timer. When used as a stand-alone channel operation function, the corresponding position of the used channel is "0".

When used as a multi-channel coordinated operation function (PWM output, single trigger pulse output and multiple PWM output), the corresponding position of the main control channel is "0" and the corresponding position of the subordinate channel is "1".

When a timer output (TOEmn=1) is allowed, the setting and reset timing of the timer output signal reflect the setting of each channel n performed by the register.

The TOMm register is set by the 16-bit memory operation instruction.

A low 8-bit of the TOMm register can be set with TOMmL and through 8-bit memory instructions. After the reset signal is generated, the value of the TOMm register changes to "0000H".

Figure 6-21 Table for 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	TOMm3	TOMm2	TOMm1	0

TOMmn	Control of timer output mode of channel n
0	Main control channel output mode (alternating output by timer interrupt request signal (INTTMmn))
1	Slave channel output mode (the output is set by timer interrupt request signal (INTTMmn) of master channel and reset by timer interrupt request signal (INTTMmp) of slave channel)

Note: bit15~4 and bit0 must be set to'0'.
Note: m: Unit number (m=0); n: Channel Number n=0~3 (when the primary channel is: n=0, 2)
p: Dependent Channel Number
n=0:p=1, 2, 3
n=2:p=3
(For details on the relationship between master and slave channels, refer to the "Basic Rules for Multi-Channel Operation Capabilities")

6.3.13 Input switch control register (ISC)

The ISC1 bit and ISC0 bit of the ISC register are used for the coordination of channel 3 and universal serial communication unit to realize LIN bus communication. If the ISC1 position" 1", the input signal of the serial data input pin (RxD0) is selected as the input of the timer.

Refer to "19.3.14 Input Switch Control Register (ISC)" for setting SSIE00 bits. The ISC register is set by an 8-bit memory operation instruction.

After the reset signal is generated, the value of the ISC register changes to "00H".

Figure 6-22 Enter a table for the Switch Control Register (ISC)

Address: 0x40040473 After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
ISC	SIE00	0	0	0	0	0	ISC1	ISC0

SIE00	SSI00 pin input settings for channel 0 in the dependent mode of CSI00 communication
0	Invalid SSI00 pin input.
1	SSI00 pin input is valid.

ISC1	Input switching of channel 3 of universal timer unit 0
0	Use the input signal of the TI03 pin as the input (usually running) of the timer.
1	The input signal of the RxD0 pin is used as the input of the timer (detecting the wake-up signal and measuring the low level width of the interval segment and the pulse width of the synchronization segment).

ISC0	External Interrupt (INTP0) input switch
0	Use the input signal of the INTP0 pin as the input for the external interrupt (usually run).
1	Use the input signal of the RxD0 pin as the input of the external interrupt (detect wake-up signal).

Note: bit6~2 must be set to'0'.

Note: When using LIN-bus for communication, you must select the input signal for the RxD0 pin by selecting the ISC1 position "1".

6.3.14 Noise Filter Admission Register (NFEN1)

The NFEN1 register sets whether the noise filter is used for the input signal of the input pin of each channel timer. For pins that need to be eliminated from noise, the corresponding position "1" must be taken to make the noise filter effective. When the noise filter is effective, the two clocks are detected $_{after}$ synchronization by the fMCK of the object channel. When the noise filter is not valid, synchronization is performed only through the object channel's runtime clock（f_{MCK}）note.

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

Note: Refer to " CCSmn=1 (2) Selecting Valid Edges of TImn Pin Input Signals" 6.5.2 Start Sequence" and "Control of 6.7 Timer Input (TImn).

Figure 6-23 Table of Noise Filter Allow Register 1(NFEN1)

Address: 0x40040471 After reset: 00H R/W

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

TNFEN03	Whether the TI03 pin or the input signal noise filter of the RxD0 pin is used or not
0	noise filter OFF
1	noise filter ON

TNFEN02	Whether the Input Signal Noise Filter of the TI02 Pin Is Used or Not
0	noise filter OFF
1	noise filter ON

TNFEN01	Whether the Input Signal Noise Filter of the TI01 Pin Is Used or Not
0	noise filter OFF
1	noise filter ON

TNFEN00	Whether the Input Signal Noise Filter of the TI00 Pin Is Used or Not
0	noise filter OFF
1	noise filter ON

Note The applicable pin can be switched by setting the ISC1 bit of the input switch control register (ISC). ISC1=0: You can choose whether to use a noise filter for the TI03 pin. ISC1=1: You can choose whether to use a noise filter for the RxD0 pin.

Note: Whether the timer input/output pins of channels 0 to 3 are different or not depends on the product. Refer to "Table 6-2 Products with Timer Input/Output Pins".

6.3.15 Register for controlling timer input/output pin port function

When using a universal timer unit, you must set the port-functional control registers (PMxx, Pxx, and PMCxx). Refer to "2.3.1 Port Mode Register (PMxx), "2.3.2 Port Register (Pxx)" and "2.3.6 Port Mode Control Register (PMCxx) " for details.

The set port mode registers (PMxx), port registers (Pxx), and port mode control registers (PMCxx) differ by product. Refer to "Register settings when 2.5 uses the multiplexing feature" for details.

When the multiplexed port of the timer output pin is used as the output of the timer, the bit of the port mode control register (PMCxx), the bit of the port mode register (PMxx) and the

(example) of using P01/TO00 as timer output

Position PMC01 of port mode control register 0 at "0.

Position PM01 of port mode register 0 "0"

Position P01 for port register 0 "0.

When the multiplexed port of the timer input pin is used as the input of the timer, the position of the port mode register (PMxx) 1. The bit of the port register (Pxx) can be "0" or "1".

(Example) of using P00/TI00 as timer input

Position PMC00 of port mode control register 0 at "0.

Position PM00 for port mode register 0 at "1.

Position P00 of port register 0 "0" or "1.

6.4 Basic rules for universal timer units

6.4.1 Basic Prinicpal of Multi-channel Coordinated Operation Function

The function of multi-channel coordinated operation is a function which combines the main control channel (the reference timer which mainly counts the period) and the slave channel (the timer which follows the main control channel operation), and it needs to abide by several rules.

The basic rules for the multi-channel coordinated operation function are as follows.

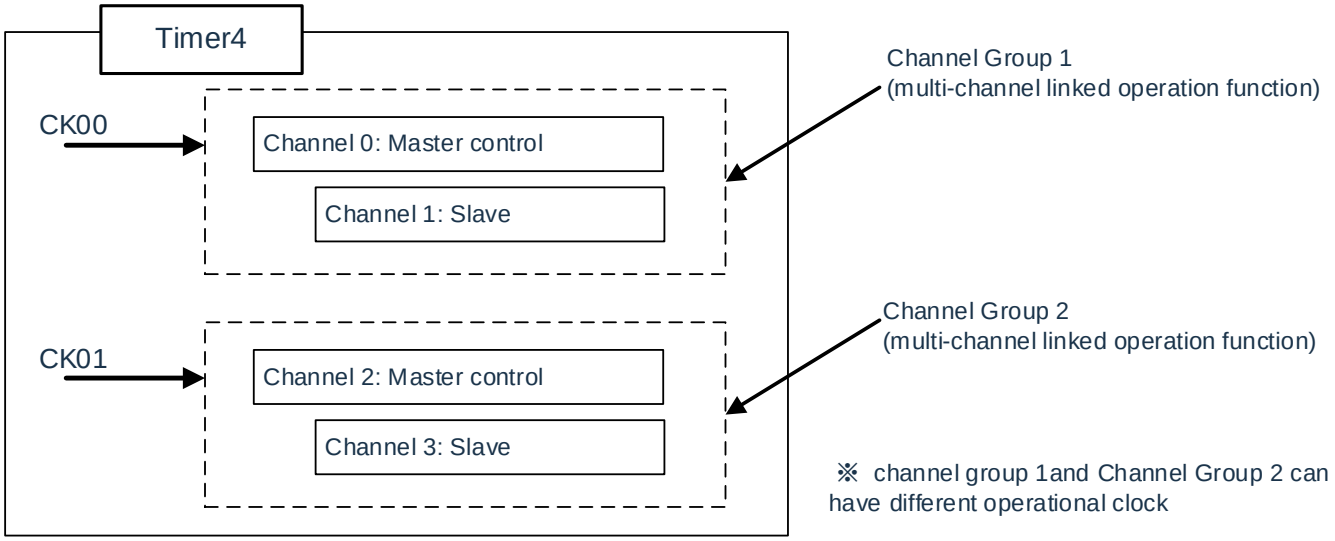
1) Only even channels (channel 0, channel 2) can be set as master channels.
2) Any channel other than channel 0 can be set as a slave channel.
3) You can only set the lower channel of the master channel as a slave channel.
   When setting channel 0 as the master channel, the channel starting channel 1 (channel 1, channel 2, channel 3) can be set as slave.
4) Multiple slave channels can be set for one master channel.
5) When using multiple master channels, you cannot set a slave channel across the master channel.
   When setting channel 0 and channel 2 as the master channel, channel 1 can be set as the slave channel of master channel 0.
6) (6) The slave channel which is linked with the master channel needs to set the same runtime clock. The CKSmn0 bit and the value of CKSmn1 bit (bit15 and bit14 of timer mode register mn(TMRmn) of the slave channel interfaced with the master channel.
7) (7) The main control channel can transmit INTTMmn, start software trigger and count clock to the low-level channel.
8) The slave channel can use the INTTMmn, start software trigger and count clocks of the master channel as the source clock, but cannot pass its INTTMmn, start software trigger
9) The master channel cannot use the INTTMmn, start software trigger, and count clocks of other high-level master channels as source clocks.
10) In order to start the channel to be linked at the same time, it is necessary to set the channel start trigger bit (TSmn) of the linked channel.
11) Only the full channel or master channel of the coordination can use the settings of the TSmn bits in the count run. You cannot use the settings of only the TSmn bits of the dependent channel.
12) In order to simultaneously stop the channel to be linked, it is necessary to simultaneously set the channel stop trigger bit (TTmn).
13) When the Coordinated Operation is running, you cannot select CKm2/CKm3 because the master channel and the slave channel require the same runtime clock.
14) The timer mode register m0 (TMRm0) is fixed to '0' without a master bit. However, because channel 0 is the highest-bit channel, channel 0 can be used as the master channel during the interaction run.

The basic rules of the multi-channel coordinated operation function are applicable to the channel group (forming a set of master and slave channels).

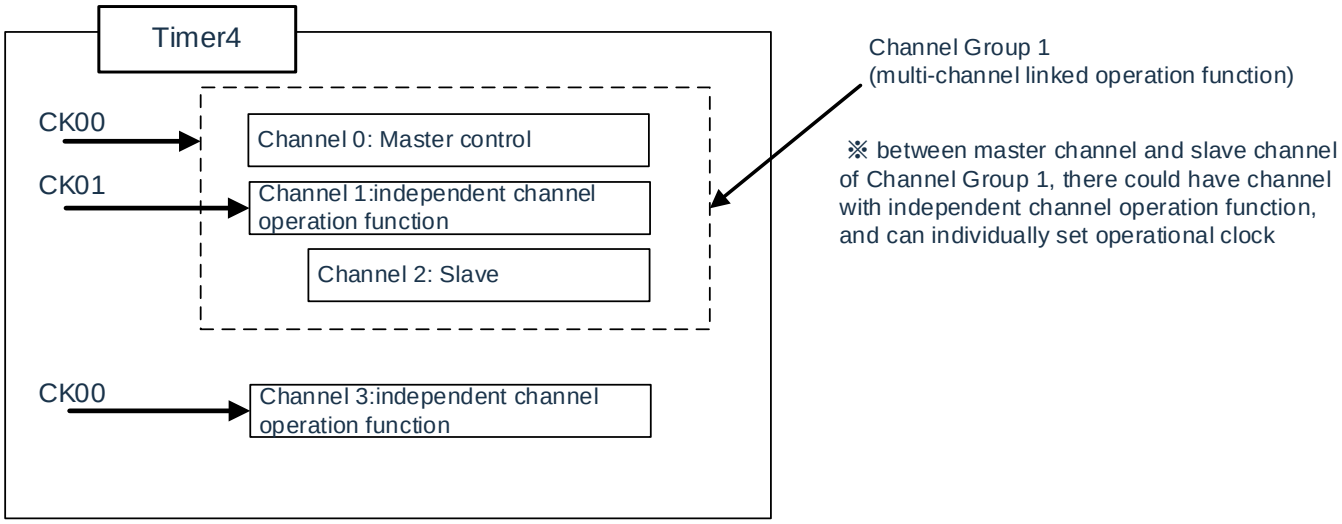
If two or more channels groups are set, the basic rules are not applicable to each other.

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

Example 1



Example 2



6.4.2 Basic Principal of the 8-bit timer operation function (only for channel 1 and channel 3).

The 8-bit timer run function is the function of using the channel of the 16-bit timer as the channel of two 8-bit timers.

Only channel 1 and channel 3 can run functions with an 8-bit timer, which requires several rules.

The basic rules for the 8-bit timer operation function are as follows.

1) The 8-bit timer run function is available only for channel 1 and channel 3.
2) When used as an 8-bit timer, the SPLIT position of timer mode register mn(TMRmn) is "1".
3) High-8-bit timers can be used as a spacer function.
4) The high 8-bit timer outputs INTTMm1H/INTTMm3H (the same as the run with the MDmn0 bit "1") at start.
5) The selection of the running clock of the high 8-bit timer depends on the setting of the CKSmn1 bit and the CKSmn0 bit of the low TMRmn register.
6) For a high 8-bit timer, the operation of the channel is started by operating the TSHm1/TSHm3 bit, and the operation of the channel is stopped. The status of the channel can be confirmed by TEHm1/TEHm3 bits.
7) The operation of the low-8-bit timer depends on the setting of the TMRmn register, which has three functions to support the operation:
   - interval timer function
   - External event counter feature
   - Delay Count Feature
8) For low-8 bit timers, channel operation is started by operating the TSm1/TSm3 bit, and channel operation is stopped by operating the TTm1/TTm3 bit. The status of the channel can be confirmed by TEm1/TEm3 bits.
9) The operation of the TSHm1/TSHm3/TTHm1/TTHm3 bit is not valid when the 16 bit timer runs. Channel 1 and channel 3 are run by operating TSm1/TSm3 and TTm1/TTm3 bits. The TEHm3 bit and the TEHm1 bit are unchanged.
10) The 8-bit timer function does not use the coordinated operation function (single trigger pulse, PWM, and multiple PWM).

Note: m: Cell number (m=0)n: Channel number (n=1,3)

6.5 Operation of counters

6.5.1 Count Clock (f_{TCLK})

The counting clock (f_{TCLK}) of the universal timer unit can select any one of the following clock by the CCSmn bit of the timer mode register mn (TMRmn):

- CKSmn0-bit and CKSmn1-bit specified runtime clocks (f_{MCK})
- Effective Edge of TImn Pin Input Signal

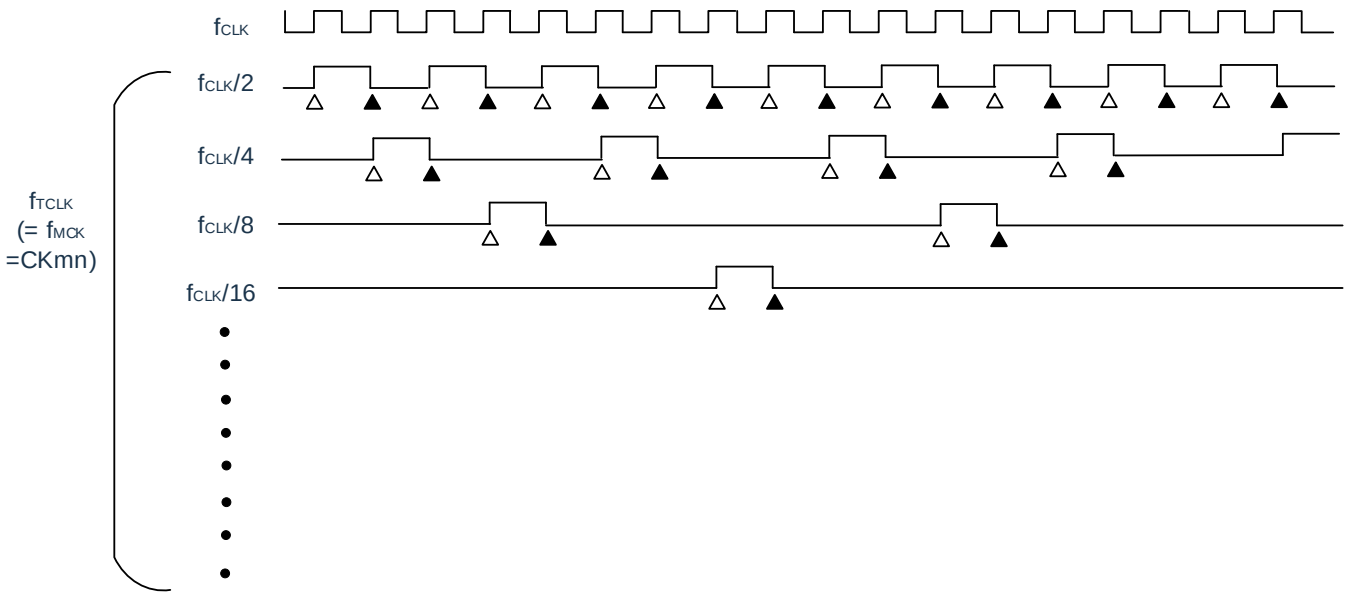
The universal timer unit is designed to run synchronously with the fCLK, so the timing of the counting clock (f_{TCLK}) is as following.

(1) Select the case of the CKSmn0 bit and the CKSmn1 bit specified runtime clock (f_{MCK}) (CCSmn=0)

According to the setting of timer clock selection register m (TPSm), the count clock (f_{TCLK}) is f_{CLK}～$f_{CLK}/2^{15}$. However, when selecting the frequency division of the fCLK, the TPSm register selects a clock that is only 1 f_{CLK} period of high level signal from the rising edge. Fixed to high level when f_{CLK} is selected.

In order to synchronize with f_{CLK} , timer counter register mn (TCRmn) counts after delaying 1 f_{CLK} clock from the rising edge of the counting clock.

Figure 6-24 Timing of fCLK and count clock (fTCLK) (CCSmn=0 case)



Remarks: 1.△: Rising edge of counting clock

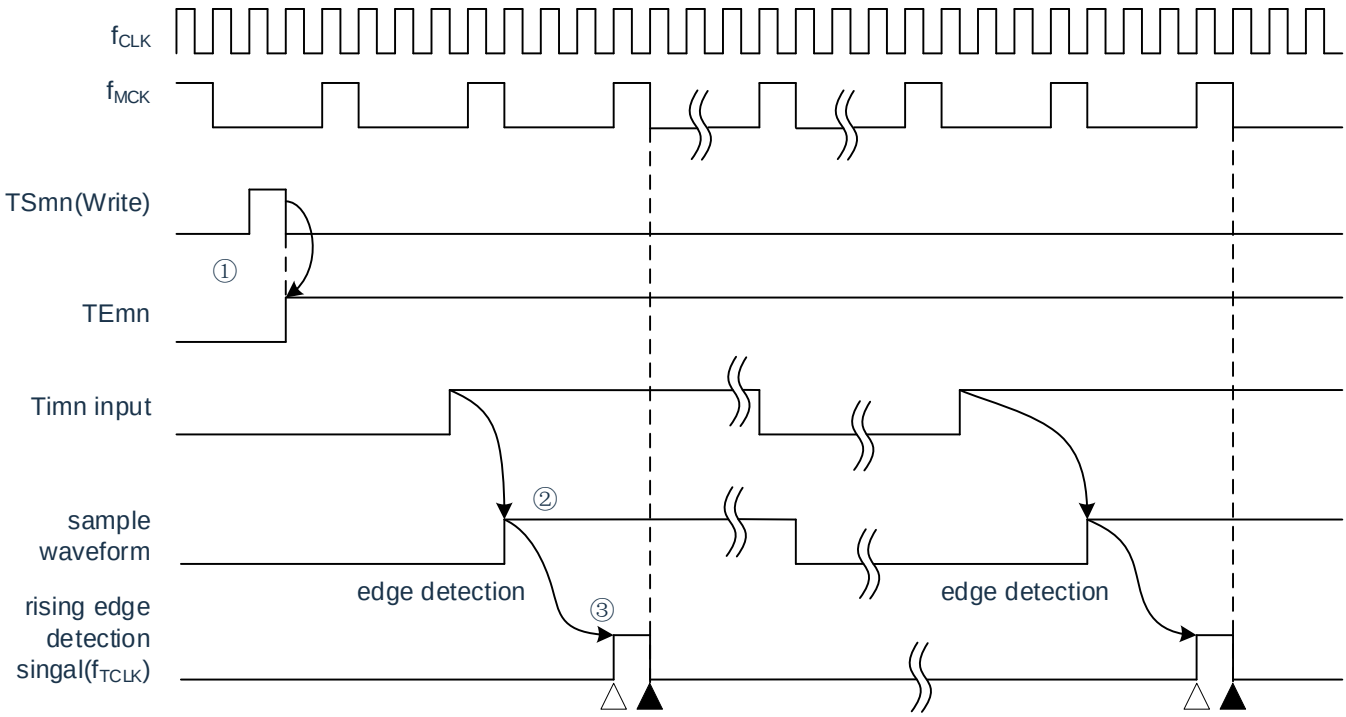
▲: Synchronization, Counter Increment/Decrement

2. f_{CLK}: Clock for CPU/peripheral hardware

(2) The case of selecting a valid edge of the TImn pin input signal (CCSmn=1)

The count clock (fTCLK) is a signal that detects an effective edge of the TImn pin input signal and is synchronized with the next fMCK rising edge. In fact, this is a signal delayed by 1~2 fMCK clocks compared to the input signal of the TImn pin (3~4 fMCK clocks when using noise filters). In order to synchronize with fCLK, timer counter register mn(TCRmn) counts after delaying 1 fCLK from the rising edge of the counting clock, which is called "counting at the effective edge of TImn pin input signal".

Figure 6-25 Counts the timing of the clock (fTCLK) (CCSmn=1, if no noise filter is used)



① The operation of the timer is started by the TSmn position bit and the valid edge of the TImn input is awaited.

② Sample the rising edge of the TImn input through fMCK.

③ A detection signal (counting clock) is output at a rising edge of the sampling signal.

Remarks: 1.△: Rising edge of counting clock

▲: Synchronization, Counter Increment/Decrement

2. f_{CLK}: CPU/Peripheral Hardware Clock

f_{MCK}: Runtime clock for channel n

3. The same waveforms are measured for input pulse interval, input signal high and low level, delay counter, and TImn input with single trigger pulse output.

6.5.2 Starting sequence of Counter

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

Execution from counting enable state up to the start of count register mn (TCRmn) is shown in Table 6-6.

Table 6-6 Run from the count allowed state until the timer count register mn (TCRmn) starts to count

Operation mode of the timer	Run after setting TSmn position as " 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 1 count clock and decremented by the subsequent count clock (refer to "6.5.3 (1) interval timer mode operation").
Event Counter Mode	Load the value of the TDRmn register into the TCRmn register by writing "1" to the TSmn bit. If an input edge of the TImn is detected, the count is decremented by a subsequent count clock. (Refer to "6.5.3(2) Run of Event Counter Mode").
capture mode	No operation is performed from the start of triggering of detection until the count clock is generated. The 0000H is loaded into the TCRmn register by the 1 count clock and counted incrementally by the subsequent count clock (reference to "6.5.3 (3) capture mode operation (interval measurement of input pulses)).
single count mode	The first trigger is entered by writing "1" to the TSmn bit in the state where the timer stops running (TEmn=0) Pending status. No operation is performed from the start of triggering of detection until the count clock is generated. The value of the TDRmn register is loaded into the TCRmn register by the first count clock, and a subsequent count is performed Count clocks for decremental counts (refer to "Operation of 6.5.3 (4) Single Count Mode").
Capture & Single Count Mode	The first trigger is entered by writing "1" to the TSmn bit in the state where the timer stops running (TEmn=0) Pending status. No operation is performed from the start of triggering of detection until the count clock is generated. The '0000H' is loaded into the TCRmn register through the first count clock and is performed through a subsequent count clock Incremental Count (refer to " 6.5.3(5) Operation of Capture & Single Count Mode (Measurement of High Level Width) "

6.5.3 Operation of counters

The following describes the counter operation for each mode.

(1) Operation of interval timer mode

(1) Enter the running permission state (TEmn=1) by writing "1" to TSmn bits. The timer count register mn (TCRmn) maintains an initial value until a count clock is generated.

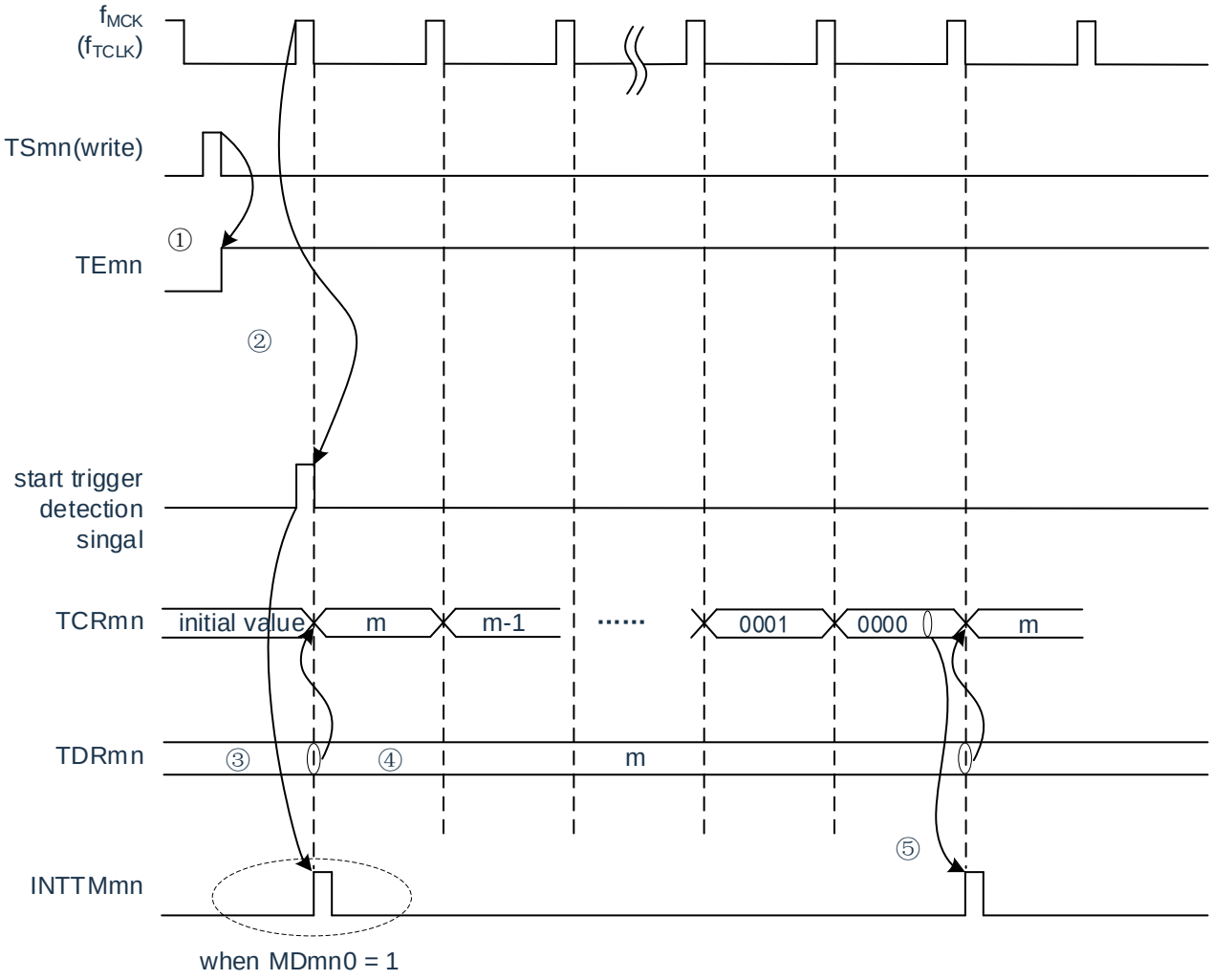
(2) A start trigger signal is generated by allowing the first count clock (fMCK) after operation.

(3) When the MDmn0 bit is "1", the INTTMmn is generated by the start trigger signal.

(4) The timer data register mn (TDRmn) is loaded into the TCRmn register by allowing the first counting clock to run.

(5) If the TCRmn register decrements to "0000H", INTTMmn is generated by the next counting clock (fMCK) and continues counting.

Figure 6-26 Runtime Sequence (Interval Timer Mode)



Note: Since the operation of the first count clock cycle is delayed after writing the TSmn bit and before the count clock is generated, an error of up to 1 clock cycle is generated. In addition, if information on that time sequence of the start count is required, the MDmn0 position" 1" is place so that an interrupt can occur at the start count.

Note: $_{fMCK}$, the start trigger heartbeat, and INTTMmn are synchronized with fCLK and are valid for 1 clock.

(2) Run of event counter mode

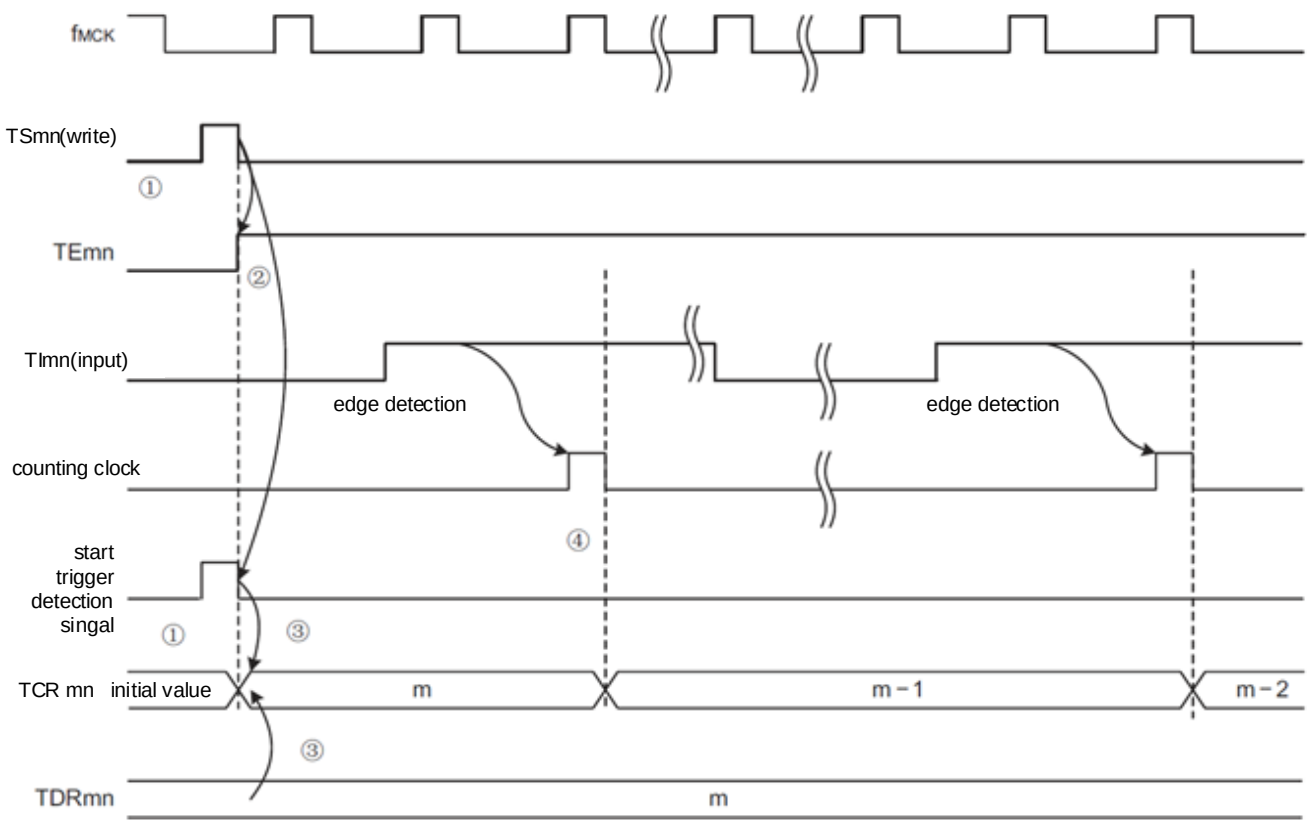
(1) The timer count register mn (TCRmn) maintains the initial value during the operation of the stop state (TEmn=0).

(2) Enter the running permission state (TEmn=1) by writing "1" to TSmn bits.

(3) Loading the value of timer data register mn (TDRmn) into the TCRmn register when both the TSmn bit and the TEmn bit become "1".

(4) Then, the value of the TImn register is decremented by the counting clock at the effective edge of the TCRmn input.

Figure 6-27 Runtime Sequence (Event Counter Mode)



Note: This is the timing when no noise filter is used. If a noise filter is used, edge detection delays an additional 2 fMCK cycles (3-4 cycles total) from the TImn input time. The error of 1 cycle is due to the TImn input being out of sync with the count clock (fMCK).

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

(1) Enter the running permission state (TEmn=1) by writing "1" to the TSmn bit.

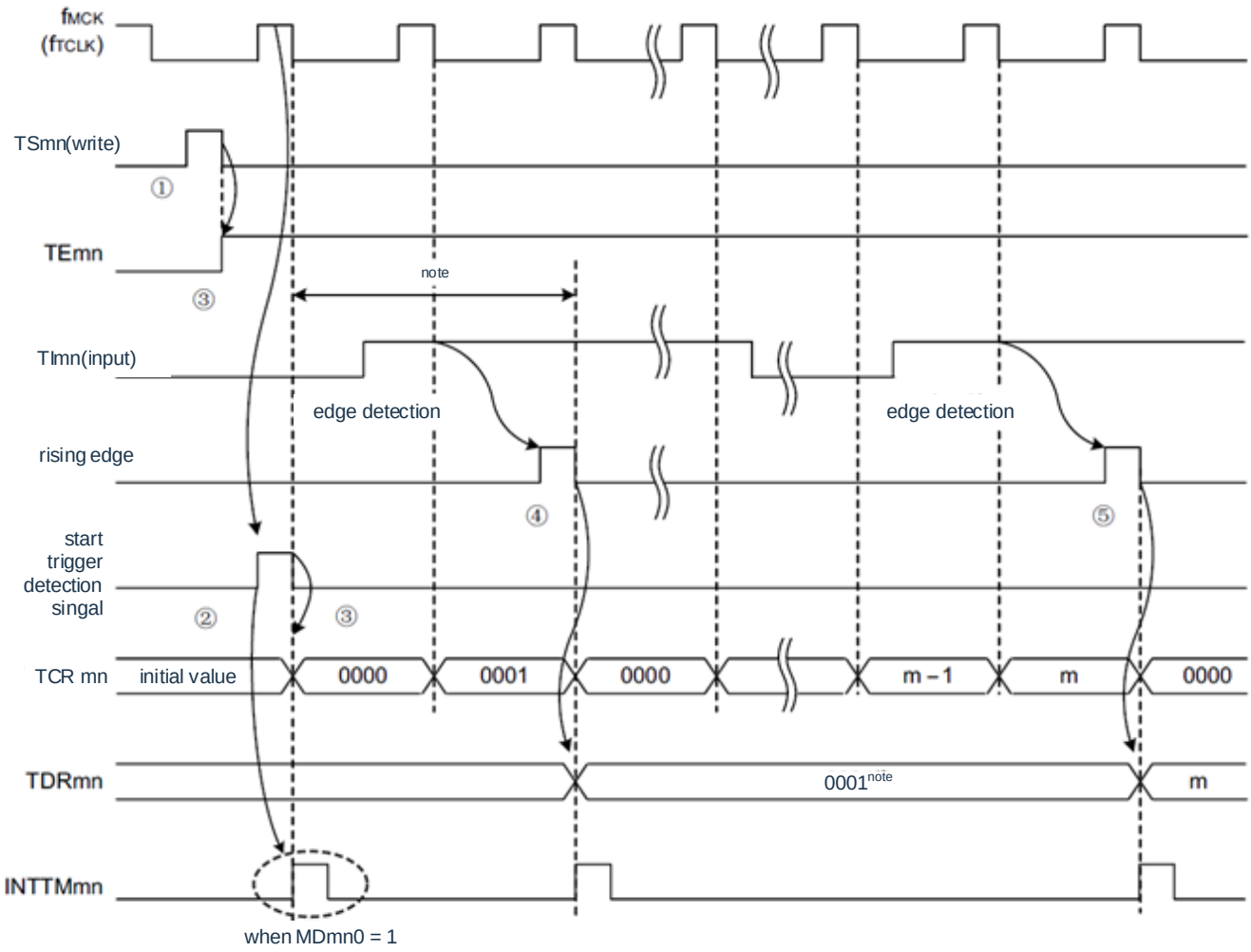
(2) The timer count register mn(TCRmn) keeps the initial value until the count clock is generated.

(3) A start trigger signal is generated by allowing the first count clock (fMCK) after operation. The '0000H' is then loaded into the TCRmn register and counts in capture mode (INTTMmn is generated by the start trigger signal when the MDmn0 bit is '1')

(4) If the valid edge of the TImn input is detected, the value of the TCRmn register is captured to the TDRmn register and INTTMmn interrupt is generated. The catch value at this time is meaningless. The TCRmn register continues to count from '0000H'.

(5) If the valid edge of the next TImn input is detected, the value of the TCRmn register is captured to the TDRmn register and INTTMmn interrupt is generated.

Figure 6-28: Runtime sequence (capture mode: interval measurement of input pulse)



Note: When the clock is input to the TImn (triggered) before starting, the count is started by detecting the trigger even if no edge is detected, therefore the capture value of the first capture is not pulse interval (in this example, 0001: 2 clock cycle interval), it shall be ignored.

Note: Since the operation of the first count clock cycle is delayed after writing the TSmn bit and before the count clock is generated, an error of up to 1 clock cycle is generated. In addition, if information on that time sequence of the

start count is required, the MDmn0 position" 1" is place so that an interrupt can occur at the start count.

Note: This is the timing when no noise filter is used. If a noise filter is used, edge detection delays an additional 2 fMCK cycles (3-4 cycles total) from the TImn input time. The error of 1 cycle is due to the TImn input being out of sync with the count clock (fMCK).

(4) Run of single count mode

(1) Enter the Run Allowed state (TEmn=1) by writing "1" to the TSmn bit.

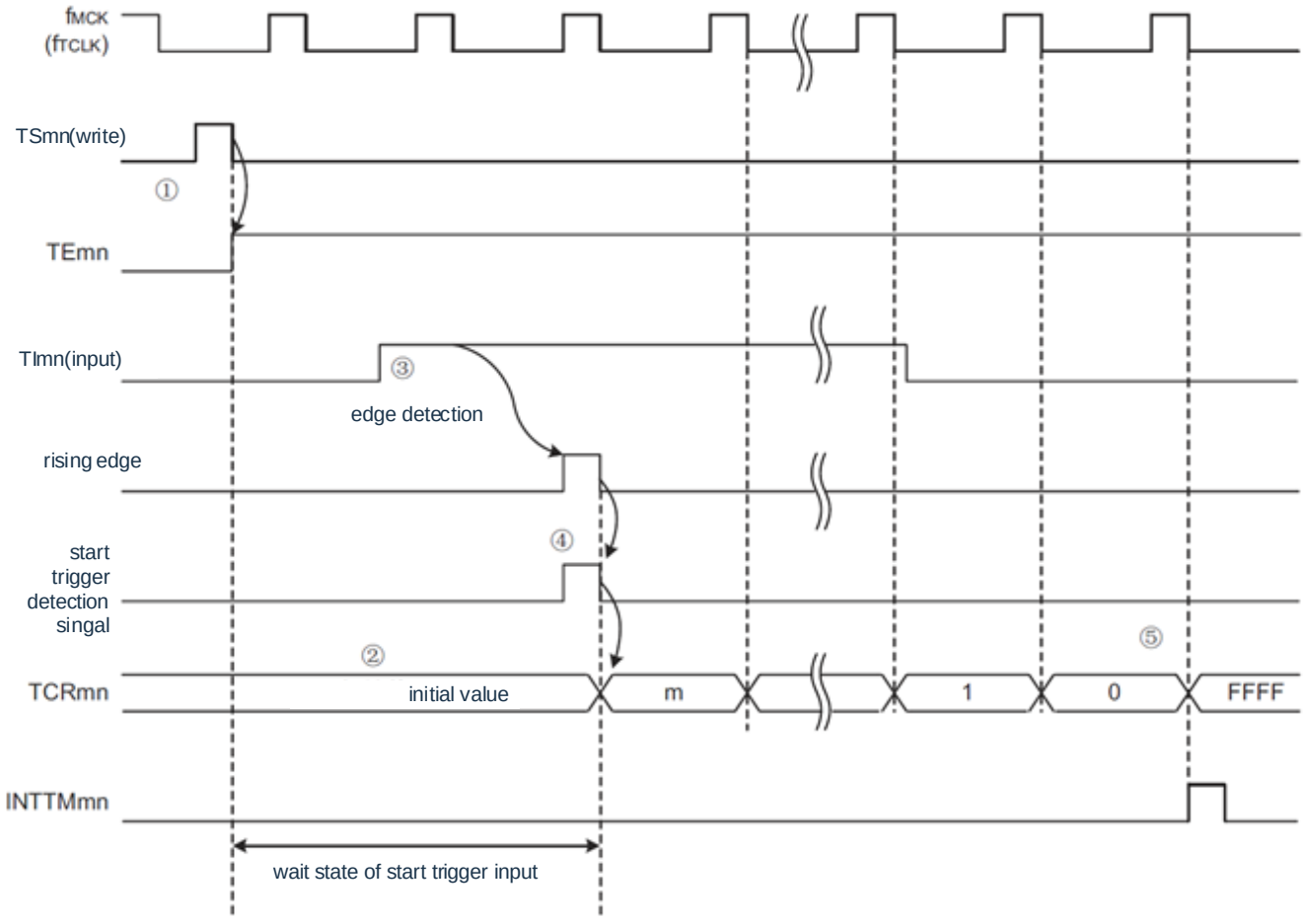
(2) The timer count register mn(TCRmn) keeps the initial value until the start trigger signal is generated.

(3) Detect the rising edge of TImn input.

(4) After the start trigger signal is generated, the value (m) of the TDRmn register is loaded into the TCRmn register and counting is started.

(5) When the TCRmn register decrements to "0000H", INTTMmn interrupts are generated, and the TCRmn register changes to "FFFFH" to stop counting.

Figure 6-29 Runtime Sequence (single count mode)



Note: This is the timing when no noise filter is used. If a noise filter is used, edge detection delays an additional 2 fMCK cycles (3-4 cycles total) from the TImn input time. The error of 1 cycle is due to the TImn input being out of sync with the count clock (fMCK).

(5) Capture & Single Count Mode Operation (Voltage High Level Width Measurement)

(1) Enter the running allowed state (TEmn=1) by writing "1" with the TSmn bit of the timer channel start register m.

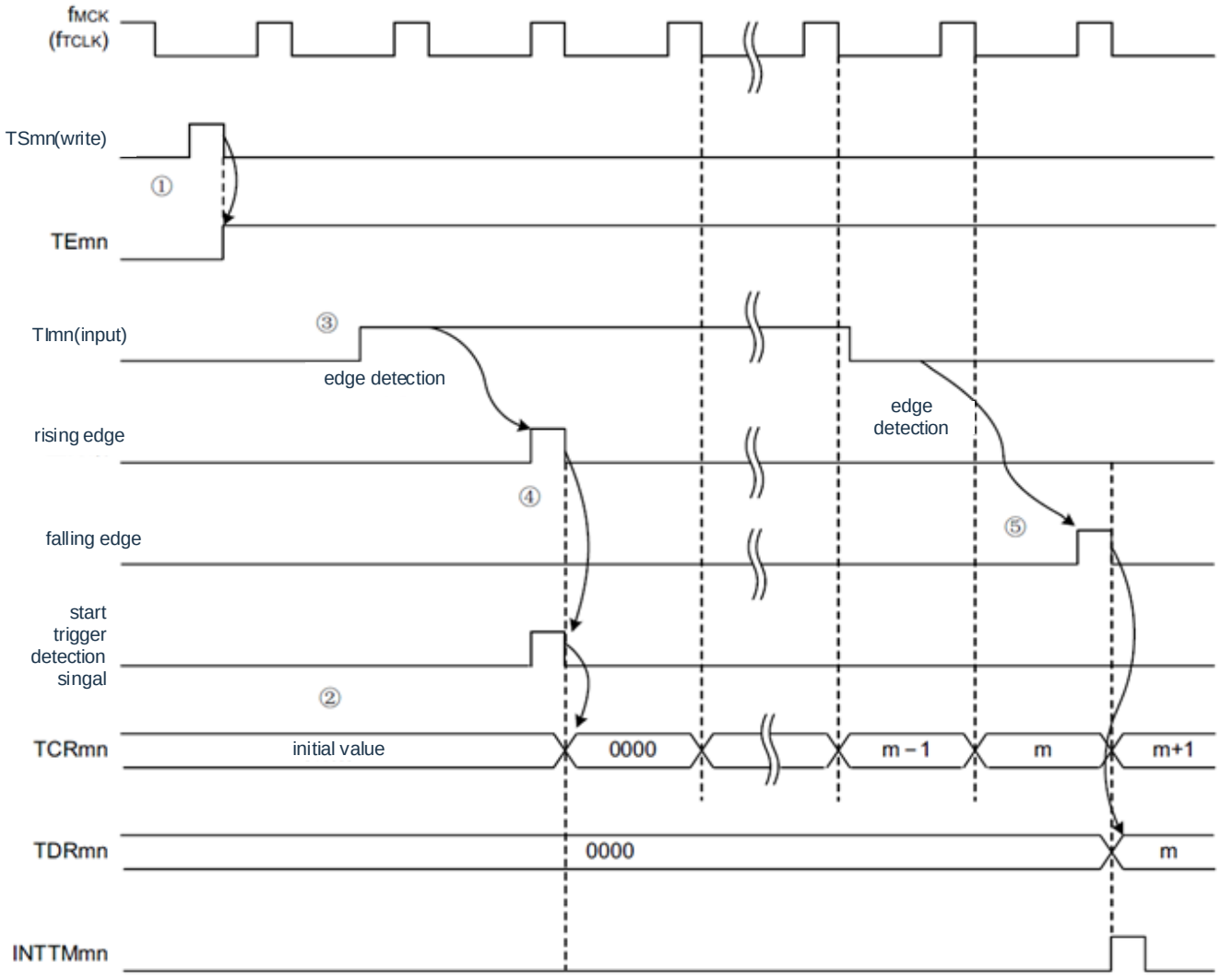
(2) The timer count register mn(TCRmn) keeps the initial value until the start trigger signal is generated.

(3) Detect the rising edge of TImn input.

(4) Loading the "0000H" into the TCRmn register after generating the start trigger signal and starting the counting.

(5) If the falling edge of the TImn input is detected, the value of the TCRmn register is captured to the TDRmn register and INTTMmn interrupt is generated.

Figure 6-30 Runtime Sequence (Capture & Single Count Mode: high level width measurement)

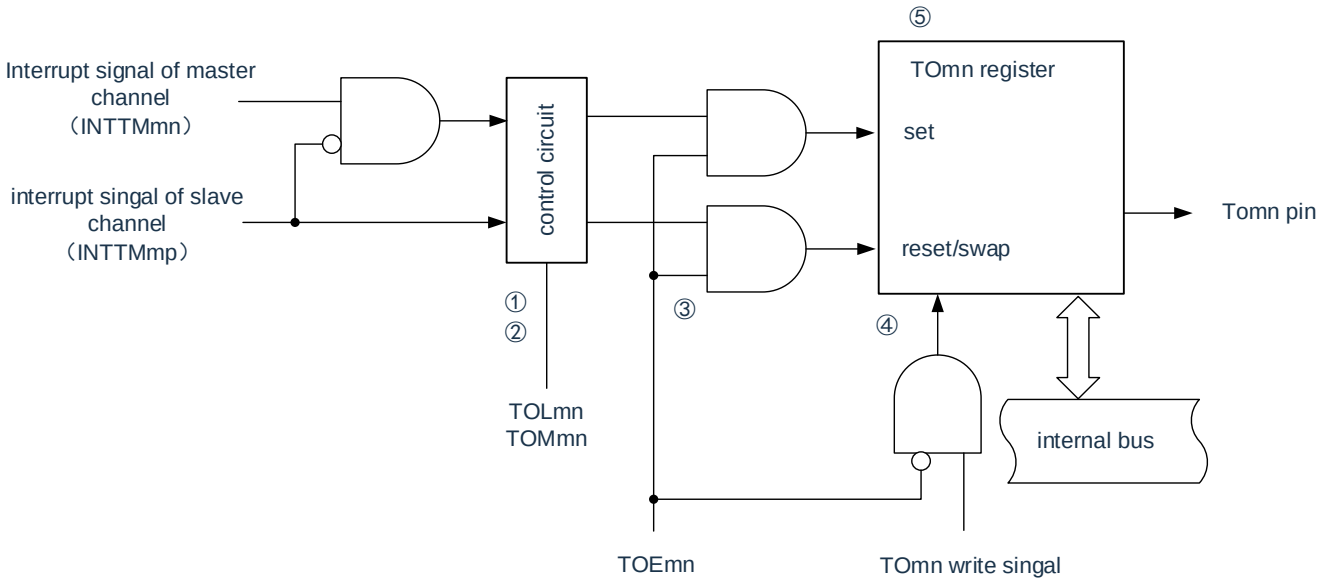


Note: This is the timing when no noise filter is used. If a noise filter is used, edge detection delays an additional 2 fMCK cycles (3-4 cycles total) from the TImn input time. The error of 1 cycle is due to the TImn input and count clock ($_{fMCK}$) being out of sync.

6.6 Control of channel output (TOmn pin)

6.6.1 Structure of TOmn pin output circuit

Figure 6-31 Structure of output circuit



The output circuit for the TOmn pin is described below.

① When the TOMmn bit is "0" (main channel output mode), the timer output level register m (TOLm) is ignored.

② When the TOMmn bit is "1" (slave channel output mode), the INTTMmn and INTTMmp (slave channel timer interrupt) are passed to TOm registers.

At this time, the TOLm register is valid and controls the following signal:

TOLmn=0 time: Forward Running (INTTMmn→ Set, INTTMmp→ Reset)

TOLmn=1 time: Invert run (INTTMmn→ reset, INTTMmp→ set)

When both INTTMmn and INTTMmp are generated (0% of PWM output), INTTMmp is preferentially used to mask INTTMmn.

③ The INTTMmn and INTTMmp are transferred to the TOm register in a state where timer output (TOEmn=1) is allowed. Write operation (TOmn write signal) is invalid for TOm register.

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, write values to the TOm register after being set to disable the timer output (TOEmn=0).

④ Write operations (TOmn write signals) of the TOmn bits of the object channel are valid in the state where timer output (TOEmn=0) is disabled. When the timer output is forbidden (TOEmn=0), INTTMmn and INTTMmp are not transferred to the TOm register.

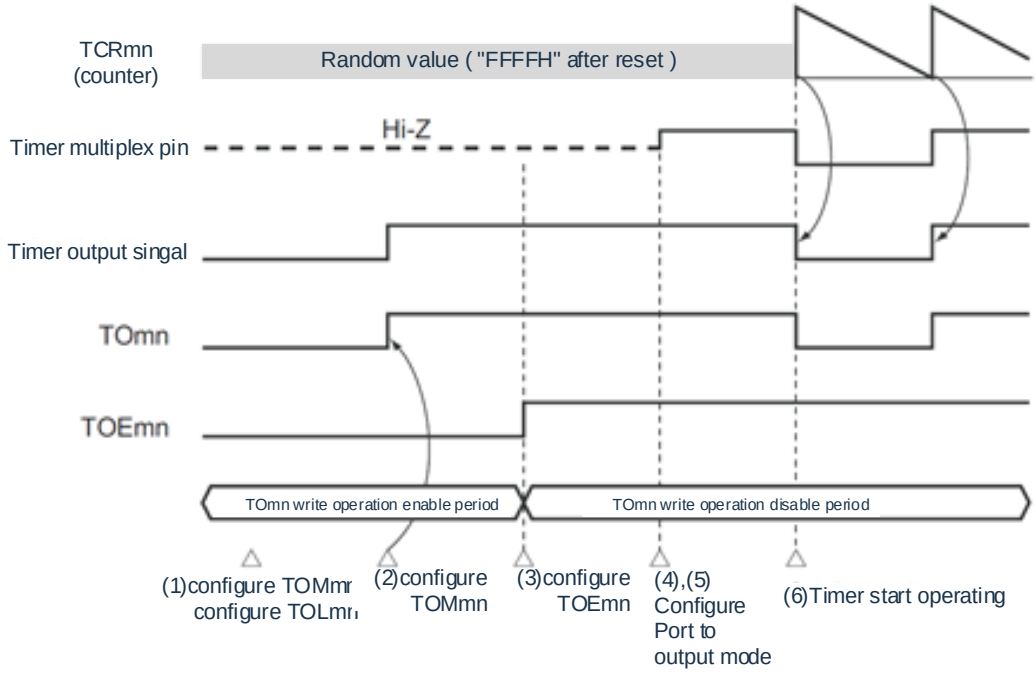
⑤ The TOm register can be read at any time and the output level of the TOmn pin can be confirmed.

Note m: Cell Number (m=0)
n: Channel Number, n=0~3 (Master Channel: n=0,2)
p: Dependent Channel Number
n=0:p=1,2,3
n=2:p=3

6.6.2 Output settings for TOmn pins

The steps and status changes from the initial setting of the TOmn output pin to the start of the timer run are shown below.

Figure 6-32 State change from output of setting timer to start running



(1) Set the operation mode output by the timer.

· TOMmn bit (0: Main control channel output mode,1: slave channel output mode)

· TOLmn bit (0: Positive Logical Output, 1: negative logical output)

(2) Setting the timer output signal to the initial state by setting the timer output register m (TOm).

(3) Write "1" to the TOEmn bit to allow timer output (TOm register forbidden).

(4) Set the port as digital input/output through the port mode control register (PMCxx) (reference to "6.3.15).

(5) Set the port input/output as output (reference "6.3.15 register for controlling the timer input/output pin port function").

(6) Allow the timer to run (TSmn=1).

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

6.6.3 Precaution for Channel Output Operation

1) Change of setting values for TOm, TOEm, TOLm, TOMm registers in timer operation

The operation of the timer (timer count register mn (TCRmn) and timer data register mn (TDRmn)) and the TOmn output circuit are independent. Therefore, the change of timer output register m (TOm), timer output permission register m (TOEm)TOLm does not affect the operation of timer. However, in order to output the expected waveform from the TOmn pin during the operation of each timer, the registers for each operation shown in 6.8 and 6.9 must be set.

If the setting values of TOEm register and TOLm register except TOm register are changed before and after the INTTMmn signal is generated for each channel.

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

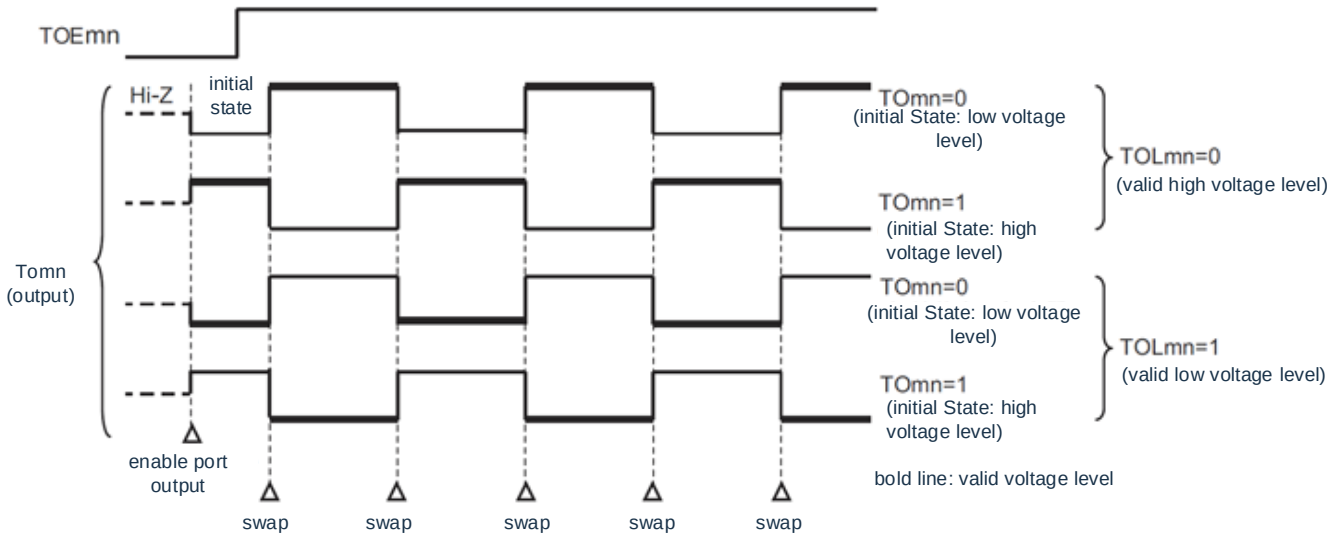
2) Initial level for the TOmn pin and output level after the timer starts to run

A timer output register m (TOm) is written before the port output is allowed and the timer output (TOEmn=0) is prohibited.

(a) The case when the operation starts in TOMmn=0 (Main Control Channel Output Mode)

The timer output level register m(TOLm) is not set in TOMmn=0. If operation of the timer is started after setting the initial level, the output level of the TOmn pin is inverted by generating alternating signals.

Figure 6-33 TOmn pin output state at alternate output (TOMmn=0)



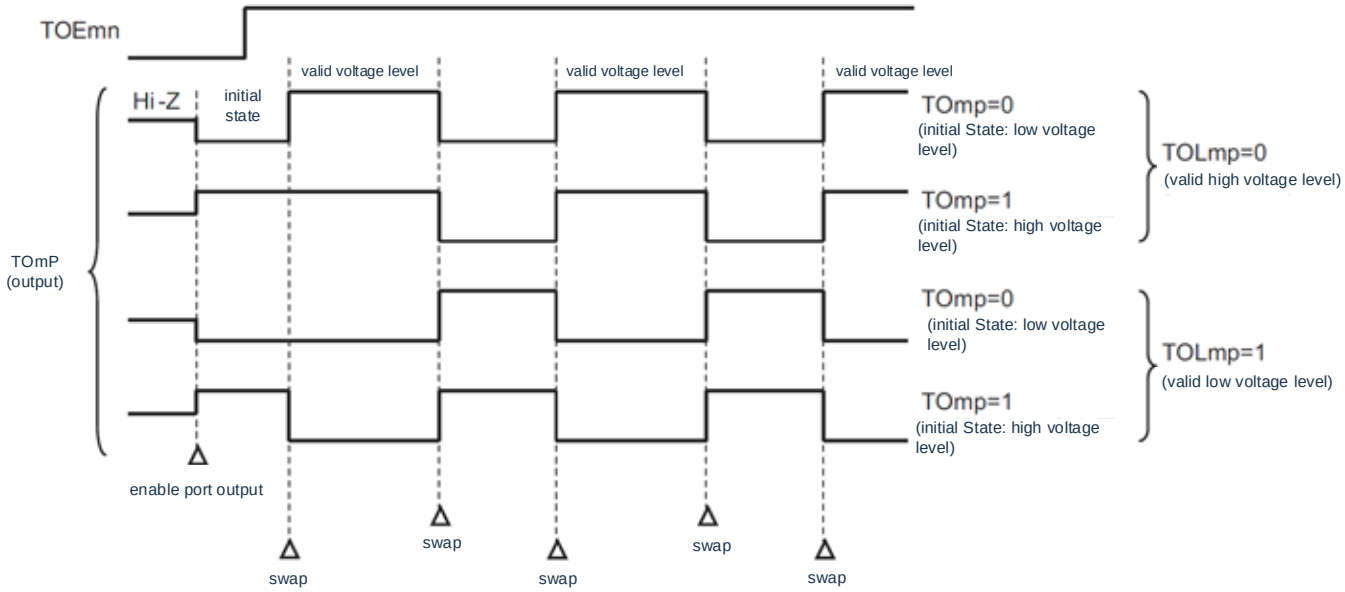
Note: 1. Alternation: Inverts the output state of the TOmn pin.

2.m: Cell number (m=0)n: Channel Number (n=0~3)

(b) The case (PWM output) of starting operation in the slave channel output mode (TOMmn=1)

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

Figure 6-34 TOmn pin output state at PWM output (TOMmn=1)



Note: 1. Set: The output signal from the TOmp pin changes from an invalid level to an active level.

Reset: The output signal from the TOmp pin changes from the active level to the invalid level.

2.m: Cell number (m=0)n: Channel number (p=1~3)

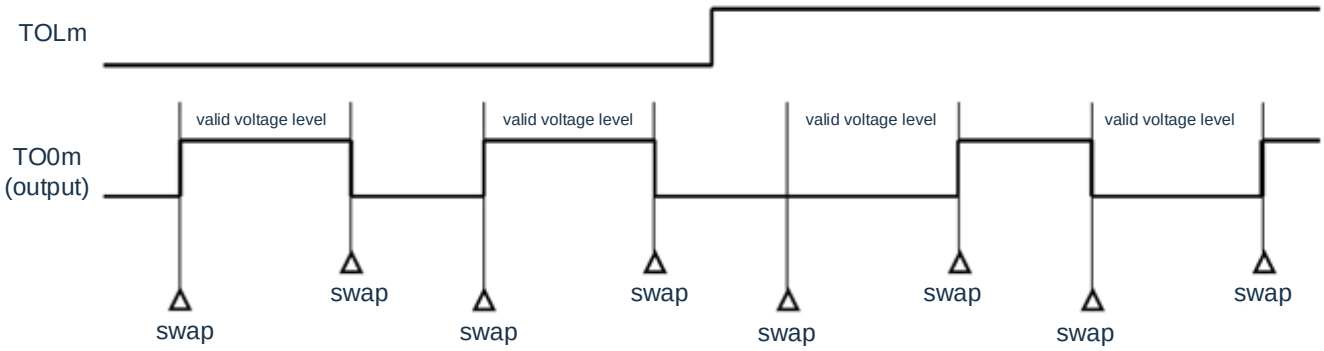
3) TOmn pin changes for dependent channel output mode (TOMmn=1)

(a) The case of changing the setting of the timer output level register m (TOLm) in the timer operation

If you change the setting of the TOLm register during the timer run, the setting is valid when the TOmn pin change condition occurs. The output level of the TOmn pin cannot be changed by rewriting the TOLm register.

When the TOMmn bit is "1", the run when the value of the TOLm register is changed during the timer run (TEmn=1) is as following.

Figure 6-35 Run when changing the contents of the TOLm register in a timer run



Note: 1. Set: The output signal from the TOmn pin changes from an invalid level to an active level.

Reset: The output signal from the TOmn pin changes from the active level to the invalid level.

2.m: Cell number (m=0)n: Channel Number (n=0~3)

(b) Set/reset sequence

In order to achieve 0% and 100% output at PWM output, through sliave channel, the timing of TOmn pin /TOmn bit of master channel timer interrupt (INTTMmn) is delayed by 1 counting clock cycle.

When the setting condition and the reset condition are generated at the same time, the reset condition is prioritized.

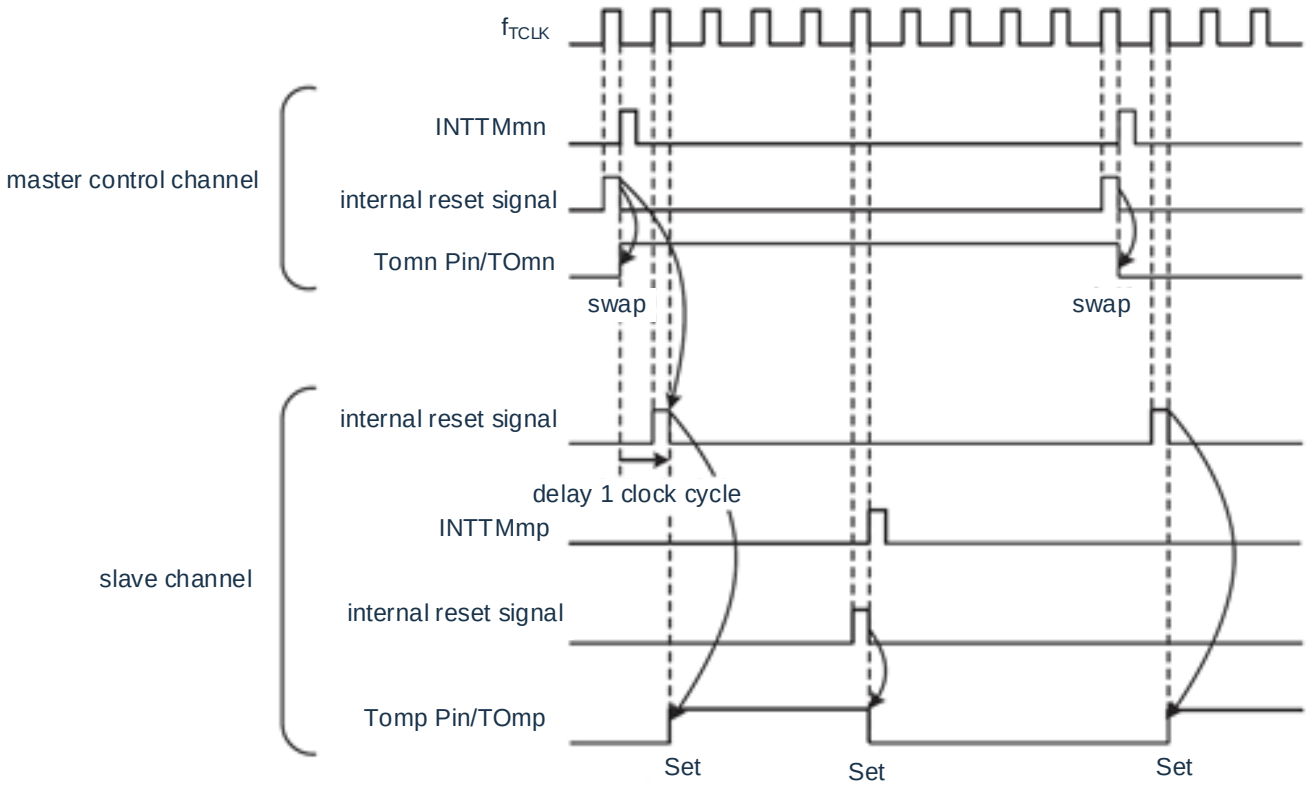
The set/reset operation state when the master/slave channel is set as shown in Figure 6-35 in this manner.

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

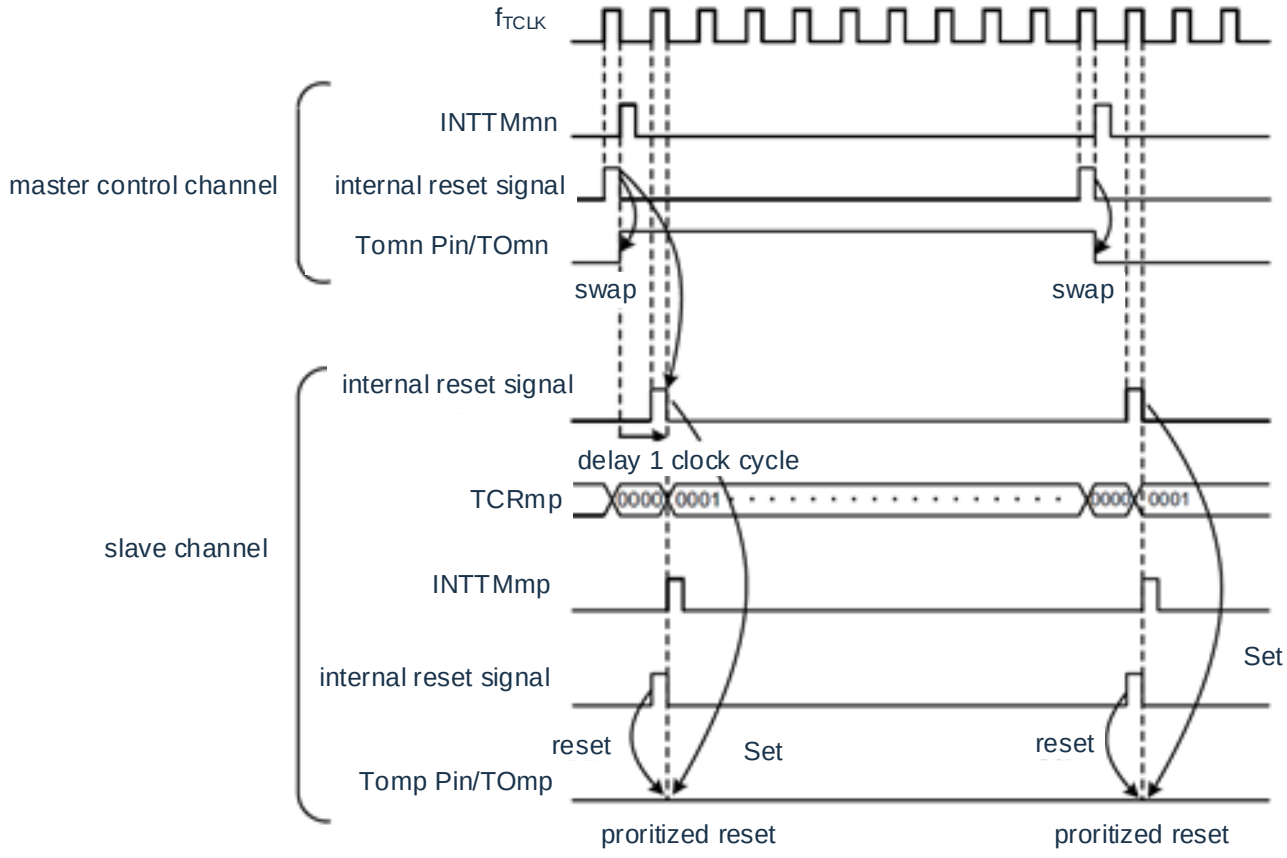
Dependent channels: TOEmp=1, TOMmp=1, TOLmp=0

Figure 6-36 Set/Reset Timing Operation State

(1) basic operation sequence



(2) Operation time sequence for 0% duty cycle



Note 1. Internal reset signal: TOmn pin reset/alternate signal

Internal Placement Signal: Placement Signal for TOmn Pins

2.m: Cell Number (m=0)

n: Channel Number n=0~3 (Master Channel: n=0,2)

p: Dependent Channel Number

n=0:p=1,2,3

n=2:p=3

6.6.4 One-time operation of TOmn bits

The timer output register m (TOm) has all channel setting bits (TOmn), so it can operate all channel TOmn bits.

Figure 6-37Example of one-time operation of TO0 n bits

Before

TO0	0	0	0	0	0	0	0	0	0	0	0	0	TO03	TO02	TO01	TO00
TOE0	0	0	0	0	0	0	0	0	0	0	0	0	TOE03	TOE02	TOE01	TOE00

Data to write

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

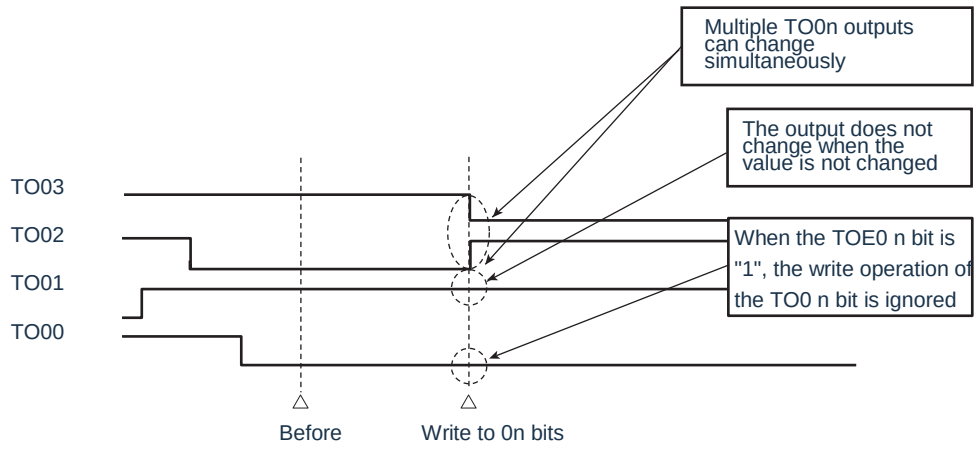
after writing

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

Only TOmn bits with TOEmn bit '0' can be written, ignoring TOmn bits with TOEmn bit '1'.

TOmn with TOEmn bit "1" is not affected by write operation, even TOmn bit is ignored.

Figure 6-38 TO0n pin status when TO0n bits are operated



Note: m: Cell number (m=0)n: Channel Number (n=0~3)

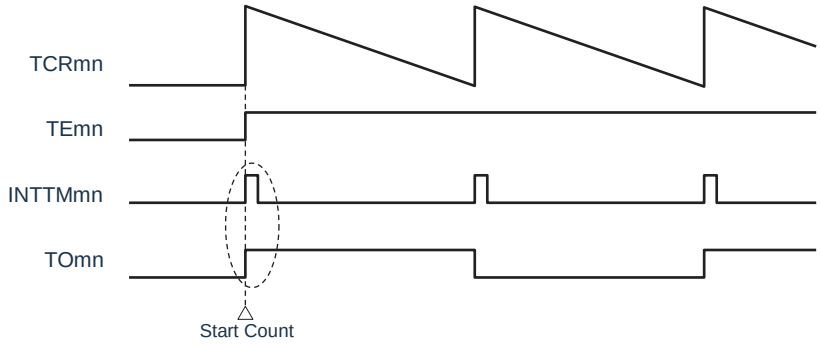
6.6.5 About timer interrupts and TOmn pin output at start of counting

In interval timer mode or capture mode, the MDmn0 bit of timer mode register mn (TMRmn) is set as the bit that generates timer interrupt when counting starts.

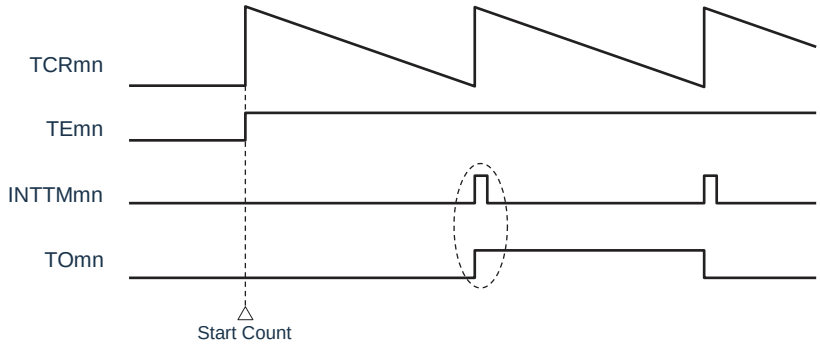
When the MDmn0 bit is '1', the start sequence of the count can be known by generating a timer interrupt (INTTMmn). In other mode, timer interrupt and TOmn output at that start of the count are not controlled. An example of a run when set to interval timer mode (TOEmn=1, TOMmn=0) is as follows.

Figure 6-39 Examples of timer interrupts and TOmn output at start of count

(a) Case where MDmn0 bit is "1"



(b) Case where MDmn0 bit is '0'



When the MDmn0 bit is "1", the output timer is interrupted (INTTMmn) at the start of the count and TOmn outputs alternately.

When the MDmn0 bit is '0', no timer interrupt (INTTMmn) is output and TOmn is unchanged at the start of the count, and INTTMmn is output alternately by TOmn.

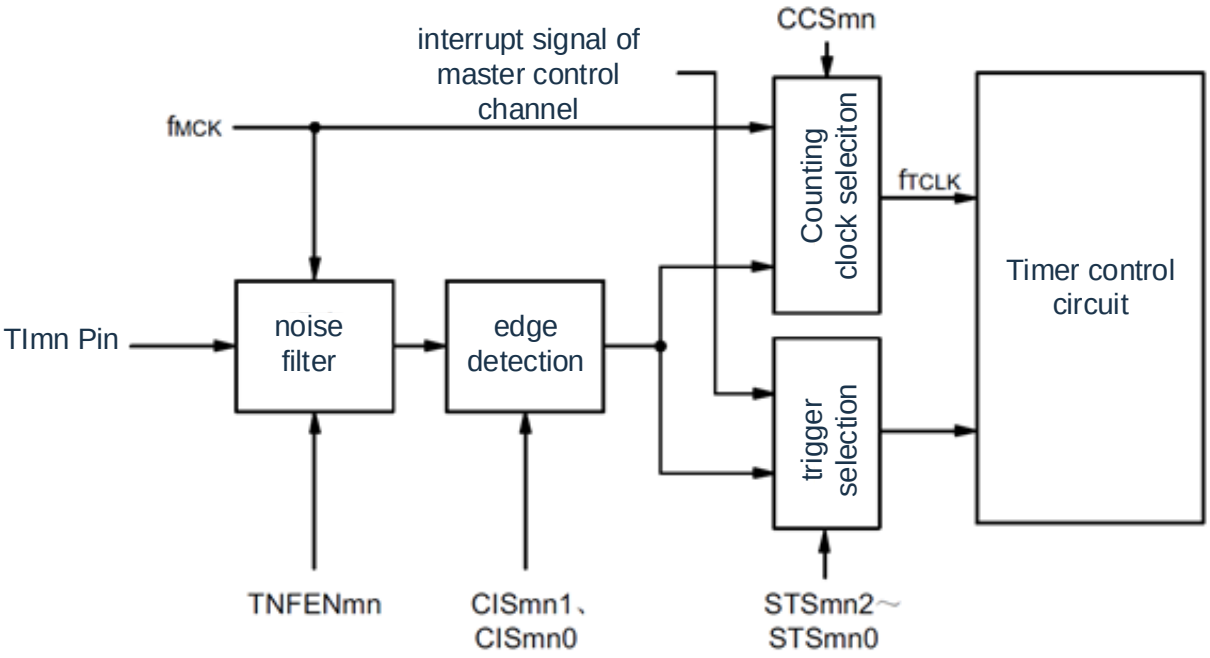
Note: m: Cell number (m=0)n: Channel Number (n=0~3)

6.7 Control of timer input (TImn)

6.7.1 Structure of TImn pin input circuit

The signal of the timer input pin is input to the timer control circuit through the noise filter and the edge detection circuit. For pins that need to be eliminated from noise, the corresponding pin noise filter must be set to be valid. The structure of the input circuit is as follows.

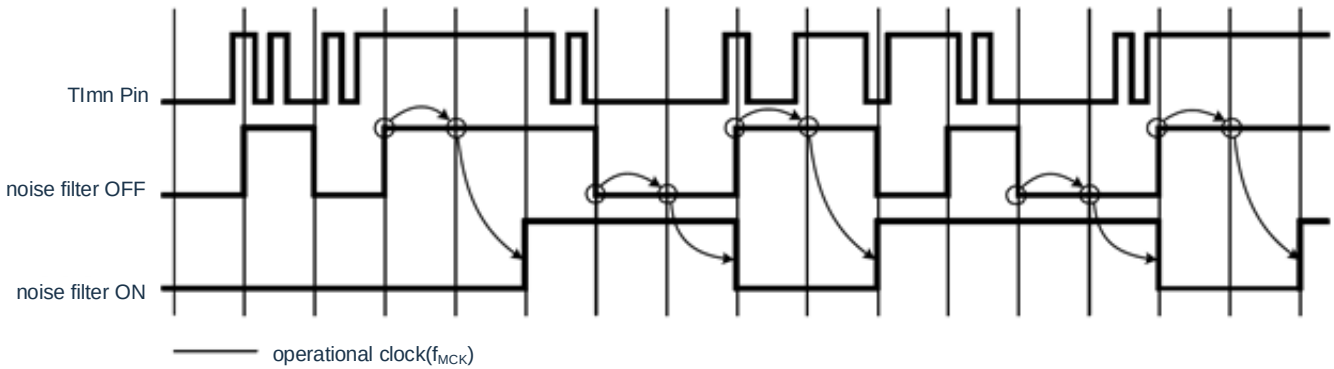
Figure 6-40(1) Structure of input circuit



6.7.2 noise filter

When the noise filter is inactive, synchronization is performed only through the runtime clock ($_{fMCK}$) of channel n. When the noise filter is active, two clocks are detected after synchronization through the $_{run}$-time clock (fMCK) of channel n. The TM4 lmn input pin in that case of a noise filter ON or OFF, the waveform aft passing through the noise filter circuit is shown below.

Figure. 6-40(2) Sampling waveform of the TImn input pin in the case of noise filter ON or OFF



Note: The input waveform of the TImn pin is used to describe the operation of the noise filter ON or OFF. In actual use, the input must be made in accordance with the TImn input high and low level width shown in AC Characteristics.

6.7.3 Precautions When Operating Channel Inputs

The noise filter circuit is not provided with a run-time clock when the timing input pin is not used. Therefore, the channel operation from the setting to use the timer input pin to the setting to input the timer corresponding to the pin allows triggering, requires the following waiting time.

(1) The case that the noise filter is OFF

If any of the timer mode register mn(TMRmn) bit12(CCSmn), bit9(STSmn1) and bit8(STSmn0) is set while all were '0', the operation enable of timer channel start register (TSm) must be triggered set after at least 2 execution clock (fMCK) cycles.

(2) The case that the noise filter is ON

If any of the timer mode register mn(TMRmn) bit12(CCSmn), bit9(STSmn1) and bit8(STSmn0) is set while all were '0', the operation enable of timer channel start register (TSm) must be triggered set after at least 4 execution clock (fMCK) cycles.

6.8 Independent channel operation function of universal timer unit

6.8.1 Operation as Interval Timer/Square wave Output

(1) interval timer

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

Generation period for INTTMmn (timer interrupt) = count clock period (set value of TDRmn+1)

(2) Operation as square wave output

TOmn generates INTTMmn and outputs 50% duty cycle square wave at the same time of alternate output.

The period and frequency of the TOmn output square wave can be calculated by the following formula:

The Square Wave Period Output by TOmn = Count Clock Period (TDRmn Set Value +1)×2

The Square Wave Frequency from TOmn = Count Clock Frequency / {(TDRmn's Set Value +1) ×2}

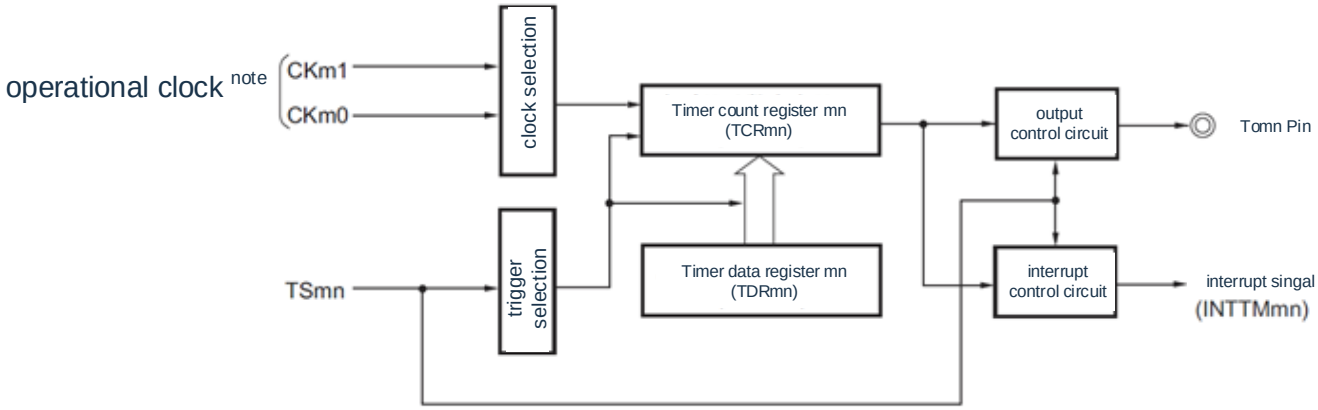
In interval timer mode, that timer count register mn (TCRmn) is use as the decrement counter.

After setting the channel start trigger bits (TSmn, TSHm1, TSHm3) of the timer channel start register m (TSm) to "1", the value of the timer data register mn (TDRmn) is loaded into the TCRmn register through one counting clock. At this time, if the MDmn0 bit of timer mode register n(TMRmn) is '0', the INTTMmn is not output and the TOmn is not output alternately. If the MDmn0 bit of the TMRmn register is "1", INTTMmn is output and TOmn is output alternately. The TCRmn register then decrements by a counting clock.

If TCRmn becomes '0000H', INTTMmn is output via the next count clock and TOmn is output alternately. At the same time, the value of the TDRmn register is loaded into the TCRmn register again. After that, the same operation continues.

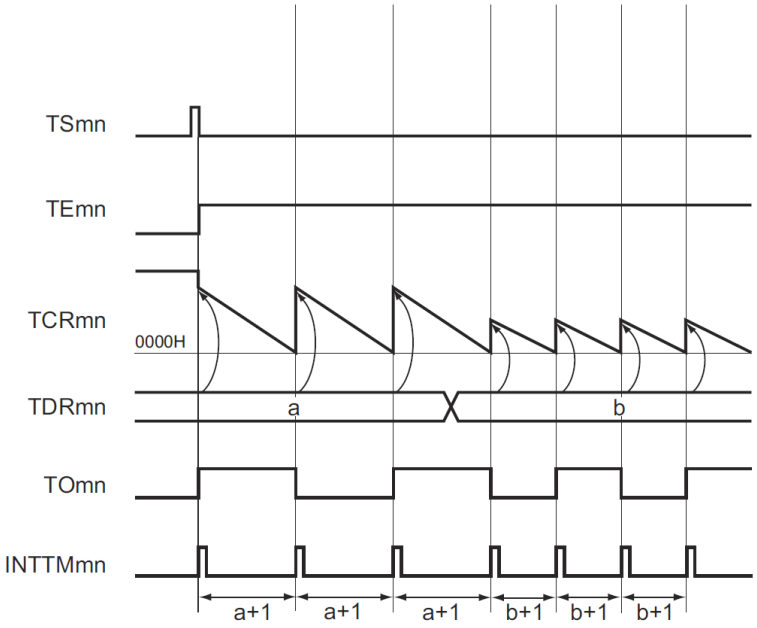
The TDRmn register can be overridden at any time, and the value of the overridden TDRmn register is valid from the next cycle.

Figure 6-41 Basic Timing Example as Interval Timer/Square Wave Output Operation (MDmn0=1)



Note: Clocks can be selected from CKm0, CKm1, CKm2, and CKm3 at Channels 1 and 3.

Figure 6-42 Basic Timing Example as Interval Timer/Square Wave Output Operation (MDmn0=1)



Remarks: 1.m: Cell number (m=0)n: Channel Number (n=0~3)

2. TSmn : bit n for timer channel start register m (TSm)

TEmn : The timer channel allows the bit n of the state register m (TEm)

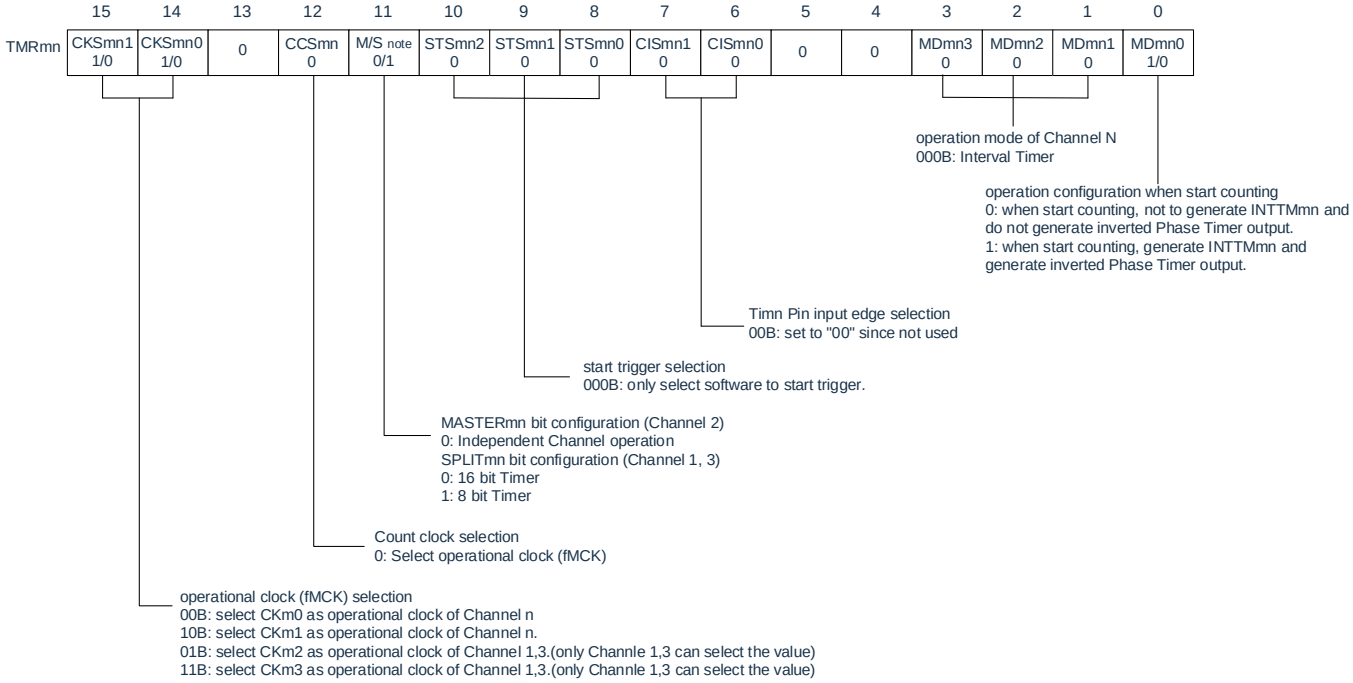
TCRmn: timer count register mn (TCRmn)

TDRmn: timer data register mn (TDRmn)

TOmn : TOmn Pin Output Signal

Figure 6-43 Example of Register Setting Content at Interval Timer/Square Wave Output

(a) timer mode register mn (TMRmn)



(b) Timer output register m (TOm).

TOm	bit	
	TOmn 1/0	0: Output "0" from TOmn. 1: Output "1" from TOmn.

(c) Timer output allow register m (TOEm).

TOEm	bit	
	TOEmn 1/0	0: Stop the TOmn output from the count run. 1: Allow TOmn output from the count run.

(d) Timer output level register m (TOLm).

TOLm	bit	
	TOLmn 0	0: "0" is set at TOMmn=0 (Master Channel Output Mode).

(e) Timer output mode register m (TOMm).

TOMm	bit	
	TOMmn 0	0: Sets the master channel output mode.

Note: TMRm2 : MASTERmn bit

TMRm1, TMRm3 :SPLITmn bit

TMRm0: Fixed as "0".

Note: m: Cell number (m=0)n: Channel Number (n=0 ~ 3)

Figure 6-44 Operation Steps of Interval Timer/Square Wave Output Function

	software operation	Hardware Status
TAU initial settings		The input clock of the timer unit m is in a stopped supply state. (stop providing clock, cannot write registers)
	Place the TM4 mEN location "1" of the peripheral enable register 0 (PER0). →	The input clock of the timer unit m is in a supplied state. (Start providing clock capable of writing registers)
	Set timer clock selection register m (TPSm). Determine the clock frequency for CKm0 to CKm3.	
channel initial setting	Set timer mode register mn (TMRmm) (determine the channel operation mode). The timer data register mn (TDRmn) is set with interval (period) value.	The channel is in an operational stop state. (Provide clocks, consume a portion of the Power)
	Using TOmn output: The TOMmn position of the timer output mode register m (TOMm) is "0" (main control channel output mode). Position TOLmn "0". → Set the TOmn bit to determine the initial level of the TOmn output. Position TOEmn "1" to allow TOmn output. → Set the port register and port mode register to "0". →	The TOmn pin is in the Hi-Z output state. When the port mode register is in output mode and the port register is "0", the initially set level of the TOmn is output. The TOmn is unchanged because the channel is in an operational stop state. The TOmn pin outputs the level set by the TOmn.
Start Run	(TOEmn position "1" only when using TOmn output and restarting) Position TSmn (TSHm1, TSHm3) "1". → Automatically returned to '0' because the TSmn (TSHm1, TSHm3) bit is the trigger bit.	The TEmn (TEHm1, THEm3) bit becomes "1" and starts counting. Load the value of the TDRmn register into the timer count register mn (TCRmn). When the MDmn 0 bit of the TMRmn register is '1', INTTMmn is generated and TOmn is output alternately.
Running	You can change the settings of the TDRmn register at will. Can read TCRmn register at any time. TSRmn register cannot be used. Can change the TOm register and TOEm register settings. Prevents the setting of the TMRmn register, TOMmn bit, and TOLmn bit from being changed.	The counter (TCRmn) counts down. If the count goes to "0000H", the value of the TDRmn register is loaded again into the TCRmn register and counting continues. When TCRmn is detected as "0000H", INTTMmn is generated and TOmn is output interleaved. This run is repeated thereafter.
Stop Running	Position TTmn (TTHm1, TTHm3) "1". → Automatically returned to '0' because the TTmn (TTHm1, TTHm3) bit is the trigger bit.	The TEmn (TEHm1, TEHmn) bit changes to "0" and stops counting. The TCRmn register maintains count values and stops counting. The TOmn output is not initialized and remains in state.
	Set the TOEmn location "0" and the TOmn bit. →	The TOmn pin outputs the level set by the TOmn bit.
TAU Stop	To maintain the TOmn pin output level: Position TOmn "0" after setting the value to be maintained for the port register. → The TOmn pin output level does not need to be maintained: No settings are required.	Maintain the output level of the TOmn pin through port functionality.
	Position the TM4 mEN of the PER0 register "0".	The input clock of the timer unit m is in a stopped supply state. Initialize the SFRs of all circuits and channels. (TOmn bit becomes "0" and TOmn pin becomes port function)

Restart Operation

6.8.2 Run as External Event Counter

Can be used as an event counter to count the valid edges (external events) of detected TImn pin inputs, and interrupt occurs if a specified count value is reached. The specified counter value can be calculated using the following formulas:

Specified count value=TDRmn's set value+1

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

The value of timer data register mn(TDRmn)is loaded into the TCRmn register by setting "1" for any channel start trigger bits (TSmn, TSHm1, TSHm3) of timer channel start registerm (TSm).

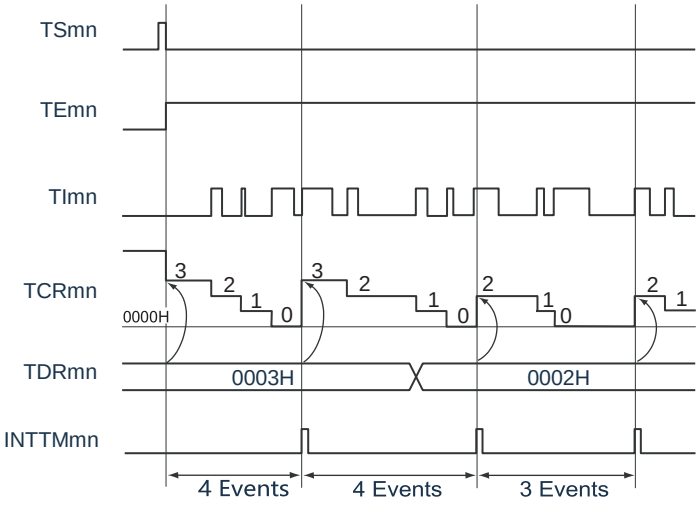
The TCRmn register decrements while detecting a valid edge of the TImn pin input. If the TCRmn becomes "0000H," the value of the TDRmn register is loaded again and the INTTMmn is output.

After that, the same operation continues.

Because the TOmn pin outputs irregular waveforms based on external events, the timer must output the TOEmn position "0" of the allowed register m (TOEm).

Can override the TDRmn register at any time, the value of the overridden TDRmn register is valid for the next count period.

Figure 6-45 A basic timing example of an external event counter running



Remarks: 1.m: Cell number (m=0)n: Channel Number (n=0~3)

2. TSmn : bit n for timer channel start register m (TSm)

TEmn : The timer channel allows the bit n of the state register m (TEm)

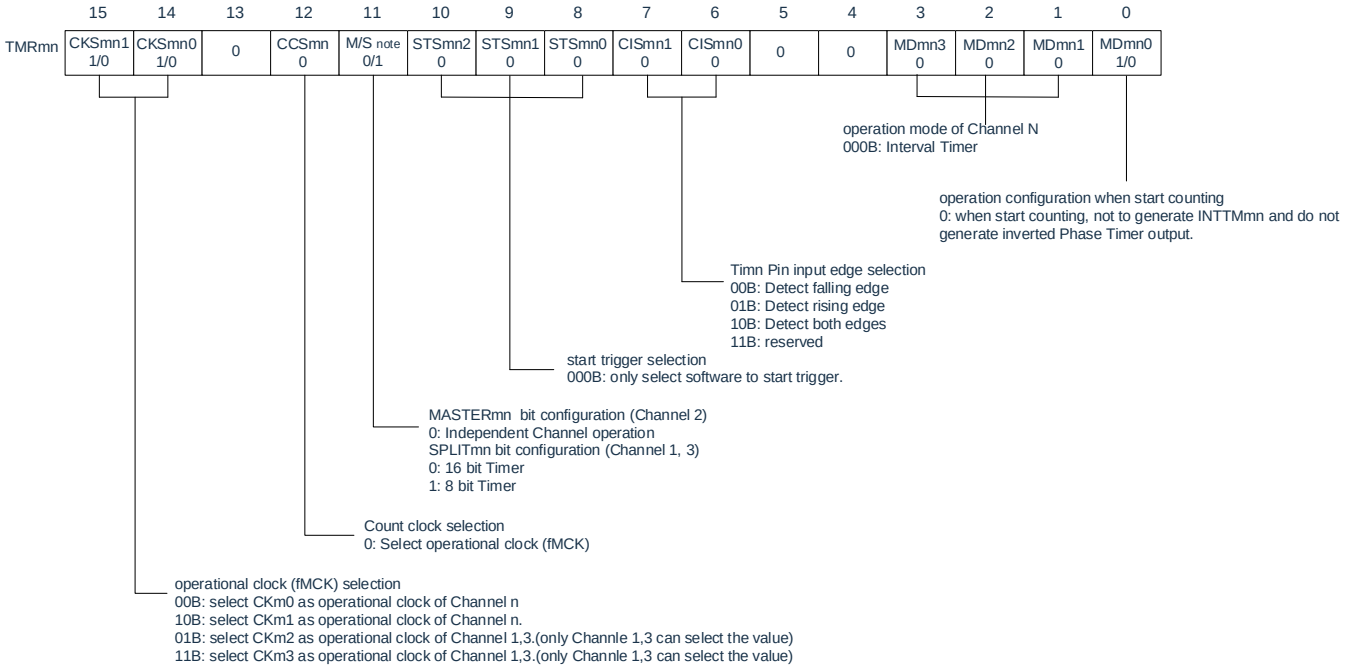
TImn :TImn Pin Input Signal

TCRmn: timer count register mn (TCRmn)

TDRmn: timer data register mn (TDRmn)

Figure 6-46 Example of register setting content in external event counter mode

(a) timer mode register mn (TMRmn)



(b) Timer output register m (TOm).

	bit	
TOm	TOmn 1/0	0: Output "0" from TOmn.

(c) Timer output allow register m (TOEm).

	bit	
TOEm	TOEmn 1/0	0: Stop the TOmn output from the count run. 1: Allow TOmn output by count run.

(d) Timer output level register m (TOLm).

	bit	
TOLm	TOLmn 0	0: "0" at TOMmn=0 (Master Channel Output Mode).

(e) Timer output mode register m (TOMm)m.

	bit	
TOMm	TOMmn 0	0: Sets the master channel output mode.

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

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

Figure 6-47 Action Steps for External Event Counter Functionality

	software operation	Hardware Status
Timer4 Initial Settings		The input clock of the timer unit m is in a state where supply is stopped. (stop providing clock, cannot write registers)
	Place the TM4 mEN location "1" of the peripheral enable register 0 (PER0). →	The input clock of the timer unit m is in a supplied state, and each channel is in an operation stop state. (Start providing clock capable of writing registers)
	A clock selection register m (TPSm) that sets the timer. Determine the clock frequency for CKm0 to CKm3.	
channel initial setting	Allow the noise filter to correspond to register 1 (NFEN12) either "OFF" or "1" (ON). A timer mode register mn (TMRmn) is set. A timer data register mn (TDRmn) is set with a count value. Output timer to allow TOEmn location "0" for register m (TOEm).	The channel is in an operational stop state. (Provide clocks, consume a portion of the Power)
Start Run	Position TSmn "1". → The TSmn bit is a trigger bit and is automatically returned 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 enters the detection waiting state of the TImn pin input edge.
Running	You can change the settings of the TDRmn register at will. Can read TCRmn register at any time. The TSRmn register is not used. Prevents the setting of TMRmn registers, TOMmn bits, TOLmn bits, TOmn bits, and TOEmn bits from being changed.	The counter (TCRmn) counts down each time an input edge of the TImn pin is detected, and if the count reaches '0 000H', loads the value of the TDRmn register again into the TCRmn register and continues counting. A INTTMmn is generated when TCRmn is detected as '0000H'. This run is repeated thereafter.
Stop Running	Position TTmn "1". → The TTmn bit is a trigger bit and is automatically returned to "0".	The TEmn bit changes to "0" and stops counting. The TCRmn register maintains count values and stops counting.
Timer4 Stop	Position the TM4 mEN of the PER0 register "0". →	The input clock of the timer unit m is in a stopped supply state. Initialize the SFRs of all circuits and channels.

Restart Operation

6.8.3 Operation as a frequency divider (only for channel 0 of cell 0)

A frequency divider capable of dividing the clock input by the TI00 pin and used as the output of the TO00 pin.

The frequency division clock frequency of TO00 output can be calculated by the following formula:

· Select a rising or falling edge:

Split Clock Frequency = Input Clock Frequency / {(TDR00's Set Value +1)×2}

· Select the case along the bilateral edge:

Split Clock Frequency ≈ Input Clock Frequency / (TDR00's setting +1)

In interval timer mode, timer count register 00 (TCR00) is used as the diminishing 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 TCR00 register by detecting an effective edge of TI00. If the MD000 bit of the timer mode register 00 (TMR00) is "0", no output of INTTM00 and TO00 is not alternately output; If the TMR00 register has a MD000 bit of "1", INTTM00 is output and TO00 is output alternately.

The TCR00 register then decrements through valid edges entered by the TI00 pin. If TCR00 becomes "0000H," TO00 will alternate output. At the same time, the value of the TDR00 register is loaded into the TCR00 register and continues to count.

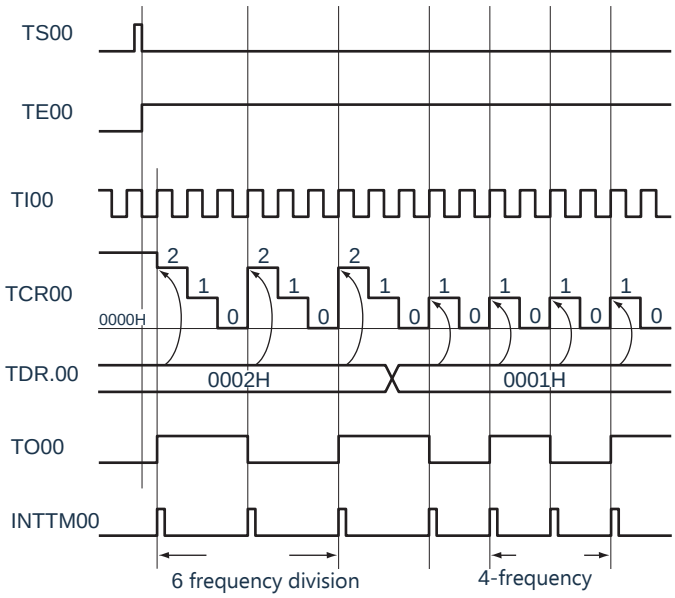
If the TI00 pin is selected to input the double edge detection, the duty cycle error of the input clock will affect the frequency division clock cycle of the TO00 output.

The clock cycle output by TO00 contains a sampling error of one running clock cycle.

Clock cycle for TO00 output=Ideal TO00 output clock cycle ±Running clock cycle (error)

The TDR00 register can be overridden at any time, and the value of the overridden TDR00 register is valid during the next count.

Figure 6-48 As a Basic Timing Example of Frequency Divider Operation (MD000=1)



Remark TS00 : Bit 0 of timer channel start register 0 (TS0)

TE00 : Timer channel allows bit 0 of status register 0 (TE0)

TI00 :TI00 pin input signal

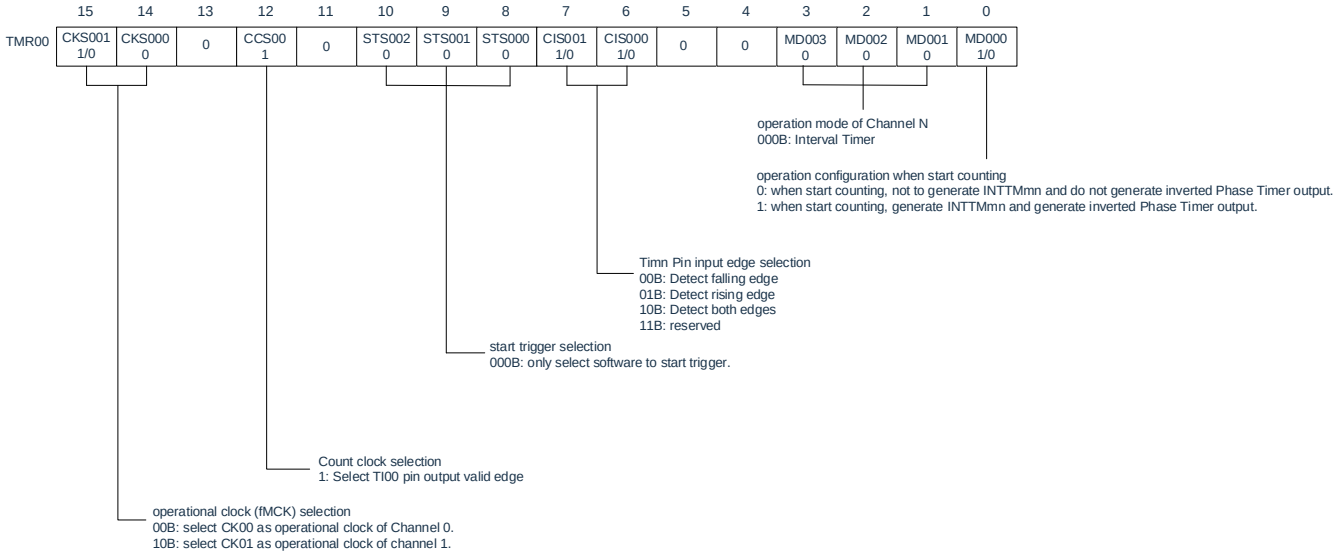
TCR00: Timer count register 00 (TCR00).

TDR.00: timer data register 00 (TDR00).

TO00 : TO00 pin output signal

Figure 6-49 as an example of register setting content when the frequency divider is running

(a) Timer Mode Register 00 (TMR00).



(b) Timer output register 0 (TO0)

	bit 0
TO0	TO00 1/0

0: Output "0" by TO00.
1: Output "1" from TO00.

(c) Timer output allow register 0 (TOE0)

	bit 0
TOE0	TOE00 1/0

0: Stop the TO00 output from the count run.
1: Allow TO00 output by count runs.

(d) Timer output level register 0 (TOL0)

	bit 0
TOL0	TOL00 0

0: Place "0" in the master channel output mode (TOM00=0).

(e) Timer Output Mode Register 0 (TOM0)

	bit
TOM0	TOM00 0

0: Sets the master channel output mode.

Figure 6-50 Operating steps when frequency divider functions

	software operation	hardware state
Timer 4 initial configuration		Timer Unit 0 input clock is in stopped state (stop providing clock, not able to write into registers)
	set TM4mEN bit of peripheral enable register 0 (PER0) to '1' →	Timer Unit 0 input clock is in active state (start providing clock, able to write into registers)
	configure Timer clock selection register 0 (TPS0), confirm CK00~CK03 clock frequency	
Channel Initial configuration	set corresponding bit of noise filter enable register 1 (NFEN1) to '0' (OFF) or '1' (ON). Configure Timer mode register 00 (TMR00) (confirm channel operation mode, select edge detection). Configure interval(period) valulue of Timer data register 00 (TDR00)	channel in operation stopped state (providing clock, consume portion of power)
	set TOM00 bit of timer output mode register 0 (TOM0) to '0' (master control channel output mode). Set TOL00 bit to '0' configure TO00 bit and confirm TO00 output initial voltage value. Set TOE00 bit to '1', aloow TO00 output. Set port register and port mode register to '0'.	TO00 pin in Hi-Z output state. When port mode register set to output mode and port register as '0', output TO00 initial configured voltage level. Because channel is in operation stopped state, thus TO00 remains unchange. TO00 pin output TO00 configured voltage level.
Start operation	set TOE00 bit to '1' (only limited to restart operation). Set TS00 bit to '1'. Because TS00 bit is trigger bit, thus automatically return to '0'.	TE00 bit turns to '1' and start counting. Load TDR00 register value into Timer count register 00 (TCR00). When MD000 bit of TMR00 register turns into '1', generate INTTM00 and TO00 swaps output
in operation	can modify any TDR00 register configuration value. Can read TCR00 register anytime. Do not use TSR00 register. Can modify TO0 register and TOE0 register value. Forbidden modifying TMR00 register. TOM00 bit and TOL00 bit configuration value.	Counter (TCR00) performs decremental counting. When count reaches '0000H', then load TDR00 register value into TCR00 register again and continue counting. When detecting TCR00 as '0000H', generate INTTM00 and TO00 swaps output. Thereafter, repeat the operation.
stop operation	set TT00 bit to '1'. Because TT00 bit is trigger bit, thus automtically return to '0'.	TE00 bit turns to '0' and stop counting. TCR00 register remains counted value and stop counting. TO00 output not been initialized and remain same state. TO00 pin outputs TO00 configured voltage.
	set TOE00 bit to '0' and configure value for TO00 bit	TO00 pin output TO00 configured voltage level.
Timer 4 stop	Scenarios to maintain TO00 pin output voltage: set TO00 bit to '0' after set hold value to port register configuration. In case TO00 pin output voltage does not need to be held: no configuration requried	hold TO00 pin output voltage level via port function.
	set TM4mEN bit of peripheral enable register 0 (PER0) to '0'	Timer Unit 0 input clock is in stopped state. Perform initialization to all circuit and SFR of all channels. (TO00 bit turns into '0' and TO00 pin becomes port function)

restart operation

6.8.4 Operation as an input pulse interval measurement

It is possible to capture count value at that effective edge of TImn and measure the interval of the TImn input pulse. During the TEmn bit "1", the software operation (TSmn=1) can also be set to capture the trigger to capture count values.

The pulse interval can be calculated using the following equation:

TImn Input Pulse Interval = Period of the counting clock ((10000H TSRmn:OVF) + (catch value of TDRmn+1))

Note: Since the TImn pin input is sampled by a running clock selected by the CKSmn bit of the timer mode register mn (TMRmn), an error occurs.

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

If the channel start trigger bit (TSmn) of the timer channel start register m (TSm) is set as '1', the TCRmn register starts counting from '0000H' by counting clock.

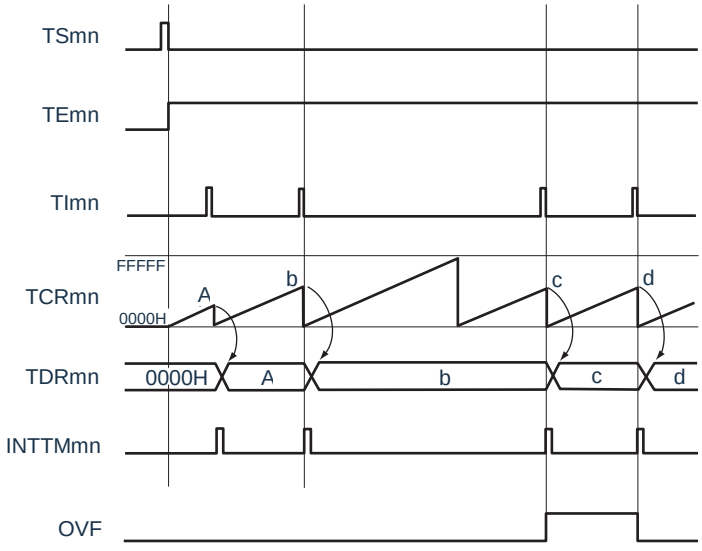
If the valid edge of the TImn pin input is detected, the count value of the TCRmn register is transferred (TDRmn) to the timer data register, TCRmn. At this time, if the counter overflows, the OVF position "1" of the timer status register mn (TSRmn). If the counter does not overflow, clear the OVF bit. After that, the same operation continues.

When the count value is captured to the TDRmn register, the OVF bit of the TSRmn register is updated according to whether overflows occur during measurement.

Even if the counter performs a full count of 2 cycles or more, it is considered that an overrun occurs to put the OVF position of the TSRmn register. However, the interval value cannot be measured normally through the OVF bit when an overflow occurs 2 or more times.

The STSmn2~STSmn0 position of the TMRmn register is "001B", and the effective edge of the TImn is used for starting trigger and capture trigger.

Figure 6-51 As an example of the running basic timing of an input pulse interval measurement (MDmn0=0)



Remarks: 1.m: Cell number (m=0)n: Channel Number (n=0~3)

2. TSmn : bit n for timer channel start register m (TSm)

TEmn : The timer channel allows the bit n of the state register m (TEm)

TImn :TImn Pin Input Signal

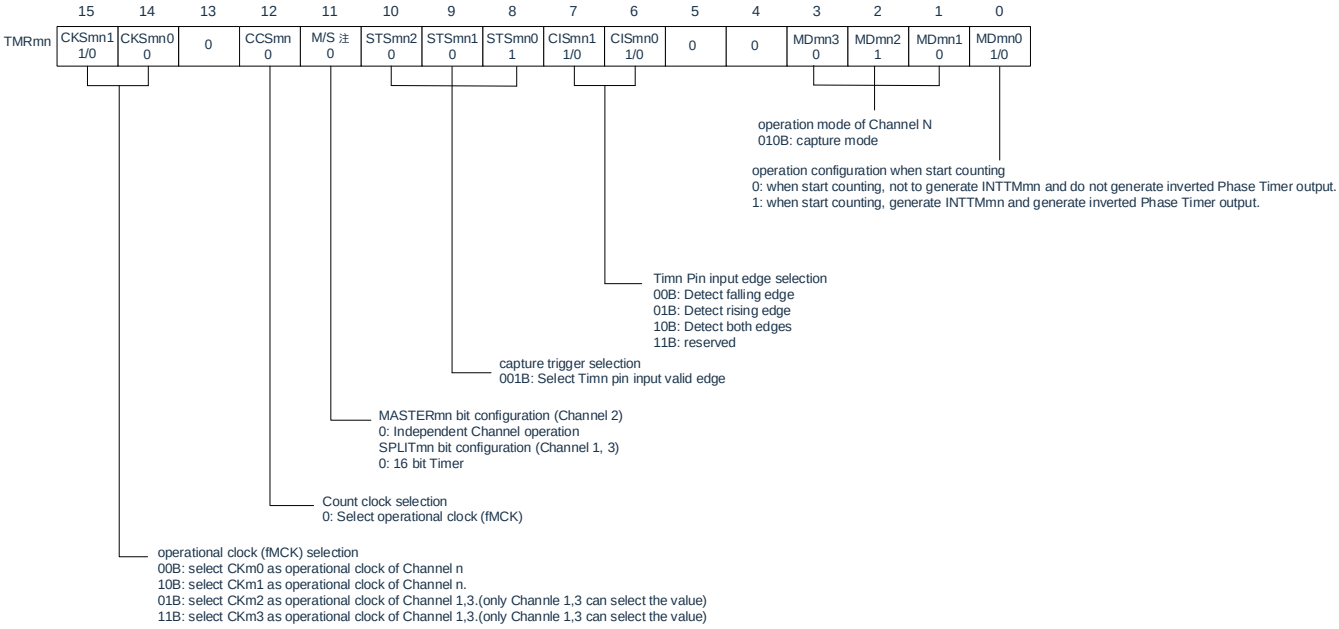
TCRmn : timer count register mn (TCRmn)

TDRmn : timer data register mn (TDRmn)

OVF : bit 0 of timer state register mn (TSRmn).

Figure 6-52 Example of Register Setting Content When Measuring Input Pulse Interval

(a) timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	bit	
TOm	TOmn	0: Output "0" from TOmn.
	0	

(c) timer output allow register m (TOEm)

	bit	
TOEm	TOEmn	0: Stop the TOmn output from the count run.
	0	

(d) Timer output level register m (TOLm)

	bit	
TOLm	TOLmn	0: Place "0" in the main channel output mode (TOMmn=0).
	0	

(e) Timer output mode register m (TOMm)

	bit	
TOMm	TOMmn	0: Sets the master channel output mode.
	0	

Note: TMRm2:MASTERmn bit

TMRm1, TMRm3:SPLITmn bit

TMRm0: Fixed as "0".

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

Figure 6-53 Operating steps when inputting pulse interval measurement functions

	software operation	hardware state
Timer 4 initial configuration		Timer Unit m input clock is in stopped state (stop providing clock, not able to write into registers)
	set TM4mEN bit of peripheral enable register 0 (PER0) to '1' →	Timer Unit m input clock is in active state, all channels in operation stopped state.
	configure Timer clock selection register m(TPSm), confirm CKm0~CKm3 clock frequency	
Channel Initial configuration	set corresponding bit of noise filter enable register 1 (NFEN1) to '0' (OFF) or '1' (ON). Configure Timer mode register mn (TMRmn) (confirm channel operation mode).	channel in operation stopped state (providing clock, consume portion of power)
Start operation	set TSmn bit to '1'. → Because TSmn bit is trigger bit, thus automatically return to '0'.	TEmn bit turns into '1' and start counting. Clear Timer counting register (TCRn) to "0000H". When MDmn0 bit of TMRmn register is '1', generate INTTMmn.
in operation	can only modify configure value of CISmn1 bit and CISmn0 bit of TMRmn register. Can read TDRmn register anytime. Can read TCRmn register aanytime. Can read TSRmn register anytime. Forbidden modifying TOMmn bit, TOLmn bit, TOmn bit and TOEmn bit configuration.	Counter(TCRmn) start incremental counting from "0000H", if detecting TImn pin input valid edge or TSmn bit set to '1', then transfer (capture) counting value to Timer data register mn(TDRmn), at the same time, clear TCRmn to "0000H" and generate INTTmn. At this time, if overflow occurs, then set OVF bit of Timer status register mn(TSRmn) . If overflow does not occur, then clear OVF bit. Thereafter, repeat the process.
stop operation	set TTmn bit to '1'. → Because TTmn bit is trigger bit, thus automatically return to '0'.	TEmn bit turns into '0' and stop counting. TCRmn register hold counted value and stop counting. 0VF bit of TSRmn register remains unchange.
timer 4 stop	set TM4mEN bit of peripheral enable register 0 (PER0) to '1' →	Timer Unit m input clock is not been provided.Perform initialization to all circuit and SFR of all channels. (TO00 bit turns into '0' and TO00 pin becomes port function)

restart operation

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

6.8.5 Operation as voltage high and low level width measurement of input signal

Note: When used as LIN-bus support, the bit1 (ISC1) of the input switch control register (ISC) must be set to "1", and use RxD0 instead of TImn.

The signal width (high and low level width) of the TImn can be measured by starting the count at one edge input by the TImn pin and capturing the count at another edge. The signal width of TImn can be calculated using the following formula.

Signal width entered by TImn = Cycle count clock ((10000H×HTSRmn:OVF) + (catch value of TDRmn+1))

Note: Since the TImn pin input is sampled by a running clock selected by the CKSmn bit of the timer mode register mn (TMRmn), an error occurs.

In capture&single count mode, the timer count register mn (TCRmn) is used as the increment counter. If the channel start trigger bit (TSmn) of the timer channel start register m (TSm) is set as "1", and the start edge of the TImn pin is detected waiting state.

If the start edge of the TImn pin input is detected (the rising edge of the TImn pin input when measuring the high level width), count incrementally starts from 0000H. Then, if an effective capture edge is detected (the falling edge of the TImn pin input when measuring the high level width), INTTMmn is output simultaneously. At this time, if the counter overflows, the OVF position of the timer state register mn (TSRmn) is set. If the counter does not overflow, clear the OVF bit. The TCRmn register stops counting by changing its value to 'Pass to TDRmn register value +1' and enters the TImn pin start edge detection wait state. After that, the same operation continues.

When the count value is captured to the TDRmn register, the OVF bit of the TSRmn register is updated according to whether overflows occur during measurement.

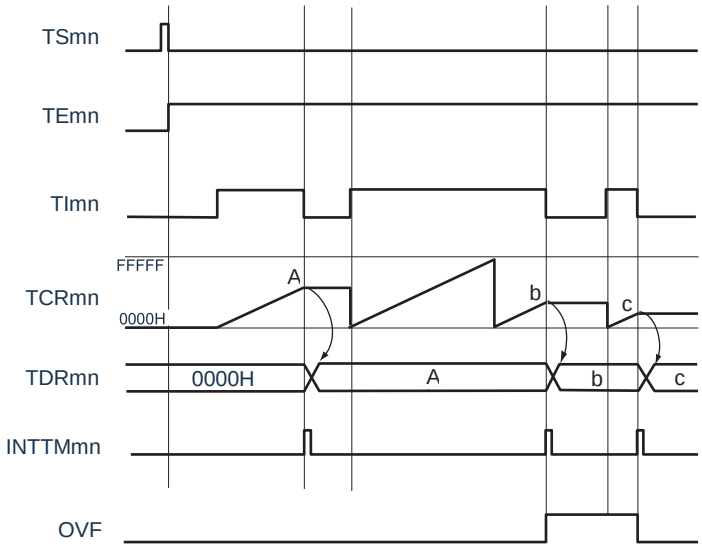
Even if the counter performs a full count of 2 cycles or more, it is considered that an overrun occurs to put the OVF position of the TSRmn register. However, the interval value cannot be measured normally through the OVF bit when an overflow occurs 2 or more times.

It is possible to set whether a high level width or a low level width of a TImn pin is measured by a CISmn1 bit and a CISmn0 bit of a TMRmn register. This feature is designed to measure the input signal width of the TImn pin, so the TSmn position "1" cannot be used during TEmn.

CISmn1, CISmn0=10B of TMRmn register: A low level width is measured.

CISmn1, CISmn0=11B of TMRmn register: A high level width is measured.

Figure 6-54 Example of a running base time sequence for a high and low level width measurement of an input signal



Remarks: 1.m: Cell number (m=0)n: Channel Number (n=0~3)

2. TSmn : Timer channel start register m (TSm) bit n

TEmn: Timer channel allows bit n of status register m (TEm).

TImn:TImn Pin Input Signal

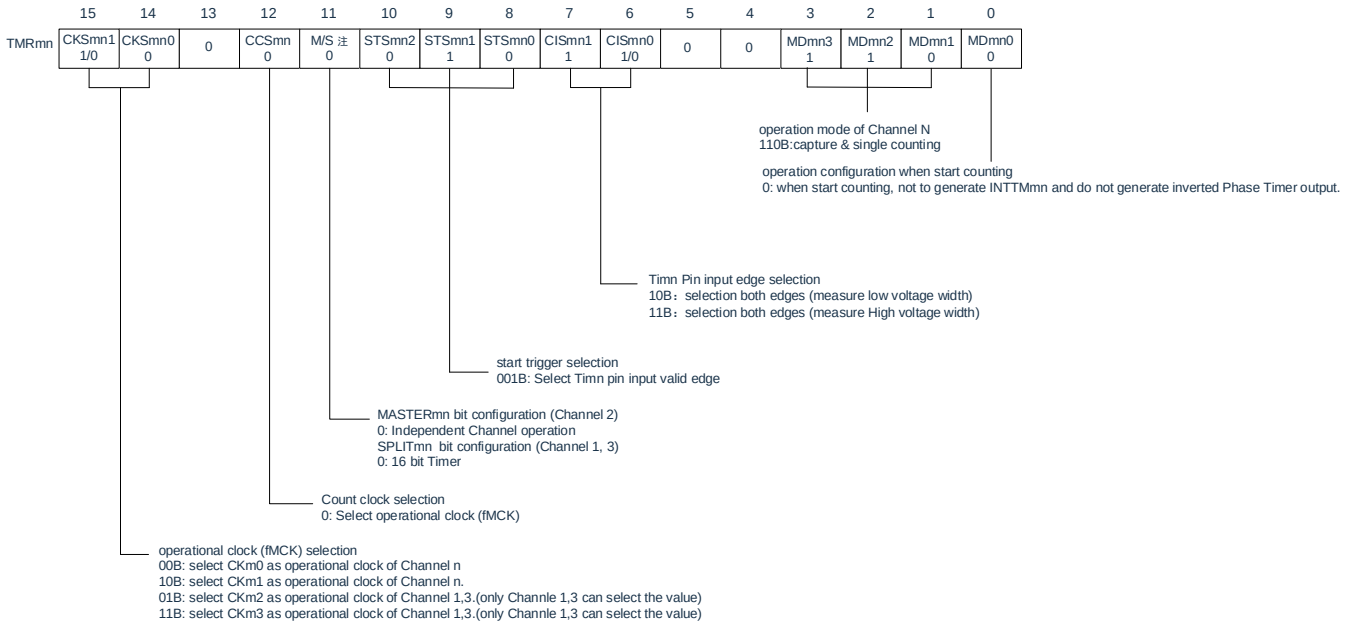
TCRmn: timer count register mn (TCRmn)

TDRmn: timer data register mn (TDRmn)

OVF: bit0 for timer state register mn(TSRmn)

Figure 6-55 Example of register setting content when measuring high and low level width of input signal

(a) timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	bit	
TOm	TOmn	0: Output "0" from TOmn.
	0	

(c) timer output allow register m (TOEm)

	bit	
TOEm	TOEmn	0: Stop the TOmn output from the count run.
	0	

(d) Timer output level register m (TOLm).

	bit	
TOLm	TOLmn	0: Place "0" in the master channel output mode (TOMmn=0).
	0	

(e) Timer output mode register m (TOMm)m.

	bit	
TOMm	TOMmn	0: Sets the master channel output mode.
	0	

Note: TMRm2:MASTERmn bit

TMRm1, TMRm3:SPLITmn bit

TMRm0: Fixed as "0".

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

Figure 6-56 Operation step for measuring width of high and low level of input signal

	software operation	hardware state
Timer 4 initial configuration		Timer Unit 0 input clock is in stopped state (stop providing clock, not able to write into registers)
	set TM4mEN bit of peripheral enable register 0 (PER0) to '1' →	Timer Unit m input clock is in active state, all channels in operation stopped state.
	configure Timer clock selection register m(TPSm), confirm CKm0~CKm3 clock frequency	
Channel Initial configuration	set corresponding bit of noise filter enable register 1 (NFEN1) to '0' (OFF) or '1' (ON). Configure Timer mode register mn (TMRmn) (confirm channel operation mode). Set T0Emn bit to '0', and stop T0mn operation.	channel in operation stopped state (providing clock, consume portion of power)
Start operation	set TSmn bit to '1'. → Because TSmn bit is trigger bit, thus automatically return to '0'.	TEmn bit turns into '1' and enter into start trigger (detect Timn pin input valid edge or set TSmn bit to '1') detection waiting state.
	detect TImn pin input counting start edge →	clear timer counting register mn (TCRmn) to '0000H" and start decremental counting.
in operation	can modify any TDRmn register configuration value. Can read TCRmn register anytime. Do not use TSRmn register. Forbidden modifying TMRmn register, TOMmn bit and TOLmn bit,Tomn and T0Emn bit configuration value.	while detecting TImn pin start edge, Counter(TCRmn) start incremental counting from "0000H", if detecting TImn pin input capture edge, then transfer counting value to Timer data register mn(TDRmn) and generate INTTmn. At this time, if overflow occurs, then set OVF bit of Timer status register mn(TSRmn) . If overflow does not occur, then clear OVF bit. TCRmn register stop counting before detecting next TImn pin start edge. Thereafter, repeat the process.
stop operation	set TTmn bit to '1'. → Because TTmn bit is trigger bit, thus automatically return to '0'.	TEmn bit turns into '0' and stop counting. TCRmn register hold counted value and stop counting. 0VF bit of TSRmn register remains unchange.
timer 4 stop	set TM4mEN bit of peripheral enable register 0 (PER0) to '1' →	Timer Unit m input clock is not been provided.Perform initialization to all circuit and SFR of all channels. (TO00 bit turns into '0' and TO00 pin becomes port function)

restart operation

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

6.8.6 Operation as delay counter

Can start decrement counts with valid edge detection (external events) entered through the TImn pin, and generate INTTMmn at arbitrary set-up intervals

(Timer interrupt).

During the TEmn bit of '1', the TSmn position of '1' can be counted down by software and INTTMmn (timer interrupt) can be generated.

Interrupt generation cycles can be calculated using the following equation:

Generation period of INTTMmn (timer interrupt) = period of the counting clock (set value of TDRmn+1)

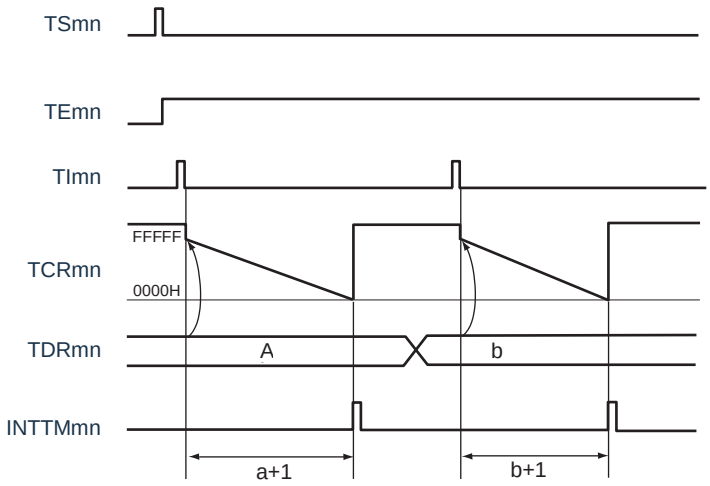
In a single count mode, the timer count register mn (TCRmn) is used as the decremental counter.

if that channel start trigger bit (TSmn, TSHm1, TSHm3) of the timer channel start register m (TSm) is set as" 1",TEmn

Bit, TEHm1 bit, and TEHm3 bit become'1' and enter the valid edge detection wait state of TImn pin. The TImn register is started by valid edge detection of TCRmn pin input and the value of timer data register mn(TDRmn) is loaded. The TCRmn register counts down from the value of the mounted TDRmn register by counting the clock. If TCRmn becomes "0000H," INTTMmn is output and count is stopped before a valid edge of the next TImn pin input is detected.

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

Figure 6-57: As an example of the basic timing of operation of a delay counter



Remarks: 1.m: Cell number (m=0)n: Channel Number (n=0~3)

2. TSmn : bit n for timer channel start register m (TSm)

TEmn : The timer channel allows the bit n of the state register m (TEm)

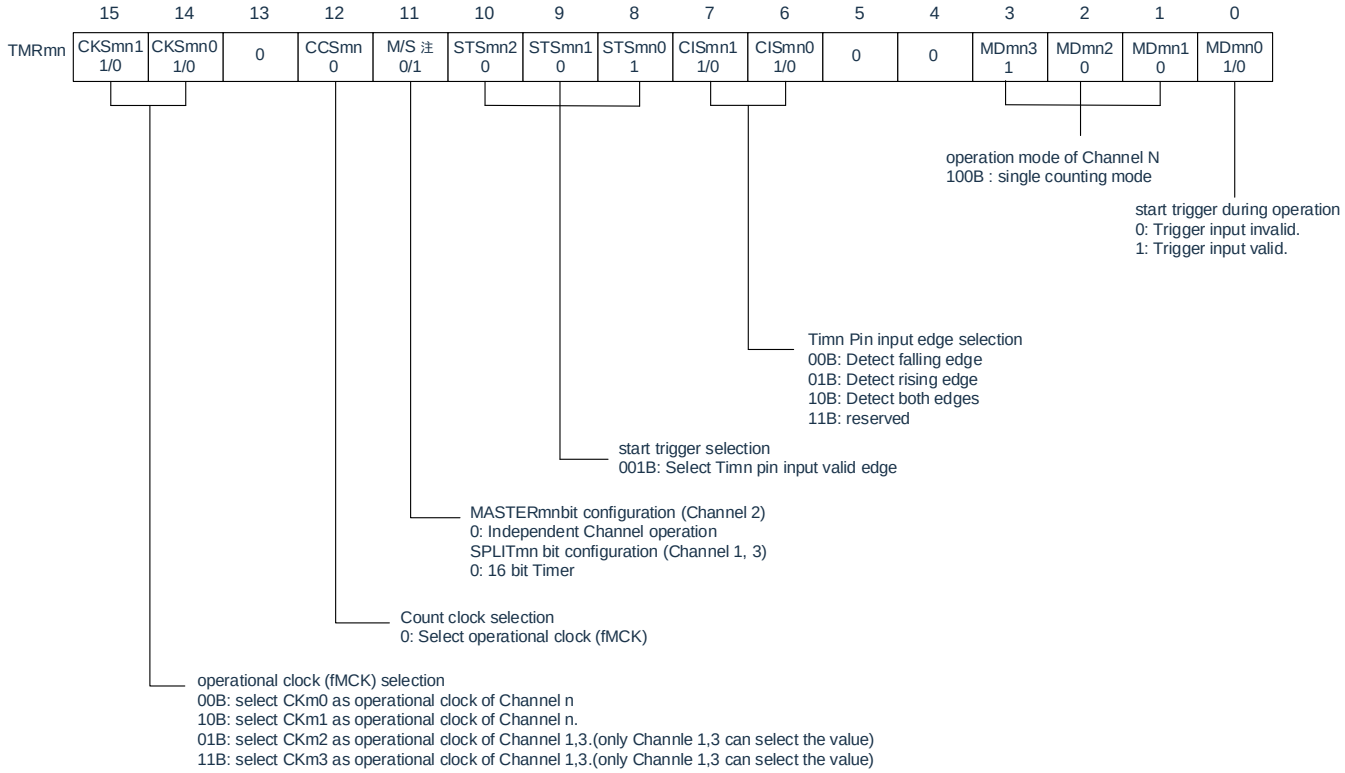
TImn :TImn Pin Input Signal

TCRmn: timer count register mn (TCRmn)

TDRmn: timer data register mn (TDRmn)

Figure 6-58 Example of register setting content when delaying counter function

(a) timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	bit	
TOm	TOmn	0: Output "0" from TOmn.
	0	

(c) timer output allow register m (TOEm)

	bit	
TOEm	TOEmn	0: Stop the TOmn output from the count run.
	0	

(d) Timer output level register m (TOLm)

	bit	
TOLm	TOLmn	0: Place "0" in the main channel output mode (TOMmn=0).
	0	

(e) Timer output mode register m (TOMm)

	bit	
TOMm	TOMmn	0: Sets the master channel output mode.
	0	

Note: TMRm2:MASTERmn bits
TMRm1, TMRm3:SPLITmn bit
TMRm0: Fixed as "0".
Note: m: Cell number (m=0)n: Channel Number (n=0~3)

Figure 6-59 Procedure when delaying counter function

	software operation	hardware state
Timer 4 initial configuration		Timer Unit m input clock is in stopped state (stop providing clock, not able to write into registers)
	set TM4mEN bit of peripheral enable register 0 (PER0) to '1' →	Timer Unit m input clock is in active state, all channels in operation stopped state. (start providing clock, can write all registers)
	configure Timer clock selection register m(TPSm), confirm CKm0~CKm3 clock frequency	
Channel Initial configuration	set corresponding bit of noise filter enable register 1 (NFEN1) to '0' (OFF) or '1' (ON). Configure Timer mode register mn (TMRmn) (confirm channel operation mode). Configure output delay time via timer data register mn (TDRmn) Set T0Emn bit to '0', and stop T0mn operation.	channel in operation stopped state (providing clock, consume portion of power)
Start operation	set TSmn bit to '1'. → Because TSmn bit is trigger bit, thus automatically return to '0'.	TEmn bit turns into '1' and enter into start trigger (detect Timn pin input valid edge or set TSmn bit to '1') detection waiting state.
	start decremental counting while detecting next start trigger. • Detect TImn pin input valid edge → • set TSmn bit to“1”via software	load TDRmn register value into Timer counting register mn (TCRmn)
in operation	can modify any TDRmn register configuration value. Can read TCRmn register anytime. Do not use TSRmn register.	Counter (TCR00) performs decremental counting. When TCRmn count reaches '0000H', then generate INTTMmn and before detecting the next start trigger (detect TImn pin input valid edge or set TSmn bit to '1'), TCRmn is "0000H" and stop counting.
stop operation	set TTmn bit to '1'. → Because TTmn bit is trigger bit, thus automatically return to '0'.	TEmn bit turns into '0' and stop counting. TCRmn register hold counted value and stop counting.
Timer 4 stop	set TM4mEN bit of peripheral enable register 0 (PER0) to '0' →	Timer Unit m input clock is not been provided.Perform initialization to all circuit and SFR of all channels.

restart operation

Note: m: Cell number (m=0)n: Channel Number (n=0~3)

6.9 Multi-channel coordinated operation function of universal timer unit

6.9.1 Operation as single trigger pulse output function

The two channels are used in pairs, and the single trigger pulse with arbitrary delay pulse width can be generated through the input of the TImn pin. The delay and pulse width can be calculated by the following formulas:

Delay={TDRmn (Master) Setting Value +2} Count Clock Cycle
Pulse Width={Set value for TDRmp (Secondary) ×} Count clock cycles

In a single count mode, that main control channel operate and count the delays. By detecting the start trigger, the timer count register mn (TCRmn) of the main control channel starts running and loads the value of timer data register mn (TDRmn). The TCRmn register counts down from the value of the mounted TDRmn register by counting the clock. If TCRmn becomes '0000H', INTTMmn is output and the count stops before the next start trigger is detected.

In a single count mode, the slave channel runs and counts the pulse width. The INTTMmn of the master channel is triggered as the start, the TCRmp register of the slave channel starts running and loads the value of the TDRmp register. The TCRmp register decrements the count from the value of the loaded TDRmp register by counting the clock. If the count value becomes "0000H," the INTTMmp is output and the count is stopped before the next start trigger (the INTTMmn of the primary channel) is detected. After the INTTMmn is generated from the main control channel and 1 counting clock passes, the output level of TOmp becomes effective level, and if TCRmp becomes "0000H".

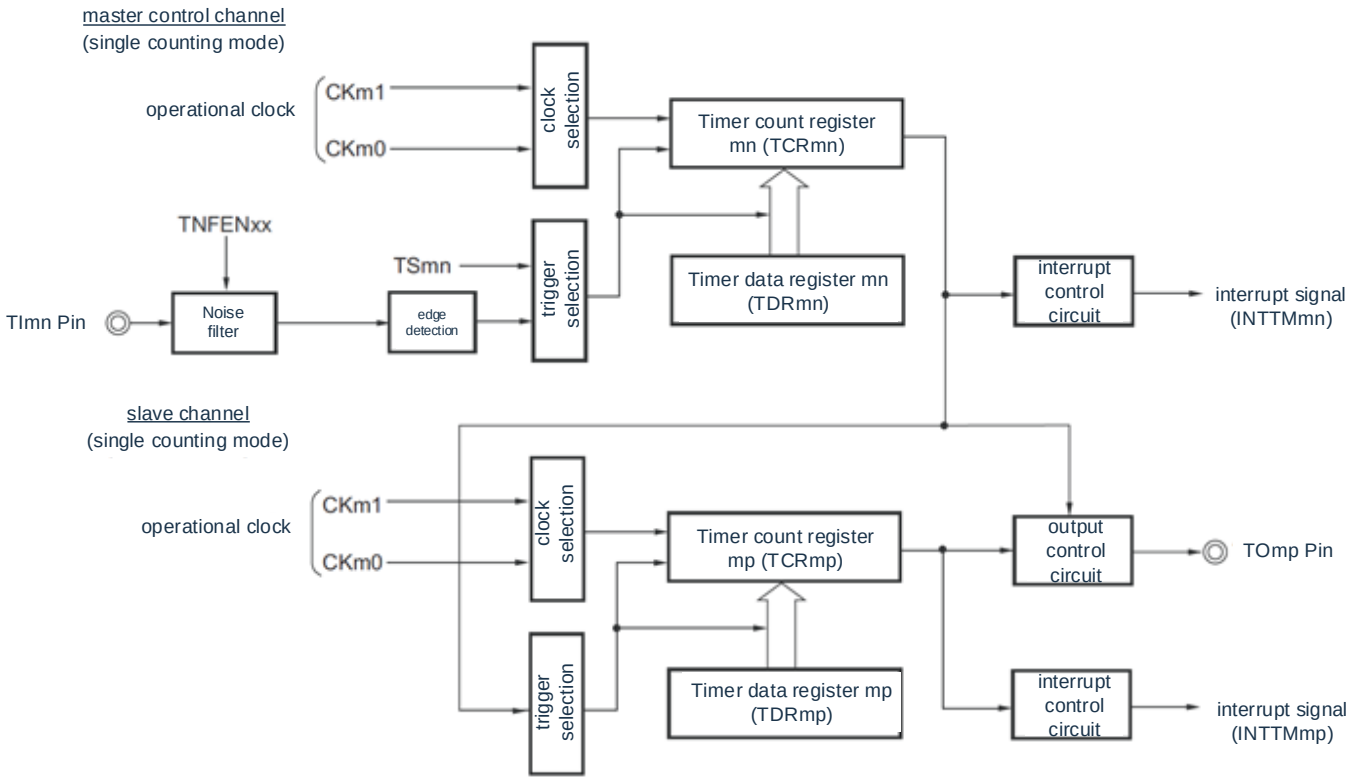
The software operation (TSmn=1) can be output as a single trigger pulse as a start trigger without using TImn pin input.

Note: Because the loading timing of TDRmn register of master channel and TDRmp register of slave channel are different, if TDRmn register and TDRmp register are rewritten during counting, abnormal waveform may be output. The TDRmn register must be overridden after the generation of INTTMmn and the TDRmp register must be overridden after the generation of INTTMmp.

Note: m: Cell number (m=0)n: Master channel number (n=0,2)

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

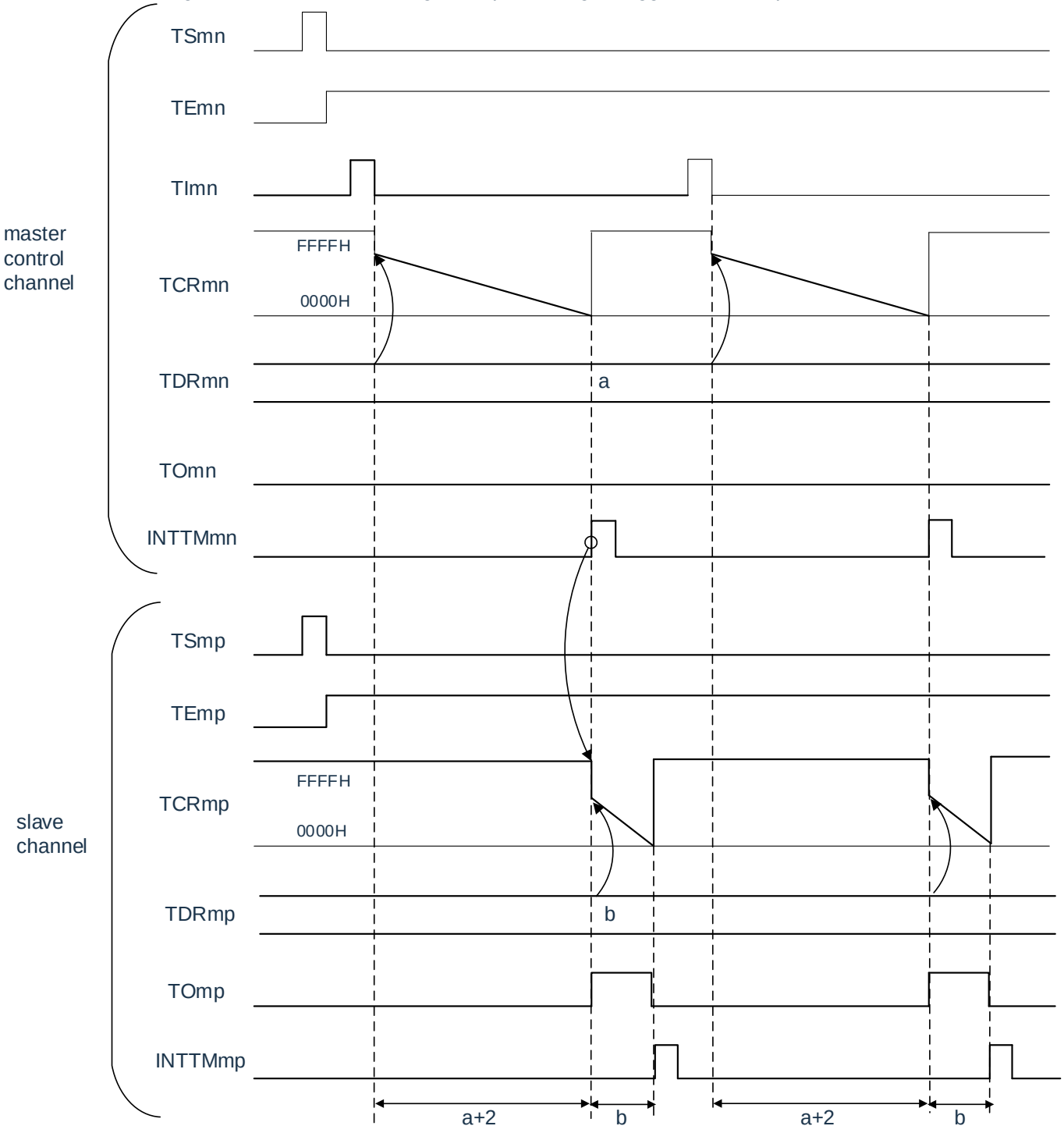
Figure 6-60 Block diagram for operating as single trigger pulse output function



Note: m: Cell number (m=0)n: Master channel number (n=0,2)

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

Figure 6-61 A Basic Timing Example of Single Trigger Pulse Output Function



Remarks: 1.m: Cell number (m=0)n: Master channel number (n=0,2)

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

2. TSmn, TSmp : bit n, p of the timer channel start register m (TSm).

TEmn, TEmp : timer channel allows bit n, p of status register m (TEm)

TImn, TImp : Input signal for TImn and TImp pins

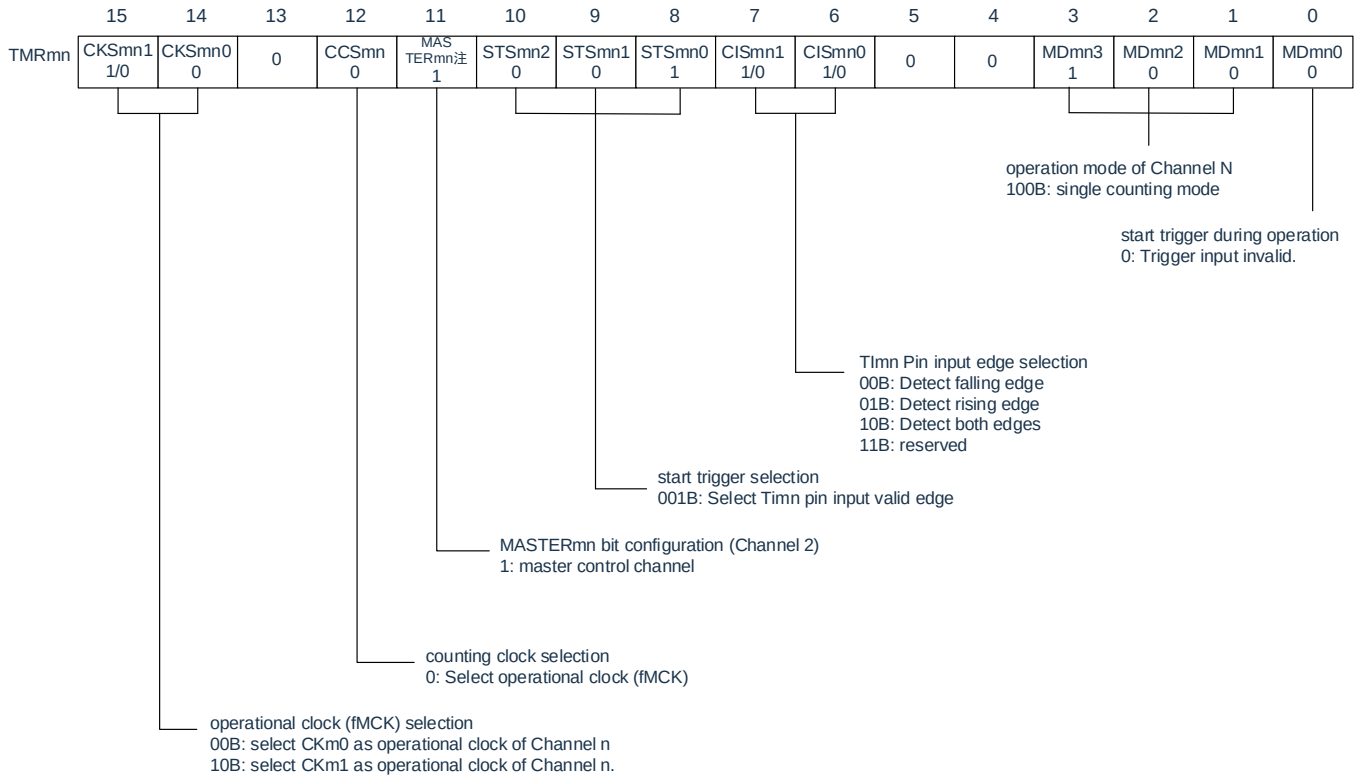
TCRmn, TCRmp : Timer count registers mn, mp (TCRmn, TCRmp)

TDRmn, TDRmp : timer data register mn, mp (TDRmn, TDRmp)

TOmn, TOmp :Output signal for TOmn and TOmp pins

Figure 6-62 Example of Register Setting Content for Single Trigger Pulse Output Function (Master Channel)

(a) Timer mode register mn (TMRmn)..



(b) Timer output register m (TOm)

	bit	
TOm	TOmn 0	0: Output "0" from TOmn.

(c) timer output allow register m (TOEm)

	bit	
TOEm	TOEmn 0	0: Stop the TOmn output from the count run.

(d) Timer output level register m (TOLm)

	bit	
TOLm	TOLmn 0	0: "0" at TOMmn=0 (Master Channel Output Mode).

(e) Timer output mode register m (TOMm)

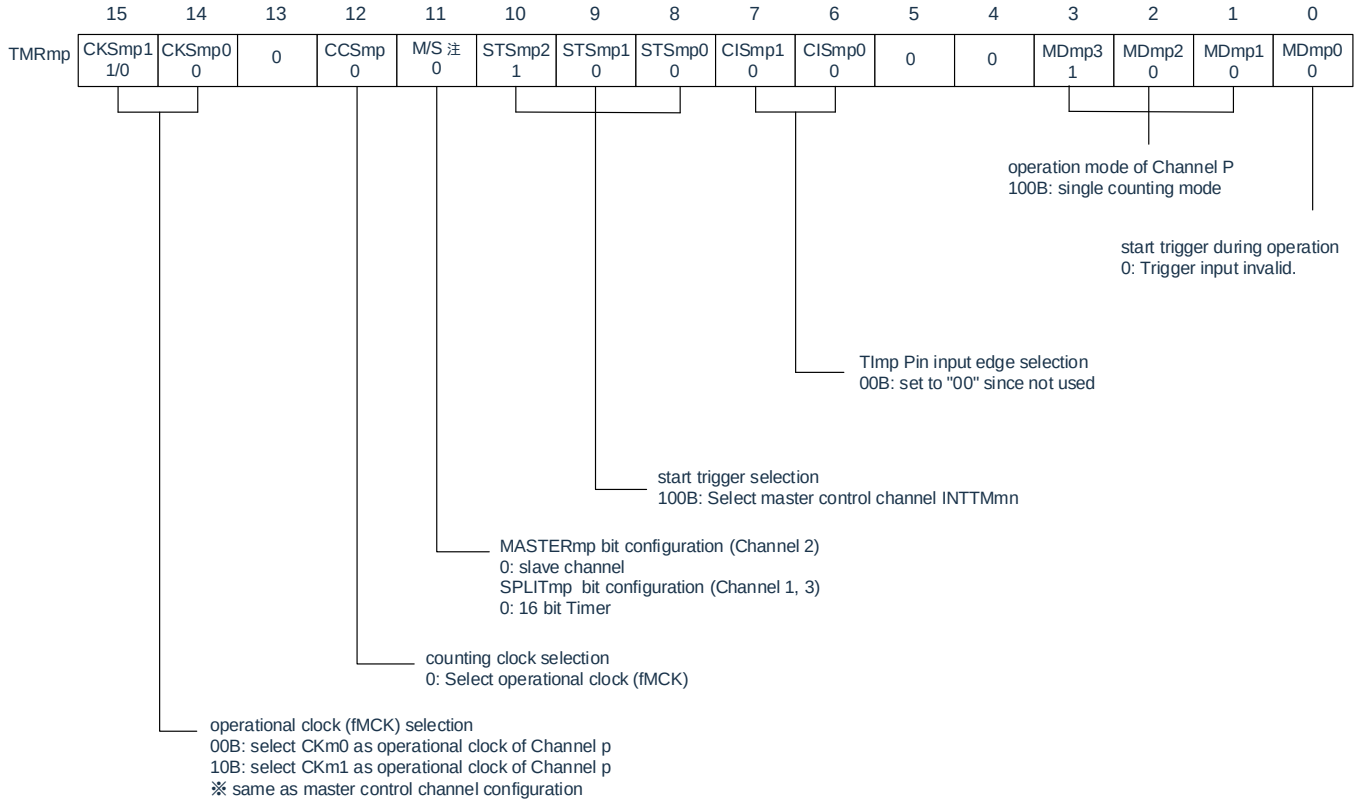
	bit	
TOMm	TOMmn 0	0: Sets the master channel output mode.

Note: TMRm2 :MASTERmn=1
TMRm0 : Fixed as "0".

Note: m: Cell number (m=0)n: Master channel number (n=0,2)

Figure 6-63 Example of Register Setting Content for Single Trigger Pulse Output Function (Slave Channel)

(a) timer mode register mp (TMRmp)



(b) Timer output register m (TOm)

	bit p
TOm	TOmp 1/0

0: Output "0" from TOmp.
1: Output "1" by TOmp.

(c) timer output allow register m (TOEm)

	bit p
TOEm	TOEmp 1/0

0: Stop the TOmp output from the count run.
1: Allow TOmp output by count run.

(d) Timer output level register m (TOLm)

	bit p
TOLm	TOLmp 1/0

0: Positive Logical Output (High Level Valid)
1: Negative Logical Output (Low Level Valid)

(e) Timer output mode register m (TOMm)m.

	bit p
TOMm	TOMmp 1

1: Sets the slave channel output mode.

Note: TMRm2:MASTERmp bit

TMRm1, TMRm3:SPLITmp bit

Note: m: Cell number (m=0)n: Master channel number (n=0,2)

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

Figure 6-64 Operating step (1/2) for a single trigger pulse output function

	software operation	hardware state
Timer 4 initial configuration		Timer Unit m input clock is in stopped state (stop providing clock, not able to write into registers)
	set TM4mEN bit of peripheral enable register 0 (PER0) to '1' →	Timer Unit m input clock is in active state, all channels in operation stopped state. (start providing clock, Start to provide clock, can write to each register)
	configure Timer clock selection register m(TPSm), confirm CKm0~CKm3 clock frequency	
Channel Initial configuration	set corresponding bit of noise filter enable register 1 (NFEN1) to 1'. Configure Timer mode registers mn,mp of 2 channels (TMRmn, TMRmp) (confirm channel operation mode). Set master control channel Timer data register mn (TDRmn) configure output delay time, and set slave channel TDRmp register pulse width.	channel in operation stopped state (providing clock, consume portion of power)
	slave channel configuration set TOMmp bit of timer output mode register m(TOMm) to '1' (slave channel output mode). Configure TOLmp bit. Configure TOmp bit and confirm TOmp otuput initial voltage. Set TOEmp bit to '1', enable TOmp output. Set port regsiter and port mode regsiter to '0'.	T0mp pin in Hi-Z output state. When port mode register set to output mode and port register as '0', output T0mp initial configured voltage level. Because channel is in operation stopped state, thus T0mp remains unchange. T0mp pin output T0mp configured voltage level.

Figure 6-65 Operating step (2/2) for a single trigger pulse output function

Start operation	set TOEmp bit (slave) to '1' (only limit to restart operation). Set TSmn bit)(master control) and TSmp bit(slave) of timer channel start register m(TSm) both to '1'. → Because TSmnn bit and TSmp bit are trigger bits, thus automatically return to '0'.	TEmn bit and Temp bit turn into '1' and master channel enter into start trigger (detect Timn pin input valid edge or set TSmn bit to '1') detection waiting state.Counter still in stop state.
	start master channel counting while detecting master channel start trigger. • Detect Tlmn pin input valid edge • set TSmn bit of master channel to“1”via software. Note.	master channel start counting
in operation	can only modify configure value of CISmn1 bit and CISmn0 bit of TMRmn register. Forbidden modifying TMRmn, TMRmp register and TOMmn bit, TOMmp bit, TOLmn bit and TOLmp bit configuration. Can read TCRmn register and TCRmp register anytime. Can not use TSRmn register and TSRmp register. can modify slave channel Tom regsiter and TOEm register configuration.	master channel load TDRmn register value into Timer technical register (TCRmn) via detecting start trigger (detecting Timn pin input valid edge or set TSmn bit of master channel to "1"), and perform decremental counting. If TCRmn counts till "0000H", then generating INTTMmn, and stop counting before next Timn pin input. Slave channel use INTTMmn of master channel as trigger, will load TDRmp register value into TCRmp regiter and counter start decremental counting. 1 counting clock cycle after master chanel outputs INTTMmn, it sets T0mp otuput voltage to valid voltage level. Then, if TCRmp count reaches "0000H", then set T0mp output voltage set to invalid votlage levle then stoop counting. Thereafter, the process repeats.
stop operation	set TTmn bit (master) and TTmp bit(slave) to '1'. → Because TTmn bit and TTmp bit are trigger bits, thus automatically return to '0'.	TEmn bit and Temp bit turn into '0' and stop counting. TCRmn register and TCRmp register hold counted value and stop counting. T0mp output not initialized and remains unchanged.
	set TOEmp bit of slave channel to '0', and configure TOmp bit. →	T0mp pin output T0mp configured voltage level.
timer 4 stop	Scenarios to maintain T0mp pin output voltage: set T0mp bit to '0' after set hold value to port register configuration. → In case T0mp pin output voltage does not need to be held: no configuration requried	maintain T0mp pin output voltage via Port function.
	set TM4mEN bit of peripheral enable register 0 (PER0) to '1' →	Timer Unit m input clock is not been provided.Perform initialization to all circuit and SFR of all channels. (TO00 bit turns into '0' and TO00 pin becomes port function)

restart operation

Note: can not set TSmn bit of slave channel to '1'.

Remark m: Unit number (m=0)

n: master control channel number (n=1,2)

p: slave channel number (n=0:p=1,2,3,n=2:p=3)

6.9.2 Operation as PWM Function

The two channels are used in pairs, and the pulse with arbitrary period and duty cycle can be generated. The period and duty cycle of the output pulse can be calculated by the following formula:

Pulse Period = {TDRmn (Master) Set Value +1×Count Clock Period Duty Ratio [%]={Setting Value for TDRmp (Dependent)}/{Setting Value for TDRmn (Master) +1}×100 0% output :Set value for TDRmp (Dependent)=0000H 100% output :Setting value for TDRmp (secondary) ≥{Setting value for TDRmn (primary) +1}

Note The duty cycle exceeds 100% when the TDRmp setting value is >{TDRmn setting value +1}, but 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 "1", the interrupt (INTTMmn) is output, and the setting value of the timer data register mn (TDRmn) is loaded into the timer count register mn (TCRmn), and counting down by counting the clocks. When counting to "0000H", the value of the TDRmn register is loaded into the TCRmn register again after outputting INTTMmn, and the count is decremented. This operation is repeated after setting the channel stop trigger bit (TTmn) of the timer channel stop register m (TTm).

When used as a PWM function, the main control channel performs a decremental count, which is a PWM output (TOmp) period until 0000H. The slave channel is used as a single count mode. Starting with the INTTMmn of the main control channel, the value of the TDRmp register is loaded into the TCRmp register and is decremented until "0000H". When counted to "0000H," INTTMmp is output and the next trigger is waited (INTTMmn of the master channel).

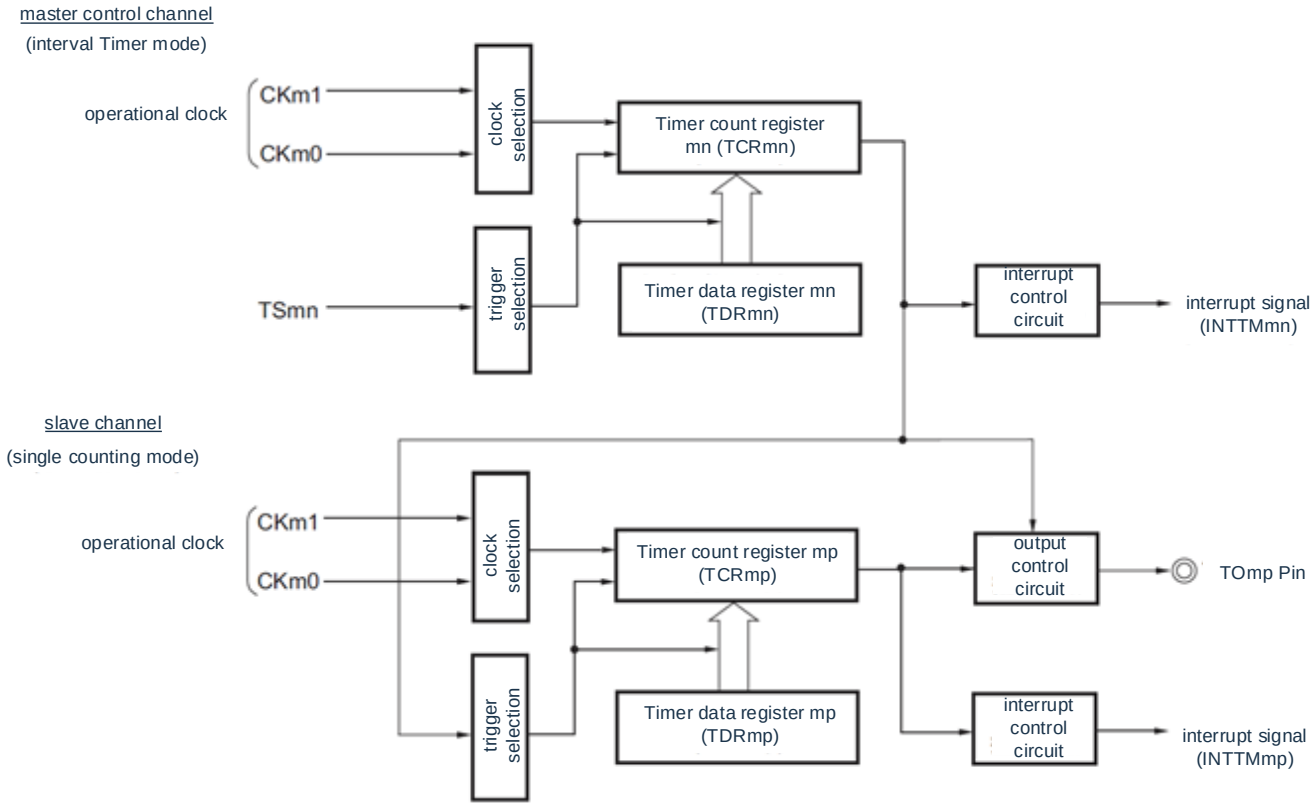
When used as a PWM function, the slave channel performs a decremental count, which is the duty cycle of the PWM output (TOmp) until '0000H'.

The PWM output (TOmp) becomes active after 1 clock generation of INTTMmn from the master channel and becomes invalid when the value of TCRmp register of slave channel is 0000H.

Note: When the timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel are to be rewritten. Because the TDRmn register and the TDRmp register are loaded into the TCRmn register and the TCRmp register when the master channel generates the INTTMmn, the TOmp pin can not output the expected waveform if the INTTMmn is rewritten before and after the master channel generates the respectively. Therefore, to override both the master TDRmn register and the slave TDRmp register, you must override these 2 registers immediately after the master channel generates INTTMmn.

Note: m: Cell number (m=0)n: Master channel number (n=0,2)
p: Slave channel number (n=0:p=1,2,3,n=2:p=3)

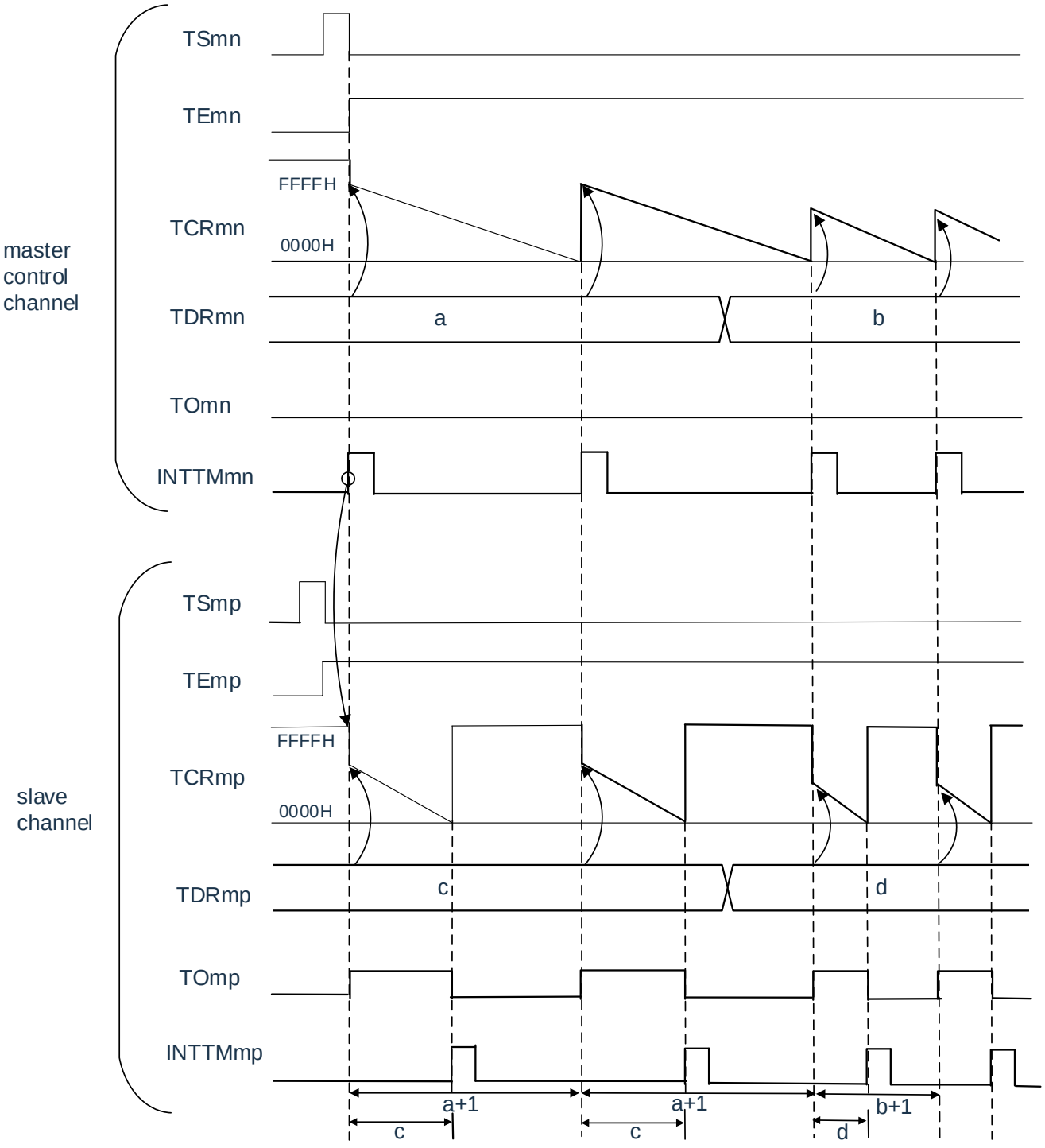
Figure 6-66 Block diagram for operation as PWM function



Note: m: Cell number (m=0)n: Master channel number (n=0,2)

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

Figure 6-67 As an example of the basic timing of the operation of the PWM function



Remarks: 1.m: Cell number (m=0)n: Master channel number (n=0,2)p: Slave channel number (n=0:p=1,2,3, n=2:p=3)

2. TSmn, TSmp: bitn, p of the timer channel start register m (TSm).

TEmn, TEmp: timer channel allows bitn, p of status register m (TEm)

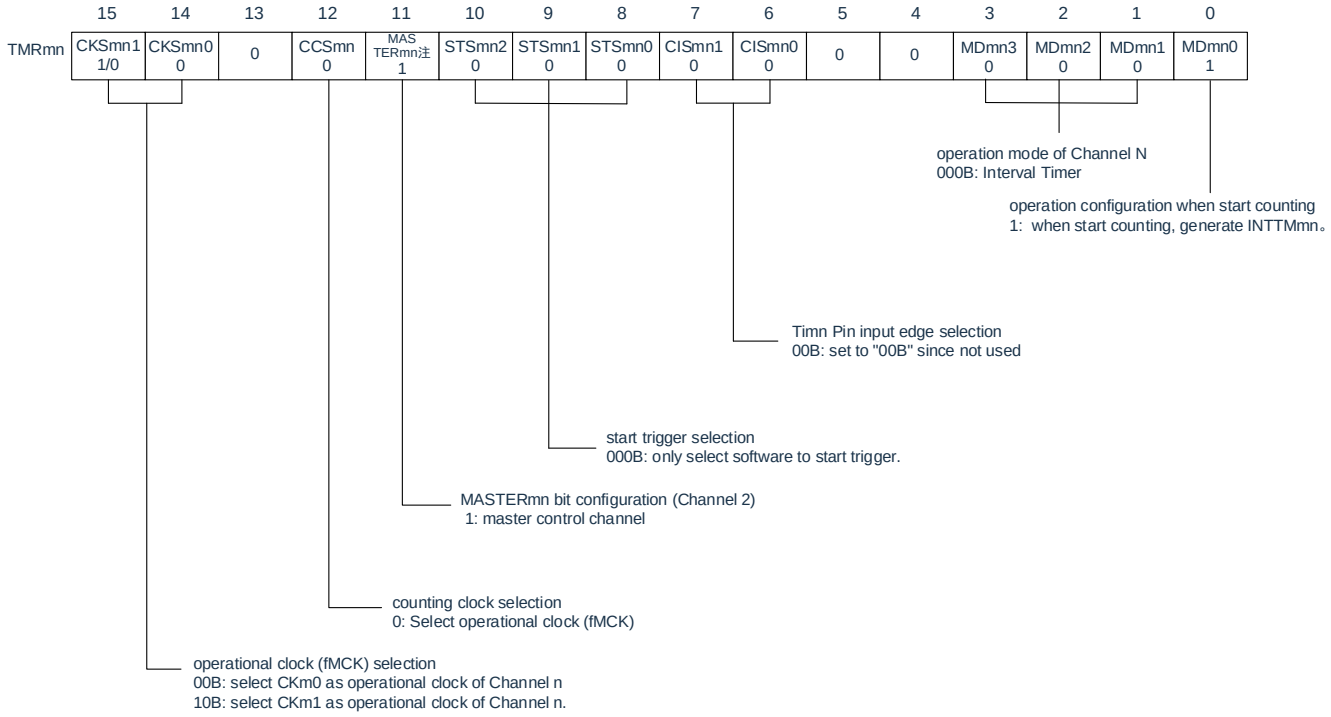
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)

TDRmn, TDRmp: timer data register mn, mp (TDRmn, TDRmp)

TOmn, TOmp: Output Signal for TOmn Pin and TOmp Pin

Figure 6-68 Example of Register Setting Content for PWM Function (Master Channel)

(a) timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	bit	
TOm	TOmn 0	0: Output "0" from TOmn.

(c) timer output allow register m (TOEm)

	bit	
TOEm	TOEmn 0	0: Stop the TOmn output from the count run.

(d) Timer output level register m (TOLm)

	bit	
TOLm	TOLmn 0	0: "0" is set at TOMmn=0 (Master Channel Output Mode).

(e) Timer output mode register m (TOMm)

	bit	
TOMm	TOMmn 0	0: Sets the master channel output mode.

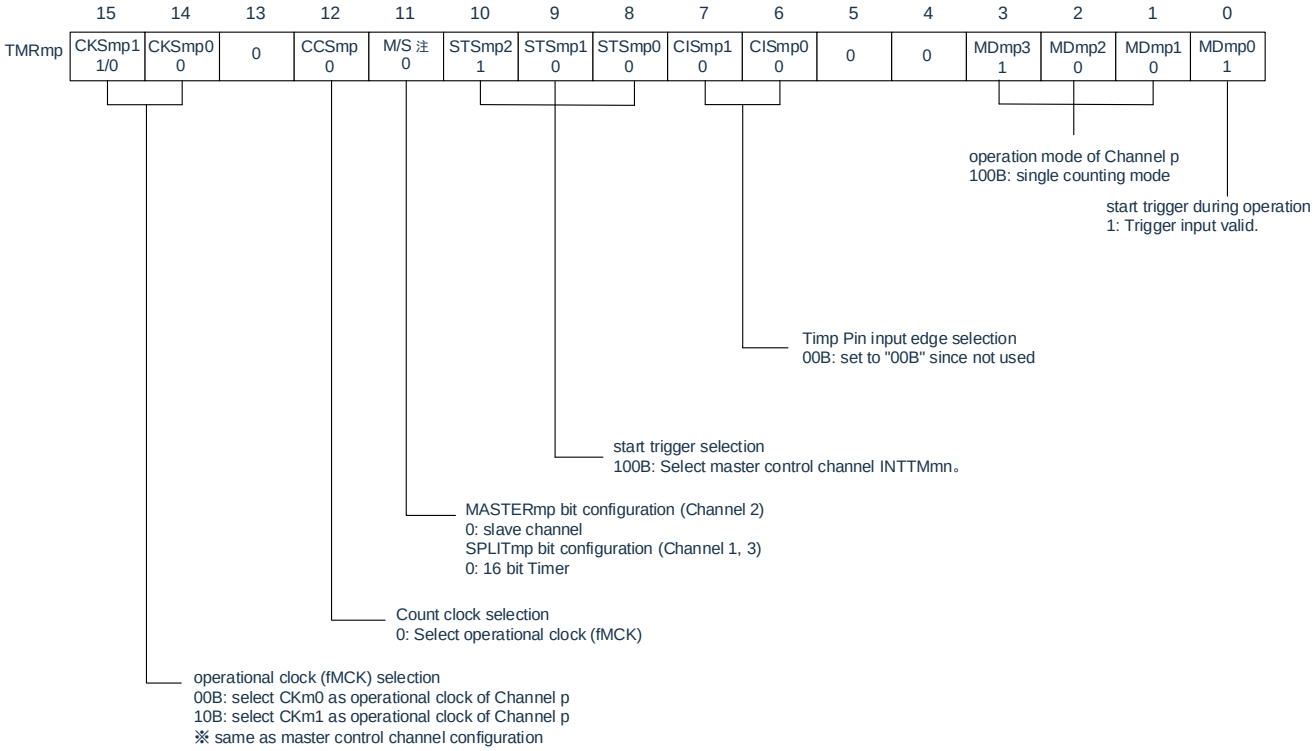
Note: TMRm2 :MASTERmn=1

TMRm0 : Fixed as "0".

Note: m: Cell number (m=0)n: Master channel number (n=0,2)

Figure 6-69 Example of Register Setting Content 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" from TOmp.
1: Output "1" by TOmp.

(c) timer output allow register m (TOEm)

	bit p
TOEm	TOEmp 1/0

0: Stop the TOmp output from the count run.
1: Allow TOmp output by count run.

(d) Timer output level register m (TOLm)

	bit p
TOLm	TOLmp 1/0

0: Positive Logical Output (High Level Valid)
1: Negative Logical Output (Low Level Valid)

(e) Timer output mode register m (TOMm)

	bit p
TOMm	TOMmp 1

1: Sets the slave channel output mode.

Note: TMRm2:MASTERmp bit

TMRm1, TMRm3:SPLITmp bit

Note: m: Cell number (m=0)n: Master channel number (n=0,2)

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

Figure 6-70 Operating steps for PWM functions (1/2)

Timer 4 initial configuration		Timer Unit m input clock is in stopped state (stop providing clock, not able to write into registers)
	set TM4mEN bit of peripheral enable register 0 (PER0) to '1'	Timer Unit m input clock is in active state, all channels in operation stopped state.
	configure Timer clock selection register m(TPSm), confirm CKm0~CKm3 clock frequency	
Channel Initial configuration	configure using timer mode register mn,mp (TMRmn,TMRmp) of 2 channels (confirm channel operation mode). Configure interal(period) value of Timer data register mn (TDRmn) of master control channel, and configure duty-cycle of slave channel TDRmp.	channel in operation stopped state (providing clock, consume portion of power)
	slave channel configuration set TOMmp bit of timer output mode register m(TOMm) to '1' (slave channel output mode). Configure TOLmp bit. Configure TOmp bit and confirm TOmp otuput initial voltage. Set TOEmp bit to '1', enable TOmp output. Set port regsiter and port mode regsiter to '0'.	T0mp pin in Hi-Z output state. When port mode register set to output mode and port register as '0', output T0mp initial configured voltage level. Because channel is in operation stopped state, thus T0mp remains unchange. T0mp pin output T0mp configured voltage level.

Figure 6-71 Operating steps for PWM functions (2/2)

restart operation	Start operation	set TOEmp bit (slave) to '1' (only limit to restart operation). Set TSmn bit)(master control) and TSmp bit(slave) of timer channel start register m(TSm) both to '1'. Because TSmnn bit and TSmp bit are trigger bits, thus automatically return to '0'.	TEmn bit and TEmp bit both turns into '1'. Master channel start counting and generate INTTMmn. Using this trigger, slave channel also start counting.
	in operation	forbidden modifying TMRmn register and TMRmp register and TOMmn bit, TOMmp bit, TOLmn bit and TOLmp bit configuration. can mmodify TDRMn register and TDRmp register configuration after master channel generates INTTMmn. Can read TCRmn reigsrer and TCRmp register anytime. can not use TSRmn register and TSRmp register.	master channel load TDRmn register value into Timer counting register (TCRmn) and perform decremental counting. If TCRmn counts till "0000H", then generating INTTMmn. At the same time, load TDRmn register value into TCRmn register and restart decremental counting. Slave channel use INTTMmn of master channel as trigger, will load TDRmp register value into TCRmp regiter and counter start decremental counting. 1 counting clock cycle after master chanel outputs INTTMmn, it sets T0mp otuput voltage to valid voltage level. Then, if TCRmp count reaches "0000H", then set T0mp output voltage set to invalid votlage levle then stoop counting. Thereafter, the process repeats.
	stop operation	set TTmn bit (master) and TTmp bit(slave) to '1'. Because TTmn bit and TTmp bit are trigger bits, thus automatically return to '0'.	TEmn bit and Temp bit turn into '0' and stop counting. TCRmn register and TCRmp register hold counted value and stop counting. T0mp output not initialized and remains unchanged.
		set TOEmp bit of slave channel to '0', and configure TOmp bit.	T0mp pin output T0mp configured voltage level.
	timer 4 stop	Scenarios to maintain T0mp pin output voltage: set T0mp bit to '0' after set hold value to port register configuration. In case T0mp pin output voltage does not need to be held: no configuration requried	maintain T0mp pin output voltage via Port function.
		set TM4mEN bit of peripheral enable register 0 (PER0) to '1'	Timer Unit m input clock is not been provided.Perform initialization to all circuit and SFR of all channels. (T0mp bit turns into '0' and T0mp pin becomes port function) (TO00 bit turns into '0' and TO00 pin becomes port function)

Remark m: Unit number (m=0)

n: master control channel number (n=0,2)

p: slave channel number (n=0:p=1,2,3,n=2:p=3)

6.9.3 Operation as Multiple PWM Output Function

This is a function of performing multiple PWM outputs with different duty cycles by extending the PWM function and using multiple slave channels.

For example, when two slave channels are used in pairs, the period and duty cycle of the output pulse can be calculated using the following equation:

Pulse Period = {TDRmn (Master) Set Value +1 Count Clock Period Duty cycle 1[%]={TDRmp (Dependent 1) Set Value}/{TDRmn (Master) Set Value+1}×100 Duty Cycle 2[%]={TDRmq (Secondary 2) Set Value}/{TDRmn (Master) Set Value+1}×100

Note: When TDRmp (Slave 1) has a set value >{TDRmn (Master) +1} or {TDRmq (Slave 2) has a set value +1} of {TDRmn (Master), the duty cycle exceeds 100% but is 100% output.

In interval timer mode, timer count register mn (TCRmn) of the main control channel runs and counts the cycles. In a single count mode, the TCRmp register of the slave channel 1 runs and counts the duty cycle and outputs a PWM waveform from the TOmp pin. Starting with the INTTMmn of the master channel, the timer data register mp (TDRmp) is loaded into the TCRmp register and decremented. If TCRmp becomes "0000H," the INTTMmp is output and counts are stopped before the next start trigger (INTTMmn of the master channel) is entered. After the INTTMmn is generated from the main control channel and 1 counting clock passes, the output level of TOmp becomes effective level, and if TCRmp becomes "0000H".

The same as the TCRmp register of the slave channel 1, in a single count mode, the TCRmq register of the slave channel 2 runs and counts the duty cycle and outputs a PWM waveform from the TOmq pin. Starting with the INTTMmn of the main control channel, the value of the TDRmq register is loaded into the TCRmq register and decremented. If TCRmq becomes "0000H," the INTTMmq is output and counts are stopped before the next start trigger (INTTMmn of the master channel) is entered. After the INTTMmn is generated from the main control channel and 1 counting clock passes, the output level of TOmq becomes effective level, and if TCRmq becomes "0000H".

When the channel 0 is used as the main control channel by such operation, up to three PWM signals can be output simultaneously.

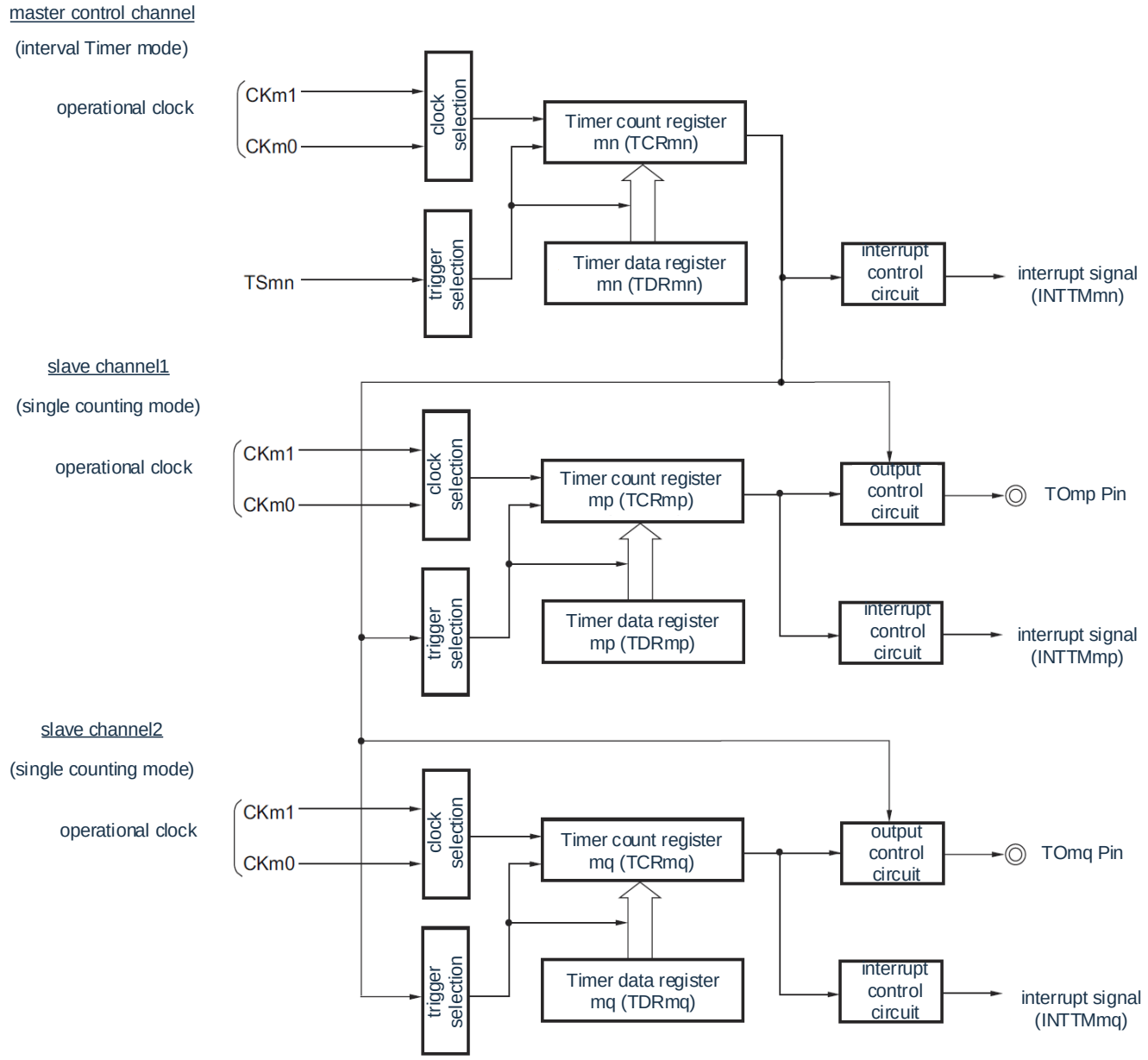
Note: At least 2 write accesses are required when the timer data register mn(TDRmn) of the master channel and the TDRmp register of slave channel 1. Because the value of the TDRmn register and the TDRmp register are loaded into the TCRmn register and the TCRmp register when the master channel generates the INTTMmn, if the TOmp pin is rewritten before the master channel generates the INTTMmn and after the generation, the expected waveform cannot be output. Therefore, to override both the master TDRmn register and the slave TDRmp register, you must override both registers immediately after generating INTTMmn in the master channel (also applies to the slave TDRmq register).

Note: m: Cell number (m=0)n: Master channel number (n=0)

p: Dependent Channel Number

q: Slave channel number n<p<q≤3 (p and q are integers greater than n)

Figure 6-72 Block diagram for operating the multiple PWM output function (output of 2 PWM cases)

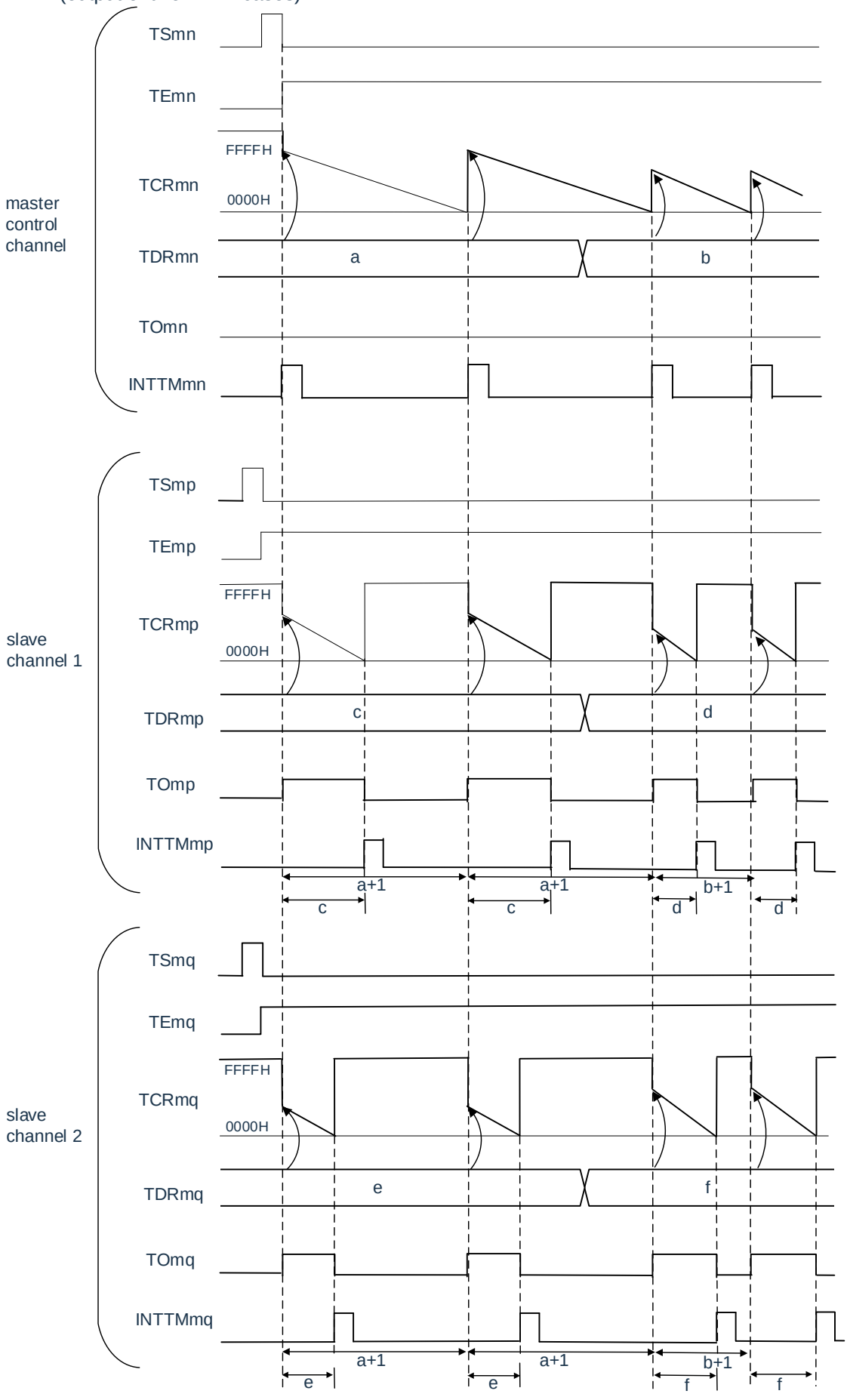


Note: m: Cell number (m=0)n: Master channel number (n=0)

p: Slave channel number q: Dependent Channel Number

n<p<q≤3 (p and q are integers greater than n)

Figure 6-73 As an example of the running basic timing of the multiple PWM output function (output of two PWM cases)



Remarks: 1, m: Cell number (m=0)n: Master channel number (n=0)

p: Slave channel number q: Dependent Channel Number

n<p<q≤3 (p and q are integers greater than n)

2. TSmn, TSmp, TSmq: bitn, p,q of timer channel start register m (TSm)

TEmn, TEmp, TEmq: Timer channel allows bitn, p, q of state register m (TEm).

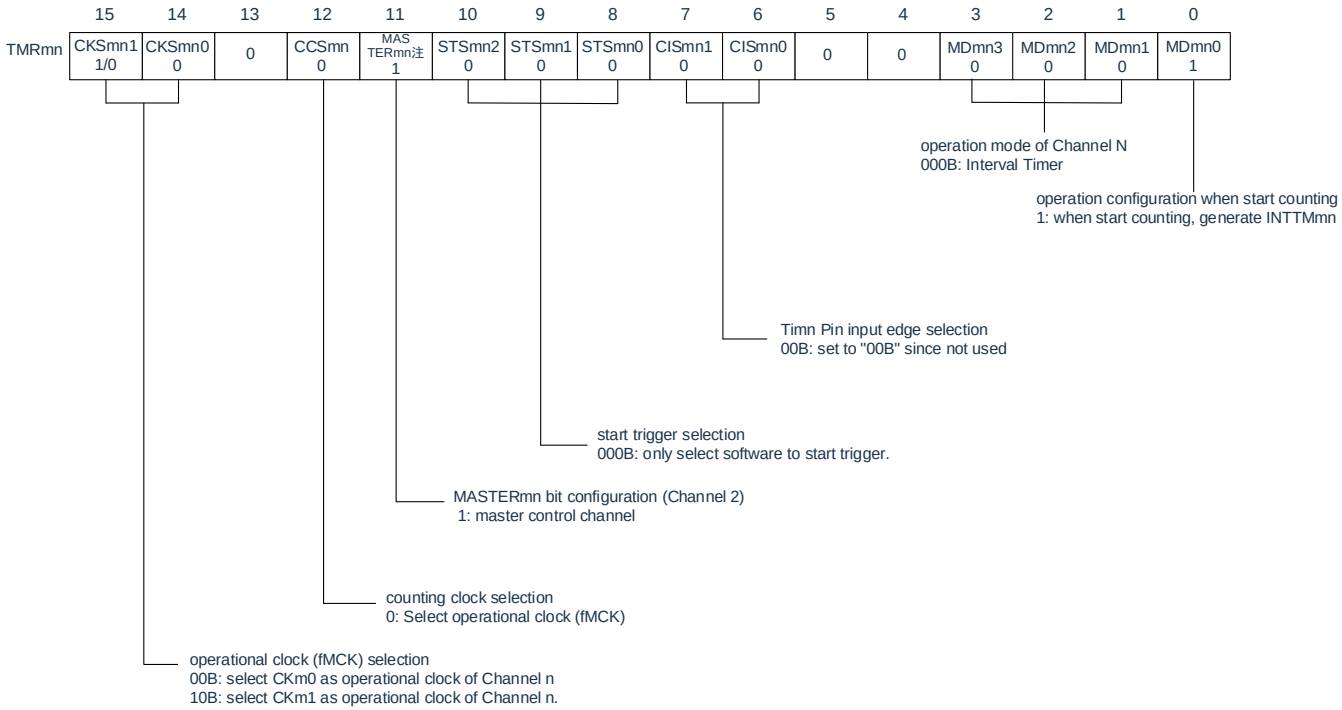
TCRmn, TCRmp, TCRmq: Timer count registers mn, mp, mq (TCRmn, TCRmp, TCRmq)

TDRmn, TDRmp, TDRmq: timer data register mn, mp, mq (TDRmn, TDRmp,TDRmq)

Output Signal of TOmn, TOmp, TOmq:TOmn, TOmp,TOmq Pin

Figure 6-74 Example of Register Setting Content for Multiple PWM Output Functions (Master Channel)

(a) timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	bit	
TOm	TOmn 0	0: Output "0" from TOmn.

(c) timer output allow register m (TOEm)

	bit	
TOEm	TOEmn 0	0: Stop the TOmn output from the count run.

(d) Timer output level register m (TOLm)

	bit	
TOLm	TOLmn 0	0: "0" is set at TOMmn=0 (Master Channel Output Mode).

(e) Timer output mode register m (TOMm)

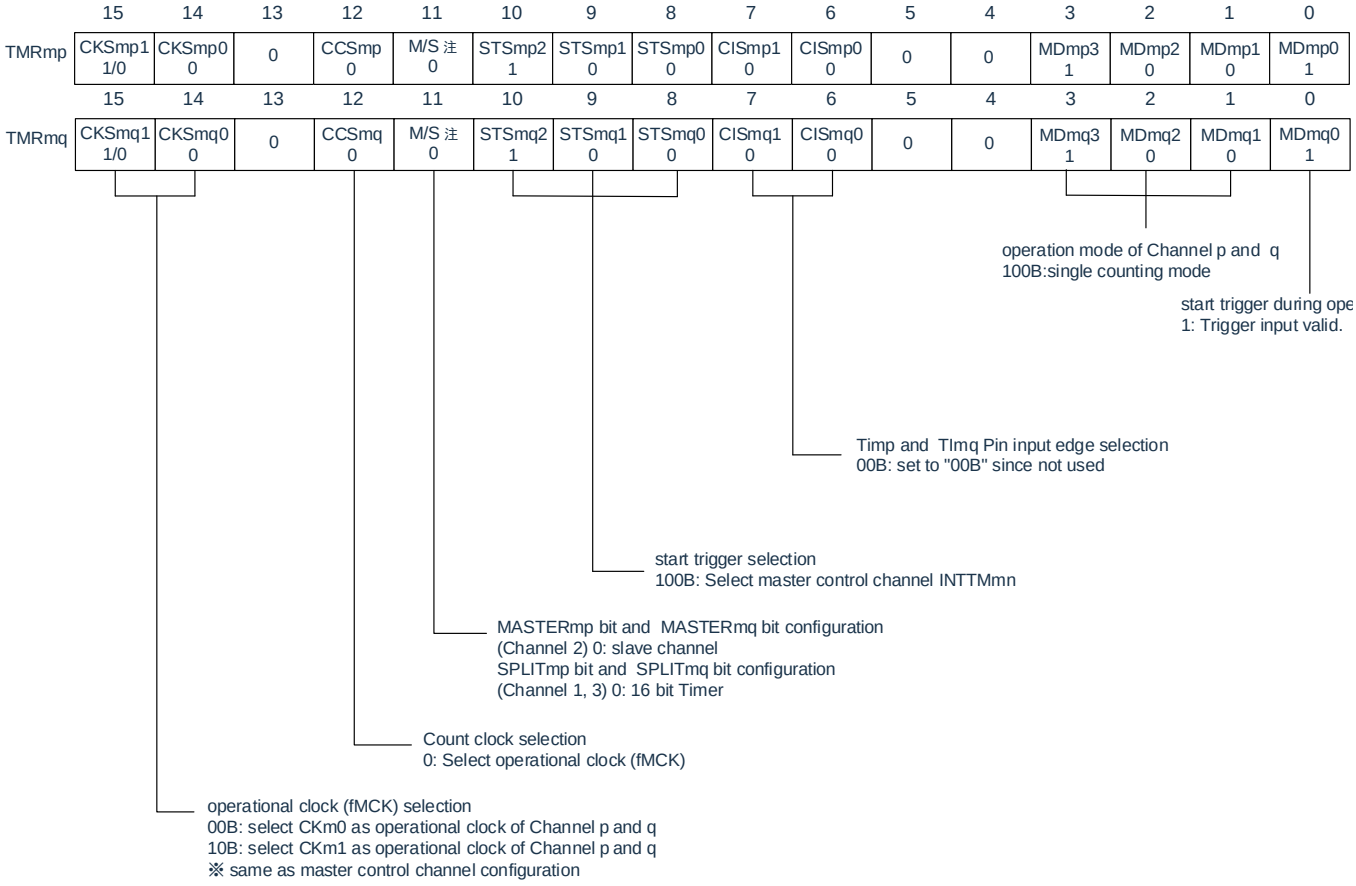
	bit	
TOMm	TOMmn 0	0: Sets the master channel output mode.

Note: TMRm2 :MASTERmn=1
TMRm0 : Fixed as "0".

Note: m: Cell number (m=0)n: Master channel number (n=0)

Figure 6-75 Example of register setting content for multiple PWM output functions (slave channel) (output of 2 PWM cases)

(a) timer mode registers mp, mq (TMRmp, TMRmq)



(b) Timer output register m (TOm)

TOm	bit q	bit p
	TOmq 1/0	TOmp 1/0

0: Output "0" by TOmp and TOmq.
1: Output "1" by TOmp and TOmq.

(c) timer output allow register m (TOEm)

TOEm	bit q	bit p
	TOEmq 1/0	TOEmp 1/0

0: Stops the TOmp and TOmq output from the count run.
1: Allow TOmp and TOmq output from count runs.

(d) Timer output level register m (TOLm)

TOLm	bit q	bit p
	TOELq 1/0	TOELp 1/0

0: Positive Logical Output (High level valid)
1: Negative Logical Output (Low level valid)

(e) Timer output mode register m (TOMm)

TOMm	bit q	bit p
	TOMLq 1	TOMLp 1

1: Sets the slave channel output mode.

NOTE: TMRm2: MASTERmp bit, MASTERmq bit
TMRm1, TMRm3: SPLITmp bit, SPLITmq bit
Note: m: Cell number (m=0)n: Master channel number (n=0)
p: Slave channel number q: Dependent Channel Number
n<p<q≤3 (p and q are integers greater than n)

Figure 6-76 Operation steps for multiple PWM output functions (output of 2 PWM cases) (1/2)

	software operation	hardware state
Timer 4 initial configuration		Timer Unit m input clock is in stopped state (stop providing clock, not able to write into registers)
	set TM4mEN bit of peripheral enable register 0 (PER0) to '1' →	Timer Unit m input clock is in active state, all channels in operation stopped state.
	configure Timer clock selection register m(TPSm), confirm CKm0~CKm3 clock frequency	
Channel Initial configuration	configure using timer mode register mn,mp (TMRmn,TMRmp) of 2 channels (confirm channel operation mode). Configure interal(period) value of Timer data register mn (TDRmn) of master control channel, and configure duty-cycle of slave channel TDRmp.	channel in operation stopped state (providing clock, consume portion of power)
	slave channel configuration set TOMmp bit and TOLmq bit of timer output mode register m(TOMm) to '1' (slave channel output mode). Configure TOLmp and Tomq bit to '0'. → Configure TOmp bit and Tomq bit, confirm TOmp and Tomq otuput initial voltage. → Set TOEmp bit and TOEmq to '1', enable TOmp and Tomq output. → Set port regsiter and port mode regsiter to '0'.	T0mp pin in Hi-Z output state. When port mode register set to output mode and port register as '0', output T0mp and T0mq initial configured voltage level. Because channel is in operation stopped state, thus T0mp and T0mq remains unchange. T0mp pin and T0mq pin output T0mp and T0mq configured voltage level.

Figure 6-77 Operation steps for multiple PWM output functions (output of 2 PWM cases) (2/2)

restart operation	Start operation	(only during restart operation, TOEmp bit and TOEmq bit (slave) will set to '1'). Set TSmn bit(master), TSmp bit and TSmq bit (slave) of timer channel start register m(TSm) all set to '1' at the same time. Because TSmn bit, TSmp and TSmq bit are all trigger bits, thus automatically return to '0'.	TEmn bit and TEmp bit both turns into '1'. Master channel start counting and generate INTTMmn. Using this trigger, slave channel also start counting.
	in operation	forbidden modifying TMRmn register and TMRmp register and TOMmn bit, TOMmp bit, TOLmn bit and TOLmp bit configuration. can mmodify TDRMn register and TDRmp register configuration after master channel generates INTTMmn. Can read TCRmn reigsrer and TCRmp register anytime. can not use TSRmn register and TSRmp register.	master channel load TDRmn register value into Timer counting register (TCRmn) and perform decremental counting. If TCRmn counts till "0000H", then generating INTTMmn. At the same time, load TDRmn register value into TCRmn register and restart decremental counting. Slave channel 1 use INTTMmn of master channel as trigger, will load TDRmp register value into TCRmp regiter and counter start decremental counting. 1 counting clock cycle after master chanel outputs INTTMmn, it sets T0mp otuput voltage to valid voltage level. Then, if TCRmp count reaches "0000H", then set T0mp output voltage set to invalid votlage levle then stoop counting. Slave channel 2 use INTTMmn of master channel as trigger, will load TDRmq register value into TCRmq regiter and counter start decremental counting. 1 counting clock cycle after master chanel outputs INTTMmn, it sets T0mq otuput voltage to valid voltage level. Then, if TCRmq count reaches "0000H", then set T0mq output voltage set to invalid votlage levle then stoop counting. Thereafter, the process repeats.
	stop operation	set TTmn bit (master), TTmp bit and TTmq bit(slave) to '1'. Because TTmn bit, TTmp bit, TTmq bit are trigger bits, thus automatically return to '0'.	TEmn bit, Temp bit and Temq turn into '0' and stop counting. TCRmn, TCRmp TCRmq registers hold counted value and stop counting. T0mp and T0mq output not initialized and remains unchanged.
		set TOEmp bit and TOEmq bit of slave channel to '0', and configure Tomp and TOmq bit.	T0mp pin and T0mq pin output T0mp and T0mq configured voltage level.
	timer 4 stop	Scenarios to maintain T0mp pin and Tomq pin output voltage: set T0mp bit and Tomq bit to '0'. In case T0mp pin and Tomq output voltage does not need to be held: no configuration requried	maintain T0mp pin and Tomq output voltage via Port function.
		set TM4mEN bit of peripheral enable register 0 (PER0) to '1'	Timer Unit m input clock is not been provided.Perform initialization to all circuit and SFR of all channels. (T0mp bit and T0mq bit turn into '0' and T0mp pin and Tomq becomes port function) (TO00 bit turns into '0' and TO00 pin becomes port function)

Note: m: Cell number (m=0)n: Master channel number (n=0)
p: Slave channel number q: Dependent Channel Number
n<p<q≤3 (p and q are integers greater than n)

6.10 Precautions when using a universal timer unit

6.10.1 Precautions When Using Timer Output

According to the product, the pins to which the timer output function is assigned may also be assigned the output of other multiplexing functions. In this case, when using the timer output, it is necessary to set the initial value of the other multiplexing function output.

Please refer to the "2.5 Register Settings When Using Multiplexing".

Chapter 7 Timer A

7.1 Function of timer A

The timer A is a 16-bit timer capable of measuring the pulse output, the pulse width and period of the external input, and counting the external events.

The 16-bit timer consists of a reload register and a decrement counter, which are assigned at the same address. If the TA0 register is accessed, the reload register and the counter can be accessed.

The specifications and block diagrams of timer A are shown in Tables 7-1 and Figure 7-1 respectively.

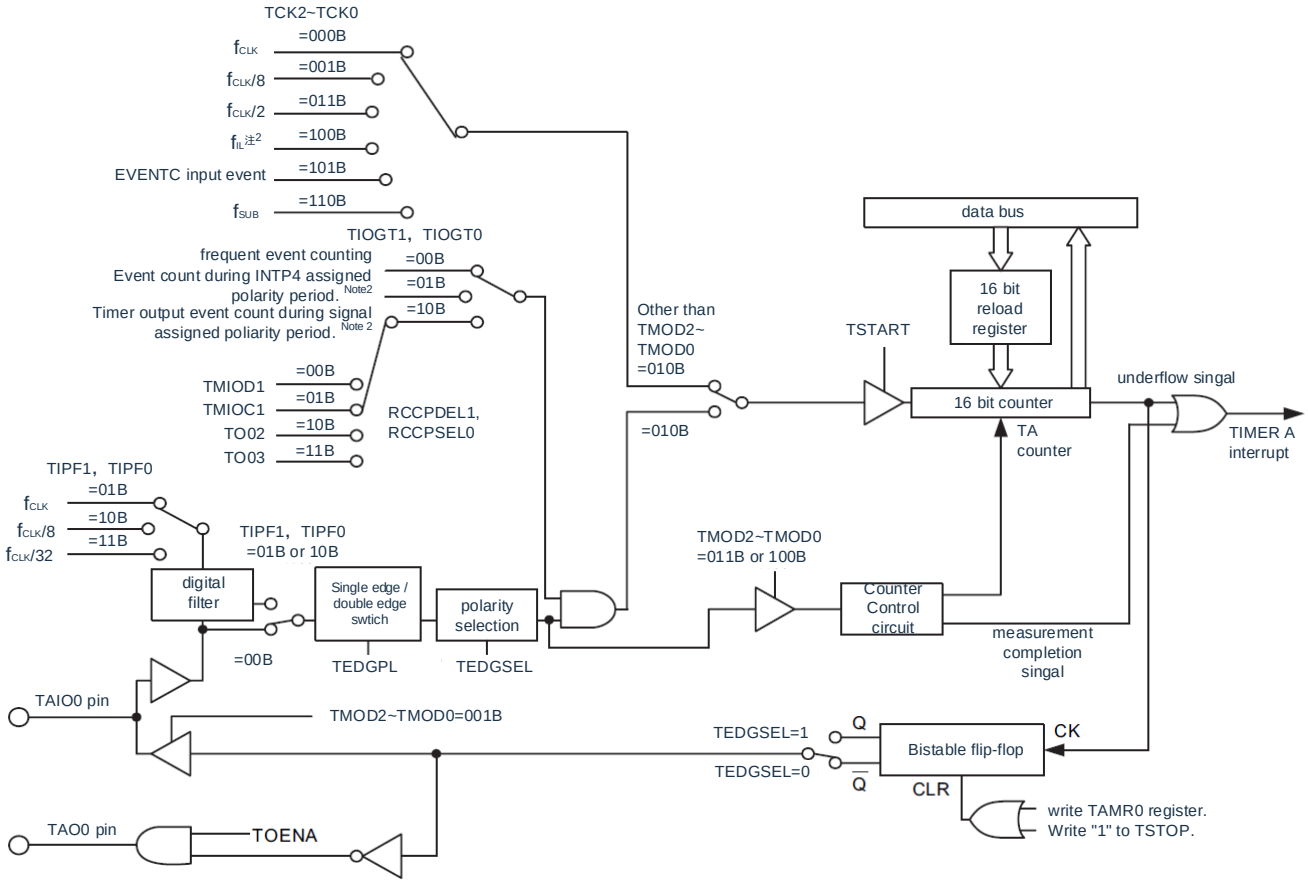
Table 7-1 Specifications for timer A

Project		Content
operation mode	timer mode	Counts the count source.
	pulse output mode	The counter source is counted, and pulses with opposite polarity are output when the timer underflows.
	Event Counter Mode	Counts external events. It can also run in deep sleep mode.
	pulse width measurement mode	The pulse width of the external input is measured.
	pulse period measurement mode	The pulse period of the external input is measured.
Count Source (Runtime Clock)		You can select events entered by f_{CLK},f_{CLK}/2,f_{CLK}/8,f_{IL}、 f_{SUB}or EVENTC.
interrupt		· When the counter overflows · At the end of the external input (TAIO) effective width measurement in pulse width measurement mode · When you enter the set-up edge of an external input (TAIO) in pulse period measurement mode
Select Features		· Collaboration with EVENTC: You can select events entered by EVENTC as the count source.

7.2 Structure of timer A

The block diagram and the pin structure of the timer A are shown in Figure 7-1 and Table 7-2 respectively.

Figure 7-1 Block diagram of timer A



Note:

1. To select fIL as the count source, the secondary system clock must be supplied with the WUTMMCK0 position "1" of the OSMC. Register. However, when fSUB is selected as the real-time clock or the count source of the 15-bit interval timer, it cannot be selected.
2. The polarity can be selected through the RCCPSEL2 bit of the TAISR0 register.

Table 7-2 Pin structure of timer A

name of the feet	Input/Output	Features
INTP4	Enter	Event counter mode control for timer A
TAIO Note	Input/Output	External event input and pulse output of timer A
TAO NOTE	output	Pulse output of timer A

Note: The configuration of the TAO pin can be selected by the PIOR12 bit and PIOR13 bit of PIOR1 register, and the configuration of the TAIO pin can be selected by PIOR10 bit and PIOR11 bit. Refer to "Chapter 2 Pin Functions" for details.

7.3 Register for control timer A

The register for controlling timer A is shown in Table 7-3.

Table 7-3 Register for control timer A

register name	symbol
Peripheral I/O Redirection Register 1	PIOR1
Peripheral Enable Register 1	PER1
Secondary system clock providing mode control register	OSMC
timer A count register 0 note·	TA0
timer A control register 0	TACR0
Timer AI/O control register 0	TAIOC0
timer A mode register 0	TAMR0
timer A event pin selection register 0	TAISR0
port register x	Px
port mode register x	PMx

Note: When accessing the TA0 register, the CPU does not enter the processing of the next instruction and is in the waiting state of the CPU processing. Therefore, when this wait occurs, the number of clocks executed by the instruction increases the number of clocks waiting. The number of read and write waiting clocks when accessing the TA0 register is 1 clock.

7.3.1 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.

You must set bit0 (TMAEN) to "1" when you want to use timer A.

The PER1 register is set by an 8-bit memory operation instruction.

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

Figure 7-2 Format of Peripheral Admission Register 1(PER1)

Location: 4002081AH After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN	TMMEN	DMAEN	PWMPEN	TMCEN	TMAEN

TMAEN	Control of an input clock of timer A is provided
0	Stop provide an input clock. · SFR used by timer A cannot be written. · Timer A is in a reset state.
1	Provides an input clock. · Read and write the SFR used by timer A.

Note 1. To set timer A, you must first set the TMAEN position "1". When the TMAEN bit is '0', the write operation of the control register of timer A is ignored and the read values are all initial (except port mode register PMx and port register Px).

7.3.2 Secondary system clock provides mode control register (OSMC)

The runtime clock of timer A can be selected by WUTMMCK0 bits.

RTCLPC bits are bits that reduce power consumption by stopping unwanted clock functionality. For RTCLPC bit settings, refer to "Chapter 4 Clock Generation Circuits."

The OSMC register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 7-3 The secondary system clock provides the format of the mode control register (OSMC)

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

symbol	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	UTMCK0	0	0	0	0

UTMCK0	Selection of a real-time clock, a run-time clock ($_{fRTC}$) of a 15-bit interval timer and a run clock of timer A
0	Secondary System Clock ($_{fSUB}$) · The secondary system clock is the real-time clock and the runtime clock of the 15-bit interval timer. · A low-speed internal oscillator cannot be selected as the count source for timer A.
1	Low-speed internal oscillator clock ($_{fIL}$) · The low-speed internal oscillator clock is the real-time clock and the runtime clock of the 15-bit interval timer. · A low-speed internal oscillator or sub-system clock can be selected as the count source of timer A.

7.3.3 timer A count register 0 (TA0)

This is a 16-bit register. If this register is written, data is written to the reload register. If you read this register, read the count value. The status of the reload registers and counters varies due to the value of the TSTART bit of the TACR0 register. Refer to "7.4.1 Reload Registers and Counter Overrides" for details.

The TA0 register is set by a 16-bit memory operation instruction. The value of the TA0 register changes to "FFFFH" after the reset signal is generated.

Figure 7-4 Format of timer A count register 0 (TA0)

Address: 40042300H After reset: FFFFH R/W

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TA0																

—	Features	Set Scope
bit15~0	16-bit counter Note1,2	0000H~FFFFH

Note:

1. If you write "1" to the TSTOP bit of the TACR0 register, the 16-bit counter is forced to stop and the count value becomes "FFFFH.
2. If the TCK2~TCK0 bit of the TAMR0 register is not set as '$_{001B}$' and '$_{011B}$' and the TA0 register is '$_{0000H}$', only one request signal is generated to the DMA and the EVENTC immediately after the count is started. However, TAO and TAIO output alternately.

In the event counter mode, the TA0 register value is "0000H", the request signal is generated to DMA and EVENTC only immediately after the counting starts, and the TAO outputs alternately even if the counting period is not specified.

If that value of the TA0 register is greater than or equal to '0001H', a request signal is generate each time the TA underflows.

Note: When accessing the TA0 register, the CPU does not enter the processing of the next instruction and is in the waiting state of the CPU processing. Therefore, when this wait occurs, the number of clocks executed by the instruction increases the number of clocks waiting. The number of read and write waiting clocks when accessing the TA0 register is 1 clock.

7.3.4 timer A control register 0 (TACR0).

The TACR0 register is a register that controls the count and stop of register A and indicates the status of timer A.

The TACR0 register is set by an 8-bit memory operation instruction.

After the reset signal is generated, the value of the TACR0 register changes to '00H'.

Figure 7-5 Timer A controls the format of register 0 (TACR0)

Location: 40042240H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
TACR0	0	0	TUNDF	TEDGF	0	TSTOP	TCSTF	TSTART

TUNDF	Overflow flag for timer A
0	No underflow occurred.
1	Underflow occurred.
[Condition for '0'] · When writing "0" to this bit through the program [Condition for "1"] · When the counter overflows	

TEDGF	Determination Flag for Active Edges
0	No valid edges.
1	There are valid edges.
[Condition for '0'] · When writing "0" to this bit through the program [Condition for "1"] · At the end of the external input (TAIO) effective width measurement in pulse width measurement mode · When you enter the set-up edge of an external input (TAIO) in pulse period measurement mode	

TSTOP	Timer A count force stop note 1
If you write "1" for this bit, force the count to stop. The read value is "0".	

TCSTF	Counter Status Flag Note 2 for Timer A
0	Stop counting.
1	Counting.
[Condition for '0'] · When writing "0" to the TSTART bit (synchronization with count source becomes "0") · When writing "1" to TSTOP bits [Condition for "1"] · When writing "1" to the TSTART bit (synced to "1")	

TSTART	Timer A starts at 2.
0	Stop counting.
1	Start counting.
Start counting by writing "1" to the TSTART bit; Stop counting by writing "0" to the TSTART bit. If the TSTART location "1" is counted, the TCSTF bit is synchronized with the count source to "1" (counting). In addition, after writing "0" to the TSTART bit, the TCSTF bit synchronizes with the count source to "0". For more information, refer to Start and Stop Controls for the 7.5.1 Count.	

Note: 1. If the TSTOP bit is written "1", the TSTART bit and the TCSTF bit are initialized simultaneously.

2. For considerations when using TSTART and TCSTF bits, refer to "Start and Stop Control of 7.5.1 Counts."

7.3.5 Timer AI/O control register 0 (TAIOC0)

The TAIOC0 register is a register that sets the input/output of the timer A. The TAIOC0 register is set by an 8-bit memory operation instruction.

After the reset signal is generated, the value of the TAIOC0 register changes to '00H'.

Figure 7-6 Format of timer AI/O control register 0 (TAIOC0)

Location: 40042241H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
TAIOC0	TIOGT1	TIOGT0	TIPF1	TIPF0	0	TOENA	0	TEDGSEL

TIOGT1	TIOGT0	Count Control Notes 1, 2 for TAIO
0	0	Events are always counted.
0	1	Counts events during the polarity specified by INTP4.
1	0	Events are counted during the polarity specified by the timer output signal.
above		Disable setting.

TIPF1	TIPF0	Selection of TAIO Input Filter
0	0	No filter.
0	1	There is a filter, sampling through fCLK.
1	0	There is a filter, sampling through fCLK/8.
1	1	There is a filter, sampling through fCLK/32.
These bits specify the sampling frequency of the TAIO input filter. Samples the input to the TAIO pin and determines the value as the input if the sample value is the same three times in a row.		

TOENA	Allow for TAO output
0	Disables TAO output (port).
1	Allow TAO output.

TEDGSEL	Input/output polarity switching
Functions vary depending on the mode in which they are run (see Tables 7-4 and 7-5).	

Note:1. Count polarity of events can be selected by RCCPSEL2 bits of the TAISR0 register when the INTP4 or timer output signal is used.

The 2.TIOGT0 bit and TIOGT1 bit are only valid in event counter mode.

Table 7-4 Edge and Polarity Switching of TAIO Input/Output

operation mode	Features
timer mode	Not used (input/output ports).
pulse output mode	0: Output from "H" level (initial level: "H") 1: Output from "L" level (initial level: "L")
Event Counter Mode	0: Count along the rising edge 1: Count down the descent
pulse width measurement mode	0: Measuring "L" level width 1: Measuring "H" level width
pulse period measurement mode	0: Measurements are made between the rising edge of the measurement pulse and the next rising edge 1: The measurement is performed between the falling edge of the measurement pulse and the next falling edge

Table 7-5 Polarity switching of TAO output

operation mode	Features
All mode	0: Output from "L" level (initial level: "L") 1: Output from "H" level (initial level: "H")

7.3.6 timer A control register 0 (TAMR0).

The TAMR0 register is a register that sets the run mode of register A. The TAMR0 register is set by an 8-bit memory operation instruction.

After the reset signal is generated, the value of the TAMR0 register changes to '00H'.

Figure 7-7 Timer A controls the format of register 0 (TAMR0)

Location: 40042242H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
TAMR0	0	TCK2	TCK1	TCK0	TEDGPL	TMOD2	TMOD1	TMOD0

TCK2	TCK1	TCK0	Counter Source Selection Note 1, 2 for Timer A
0	0	0	fCLK
0	0	1	$f_{CLK}/8$
0	1	1	$f_{CLK}/2$
1	0	0	fIL
1	0	1	Events entered by EVENTC
1	1	0	fSUB
Other than above			Disable setting.

TEDGPL	TAIO Edge Polarity Selection Note 5
0	single edge
1	double edge

TMOD2	TMOD1	TMOD0	Select Note 3 for Timer A Operation mode
0	0	0	timer mode
0	0	1	pulse output mode
0	1	0	Event Counter Mode
0	1	1	pulse width measurement mode
1	0	0	pulse period measurement mode
Other than above			Disable setting.

Note:

1. If the event counter mode is selected, the external input (TAIO) is selected as the count source regardless of the setting of TCK0~TCK2 bits.
2. Cannot switch count source during count. If you want to switch the count source, you must switch when both the TSTART and TCSTF bits of the TACR0 register are "0".
3. The run mode can only be changed when stopping the count (both TSTART and TCSTF bits of the TACR0 register are "0") and cannot be changed during the count.
4. To select fIL as the count source, the secondary system clock must be supplied with the WUTMMCK0 position "1" of the OSMC. Register. However, when fSUB is selected as the count source of the real-time clock or 12-bit interval timer, fIL cannot be selected as the count source of timer A.
5. The TEDGPL bit is only valid in event counter mode.
6. The outputs of the TAO pin and the TAIO pin of timer A are initialized by writing the TAMR0 register. Refer to the "Figure 7-6 Timer AI/O Control Register 0 (TAIOC0) Format" for initialization.

7.3.7 Timer A Event Pin Selection Register 0 (TAISR0).

The TAISR0 register is a register that selects a timer that controls the period of the event count in the event counter mode and sets the polarity. The TAISR0 register is set by an 8-bit memory operation instruction.

After the reset signal is generated, the value of the TAISR0 register changes to '00H'.

Figure 7-8 Timer A Event Pin Selection Register 0 (TAISR0) Format

Location: 40042243H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
TAISR0	0	0	0	0	0	RCCPSEL2 Note	RCCPSEL1 Note	RCCPSEL0 Note

RCCPSEL2 Note	The Choice of Output Signal of Timer and Polarity of INTP4
0	Event are counted during that" L" level.
1	Event are counted during that" H" level.

RCCPSEL1 Note	RCCPSEL0 Note	Selection of Output Signal of Timer
0	0	TMIOD1
0	1	TMIOC1
1	0	TO02
1	1	TO03

Note: RCCPSEL0~RCCPSEL2 bits are only valid in the event counter mode.

7.3.8 Port Mode Register x (PMx)

This is a register that sets the port input/output.

When the multiplexed ports (TAIO, TAO, etc.) of the timer output pin are used as the output of the timer, the corresponding bits of the port mode register (PMxx) and port register (Pxx) must "0".

(Example) If P01 is used as the timer output TAIO, the PM01 position "0" of port mode register 0. Position P01 for port register 0 "0.

To use the multiplexed ports of the timer input pins (P01/TAIO, etc.) as input to the timer, the locations of the port mode registers (PMxx) corresponding to each port must The bit of the port register (Pxx) can be "0" or "1".

(Example) If P01 is used as timer input TAIO, the PM01 position "1" of port mode register 0. Position P01 of port register 0 "0" or "1.

The PMxx register is set by an 8-bit memory operation instruction. After the reset signal is generated, the values of these registers become "FFH".

For port mode register formats, refer to "Table 2-2 Product Assigned PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and their bits."

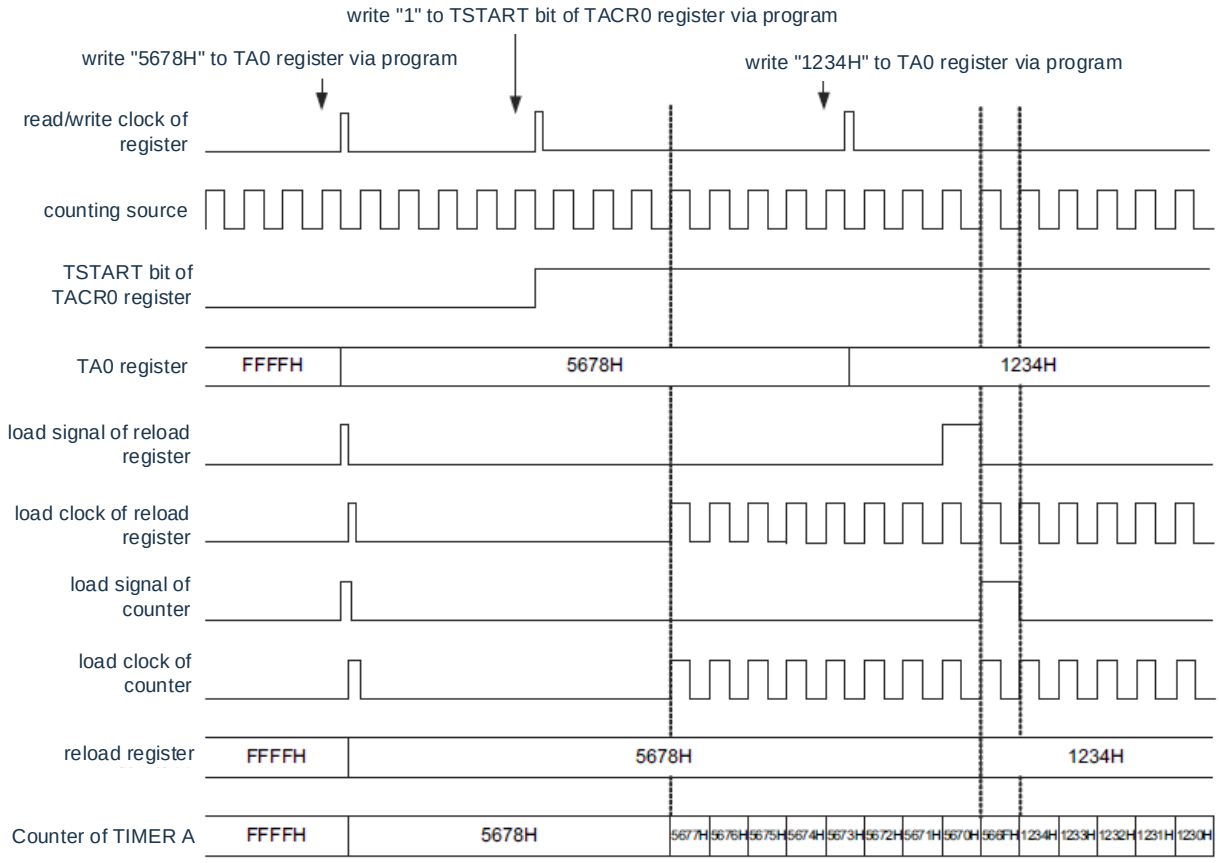
7.4 Operation of timer A

7.4.1 Reload register and counter override

Regardless of the run mode, the rewriting timing of the reload register and counter varies with the value of the TSTART bit of the TACR0 register. When the TSTART bit is "0" (stop counting), the direct write reload register and counter; When the TSTART bit is '1' (Start Count), the write counter is synchronized with the next count source after the write reload register is synchronized with the count source.

The rewriting sequence diagram, determined by the value of the TSTART bit, is shown in Figure 7-9.

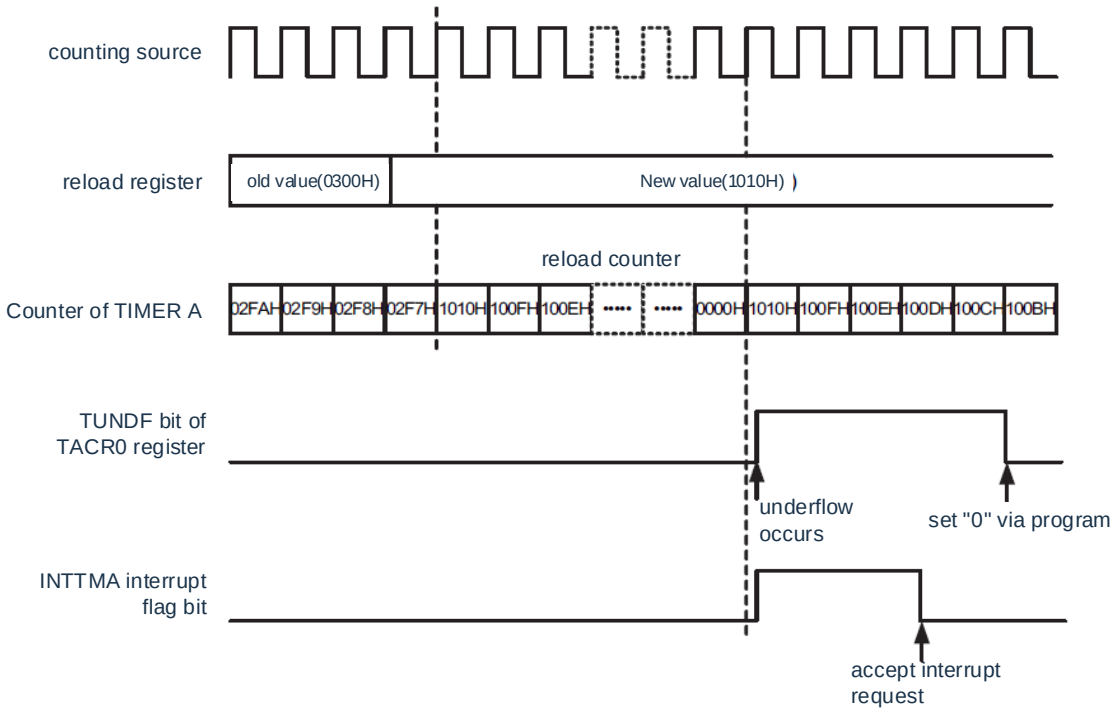
Figure 7-9 Rewriting Time Series Figure Determined by the Value of the TSTART Bit



7.4.2 timer mode

This is the mode of decreasing count by the TCK0~TCK2 bit selection of the TAMR0 register. In timer mode, count value is reduced by 1 whenever a count source is entered, and underflow occurs and an interrupt request is generated. An example of the timer mode operation is shown in Figure 7-10.

Figure 7-1 0 Running example of timer mode



7.4.3 pulse output mode

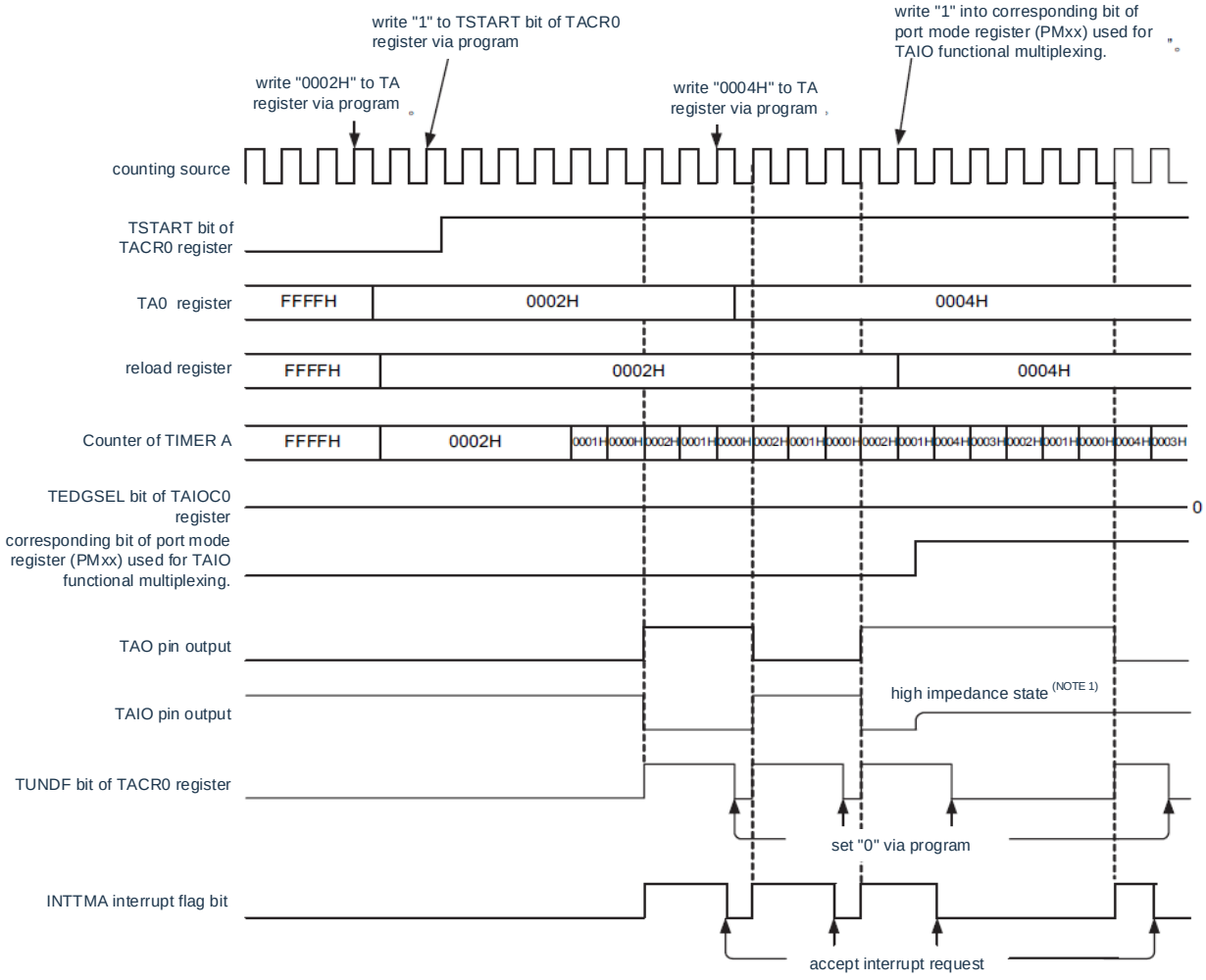
In this mode, the output level of the TAIO pin and the TAO pin is inverted every time the underflow occurs by the counting source selected by the TAMR0 register TCK0~TCK2 bit.

In the pulse output mode, the count value is reduced by 1 each time the count source is input, and if the count value becomes '0000H', underflow occurs and interrupt request occurs.

Pulses can be output from the TAIO pin and the TAO pin, and the output level is inverted each time an underflow occurs. The TAO pin pulse output can be stopped by the TOENA bit of the TAIOC0 register.

In addition, the output level can be selected through the TEDGSEL bit of the TAIOC0 register. An example of operation of the pulse output mode is shown in Figure 7-11.

Figure 7-11 Running example of pulse output mode



NOTE 1: configure to high impedance state via port output enable control of the selected TAIO function.

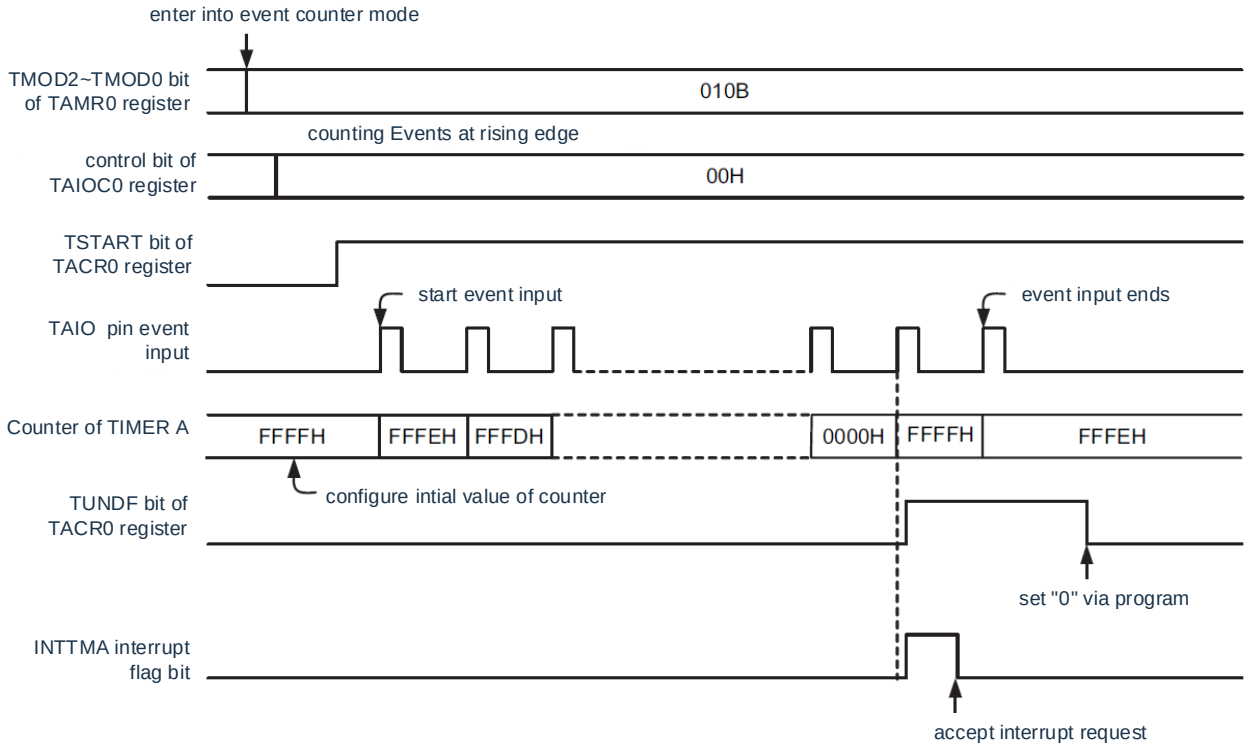
7.4.4 Event Counter Mode

This is the pattern of decreasing counts by the external event signal (count source) entered through the TAIO pin.

Various settings during event counting can be performed through the TIOGT0~TIOGT1 bit and TAISR0 register of TAIOC0 register, and filter function of TAIO input can be specified through TIPF0~TIPF1 bit of TAIOC0 register.

The TAO pin can be alternately output even in event counter mode. To use event counter mode, refer to the "Set-up steps for 7.5.5 TAO and TAIO pins". An example of the Event Counter mode running 1 is shown in Figure 7-12.

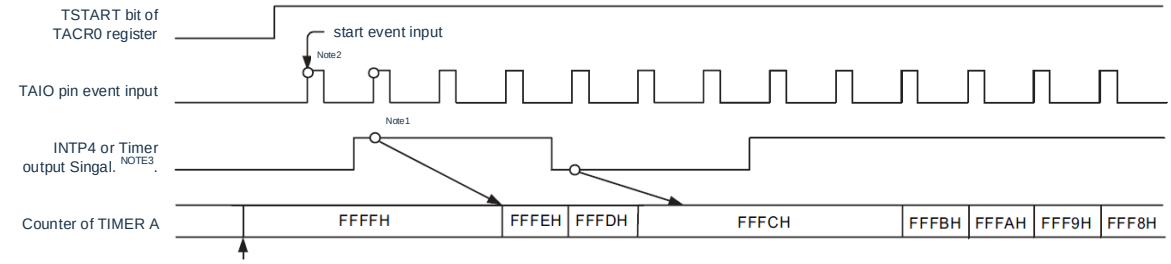
Figure 7-12 Running example of event counter mode



An example of running a specified time count in event counter mode (TIOGT1 and TIOGT0 bits of the TAIOC0 register are "01B" or "10B").

Figure 7-13 Running example of event counter mode

■ example of timing sequence to configure operational mode to following scenario.
TAMR0 register: TMOD2, 1, 0=010B (Event counter mode)
TAIOC0 register: TIOGT1,0=01B(event count during external interrupt pin defined period)
TIPF1, 0=00B (no filter)
TEDGSEL=0 (counting at rising edge)
TAISR0 register: RCCPSEL2=1(counting during H period)



The below precaution note only is relevant to the event counting mode configuration while TIOGT1 and TIOGT0 bit of TAIOC0 register is configured as "01B" or "10B".

NOTE1. To have synchronization control, 2 cycles of counting source clock delay can be reflected before counting execution starts.
2. the 2 counting source clock can start counting based on the state of previous counting stop, initialization shall be done towards internal circuit and start counting after operational configuration. In order to invalid the change of 2 counting source clock after counting starts, TSTOP bit of TACR0 register shall be set to '1'.
3.To timer output singal selected by RCCPSEL1 and RCCPSEL0 bit of TAISR0 register, the pins which are allocated to the timer output pin can not be used as other multiplex function output.

7.4.5 pulse width measurement mode

This is the mode in which the external signal pulse width of the TAIO pin input is measured.

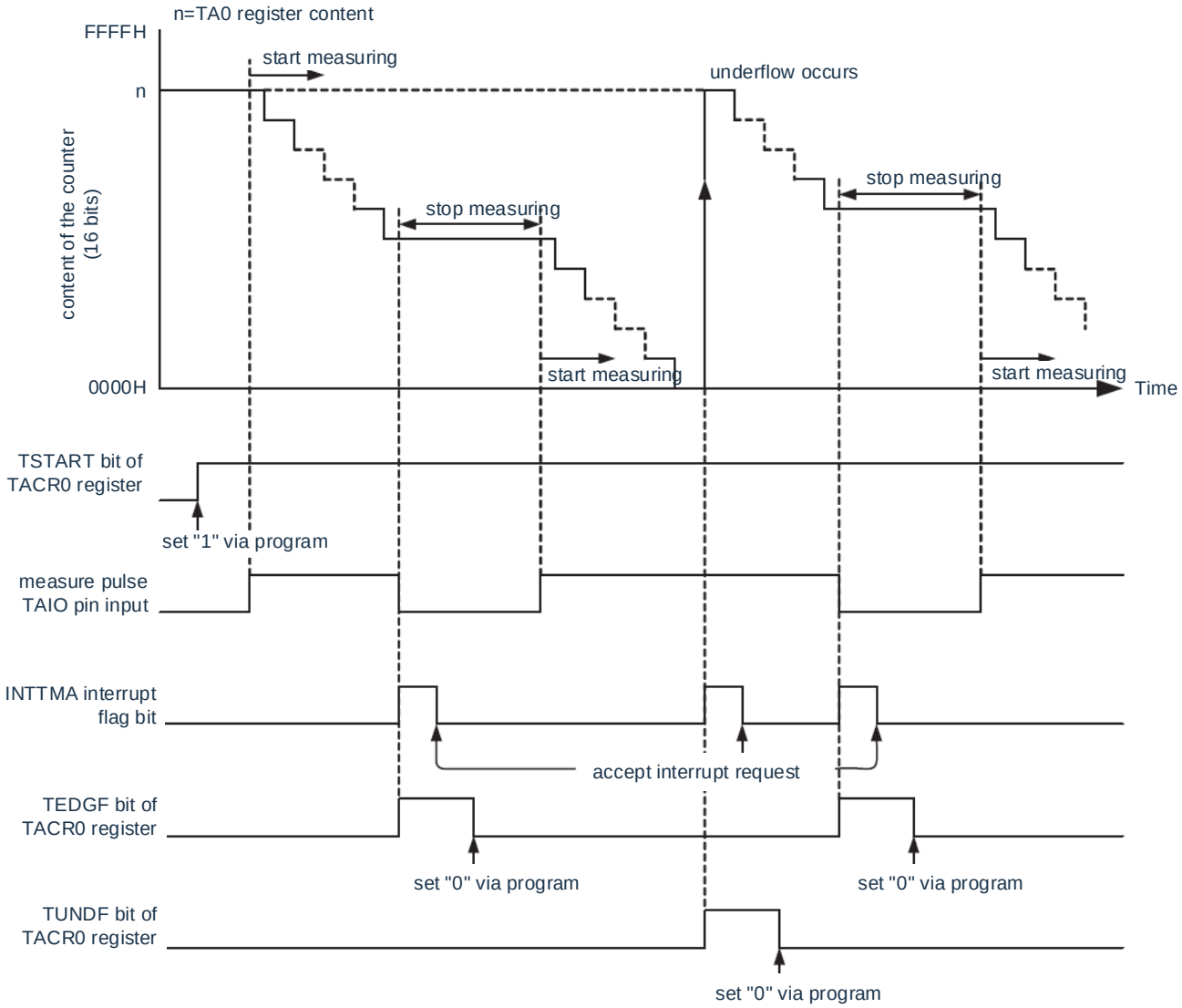
In pulse width measurement mode, if a level specified by the TEDGSEL bit of the TAIOC0 register is input to the TAIO pin, counting is started by the selected count source. If the specified level of the TAIO pin input ends, the counter stops counting, the TEDGF bit of the TACR0 register becomes "1" with a valid edge and generates an interrupt request. Pulse width data is measured by reading the count value when the counter stops counting. If the counter underflows during measurement, the TUNDF bit of the TACR0 register becomes "1" (underflows occur) and generates an interrupt request.

An example of the pulse width measurement mode is shown in Figure 7-14.

When accessing TEDGF and TUNDF bits of the TACR0 register, refer to "Access of the 7.5.2 flag (TEDGF and TUNDF bits of the TACR0 register).

Figure 7-14 Running example of pulse measurement mode

This is the case where the measurement is performed on the "H" level of the measurement pulse (TEDGSEL=1 in the TAIOC0 register).



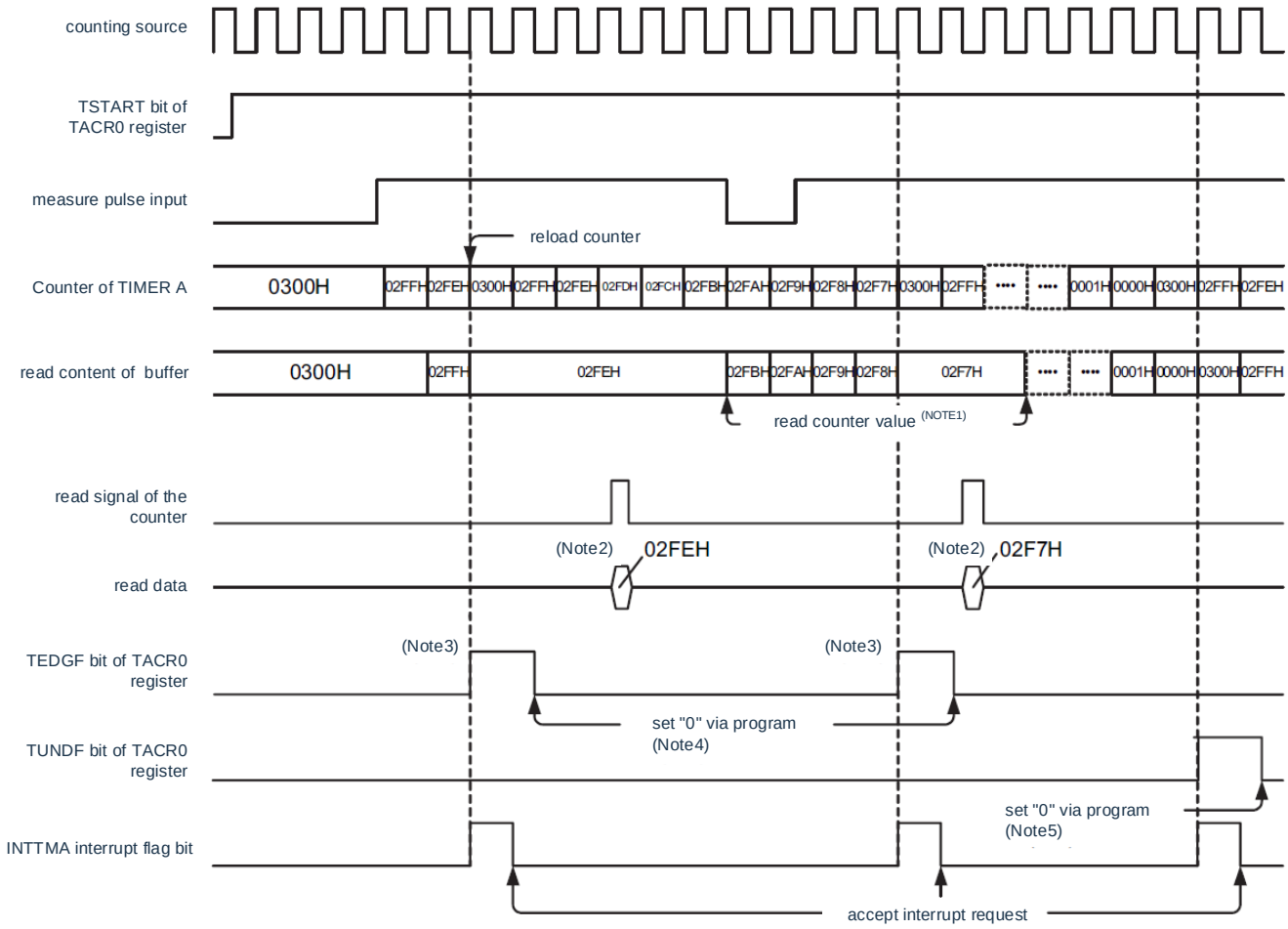
7.4.6 pulse period measurement mode

This is the mode in which the pulse period of the external signal of the TAIO pin input is measured.

The counter performs decrement counting by the counter source selected by the TCK0~TCK2 bit of the TAMR0 register. If the TAIO pin is inputted with a pulse of the TEDGSEL bit of the TAIOC0 register, the count value is transmitted to the read buffer at the rising edge of the counter source. At this time, the TA0 register (read buffer) is read, and the difference between the read value and the reloaded value is the periodic data of the input pulse. Cycle data is held until the read buffer. If the counter underflows, the TUNDF bit of the TACR0 register becomes "1" (underflows occur) and generates an interrupt request. An example of operation of the pulse period measurement mode is shown in Figure 7-15.

Pulses greater than 2 times the count source period must be entered, and the 'L' and 'H' levels entered must be greater than pulses of count source period. If the input pulse period and width do not meet these conditions, the input pulse may be ignored.

Figure 7-15 Running example of pulse period measurement mode



This is the scenario done while TA0 register initial value as "0300H" and TEDGSEL bit of TAIOC0 register set to 0 and measurement done before pulse arises.

Note1.reading TA0 register must be done from the moment TEDGF bit changes to 1 till next valid edge input. Content of the read buffer will be preserved till reading TA0 regsiter, thus, if the TA0 register is not read before the input valid edge, it will remain the measurement result of previous cycle.

2.if reading TA0 register in pulse period measurement mode, the read value is the content of read buffer.

3.TEDGF bit of the TACR0 register will change to 1 (valid edge), if specified edge of external pulse input occurs after the input mesurement pulse valid edge.

4. TEDGF bit of the TACR0 register must be set to 0 via 8 bit operation instruction if program wants to set it to 0.

5. TUNDF bit of the TACR0 register must be set to 0 via 8 bit operation instruction if program wants to set it to 0.

7.4.7 Collaboration with EVENTC

The ability to set events entered by EVENTC as count sources by working with EVENTC.

The TCK0~TCK2 bits of the TAMR0 register are counted at the rising edge of the ELC input. However, EVENTC input does not work in event counter mode.

The EVENTC set-up steps are as follows.

· Steps to Start Running

(1) Set the event output target selection register (ELSELRn) of EVENTC.

(2) Set the running mode of event source.

(3) Setting the mode of timer A.

(4) Start the count of timer A.

(5) Start the source of event occurrence.

· To stop running

(1) Stopping the source of event occurrence.

(2) Stopping the count of timer A.

(3) Set the EVENTC event output target selection register (ELSELRn) to "0"

7.4.8 Output settings for each mode

The status of the TAO and TAIO pins in each mode is shown in Tables 7-6 and 7-7.

Table 7-6 TAO Pin Settings

operation mode	TAIOC0 register		TAO Pin Output
	TOENA bit	TEDGSEL bit	
All mode	1	1	invert output
		0	forward output
	0	0 or 1	Suppress Output

Table 7-7 TAIO Pin Settings

operation mode	TAIOC0 register		Input/Output of TAIO Pin
	PMXX Bitmap	TEDGSEL bit	
timer mode	0 or 1	0 or 1	Input (not used)
pulse output mode	1	0 or 1	Suppress Output (Hi-Z Output)
	0	1	forward output
		0	invert output
Event Counter Mode	1	0 or 1	Enter
pulse width measurement mode			
pulse period measurement mode			

Note This is the bit of the port mode register (PMxx) corresponding to the TAIO functional multiplexing port.

7.5 Precautions when using timer A

7.5.1 Start and stop control of count

· Event Count Mode or Setting Count Source to Non-EVENTC

If the TACR0 register's TSTART bit is written "1" during count stop, the TACR0 register's TCSTF bit is "0" during 3 counts. In addition to the TCSTF bit, the relevant register entry for timer A cannot be accessed before the TCSTF bit becomes "1"note.

If you write "0" to the TSTART bit during the count (stop counting), the TCSTF bit is "1" in 3 count source cycles. Stop counting when the TCSTF bit becomes '0'. In addition to the TCSTF bit, the associated register entry for timer A cannot be accessed before the TCSTF bit becomes "0". Interrupt register must be cleared before changing TSTART bit from '0' to '1'. Refer to "Chapter 25 Interrupt Features" for more information.

Note Related registers for timer A: TA0, TACR0, TAIOC0, TAMR0,TAISR0

· Event Count Mode or Case with Count Source set to EVENTC

If the TSTART bit of the TACR0 register is written "1" during count stop, the TCSTF bit of the TACR0 register is "0" during 2 CPU clock cycles. In addition to the TCSTF bit, the relevant register entry for timer A cannot be accessed before the TCSTF bit becomes "1" note.

If you write "0" to the TSTART bit during the count (stop counting), the TCSTF bit is "1" for 2 CPU clock cycles. Stop counting when the TCSTF bit becomes '0'. In addition to the TCSTF bit, the associated register entry for timer A cannot be accessed before the TCSTF bit becomes "0" note.

Interrupt register must be cleared before changing TSTART bit from '0' to '1'. Refer to "Chapter 25 Interrupt Features" for more information.

Note Related registers for timer A: TA0, TACR0, TAIOC0, TAMR0,TAISR0

7.5.2 Access to flags (TEDGF and TUNDF bits of the TACR0 register)

If you write "0" to the TEDGF and TUNDF bits of the TACR0 register through the program, these bits become "0". However, the value of "1" does not change. If a read-modify-write instruction is used for TACR0 registers, the TEDGF bit and TUNDF position "0" may be mistimed during instruction execution. The TACR0 register must be accessed through the 8-bit memory operation instruction.

7.5.3 Access to a counter register

When writing the TACR0 0 register continuously with TSTART bits and TCSTF bits of the register being "1", at least 3 count source clock cycles must be separated.

7.5.4 Changes in Patterns

The Run Mode Correlation register (TAIOC0, TAMR0, TAISR0) for timer A can only be changed when the TACR0 register has TSTART and TCSTF.

When changing the run mode dependent register of timer A, the values of the TEDGF bit and the TUNDF bit are indefinite. Count must start after writing "0" to the TEDGF bit (no valid edge) and "0" to the TUNDF bit.

7.5.5 Set-up steps for TAO and TAIO pins

After reset, the TAO and TAIO pins are the input ports for multiplexing I/O. To output from the TAO and TAIO pins, you must follow the steps below.

Change Step

(1) Setting mode.

(2) Set the initial value to allow output.

(3) Position "0" of the port register corresponding to the TAO pin and TAIO pin.

(4) Set the bit of the port mode register corresponding to the TAO pin and the TAIO pin to the output mode.

(Export from TAO and TAIO pins)

(5) Start counting (TSTART=1 of TACR0 register).

To enter from the TAIO pin, you must follow these steps to set it up.

(1) Setting mode.

(2) Set the initial value and select the edge.

(3) Set the bit of the port mode register corresponding to the TAIO pin to the input mode.

(Starting with the TAIO pin)

(4) Start counting (TSTART=1 of TAMR0 register).

(5) Wait until the TCSTF bit to the TACR0 register becomes "1" (counting).

(event counter mode only)

(6) Enter an external event from the TAIO pin.

(7) The invalid treatment of the measured value must be performed at the end of the first measurement (the second and subsequent measurements are valid).

(Pulse width measurement mode and pulse period measurement mode only)

7.5.6 When timer A not used

When timer A is not used, the TMOD2~TMOD0 position of the TAMR0 register must be "000B" and the The TOENA location of the TAIOC0 register is "0" (TAO output is disabled).

7.5.7 Stopping execution clock of timer A

It is possible to control the supply or stop of the clock of the timer A through the TMAEN bit of the PER1 register. However, the following SFRs cannot be accessed when the clock of timer A stops, but must be accessed in the state providing the timer A clock.

TA0 register, TACR0 register, TAMR0 register, TAIOC0 register, and TAISR0 register

7.5.8 Configuration steps for deep sleep mode (event counter mode)

To make the event counter mode run in deep sleep mode, you must follow the steps below to transfer to deep sleep mode after providing the clock of timer A.

Configuration Steps

(1) Set the running mode.

(2) Start counting (TSTART=1, TCSTF=1). (3) Stopping providing the clock of timer A.

To stop the event counter mode in deep sleep mode, you must follow these steps to run the stop process.

(1) Providing a clock of timer A.

(2) Stop Count (TSTART=0, TCSTF=0)

7.5.9 Functional limitations in deep sleep mode (event counter mode only)

You cannot use the digital filter feature to make event counter mode run in deep sleep mode.

7.5.10 Forced count stop with TSTOP bit

SFRs below 1 count source cycle memory after the counter is forcibly stopped by the TSTOP bit of the TACR0 register. TA0 register, TACR0 register, and TAMR0 register

7.5.11 digital filter

When a digital filter is used, the timer cannot be started within 5 digital filter clock cycles after setting the TIPF1 bit and TIPF0 bit of TAIOC register.

In addition, in the state of using digital filter, the timer can not start in 5 digital filter clock cycle even if changing TAIOC register TEDGSEL bit.

7.5.12 The Case of Selecting fIL as Count Source

To select fIL as the count source, the secondary system clock must be supplied with the WUTMMCK0 position "1" of the OSMC (). However, when fSUB is selected as the real-time clock or the count source of the 15-bit interval timer, it cannot be selected.

Chapter 8 Timer B

8.1 Function of timer B

Timer B has the following three modes:

- Timer mode:
  - Input capture: Counts are made along the rising edge, the falling edge, or the double edges of the rising edge/falling edge.
  - output comparison function: 'L' level output,'H' level output, or alternate output
- PWM mode: PWM output with arbitrary duty cycle can be performed.
- Phase count mode: It can automatically measure the count value of 2 phase encoder.

8.2 Structure of timer B

The block diagram and the pin structure of the timer B are shown in Figure 8-1 and Table 8-1 respectively.

Figure 8-1: Block diagram of timer B

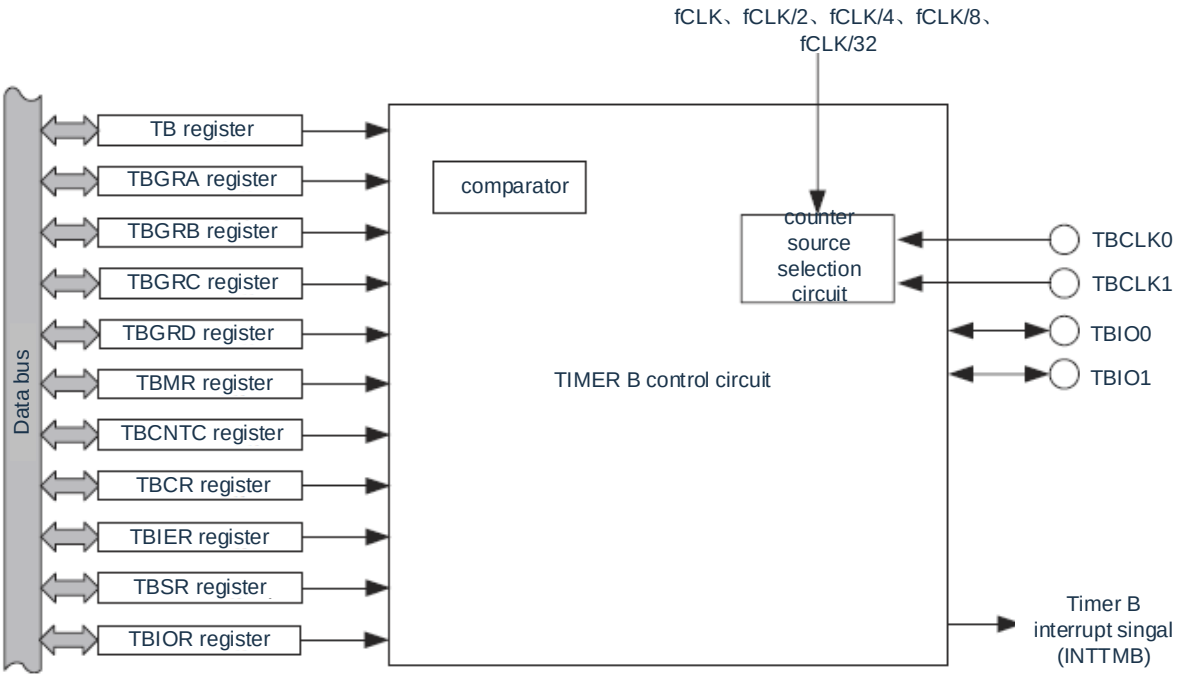


Table 8-1 Pin structure of timer B

name of the feet	Port name for multiplexing	Input/Output	Features
TBCLK0	P00	Enter	· Phase count mode Phase A input · Input of external clock 0 in non-phase count mode
TBCLK1	P01	Enter	· Phase count mode phase B input · Input of external clock 1 in non-phase count mode
TBIO0	P50	Input/Output	· Timer mode (output comparison function) Output of TBGRA output comparison · Timer mode (input capture function) TBGRA Input Captured Input • PWM mode PWM output
TBIO1	P51	Input/Output	· Timer mode (output comparison function) Output of TBGRB output comparison · Timer mode (input capture function) TBGRB Input Captured Input

8.3 Register for control timer B

The register for controlling timer B is shown in Table 8-2.

Table 8-2 Register for control timer B

register name	symbol
Peripheral Enable Register 1	PER1
timer B mode register	TBMR.
timer B count control register	TBCNTC
timer b control register	TBCR
timer b interrupt enable register	TBIER
timer B status register	TBSR
timer BI/O control register	TBIOR
timer B counter	TB
timer b general register a	TBGRA
timer b general register b	TBGRB
timer b general register c	TBGRC
timer b general register d	TBGRD
port register	Pxx
port mode register	PMxx

8.3.1 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.

When you want to use timer B, you must set bit6(TMBEN) to '1'.

The PER1 register is set by an 8-bit memory operation instruction.

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

Figure 8-2 Format of Peripheral Admission Register 1(PER1)

Address: 0x4002081A After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN	TMMEN	DMAEN	PWMPEN	TMCEN Note 2	TMAEN

TMBEN	Control of input clock of timer B is provided
0	Stop provide an input clock. · SFR used by timer B cannot be written. · Timer B is in the reset state.
1	Provides an input clock. · Read and write the SFR used by timer B.

Note: To set timer B, you must first set the TMBEN location "1". When the TMBEN bit is "0", the write operation of the control register of timer B is ignored, and the read values are all initial (except port mode register (PMxx) and port register (Pxx).

8.3.2 timer B mode register (TBMR)

Figure 8-3 Format of timer B mode register (TBMR)

Address: 40042650H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
TBMR	TBSTART	TBELCICE	TBDFCK1	TBDFCK0	TBDFB	TBDFA	TBMDF	TBPWM

TBSTART	Start of TB Count
0	The count is stopped and the PWM output signal (TBIO0 pin) is initialized (PWM mode).
1	Start counting.

TBELCICE	EVENTC Enter the selection note for capture request Note1,2
0	The external input signal 1/digital filter passes the signal 1.
1	Select the event entered by EVENTC (Enter Capture).

TBDFCK1	TBDFCK0	Selection of clocks for digital filter functions Note 1
0	0	$f_{CLK}/32$
0	1	$f_{CLK}/8$
1	0	f_{CLK}
1	1	Clock for TBTCK0~TBTCK2 bit selection of TBCR register

TBDFB	The Choice of Digital Filter Function of TBIO1 Pin
0	There is no digital filter function.
1	It has digital filter function.
When the digital filter function is available, the sampling clock cycle of up to 5 digital filters is needed for	

TBDFA	The Choice of Digital Filter Function of TBIO0 Pin
0	There is no digital filter function.
1	It has digital filter function.
When the digital filter function is available, the sampling clock cycle of up to 5 digital filters is needed for	

TBMDF	Selection of Phase Counting Mode
0	incremental count
1	phase counting mode
When the TBMDF bit is "0", the counter counts the counting source set by the TBTCK0~TBTCK2 bit of the TBCR register. When the TBMDF bit is "1", the counter counts the phase of the TBCLKj pin (j=0, 1) shown in Table 8-15 TB Register Addition and Subtraction Conditions.	

TBPWM	Selection of PWM Mode
0	timer mode
1	PWM mode

Note: 1. You cannot set this bit when the TBSTART bit is "0".

2. To make the EVENTC-entered event (input capture) valid, the TBIO12 position "1" of the TBIOR register and the TBIO11 position "00B".

8.3.3 Timer B count control register (TBCNTC)

A TBCNTC register is used in the phase count mode to set the count condition of the phase count mode.

Figure 8-4 Format of timer B count control register (TBCNTC)

Location: 40042651H. After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
TBCNTC	CNTEN7	CNTEN6	CNTEN5	CNTEN4	CNTEN3	CNTEN2	CNTEN1	CNTEN0

CNTEN7	Allowed for Count 7
0	invalid
1	incremental count The TBCLK0 input is an 'L' level and at the rising edge of the TBCLK1 input

CNTEN6	Allowed 6 for Count
0	invalid
1	incremental count TBCLK1 input is an 'H' level and at the rising edge of the TBCLK0 input

CNTEN5	Allowed 5 for Count
0	invalid
1	incremental count The TBCLK0 input is an 'H' level and at the descent edge of the TBCLK1 input

CNTEN4	Allowed 4 for Count
0	invalid
1	incremental count The TBCLK1 input is an 'L' level and at the descent edge of the TBCLK0 input

CNTEN3	Allowed 3 for Count
0	invalid
1	decremental count The TBCLK1 input is an 'H' level and at the descent edge of the TBCLK0 input

CNTEN2	Allowed 2 for count
0	invalid
1	decremental count The TBCLK0 input is an 'L' level and at the descent edge of the TBCLK1 input

CNTEN1	Allowed 1 for count
0	invalid
1	decremental count The TBCLK1 input is an 'L' level and is used when the TBCLK0 input is rising

CNTEN0	Allowed 0 for count
0	invalid
1	decremental count The TBCLK0 input is an 'H' level and when the TBCLK1 input is rising

8.3.4 Timer B Control Register (TBCR)

The TBMR register must be written with the TBSTART bit of the TBCR register in the state of '0'.

Figure 8-5 Timer B Control Register (TBCR) Format

Location: 40042652H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
TBCR	0	TBCCLR1	TBCCLR0	TBCKEG1	TBCKEG0	TBTCK2	TBTCK1	TBTCK0

TBCCLR1	TBCCLR0	Purging source selection for TB registers
0	0	Do not clear.
0	1	Clears when an input capture or comparison match for TBGRA.
1	0	Clears when an input capture or comparison match for TBGRB.
Oher than above		Disable setting.

TBCKEG1	TBCKEG0	Effective edge selection for external clock Notes 1, 2
0	0	Count along the rising edge.
0	1	Count along the descent.
1	0	Count along the double edges of the rising/falling edges.
Oher than above		Disable setting.

TBTCK2	TBTCK1	TBTCK0	Count Source Selection Note 1
0	0	0	f_{CLK}
0	0	1	$f_{CLK}/2$
0	1	0	$f_{CLK}/4$
0	1	1	$f_{CLK}/8$
1	0	0	$f_{CLK}/32$
1	0	1	Input for TBCLK0
1	1	1	Input for TBCLK1
Oher than above			Disable setting.

Note:1. In phase count mode, the settings of TBTCK0~TBTCK2 bits, TBCKEG0 bits and TBCKEG1 bits are invalid, so the operation of the preferential phase count mode is invalid.

The 2.TBCKEG0 bit and TBCKEG1 bit are valid when the TBTCK0~TBTCK2 bit is set to an external clock (TBCLK0, TBCLK1), otherwise they are invalid.

8.3.5 Timer B interrupt enable register (TBIER)

Figure 8-6 Timer B interrupts the format of the allow register (TBIER)

Location: 400426 53H. After reset: 00H R/W

symbol	6	5	4	3	2	1	0
TBIER	0	0	0	0	TBOVIE	TBUDIE	TBIMIEB	TBIMIEA

TBOVIE	Permission for overflow interrupts
0	Interrupts due to TBOVF bits are prohibited.
1	The interrupt due to the TBOVF bit is valid.

TBUDIE	Permission for overflow interrupts
0	Interrupts due to TBUDF bits are prohibited.
1	The interrupt due to the TBUDF bit is valid.

TBIMIEB	Input capture/compare matching interrupt allowed B
0	Interrupts due to TBIMFB bits are prohibited.
1	The interrupt due to the TBIMFB bit is valid.

TBIMIEA	Enter Snap/compare matching interrupt allowed A
0	Interrupts due to TBIMFA bits are prohibited.
1	The interrupt due to the TBIMFA bit is valid.

Notes: Bit of TBIMFA, TBIMFB, TBUDF, TBOVF:TBSR register

8.3.6 Timer B Status Register (TBSR)

Figure 8-7 Format of timer B status register (TBSR)

Address: 40042654H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
TBSR	0	0	0	TBDIRF	TBOVF	TBUDF	TBIMFB	TBIMFA

TBDIRF	Count Direction Flag
0	The TB register performs decrement counting.
1	The TB register performs incremental count.

TBOVF	Overflow Flag Note 1
[Condition for '0']Read and write "0" Note 2. [Condition for "1"] Refer to Table 8-3 Conditions marked "1".	

TBUDF	overflow mark
[Condition for '0']Read and write "0" Note 2. [Condition for "1"] Refer to Table 8-3 Conditions marked "1".	

TBIMFB	Enter capture/compare match flag B
[Condition for '0']Read and write "0"Notes 2, 3. [Condition for "1"] Refer to Table 8-3 Conditions marked "1".	

TBIMFA	Enter capture/compare match flag A
[Condition for '0']Read and write "0"Notes 2, 3. [Condition for "1"] Refer to Table 8-3 Conditions marked "1".	

Note:

1. The TBOVF bit changes to '1' when the counter value of timer B changes from 'FFFFH' to '0000H. In addition, the TBCCLR0 bit and the TBCCLR1 bit of the TBCR register change the count value of timer B from FFFFH to 0000H.
2. The results are as follows:
   · This bit does not change when writing "1".
   · If the read value is "0", even if the same bit is written "0" (from "0" to "1").
   · If the read value is "1", the bit becomes "0" if the same bit is given "0".

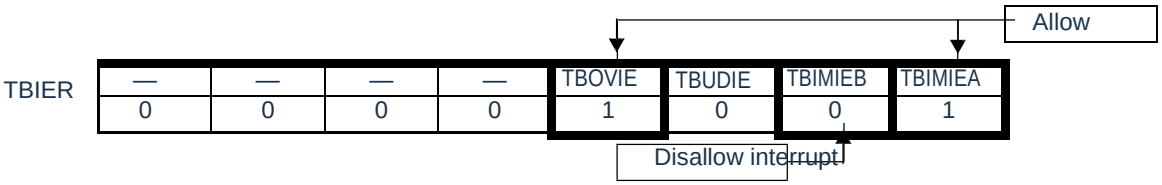
However, if the interrupt is set to 0 by timer B interrupt enable register TBIER, it must be set 0.

a) The object status flag must be written "0" after setting timer B interrupt permit register (TBIER) to "00H".

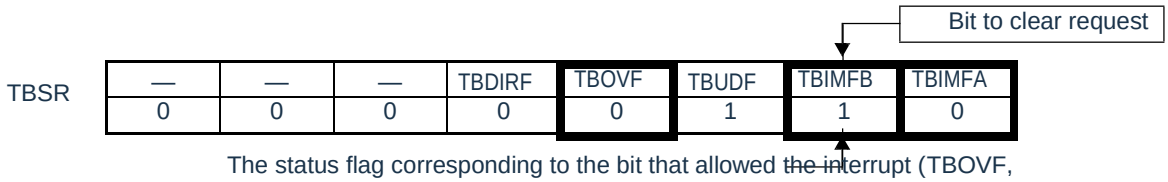
b) When timer B interrupt enable register (TBIER) has bit of ' 1 ' and interrupt source status flag of '0 ' allowed by this bit.

(Example) Cleaning TBIMFB in a state where TBIMIEA and TBOVIE are interrupts allowed and TBIMIEB is interrupts prohibited

- Timer B interrupts the state of the Permit Register (TBIER)



- Status of timer B status register (TBSR)

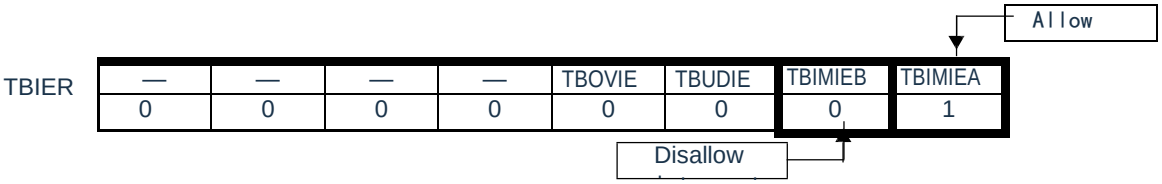


The status flag corresponding to the bit that allowed the interrupt (TBOVF, TBIMFA) is "0", so you must write "0" to TBIMFB.

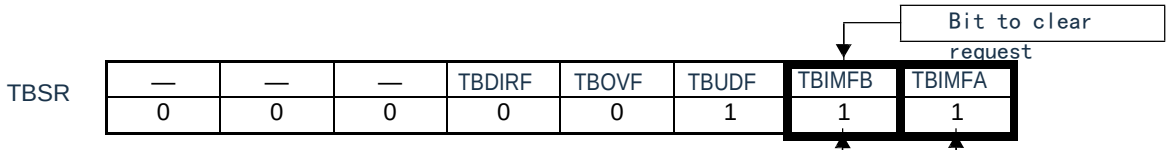
c) When timer B interrupt enable register (TBIER) has bit of ' 1' allowed and interrupt source state flag of '1' allowed by this bit, must write '0'.

(Example) When TBIMIEA is Interrupt Allowed and TBIMIEB is Interrupt Disabled to clear TBIMFB

- Timer B interrupts the state of the Permit Register (TBIER)



- Status of timer B status register (TBSR)



The status flag (TBIMFA) corresponding to the bit that allowed the interrupt is the "1", so you must write "0" to both TBIMFA and TBIMFB.

2. When using DMA, the TBIMFA bit and the TBIMFB bit become "1" after the end of DMA transfer.

Table 8-3 Conditions for each mark to be "1"

bit symbol	Timer Mode Note 1		PWM mode
	Input capture	output comparison function	
TBOVF	When TB overflows		
TBUDF	When TB underflows (phase count mode only)		
TBIMFB	Input Edge for TBIO1 Pin Note 2	When TB and TBGRB have the same value	
TBIMFA	Input Edge for TBIO0 Pin Note 2	When TB and TBGRA have the same value	

Note: 1. The phase count mode is a counting method of a timer B counter register, which can use the timer mode and PWM mode.

2. This is the edge selected by the TBIOj0 bit and TBIOj1 bit (j=0,1) of the TBIOR register.

8.3.7 Timer BI/O control register (TBIOR)

Figure 8-8 The format of the timer BI/O control register (TBIOR)

Address: 00H R/W after 40042655H reset

	7	6	5	4	3	2	1	0
TBIOR	TBBUFB	TBIOB2	TBIOB1	TBIOB0	TBBUFA	TBIOA2	TBIOA1	TBIOA0

TBBUFB	Selection of TBGRD Register Function
0	Buffer register not used as TBGRB register.
1	Buffer register used as TBGRB register.

TBIOB2	Select TBGRB Mode Notes 1, 2
0	output comparison function
1	Input capture function

TBIOB1	TBIOB0	TBGRB control
0	0	Prevents comparing matching pin outputs.
0	1	Output "L" level.
1	0	Output "H" level.
1	1	Alternate output.
The output comparison function is used to compare the output of the TB register and the TBGRB register.		

TBIOB1	TBIOB0	TBGRB control
0	0	TBIO1's rising edge
0	1	TBIO1's descent edge
1	0	TBIO1's Double Frontier
Other than above		Disable setting.
Capture the contents of the TB register to the TBGRB with input capture.		

TBBUFA	Selection of TBGRC Register Function
0	Buffer register not used as TBGRA register.
1	Buffer register used as TBGRA register.

TBIOA2	Select TBGRA Mode Notes 1, 2
0	output comparison function
1	Input capture function

TBIOA1	TBIOA0	TBGRA control
0	0	Prevents comparing matching pin outputs.
0	1	Output "L" level.
1	0	Output "H" level.
1	1	Alternate output.
The output comparison function is used to compare the output of the TB register and the TBGRA register.		

TBIOA1	TBIOA0	TBGRA control
0	0	TBIO0's rising edge
0	1	TBIO0's descent edge
1	0	TBIO0's Double Frontier
Othan than above		Disable setting.
Capture the contents of the TB register to the TBGRA with input capture.		

Note:

1. The TBIOj2 register is used as an input capture register when the TBGRj bits (j=A, B) are "1".
2. The TBGRj register is used as a comparison matching register when the TBIOj2 bits (j=A, B) are "0". The TBIOj0 bit and the TBIOj1 bit are set after reset and the following levels are output from the TBIOj pin before the first comparison match occurs:
   When TBIOj1, TBIOj0=01B, the "H" level is output.
   When TBIOj1, TBIOj0=10B, the "L" level is output.
   When TBIOj1, TBIOj0=11B, the "L" level is output.

The TBIOR register controls the pin of the input/output in the timer mode. Invalid in PWM mode. The TBIOR register must be set under the state of stopping the count (TBSTART=0 of the TBMR register).

8.3.8 Timer B Counter (TB)

The TB register is connected to the CPU by a 16-bit internal bus, so it must be accessed in 16-bit units. The TB register may be capable of incremental count, free run, periodic count, or external event count. The TB register can be cleared "0000H" by comparing and matching with the corresponding TBGRA register and TBGRB register or capturing the TBGRA register and TBGRB register.

When the TB register overflows ('FFFFH '0000H'), the TBOVF bit of the TBSR register becomes '1' When the TB register underflows ('0000H 'FFFFH'), the TBUDF bit of the TBSR register becomes '1'

Figure 8-9 Format of timer B counter (TB)

Location: 40042656H After reset: 0000H R/W

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

—	Features	Set Scope
bit15~0	Increase/decrease counts are performed in phase count mode and increase counts are performed in other modes.	0000H~FFFFH

8.3.9 timer b general register a, b, c, d (TBGRA, TBGRB, TBGRC, TBGRD)

The TBGRA register and the TBGRB register are 16-bit readable and writable registers, which have the functions of output comparison register and input capture register. Function conversion is performed through the TBIOR register.

When used as the output comparison register, the values of the TBGRA register and the TBGRB register are always compared to the values of the TB register. If the values are the same (comparison match), the TBIMFA bit or TBIMFB bit of the TBSR register becomes "1". The output can be compared through the TBIOR register setting.

When used as an input capture register, the value of the TB register is saved after detecting an external input capture signal. At this point, the TBSR register's

The TBIMFA bit or TBIMFB bit becomes'1'. The detection edge of the input capture signal is selected through the TBIOR register.

TBGRC register and TBGRD register can also be used as buffer register of TBGRA register and TBGRB register respectively, and can select this function through TBBUFA bit and TBBUFB bit of TBIOR register.

For example, if the TBGRA register is set as the output comparison register and the TBGRC register is set as the buffer register for the TBGRA register, the TBGRC register is transferred to the TBGRA register.

If the TBGRA register is set as the input capture register and the TBGRC register is set as the buffer register of the TBGRA register, the TB register and TBGRA register are transferred to the TBGRA register and TBGRC register respectively.

It can read and write TBGRA, TBGRB, TBGRC, TBGRD register in 16 bit.

FIG. 8-10 Timer B Format of Universal Registers A, B, C, D (TBGRA, TBGRB, TBGRC,TBGRD)

Address: 40042658H (TBGRA), 4004265AH (TBGRB), 4004260H (TBGRC), 40042662H (TBGRD) After reset: FFFFH R/W

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

—	Features
bit15~0	Vary depending on timer mode and function. TBGRA, TBGRB, TBGRC, TBGRD register functions are shown in Table 8-4.

Notes: i=A, B, C, D

Table 8-4 TBGRA, TBGRB, TBGRC, TBGRD register features

Patterns and Features	register	Settings	Features
input capture	TBGRA	TBIOR (TBIOA2=1) TBMR. (TBPWM=0)	Enter a capture register (holds the value of the TB register)
	TBGRB	TBIOR (TBIOB2=1) TBMR. (TBPWM=0)	Enter a capture register (holds the value of the TB register)
output comparison	TBGRA	TBIOR (TBIOA2=0) TBMR. (TBPWM=0)	Output comparison register (stored and TB register comparison value, when comparison matches TBIO0 output setting)
	TBGRB	TBIOR (TBIOB2=0) TBMR. (TBPWM=0)	Output comparison register (stored and TB register comparison value, when comparison matches TBIO1 output setting)
PWM	TBGRA	TBMR. (TBPWM=1)	Output comparison register (TBIO0 output "H" level at comparison match)
	TBGRB		Output comparison register (TBIO0 output "L" level at comparison match)
common	TBGRC	TBIOR (TBBUFA=0)	Not used.
	TBGRD	TBIOR (TBBUFB=0)	Not used.
	TBGRC	TBIOR (TBBUFA=1)	Buffer register for TBGRA (and TBGRA for transfer) When TBIOA2=1 The previous input capture value is taken from the TBGRA by entering the capture signal. When TBIOA2=0 The next comparison expectation value is transferred to the TBGRA by comparison matching of TB and TBGRA.
	TBGRD	TBIOR (TBBUFB=1)	Buffer register for TBGRB (and TBGRB for transfer) When TBIOB2=1 The previous input capture value is taken from the TBGRB by entering the capture signal. When TBIOB2=0 The next comparison expectation value is transferred to the TBGRB by comparison matching of TB and TBGRB.

Note: If the TBTCK2~TBTCK0 position of the TBCR register is '000B' ($_{fCLK}$) and the comparison value is '0000H', the request signal is generated to DMA and EVENTC only 11 times. If that compare value is greater than or equal to" 0001H", a request signal is generate each time the compare matches.

8.3.10 port register and port mode register

When using the multiplexed port of the timer output pin as the output of the timer, the bit of the port mode register (PMxx) and the location of the port register (Pxx) must

(Example) When P50/TBIO0 is used as the timer output, the PM50 position of port mode register 5 is "0". Position P50 of port register 5 at "0.

When using the multiplexed port of the timer input pin as the input of the timer, the location of the port mode register (PMxx) corresponding to each port must be "1" The bit of the port register (Pxx) can be "0" or "1".

(Example) the case where P50/TBIO0 is used as timer input

Position PM50 of port mode register 5 at "1.

Position P50 of port register 5 at "0" or "1.

Refer to "2.3.1 Port Mode Register (PMxx)" 2.3.2 Port Register (Pxx) and "2.3.6 Port Mode Control Register (PMCxx)" for details.

The set port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx) differ by product. Refer to "Register settings when 2.5 uses the multiplexing feature" for details.

8.4 Operation of timer B

8.4.1 Common Issues Concerning Multiple Models and Functions

(1) count source

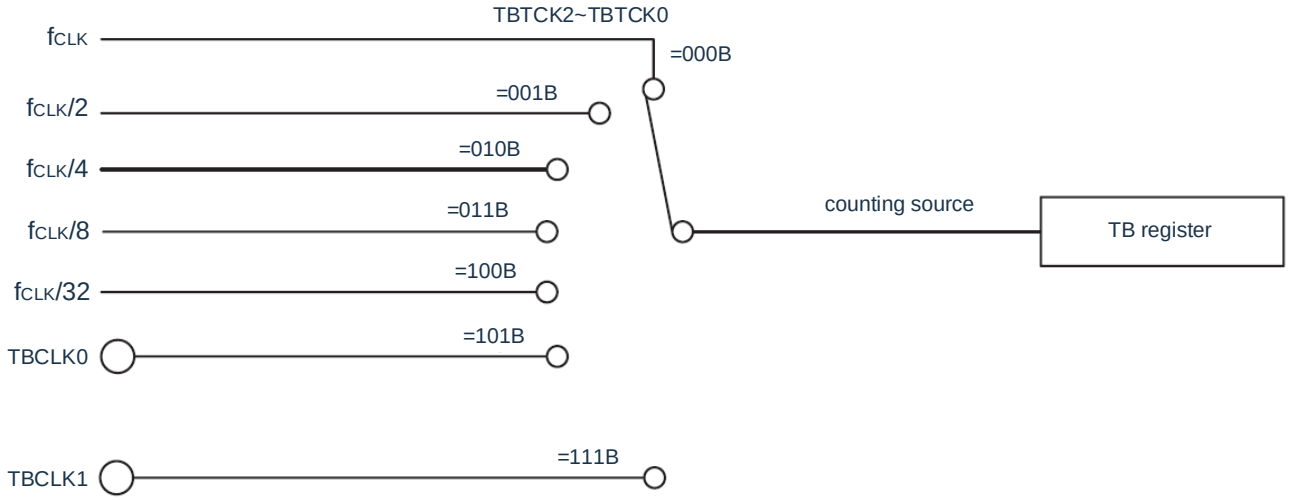
The selection of the count source and the block diagram are shown in Tables 8-5 and Figures 8-12, respectively.

When phase count mode is selected, the settings of TBTCK0~TBTCK2 bit, TBCKEG0 bit, and TBCKEG1 bit of the TBCR register are invalid.

Table 8-5 Selection of Counting Source

count source	selection method
fCLK, fCLK/2, fCLK/4, fCLK/8, fCLK/32	The counting source is selected by the TBTCK0~TBTCK2 bits of the TBCR register.
TBCLK0 Pin and TBCLK1 External Input Signal for Pins	The TBTCK2~TBTCK0 bits of the TBCR register are "101B" (TBCLK0 input) or "111B" (TBCLK1 input). Valid edges are selected through the TBCKEG0 and TBCKEG1 bits of the TBCR register. The corresponding bit of the port mode register is "1" (input mode).

Figure 8-12 Block diagram of count source



Remark: TBTCLK2~TBTCLK0: TBCR register bits

The external clock input by the TBCLKj pin (j=0,1) must have at least 3 clock cycles of runtime ($_{fCLK}$) of timer B.

(2) buffer operation

The TBGRC register and the TBGRD register can be set as the buffer register of the TBGRA register and the TBGRB register respectively through the TBBUFA bit and the TBBUFB bit of the TBIOR register.

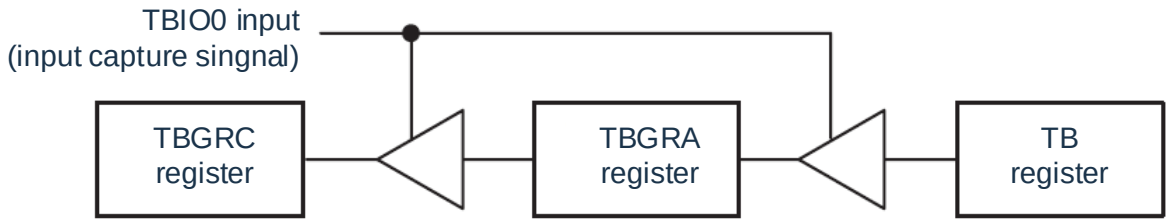
- TBGRA buffer register: TBGRC register
- TBGRB's buffer register: TBGRD register buffer runs differently depending on the timer mode. Buffer operation for each mode is shown in Table 8-6, with a slow input capture function and an output comparison function

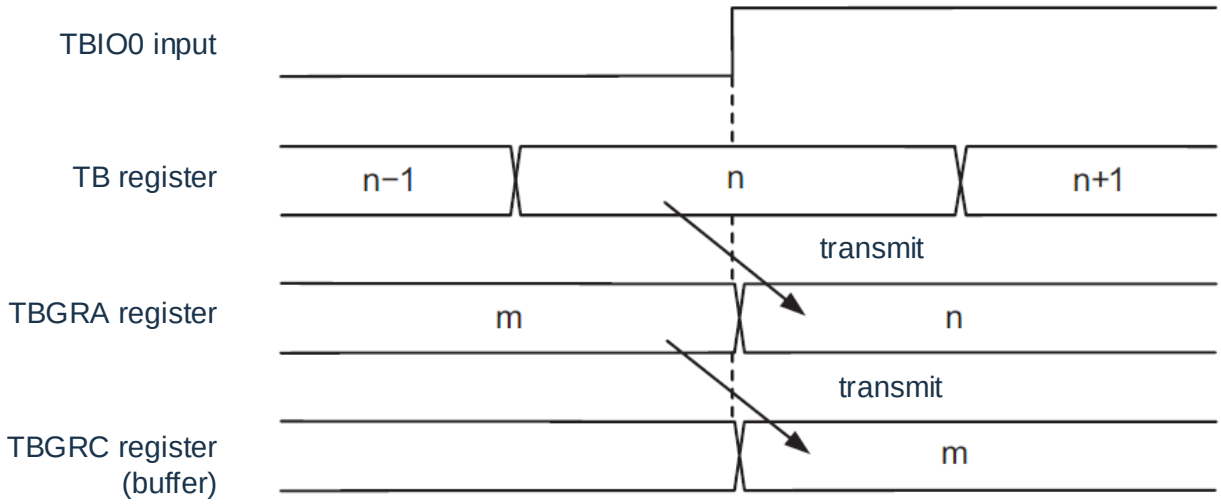
The punch operation is shown in Figure 8-13 and Figure 8-14 respectively.

Table 8-6 Buffer operation by mode

Features and Modes	Transfer Sequence	Transferred register
Input capture	input of input capture signal	The contents of the TBGRA (TBGRB) register are transferred to the buffer register.
output comparison function	TB register and TBGRA (TBGRB) Comparison matching of registers	Transfer the contents of the buffer register to the TBGRA (TBGRB) register.
PWM mode		

Figure 8-13 Buffer Operation for Input Capture



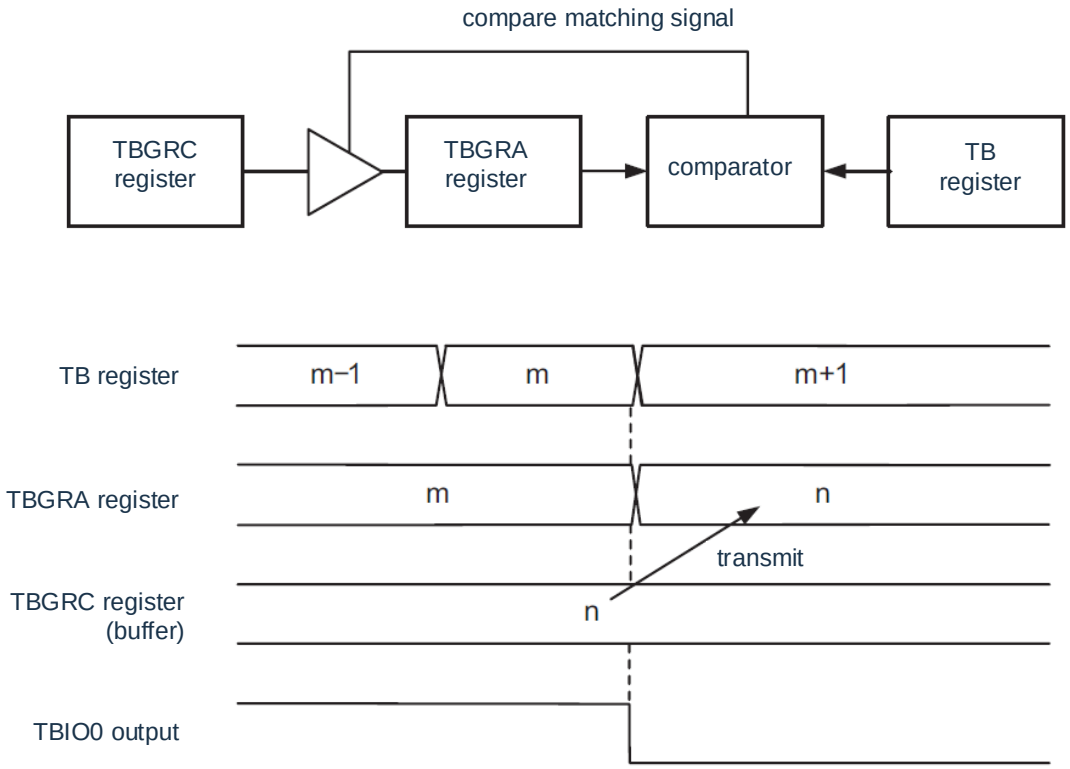


above diagram condition as following:

・TBBUFA bit of TBIOR register is 1 (TBGRC register is the buffer register of TBGRA)

・TBIOA~TBIOA0 bit of TBIOR register is "100B" (input capture at rising edge)

Figure 8-14 Buffer operation for output comparison function



above diagram condition as following:

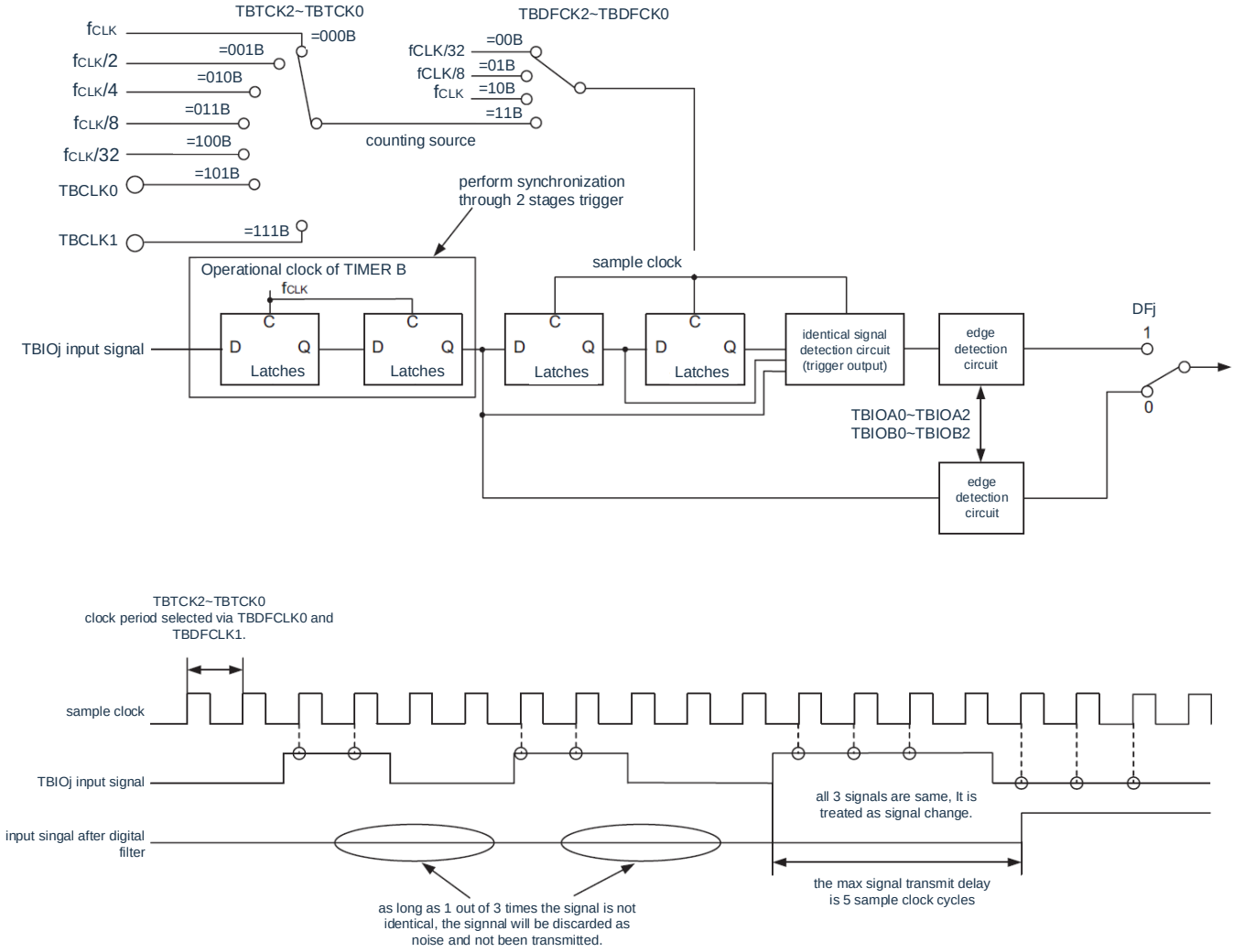
- TBBUFA bit of TBIOR register is 1 (TBGRC register is the buffer register of TBGRA)
- TBIOA2~TBIOA0 bit of TBIOR register as "001B" (while compare matching, output "L" voltage level).

(3) digital filter

The TBIOj input (j=0, 1) is sampled and if that signal is the same 3 times, the level is determined. The function of the digital filter and the sampling clock must be selected through the TBMR register.

The block diagram of the digital filter is shown in Figure 8-15.

Figure 8-15 Block diagram of digital filter



Note j=0, 1

TBTCK2~TBTCK0: bits of TBCR register
TBDFCK0，TBDFCK1，TBDFA，TBDFB: bits of TBMR register
TBIOA0~TBIOA2，TBIOB0~TBIOB2?: bits of TBIOR register

(4) Events entered from EVENTC

The event entered through the EVENTC, timer B makes the input capture run B. At this point, the TBIMFB bit of the TBSR register is "1". To use this feature, you must select the input capture function for timer mode/phase count mode and position "1" for the TBMR register. This feature is not valid in other modes (output comparison function of timer mode/phase count mode and PWM mode).

Configuration Steps

(a) Set the EVENTC event link target to timer B.

(b) Position "1" for the TBELCICE of the TBMR register.

(5) Events exported to EVENTC

Events output to EVENTC through the TBIMFA and TBIMFB bits are shown in Tables 8-7 and 8-8 respectively.

Table 8-7 Events output to EVENTC via TBIMFA bit

Features and Modes	EVENTC Source
Input capture (TBPWM=0, TBIO02=1)	TBIO0 edge detection via TBIOA0 and TBIOA1 bits
output comparison function (TBPWM=0, TBIO02=0)	Comparison and matching of TB register and TBGRA register
PWM mode (TBPWM=1)	Comparison and matching of TB register and TBGRA register

Note: TBPWM:bit of TBMR register
Bit of TBIOA0, TBIOA1, TBIOA2:TBIOR register

Table 8-8 Events output to EVENTC via TBIMFB bit

Features and Modes	EVENTC Source
Input capture (TBPWM=0, TBIO12=1)	TBIO1 edge detection via TBIOB0 and TBIOB1 bits
output comparison function (TBPWM=0, TBIO12=0)	Comparison and matching of TB register and TBGRB register
PWM mode (TBPWM=1)	Comparison and matching of TB register and TBGRB register

Note: TBPWM:bit of TBMR register
Bit of TBIOB0, TBIOB1, TBIOB2:TBIOR register

8.4.2 timer mode (input capture function)

The TB register value can be transferred to the TBGRA register and the TBGRB register after the input edge of the input capture/output comparison pin (TBIO0, TBIO1) is detected. Detection edges can be selected from the rising, falling, and double edges. It can measure pulse width and pulse period by using input capture function.

The specifications for the input capture functions are shown in Table 8-9.

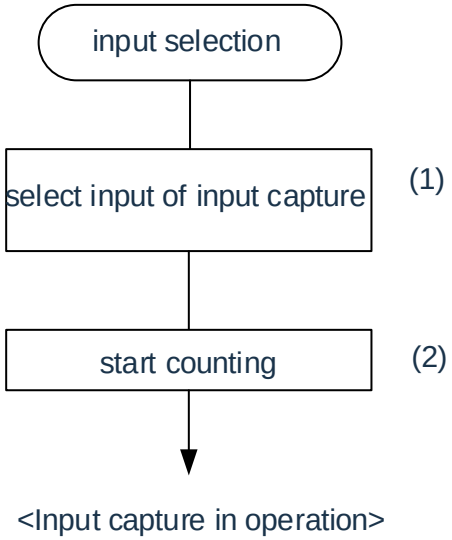
Table 8-9 Enter the specifications for the capture function

Project	Specifications
count source	fCLK, fCLK/2, fCLK/4, fCLK/8, fCLK/32 External input signal for TBCLK0 and TBCLK1 pins (program selection of effective edges)
Count	incremental count
counting period	When the TBCR register has a TBCCLR1 bit and a TBCCLR0 bit of "00B" (free running) 1/fk 65536fk: Frequency of the count source
Count Start Condition	Write "1" to the TBSTART bit of the TBMR register.
Count stop condition	Write "0" to the TBSTART bit of the TBMR register.
Generation sequence of interrupt requests	· Input capture (valid edges for TBIO0 and TBIO1 pin inputs) · Overspill of TB register
TBIO0 Pin and TBIO1 Functions of pins	I/O port or input captured input (select by pin)
TBCLK0 Pin and The Function of TBCLK1 Pins	I/O port or external clock input
read timer	If you read the TB register, you can read the count value.
write timer	Can write TB register.
Select Features	· Enter the selection of input pins for the capture 1 or 2 pins in the TBIO0 and TBIO1 pins · Selection of input valid edges for input capture rising, falling or double · Time sequence in which the TB register is set "0000H" Overspill or Input Capture · Buffer Run (refer to "8.4.1 (2) Buffer Run") · Digital filter (reference to "8.4.1 (3) Digital filter") · Input capture run via EVENTC's event input signal (input capture)

(1) Enter an example of set-up steps for a capture run

Enter an example of the set-up steps for the capture run as shown in Figure 8-16.

Figure 8-16 Enter an example of set-up steps for a capture run



(1) must configure TBGRj register(j=A,B) as input capture register via TBIOR regsiter, and select capture signal input edge from rising, falling and double edge.

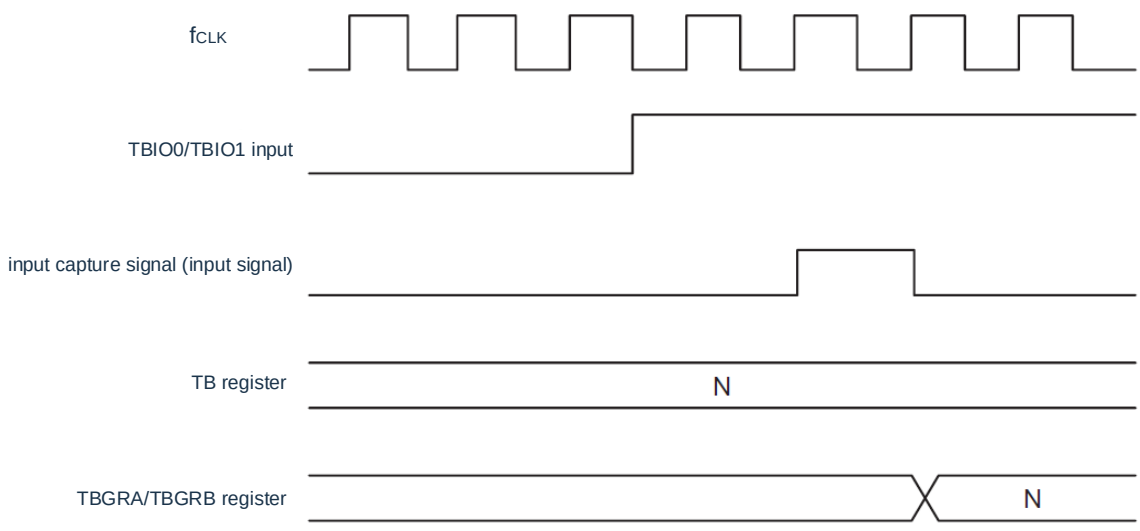
(2) must set TBSTART bit of TBMR register to 1, and start counting of TB register.

(2) Time sequence of input capture signal

The input captured can be selected by setting the TBIOR register to select the rising, falling, or double edges. The timing of the input signal captured is shown in Figure 8-17.

In the case of one edge, the pulse width of input signal captured must be at least 1.5 f_{CLK}. In case of a double edge, the input signal captured must have a pulse width of at least 2.5 fCLK.

Figure 8-17 Input captured input signal timing



(3) running example

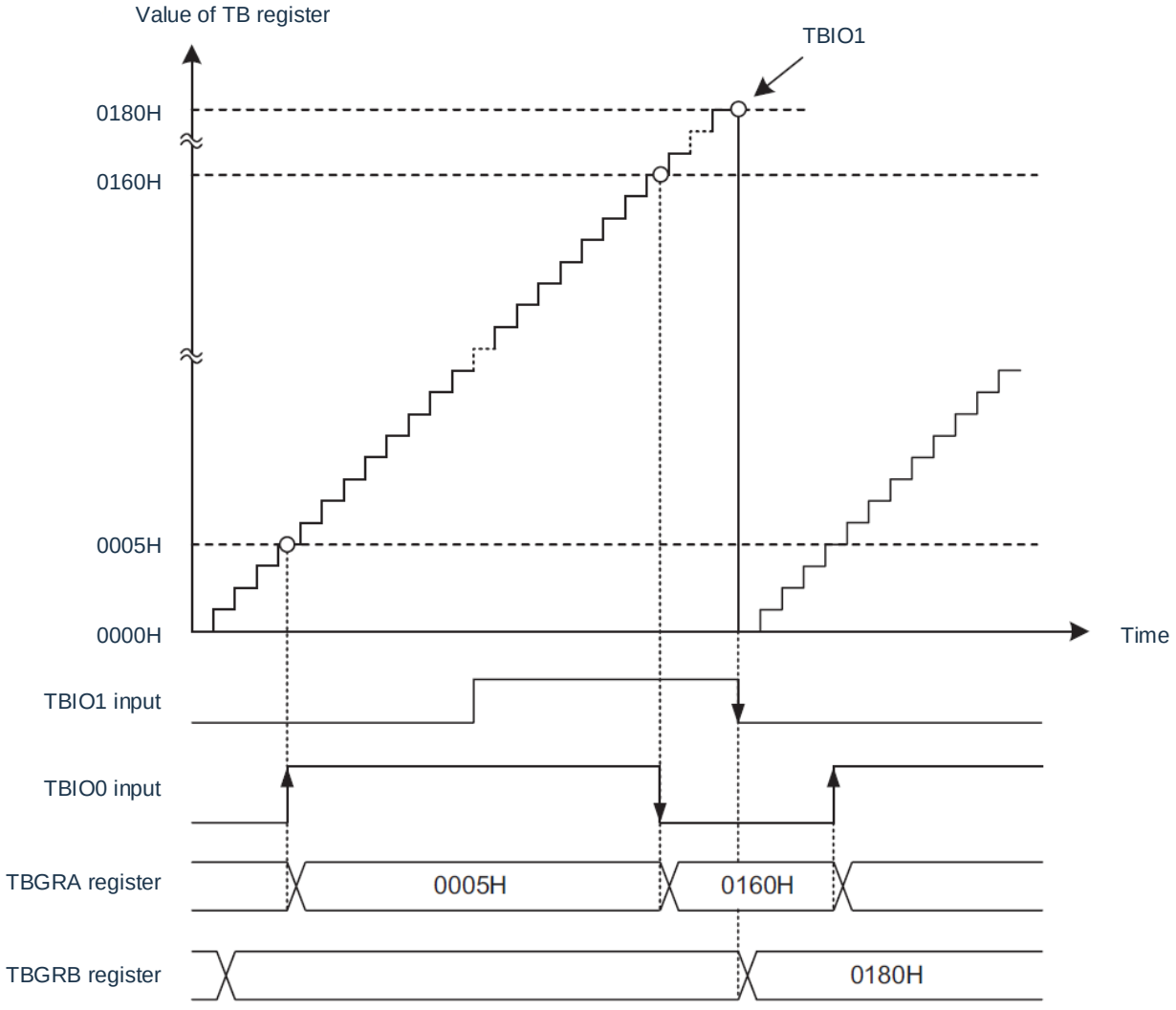
The input capture is shown in Figure 8-18.

In this example, the rising/falling edge is selected as the input edge of the TBIO0 pin, the falling edge is selected as the input edge.

a) The TBGRA register and the TBGRB register must be set as input capture registers through the TBIOR register, and the input edge of the input capture signal must be selected from the rising edge, the falling edge, and the double edge.

b) You must start the TB register count with the TBMR register's TBSTART location '1'.

Figure 8-18 Running Example of Input Capture



It is possible to clear the count when an input capture A or an input capture B occurs by setting the TBCCLR0 bit and the TBCCLR1 bit of the TBCR register. Figure 8-18 is an example of running with the TBCCLR1 bit and the TBCCLR0 position "10B". If set to count clear by input capture during run and input capture is performed when the count value of the timer is 'FFFFH', then the TBIMFA bit and TBIMFB bit interrupt flags and TBOVF bits may become '1'.

8.4.3 timer mode (output comparison function)

This is the mode of detecting whether the contents of the TB register and the contents of the TBGRA register or the TBGRB register are the same (comparison match). If the contents are the same, output any level from the TBIO0 pin or from the TBIO1 pin.

The specifications for the output comparison function are shown in Table 8-10.

Table 8-10. Output Comparison Feature Specifications

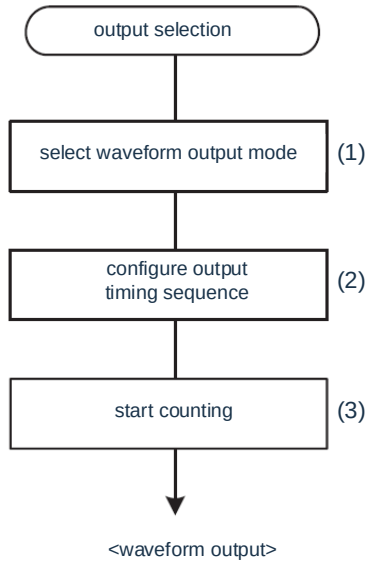
Project	Specifications
count source	fCLK, fCLK/2, fCLK/4, fCLK/8, fCLK/32 External input signal for the TBCLKj pin (program selection of effective edges)
Count	incremental count
counting period	· When the TBCCLR1 and TBCCLR0 bits of the TBCR register are "00B" (free running) 1/fk 65536fk: Frequency of the count source • When the TBCCLR1 bit and the TBCCLR0 bit of the TBCR register are "01B" or "10B" at TBGRj Compare settings for the 1/fk(n+)n:TBGRj register when TB is set to "0000H")
waveform output timing	Comparison matching (same contents of TB register and TBGRj register)
Count Start Condition	Write "1" to the TBSTART bit of the TBMR register.
Count stop condition	Write "0" to the TBSTART bit of the TBMR register.
Generation sequence of interrupt requests	· Comparison matching (same contents of TB register and TBGRj register) · Overspill of TB register
TBIO0 Pin and TBIO1 Functions of pins	I/O port or output compared output (select by pin)
TBCLK0 Pin and The Function of TBCLK1 Pins	I/O port or external clock input
read timer	If you read the TB register, you can read the count value.
write timer	Can write TB register.
Select Features	· Selection of output pins for output comparisons 1 or 2 pins in the TBIO0 and TBIO1 pins · Selection of output levels for comparison matching 'L' level output,'H' level output, or level inversion output · Time sequence in which the TB register is set "0000H" Comparison matching of overflow or TBGRj registers · Buffer Run (see "8.4.1 (2) Buffer Run")

Remark j=A, B

(1) Examples of set-up steps for comparing matched waveform outputs

The set-up steps for comparing the matched waveform outputs are shown in Figure 8-19.

Figure 8-19 Set-up steps for comparing matched waveform outputs



(1) must select compare matching output via TBIOR register between "L" voltage output, "H" voltage input and switching output. If configure as waveform output mode, the port is the output pins (TBIO0, TBIO1) of compare matching. Before first compare matching, the output voltage level of compare matching pin is determined by TBIOA0 and TBIOA1 of TBIOR register, TBIOB must conigure TBGRA regsiter and TBGRB register compare matching occuring timing sequence.

(2) must conigure TBGRA regsiter and TBGRB register compare matching occuring timing sequence.

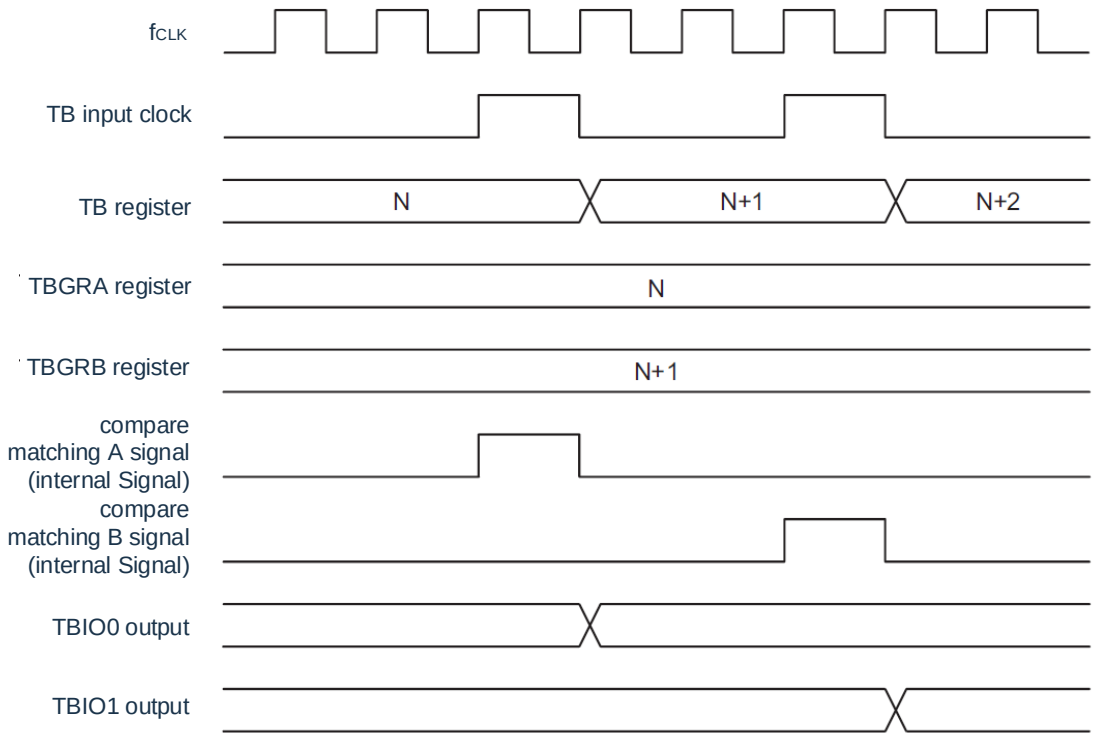
(3) must set TBSTART bit of TBMR register to 1, and start counting of TB register.

(2) Output Timing for Output Comparison

A comparison matching signal is generated in the same final state of the TB register as the TBGRA register or the TBGRB register (when updating the same count value of the TB register). After generating a comparison matching signal, output setting values of the TBIOR register are output from output pins (TBIO0, TBIO1) of the comparison. A comparison matching signal is not generated from the same contents of the TB register and the TBGRA register or the TBGRB register until the input clock of the TB register is generated.

The output timing of the output comparison is shown in Figure 8-20.

Figure 8-20 Output Timing for Output Comparison

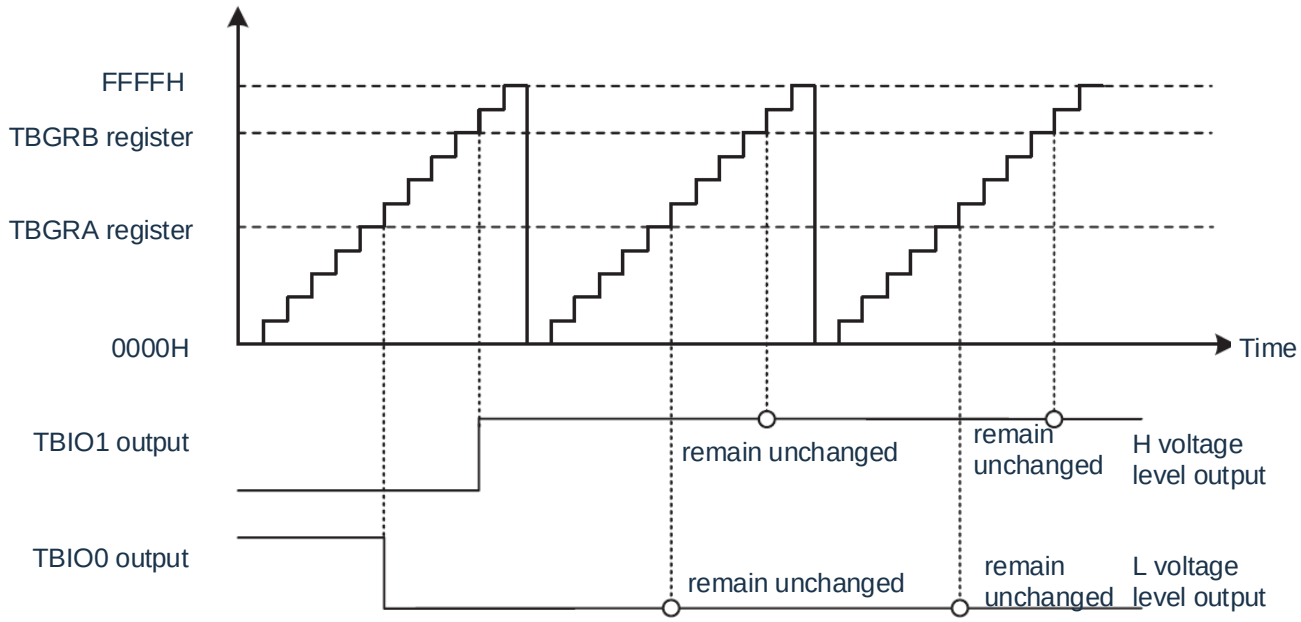


(3) running example

Examples of the operation of the 'L' level output and the 'H' level output are shown in FIG. 8-21.

In this example, that TB register is set to run freely and output an 'L' level when compare match A and an 'H' level. If you set the same level as the pin, the pin level does not change.

Figure 8-21: Examples of operation of the "L" level output and the "H" level output



An example of the operation of the alternate outputs is shown in Figure 8-22. In this example, that TB register is set to run on a periodic count (clear the counter when comparing match B) and output alternately when comparing match A or matching B.

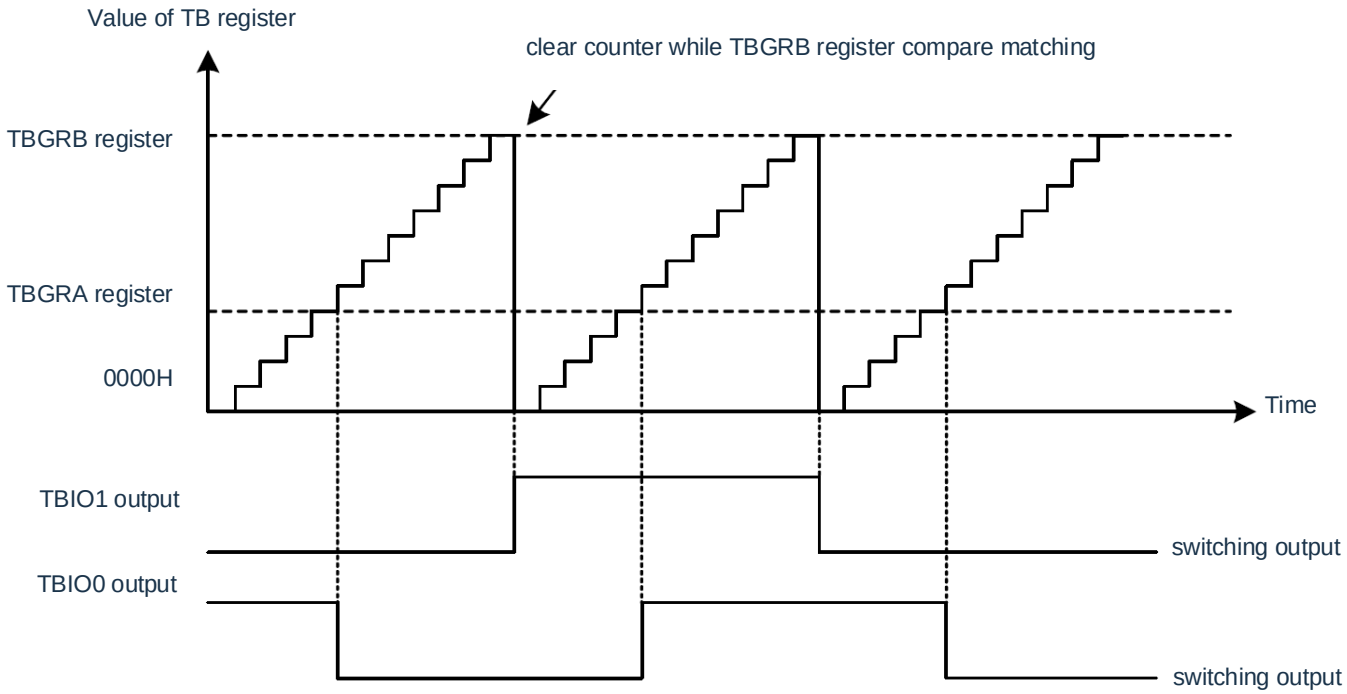
a) The comparison match output must be selected from 'L' level output, 'H' level output, and alternate output through the TBIOR register. If set to waveform output mode, the port is the output pin (TBIO0, TBIO1) that matches.

b) The TBGRA register and the TBGRB register must be set to match the occurrence sequence.

c) You must start the TB register count with the TBMR register's TBSTART location '1'.

Even if the TBSTART position "0" is run, the output pin (TBIO0, TBIO1) of the comparison match is not initialized. To return to the initial value, the output is initialized by writing the TBIOR register (but only if the TBIO00, TBIO01, TBIO10, and TBIO11 bits through the TBIOR register are set to L level output). By setting the TBCCLR0 bit and TBCCLR1 bit of the TBCR register, the counter value of timer B is reset when the input capture/comparison matches. In this case, if the comparison expectation value is 'FFFFH', it changes from 'FFFFH' to '0000H' and the TBOVF bit '1. The mode of the output comparison function using the count value and the comparison expected value of the timer B is also the same.

Figure 8-22 Running Example of Alternate Output



8.4.4 PWM mode

The PWM mode pairing uses the TBGRA register and the TBGRB register to output the PWM waveform from the TBIO0 output pin. The output setting of the TBIOR register is not valid for output pins set to PWM mode. The "H" level output sequence of PWM waveform is set to the TBGRA register, and the "L" level output sequence of PWM waveform is set to TBGRB register.

A PWM waveform of 0-100% duty cycle can be output from a TBIO0 pin by clearing the source by setting the TBGRA register or TBGRB register comparison match to TB register.

The specification of the PWM mode and the combination of the PWM output pins and registers are shown in Tables 8-11 and Tables 8-12 respectively.

When the setting values of the TBGRA register and the TBGRB register are the same, the output values do not change even if the comparison matching occurs.

Table 8-11. Specifications for PWM mode

Project	Specifications
count source	fCLK, fCLK/2, fCLK/4, fCLK/8, fCLK/32 TBCLK0,External input signal for the TBCLK1 pin (program selection of effective edges)
Count	incremental count
PWM waveform	· Set the "H" level output timing of the PWM waveform to the TBGRA register. · Set the "L" level output timing of the PWM waveform to the TBGRB register.
Count Start Condition	Write "1" to the TBSTART bit of the TBMR register.
Count stop condition	Write "0" to the TBSTART bit of the TBMR register.
Generation sequence of interrupt requests	· Comparison matching (same contents of TB register and TBGRj register) · Overspill of TB register
TBIO0 Pin Functionality	PWM output
TBIO1 Pin Functionality	I/O port
TBCLK0 Pin and The Function of TBCLK1 Pins	I/O port or external clock input
read timer	If you read the TB register, you can read the count value.
write timer	Can write TB register.
Select Features	· Time-overflows of TB registers set to "0000H" or comparison matching of TBGRj registers · Buffer Run (see "8.4.1 (2) Buffer Run")

Notes: j=A, B

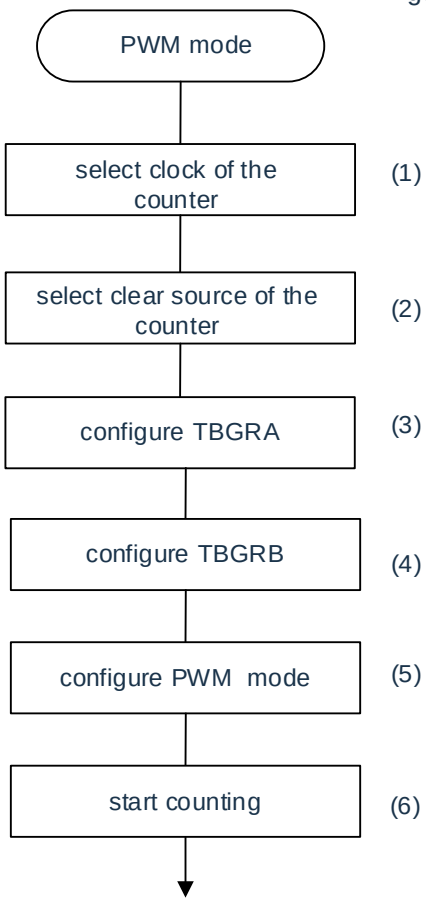
Table 8-12. Combination of PWM output pin and register

output pin	"H" level output	"L" level output
TBIO0	TBGRA	TBGRB
TBIO1	Used as an I/O port.	

(1) Example of set-up steps for PWM mode

Examples of the set-up steps of the PWM mode are shown in Figure 8-23.

Figure 8-23 Example of set-up steps for PWM mode



(1) must select counting source via TBTCK0~TBTCK2 bit of TBCR register. When select external clock, must select external clock edge via TBCKEG0 bit and TBCDEG1 bit of TBCR register.

(2) must select clear source of counter via TBCCLR0 bit and BCCLR1 bit of the TBCR register.

(3) must configure PWM output waveform "H" voltage level output timing sequence via TBGRA register

(4) must configure PWM output waveform "L" voltage level output timing sequence via TBGRB register

(5) The PWM mode must be set via the TBPWM bits in the TBMR register. When set to PWM mode, the TBGRA register and the TBGRB register become the output compare registers for setting the "H" level output/"L" level output timing of the PWM output waveform, regardless of the contents of the TBIOR register. When the PMxx bit of the PM register is "0" and the Pxx bit of the PL register is "0", the TBIO0 pin automatically becomes a PWM output pin. However, the TBIO1 pin is used as an I/O port regardless of the setting of the TBIOR register.

(6) must set TBSTART bit of TBMR register to 1, and start counting of TB register.

(2) running example

An operating example (1) of the PWM mode is shown in FIG. 8-24.

When the PMxx bit of the PM register is '0' and the Pxx bit of the PL register is '0', the TBIO0 pin automatically becomes the output pin when set to the PWM mode. However, the TBIO1 pin is used as an I/O port regardless of the setting of the TBIOR register.

In this example, the comparison match between the TBGRA register and the TBGRB register is set to the counter of the TB register to clear the source. The initial state of the TBIO0 pin depends only on the counter's clear source, as shown in Table 8-13.

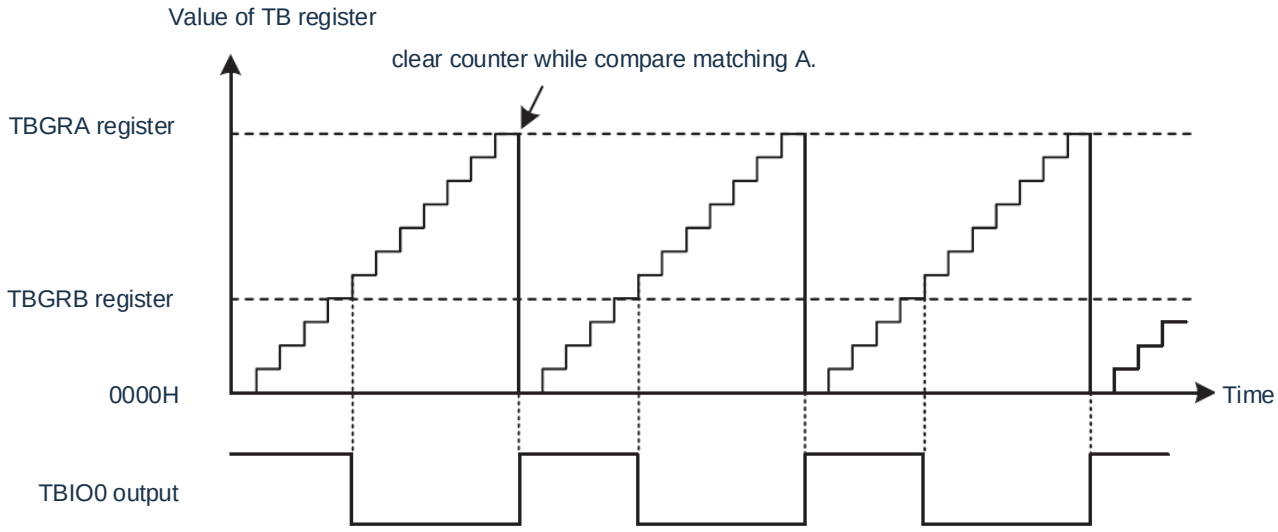
This initialization occurs when the TBSTART bit of the TBMR register is "0" (stop count).

Table 8-13. The Initial State of the TBIO0 Pin and the Correspondence of the Counter Scavenging Source

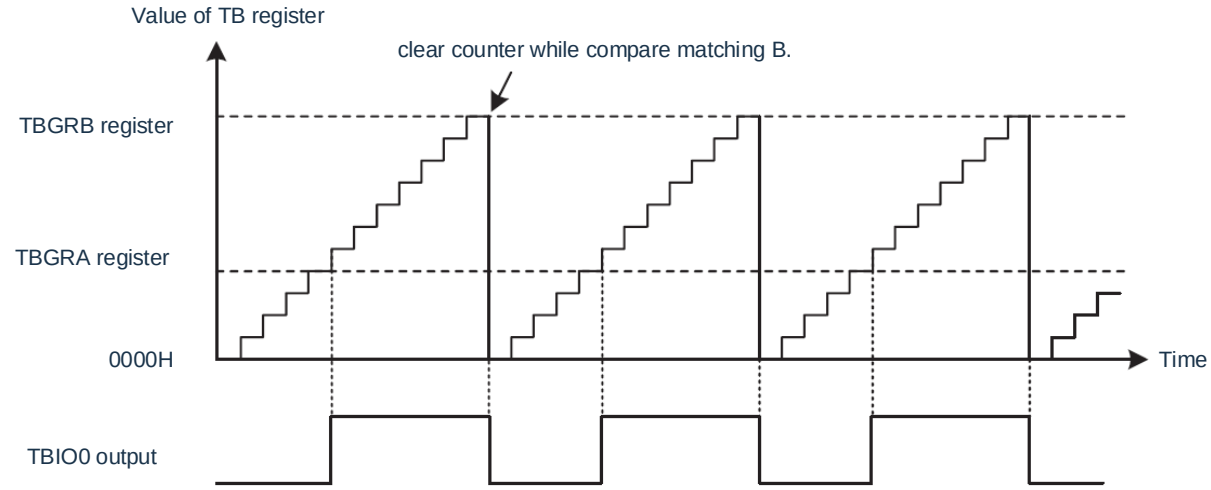
Counter Clear Source	Initial state of the TBIO0 pin
Comparison matching of TBGRA registers	"H" level
Comparison matching of TBGRB registers	"L" level

When the TBCR register has a TBCCLR1 bit and a TBCCLR0 bit of '00B' (no clear), the TBIO0 pin has H level.

Figure 8-24 Running example of PWM mode (1)



(a) clear counter while TBGRA register compare matching。



(b)clear counter while TBGRB register compare matching

Examples of PWM waveforms outputting 0% and 100% duty cycles in PWM mode are shown in FIG. 8-25.

The duty cycle of the PWM waveform is 0% when the comparison match of the TBGRB register is set as the counter's clear source.

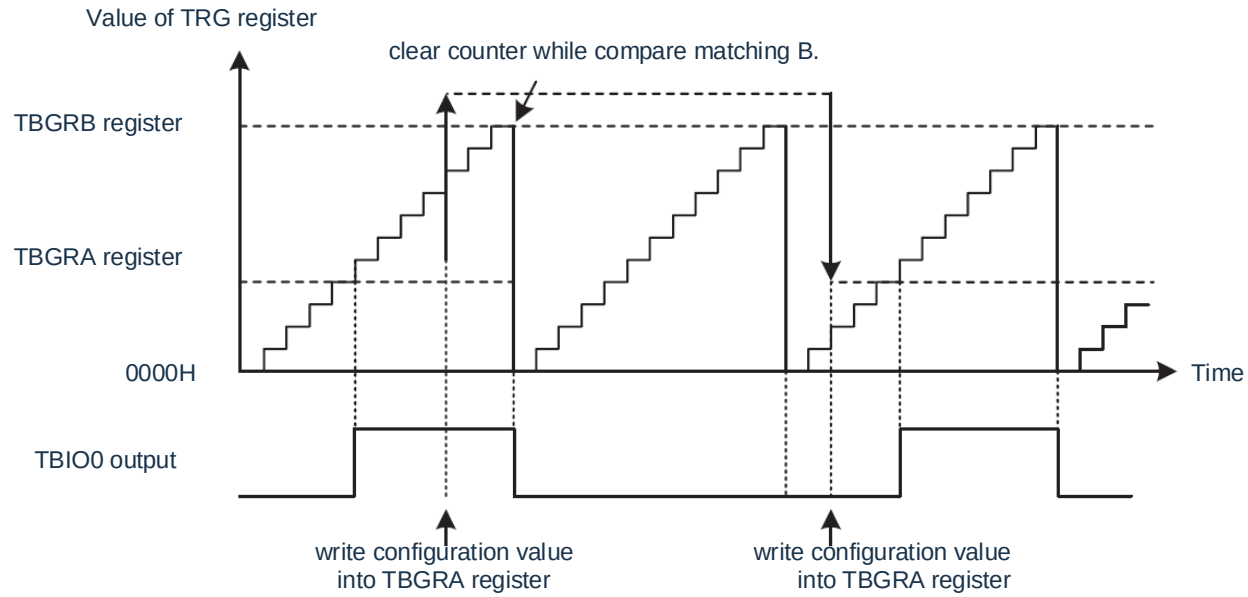
· setting value for the TBGRA register > setting value for the TBGRB register

The duty cycle of the PWM waveform is 100% when the comparison match of the TBGRA register is set as the counter's clear source.

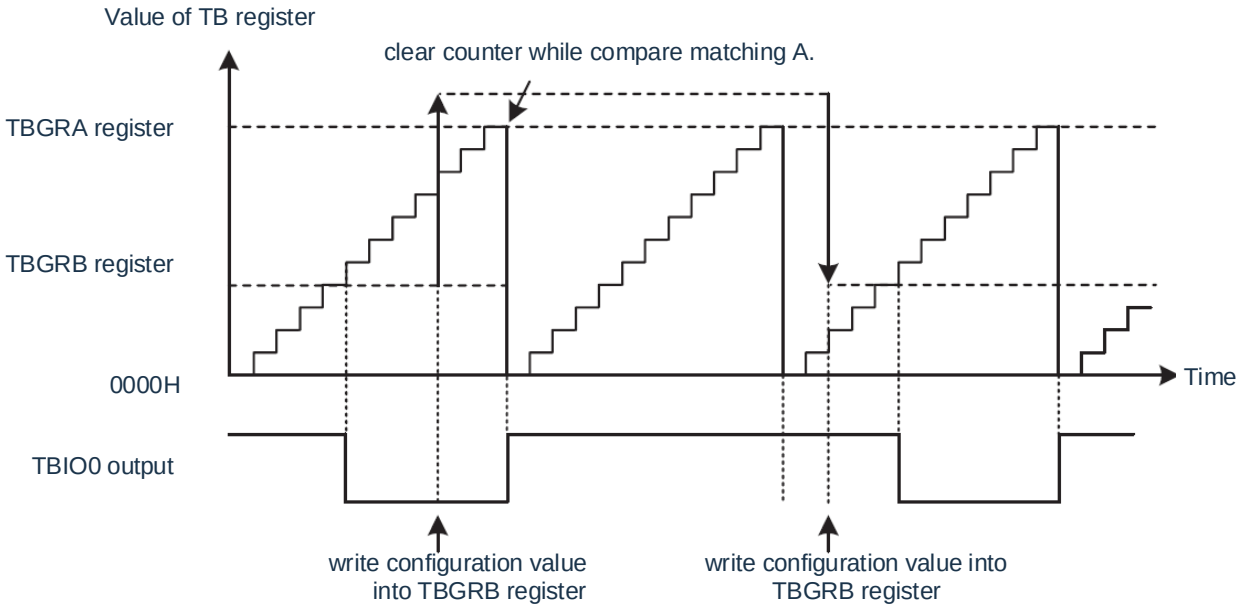
· The setting value of the TBGRB register > The setting value of the TBGRA register The output value will not change even if a comparison match occurs when the following conditions are met.

· Setting value of the TBGRA register = Setting value of the TBGRB register

Figure 8-25 Operating example of PWM mode (2)



(a)Duty cycle 0%.



(b) Duty cycle100%

8.4.5 phase counting mode

The phase count mode detects phase differences of the external input signals of the 2 TBCLK0 pins and the TBCLK1 pins and the TB register performs decrement.

When the PMxx bit of the PM register is "1", the TBCLK0 pin and the TBCLK1 pin are automatically used as the external clock input pin, and the TB register counts according to the CNTEN0CNTEN7 bit setting of the TBCNTC register. However, since the TBCR register has valid TBCCLR0 bits and TBCCLR1 bits, TBIOR, TBIER, TBSR, TBGRA, TBGRB registers, the input capture/output comparison function, PWM output function and interrupt source can be used.

The TB register counts the two-sided rising/falling edges of the TBCLK0 pin and the TBCLK1 pin according to the CNTEN0~CNTEN7 bit settings.

The specifications of the phase count mode and the addition and subtraction conditions of the TB register are shown in Tables 8-14 and Tables 8-15 respectively.

Table 8-14. Specifications for Phase Count Mode

Project	Specifications
count source	External input signal for TBCLK0/TBCLK1 pin
Count	Count Up/Count Down
Count Start Condition	Write "1" to the TBSTART bit of the TBMR register.
Count stop condition	Write "0" to the TBSTART bit of the TBMR register.
Generation sequence of interrupt requests	· Input capture (valid edge of TBIO0/TBIO1 input) · Comparison matching (same contents of TB register and TBGRA/TBGRB register) · Overspill of TB register · Underflow of TB register
TBIO0 Pin Functionality	I/O port, input captured, output compared, or PWM output
TBIO1 Pin Functionality	I/O port, input captured, or output compared
TBCLK0 Pin and The Function of TBCLK1 Pins	external clock input
read timer	If you read the TB register, you can read the count value.
write timer	Can write TB register.
Select Features	· Selection of the addition and subtraction conditions of the counter The selection is performed by the CNTEN0~CNTEN7 bits of the TBCNTC register. · Can use input capture/output comparison function and PWM function.

Table 8-15. TB register addition and subtraction condition

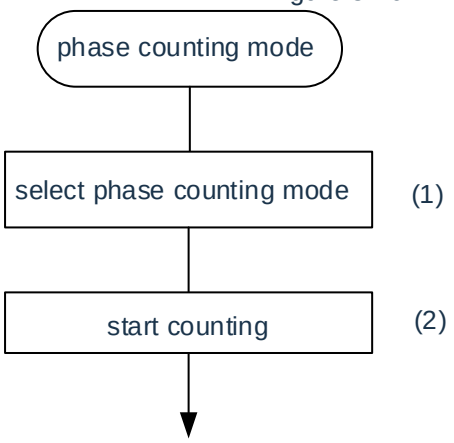
TBCLK1 Pin	↑	"H"	↓	"L"	"H"	↓	"L"	↑
TBCLK0 Pin	"L"	↑	"H"	↓	↓	"L"	↑	"H"
TBCNTC register's CNTEN0~CNTEN7 bits	CNTEN7	CNTEN6	CNTEN5	CNTEN4	CNTEN3	CNTEN2	CNTEN1	CNTEN0
Counter Directional Note	+1	+1	+1	+1	-1	-1	-1	-1

Note: Indicates the direction of the count when each of the TBCNTC register is "1" (Ascending or Decremental). Do not count when "0" is invalid.

(1) Example of set-up steps for phase count mode

An example of the set-up step of the phase count mode is shown in Fig. 8-26.

Figure 8-26 Example of set-up steps for phase count mode



(1)must set TBMDF bit of TBMR register to 1, select phase counting mode

(2)must set TBSTART bit of TBMR register to 1, and start counting of TB register.

<phase counting mode>

(2) running example

Examples of the operation of the phase count mode are shown in Figure 8-27 to Figure 8-30.

In the phase count mode, the TBCLK0 pin and the TBCNTC register are set according to the CNTEN0~CNTEN7 bit

Both the rising(↑) and falling edges(↓) of the TBCLK1 pin count up and down

Figure 8-27 Running example of phase count mode 1

・while TBCNTC register value as "FFH"

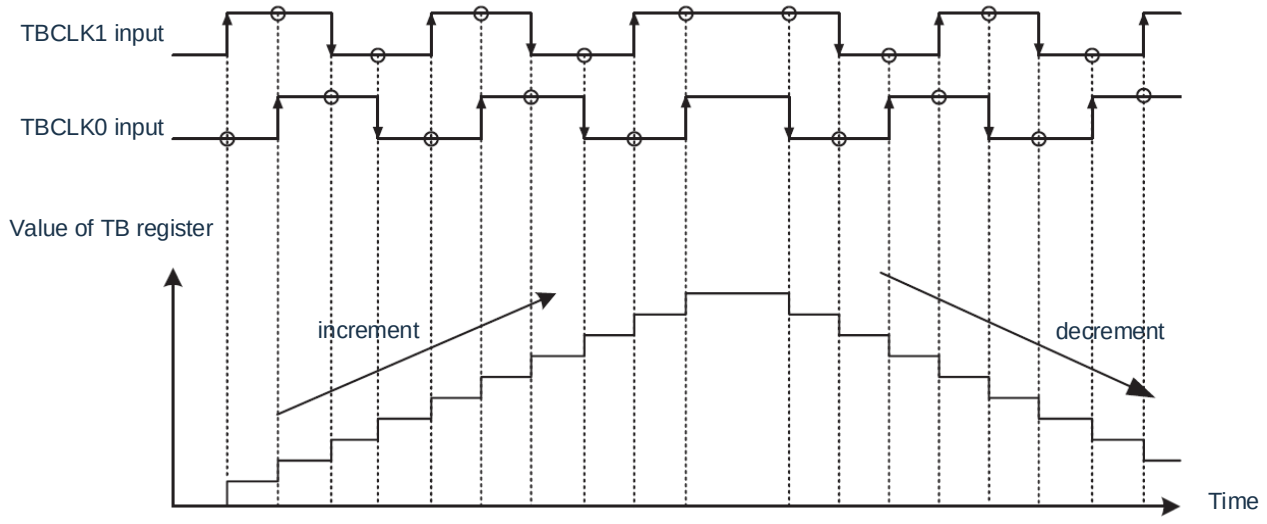


Figure 8-28 Running example of phase count mode 2

・whlie TBCNTC register value as "24H"

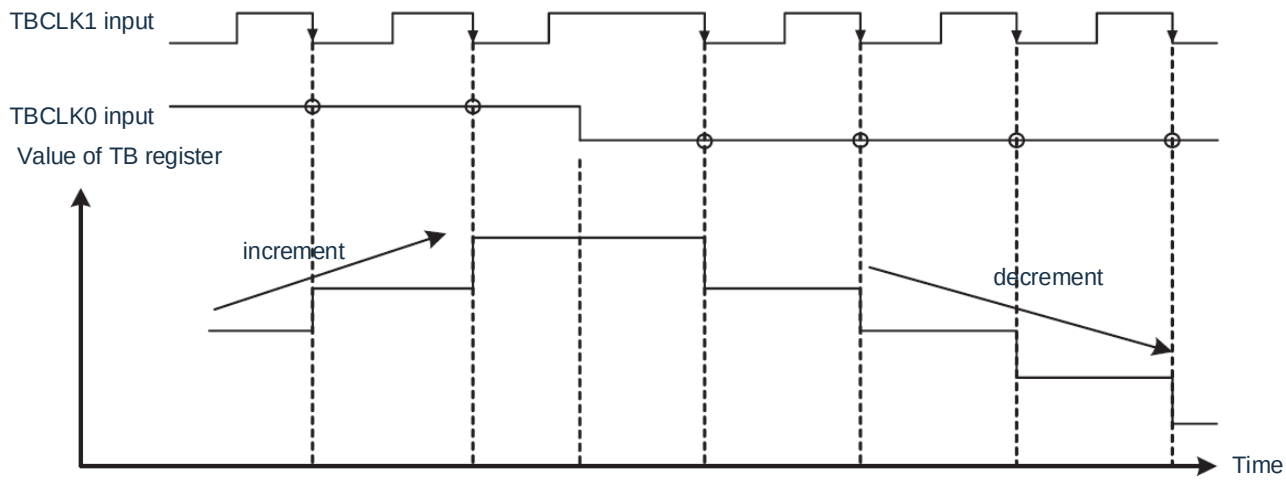


Figure 8-29 Running example of phase count mode 3

・while TBCNTC register value as "28H"

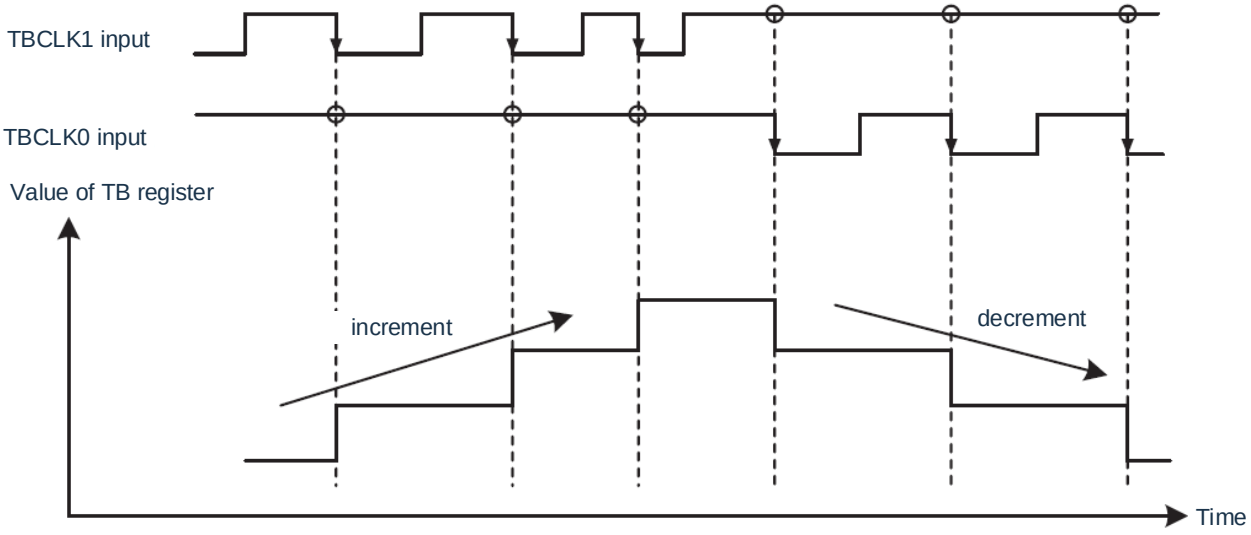
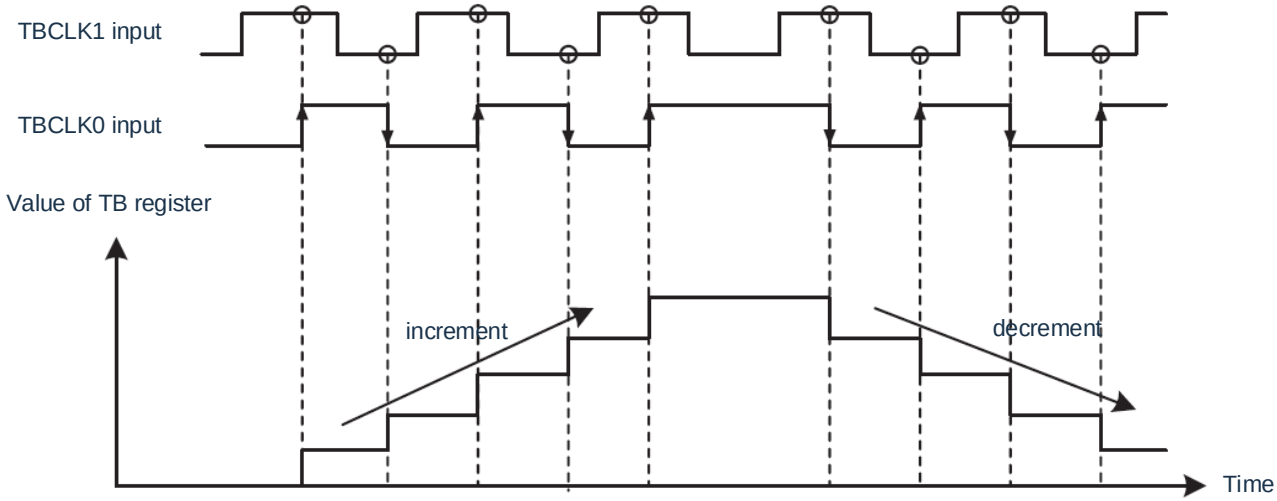


Figure 8-30 Running example of phase count mode 4

・whlie TBCNTC register value as "5AH"



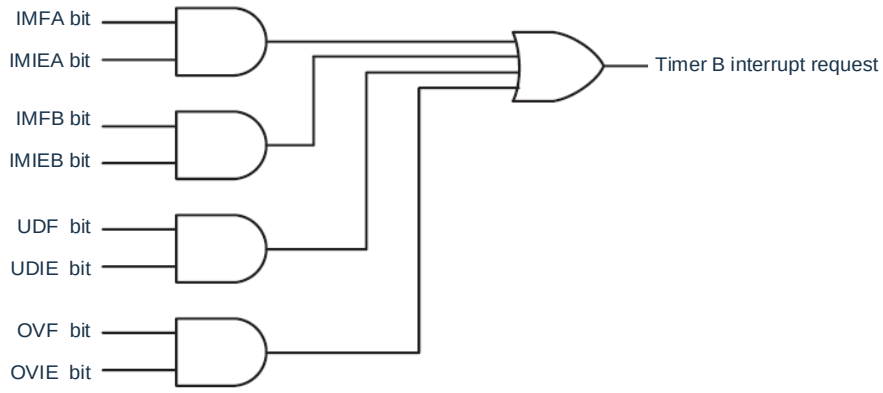
8.5 timer B interrupt

Timer b generate a timer b interrupt request from four interrupt sources. The relevant registers for timer B interrupts are shown in Table 8-16, and the block diagram for timer B interrupts is shown in Figure 8-31.

Table 8-16. Correlation register interrupted by timer B

	State register for timer B	Interrupt Admission Register for	interrupt request flag (Register)	interrupt mask flag (Register)	priority assignment flag (Register)
Timer B	TBSR	TBIER	TBIF (IF2H)	TBMK (MK2H)	TBPR0 (PR02H) TBPR1 (PR12H)

Figure 8-31 Block diagram of timer B interrupt



IMFA, IMFB, UDF, OVF: TBSR register Bits
IMIEA, IMIEB, UDIE, OVIE: TBIER register Bits

Since timer B generate 1 interrupt requests (interrupt of timer B) from a plurality of interrupt request sources, there are differences from other maskable interrupts:

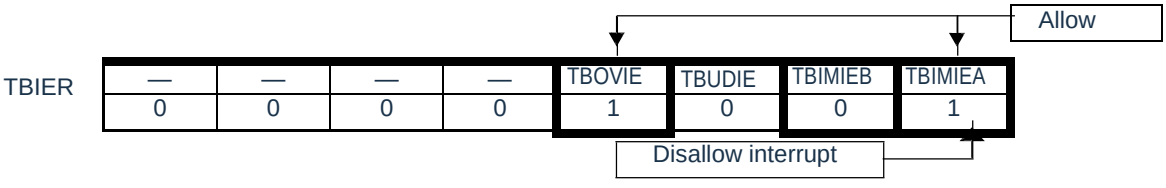
- When the bit of the TBSR register is '1' and the corresponding bit of the TBIER register is '1' (interrupt allowed), the Bit0 bit of the IF1D register becomes '1'.
- When multiple bits of the TBIER register are '1', it is necessary to determine which interrupt is generated by which request source through the TBSR register.
- Because even if you accept interrupts, the TBSR register members do not automatically become "0", you must "0" these locations in the interrupt.
- If the interrupt is set to 0 by the interrupt permission register (TBIER) of the timer B, any of the following methods.

a) The object status flag must be written "0" after setting timer B interrupt permit register (TBIER) to "00H".

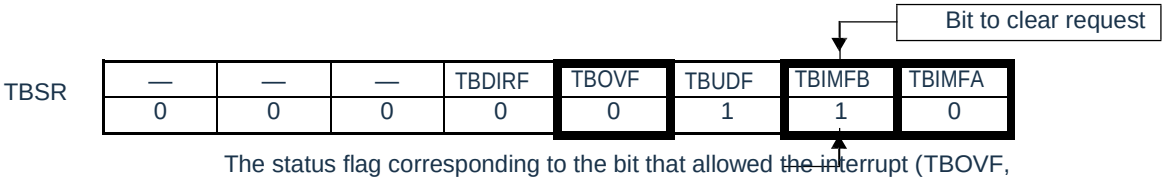
b) When timer B interrupt enable register (TBIER) has bit of ' 1 ' and interrupt source status flag of '0 ' allowed by this bit.

(Example) Cleaning TBIMFB in a state where TBIMIEA and TBOVIE are interrupts allowed and TBIMIEB is interrupts prohibited

- Timer B interrupts the state of the Permit Register (TBIER)



- Status of timer B status register (TBSR)

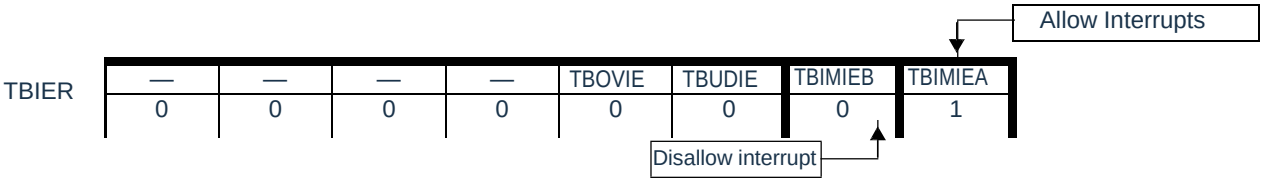


The status flag corresponding to the bit that allowed the interrupt (TBOVF, TBIMFA) is "0", so you must write "0" to TBIMFB.

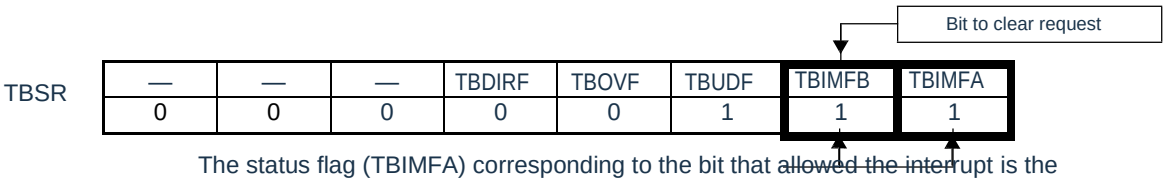
c) When timer B interrupt enable register (TBIER) has bit of ' 1' allowed and interrupt source state flag of '1' allowed by this bit, must write '0'.

(Example) When TBIMIEA is Interrupt Allowed and TBIMIEB is Interrupt Disabled to clear TBIMFB

- Timer B interrupts the state of the Permit Register (TBIER)



- Status of timer B status register (TBSR)



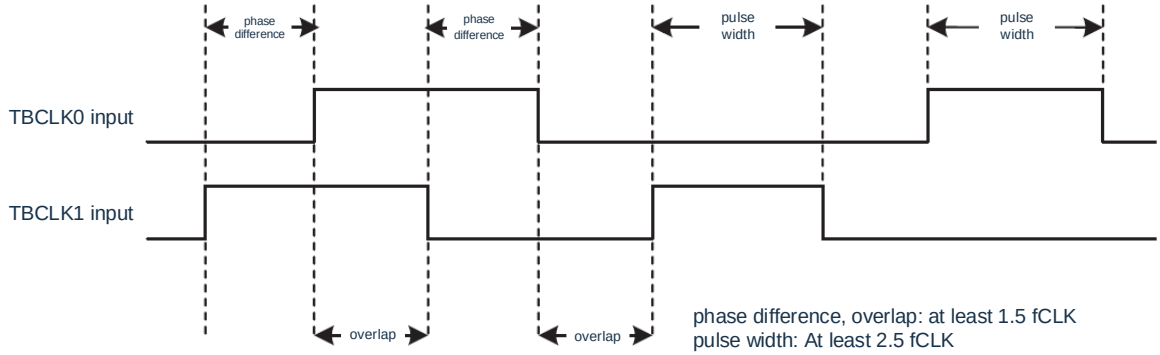
The status flag (TBIMFA) corresponding to the bit that allowed the interrupt is the "1", so you must write "0" to both TBIMFA and TBIMFB.

8.6 Precautions when using timer B

8.6.1 Phase difference, overlap and pulse width in phase counting mode

The phase difference and overlap of the external input signals of the TBCLK0 pin and the TBCLK1 pin must be at least 1.5 fCLK and at least 2.5 fCLK. The phase difference, overlap, and pulse width in the phase count mode are shown in Figure 8-32.

Figure 8-32 Phase difference, overlap and pulse width in phase counting mode



8.6.2 mode switching

- To switch modes during the run, you must switch after "0" (stop count) of the TBSTART location of the TBMR register.
- The bit0 of the IF1D register must be set to "0" after switching mode and before starting running. Refer to "Chapter 25 Interrupt Features" for more information.

8.6.3 Switch of count source

- To switch count sources, you must make a switch note after stopping the count. Change Steps
  a) Place the TBSTART location of the TBMR register "0" (stop count).
  b) Change the TBTCK0~TBTCK2 bit of the TBCR register.

Note: The registers and bits that are not overridden during the count are:

a) All bits except TBSTART bits of the TBMR register
b) TBCNTC register
c) TBCR register
d) TBIOR register

8.6.4 Set-up steps for TBIO0 and TBIO1 pins

After reset, the TBIO0 pin and the TBIO1 pin's multiplexing I/O port are used as input ports.

- When you want to export from the TBIO0 and TBIO1 pins, you must follow the steps below.

Change Steps

1) Make the allowed settings for mode, initial value, and output (because both initial and allowed settings are made via SFR).
2) Position "0" of port register corresponding to TBIO0 and TBIO1 pins.
3) Set the bit of the port mode register corresponding to the TBIO0 pin and the TBIO1 pin to output mode (starting with the TBIO0 pin and TBIO1 pin).
4) Start counting (the TBSTART bit of the TBMR register is "1").

- To change the bit of the port mode register for the TBIO0 pin and TBIO1 pin from output to input, you must follow these steps.

Change Steps

1) Set the bit of the port mode register corresponding to the TBIO0 and TBIO10 pins to the input mode (starting with the TBIO0 and TBIO1 pins).
2) Set as input capture.
3) Start counting (the TBSTART bit of the TBMR register is "1").

- When you switch the TBIO0 and TBIO1 pins from output to input mode, you may run input capture based on the status of the pins. when a digital filter is not used, edge detection is performed at least after two CPU clock cycles; When digital filter is used, it needs 5 sampling clock cycles of digital filter to detect the edge.

8.6.5 external clock TBCLK0 and TBCLK1

The external clock input by the TBCLKj pin (j=0,1) must have at least 3 clock cycles of runtime ($_{fCLK}$) of timer B.

8.6.6 Read and write access to SFR

To set timer B, you must first set the TMBEN location "1" of the PER1 register. When the TMBEN bit is '0', the write operation of the control register of timer B is ignored and the read values are initial (except for port registers and port mode registers).

1) TBMR register

When switching the clock of a digital filter, you must follow these steps:

a) The TBSTART bit is set with '0' (stop counting) the TBDFA bit and TBDFB bit of the TBMR register (TBIO0 pin and TBIO1 pin), the TBDFCK0 bit and TBDFCK1 bit of the TBMR register (clock bit of the digital filter function).
b) Position TBSTART "1".

However, a 1-shot setting can be performed without setting the digital filter and without changing the TBDFCK1 and TBDFCK0 bits after reset.

In addition to external input pins (TBIO0, TBIO1), events entered by EVENTC can be selected as the source of operation for input capture. When this feature is not used, the TBELCICE position of the TBMR register "1" must be set with the input capture function (TBIOB2~TBIOB0=100B). This feature is not valid when using the output compare

function in PWM mode or timer mode (TBPWM=1, TBIOB2=0).

2) TB register

· The write operation of the TBMR register preferentially resets the count generated by the running condition of timer B.

8.6.7 Input capture operation when stopping count

In input capture mode, if the TBSTART bit of the TBMR register is "0" (stop count), enter the TBIO 0/TBIO1 pin

The TBIOj0 bit and TBIOj1 bit of the TBIOR register select edges, which generate input capture interrupt requests on the valid edges of TBIO 0/TBIO1 inputs

(j=A, B).

Chapter 9 timer C

9.1 Function of timer C

Timer C is a timer that triggers an input capture function through software, comparator 1, and timer M.

The actions are as follows:

Count Start: The count action is triggered by software or timer M

Count stopped: The count is stopped by the output of the software or comparator 1

Input capture: When the interruption of the comparator 1 occurs, the count value is transmitted to the buffer

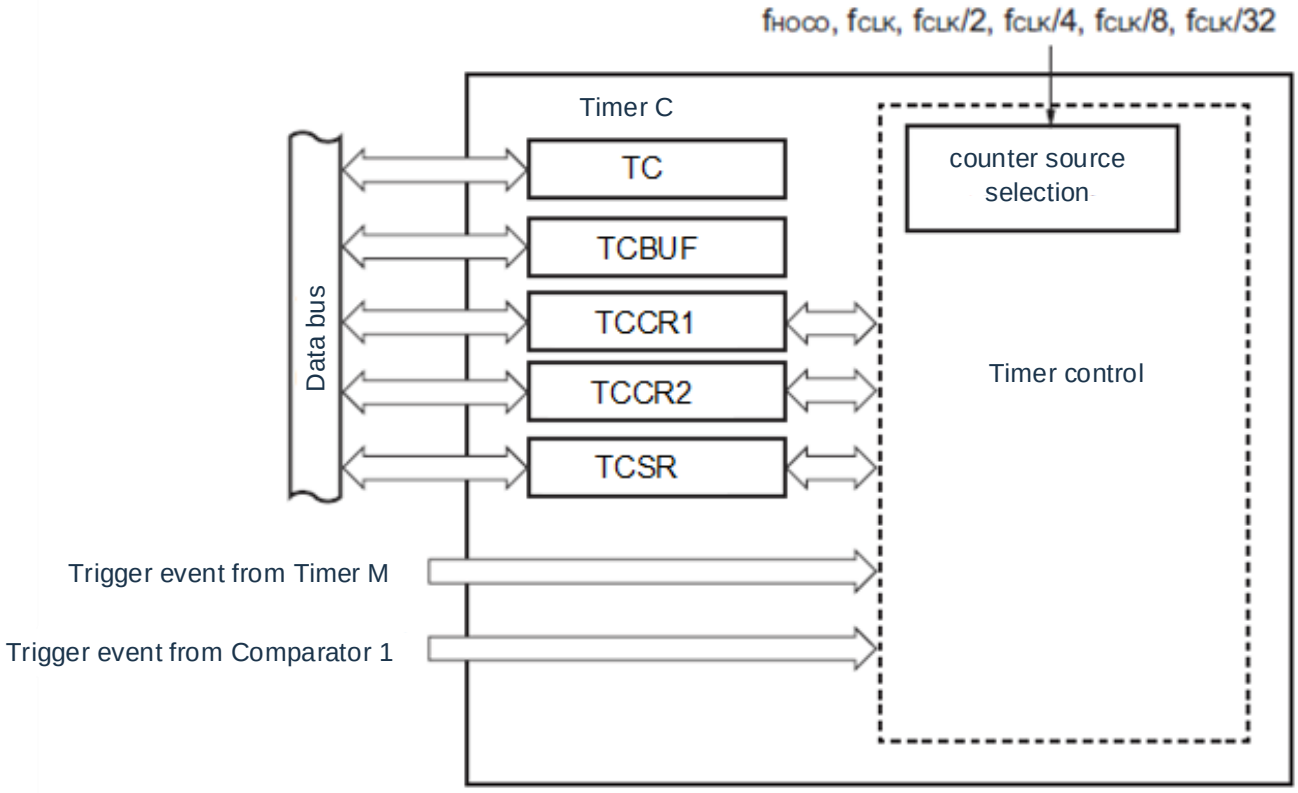
Count reset: The count reset is triggered by timer m or comparator 1

The action clock for Timer C can be selected from fCLK or fHOCO.

9.2 Structure of timer C

The block diagram of timer C is shown in Figure 9-1.

Figure 9-1 Block diagram of timer C



9.3 Register for controlling timer C

The register for controlling timer C is shown in Table 9-1.

Table 9-1 Register for controlling timer C

register name	symbol
Peripheral Enable Register 1	PER1
timer C count register	TC
timer C count buffer register	TCBUF0
timer C control register 1	TCCR1
timer C control register 2	TCCR2
timer C status register	TCSR

9.3.1 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.

You must set bit1 (TMCEN) to "1" when you want to use timer C. The PER1 register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 9-2 Format of Peripheral Admission Register 1(PER1)

Address: 0x4002081A After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN	TMMEN	DMAEN	PWMPEN	TMCEN	TMAEN

TMCEN	Control of an input clock of a timer C is provided
0	Stop provide an input clock. · SFR used by timer C cannot be written. · Timer A is in a reset state.
1	Provides an input clock. · Read and write the SFR used by timer C.

Note:

1) The TMCEN location "1" must be first used to set timer C. When the TMCEN bit is '0', the write operation of the control register of timer C is ignored, and the read values are all initial values.

2) To select fHOCO as the counter source for timer C, fCLK must be set to fIH before bit2 (TMCEN) of Peripheral Admission Register 1 (PER1). If you want to change fCLK to a clock other than fIH, you must change it2 (TMCEN) after clearing the peripheral enable register 1 (PER1).

9.3.2 timer C count register (TC)

This is a 16-bit register. If this register is written, data is written to the reload register. If you read this register, read the count value.

Figure 9-3 Format of timer C count register (TC)

Address: 0x40042C50 After reset: 0000H R/W

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

—	Features	Set Scope
bit15~0	Increment count, TCSR's TCOVF bit is set to 1 when overflows occur	0000H~FFFFH

Note: FRQSEL4=1 for option bytes (000C2H/010C2H) and TMCEN=0 for PER1, the reset value is variable. If a readout of that initial value is required, it is read out with fCLK selected and TMCEN=1.

9.3.3 timer C count buffer register (TCBUF)

Figure 9-4 Format of timer C count buffer register (TCBUF)

Address: 0x40042C52 After reset: 0000H R

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

—	Features	Set Scope
bit15~0	When the interruption of comparator 1 occurs, the value of TC register is transferred to buffer register	0000H~FFFFH

Note: FRQSEL4=1 for option bytes (000C2H/010C2H) and TMCEN=0 for PER1, the reset value is variable. If a readout of that initial value is required, it is read out with fCLK selected and TMCEN=1.

9.3.4 timer C control register 1 (TCCR1)

Figure 9-5: Timer C controls the format of register 1 (TCCR1)

Address: 0x40042C54 After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
TCCR1	TCK2	TCK1	TCK0	START_MD	TRIG_MD_SW	TRIG_MD_HW	TM_TRIG	OVIE

TCK2	TCK1	TCK0	Count Source Selection Note
0	0	0	fCLK, fHOCO note 1
0	0	1	fCLK/2 Note 2
0	1	0	fCLK/4 Note 2
0	1	1	fCLK/8 Note 2
1	0	0	fCLK/32 Note 2
above			Disable setting

START_MD	Selection of Count Start Source
0	software setting Timer C count start
1	Output signal from Timer M triggers start of Timer C count
Count starts when START_MD=1 after the TSTART bit of TCCR2 is set to 1	

TRIG_MD_SW	Effective Signal of Software Reset Timer C
0	Prevent software reset of counter for Timer C
1	Counter that allows software to reset Timer C
Invalid when START_MD=1	

TRIG_MD_HW	Action selection when Timer C is triggered from the output of Timer M
0	Count after Timer C reset
1	Timer C Start Count
Invalid when START_MD=0	

TM_TRIG	Hardware triggered selection from Timer M
0	Count Start Action for TM0 (TSTART0 set to 1) Triggers the Timer C Start Count
1	Count Start Action for TM 1 (TSTART 1 set to 1) Triggers the Timer C Start Count
Invalid when START_MD=0	

OVIE	Allow overspill interrupt signal generation
0	Prevent Interrupt When TC Register Overflows
1	Allow interrupt to occur when TC register overflows

Note: When interacting with Timer M, the action clock of Timer C must be set to match the action clock frequency of Timer M.

Note 1: Select fCLK when FRQSEL4=0 for option bytes (0002H/010C2H). Select fHOCO when FRQSEL4=1 for option bytes (0002H/010C2H).

Note 2: Option byte (000C2H/010C2H) cannot be set with FRQSEL4=1.

Note: FRQSEL4 = 1 for option bytes (000C2H/010C2H) and TMCEN = 0 for PER1, the reset value is indefinite. If a readout of that initial value is required, it is read out with fCLK selected and TMCEN=1.

9.3.5 timer C control register 1 (TCCR2)

Figure 9-6: Timer C controls the format of register 1 (TCCR2)

Address: 0x40042C55 After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
TCCR2	0	0	0	0	0	CMP_TCR1	CMP_TCR0	TSAT

CMP1_TCR1	CMP1_TCR0	Action selection when triggering Timer C by output from comparator 1
0	0	Timer C count stopped
0	1	The Timer C count value is transferred to the buffer register and the count continues
1	0	Timer C count value changed to 0000H, count continues
1	1	The Timer C count value is transferred to the buffer register, the Timer C count value becomes 0000H, and the count continues

TSTART	Timer C Action Start Control Note
0	TC Count Stop
1	TC Count Start

Note: If that stop signal control from the CMP1 and the TSTART control compete, the stop signal from the comparator 1 has a higher priority.

Note: FRQSEL4=1 for option bytes (000C2H/010C2H) and TMCEN=0 for PER1 is an indefinite reset. If read out initial value is required, read out is required under condition that fCLK is selected and TMCEN=1.

9.3.6 timer C status register (TCSR)

Figure 9-7: Timer C controls the format of register 1 (TCSR)

Address: 0x40042C56 After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
TCSR	0	0	0	0	0	0	TCSB	TCOVF

TCSB	Timer C Counter Status Flag Note 1
0	Count stopped
1	Counting

TCOVF	Overflow Status Flag Bit for the TimerC Counter Note 2,3
0	No overflow occurred
1	Overspill occurred

Note 1: Read-only, not writable.

Note 2: Only 0 can be written, write 1 is invalid.

Note When the 3:Timer C counter overflow and TCOVF write 0 occur simultaneously, the overflow has a higher priority.

Note: FRQSEL4 = 1 for option bytes (000C2H/010C2H) and TMCEN = 0 for PER1, the reset value is indefinite. If a readout of that initial value is required, it is read out with fCLK selected and TMCEN=1.

9.4 Operation of timer C

The timer C may be started by a signal trigger count of the timer M and stopped by a signal trigger count of the comparator 1.

9.4.1 count source

The action clock of the timer C is determined by the option byte and the frequency division setting of the timer C.

1) Clock source for timer C

- When FRQSEL4=1 for option byte (000C2H/010C2H), the CPU/peripheral action clock (fcpu) selects the high speed oscillation clock fHOCO and the clock source of the timer C selects fHOCO.
- FRQSEL4=0 for option bytes (000C2H/010C2H) or fcpu for high speed system clock (fMX) and fcpu for timer C.

2) timer C count clock source

Set the count frequency using TCCR1.

If you use Timer M to trigger the Timer C count start, you must set the Timer C action clock to match the Timer M action clock frequency.

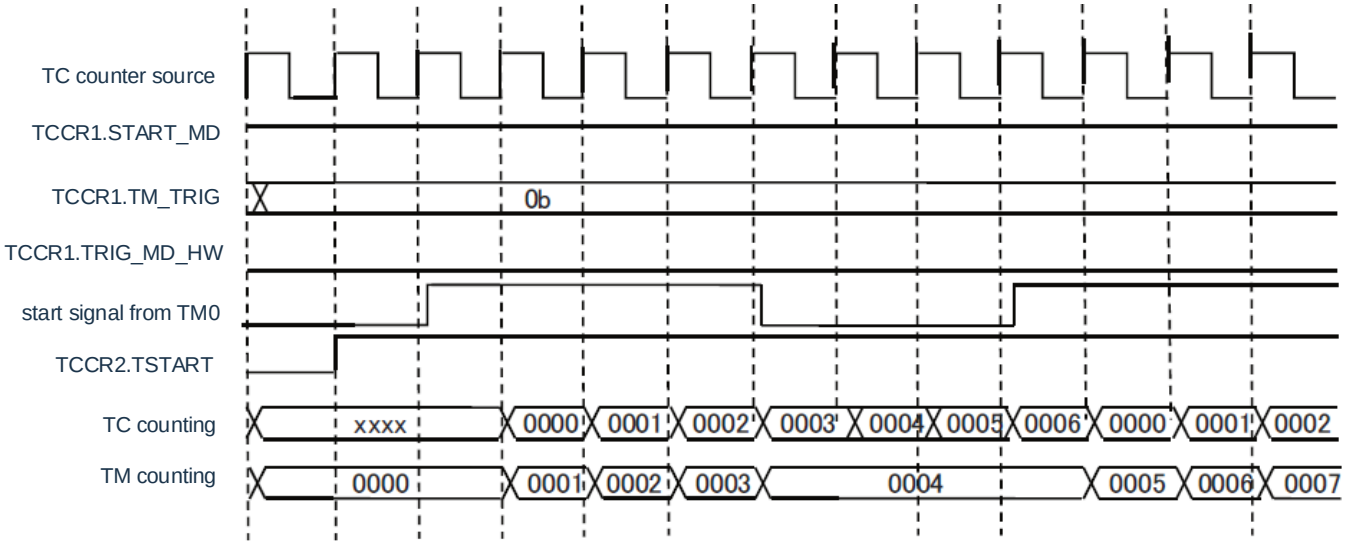
9.4.2 Timer C starts counting actions

Start the count action for Timer C with Timer M or software settings.

9.4.2.1 Select the Timer M signal as the setting and action for the trigger

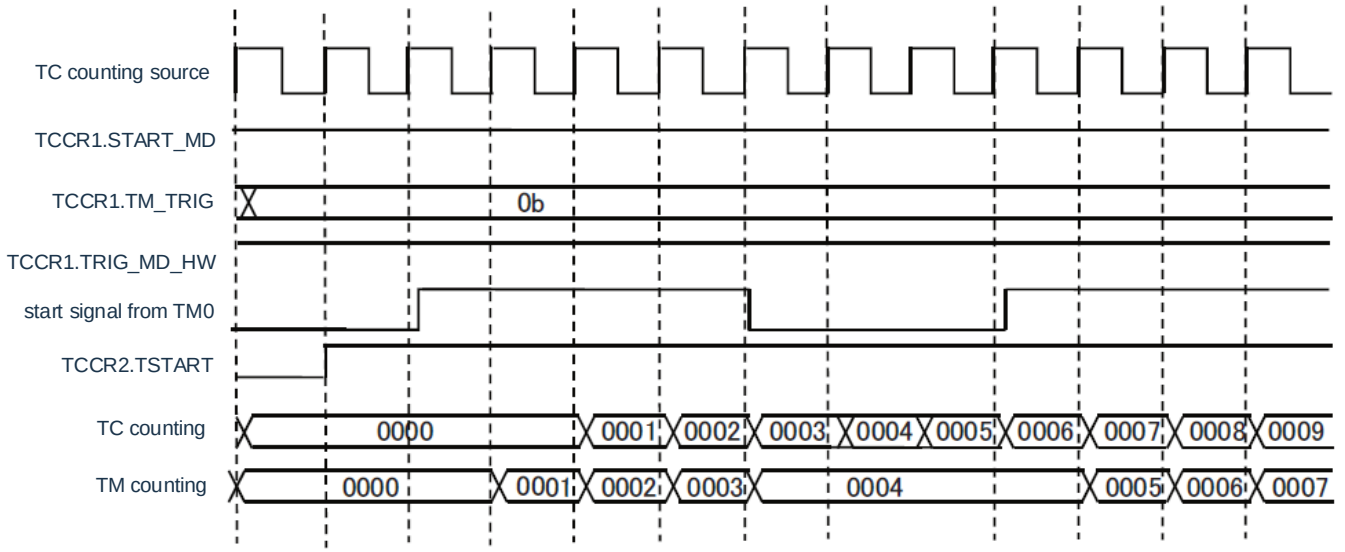
1) Reset and start set steps for the Timer C count when TRIG_MD_HW=0:
   a) Select the Timer M output signal as the trigger source for the count start: TCCR1.START_MD=1
   b) Select the trigger for Timer C: TCCR1.TRIG_MD_HW=0
   c) Select the trigger for Timer M_0/1: TCCR1.TC_TRIG=1/0
   d) Timer C count start: TCCR2.TSTART=1

Figure 9-8 Example of Timer C count reset and start (TRIG_MD_HW=0)



2) To set the start of the Timer C count when TRIG_MD_HW=1:
   a) Select the Timer M output signal as the trigger source for the count start: TCCR1.START_MD=1
   b) Select the trigger for Timer C: TCCR1.TRIG_MD_HW=1
   c) Select the trigger for Timer M_0/1: TCCR1.TC_TRIG=1/0
   d) Timer C count start: TCCR2.TSTART=1

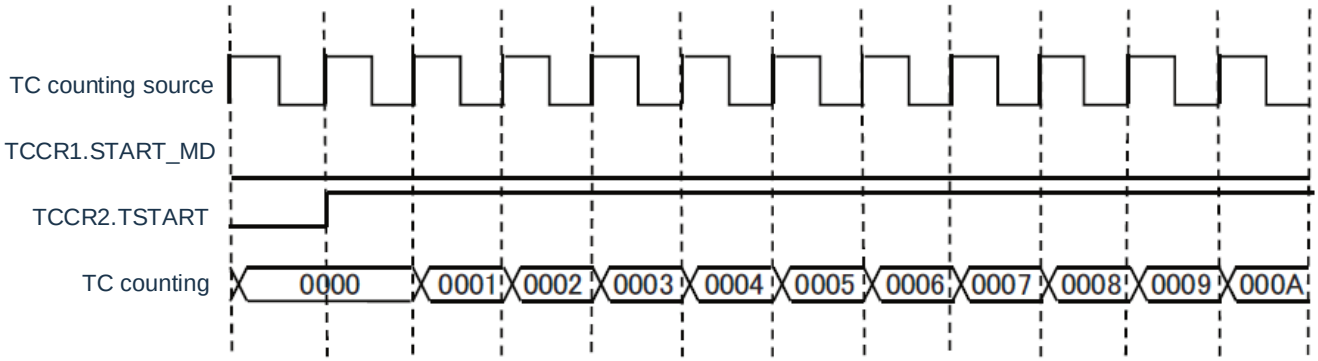
Figure 9-9 Example of Timer C count start (TRIG_MD_HW=1)



9.4.2.2 Select the settings and actions when the software fires

1. Count Start Source Selection Software Trigger: TCCR1.START_MD=0
2. Timer C count start: TCCR2.TSTART=1

Figure 9-10 Example of Software Trigger Timer C Count Start



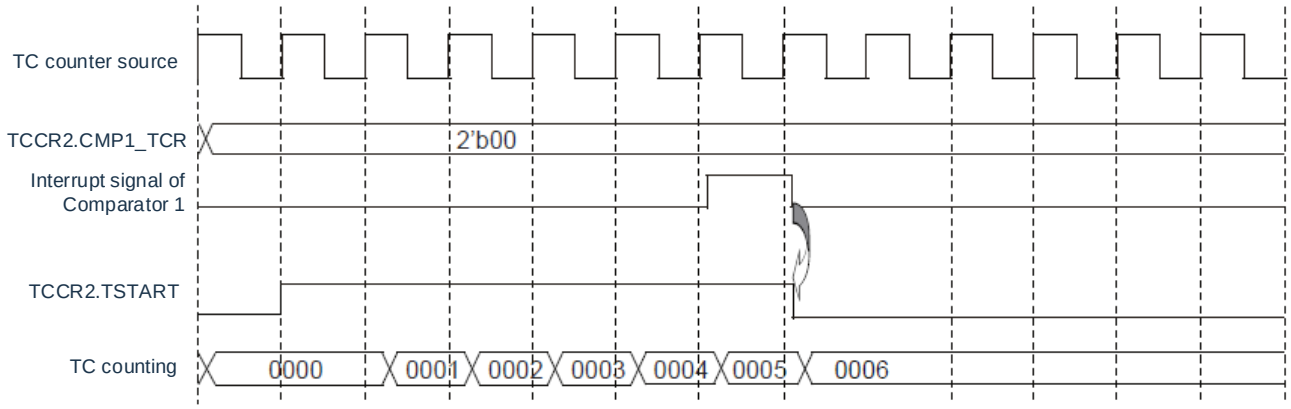
9.4.3 Timer C counts stop actions

While counting, timer c may stop the count action by the trigger of comparator 1 or by software setting.

9.4.3.1 Select Comparator 1 as Trigger Settings and Actions

1. Select Comparator 1 as the trigger: TCCR2.CMP1_TCR=00
2. Timer C Count Start: TCCR2.TSTART=1

FIG. 9-11 Select the example of comparator 1 triggering Timer C to stop



9.4.3.2 Settings and actions when software is triggered

1. TimerC Count Start: TCCR2.TSTART=0
2. Software setting TCCR2.TSTART to 0, Timer C count stopped

9.4.4 Input capture action

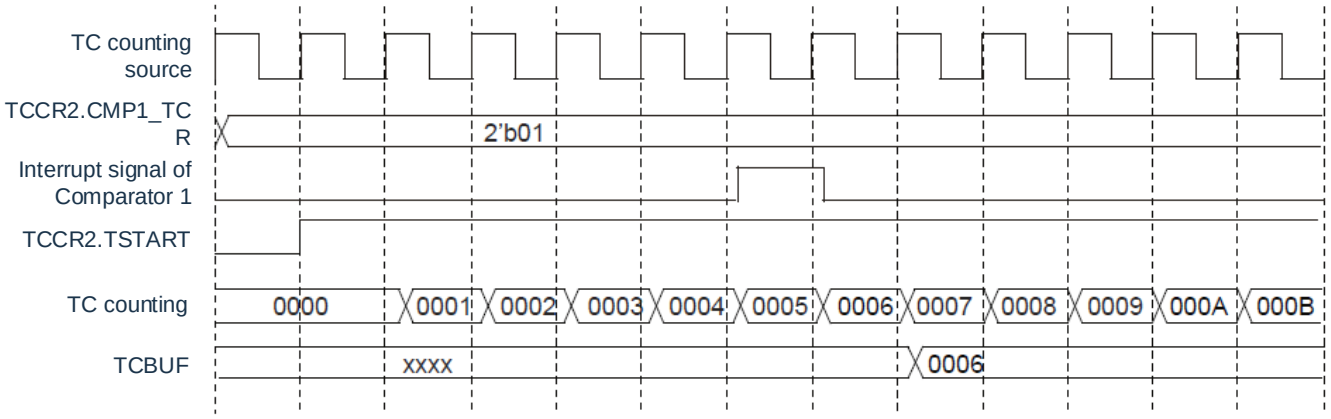
If an interrupt is generate by that comparator 1 during the Timer C action, the action of the Timer C may change.

1) Case 1:

A count value of TCCR2.CMP1_TCR =01,TimerC is transferred to the count buffer.

- TCCR2.CMP1_TCR=01 (select input capture feature)
- TCCR2.TSTART=1 (Timer C count starts)

Figure 9-12 Example of Input Capture Action

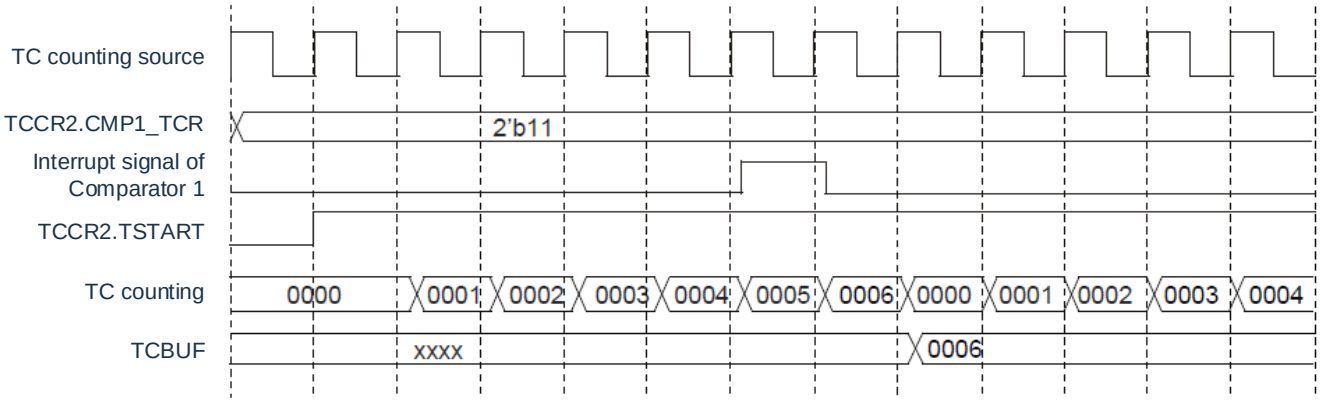


2) Case 2:

TCCR2.CMP1_TCR=11, the count value of TimerC is transferred to the count buffer, the count value of TimerC is reset.

- TCCR2.CMP1_TCR=11 (select Input Capture and Reset function)
- TCCR2.TSTART=1 (Timer C count starts)

Figure 9-13 Example of Input Captured Action (Also Reset Count Value)

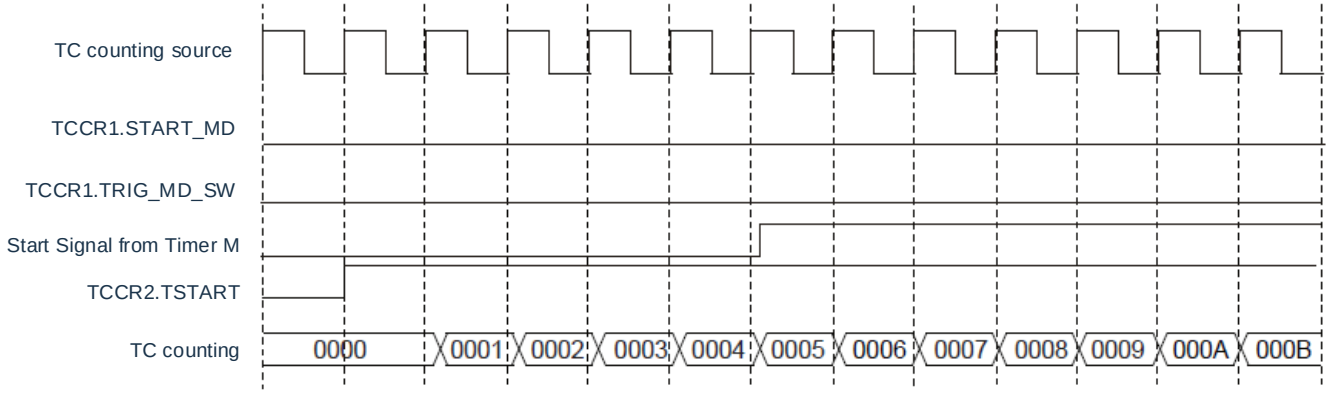


9.4.5 timer C count reset action

When the Timer C action is started, the output signal of Timer M and the output signal of comparator 1 can reset the counter of Timer C.

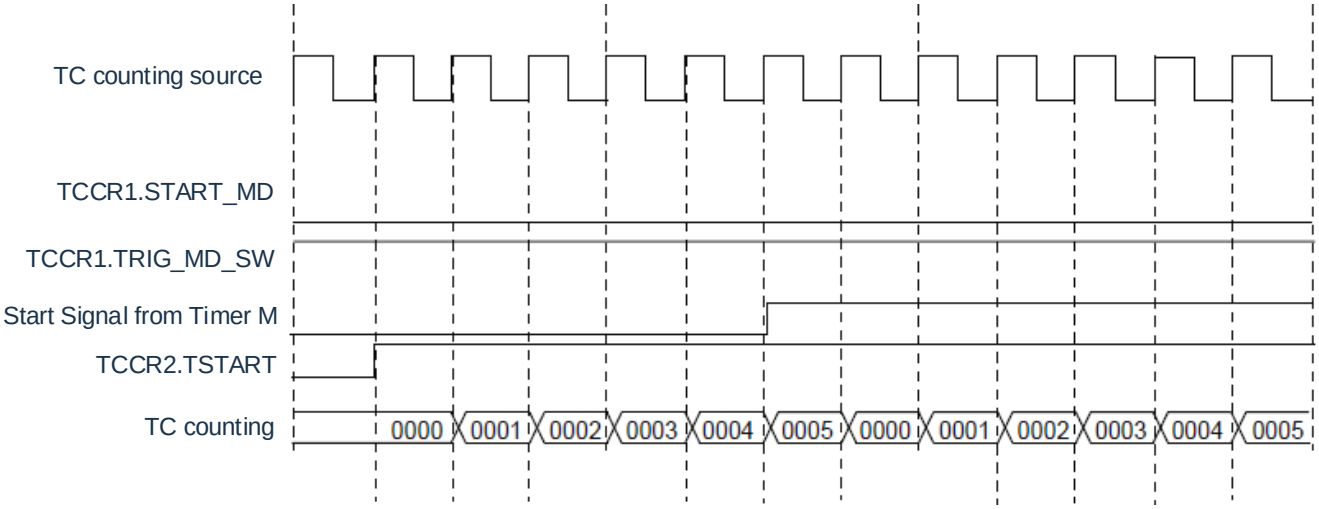
1. TRIG_MD_SW=0, the output signal of Timer M cannot reset the count value at the beginning of the software trigger count.
   1) Software trigger count start: TCCR1.START_MD=0
   2) Allow software reset counter: TCCR1.TRIG_MD_SW=0
   3) Timer C Count Start: TCCR2.TSTART=1

Figure 9-14 Example of Timer M triggering Timer C count reset



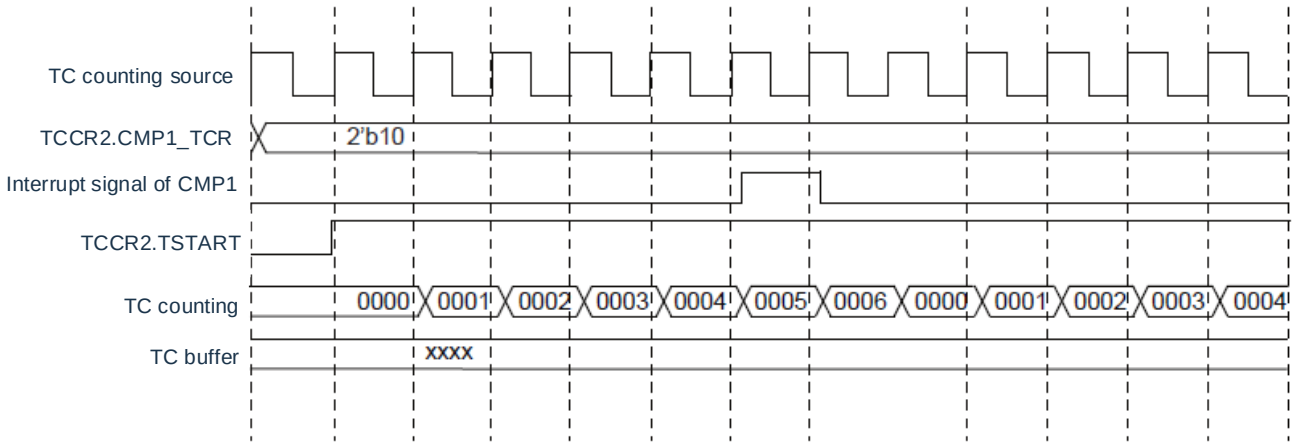
2. TRIG_MD_SW=1, when the software trigger count starts, the output signal of Timer M will reset the count value.
   1) Software trigger count start: TCCR1.START_MD=0
   2) Disable software reset counter: TCCR1.TRIG_MD_SW=1
   3) Timer C Count Start: TCCR2.TSTART=1

Figure 9-15 Example of Timer M triggering Timer C count reset



3. The output signal of CMP1 resets the count value at TCCR2.CMP1_TCR=10.
   1) Count value reset, count action continues: TCCR2.CMP1_TCR=10 (input capture is not allowed)
   2) Timer C Count Start: TCCR2.TSTART=1

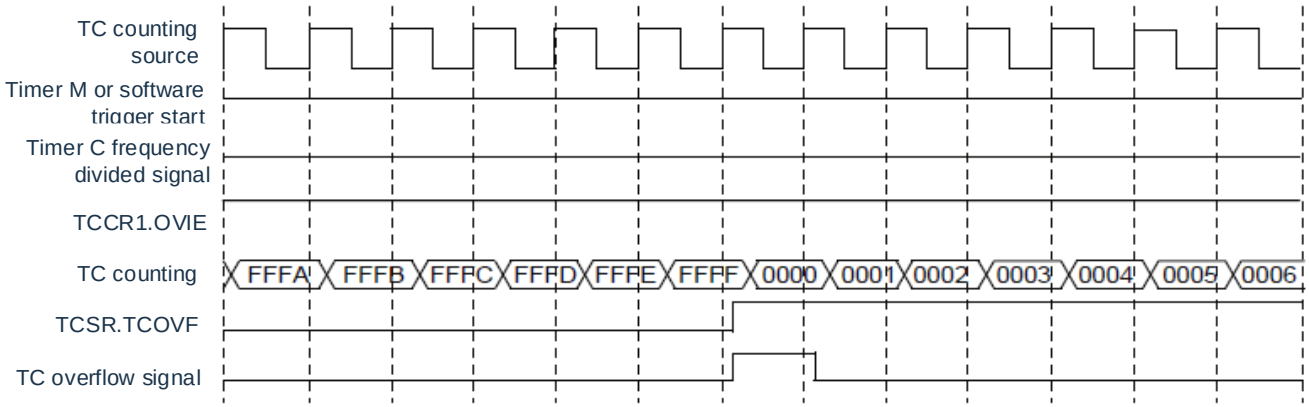
Figure 9-16 Example of CMP1 triggering a Timer C count reset



9.4.6 Interruption of timer C

When the counter of timer C overflows, if TCCR1.OVIE=1 is set, the overflowing interrupt signal will be generated.

Figure 9-17 Example of interrupt generation when Timer C overflows



9.5 Precautions for Using Timer C

9.5.1 Read/write of register

To set timer C, you must first set the TMCEN location "1" for PER1. When the TMCEN bit is '0', the write operation of the control register of timer C is ignored, and the read values are all initial values.

When the clock stops, other registers can read and write except that the TC and TCBUF registers cannot write.

Note: The following registers cannot be overridden in a count action.

- TCCR1
- TCCR2

9.5.2 overflow interrupt

When the count value of TimerC is FFFFH, an overflow interrupt does not occur if the counter is reset by the trigger event entered externally before the overflow reaches 0000 H.

9.5.3 Input capture and timer C count reset actions

Even if TCSR.TCSB=0 (count stop) is set, the input signals of Timer M and CMP1 trigger input capture and count reset actions of Timer C.

9.5.4 Steps when timer C and timer M and comparator 1 are linked

When the timer C and the timers M and the comparator 1 are linked, the setting steps are as follows:

1. Provides clock input for Comparator 1: PGACMPEN=1
2. Allow comparator 1 to interrupt generation and output: Refer to Chapter 17 Comparator CMP for more details.
3. Provides clock input for timer C: TMCEN=1
4. Set TCCR1
5. Set TCCR2
6. Set timer M, timer M count start: TMOEN=1
7. Timer C count start: TCCR2.START=1

Note 1:Timer C and Timer M, CMP1 must set the action clock of Timer C to match the action clock frequency of Timer M.

Note 2: When you set up a register, you must first set the control register TCCR1, and then set TCCR2.TSTART.

Chapter 10 Timer M

10.1 Function of timer M

Timer M has the following four modes:

- timer mode
- Enter capture: The count value is taken to the register, triggered by an external signal.
- Output Comparison: Checks if the count value and register value are the same (changes the output of the pin when the test is performed).
- PWM function: Continuously output any pulse width.

The following 3 modes use PWM functionality:

- Reset synchronous PWM mode: This is the mode of the three-phase waveform (6 of them) with output sawtooth modulation and no dead time.
- Complementary PWM mode: This is the output triangle wave modulation, three-phase waveform with dead time (6) mode.
- PWM3 Mode: This is the mode in which the same period PWM waveform (2) is output.

The timer M0 and the timer M1 have the same input capture function, output comparison function and PWM function in the timer mode.

Reset synchronous PWM mode, complementary PWM mode and PWM3 mode output waveform by the combination of counter and register of timer M0 and timer M1.

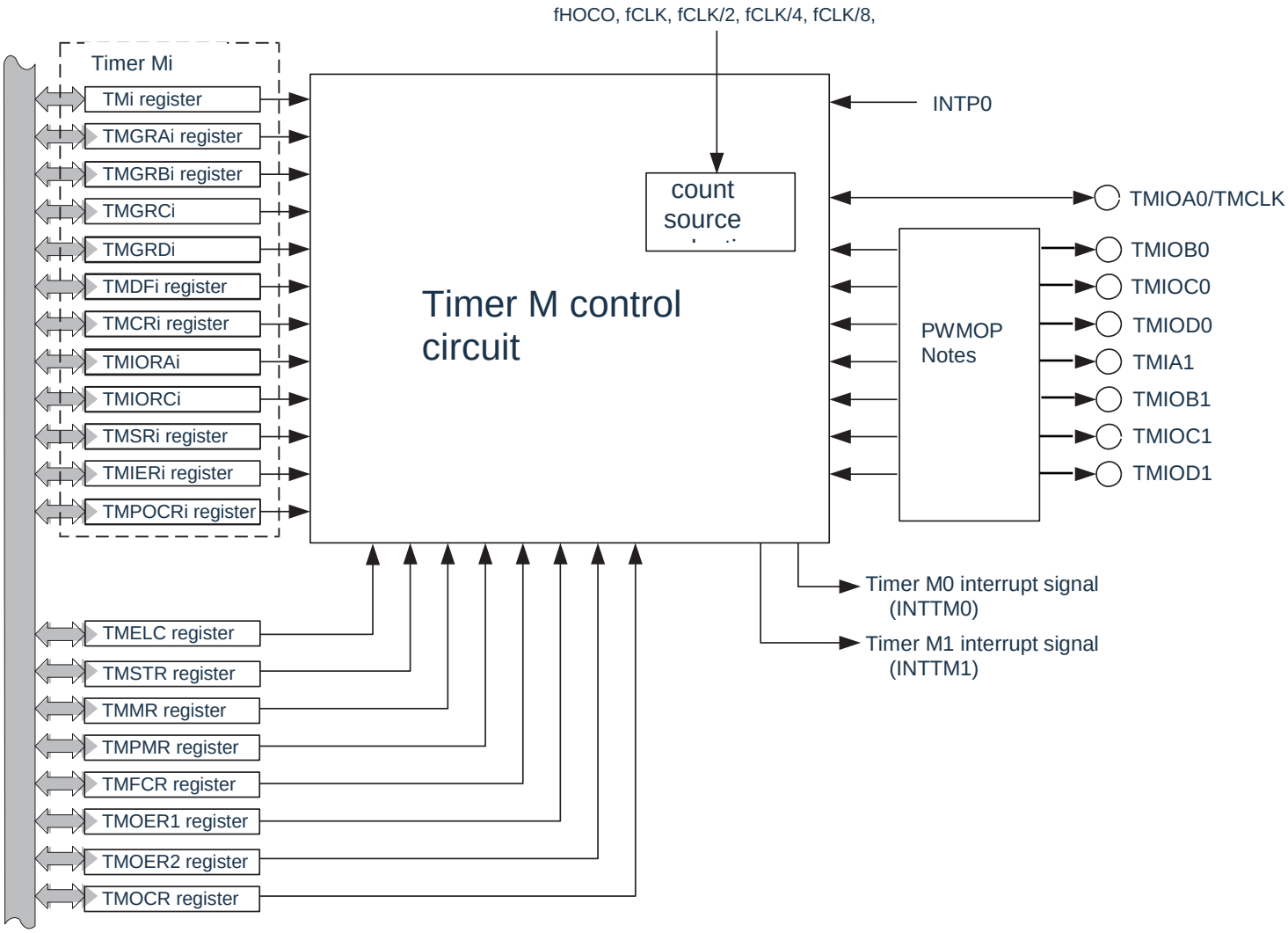
The timer M has four input/output pins.

The operating clock of the timer M is fCLK or fHOCO.

10.2 Structure of timer M

The block diagram and the pin structure of the timer M are shown in Figure 10-1 and Table 10-1 respectively.

Figure 10-1 Block diagram of timer M



Note: Can only cut off output

Remarks: i=0,1

Table 10-1 Pin structure of timer M

name of the feet	Port name for multiplexing	Input/Output	Features
TMIOA0/TMCLK	P17	Input/Output	Functions vary by mode of operation, see each mode for details.
TMIOB0	P14	Input/Output	
TMIOC0	P16	Input/Output	
TMIOD0	P1 5	Input/Output	
TMIA1	P12	Input/Output	
TMIOB1	P1 0	Input/Output	
TMIOC1	P1 3	Input/Output	
TMIOD1	P11	Input/Output	

10.3 Register for controlling timer M

The register for controlling the timer M is shown in Table 10-2.

Table 10-2 Register for controlling timer M

register name	symbol
Peripheral Enable Register 1	PER1
timer M EVENTC register	TMELC
timer M start register	TMSTR
timer M mode register	TMMR.
timer MPWM function selection register	TMPMR.
timer M function control register	TMFCR
Timer M outputs Master Allow Register 1	TMER1
Timer M outputs Master Allow Register 2	TMER2
timer M output control register	TMOCR
timer M digital filter function selection register 0	TMDF0
Timer M digital filter function selection register 1	TMDF1
timer M control register 0	TMCR0
Timer M I/O control register A0	TMIORA0
Timer M I/O control register C0	TMIORC0
Timer M Status Register 0	TMSR0
Timer M interrupt enable register 0	TMIR0
Timer M PWM function output level control register 0	TMPOCR0
Timer M Counter 0	TM0
Timer M General register A0	TMGRA0
timer m general register b0	TMGRB0
timer m general register c0	TMGRC0
timer m general register d0	TMGRD0
Timer M control register 1	TMCR1
Timer MI/O control register A1	TMIORA1
Timer MI/O control register C1	TMIORC1
Timer M Status Register 1	TMSR1
Timer M interrupt enable register 1	TMIR1
Timer M PWM function output level control register 1	TMPOCR1
Timer M Counter 1	TM1
Timer M General register A1	TMGRA1
Timer M General register B1	TMGRB1
timer m general register c1	TMGRC1
timer m general register d1	TMGRD1
port register	Pxx
port mode register	PMx, PMCx

10.3.1 Peripheral Enable Register 1 (PER1)

The PER1 register is a register that is set to allow or prohibit providing clocks to each peripheral hardware. Reduce power consumption and noise by stopping clock supply to unused hardware.

When you want to use timer M, you must set bit4(TMMEN) to '1'. The PER1 register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 10-2 Format of Peripheral Admission Register 1 (PER1)

Address: 0x400208 1A After reset:00H R/W

symbol	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN Notes	TMMEN	DMAEN	PWMPEN	TMCEN	TMAEN

TMMEN	Control of an input clock of a timer M is provided
0	Stop provide an input clock. · SFR used by timer M cannot be written. · The timer M is in a reset state.
1	Provides an input clock. · Can read and write the SFR used by the timer M.

PWMPEN	Control of input clock of PWM cut-off control circuit
0	Stop provide an input clock. · Cannot write the SFR used by the PWM cutoff circuit. · The PWM cut-off circuit is in a reset state.
1	Provides an input clock. · Can read and write SFR used by PWM cut-off circuit.

Note 1. To set the timer M, you must first set the TMMEN position "1". When the TMMEN bit is '0', the write operation of the control register of timer M is ignored, and the read values are initial (PM1 and P1).

2. To select fHOCO as the counter source for timer M, fCLK must be set to fIH before bit4(TMMEN) of peripheral enable register 1 (PER1). If you want to change fCLK to a clock other than fIH, you must change it after clearing bit4 (TMMEN) of Peripheral enable register 1 (PER1).

3. To set up PWMOP, you must first set the PWMOPEN location "1". When the PWMOPEN bit is '0', the write operation of the control register of the PWMOP is ignored and the read values are initial. See "10.8 PWMOP" for details.

10.3.2 timer M EVENTC register (TMELC)

Figure 10-3 Format of the timer MEVENTC register (TMELC)

Address: 0x40042A60 After Reset: 00H Note R/W

symbol	7	6	5	4	3	2	1	0
TMELC	0	0	ELCOBE1	ELCICE1	0	0	ELCOBE0	ELCICE0

ELCOBE1	Permission for EVENTC event input 1 (for forcing the cutoff of the pulse output
0	Forbids forced cutoff.
1	Allow forced cutoff.

ELCICE1	Selection of EVENTC event input 1 (input capture D1 for timer M)
0	Select Input Capture TMIOD1.
1	Select Event Input 1 from the Event Coordination Controller(EVENTC).

ELCOBE0	Permission for EVENTC event input 0 (used to force a cutoff of the pulse
0	Forbids forced cutoff.
1	Allow forced cutoff.

ELCICE0	Selection of EVENTC event input 0 (input capture D0 for timer M)
0	Select Input Capture TMIOD0.
1	Select event input 0 from the event Coordination Controller(EVENTC).

Note: The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

10.3.3 Timer M Start Register (TMSTR)

The TMSTR register can be set by an 8-bit memory operation instruction. Refer to the "10.7.1(1)TMSTR register" note when using timer M.

Figure 10-4 The format of the timer M start register (TMSTR)

Address: 0x40042A63 After reset: 0CH note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMSTR	0	0	0	0	CSEL1	CSEL0	TSTART1	TSTART0

CSEL1	TM1 Count Run Selection Note 2
0	Stops counting when comparing to a match with the TMGRA1 register.
1	Continue counting note 3 after matching with the TMGRA1 register comparisons.

CSEL0	Selection of TM0 Count Run
0	Stops counting when comparing to a match with the TMGRA0 register.
1	Continue counting note 3 after matching with the TMGRA0 register comparisons.

TSTART1	Start Flag for TM1 Count Note 4, 5
0	Stop counting.
1	Start counting.

TSTART0	TM0 Count Start Flag Note 6,7
0	Stop counting.
1	Start counting.

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. Cannot use PWM3 mode.
3. This location must be "1" when using the input capture feature.
4. You must write "0" to the CSEL1 bit when the TSTART1 bit is 1.
5. This flag is "0" when the CSEL1 bit is "0" and a comparison match signal (TMIOA1) is generated.
6. You must write "0" to the CSEL0 bit when the TSTART0 bit is 1.
7. This flag is "0" when the CSEL0 bit is "0" and a comparison match signal (TMIOA0) is generated.

10.3.4 timer M mode register (TMMR)

Figure 10-5 The format of the timer M mode register (TMMR)

Address: 0x40042A64 After reset: 00H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMMR	TMBFD1	TMBFC1	TMBFD0	TMBFC0	0	0	0	TMSYNC

TMBFD1	Selection of TMGRD1 Register Functions Note 2
0	universal register
1	Buffer register for TMGRB1 register

TMBFC1	Selection of TMGRC1 Register Functions Note 2
0	universal register
1	Buffer register for TMGRA1 register

TMBFD0	Selection of TMGRD0 Register Functions Note 2
0	universal register
1	Buffer register for TMGRB0 register

TMBFC0	Selection of TMGRC0 Register Features Note 2,3
0	universal register
1	Buffer register for TMGRA0 register

TMSYNC	Synchronization note 4 for timer M
0	TM0 and TM1 operate independently.
1	TM0 and TM1 run synchronously.

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. When using the output comparison function, if you select "0" (change TMGRji) by IOj3 (j=C or D) bits of the TMIORCi register (i=0,1

output pin of the register), the TMBFji position of the TMMR register must '0'.

3. In complementary PWM mode, this position "0" (universal register) is

required.

4. In reset synchronous PWM mode, complementary PWM mode, or PWM3 mode, this position '0' (TM0 and TM1).

10.3.5 Timer M PWM Function Selection Register (TMPMR)

Figure 10-6 Timer M PWM Function Selection Register (TMPMR) Format [Timer Mode]

Address: 0x40042A65 After Reset: 00H Note R/W

symbol	7	6	5	4	3	2	1	0
TMPMR.	0	TMPWMD1	TMPWMC1	TMPWMB1	0	TMPWMD0	TMPWMC0	TMPWMB0

TMPWMD1	The Choice of PWM Function of TMIOD1
0	input capture function or output comparison function
1	PWM function

TMPWMC1	The Choice of PWM Function of TMIOC1
0	input capture function or output comparison function
1	PWM function

TMPWMB1	The Choice of PWM Function of TMIOB1
0	input capture function or output comparison function
1	PWM function

TMPWMD0	The Choice of PWM Function of TMIOD0
0	input capture function or output comparison function
1	PWM function

TMPWMC0	The Choice of PWM Function of TMIOC0
0	input capture function or output comparison function
1	PWM function

TMPWMB0	The Choice of PWM Function of TMIOB0
0	input capture function or output comparison function
1	PWM function

Note: The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

10.3.6 Timer M Functional Control Register (TMFCR)

Figure 10-7 Format of timer M function control register (TMFCR)

Address: 0x40042A66 After reset: 80H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMFCR	PWM3	STCLK	0	0	OLS1	OLS0	CMD1	CMD0

PWM3	Selection of PWM3 Mode Note 2
· In timer mode, '1' must be set (non-PWM3 mode). · In PWM3 mode, '0' (PWM3 mode) must be set. · Invalid in reset synchronous PWM mode and complementary PWM mode.	

STCLK	External clock input selection
· Timer mode, reset synchronous PWM mode, complementary PWM mode 0: Invalid external clock input 1: External clock input is valid · In PWM3 mode, '0' must be set (external clock input is invalid).	

OLS1	Selection of inverting output levels (reset synchronous PWM mode or complementary PWM mode)
· Reset synchronous PWM mode, complementary PWM mode 0: Initial output "H" level, "L" level valid. 1: The initial output" L" level and "H" level are valid. · Not valid in timer mode and PWM3 mode.	

OLS0	Selection of positive phase output level (reset synchronous PWM mode or complementary PWM mode)
· Reset synchronous PWM mode, complementary PWM mode 0: Initial output "H" level, "L" level valid. 1: The initial output" L" level and "H" level are valid. · Not valid in timer mode and PWM3 mode.	

CMD1	CMD0	Selection Notes for Combined Modes Note 3,4
· In timer mode and PWM3 mode, "00B" (timer mode or PWM3 mode) must be set. · In reset synchronous PWM mode, "01B" must be set. · Complementary PWM mode CMD1 CMD0 1 0: Complementary PWM mode (transfers data from buffer registers to universal registers when TM1 underflows) 1 1 : Complementary PWM mode (transfers data from buffer registers to universal registers when TM0 and TMGRA0 registers match) Above: Disable setting.		

Note 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position '1'.

2. The PWM3 bit is set effectively when the CMD1 bit and the CMD0 bit are '00B' (timer or PWM3).

3. The CMD0 bit and the CMD1 bit must be written when both the TSTART0 bit and the TSTART1 bit of the TMSTR register are '0'.

4. When the CMD1 bit and CMD0 bit are "01B", "10B" or "11B, it is not related to the setting of the TMPMR register, it is reset synchronous PWM mode or complementary PWM mode.

10.3.7 Timer M outputs master permissive register 1 (TMOER1)

Figure 10-8 Timer M outputs the format of the master permissive register 1 (TMOER1)

[Output comparison function, PWM function, reset synchronous PWM mode, complementary PWM mode and PWM3 mode]

Address: 0x40042A67 After Reset: FFHnote 1 R/W

symbol	7	6	5	4	3	2	1	0
TMER1	ED1	EC1	EB1	EA1	ED0	EC0	EB0	EA0

ED1	Forbidden for TMIOD1 Output Note 2
0	Allow output.
1	Suppress output (TMIOD1 pin is an I/O port).

EC1	Forbidden for TMIOC1 Output Note 2
0	Allow output.
1	Output is disabled (TMIOC1 pin is I/O port).

EB1	Forbidden for TMIOB1 Output Note 2
0	Allow output.
1	Output is disabled (TMIOB1 pin is I/O port).

EA1	Disable for TMIOA1 output Note 2,3
0	Allow output.
1	Output is disabled (TMIOA1 pin is I/O port).

ED0	Forbidden for TMIOD0 Output Note 2
0	Allow output.
1	Output is disabled (TMIOD0 pin is I/O port).

EC0	Forbidden for TMIOC0 Output Note 2
0	Allow output.
1	Output is disabled (TMIOC0 pin is I/O port).

EB0	TMIOB0 Output Disallowed
0	Allow output.
1	Output is disabled (TMIOB0 pin is I/O port).

EA0	Disable Note for TMIOA0 Output Note 3,4
0	Allow output.
1	Output is disabled (TMIOA0 pin is I/O port).

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. In PWM3 mode, you must have this location "1".
3. When using the PWM feature, you must have this position "1".
4. In that reset synchronous PWM mode and the complementary PWM mode, this position" 1" must be use.

10.3.8 Timer M outputs Master Permissive Register 2 (TMOER2).

Figure 10-9 Timer M outputs the format of master permissive register 2 (TMOER2)
[PWM Function, Reset Synchronous PWM Mode, Complementary PWM Mode, and PWM3 Mode]

Address: 0x40042A68 After reset: 00^{H} note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMER2	TMPTO	0	0	0	0	0	0	TMSHUTS

TMPTO	Valid for Pulse Output Forced Cutoff Signal INTP0 Pin Input Note 2
0	The pulse output forces the cutoff input to be invalid.
1	The pulse output forces the cutoff input to be valid (the TMSHUTS bit is "1" if the INTP0 pin is given an "L" level).

TMSHUTS	Force Cutoff Flag
0	No forced cutoff.
1	At the mandatory cutoff.
When the cutoff pulse is forced by a INTP0 pin or an ELC input event, this bit becomes "1" and is not automatically cleared. Therefore, to stop the force cutoff pulse, you must write "0" to this bit during the stop count (TSTARTi=0). The TMSHUTS bit write" 1" is force off pulses even in an active mode.	

Note 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. Refer to "Forced cut off of 10.4.4 pulse output".

10.3.9 Timer M Output Control Register (TMOCR)

The TMOCR register must be written when both the TSTART0 bit and the TSTART1 bit of the TMSTR register are '0'.

Figure 10-10 Timer M Output Control Register (TMOCR) Format [Output Comparison Function]

Address: 0x40042A69 After reset: 00H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0

TOD1	Selection of the initial output level of TMIOD1 Note 2
0	Initial output "L" level.
1	Initial output "H" level.

TOC1	Selection of the initial output level of TMIOC1 Note 2
0	Initial output "L" level.
1	Initial output "H" level.

TOB1	Selection of the initial output level of TMIOB1 Note 2
0	Initial output "L" level.
1	Initial output "H" level.

TOA1	Selection of initial output level of TMIOA1
0	Initial output "L" level.
1	Initial output "H" level.

TOD0	Selection of the initial output level of TMIOD0 Note 2
0	Initial output "L" level.
1	Initial output "H" level.

TOC0	Selection of the initial output level of TMIOC0 Note 2
0	Initial output "L" level.
1	Initial output "H" level.

TOB0	Selection of the initial output level of TMIOB0 Note 2
0	Initial output "L" level.
1	Initial output "H" level.

TOA0	Selection of initial output level of TMIOA0
0	Initial output "L" level.
1	Initial output "H" level.

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. When the TMOCR register is set in the case that the pin function of the TMOCR register is waveform output, the initial output level is output.

Figure 10-11 Timer M Output Control Register (TMOCR) Format [PWM Function]

Address: 0x40042A69 After reset: 00^{H} note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0

TOD1	Selection of the initial output level of TMIOD1 Note 2
0	The initial output is an invalid level.
1	The initial output is the effective level.

TOC1	Selection of the initial output level of TMIOC1 Note 2
0	The initial output is an invalid level.
1	The initial output is the effective level.

TOB1	Selection of the initial output level of TMIOB1 Note 2
0	The initial output is an invalid level.
1	The initial output is the effective level.

TOA1	Selection of initial output level of TMIOA1
Must be set '0'.	

TOD0	Selection of the initial output level of TMIOD0 Note 2
0	The initial output is an invalid level.
1	The initial output is the effective level.

TOC0	Selection of the initial output level of TMIOC0 Note 2
0	The initial output is an invalid level.
1	The initial output is the effective level.
Effective in reset synchronous PWM mode and complementary PWM mode.	

TOB0	Selection of the initial output level of TMIOB0 Note 2
0	The initial output is an invalid level.
1	The initial output is the effective level.

TOA0	Selection of initial output level of TMIOA0
Must be set '0'.	

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. When the TMOCR register is set in the case that the pin function of the TMOCR register is waveform output, the initial output level is output.

Figure 10-12 Timer M Output Control Register (TMOCR) Format [PWM3 Mode]

Address: 0x40042A69 After reset: 00H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0

TOD1	Selection of initial output level of TMIOD1
Invalid in PWM3 mode.	

TOC1	Selection of initial output level of TMIOC1
Invalid in PWM3 mode.	

TOB1	Selection of initial output level of TMIOB1
Invalid in PWM3 mode.	

TOA1	Selection of initial output level of TMIOA1
Invalid in PWM3 mode.	

TOD0	Selection of initial output level of TMIOD0
Invalid in PWM3 mode.	

TOC0	Selection of initial output level of TMIOC0
Invalid in PWM3 mode.	

TOB0	Selection of the initial output level of TMIOB0 Note 2
0	The initial output" L" level and "H" level are valid. The "H" level is output when the TMGRB1 comparison match and the "L" level when the TMGRB0 comparison match.
1	Initial output "H" level, "L" level valid. Output "L" level when TMGRB1 comparison matches and "H" level when TMGRB0 comparison matches.

TOA0	Selection of initial output level of TMIOA0
0	The initial output" L" level and "H" level are valid. The "H" level is output when the TMGRA1 comparison match and the "L" level when the TMGRA0 comparison match.
1	Initial output "H" level, "L" level valid. Output "L" level when TMGRA1 comparison matches and "H" level when TMGRA0 comparison matches.

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. When the TMOCR register is set in the case that the pin function of the TMOCR register is waveform output, the initial output level is output.

10.3.10 timer m digital filter function selection register i (TMDFi) (i=0,1)

Figure 10-13 Timer M Digital Filter Function Selection Register i (TMDFi) (i=0,1) Format [Input capture]

Addresses: 0x40042A6A (TMDF0), 0x40042A6B (TMDF1) After reset: 00H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMDFi	DFCK1	DFCK0	PENB1	PENB0	DFD	DFC	DFB	DFA

DFCK1	DFCK0	Clock Selection for Digital Filter Functions Note 2
0	0	fCLK/32 Note 3
0	1	fCLK/8 Note 3
1	0	fCLK Note 3
1	1	Count source (TCK0~TCK2 bit selection clock for TMCRi register)

PENB1	PENB0	Control of Impulse Forced Cut-off of TMIOB Pin
0	0	Must be set "00B".

DFD	Selection of Digital Filter Function of TMIODi Pin
0	There is no digital filter function.
1	It has digital filter function.
When the digital filter function is available, the sampling clock cycle of up to 5 digital filters is needed for	

DFC	Selection of Digital Filter Function of TMIOCi Pin
0	There is no digital filter function.
1	It has digital filter function.
When the digital filter function is available, the sampling clock cycle of up to 5 digital filters is needed for	

DFB	Selection of Digital Filter Function of TMIOBi Pin
0	There is no digital filter function.
1	It has digital filter function.
When the digital filter function is available, the sampling clock cycle of up to 5 digital filters is needed for	

DFA	Selection of Digital Filter Function of TMIOAi Pin
0	There is no digital filter function.
1	It has digital filter function.
When the digital filter function is available, the sampling clock cycle of up to 5 digital filters is needed for	

Note 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. The count must start after setting the DFCK0 bit and the DFCK1 bit.

3. fCLK/32, fCLK/8 and fCLK are fHOCO/32, fHOCO/8 and fHOCO when the FRQSEL4 bit of the user option byte (000C2H) is 1.

Figure 10-14 Timer M Digital Filter Function Select the format of register i(TMDFi) (i=0,1)[PWM function, reset synchronous PWM mode, complementary PWM mode and PWM3 mode]

Addresses: 0x40042A6A (TMDF0), 0x40042A6B (TMDF1) After reset: 00H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMDFi	DFCK1	DFCK0	PENB1	PENB0	DFD	DFC	DFB	DFA

DFCK1	DFCK0	Control of Impulse Forced Cut-off of TMIOA Pin
0	0	Forbids forced cutoff.
0	1	high impedance output
1	0	"L" level output
1	1	"H" level output
If the corresponding pin is not used as the output port of the timer M in these modes, '00B' must be set (forbidding forced cutoff). Also, these bits must be set during stopping the count.		

PENB1	PENB0	Control of Impulse Forced Cut-off of TMIOB Pin
0	0	Forbids forced cutoff.
0	1	high impedance output
1	0	"L" level output
1	1	"H" level output
If the corresponding pin is not used as the output port of the timer M in these modes, '00B' must be set (forbidding forced cutoff). Also, these bits must be set during stopping the count.		

DFD	DFC	Control of Impulse Forced Cut-off of TMIOC Pin
0	0	Forbids forced cutoff.
0	1	high impedance output
1	0	"L" level output
1	1	"H" level output
If the corresponding pin is not used as the output port of the timer M in these modes, '00B' must be set (forbidding forced cutoff). Also, these bits must be set during stopping the count.		

DFB	DFA	Control of Impulse Forced Cut-off of TMIOD Pin
0	0	Forbids forced cutoff.
0	1	high impedance output
1	0	"L" level output
1	1	"H" level output
If the corresponding pin is not used as the output port of the timer M in these modes, '00B' must be set (forbidding forced cutoff). Also, these bits must be set during stopping the count.		

Note: The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

10.3.11 Timer M control register i (TMCRi) (i=0,1)

The TMCR1 register is not used in the reset synchronization PWM mode and the PWM3 mode.

Figure 10-15 Timer M Control Register i (TMCRi) (i=0,1) [Input capture and output comparison]

Addresses: 0x40042A70 (TMCR0), 0x40042A80 (TMCR1) After reset: 00H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMCRi	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	Clear Selection of TMi Counters
0	0	0	Do not clear (run free).
0	0	1	Clears when the TMGRAi's input capture/comparison matches.
0	1	0	Clears when the TMGRBi's input capture/comparison matches.
0	1	1	Synchronize Purge (Purge at the same time as the counter of other timer Mi) Note 1
1	0	1	Clears when the TMGRCi's input capture/comparison matches.
1	1	0	Clears when the TMGRDi's input capture/comparison matches.
Other than above			Disable setting.

CKEG1	CKEG0	External clock edge selection note 2
0	0	Count along the rising edge.
0	1	Count along the descent.
1	0	Count along double edges.
Other than above		Disable setting.

TCK2	TCK1	TCK0	Selection of Counting Source
0	0	0	fCLK, fHOCO Note3
0	0	1	fCLK/2
0	1	0	fCLK/4
0	1	1	fCLK/8
1	0	0	fCLK/32
1	0	1	Input for TMCLK note 4
Other than above			Disable setting.

Note:

1. Valid when the TMSYNC bit of the TMMR register is "1" (TM0 and TM1 run synchronously).
2. Valid when TCK2~TCK0 bit is '101B' (input of TMCLK) and STCLK bit is'1' (input of external clock is valid).
3. When the FRQSEL4 and FRQSEL3 bits of the user option byte (000C2H) are both 1, select fHOCO. For other values, select fCLK. To select fHOCO as the count source for timer M, you must set fCLK to fIH before setting bit4(TMMEN) of peripheral admissible register 1(PER1). If you want to change fCLK to a clock other than fIH, you must make the change after clearing bit4(TMMEN) of peripheral allow register 1(PER1).
4. Valid when the STCLK bit of the TMFCR register is "1" (External clock input is valid).

Figure 10-16 Timer M Control Register i (TMCRi) (i=0,1) Format [PWM]

Addresses: 0x40042A70 (TMCR0), 0x40042A80 (TMCR1) After reset: 00H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMCRi	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	Clear Selection of TMi Counters
Must be '001B'.			

CKEG1	CKEG0	External clock edge selection note 2
0	0	Count along the rising edge.
0	1	Count along the descent.
1	0	Count along double edges.
Other than above		Disable setting.

TCK2	TCK1	TCK0	Selection of Counting Source
0	0	0	fCLK, fHOCO Note 3
0	0	1	fCLK/2 Note 4
0	1	0	fCLK/4 Note 4
0	1	1	fCLK/8 Note 4
1	0	0	fCLK/32 Note 4
1	0	1	Input note 5 for TMCLK
Other than above			Disable setting.

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. Valid when TCK2~TCK0 bit is '101B' (input of TMCLK) and STCLK bit is'1' (input of external clock is valid).
3. When the FRQSEL4 bit of the user option byte (000C2H) is "0", select fCLK; Select fHOCO when the FRQSEL4 bit is "1" To select fHOCO as the counter source for timer M, fCLK must be set to fIH before bit4 (TMMEN) of Peripheral Admission Register 1 (PER1). If you want to change fCLK to a clock other than fIH, you must change it after clearing the bit4 (TMMEN) of peripheral enable register 1 (PER1).
4. Cannot set when the FRQSEL4 bit of user option byte (000C2H) is '1'.
5. Valid when the STCLK bit of the TMFCR register is "1" (External clock input is valid).

Figure 10-17 Timer M Control Register 0 (TMCR0) Format [Reset Synchronous PWM Mode]

Address: 0x40042A70 After reset: 00H note1 R/W

symbol	7	6	5	4	3	2	1	0
TMCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	Clear Selection of TM0 Counter
Must be '001B' (clear TM0 register when matching with TMGRA0 register).			

CKEG1	CKEG0	External clock edge selection note 2
0	0	Count along the rising edge.
0	1	Count along the descent.
1	0	Count along double edges.
Other than above		Disable setting.

TCK2	TCK1	TCK0	Selection of Counting Source
0	0	0	fCLK, fHOCO Note 3
0	0	1	fCLK/2 Note 4
0	1	0	fCLK/4 Note 4
0	1	1	fCLK/8 Note 4
1	0	0	fCLK/32 Note 4
1	0	1	Input note 5 for TMCLK
Other than above			Disable setting.

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. Valid when TCK2~TCK0 bit is '101B' (input of TMCLK) and STCLK bit is'1' (input of external clock is valid).
3. When the FRQSEL4 bit of the user option byte (000C2H) is "0", select fCLK; Select fHOCO when the FRQSEL4 bit is "1" To select fHOCO as the counter source for timer M, fCLK must be set to fIH before bit4 (TMMEN) of Peripheral Admission Register 1 (PER1). If you want to change fCLK to a clock other than fIH, you must change it after clearing the bit4 (TMMEN) of peripheral enable register 1 (PER1).
4. Cannot set when the FRQSEL4 bit of user option byte (000C2H) is '1'.
5. Valid when the STCLK bit of the TMFCR register is "1" (External clock input is valid).

Figure 10-18 Timer M Control Register 0 (TMCR0) Format [Complementary PWM Mode]

Address: 0x40042A70 After reset: 00H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	Clear Selection of TM0 Counter
Must set "000B".			

CKEG1	CKEG0	External clock edge selection note 2,3
0	0	Count along the rising edge.
0	1	Count along the descent.
1	0	Count along double edges.
Other than above		Disable setting.

TCK2	TCK1	TCK0	Selection of Counting Source
0	0	0	fCLK, fHOCO Note 4
0	0	1	fCLK/2 Note 5
0	1	0	fCLK/4 Note 5
0	1	1	fCLK/8 Note 5
1	0	0	fCLK/32 Note 5
1	0	1	Input for TMCLK Note 6
Other than above			Disable setting.

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. Valid when TCK2~TCK0 bit is '101B' (input of TMCLK) and STCLK bit is'1' (input of external clock is valid).

3. You must set the same values for TCK0~TCK2 bits, CKEG0 bits, and CKEG1 bits of the TMCR0 register and TMCR1 register.

4. When the FRQSEL4 bit of the user option byte (000C2H) is "0", select fCLK; Select fHOCO when the FRQSEL4 bit is "1" To select fHOCO as the counter source for timer M, fCLK must be set to fIH before bit4 (TMMEN) of Peripheral Admission Register 1 (PER1). If you want to change fCLK to a clock other than fIH, you must change it after clearing the bit4 (TMMEN) of peripheral enable register 1 (PER1).

5. Cannot set when the FRQSEL4 bit of user option byte (000C2H) is '1'.

6. Valid when the STCLK bit of the TMFCR register is "1" (External clock input is valid).

Figure 10-19 Timer M Control Register 0 (TMCR0) Format [PWM3 Mode]

Address: 0x40042A70 After reset: 00H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	Clear Selection of TM0 Counter
Must be '001B' (clear TM0 register when matching with TMGRA0 register).			

CKEG1	CKEG0	Selection of external clock edge
Invalid in PWM3 mode.		

TCK2	TCK1	TCK0	Selection of Counting Source
0	0	0	fCLK, fHOCO Note 2
0	0	1	fCLK/2 Note 3
0	1	0	fCLK/4 Note 3
0	1	1	fCLK/8 Note 3
1	0	0	fCLK/32 Note 3
Other than above			Disable setting.

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. When the FRQSEL4 bit of the user option byte (000C2H) is "0", select fCLK; Select fHOCO when the FRQSEL4 bit is "1" To select fHOCO as the counter source for timer M, you must set fCLK to fIH before the timer starts counting.

3. Cannot set when the FRQSEL4 bit of user option byte (000C2H) is '1'.

10.3.12 Timer MI/O control register Ai(TMIORAi) (i=0,1)

Figure 10-20 Timer MI/O Control Register Ai (TMIORAi) (i=0,1) Format [input capture]

Addresses: 0x40042A71 (TMIORA0), 0x40042A81H (TMIORA1) After reset: 00^{H} note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMIORAi	0	IOB2	IOB1	IOB0	0	IOA2	IOA1	IOA0

IOB2	Selection for TMGRB Mode Note 2
When using the input capture feature, you must set "1" (input capture).	

IOB1	IOB0	TMGRB control
0	0	Catch TMGRBi on the rising edge input.
0	1	Caught TMGRBi on descent input.
1	0	TMGRBi is captured along the input at the double edge.
Other than above		Disable setting.

IOA2	Select for TMGRA Mode Note 3
When using the input capture feature, you must set "1" (input capture).	

IOA1	IOA0	TMGRA control
0	0	Catch TMGRAi on the rising edge input.
0	1	Caught TMGRAi on descent input.
1	0	TMGRAi is captured along the input at the double edge.
Other than above		Disable setting.

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. If you select "1" through the TMBFDi bit of the TMMR register, you must set the same value for the IOB2 bit of the TMIORAi register.

3. If you select "1" through the TMBFCi bit of the TMMR register, you must set the same value for the IOA2 bit of the TMIORAi register.

Figure 10-21 Timer MI/O Control Register Ai (TMIORAi) (i=0,1) Format [output comparison]

Addresses: 0x40042A71 (TMIORA0), 0x40042A81H (TMIORA1) After reset: 00H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMIORAi	0	IOB2	IOB1	IOB0	0	IOA2	IOA1	IOA0

IOB2	Selection for TMGRB Mode Note 2
When you use the output comparison feature, you must set "0" (output comparison).	

IOB1	IOB0	TMGRB control
0	0	Prevent comparing matching pin outputs (I/O ports with TMIOBi pins).
0	1	Output "L" level when TMGRBi comparison matches.
1	0	Output "H" level when TMGRBi comparison matches.
1	1	Alternate output when TMGRBi compares matches.

IOA2	Select for TMGRA Mode Note 3
When you use the output comparison feature, you must set "0" (output comparison).	

IOA1	IOA0	TMGRA control
0	0	Prevent comparing matching pin outputs (I/O ports with TMIOAi pins).
0	1	Output "L" level when TMGRAi comparison matches.
1	0	Output "H" level when TMGRAi comparison matches.
1	1	Alternate output when TMGRAi compares matches.

Note 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. If you select "1" through the TMBFDi bit of the TMMR register, you must set the same value for the IOB2 bit of the TMIORAi register.

3. If you select "1" through the TMBFCi bit of the TMMR register, you must set the same value for the IOA2 bit of the TMIORAi register.

10.3.13 Timer MI/O control register Ci(TMIORCi) (i=0,1)

Figure 10-22 Timer MI/O Control Register Ci(TMIORCi) (i=0,1) Format [input capture]

Addresses: 0x40042A72 (TMIORC0), 0x40042A82 (TMIORC1) After reset: 88H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMIORCi	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0

IOD3	Selection of TMGRD Register Function
When using the input capture function, '1' (universal or buffer register) must be set.	

IOD2	Selection for TMGRD Mode Note 2
When using the input capture feature, you must set "1" (input capture).	

IOD1	IOD0	TMGRD control
0	0	Catch TMGRDi on the rising edge input.
0	1	Caught TMGRDi on descent input.
1	0	TMGRDi is captured along the input at the double edge.
Other than above		Disable setting.

IOC3	Selection of TMGRC Register Function
When using the input capture function, '1' (universal or buffer register) must be set.	

IOC2	Select for TMGRC Mode Note 3
When using the input capture feature, you must set "1" (input capture).	

IOC1	IOC0	TMGRC control
0	0	Catch TMGRCi on the rising edge input.
0	1	Caught TMGRCi on descent input.
1	0	TMGRCi is captured along the input at the double edge.
Other than above		Disable setting.

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. If you select "1" through the TMBFDi bit of the TMMR register, you must set the same value for the IOB2 bit of the TMIORAi register.

3. If you select "1" through the TMBFCi bit of the TMMR register, you must set the same value for the IOA2 bit of the TMIORAi register.

Figure 10-23 Timer M I/O Control Register Ci (TMIORCi) (i=0,1) [output comparison function]

Addresses: 0x40042A72 (TMIORC0), 0x40042A82 (TMIORC1) After reset: 88H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMIORCi	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0

IOD3	Selection of TMGRD Register Function
0	TMIOB output register (Refer to "Changes to the Output Pin of the 10.5.2 (2) TMGRCi Register and TMGRDi Register (i=0,1)")
1	Universal or buffer registers

IOD2	Selection for TMGRD Mode Note 2
When you use the output comparison feature, you must set "0" (output comparison).	

IOD1	IOD0	TMGRD control
0	0	Prevents comparing matching pin outputs.
0	1	Output "L" level when TMGRDi comparison matches.
1	0	Output "H" level when TMGRDi comparison matches.
1	1	Alternate output when TMGRDi compares matches.

IOC3	Selection of TMGRC Register Function
0	TMIOA output register (Refer to "Changes to the Output Pin of the 10.5.2 (2) TMGRCi Register and TMGRDi Register (i=0,1)")
1	Universal or buffer registers

IOC2	Select for TMGRC Mode Note 3
When you use the output comparison feature, you must set "0" (output comparison).	

IOC1	IOC0	TMGRC control
0	0	Prevents comparing matching pin outputs.
0	1	Output "L" level when TMGRCi comparison matches.
1	0	Output "H" level when TMGRCi comparison matches.
1	1	Alternate output when TMGRCi compares matches.

Note 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. If you select "1" through the TMBFDi bit of the TMMR register, you must set the same value for the IOB2 bit of the TMIORAi register.

3. If you select "1" through the TMBFCi bit of the TMMR register, you must set the same value for the IOA2 bit of the TMIORAi register.

10.3.14 Timer M Status Register 0 (TMSR0)

Figure 10-24 Timer M Status Register 0 (TMSR0) Format [Input Capture Function]

Address: 0x40042A73 After reset: 00H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMSR0	0	0	0	OVF	IMFD	IMFC	IMFB	IMFA

OVF	Overflow Flag Note 2
[Condition for '0'] Read and write "0" note 3. [Condition for "1"] When TM0 overflows	

IMFD	Enter Capture/Compare Match Flag D Note 6
[Condition for '0'] Read and write "0" note 3. [Condition for "1"] Input Edge Note 4 for TMIOD0 Pin	

IMFC	Enter Capture/Compare Match Flag C Note 6
[Condition for '0'] Read and write "0" note 3. [Condition for "1"] Input Edge Note 4 for TMIOC0 Pin	

IMFB	Enter Capture/Compare Match Flag B Note 6
[Condition for '0'] Read and write "0" note 3. [Condition for "1"] Input Edge Note 5 for TMIOB0 Pin	

IMFA	Enter Capture/Compare Match Flag A Note 6
[Condition for '0'] Read and write "0" note 3. [Condition for "1"] Input Edge Note 5 for TMIOA0 Pin	

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. When the count value of timer M0 changes from 'FFFFH' to '0000H', the overflow flag changes to '1'. In addition, according to the setting of CCLR0~CCLR2 bit of TMCR0 register, if the count value of timer M0 changes from FFFFH to 0000H.

Note: 3. The results are as follows:

- When writing "1," this bit does not change.
- In the case of reading "0", even writing "0" remains unchanged (even writing "0" remains).
- If the read value is "1", the bit becomes "0" if the same bit is given "0" However, when the status flag of an interrupt source of timer M (hereinafter referred to as the object status flag) is set to 0

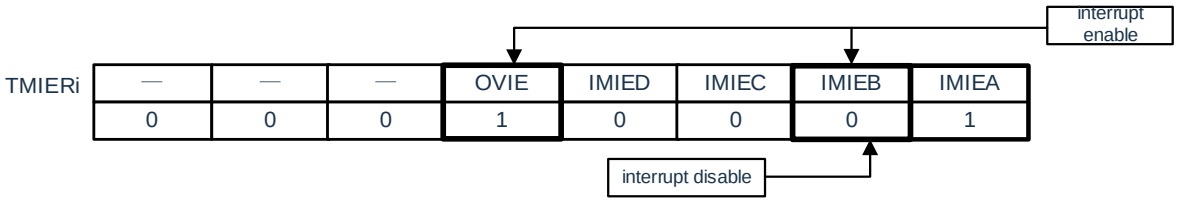
M interrupt enable register i (TMIERi) to be set to disable interrupt must be set" 0" in any of that follow (a) c.

a) The object status flag must be written "0" after setting timer M interrupt enable register i (TMIERi).

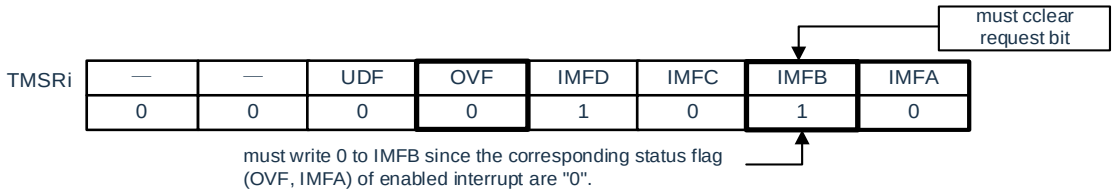
b) When timer M interrupt enable register i (TMIERi) has bit with" 1" (allowed) and the bit allow interrupt source state flag" 0.

(Example) Cleaning IMFB in a state where IMIEA and OVIE are interrupts allowed and IMIEB is interrupts prohibited

- Timer M interrupts the state of the allow register i (TMIERi)



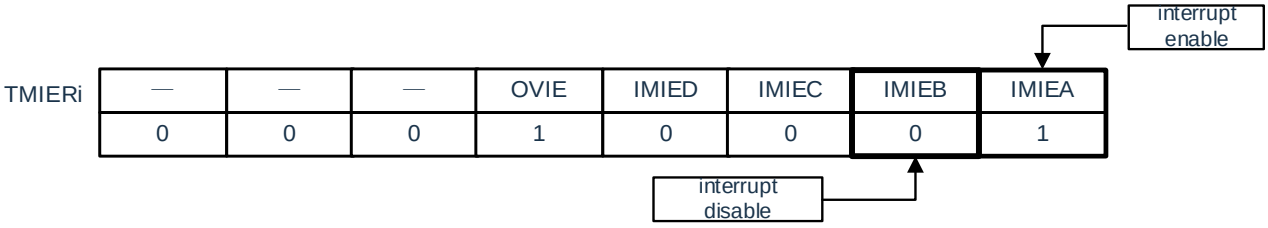
- Status of timer M status register i (TMSRi)



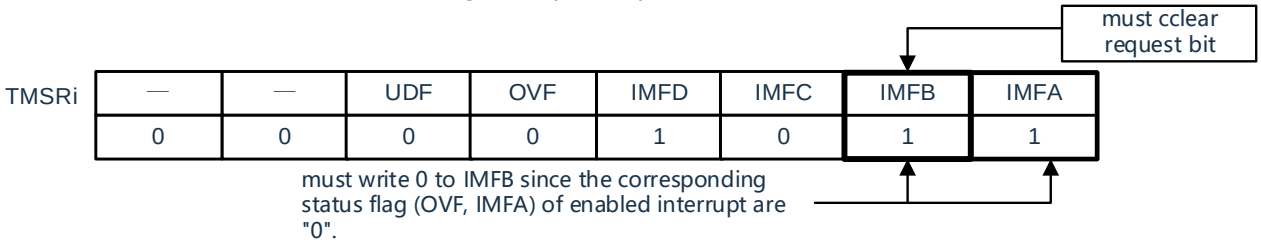
c) When timer M interrupt enable register i (TMIERi) has bit with '1' (allowed) and the bit allows interrupt source state flag '1'.

(Example) When IMIEA is Interrupt Allowed and IMIEB is Interrupt Disabled to clear IMFB

- Timer M interrupts the state of the allow register i (TMIERi)



- Status of timer M status register i (TMSRi)



4. This is the edge selected by the IOk1 bit and IOk0 bit (k=C or D) of the TMIORC0 register. Include that case where the TMBFk0 bit of the TMMR register is" 1" (TMGRk0 is a buffer register).
5. This is the edge selected by the IOj1 bit and IOj0 bit (j=A or B) of the TMIORA0 register.
6. When using DMA, the IMFA bits, IMFB bits, IMFC bits, and IMFD bits become "1" after the DMA transfer is completed.

Figure 10-25 Timer M Status Register 0 (TMSR0) Format [Features Other Than Input Capture]

Address: 0x40042A73 After reset: 00H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMSR0	0	0	0	OVF	IMFD	IMFC	IMFB	IMFA

OVF	Overflow Flag Note 3
[Condition for '0'] Read and write "0" note 2. [Condition for "1"] When TM0 overflows	

IMFD	Enter Capture/Compare Match Flag DNote 5
[Condition for '0'] Read and write "0" note 2. [Condition for "1"] When TM0 and TMGRD0 have the same values Note 4	

IMFC	Enter Capture/Compare Match Flag C Note 5
[Condition for '0'] Read and write "0" note 2. [Condition for "1"] When TM0 and TMGRC0 have the same values Note 4	

IMFB	Enter Capture/Compare Match Flag B Note 5
[Condition for '0'] Read and write "0" note 2. [Condition for "1"] When TM0 and TMGRB0 have the same value	

IMFA	Enter Capture/Compare Match Flag ANote 5
[Condition for '0'] Read and write "0" note 2. [Condition for "1"] When TM0 and TMGRA0 have the same value	

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. The results are as follows:

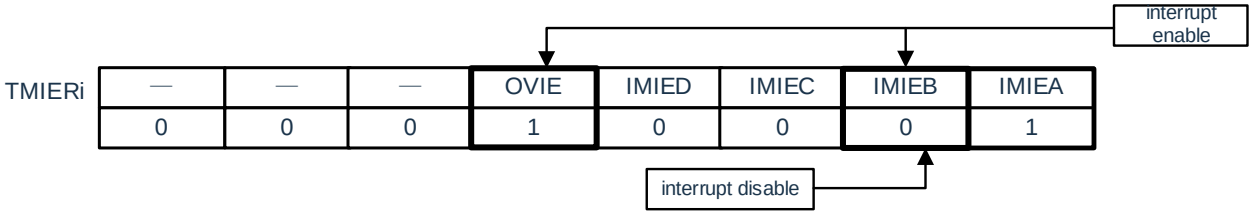
- When writing "1," this bit does not change.
- In the case of reading "0", even writing "0" remains unchanged (even writing "0" remains).
- If the read value is "1", the bit becomes "0" if the same bit is given "0"
  However, if the interrupt is set to 0 by timer M interrupt enable register i(TMIERi), then any of 0.

a) The object status flag must be written "0" after setting timer M interrupt enable register i (TMIERi).

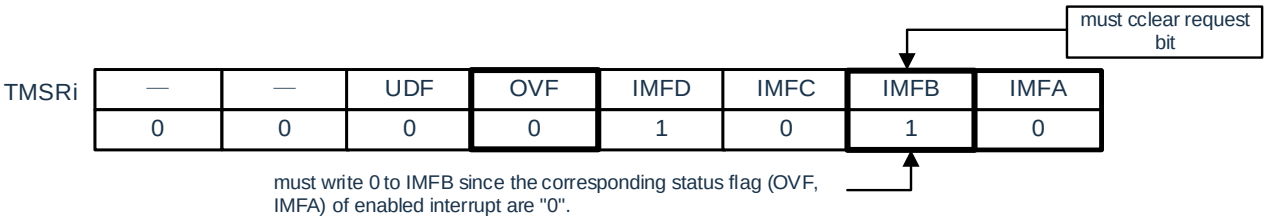
b) When timer M interrupt enable register i (TMIERi) has bit with" 1" (allowed) and the bit allow interrupt source state flag" 0.

(Example) Cleaning IMFB in a state where IMIEA and OVIE are interrupts allowed and IMIEB is interrupts prohibited

- Timer M interrupts the state of the allow register i (TMIERi)



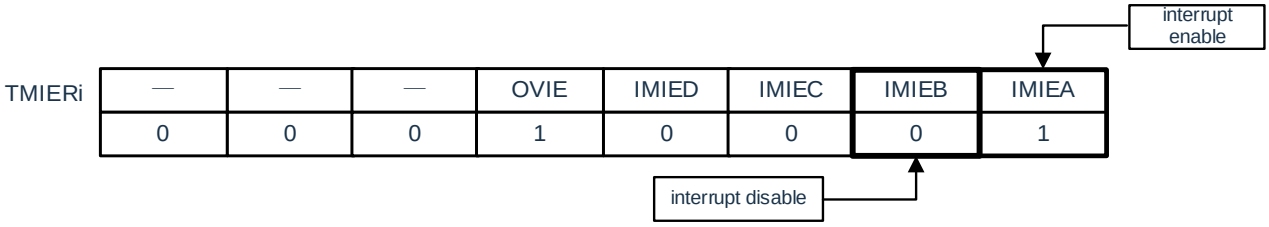
- Status of timer M status register i (TMSRi)



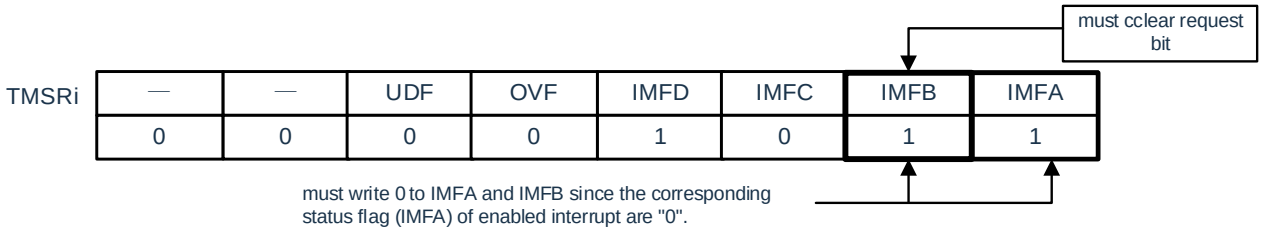
c) When timer M interrupt enable register i (TMIERi) has bit with '1' (allowed) and the bit allows interrupt source state flag '1'.

(Example) When IMIEA is Interrupt Allowed and IMIEB is Interrupt Disabled to clear IMFB

- Timer M interrupts the state of the allow register i (TMIERi)



- Status of timer M status register i (TMSRi)



3. When the count value of timer M0 changes from 'FFFFH' to '0000H', the overflow flag changes to '1'. In addition, according to the setting of CCLR0~CCLR2 bit of TMCR0 register, if the count value of timer M0 changes from FFFFH to 0000H.
4. Include that case where the TMBFk0 bit (k=C or D) of the TMMR register is '1' (TMGRk0 is a buffer register).
5. When using DMA, the IMFA bits, IMFB bits, IMFC bits, and IMFD bits become "1" after the DMA transfer is completed.

10.3.15 Timer M Status Register 1 (TMSR1)

Figure 10-26 Timer M Status Register 1 (TMSR1) Format [Input Capture Function]

Address: 0x40042A83 After reset: 00H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMSR1	0	0	UDF	OVF	IMFD	IMFC	IMFB	IMFA

UDF	overflow mark
Invalid when using input capture feature.	

OVF	Overflow Flag Note 2
[Condition for '0'] Read and write "0" note 3. [Condition for "1"] When TM1 overflows	

IMFD	Enter Capture/Compare Match Flag D Note 6
[Condition for '0'] Read and write "0" note 3. [Condition for "1"] Input Edge Note 4 for TMIOD1 Pin	

IMFC	Enter Capture/Compare Match Flag C Note 6
[Condition for '0'] Read and write "0" note 3. [Condition for "1"] Input Edge Note 4 for TMIOC1 Pin	

IMFB	Enter Capture/Compare Match Flag B Note 6
[Condition for '0'] Read and write "0" note 3. [Condition for "1"] Input Edge Note 5 for TMIOB1 Pin	

IMFA	Enter Capture/Compare Match Flag A Note 6
[Condition for '0'] Read and write "0" note 3. [Condition for "1"] Input Edge Note 5 for TMIOA1 Pin	

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

2. The overflow flag changes to '1' when the counter value of timer M1 changes from 'FFFFH' to '0000H'. In addition, according to the setting of CCLR0~CCLR2 bit of TMCR1 register, the overflow flag becomes '1' if FFFFH.

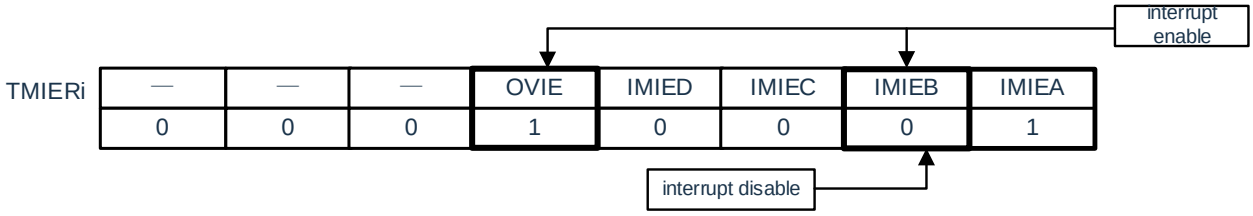
Note: 3. The results are as follows:

- When writing "1," this bit does not change.
- In the case of reading "0", even writing "0" remains unchanged (even writing "0" remains).
- If the read value is "1", the bit becomes "0" if the same bit is given "0"
  However, if the interrupt is set to 0 by timer M interrupt enable register i(TMIERi), then any of 0.

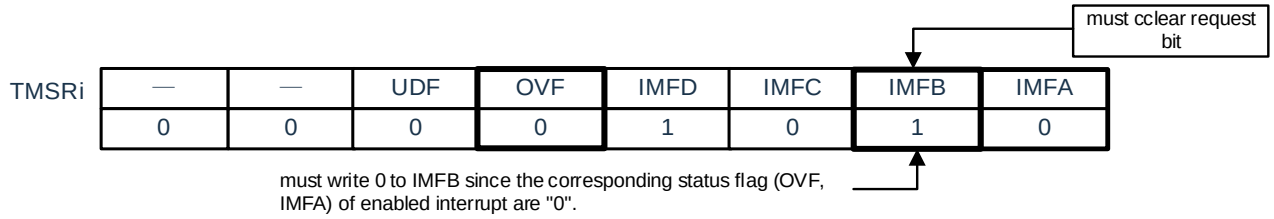
a) The object status flag must be written "0" after setting timer M interrupt enable register i (TMIERi).
b) When timer M interrupt enable register i (TMIERi) has bit with" 1" (allowed) and the bit allow interrupt source state flag" 0.

(Example) When IMIEA and OVIE are interrupts-allowed and IMIEB is interrupts-forbidden. IMFB is cleared.

- Timer M interrupts the state of the allow register i (TMIERi)



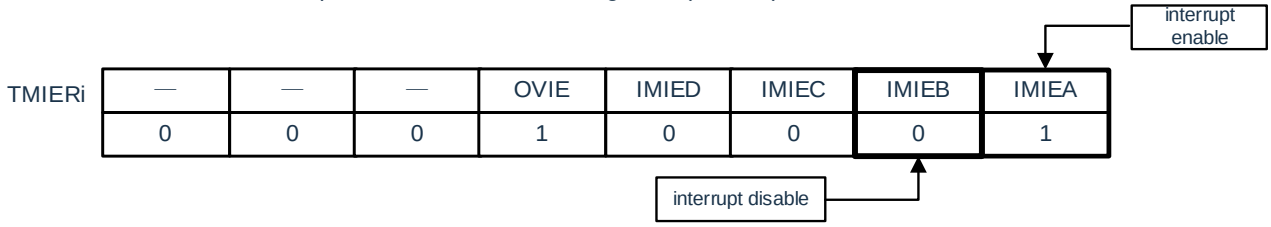
- Status of timer M status register i (TMSRi)



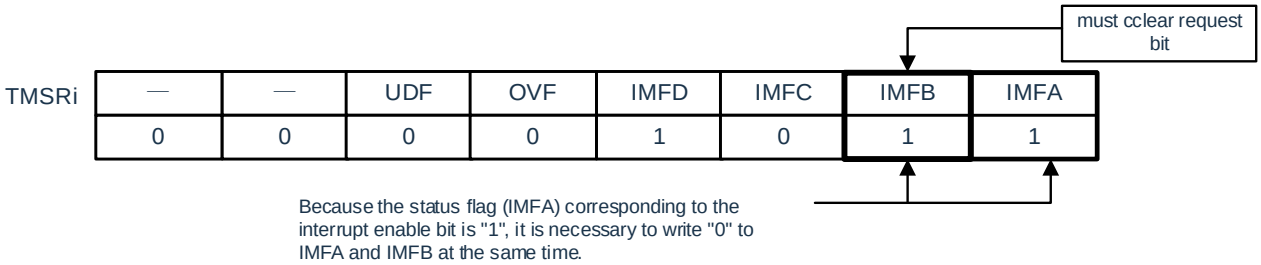
c) When timer M interrupt enable register i (TMIERi) has bit with '1' (allowed) and the bit allows interrupt source state flag '1'.

(Example) When IMIEA is an interrupt-allowed state and IMIEB is an interrupt-prohibited state, clear IMFB.

- Timer M interrupts the state of the allow register i (TMIERi)



- Status of timer M status register i (TMSRi)



4. This is the edge selected by the IOk1 bit and IOk0 bit (k=C or D) of the TMIORC1 register.
   Include that case where the TMBFk1 bit of the TMMR register is" 1" (TMGRk1 is a buffer register).
5. This is the edge selected by the IOj1 bit and IOj0 bit (j=A or B) of the TMIORA1 register.
6. When using DMA, the IMFA bits, IMFB bits, IMFC bits, and IMFD bits become "1" after the DMA transfer is completed.

Figure 10-27 Timer M Status Register 1 (TMSR1) Format [Input capture]

Address: 0x40042A83 After reset: 00H note 1 R/W

symbol	7	6	5	4	3	2	1	0
TMSR1	0	0	UDF	OVF	IMFD	IMFC	IMFB	IMFA

UDF	overflow mark
The case of complementary PWM mode [Condition for '0'] Read and write "0" note 2. [Condition for "1"] Invalid in non-complementary PWM mode when TM1 underflows.	

OVF	Overflow Flag Note 3
[Condition for '0'] Read and write "0" note 2. [Condition for "1"] When TM1 overflows	

IMFD	Enter Capture/Compare Match Flag D Note 5
[Condition for '0'] Read and write "0" note 2. [Condition for "1"] When TM1 and TMGRD1 have the same values Note5	

IMFC	Enter Capture/Compare Match Flag C Note 5
[Condition for '0'] Read and write "0" note 2. [Condition for "1"] When TM1 and TMGRC1 have the same values Note5	

IMFB	Enter Capture/Compare Match Flag B Note 5
[Condition for '0'] Read and write "0" note 2. [Condition for "1"] When TM1 and TMGRB1 have the same value	

IMFA	Enter Capture/Compare Match Flag A Note 5
[Condition for '0'] Read and write "0" note 2. [Condition for "1"] When TM1 and TMGRA1 have the same value	

Note: 1. The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

Note:2. The results are as follows:

- When writing "1," this bit does not change.
- In the case of reading "0", even writing "0" remains unchanged (even writing "0" remains).
- If the read value is "1", the bit becomes "0" if the same bit is given "0"

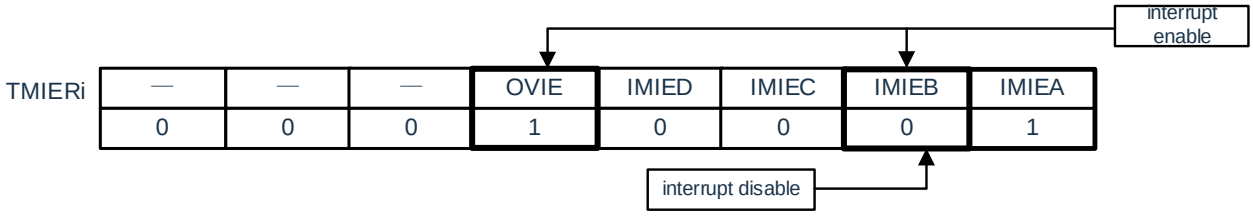
  However, if the interrupt is set to 0 by timer M interrupt enable register i(TMIERi), then any of 0.

a) The object status flag must be written "0" after setting timer M interrupt enable register i (TMIERi).

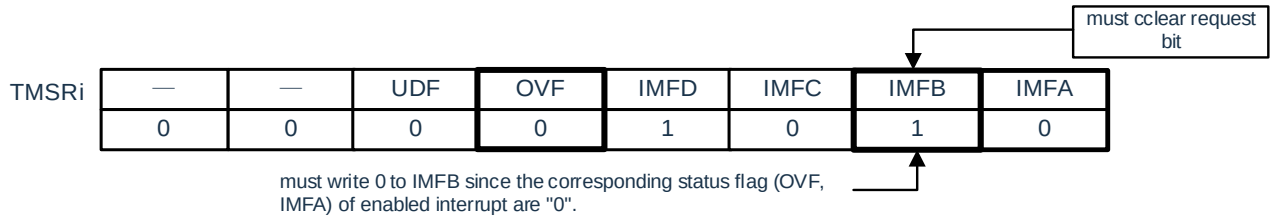
b) When timer M interrupt enable register i (TMIERi) has bit with" 1" (allowed) and the bit allow interrupt source state flag" 0.

(Example) Cleaning IMFB in a state where IMIEA and OVIE are interrupts allowed and IMIEB is interrupts prohibited

- Timer M interrupts the state of the allow register i (TMIERi)



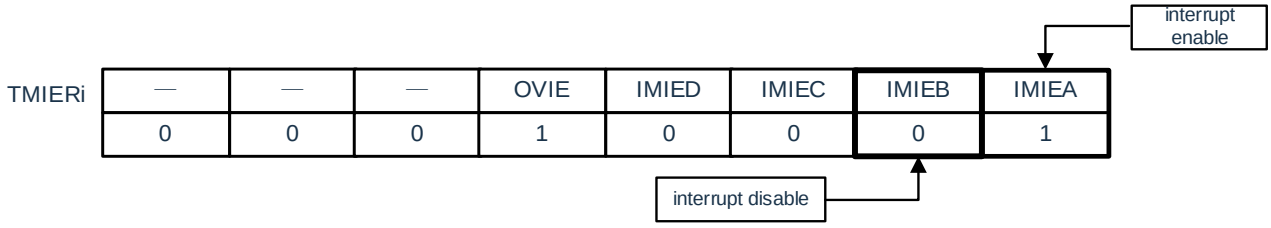
- Status of timer M status register i (TMSRi)



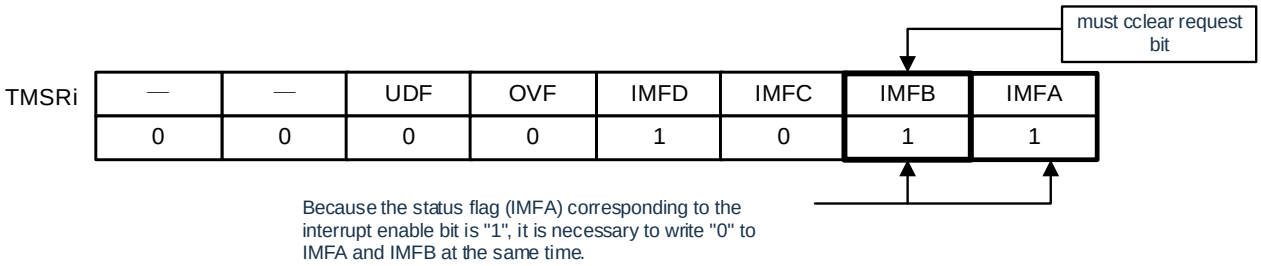
c) When timer M interrupt enable register i (TMIERi) has bit with '1' (allowed) and the bit allows interrupt source state flag '1'.

(Example) When IMIEA is Interrupt Allowed and IMIEB is Interrupt Disabled to clear IMFB

- Timer M interrupts the state of the allow register i (TMIERi)



- Status of timer M status register i (TMSRi)



3. The overflow flag changes to '1' when the counter value of timer M1 changes from 'FFFFH' to '0000H'. In addition, according to the setting of CCLR0~CCLR2 bit of TMCR1 register, the overflow flag becomes '1' if FFFFH.
4. Include the case where the TMBFk1 bit (k=C or D) of the TMMR register is a buffer register.
5. When using DMA, the IMFA bit, the IMFB bit, the IMFC bit, and the IMFD bit become "1" after the DMA transfer.

10.3.16 timer m interrupt enable register i (TMIERi) (i=0,1)

Figure 10-28 Timer M interrupts the format of register i(TMIERi) (i=0,1)

Addresses: 0x40042A74 (TMIER0), 0x40042A84 (TMIER1) After Reset: 00^{H} Note R/W

symbol	7	6	5	4	3	2	1	0
TMIRi	0	0	0	OVIE	IMIED	IMIEC	IMIEB	IMIEA

OVIE	Permission for overflow/overflow interrupts
0	Interrupts (OVIs) due to OVF and UDF bits are prohibited.
1	Allow interrupts (OVIs) due to OVF and UDF bits.

IMIED	Input capture/compare matching interrupt allowed D
0	Interrupts due to IMFD bits are prohibited (IMID).
1	Allow interrupts due to IMFD bits (IMID).

IMIEC	Input capture/compare matching interrupt allow C
0	Interrupts due to IMFC bits are prohibited (IMIC).
1	Allow interrupts due to IMFC bits (IMIC).

IMIEB	Input capture/compare matching interrupt allowed B
0	Interrupts due to IMFB bits are prohibited (IMIB).
1	Allow interrupts due to IMFB bits (IMIB).

IMIEA	Enter Snap/compare matching interrupt allowed A
0	Interrupts due to IMFA bits are prohibited (IMIA).
1	Allow interrupts due to IMFA bits (IMIA).

Note: The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

10.3.17 timer MPWM function output level control register i (TMPOCRi) (i=0,1)

The TMPOCRi register settings are valid only when using the PWM function, otherwise the TMPOCRi register settings are not valid.

Figure 10-29 Timer MPWM Function Output Level Control Register i (TMPOCRi) (i=0,1) Format [PWM]

Addresses: 0x40042A75 (TMPOCR0), 0x40042A85 (TMPOCR1) After Reset: $00^{H\ Note}$ R/W

symbol	7	6	5	4	3	2	1	0
TMPOCRi	0	0	0	0	0	COLD	POLC	POLB

COLD	Output level control D of PWM function
0	The TMIODi output level is a valid 'L' level.
1	The TMIODi output level is a valid 'H' level.

POLC	Output level control C of PWM function
0	The TMIOCi output level is a valid 'L' level.
1	The TMIOCi output level is a valid 'H' level.

POLB	Output level control B of PWM function
0	The TMIOBi output level is a valid 'L' level.
1	The TMIOBi output level is a valid 'H' level.

Note: The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

10.3.18 timer M counter i (TMi) (i=0,1)

[Timer Mode]
The TMi register must be accessed in 16 bits and not in 8 bits.

[Reset synchronous PWM mode and PWM3 mode]
The TM0 register must be accessed in 16 bits and not in 8 bits. The TM1 register is not used in reset synchronization PWM mode and PWM3 mode.

[Complementary PWM mode (TM0)]
The TM0 register must be accessed in 16 bits and not in 8 bits.

[Complementary PWM mode (TM1)]
The TM1 register must be accessed in 16 bits and not in 8 bits.

Figure 10-30 Format of timer M counter i(TMi) (i=0,1) [timer mode]

Address: 0x40042A76(TM0), 0x40042A86(TM1) After Reset: 0000H Note R/W

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

—	Features	Set Scope
bit15~0	Count the sources incrementally. If overflows occur, the OVF bit of the TMSRi register becomes "1".	0000H~FFFFH

Note: The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

Figure 10-31 Timer M Counter i(TMi) (i=0,1) Format [Reset Synchronous PWM Mode and PWM3 Mode]

Address: 0x40042A76(TM0), 0x40042A86(TM1) After Reset: 0000H Note R/W

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

—	Features	Set Scope
bit15~0	Count the sources incrementally. If overflows occur, the OVF bit of the TMSR0 register becomes "1".	0000H~FFFFH

Note: The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

Figure 10-32 Timer M Counter i(TMi) (i=0,1) Format [Complementary PWM mode (TM0)]

Address: 0x40042A76(TM0), 0x40042A86(TM1) After Reset: 0000H Note R/W

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

—	Features	Set Scope
bit15~0	You must set the time of the dead zone. The count source is counted incrementally or decrementally. If overflows occur, the OVF bit of the TMSR0 register becomes "1".	0001H~FFFFH

Note: The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

Figure 10-33 Timer M Counter i(TMi) (i=0,1) Format [Complementary PWM Mode (TM1)]

Address: 0x40042A76(TM0), 0x40042A86(TM1) After Reset: 0000H Note R/W

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

—	Features	Set Scope
bit15~0	Must be '0000H'. The count source is counted incrementally or decrementally. If overflows occur, the UDF bit of the TMSR1 register becomes "1".	0000H~FFFFH

Note: The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

10.3.19 timer m general register ai, bi, ci, di (TMGRAi, TMGRBi, TMGRCi, TMGRDi) (i=0,1)

[Enter capture feature]

The TMGRAi~TMGRDi register must be accessed in 16 bits and not in 8 bits. The following registers are invalid when using the input capture feature:

TMORE1, TMORE2, TMOCR, TMPOCR0,TMPOCR1

When a digital filter is not used (DFj bit of TMDFi register is '0'), the pulse width of the capture signal input by the TMIOji pin must be at least 3 $_{fCLK}$ clock cycles.

[Output Comparison Function]

The TMGRAi~TMGRDi register must be accessed in 16 bits and not in 8 bits. The following registers are not valid when using the output compare function:

TMDF0, TMDF1, TMPOCR0, TMPOCR1

[PWM Features]

The TMGRAi~TMGRDi register must be accessed in 16 bits and not in 8 bits. When using the PWM function, the following registers are invalid:

TMDF0, TMDF1, TMIORA0, TMIORC0,TMIORA1,TMIORC1

[Reset Synchronous PWM Mode]

The TMGRAi~TMGRDi register must be accessed in 16 bits and not in 8 bits. In reset synchronization PWM mode, the following registers are invalid:

TMPMR, $^{TMOCR\ Note,}$ TMDF0, TMDF1, TMIORA0, TMIORC0, TMPOCR0, TMIORA1,TMIORC1,TMPOCR1

Note: Only the TOC0 bit of the TMOCR register is valid as the initial output setting of the TMIOC0 in the reset synchronous PWM mode and complementary PWM mode.

[Complementary PWM Mode]

The TMGRAi~TMGRDi register must be accessed in 16 bits and not in 8 bits. In complementary PWM mode, TMGRC0 registers are not used.

In complementary PWM mode, the following registers are invalid:

TMPMR, $^{TMOCR\ Note,}$ TMDF0, TMDF1, TMIORA0, TMIORC0, TMPOCR0, TMIORA1,TMIORC1,TMPOCR1

Note: Only the TOC0 bit of the TMOCR register is valid as the initial output setting of the TMIOC0 in the reset synchronous PWM mode and complementary PWM mode.

Since TMGRB0, TMGRA1, TMGRB1 registers (forbidden) cannot be written immediately after counting, TMGRD0, TMGRC1,TMGRD1 registers must be used as buffer registers.

However, when writing TMGRD0, TMGRC1, TMGRD1 registers, write them after TMBFD0 bits, TMBFC1 bits, and TMBFD1 position "0" (generic register). Thereafter, the TMBFD0 bit, TMBFC1 bit and TMBFD1 position "1" (buffer register) are enabled.

[PWM3 Mode]

The TMGRAi~TMGRDi register must be accessed in 16 bits and not in 8 bits. In PWM3 mode, the following registers are invalid:

TMPMR, TMDF0, TMDF1, TMIORA0, TMIORC0, TMPOCR0, TMIORA1, TMIORC1, TMPOCR1In PWM3 mode, TMGRC0, TMGRC1, TMGRD0, TMGRD1 registers are not used. However, to use these registers as buffer registers, write values to the TMGRC0, TMGRC1, TMGRD0,TMGRD1 registers after TMBFC0, TMBFC1, TMBFD0, and TMBFD1 locations "0". Thereafter, TMBFC0 bits, TMBFC1 bits, TMBFD0 bits and TMBFD1 positions "1" (buffer registers) can be used.

Figure 10-34 timer m general register ai, bi, ci, di

(TMGRAi, TMGRBi, TMGRCi, TMGRDi) (i=0,1) in [Input capture] format

Address: 0x40042A78 (TMGRA0), 0x40042A7A (TMGRB0), reset:FFFFH note R/W

0x40042B58 (TMGRC0), 0x40042B5A (TMGRD0),

0x40042A88 (TMGRA1), 0x40042A8A (TMGRB1),

0x40042B5C (TMGRC1), 0x40042B5E (TMGRD1)

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMGRAi TMGRBi TMGRCi TMGRDi																

—	Features
bit15~0	Refer to Table 10-3 TMGRji register function when using the input capture function.

Note: The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

Table 10-3 TMGRji register function when using input capture function

register	Settings	register function	Input pin for input capture
TMGRAi	—	Generic register that reads the value of the TMi register when the input is captured.	TMIOI
TMGRBi			TMIOBi
TMGRCi	TMBFCi=0	Generic register that reads the value of the TMi register when the input is captured.	TMIOCi
TMGRDi	TMBFDi=0		TMIODi
TMGRCi	TMBFCi=1	Generic register that reads the value of the TMi register when the input is captured (reference "10.4.2 Buffer Run").	TMIOI
TMGRDi	TMBFDi=1		TMIOBi

Note: i=0,1,j=A,B,C,D

TMBFCi, TMBFDi:bit of the TMMR register

Figure 10-35 timer m general register ai, bi, ci, di (TMGRAi, TMGRBi, TMGRCi, TMGRDi) (i=0,1) format [output comparison function]

Address: 0x40042A78 (TMGRA0), 0x40042A7A (TMGRB0), After reset: FFFFH Note R/W
0x40042B58 (TMGRC0), 0x40042B5A (TMGRD0), 0x40042A88 (TMGRA1), 0x40042A8A (TMGRB1), 0x40042B5C mgrc1), 0x40042B5E (TMGRD1)

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	10
TMGRAi															
TMGRBi															
TMGRCi															
TMGRDi															

—	Features
bit15~0	Refer to Table 10-4 TMGRji register function when using output comparison function.

Note: The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

Table 10-4 TMGRji register function when using output comparison function

register	Settings		register function		Output pin for output comparison
	TMBFki	IOj3			
TMGRAi	—	—	Universal register, must write compare value.		TMIOI
TMGRBi					TMIOBi
TMGRCi	0	1	Universal register, must write compare value.		TMIOCi
TMGRDi					TMIODi
TMGRCi	1	1	buffer register, must write next compare value (Refer to "10.4.2 Buffer Operation").		TMIOI
TMGRDi					TMIOBi
TMGRCi	0	0	TMIOAi output control	(Refer to "Changes to the Output Pin of the 10.5.2 (2) TMGRCi Register and TMGRDi Register (i=0,1)"	TMIOI
TMGRDi			TMIOBi output control		TMIOBi

Note: If the TCK2~TCK0 position of the TMCRi register is '000B' ($_{fCLK, fHOCO)}$ and the comparison value is '0000H', request signals are generated to the DMA and the ELC only 11 times immediately after the start of counting. If that compare value is greater than or equal to" 0001H", a request signal is generate each time the compare matches.

Note: i=0,1, j=A, B, C, D,k=C,D
TMBFki:bit of TMMR register, bit of IOj3:TMIORCi register

Figure 10-36 timer m general register ai, bi, ci, di
(TMGRAi, TMGRBi, TMGRCi, TMGRDi) (i=0,1) format [PWM]

Address: 0x40042A78 (TMGRA0), 0x40042A7A (TMGRB0), reset:FFFFH [note] R/W
0x40042B58 (TMGRC0), 0x40042B5A (TMGRD0),
0x40042A88 (TMGRA1), 0x40042A8A (TMGRB1),
0x40042B5C (TMGRC1), 0x40042B5E (TMGRD1)

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMGRAi																
TMGRBi																
TMGRCi																
TMGRDi																

—	Features
bit15~0	Refer to Table 10-5 TMGRji register function when using PWM function.

Note: The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

Table 10-5 TMGRji register function when using PWM function

register	Settings	register function	PWM output pin
TMGRAi	—	Universal register, PWM cycle must be set.	—
TMGRBi	—	Universal register, must set the change point of the PWM output.	TMIOBi
TMGRCi	TMBFCi=0	Universal register, must set the change point of the PWM output.	TMIOCi
TMGRDi	TMBFDi=0		TMIODi
TMGRCi	TMBFCi=1	Buffer register, must set the next PWM cycle (reference "10.4.2 buffer operation").	—
TMGRDi	TMBFDi=1	Buffer register, must set the change point of the next PWM output (refer to "10.4.2 buffer operation").	TMIOBi

Note: If the TCK2~TCK0 position of the TMCRi register is '000B' ($_{fCLK, fHOCO)\ and\ the\ comparison\ value}$ is '0000H', request signals are generated to the DMA and the ELC only 11 times immediately after the start of counting. If that compare value is greater than or equal to" 0001H", a request signal is generate each time the compare matches.

Note: i=0,1,j=A,B,C,D
TMBFCi, TMBFDi:bit of the TMMR register

Figure 10-37 timer m general register ai, bi, ci, di
(TMGRAi, TMGRBi, TMGRCi, TMGRDi) (i=0,1) format [reset synchronous PWM mode]

Address: 0x40042A78 (TMGRA0), 0x40042A7A (TMGRB0), reset:FFFFH note R/W
0x40042B58 (TMGRC0), 0x40042B5A (TMGRD0),
0x40042A88 (TMGRA1), 0x40042A8A (TMGRB1),
0x40042B5C (TMGRC1), 0x40042B5E (TMGRD1)

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMGRAi TMGRBi TMGRCi TMGRDi																

—	Features
bit15~0	Refer to Table 10-6 Reset the TMGRji register function in synchronous PWM mode.

Note The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

Table 10-6 TMGRji register function in reset synchronous PWM mode

register	Settings	register function	PWM output pin
TMGRA0	—	Universal register, PWM cycle must be set.	(TMIOC0 at each PWM Periodic Invert)
TMGRB0	—	Universal register, must set the change point of PWM1 output.	TMIOB0T MIOD0
TMGRC0	TMBFC0=0	(not used in reset synchronization PWM mode)	—
TMGRD0	TMBFD0=0		
TMGRA1	—	Universal register, must set the change point of PWM2 output.	TMIOA1T MIOC1
TMGRB1	—	Universal register, must set the change point of PWM3 output.	TMIOB1 TMIOD1
TMGRC1	TMBFC1=0	(not used in reset synchronization PWM mode)	—
TMGRD1	TMBFD1=0		
TMGRC0	TMBFC0=1	Buffer register, must set the next PWM cycle (reference "10.4.2 buffer operation").	(TMIOC0 at each PWM Periodic Invert)
TMGRD0	TMBFD0=1	buffer register, must set the change point of the next PWM1 output (reference) "10.4.2 Buffer Run").	TMIOB0T MIOD0
TMGRC1	TMBFC1=1	buffer register, must set the change point of the next PWM2 output (reference) "10.4.2 Buffer Run")	TMIOA1T MIOC1
TMGRD1	TMBFD1=1	buffer register, must set the change point of the next PWM3 output (reference) "10.4.2 Buffer Run")	TMIOB1 TMIOD1

Note: If the TCK2~TCK0 position of TMCR0 register is '000B' (fCLK, fHOCO) and the comparison value is '0000H'. If that compare value is greater than or equal to" 0001H", a request signal is generate each time the compare matches.

Note: i=0,1,j=A,B,C,D
Bit of TMBFC0, TMBFD0, TMBFC1, TMBFD1:TMMR register

Figure 10-38 timer m general register ai, bi, ci, di
(TMGRAi, TMGRBi, TMGRCi, TMGRDi) (i=0,1) format [complementary PWM mode]

Address: 0x40042A78 (TMGRA0), 0x40042A7A (TMGRB0), reset:FFFFH [note] R/W
0x40042B58 (TMGRC0), 0x40042B5A (TMGRD0),
0x40042A88 (TMGRA1), 0x40042A8A (TMGRB1),
0x40042B5C (TMGRC1), 0x40042B5E (TMGRD1)

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMGRAi																
TMGRBi																
TMGRCi																
TMGRDi																

—	Features
bit15~0	Refer to Table 10-7 Complementary TMGRji register function in PWM mode.

Note: The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

Table 10-7 TMGRji register function in complementary PWM mode

register	Settings	register function	PWM output pin
TMGRA0	—	Generic register, PWM period must be set at the initial setting. Setting Range: Setting Value (Initial Count) for TM0 Register ⩽ Set Value for TMGRA0 ⩽ Set value for the FFFFH-TM0 register When the TSTART0 bit and the TSTART1 bit of the TMSTR register are "1" (Start Count), this register cannot be written.	(TMIOC0 inverts every half cycle)
TMGRB0	—	Generic register, which must set the change point of the PWM1 output at the initial setting. Setting Range: Setting Value (Initial Count) for TM0 Register ⩽ Setting value for TMGRB0 ⩽ Setting value for TMGRA0 - Setting value for TM0 register When the TSTART0 bit and the TSTART1 bit of the TMSTR register are "1" (Start Count), this register cannot be written.	TMIOB0TMIOD0
TMGRA1	—	Generic register, which must set the change point of the PWM2 output at the initial setting. Setting Range: Setting Value (Initial Count) for TM0 Register ⩽ Setting value for TMGRA1 ⩽ Setting value for TMGRA0 - Setting value for TM0 register When the TSTART0 bit and the TSTART1 bit of the TMSTR register are "1" (Start Count), this register cannot be written.	TMIOA1TMIOC1
TMGRB1	—	Generic register, which must set the change point of the PWM3 output at the initial setting. Setting Range: Setting Value (Initial Count) for TM0 Register ⩽ Setting value for TMGRB1 ⩽ Setting value for TMGRA0 - Setting value for TM0 register When the TSTART0 bit and the TSTART1 bit of the TMSTR register are "1" (Start Count), this register cannot be written.	TMIOB1 TMIOD1
TMGRC0	—	(not used in complementary PWM mode)	—
TMGRD0	TMBFD0=1	Buffer register, must set the change point of the next PWM1 output (reference "10.4.2 buffer operation"). Setting Range: Setting Value (Initial Count) for TM0 Register ⩽ Setting value for TMGRD0 ⩽ Setting value for TMGRA0 - Setting value for TM0 register The same value of the TMGRB0 register must be set at the initial set-up.	TMIOB0TMIOD0
TMGRC1	TMBFC1=1	Buffer register, must set the change point of the next PWM2 output (reference "10.4.2 buffer operation"). Setting Range: Setting Value (Initial Count) for TM0 Register ⩽ Setting value for TMGRC1 ⩽ Setting value for TMGRA0 - Setting value for TM0 register The same value of the TMGRA1 register must be set at the initial set-up.	TMIOA1TMIOC1
TMGRD1	TMBFD1=1	Buffer register, must set the change point of the next PWM3 output (reference "10.4.2 buffer operation"). Setting Range: Setting Value (Initial Count) for TM0 Register ⩽ Setting value for TMGRD1 ⩽ Setting value for TMGRA0 - Setting value for TM0 register The same value of the TMGRB1 register must be set at the initial set-up.	TMIOB1 TMIOD1

Note: If the TCK2~TCK0 position of the TMCRi register is '000B' ($_{fCLK, fHOCO)}$ and the comparison value is '0000H', request signals are generated to the DMA and the ELC only 11 times immediately after the start of counting. If that compare value is greater than or equal to" 0001H", a request signal is generate each time the compare matches.

Note: i=0,1,j=A,B,C,D
Bit of TMBFD0, TMBFC1, TMBFD1:TMMR register

Figure 10-39 timer m general register ai, bi, ci, di
(TMGRAi, TMGRBi, TMGRCi, TMGRDi) (i=0,1) format [PWM3 mode]

Address: 0x40042A78 (TMGRA0), 0x40042A7A (TMGRB0), reset:FFFFH [note] R/W
0x40042B58 (TMGRC0), 0x40042B5A (TMGRD0), 0x40042A88 (TMGRA1), 0x40042A8A (TMGRB1),
0x40042B5C mgrc1), 0x40042B5E (TMGRD1)

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMGRAi TMGRBi TMGRCi TMGRDi																

—	Features
bit15~0	Refer to TMGRji register function in Table 10-8PWM3 mode.

Note: The reset value is indefinite if the FRQSEL4 bit of the user option byte (000C2H) is '1' and the TMMEN bit of the PER1 register. When you need to read an initial value, you must read the initial value after setting fCLK to fIH and TMMEN position "1".

Table 10-8 TMGRji register function in PWM3 mode

register	Settings	register function	PWM output pin
TMGRA0	—	Universal register, PWM cycle must be set. Setting Range: Setting Value for the TMGRA1 Register ⩽ Setting Value for the TMGRA0	TMIA0
TMGRA1		Universal register, must set the change point of the PWM output (the time sequence to the effective level). Setting Range: Setting Value for TMGRA1 ⩽ Setting Value for the TMGRA0 Register	
TMGRB0		Universal register, which must set the change point of the PWM output (the timing back to the initial output level). Setting Range: Setting value for the TMGRB1 register ⩽ Setting value for the TMGRB0 ⩽ Setting value for the TMGRA0 register	TMIOB0
TMGRB1		Universal register, must set the change point of the PWM output (the time sequence to the effective level). Setting Range: Setting Value for TMGRB1 ⩽ Setting Value for the TMGRB0 Register	
TMGRC0	TMBFC0=0	(not used in PWM3 mode)	—
TMGRC1	TMBFC1=0		
TMGRD0	TMBFD0=0		
TMGRD1	TMBFD1=0		
TMGRC0	TMBFC0=1	Buffer register, must set the next PWM cycle (reference "10.4.2 buffer operation"). Setting Range: Setting Value for the TMGRC1 Register ⩽ Setting Value for the TMGRC0	TMIA0
TMGRC1	TMBFC1=1	Buffer register, must set the change point of the next PWM output (refer to "10.4.2 buffer operation"). Setting Range: Setting Value for TMGRC1 ⩽ Setting Value for the TMGRC0 Register	
TMGRD0	TMBFD0=1	Buffer register, must set the change point of the next PWM output (refer to "10.4.2 buffer operation"). Setting Range: TMGRD1 Register ⩽ Setting value of TMGRD0 ⩽ Setting value for the TMGRC0 register	TMIOB0
TMGRD1	TMBFD1=1	Buffer register, must set the change point of the next PWM output (refer to "10.4.2 buffer operation"). Setting Range: Setting Value for TMGRD1 ⩽ Setting Value for the TMGRD0 Register	

Note: If the TCK2~TCK0 position of TMCR0 register is '000B' (fCLK, fHOCO) and the comparison value is '0000H'. If that compare value is greater than or equal to" 0001H", a request signal is generate each time the compare matches.

Note: i=0,1,j=A,B,C,D
Bit of TMBFC0, TMBFD0, TMBFC1, TMBFD1:TMMR register

10.3.20 Port Mode Register (PMxx, PMCxx)

This is a register that sets the input/output of the port or the analog input.

When the multiplexed ports (Pxx/TMIOD1, Pxx/TMIOC1, etc.) of the timer output pin are used as the output of the timer, the bits of the port mode register (PMxx, PMCxx) and Pxx must be

(Example) When P10/TMIOD1 is used as timer output

Position "0" PM10 and PMC10 for port mode register 1.

Position P10 of port register 1 at "0.

When using the multiplexed ports (P10/TMIOD1, P11/TMIOC1, etc.) of the timer input pin as the input of the timer, the position of the port mode register (PMCxx) must The bit of the port register (Pxx) can be "0" or "1".

(example) the case where P10/TMIOD1 is used as timer input

Position "1" for PM10 for port mode register 1.

Position PMC10 for port mode register 1 at "0.

Position P10 of port register 1 at "0" or "1.

The PM1 and PMC1 registers are set by the 8-bit memory operation instruction. After the reset signal is generated, the value of the register changes to "FFH".

Figure 10-40 Format for Port Mode Register 1 (PM1, PMC1)

Address: 0x40040321 After Reset: FFH R/W

symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PMmn	Selection of Input/Output Modes of Pmn Pins (m=1, n=0 ~7)
0	Output Mode (Output Buffer ON)
1	Input mode (output buffer OFF)

Address: 0x40040061 After Reset: FFH R/W

symbol	7	6	5	4	3	2	1	0
PMC1	PMC17	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10

PMCmn	Selection of Input/Output Modes of Pmn Pins (m=1, n=0~7)
0	Digital Input/Output (multiplexing function other than analog input)
1	analog input

10.4 Common Issues on Multiple Models

10.4.1 count source

The counting source selection method is the same for all modes. However, in PWM3 mode, you cannot select an external clock.

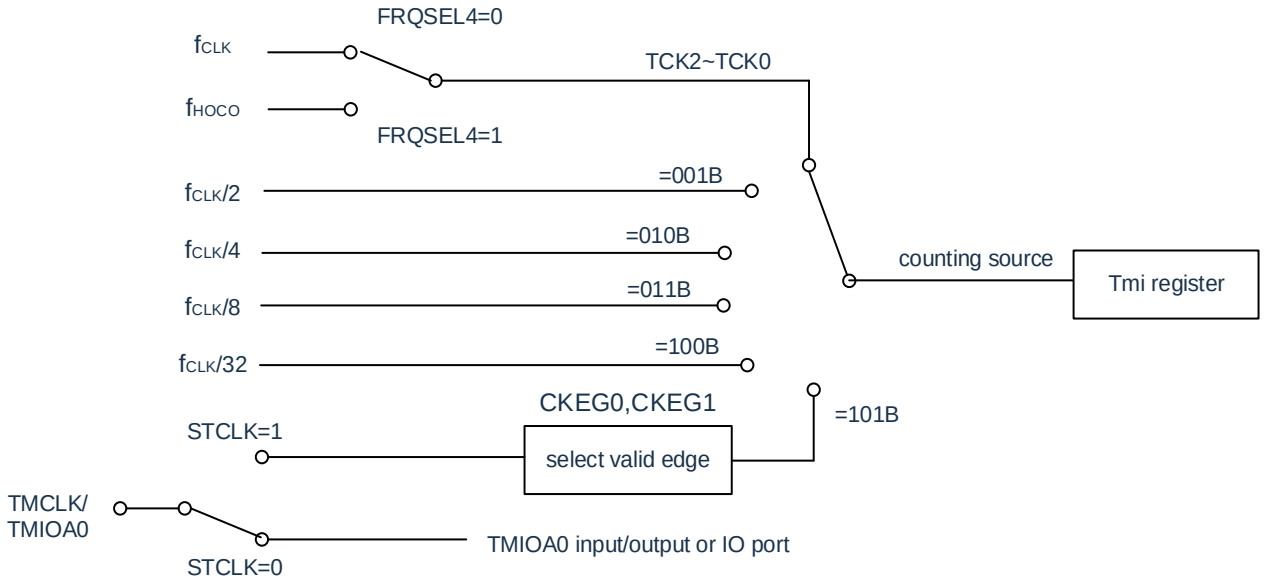
Table 10-9 Selection of Counting Source

count source	selection method
f_{CLK}、f_{HOCO}[note]、$f_{CLK}/2$、$f_{CLK}/4$、$f_{CLK}/8$、$f_{CLK}/32$	The counting source is selected by the TCK2~TCK0 bits of the TMCRi register.
External Input Signal for TMCLK Pin	The STCLK bit of the TMFCR register is "1" (external clock input is valid). The TCK2~TCK0 bit of the TMCRi register is '101B' (count source is external clock). The effective edge is selected by the CKEG1~CKEG0 bits of the TMCRi register. The port mode register for the multiplexing I/O port of the TMCLK pin has the bit '1'

Note: i=0,1

NOTE: Select fCLK when the FRQSEL4 bit of the user option byte (000C2H) is " 0 Select fHOCO when the FRQSEL4 bit is "1" To select fHOCO as the counter source for timer M, fCLK must be set to fIH before bit4 (TMMEN) of Peripheral Admission Register 1 (PER1). If you want to change fCLK to a clock other than fIH, you must change it after clearing the bit4 (TMMEN) of peripheral enable register 1 (PER1).

Figure 10-41 Block diagram of count source



Note: i=0,1

The external clock input by the TMCLK pin must have a pulse width of at least 3 Runtime Clock (f_{CLK}) cycles.

10.4.2 buffer operation

The TMGRCi register and the TMGRDi register can be set as the buffer register of the TMGRAi register and the TMGRBi register through the TMBFCi (i=0, 1) and TMBFDi bits of the TMMR register.

- TMGRAi buffer register: TMGRCi register
- TMGRBi buffer register: TMGRDi register

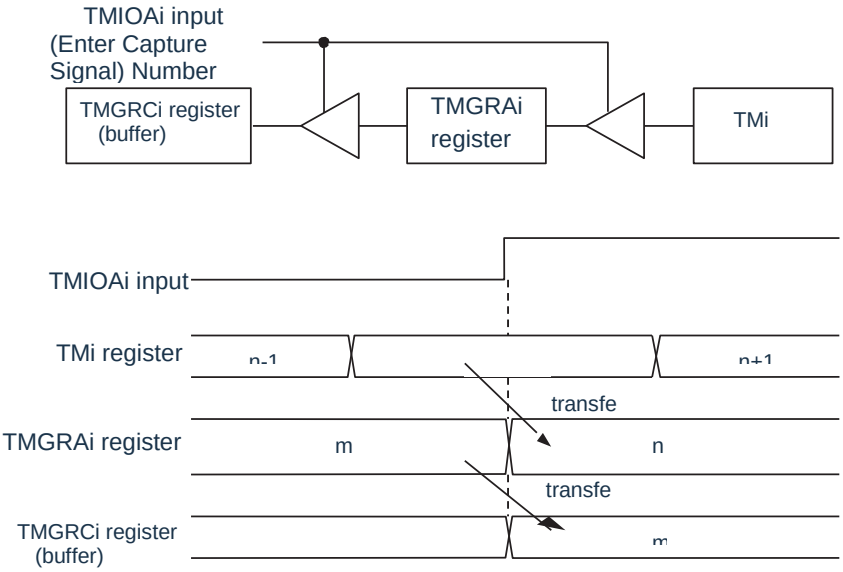
The buff run varies depending on that timer mode, and the buff run for each mode is shown in Table 10-10.

Table 10-10. Buffer operation by mode

Features and Modes		Transfer Sequence	Transferred register
timer mode	Input capture	Input signal for TMIOAi (Input to capture signal)	The contents of the TMGRAi register are transferred to the TMGRCi register (buffer register).
		Input signal for TMIOBi (Input to capture signal)	The contents of the TMGRBi register are transferred to the TMGRDi register (buffer register).
	output comparison function	Comparison Matching of TMi Register and TMGRAi Register	Transfers the contents of the TMGRCi register (buffer register) to the TMGRAi register.
		Comparison Matching of TMi Register and TMGRBi Register	Transfers the contents of the TMGRDi register (buffer register) to the TMGRBi register.
	PWM function	Comparison Matching of TMi Register and TMGRAi Register	Transfers the contents of the TMGRCi register (buffer register) to the TMGRAi register.
		Comparison Matching of TMi Register and TMGRBi Register	Transfers the contents of the TMGRDi register (buffer register) to the TMGRBi register.
Reset synchronous PWM mode		Comparison and Matching of TM0 Register and TMGRA0 Register	Transfers the contents of the TMGRCi register (buffer register) to the TMGRAi register. Transfers the contents of the TMGRDi register (buffer register) to the TMGRBi register.
complementary PWM mode		• The underflow of the TM1 register when the CMD1 and CMD0 bits of the TMFCR register are "11B". • When the CMD1 bit and CMD0 bit of the TMFCR register are "10B", the comparison match between the TM0 register and the	Transfer the contents of the TMGRC1 register (buffer register) to the TMGRA1 register. Transfers the contents of the TMGRDi register (buffer register) to the TMGRBi register.
PWM3 Mode		Comparison and Matching of TM0 Register and TMGRA0 Register	Transfers the contents of the TMGRCi register (buffer register) to the TMGRAi register. Transfers the contents of the TMGRDi register (buffer register) to the TMGRBi register.

Note: i=0,1

Figure 10-42 Buffer Operation for Input Capture

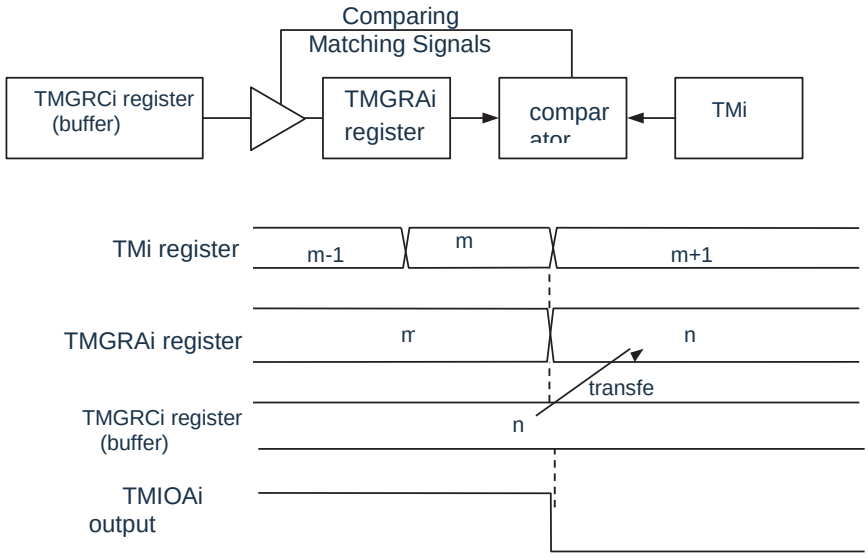


Notes: i=0,1

The conditions in the above diagram are as follows:

- The TMBFCi bit of the TMMR register is "1" (the TMGRCi register is a buffer register for the TMGRAi register)
- The IOA2 IOA0 bit ¬ of the TMIORAi register is "100B"

Figure 10-43 Buffer operation for output comparison function



Notes: i=0,1

The conditions in the above diagram are as follows:

- The TMBFCi bit of the TMMR register is "1" (the TMGRCi register is a buffer register for the TMGRAi register)
- The IOA2¬IOA0 bit of the TMIORAi register is "001B" (output "L" level)

In timer mode (input capture function and output comparison function), the following settings must be made.

Use TMGRCi (i=0,1) registers as buffer registers for TMGRAi registers:

- The IOC3 location of the TMIORCi register must be "1" (universal or buffer register).
- You must set the same value for the IOC2 bit of the TMIORCi register and for the IOA2 bit of the TMIORAi register.

Use of TMGRDi registers as buffer registers for TMGRBi registers:

- The IOD3 location of the TMIORCi register must be "1" (universal or buffer register).
- You must set the same value for the IOD2 bit of the TMIORCi register and for the IOB2 bit of the TMIORAi register.

When using the input capture function, even if the TMGRCi register and the TMGRDi register are used as buffer registers, the IMFC bit and the IMFD bit of the TMSRi register become "1" at the input edges of the TMIOCi pin and TMIODi pin.

When the output comparison function or PWM function is used or in reset synchronous PWM mode, complementary PWM mode and PWM3 mode, even if the TMGRCi register and the TMGRDi register are used as buffer registers, when compare with TMi register, the IMFC bit and IMFD bit of TMSRi register also become '1'.

10.4.3 synchronous operation

Synchronize the TM0 register with the TM1 register.

- synchronization preset

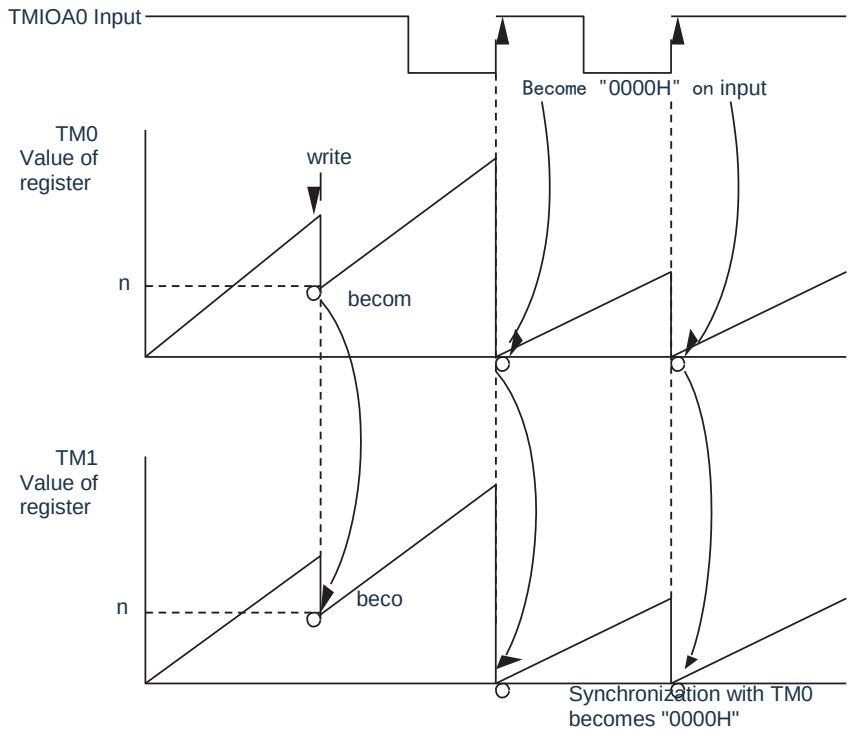
If the TMi register is written when the TMSYNC bit of the TMMR register is "1", the data is written to both the TM0 register.

- Synchronous Purge

When the TMSYNC bit of the TMMR register is "1" and the CCLR2~CCLR0 bit of the TMCR0 register is "011B".

Similarly, when the TMSYNC bit of the TMMR register is "1" and the CCLR2~CCLR0 bit of the TMCR1 register is "011B".

Figure 10-44 synchronous operation



The conditions in the above diagram are as follows:

The TMSYNC bit of the TMMR register is "1"
(synchronous operation)
The CCLR2~CCLR0 bit of the TMCR0 register is '001B' (set the TM0 to '0000H')
The CCLR2~CCLR0 bit of the TMCR1 register is "011B" (synchronized with TM0 to "0000H")
The IOA2~IOA0 bit of the TMIORA0 register is '100B'
The CMD1 and CMD0 bits of the TMFCR register are "00B"
The PWM3 bit of the TMFCR register is "1"

} (Capture on the rising edge of the TMIOA0 input)

10.4.4 Forced Cutoff of Pulse Output

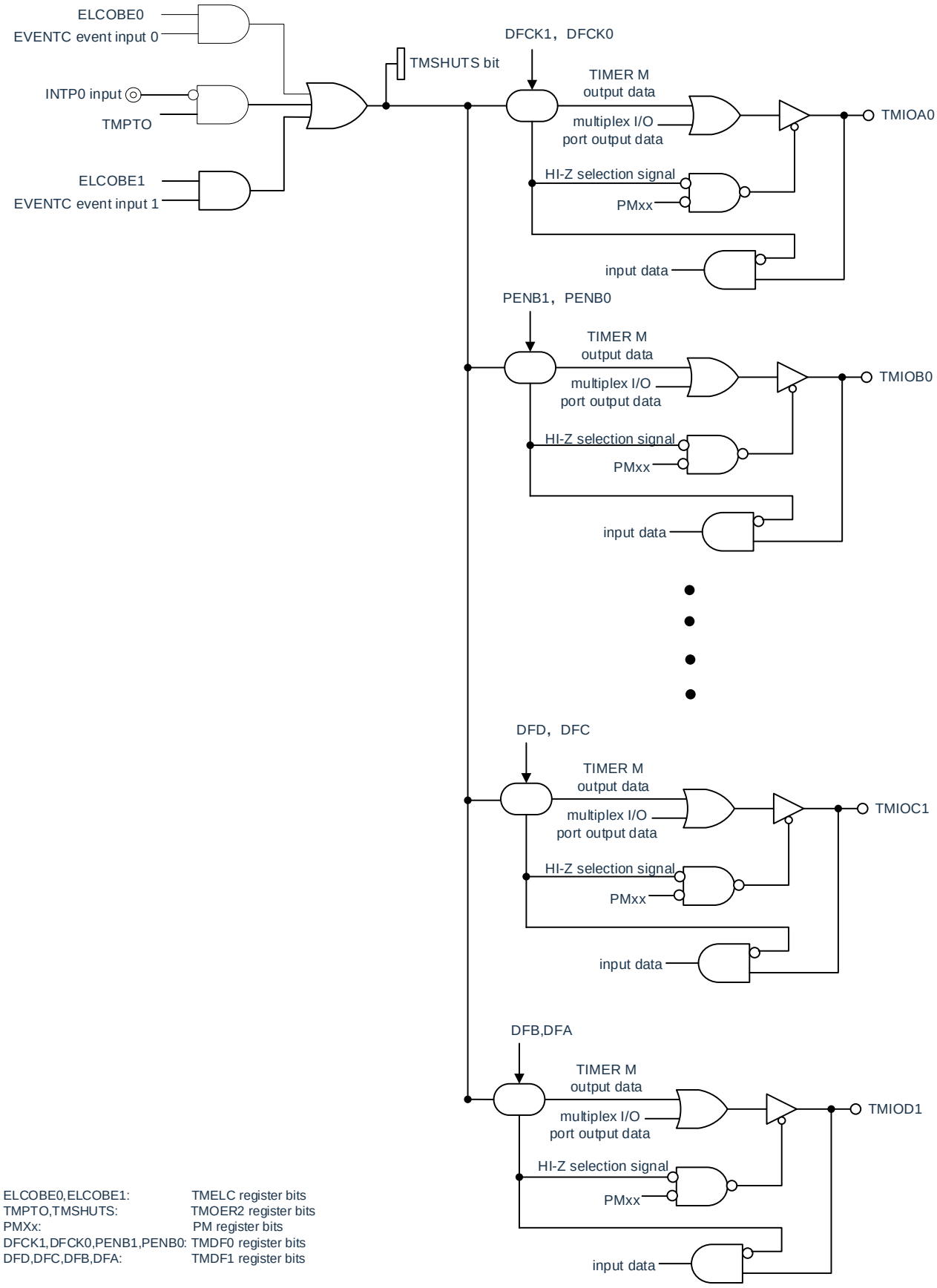
When the PWM function is used, or in reset synchronous PWM mode, complementary PWM mode and PWM3 mode, the TMIOji output pin (i=0,1, B,D) can be cut off.

If the corresponding position of the TMOER1 register is "0" (the output of timer M is allowed), the output pin used in these functions or modes is used as the output pin of timer M. When the TMOER2 register has a TMPTO bit of '1' (the pulse output forced cut-off signal INTP0 pin input is valid), the TMDF0 register or the TMDF1 register DFCK1, DFCK0, PENB1, PENB0,DFD,DFC, DFB, DFA bit settings output from the output pin used as the timer M output port.

When you use this feature, you must have the following settings:

- The pin state (high impedance, 'L' level output, or 'H' level output) when the pulse output is forced off is set through TMDFi registers.
- For pulse-forced cutoff with EVENTC event input, refer to "Events Entered from the Event Coordination Controller(EVENTC)".
- The TMSHUTS bit of the TMOER2 register is "1" when forcing the output of the cutoff pulse. To abort the output of the force cutoff pulse, the TMSHUTS position "0" must be set during stopping the count (TSTARTi=0).
- Put the TMPTO position of the TMOER2 register "1" (the pulse output forced cutoff signal INTP0 pin input is valid).

Figure 10-45 Forced Cutoff of Pulse Output



ELCOBE0,ELCOBE1: TMELC register bits
TMPTO,TMSHUTS: TMOER2 register bits
PMXx: PM register bits
DFCK1,DFCK0,PENB1,PENB0: TMDF0 register bits
DFD,DFC,DFB,DFA: TMDF1 register bits

10.4.5 Events entered from the Event Coordination Controller(EVENTC)

The timer M runs two types of events entered through the EVENTC.

(a) Input capture for TMIOD0/TMIOD1

The timer M captures the input of the TMIOD0/TMIOD1 by the events input by EVENTC. At this point, the IMFD bit of the TMSRi register is "1".

To use this feature, you must select the timer mode input capture feature and either the ELCICE0 bit of the TMELC register or the ELCICE1 position "1. In other modes (output comparison function of timer mode, PWM function, reset synchronous PWM mode, complementary PWM mode, PWM3 mode) this function is invalid.

(b) Running Note for Forced Cutoff Pulse Output

Events entered through the EVENTC force the output of the cutoff pulse. To use this feature, you must select Pulse Output Mode (PWM function, Reset Synchronous PWM mode, Complementary PWM mode, PWM3 mode) and either the ELCOBE0 or the ELCOBE1 position"1. This feature is not valid when using input capture in timer mode.

Note: The INTP0 pin's mandatory cutoff function cuts off the pulse output during the "L" level input, but the EVENTC event's mandatory cutoff function.

Configuration Steps

1) Set the EVENTC event linkage target to timer M.
2) Set ELCICEi bits (i=0, 1) and ELCOBEi bits (i=0, 1) of the TMELC register.

10.4.6 Events output to the event Coordination Controller(EVENTC)/data transfer controller (DMA)

The mode of the timer M and the events output to the EVENTC/DMA are shown in Table 10-11.

Table 10-11. Mode of timer M and event output to ELC/DMA

Usage Mode	output source	EVENTC	DMA
Input capture	TMIOA0 edge detection via IOA1 and IOA0 bit settings of TMIORA0 registers	○	○
	TMIOB0 edge detection via IOB1 and IOB0 bit settings of TMIORA0 registers	○	○
	TMIOC0 edge detection via IOC1 and IOC0 bit settings of TMIORC0 registers	—	○
	TMIOD0 edge detection via IOD1 and IOD0 bit settings of TMIORC0 registers	—	○
	TMIOA1 edge detection via IOA1 and IOA0 bit settings of TMIORA1 registers	○	○
	TMIOB1 edge detection via IOB1 and IOB0 bit settings of TMIORA1 registers	○	○
	TMIOC1 edge detection via IOC1 and IOC0 bit settings of TMIORC1 registers	—	○
	TMIOD1 edge detection via IOD1 and IOD0 bit settings of TMIORC1 registers	—	○
output comparison function, PWM functions, reset synchronous PWM mode, complementary PWM mode, PWM3 mode	Comparison and Matching of TM0 Register and TMGRA0 Register	○	○
	Comparison and Matching of TM0 Register and TMGRB0 Register	○	○
	Comparison and Matching of TM0 Register and TMGRC0 Register	—	○
	Comparison and Matching of TM0 Register and TMGRD0 Register	—	○
	Comparison and Matching of TM1 Register and TMGRA1 Register	○	○
	Comparison and Matching of TM1 Register and TMGRB1 Register	○	○
	Comparison and Matching of TM1 Register and TMGRC1 Register	—	○
	Comparison and Matching of TM1 Register and TMGRD1 Register	—	○
complementary PWM mode	Underflow of TM1 Register	○	—

10.5 Operation of timer M

10.5.1 Input capture function

This is a function of measuring the width and period of an external signal. The TMi register (counter) is transferred to the TMGRji register (input capture) by using the external signal of the TMIOji pin (i=0,1, j=A,B,C,D) as the trigger. Because the TMIOji pin and the TMGRji register are used in combination, the pin can be selected as an input capture function, or other mode and function.

The block diagram for the input capture function and the running example are shown in Figures 10-46 and 10-47 respectively, and the specification for the input capture function is shown in Table 10-12.

Figure 10-46 Block Diagram of Input Capture Function

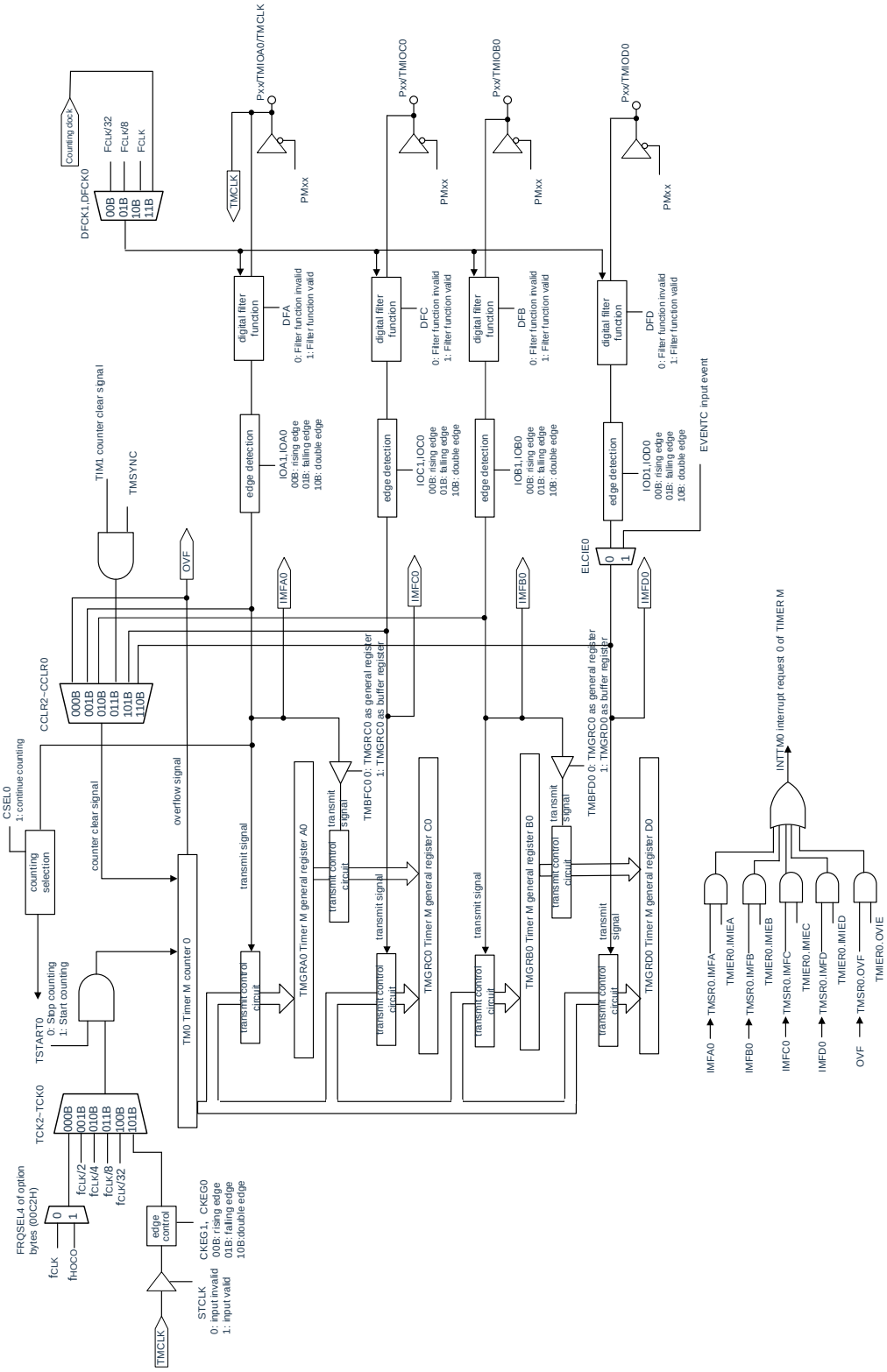


Table 10-12 Enter the specifications for the capture function

Project	Specifications
count source	fHOCO [note], fCLK, fCLK/2, fCLK/4, fCLK/8,fCLK/32 External input signal for the TMCLK pin (program selection of effective edges)
Count	incremental count
counting period	When the CCLR2~CCLR0 bit of the TMCRi register is '000B' 1/fk 65536 fk: Frequency of the count source
Count Start Condition	Write "1" to the TSTARTi bit of the TMSTR register.
Count stop condition	When the CSELi bit of the TMSTR register is "1", write "0" to the TSTARTi bit.
Generation sequence of interrupt requests	· Input capture (valid edges for TMIOji inputs) • TMi Overspill
TMIOA0 Pin Functionality	I/O Port, Input Captured Input, or TMCLK (External Clock) Input
TMIOB0, TMIOC0, TMIOD0, TMIOA1~ TMIOD1 Pin Functionality	I/O port or input captured input (select by pin)
INTP0 Pin Functionality	Not used (enter a private port or INTP0 interrupt input).
read timer	If you read the TMi register, you can read the count value.
write timer	· TMi registers can be written when the TMSYNC bit of the TMMR register is "0". · When the TMMR register has a TMSYNC bit of "1" (timer M0 and timer M1 running synchronously), data is written to both TM register.
Select Features	· Enter the selection of input pins for the capture 1 or more pins in the TMIOAi, TMIOBi, TMIOCi, TMIODi pins · The selected rising, falling, or double edges of the input effective edges captured by the input · Overflows or captures the timing of the TMi setting to "0000H" · Buffer Run (reference to "10.4.2 Buffer Run") · Run synchronously (see "10.4.3 Run Synchronously") · Digital filter The TMIOji input is sampled and if that signal is the same three time, the level is

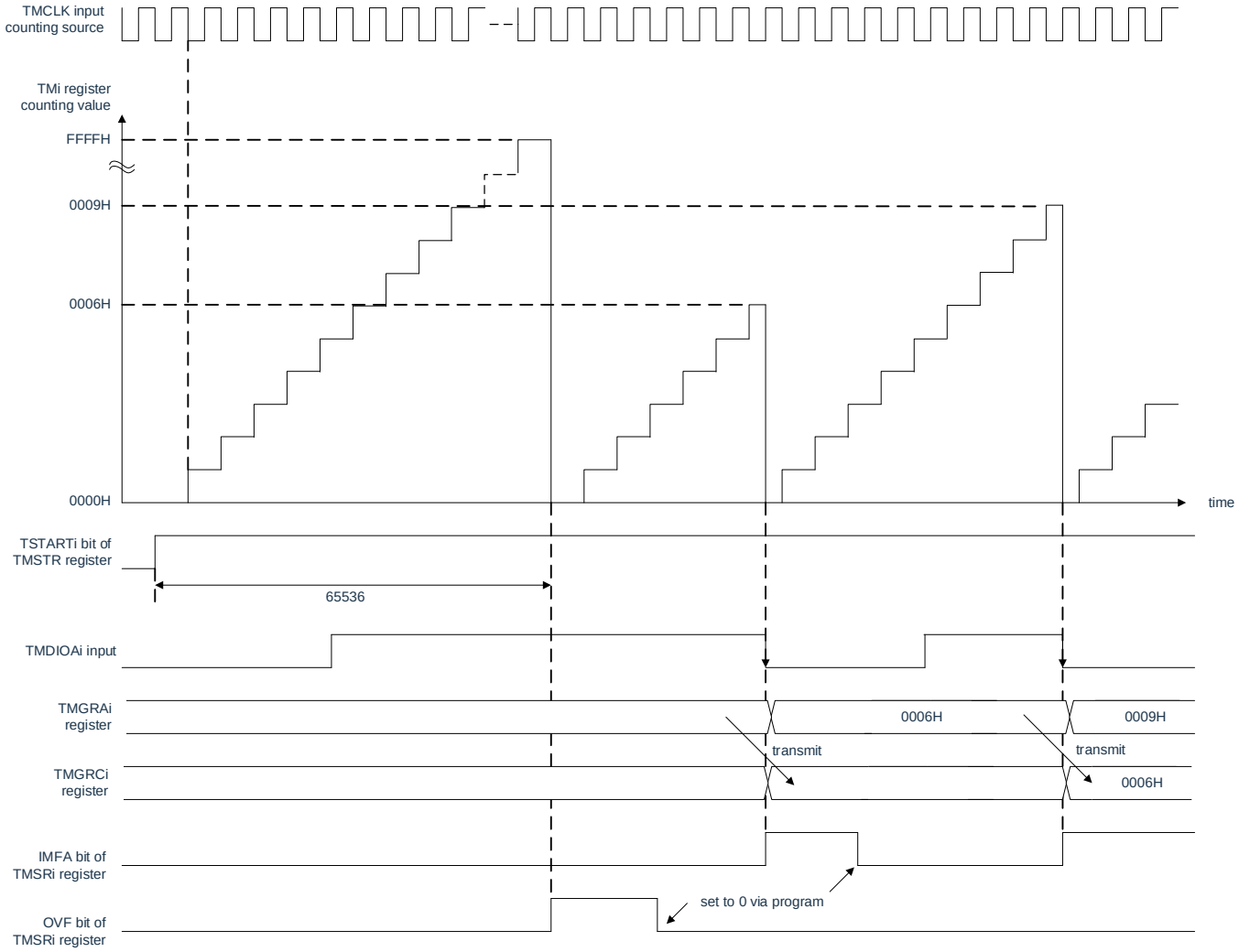
Note: fHOCO can only be selected if the FRQSEL4 bit of the user option byte ($_{000C2H)\ is\ "1"}$. To select fHOCO as the counter source for timer M, fCLK must be set to fIH before bit4 ($TMMEN_{)\ of\ Peripheral}$ Admission Register 1 ($PER1_{)}$. If you want to change fCLK to a clock other than fIH, you must change it after clearing the bit4 (TMMEN) of peripheral enable register 1 (PER1).

Note: i=0,1,j=A,B,C,D

(1) running example

The counter value of the timer Mi is reset when input capture or comparison match occurs by setting CCLR0~CCLR2 bits of the TMCRi register (i=0,1). FIGS. 10-47 is an example of operation when the position CCLR2~CCLR0 is "001B". If it is set to clear the count by the input capture during running and capture the input when the count value of the timer is 'FFFFH', the interrupt flag of IMFA~IMFD and OVF bits of the TMSRi register.

Figure 10-47 Running Example of Input Capture



Note i=0, 1

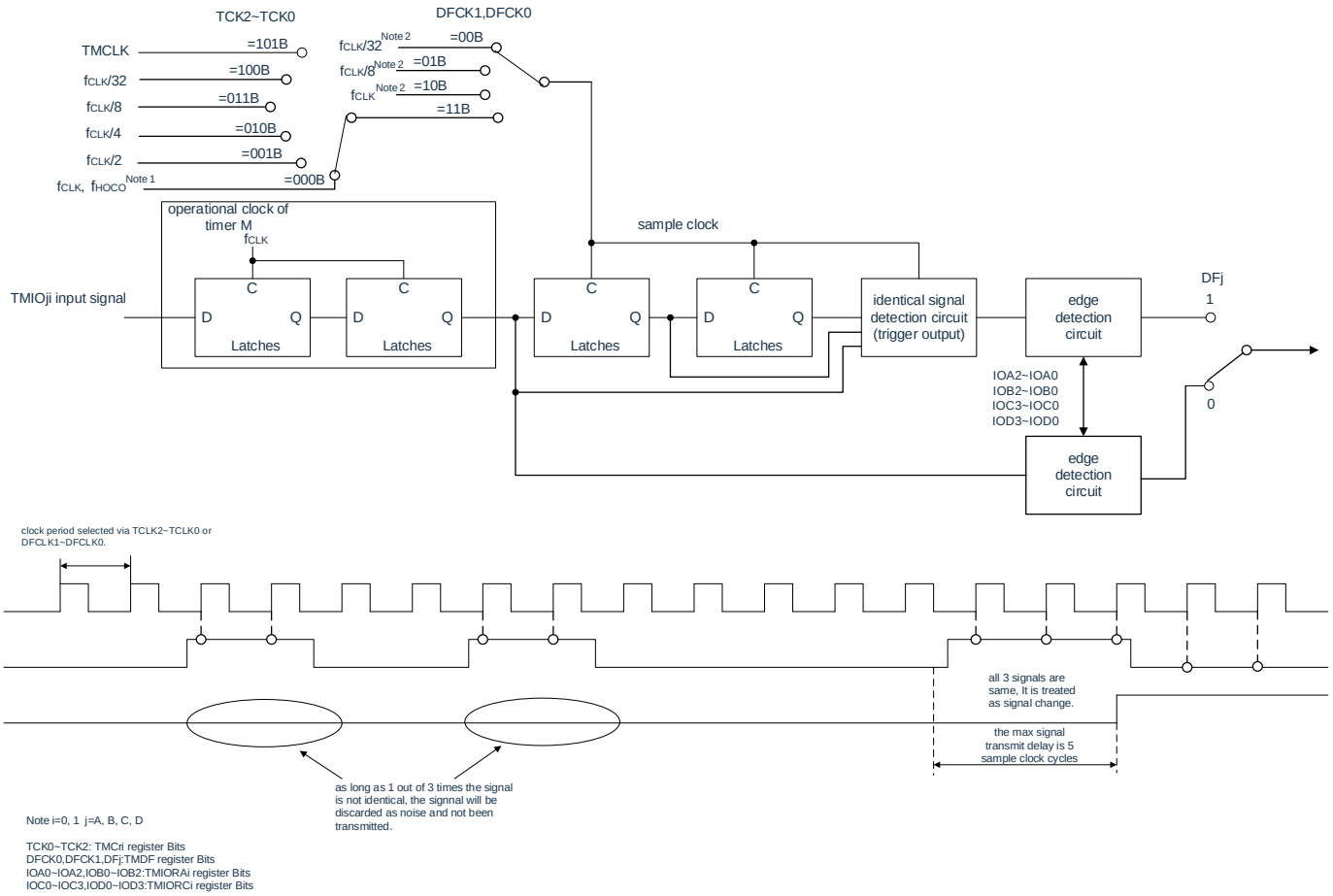
above diagram condition as following:
CCLR2~CCLR0 bit of TMCRi register as "001B" (set Tmi bit as "0000H" while TMGRAi input capture)
TCLK2~TCLK0 bit of TMCRi register as "101B" (Counting source as TMCLK input)
CKEG1 bit and CKEG0 bit of TMCRi register as "01B" (counting at falling edge of counting source).
IOA2~IOA0 bit of TMIORAi register as "101B" (input capture at falling edge of TMIOAi input)
TMBFCi bit of TMMR register as "1" (TMGRCi register is the buffer register of TMGRAi register)

(2) digital filter

The TMIOji input (i=0, 1,j=A,B,C,D) is sampled and if that signal is identical 3 times, the level is determined. The function of the digital filter and the sampling clock must be selected through the TMDFi register.

The block diagram of the digital filter is shown in Figure 10-48.

Figure 10-48 Block diagram of digital filter



Note: 1. When the user option byte (000C2H) has the FRQSEL4 bit of "0", select fCLK; Select fHOCO when the FRQSEL4 bit is "1"

2. fCLK/32, fCLK/8 and fCLK are fHOCO/32, fHOCO/8 and fHOCO when the FRQSEL4 bit of user option byte (000C2H) is "1".

10.5.2 output comparison function

This is a mode to detect whether the contents of TMi registers (counters) (i=0,1) and TMGRji registers (j=A,D) are identical (compare matching). If the contents are the same, output any level from the TMIOji pin. Because the TMIOji pin and the TMGRji register are used in combination, the pin can be selected as the output comparison function, or other modes and functions. The block diagram and running example of the output comparison function are shown in Figure 10-49 and Figure 10-50 respectively, and the specification of the output comparison function is shown in Table 10-13.

Figure 10-49 Block Diagram of Output Comparison Function

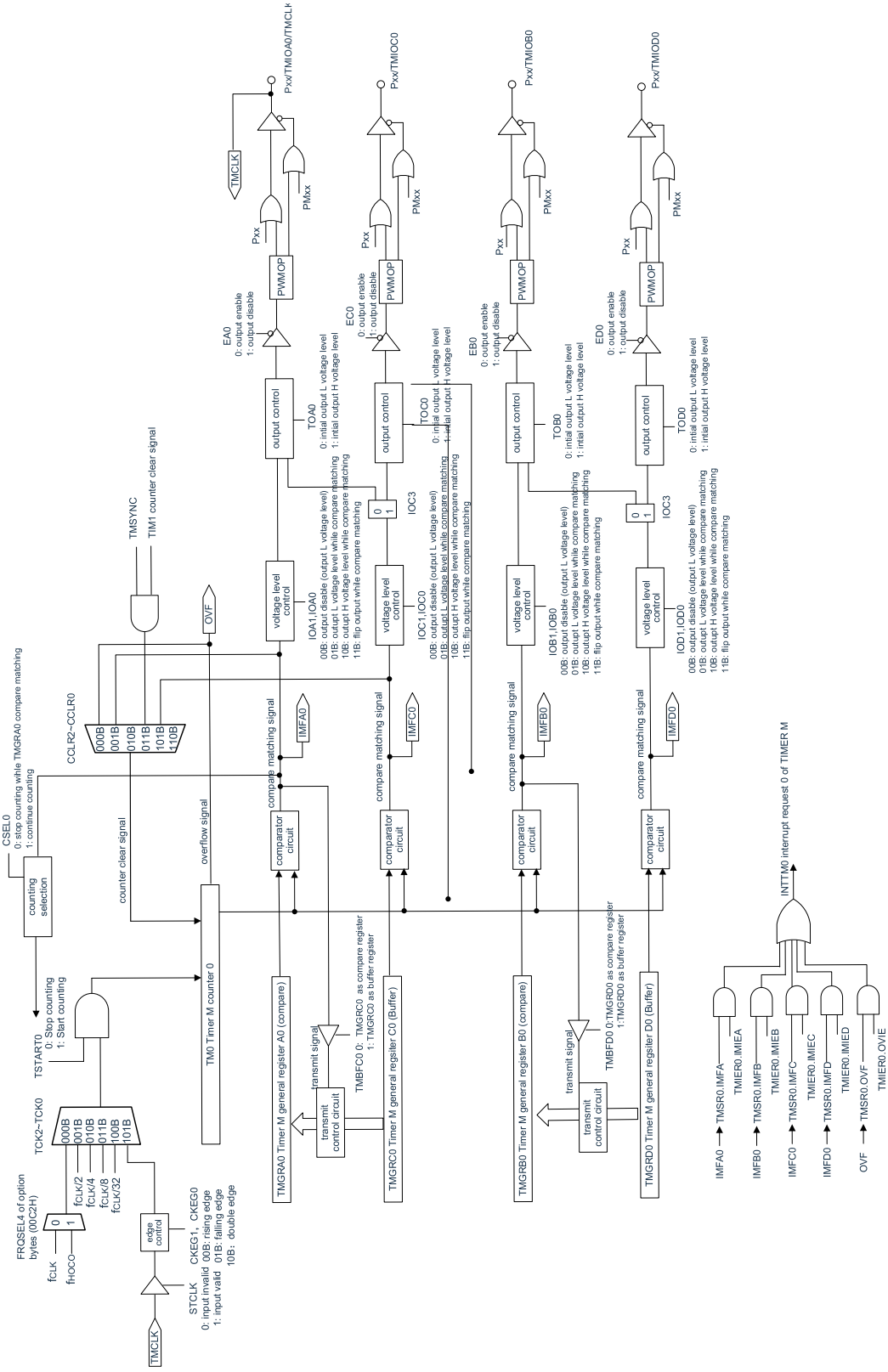


Table 10-13. Output Comparison Feature Specifications

Project	Specifications
count source	fHOCO note, fCLK, fCLK/2, fCLK/4, fCLK/8,fCLK/32 External input signal for the TMCLK pin (program selection of effective edges)
Count	incremental count
counting period	· When the CCLR2~CCLR0 bit of the TMCRi register is '000B' 1/fk 65536 fk: Frequency of the count source · When the CCLR1~CCLR0 bit of the TMCRi register is "01B" or "10B" (TMi is set "0000H" when TMGRji matches) 1/fk (n+1)n:TMGRji register's setting value
waveform output timing	Comparison matching (same contents of TMi register and TMGRji register)
Count Start Condition	Write "1" to the TSTARTi bit of the TMSTR register.
Count stop condition	· Write "0" to the TSTARTi bit when the CSELi bit of the TMSTR register is "1". The output pin of the output comparison maintains the output level before stopping counting. · Stop counting when the CSELi bit of the TMSTR register is "0" and a comparison match for TMGRAi occurs. The output pin of the output comparison maintains the level
Generation sequence of interrupt requests	· Comparison matching (same contents of TMi register and TMGRji register) • TMi Overspill
TMIOA0 Pin Functionality	I/O port, output for output comparison, or TMCLK (external clock) input
TMIOB0, TMIOC0, TMIOD0, TMIOA1~ TMIOD1 Pin Functionality	I/O port or output compared output (select by pin)
INTP0 Pin Functionality	Not used (enter a private port or INTP0 interrupt input).
read timer	If you read the TMi register, you can read the count value.
write timer	· TMi registers can be written when the TMSYNC bit of the TMMR register is "0". · When the TMMR register has a TMSYNC bit of "1" (timer M0 and timer M1 running synchronously), data is written to both TM register.
Select Features	· Selection of output pins for output comparisons 1 or more pins in the TMIOAi, TMIOBi, TMIOCi, TMIODi pins · Selection of output levels for comparison matching 'L' level output,'H' level output, or level inversion output · The selection setting of the initial output level is the level from the start of the count until the comparison match. · Set TMi to the time sequence of 0000H Comparison matching of overflow or TMGRAi registers · Buffer Run (see "10.4.2 Buffer Run") · Run synchronously (refer to "10.4.3 Run Synchronously") • Changes to Output Pins for TMGRCi and TMGRDi TMGRCi and TMGRDi can be used for output control of TMIOAi and TMIOBi pins, respectively. · The timer M can be used as an internal timer without output.

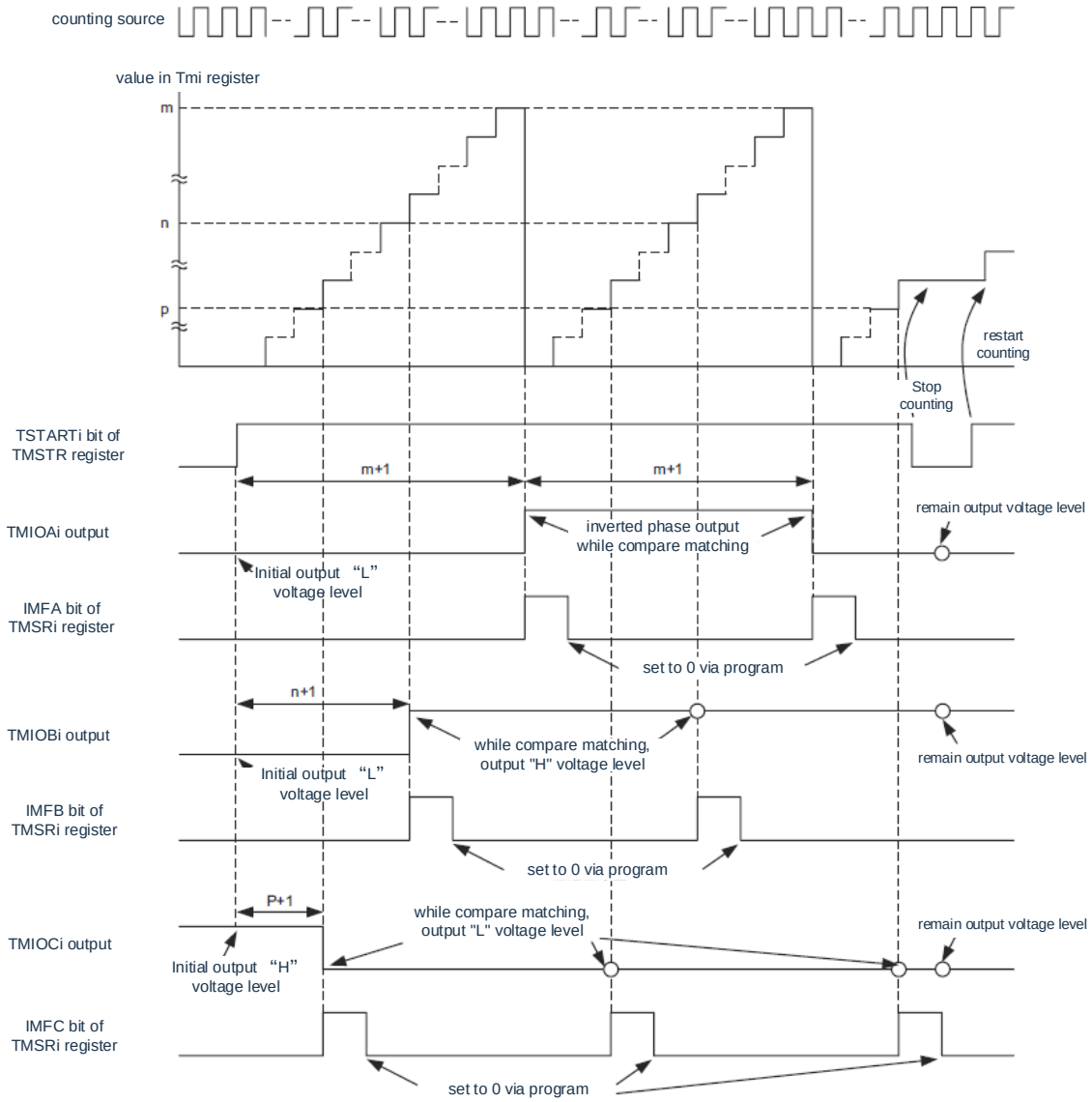
Note: fHOCO can only be selected if the FRQSEL4 bit of the user option byte (000C2H) is "1". To select fHOCO as the counter source for timer M, fCLK must be set to fIH before bit4 (TMMEN) of Peripheral Admission Register 1 (PER1). If you want to change fCLK to a clock other than fIH, you must change it after clearing the bit4 (TMMEN) of peripheral enable register 1 (PER1).

Notes: i=0, 1, j=A, B,C,D

(1) running example

The counter value of the timer Mi is reset when input capture or comparison match occurs by setting CCLR0~CCLR2 bits of the TMCRi register (i=0,1). If that expect value is" FFFFH", it will change from" FFFFH" to" 0000H" and the overflow flag to" 1".

Figure 10-50 Running Example of the Output Comparison Function



Notes: i=0,1
Setting value for the m:TMGRAi register
Setting value for the n:TMGRBi register
Setting value for the p:TMGRCi register

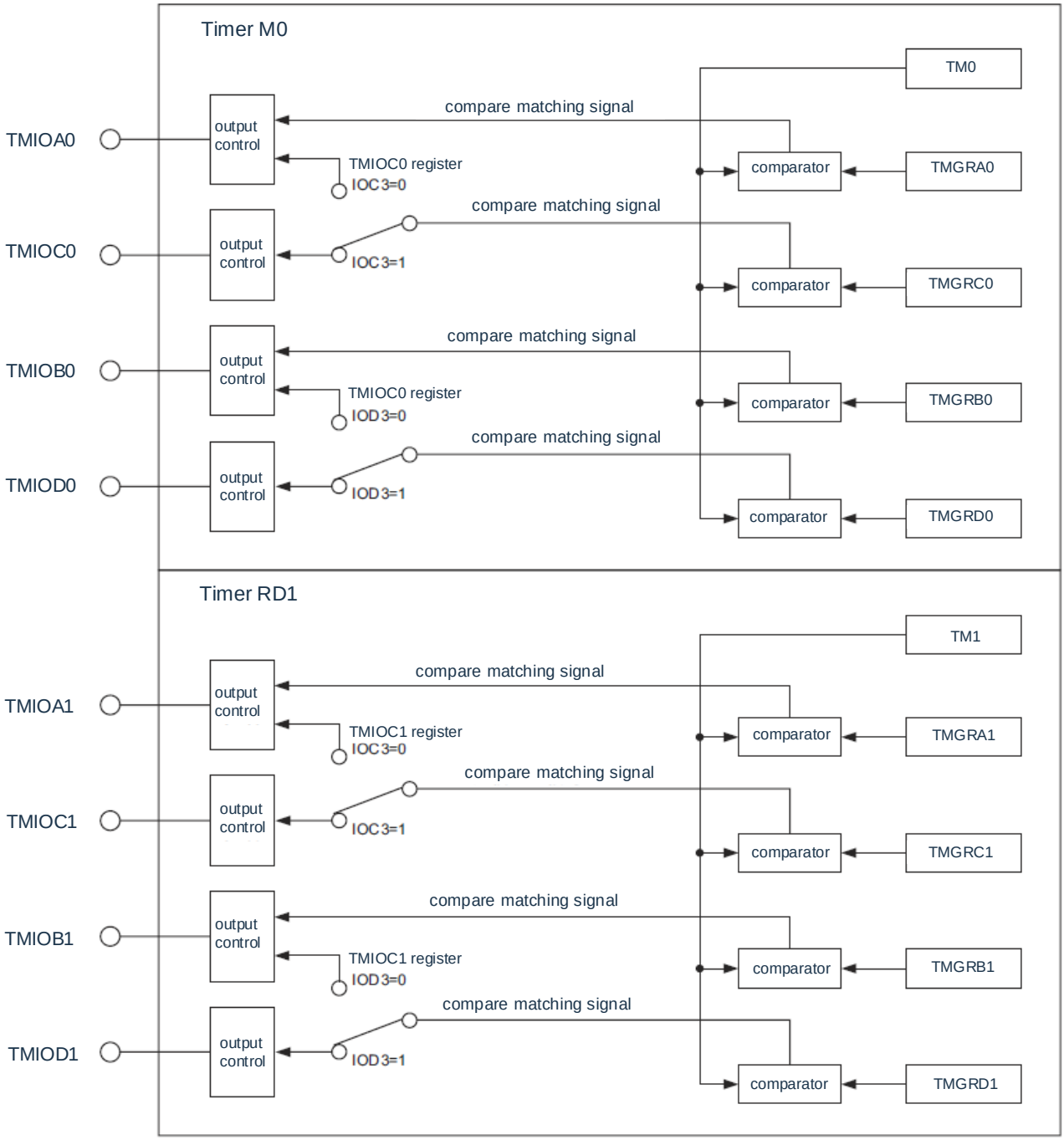
The conditions in the above diagram are as follows:
The CSELi bit of the TMSTR register is "1" (TMi does not stop when comparing matches).
The TMBFCi and TMBFDi bits of the TMMR register are "0" (TMGRCi and TMGRDi do not run as buffers).
EAi bit, EBi bit, and ECi bit of the TMOER1 register are "0" (output of TMIOAi, TMIOBi, and TMIOCi allowed).
The CCLR2~CCLR0 bit of the TMCRi register is "001B".
TMOCR register TOAi bit and TOBi bit are '0' (initial output 'L' level before comparison match) and TOCi bit is '1' (initial output 'H' level before comparison match).
The IOA2~IOA0 bit of the TMIORAi register is "011B" (TMIOAi is inverted when TMGRAi matches).
The IOB2~IOB0 of the TMIORAi register is "010B" (TMIOBi outputs "H" levels when TMGRBi comparisons match).
The IOC3~IOC0 of the TMIORCi register is "1001B" (TMIOCi outputs "L" levels when TMGRCi comparisons match).
The IOD3~IOD0 of the TMIORCi register is "1000B".

(2) Changes to the output pin of the TMGRCi register and the TMGRDi register (i=0,1)

The TMGRCi register and the TMGRDi register can be used for the output control of the TMIOAi pin and the TMIOBi pin, respectively. Therefore, each pin can be output control as follows:

- The TMIOAi output is controlled by the value of the TMGRAi register and the value of the TMGRCi register.
- The TMIOBi output is controlled by the value of the TMGRBi register and the value of the TMGRDi register.

Figure 10-51 Changes to the output pin for TMGRCi and TMGRDi



To change the output pin of the TMGRCi register and the TMGRDi register, you must have the following settings:

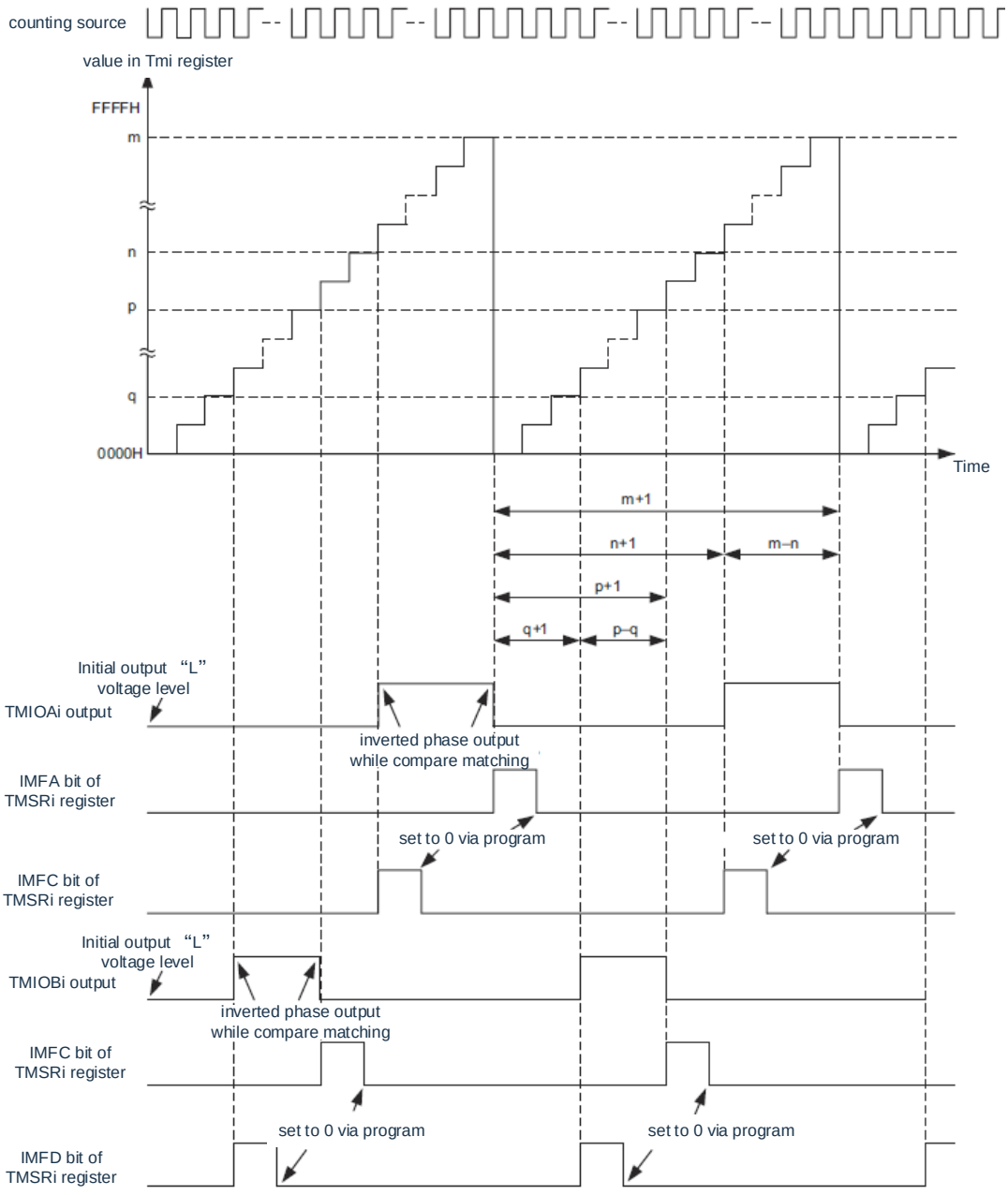
· Select "0" (change of output pin of the TMGRji register) by the TMIORCi register's IOj3 bits (j=C, D).

· Place the TMBFji location of the TMMR register "0" (universal register).

- Different values are set for the TMGRAi register and the TMGRCi register, and different values are set for the TMGRBi register and the TMGRDi register.

Examples of the operation of using TMGRCi and TMGRDi for output control of the TMIOAi and TMIOBi pins, respectively, are shown in Figure 10-52.

Figure 10-52 Example of running TMGRCi and TMGRDi for output control of TMIOAi and TMIOBi pins, respectively



Notes: i=0,1
Setting value for the m:TMGRAi register
Setting value for the n:TMGRBi register
Setting value for the p:TMGRCi register
q:Setting value for the TMGRDi register

The conditions in the above diagram are as follows:
The CSELi bit of the TMSTR register is "1" (TMi does not stop when comparing matches).
The TMBFCi and TMBFDi bits of the TMMR register are "0" (TMGRCi and TMGRDi do not run as buffers).
EAi bits, EBi bits, and ECi bits of the TMOER1 register are "0" (allow output of TMIOAi, TMIOBi, and TMIOCi).
The CCLR2~CCLR0 bit of the TMCRi register is "001B".
The TMOCR register has a TOAi bit and a TOBi bit of "0" (the initial output "L" level before the comparison match).
The IOA2~IOA0 bit of the TMIORAi register is "011B" (TMIOAi is inverted when TMGRAi matches).
The IOB2~IOB0 of the TMIORAi register is "011B" (TMIOBi is inverted when TMGRBi matches).
The IOC3~IOC0 of the TMIORCi register is "0011B" (TMIOAi is inverted when TMGRCi matches).
The IOD3~IOD0 of the TMIORDi register is "1000B" (TMIOBi is inverted when TMGRCi comparisons match).

10.5.3 PWM function

This is the function of the output PWM waveform. At most 3 PWM waveforms can be output through the timer Mi (i=0,1). By synchronizing the timer M0 with the timer M1, up to 6 PWM waveforms can be output in the same cycle.

Because TMIOji pins (j=B, C, D) and TMGRji registers are used in combination, the pin can be selected to be PWM function, or other modue and function (but, no matter which pin is used as PWM function, TMGRAi register must be used, therefore the TMGRAi register can not be used for other modes).

The block diagram and specification of PWM function are shown in Figure 10-53 and Table 10-14, respectively, and the operation example is shown.

Figure 10-53 Block diagram of PWM function

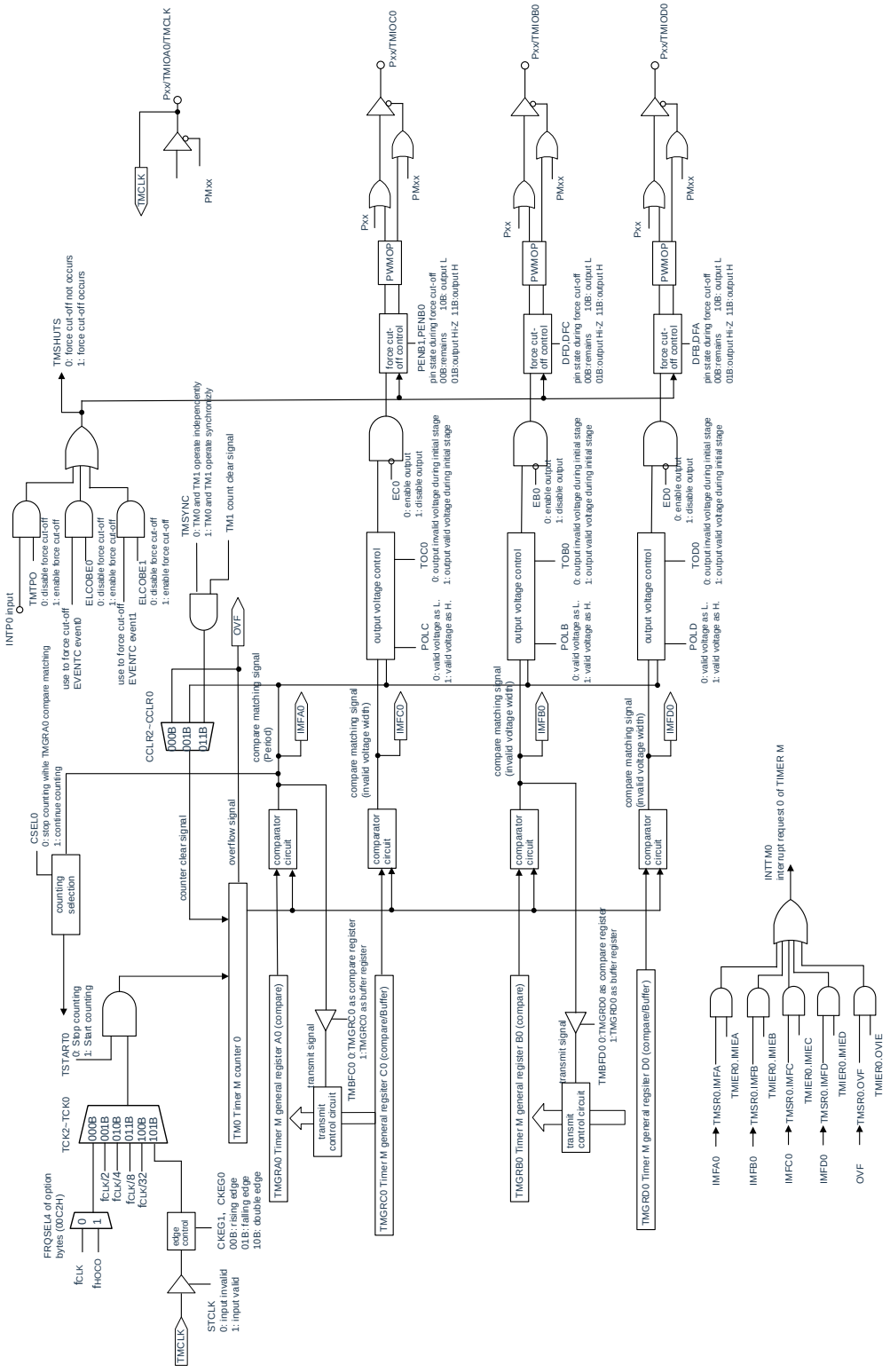
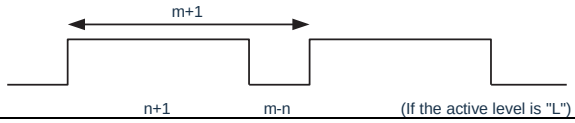


Table 10-14. Specifications of PWM Functions

Project	Specifications
count source	fHOCO note, fCLK, fCLK/2, fCLK/4, fCLK/8, fCLK/32 External input signal for the TMCLK pin (allows program selection of effective edges)
Count	incremental count
PWM waveform	PWM Cycle: 1/fk (m+1) Effective Level Width: 1/fk

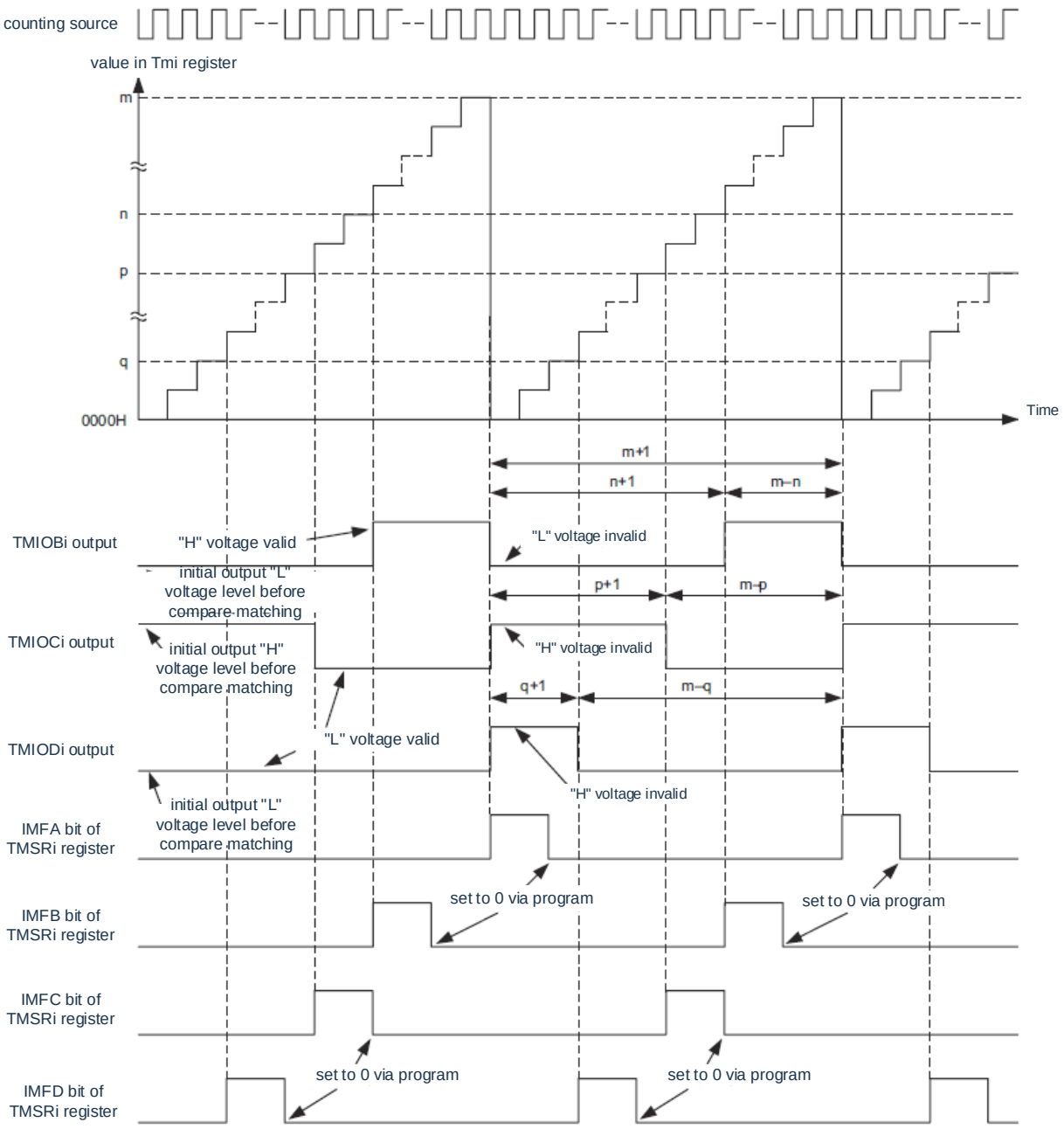
	(m-n) Invalid Level Width: 1/fk (n+1) fk: Frequency of the count source m:Setting value for TMGRAi register n:Setting value for the TMGRji register 
Count Start Condition	Write "1" to the TSTARTi bit of the TMSTR register.
Count stop condition	· Write "0" to the TSTARTi bit when the CSELi bit of the TMSTR register is "1". The PWM output pin maintains the output level before stopping the count. · Stop counting when the CSELi bit of the TMSTR register is "0" and a comparison match for TMGRAi occurs. The PWM output pin maintains the level after the comparison match causes the output to change.
Generation sequence of interrupt requests	· Comparison matching (same contents of TMi register and TMGRhi register) • TMi Overspill
TMIOA0 Pin Functionality	I/O port or TMCLK (external clock) input
TMIOA1 Pin Functionality	I/O port
TMIOB0, TMIOC0, TMIOD0, TMIOB1,TMIOC1,TMIOD1 pin function	I/O port or PWM output (select by pin)
INTP0 Pin Functionality	Input of the pulse output forcing cut-off signal (input private port or INTP0 interrupt input)
read timer	If you read the TMi register, you can read the count value.
write timer	Can write TMi register.
Select Features	· Selection of 1~3 PWM output pins by timer Mi 1 or more pins in the TMIOBi, TMIOCi, TMIODi pin · Selection of effective levels for each pin · Selection of initial output levels for each pin · Run synchronously (see "10.4.3 Run Synchronously") · Buffer Run (reference to "10.4.2 Buffer Run") · The input of the pulse output compulsory cutoff signal (reference to "10.4.4 compulsory cutoff of pulse output")

Note: fHOCO can only be selected if the FRQSEL4 bit of the user option byte (000C2H) is "1". To select fHOCO as the counter source for timer M, fCLK must be set to fIH before bit4 (TMMEN) of Peripheral Admission Register 1 (PER1). If you want to change fCLK to a clock other than fIH, you must change it after clearing the bit4 (TMMEN) of peripheral enable register 1 (PER1).

Note: i=0,1, j=B,C,D, h=A, B,C,D

(1) running example

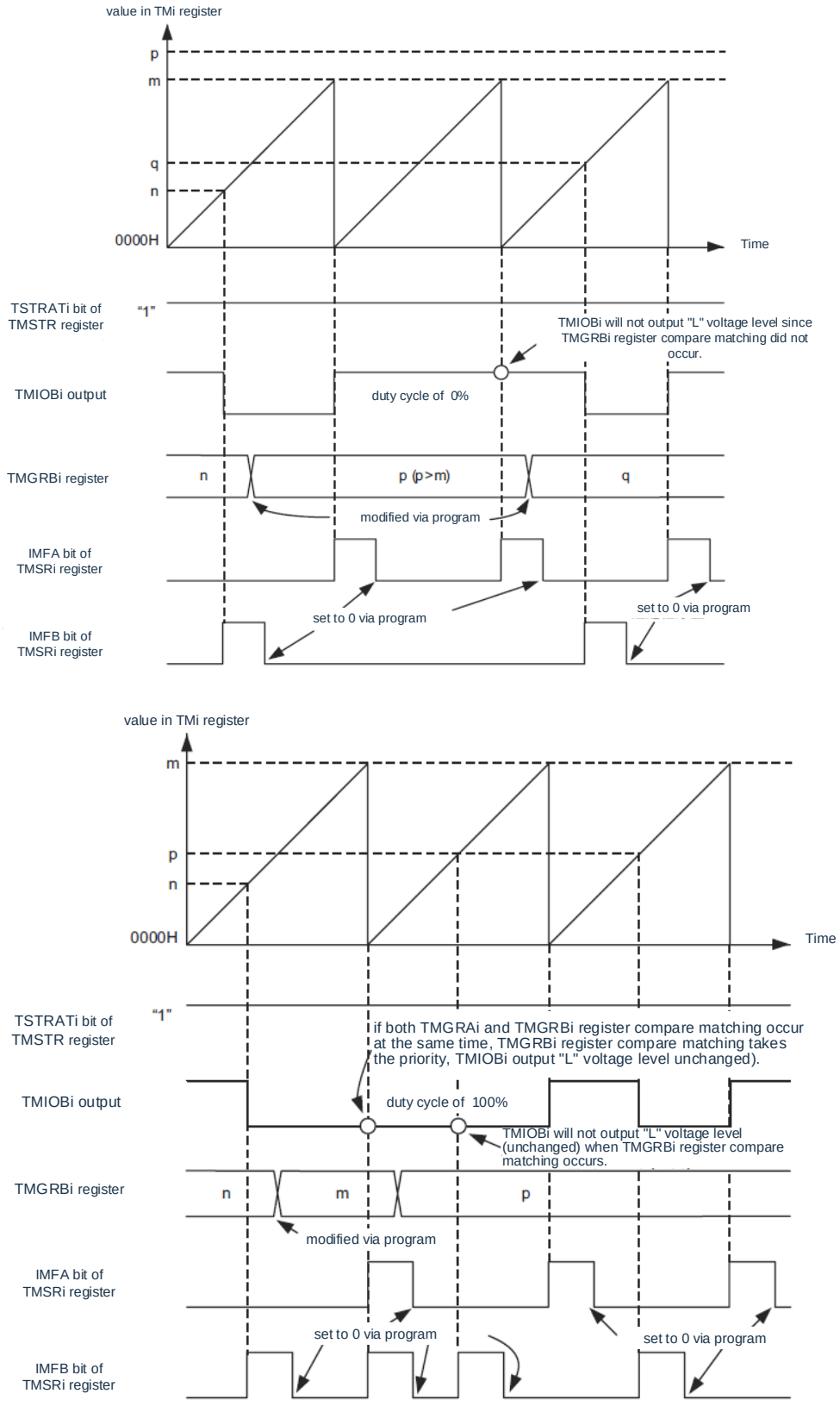
Figure 10-54 Running example of PWM function



Notes: i=0,1
Setting value for the m:TMGRAi register
Setting value for the n:TMGRBi register
Setting value for the p:TMGRCi register
q:Setting value for the TMGRDi register

The conditions in the above diagram are as follows:
The TMBFCi and TMBFDi bits of the TMMR register are "0" (TMGRCi and TMGRDi do not run as buffers).
The EBi and ECi bits of the TMOER1 register are "0" (allow output of TMIOBi and TMIOCi).
The TMOCR register has a TOBi bit and a TOCi bit of '0' (invalid level) and a TODi bit of '1' (valid level).
The TMPOCRi register has a POLB bit of "1" (valid at the "H" level) and a POLC bit and a POLD bit of "0" (valid at the "L" level)

Figure 10-55 Example of PWM function running (duty cycles of 0% and 100%)



Notes: i=0,1
Setting value for the m:TMGRAi register

The conditions in the above diagram are as follows:
The EBi bit of the TMOER1 register is "0" (TMIOBi output allowed).
The POLB bit of the TMPOCRi register is "0" ("L" level valid)

10.5.4 Reset synchronous PWM mode

Output 3 positive and 3 inverse (total 6) PWM waveforms (three-phase, sawtooth modulation, dead time).

The block diagram and the running example of the reset synchronous PWM mode are shown in Figure 10-56 and Figure 10-57 respectively, and the specification of the reset synchronous PWM mode is shown in Table 10-15.

For examples of PWM operation with duty cycles of 0% and 100%, refer to the Figure 10-55 for PWM operation examples.

Figure 10-56 Block diagram of reset synchronous PWM mode

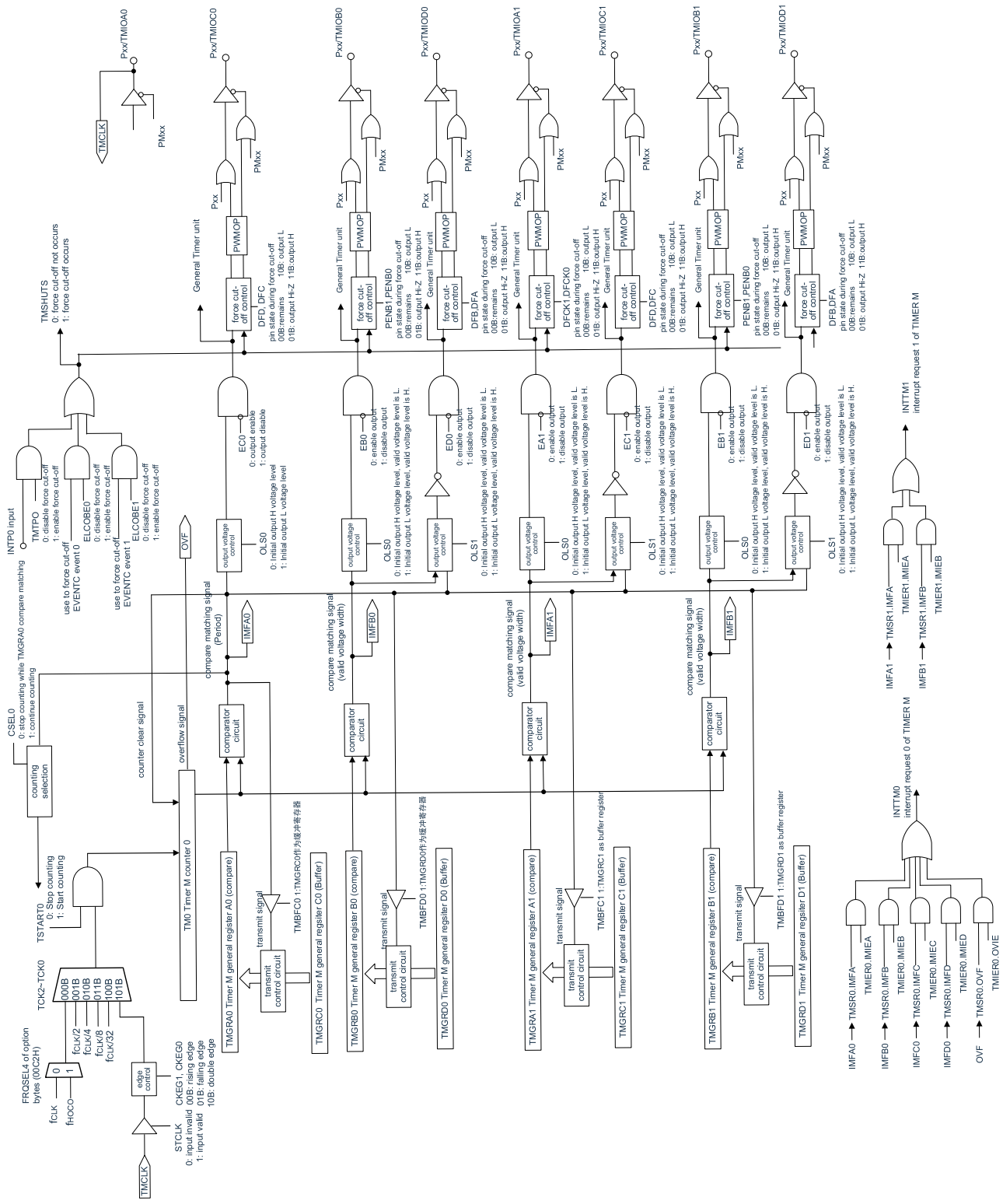
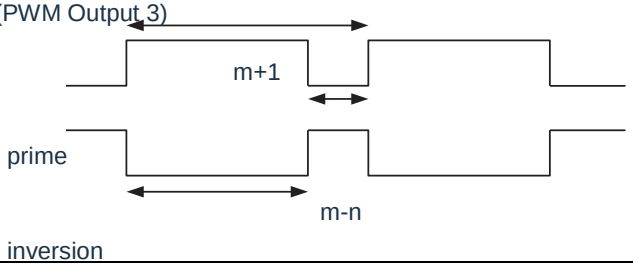


Table 10-15. Specifications for Reset Synchronous PWM Mode

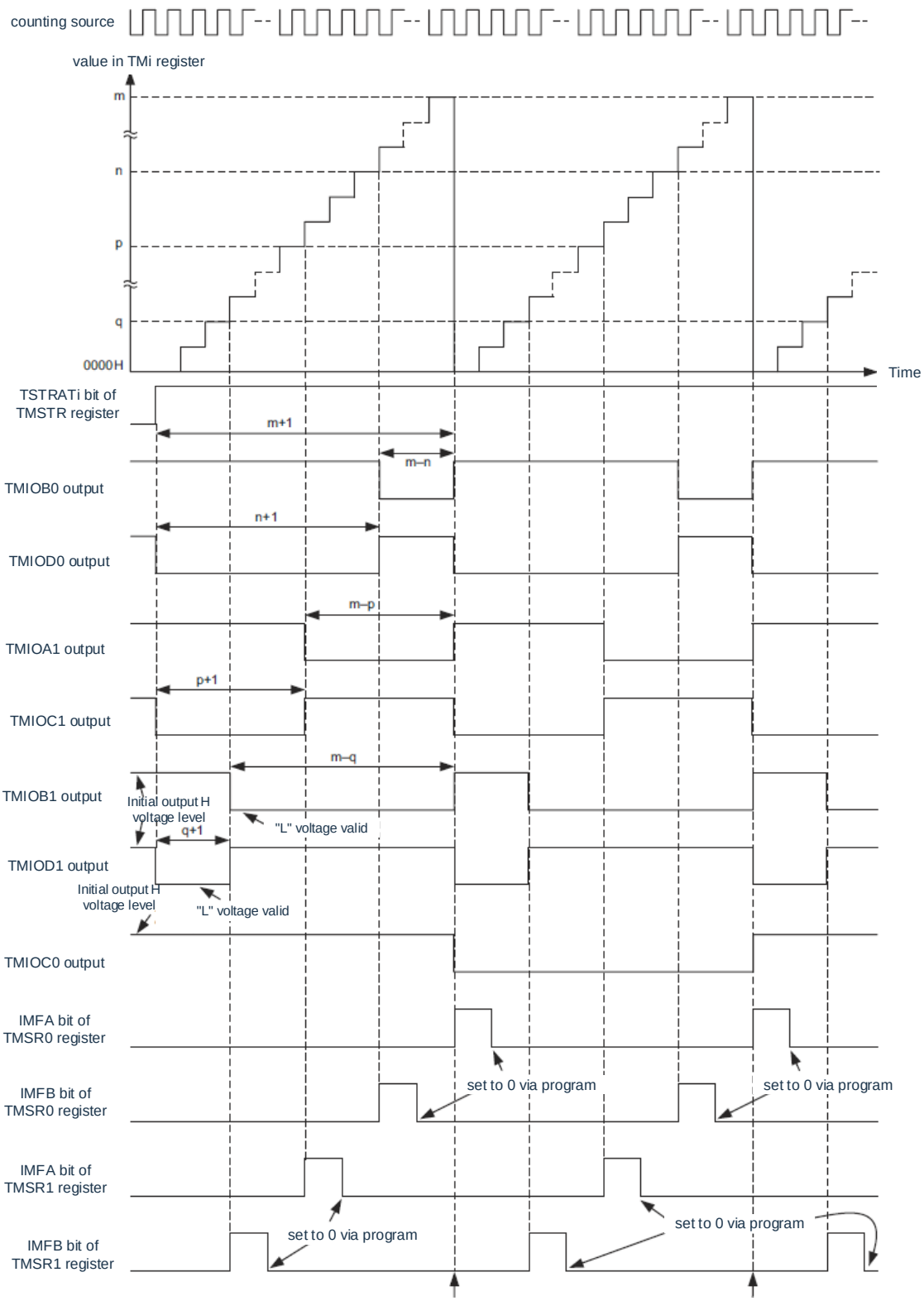
Project	Specifications
count source	fHOCO Note, fCLK, fCLK/2, fCLK/4, fCLK/8, fCLK/32TMCLK pin external input signal (can be programmed to select effective edges)
Count	TM0 is an incremental count (TM1 is not used).
PWM waveform	PWM Period: 1/fk (m+1) Forward Effective Level Width: 1/fk (m-n) Inverted Effective Level Width:1/fk (n+1) fk: Count the frequency of the source m:The setting value of the TMGRA0 register n: Setting Value of TMGRB0 Register (PWM Output 1) Setting Value of TMGRA1 Register (PWM Output 2) Setting Value of TMGRB1 Register (PWM Output 3) 
Count Start Condition	Write "1" to the TSTART0 bit of the TMSTR register.
Count stop condition	· Write "0" to the TSTART0 bit when the CSEL0 bit of the TMSTR register is "1". The PWM output pin outputs an initial output level selected by the OLS0 bit and the OLS1 bit of the TMFCR register. · Stop counting when the CSEL0 bit of the TMSTR register is "0" and a comparison match of the TMGRA0 occurs.
Generation sequence of interrupt requests	· Comparison matching (TM0 register and TMGRj0, TMGRA1, TMGRB1 register are identical) ·TM0 overspill
TMIOA0 Pin Functionality	I/O port or TMCLK (external clock) input
TMIOB0 Pin Functionality	PWM Output 1 Positive Phase Output
TMIOD0 Pin Functionality	PWM Output 1 Invert Output
TMIOA1 Pin Functionality	PWM Output 2 Forward Phase Output
TMIOC1 Pin Functionality	PWM Output 2 Inverting Output
TMIOB1 Pin Functionality	PWM Output 3 Forward Phase Output
TMIOD1 Pin Functionality	PWM Output 3 Invert Output
TMIOC0 Pin Functionality	Inverting output is performed at each PWM cycle.
INTP0 Pin Functionality	Input of the pulse output forcing cut-off signal (input private port or INTP0 interrupt input)
read timer	If you read the TM0 register, you can read the count value.
write timer	Can write TM0 register.
Select Features	· Selection of active and initial output levels for forward and reverse phases · Buffer Run (reference to "10.4.2 Buffer Run") · The input of the pulse output compulsory cutoff signal (reference to "10.4.4 compulsory cutoff of pulse output")

Note: fHOCO can only be selected if the FRQSEL4 bit of the user option byte (000C2H) is "1". To select fHOCO as the counter source for timer M, fCLK must be set to fIH before bit4 (TMMEN) of Peripheral Admission Register 1 (PER1). If you want to change fCLK to a clock other than fIH, you must change it after clearing the bit4 (TMMEN) of peripheral enable register 1 (PER1).

Note: j=A, B, C, D

(1) running example

Figure 10-57 Operating example of reset synchronous PWM mode



Notes: i=0,1
Setting value for the m:TMGRA0 register
Setting value for the n:TMGRB0 register
Setting value for the p:TMGRA1 register
q:Setting value for the TMGRB1 register

The conditions in the above diagram are as follows:
The OLS1 and OLS0 bits of the TMFCR register are "0" (initial output "H" level, "L" level is valid)

10.5.5 complementary PWM mode

Output 3 positive and 3 inverse (total 6) PWM waveforms (three-phase, triangular modulation, dead time).

The complementary PWM mode specification is shown in Table 10-16, and the output model and operation example are shown in Figure 10-60.

Figure 10-58 Block diagram of complementary PWM mode

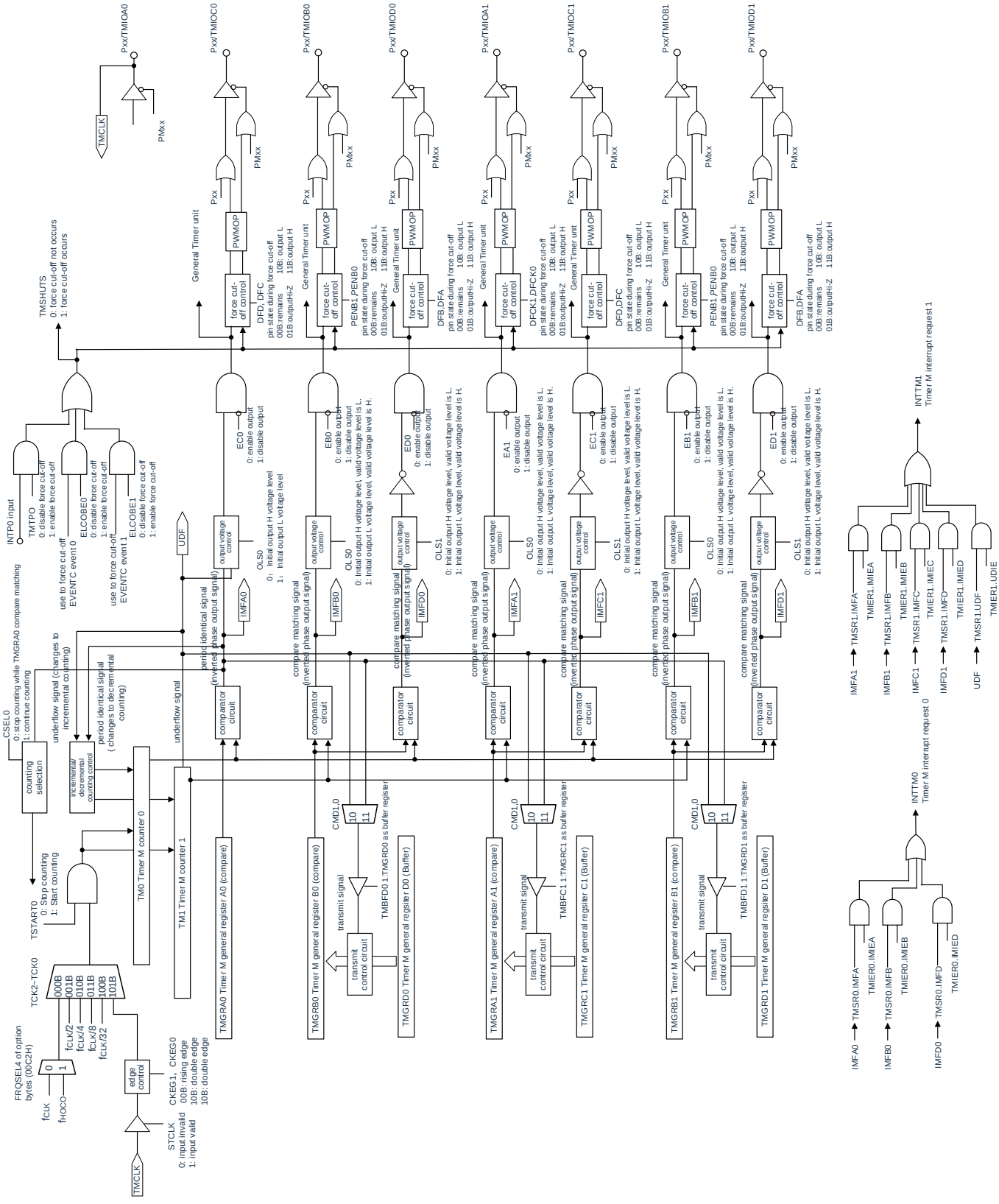
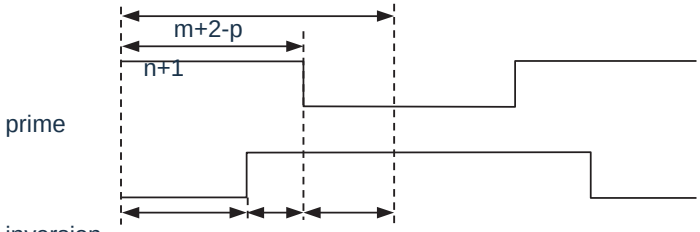


Table 10-16 Specifications for Complementary PWM Modes

Project	Specifications
count source	fHOCO Note 1, fCLK, fCLK/2, fCLK/4, fCLK/8, fCLK/32 External input signal for the TMCLK pin (allows program selection of effective edges) The TMCR0 register's TCK0~TCK2 bit and the TMCR1 register's TCK0~TCK2 bit must be set to the same value (same count source).
Count	An increasing or decreasing count. if that TM0 register and the TMGRA0 register match in the process of increment count, both TM0 and TM1 become decrements; If the TM1 register changes from '0000H' to 'FFFFH' during the process of decreasing the count, both TM0 and TM1 become incremental.
PWM waveform	PWM Cycle: 1/fk (m+2-p) 2 Note2 Dead time: p Phase Active Level Width: 1/fk(m-n-p+1) 2 Inverted Effective Level Width: 1/fk(n+1-p) 2 fk: Frequency of the count source m:Setting value for the TMGRA0 register n: Setting value for the TMGRB0 register (PWM output 1) Setting value for the TMGRA1 register (PWM output 2) Setting value for the TMGRB1 register (PWM output 3) p:Setting Value for the TM0 Register  n+1-p p m-p-n+1 (With L Active)
Count Start Condition	Write "1" to the TSTART0 and TSTART1 bits of the TMSTR register.
Count stop condition	Write "0" to the TSTART0 bit and the TSTART1 bit when the CSEL0 bit of the TMSTR register is "1" (The PWM output pin outputs the initial output level selected by the OLS0 bit and the OLS1 bit of the TMFCR register).
Generation sequence of interrupt requests	· Comparison matching (same contents of TMi register and TMGRji register) ·TM1 Underflow
TMIOA0 Pin Functionality	I/O port or TMCLK (external clock) input
TMIOB0 Pin Functionality	PWM Output 1 Forward Phase Output Pin
TMIOD0 Pin Functionality	PWM output 1 inverting output pin
TMIOA1 Pin Functionality	PWM Output 2 Forward Phase Output Pin
TMIOC1 Pin Functionality	PWM output 2 inverting output pin
TMIOB1 Pin Functionality	PWM Output 3 Forward Phase Output Pin
TMIOD1 Pin Functionality	PWM output 3 inverting output pin
TMIOC0 Pin Functionality	Inverting output is performed every 1/2 PWM cycles.
INTP0 Pin Functionality	Input of the pulse output forcing cut-off signal (input private port or INTP0 interrupt input)

read timer	If you read the TMi register, you can read the count value.
write timer	Can write TMi register.
Select Features	· The input of the pulse output compulsory cutoff signal (reference to "10.4.4 compulsory cutoff of pulse output") · Selection of active and initial output levels for forward and reverse phases · Selection of transmission timing of buffer registers

Note:1. Select fHOCO only if the FRQSEL4 bit of the user option byte (000C2H) is "1" To select fHOCO as the counter source for timer M, fCLK must be set to fIH before bit4 (TMMEN) of Peripheral Admission Register 1 (PER1). If you want to change fCLK to a clock other than fIH, you must change it after clearing the bit4 (TMMEN) of peripheral enable register 1 (PER1).

2. The PWM cycle is fixed after the start of count.

Note: i=0,1,j=A,B,C,D

(1) running example

Figure 10-59 Output Model of Complementary PWM Mode

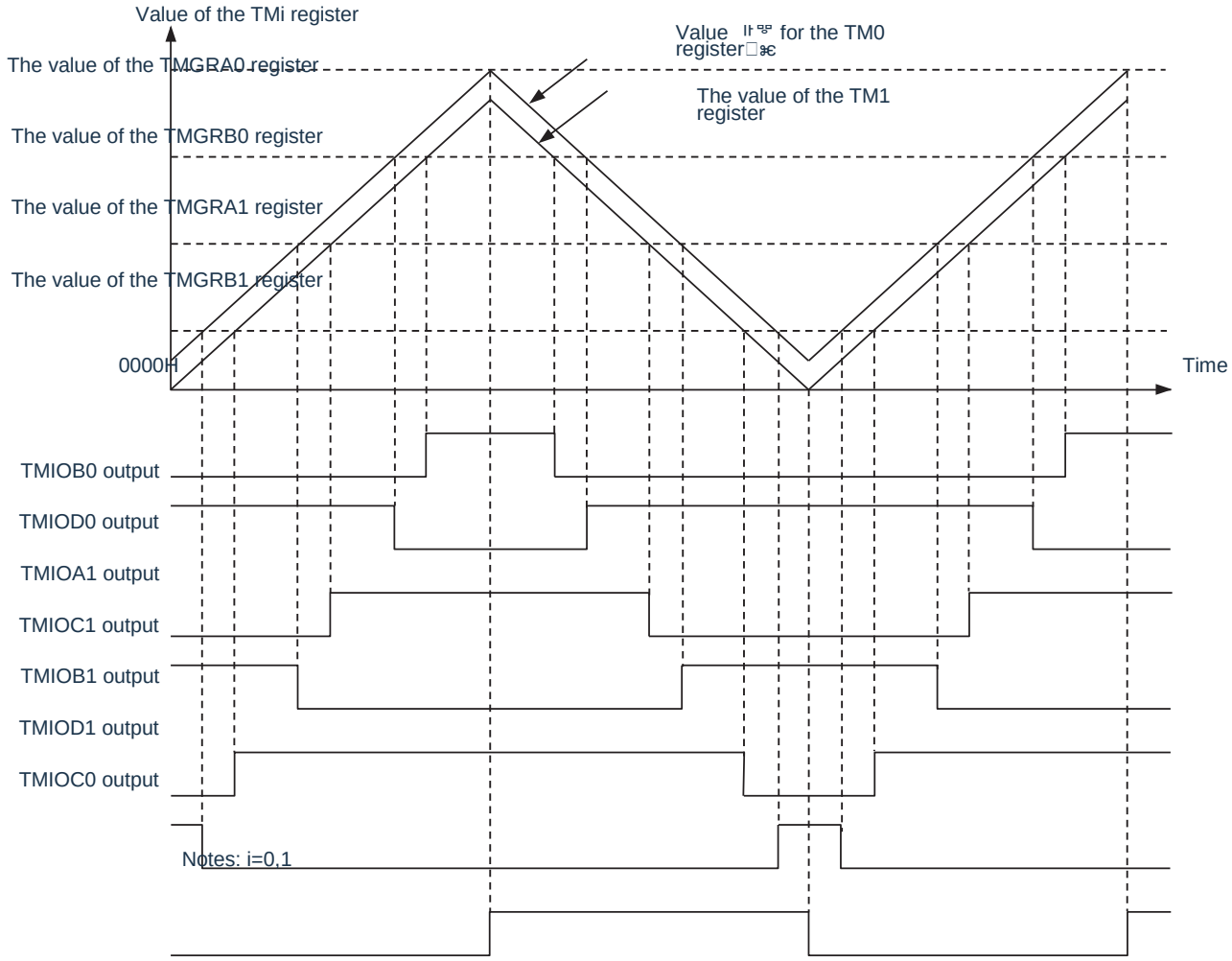
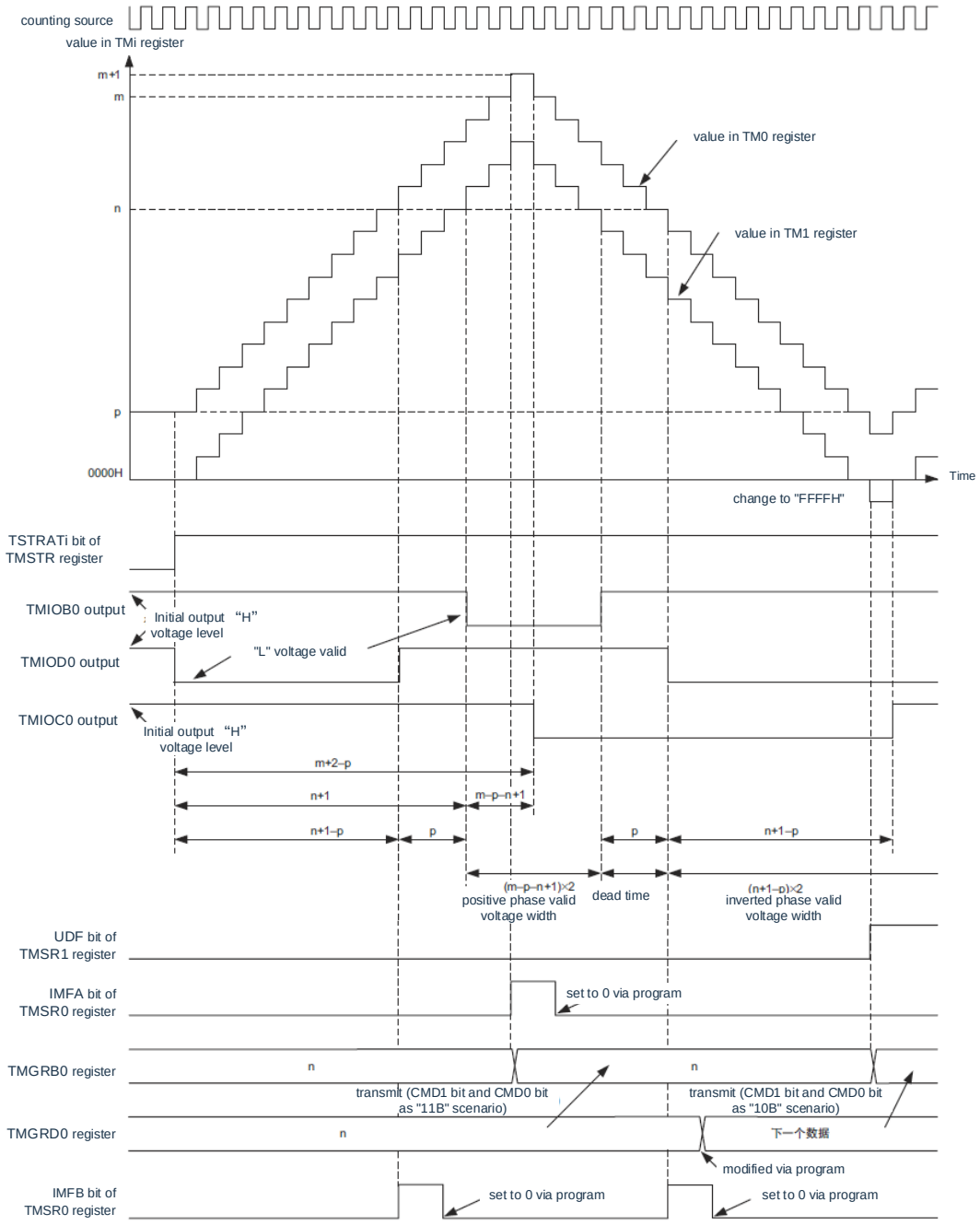


Figure 10-60 Running example of complementary PWM mode



Note: CMD0, CMD1:bit of the TMFCR register
i=0,1
Setting value for the m:TMGRA0 register
Setting value for the n:TMGRB0 register
Setting value for the p:TM0 register
The conditions in the above diagram are as follows:
The OLS1 and OLS0 bits of the TMFCR register are "0" (initial output "H" level, "L" level is valid)

(2) Data transfer sequence of buffer register

TMGRD0, TMGRC1, TMGRD1 register to TMGRB0, TMGRA1, TMGRB1 register data transfer when the TMFCR register has CMD1 bit and CMD0 bit overflow.

Data transfer occurs when the CMD1 bit and the CMD0 bit are "11B" and the TM0 register and TMGRA0 register match.

10.5.6 PWM3 mode

Output 2 PWM waveforms with same period.

The block diagram and running examples of PWM3 mode are shown in Figure 10-61 and Figure 10-62, respectively, and the specification of PWM3 mode is shown in Table 10-17.

Figure 10-61 Block diagram for PWM3 mode

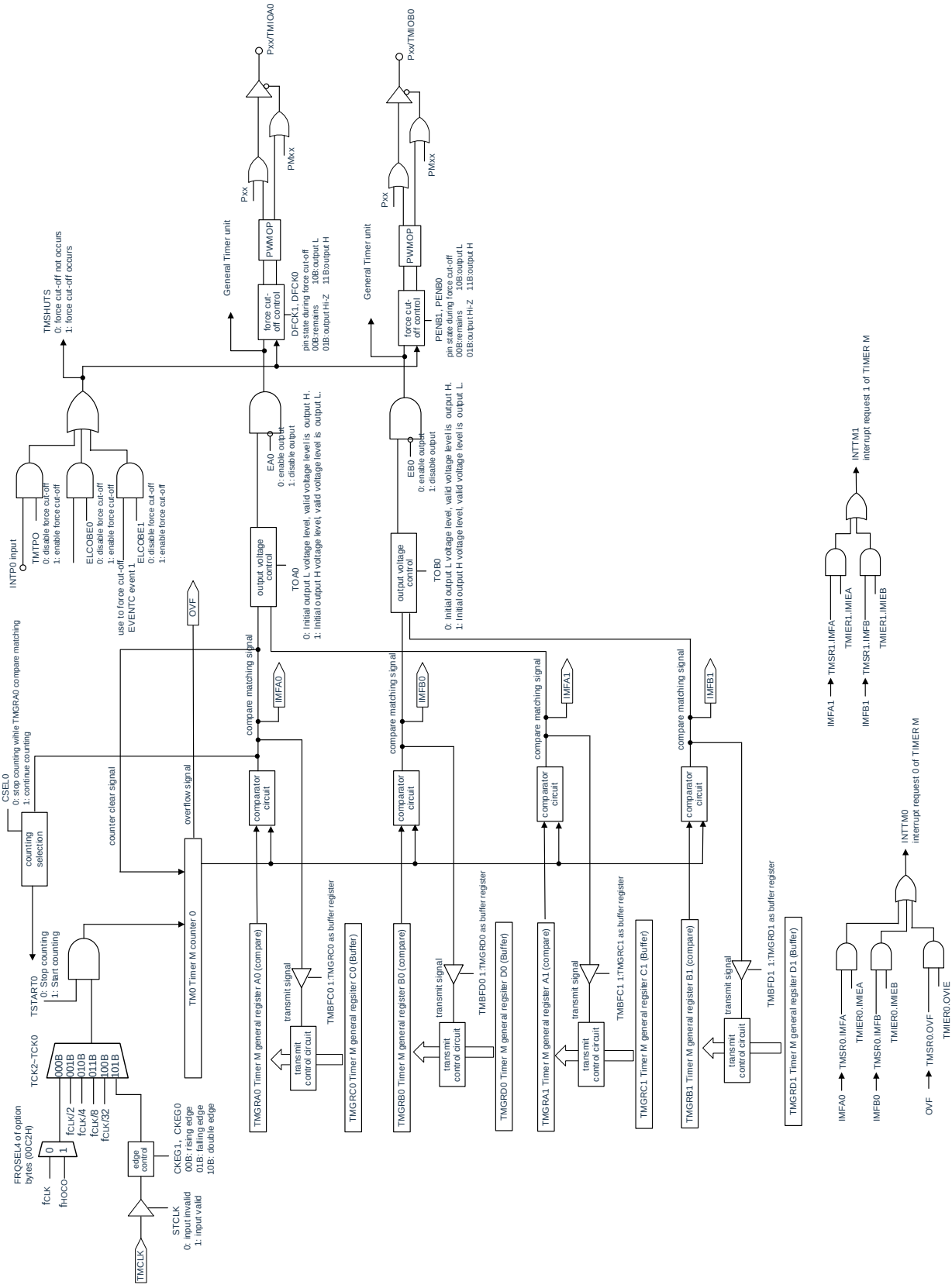
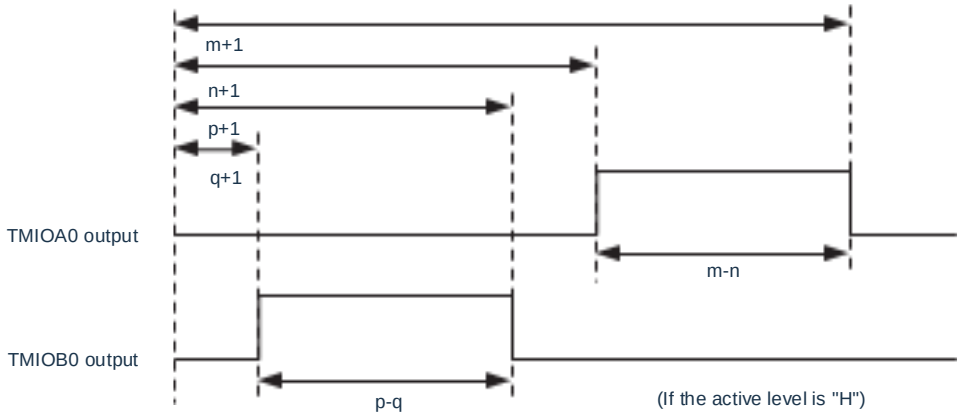


Table 10-17. Specifications for PWM3 Mode

Project	Specifications
count source	fHOCO note, fCLK, fCLK/2, fCLK/4, fCLK/8, fCLK/32
Count	TM0 is an incremental count (TM1 is not used).
PWM waveform	PWM Cycle: 1/fk (m+1) Valid level width for TMIOA0 output: 1/fk (m-n) Effective level width of TMIOB0 output: 1/fk (p-q) fk: Frequency of the count source m:Setting value for the TMGRA0 register n:Setting value for the TMGRA1 register p:Setting value for the TMGRB0 register q:Setting value for the TMGRB1 register 
Count Start Condition	Write "1" to the TSTART0 bit of the TMSTR register.
Count stop condition	· Write "0" to the TSTART0 bit when the CSEL0 bit of the TMSTR register is "1". The PWM output pin maintains the output level before stopping the count. · Stop counting when the CSEL0 bit of the TMSTR register is "0" and a comparison match of the TMGRA0 occurs. The PWM output pin maintains the level after the comparison match causes the output to change.
Generation sequence of interrupt requests	· Comparison matching (same contents of TMi register and TMGRji register) ·TM0 overspill
TMIOA0 Pin and The Function of TMIOB0 Pins	PWM output
TMIOC0, TMIOD0, TMIOA1~TMIOD1 pin function	I/O port
INTP0 Pin Functionality	Input of the pulse output forcing cut-off signal (input private port or INTP0 interrupt input)
read timer	If you read the TM0 register, you can read the count value.
write timer	Can write TM0 register.
Select Features	· The input of the pulse output compulsory cutoff signal (reference to "10.4.4 compulsory cutoff of pulse output") · Selection of effective levels for each pin · Buffer Run (reference to "10.4.2 Buffer Run")

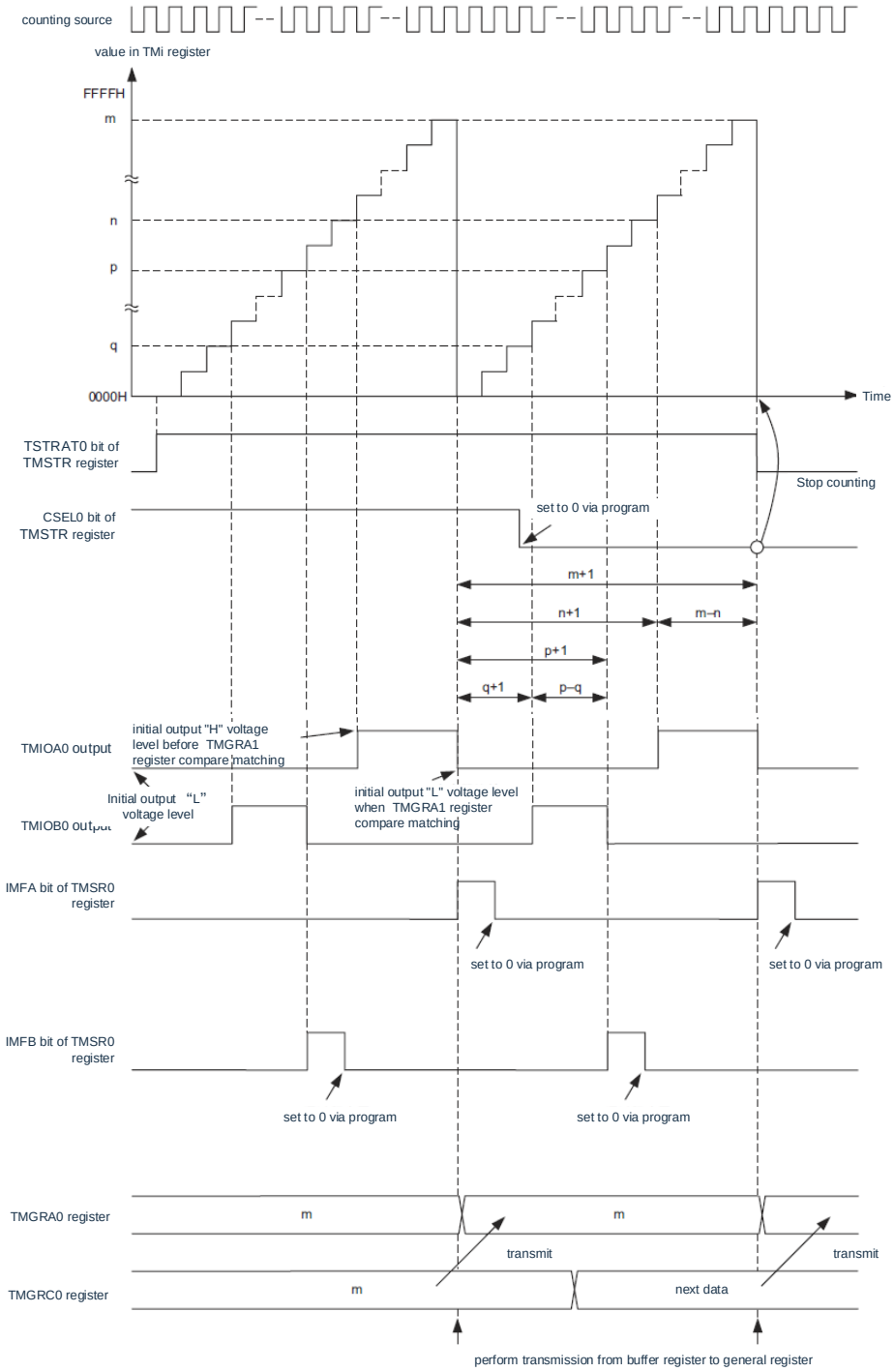
Note: fHOCO can only be selected if the FRQSEL4 bit of the user option byte (000C2H) is "1". To select fHOCO as the counter source for timer M, fCLK must be set to fIH before bit4 (TMMEN) of Peripheral Admission Register 1 (PER1). If you want

to change fCLK to a clock other than fIH, you must change it after clearing the bit4 (TMMEN) of peripheral enable register 1 (PER1).

Note: i=0,1,j=A,B,C,D

(1) running example

Figure 10-62 Running example of PWM3 mode



Remarks: j=A,B
Setting value for the m:TMGRA0 register
Setting value for the n:TMGRA1 register
Setting value for the p:TMGRB0 register
q:Setting value for the TMGRB1 register

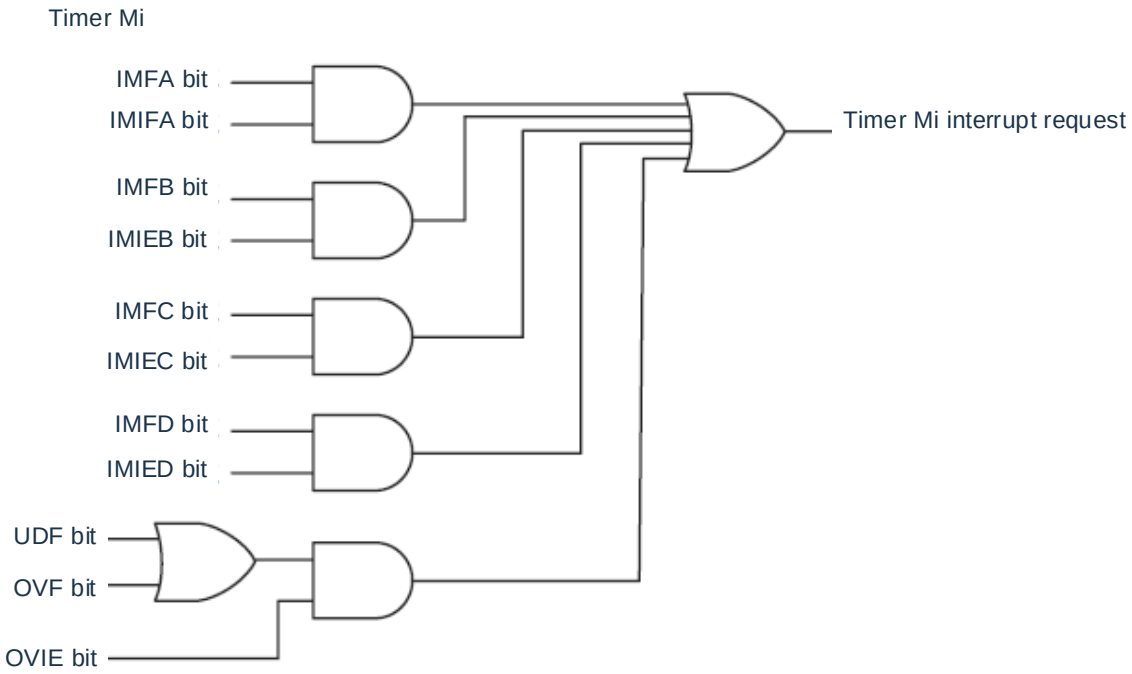
The conditions in the above diagram are as follows:
The TMOCR register TOA0 bit and TOB 0 bit are "0" (initial output "L"level, output "L" level when TMGRj0 registers match).
The TMBFC0 bit of the TMMR register is '1' (TMGRC0 register is a buffer register for TMGRA0 register)

and timer M1. The relevant registers for timer M interrupts are shown in Table 10-18, and the block diagram for timer M interrupts is shown in Figure 10-63.

Table 10-18. Correlation register interrupted by timer M

	State register for timer M	Interrupt permit register for timer M	interrupt request flag (Register)	interrupt mask flag (Register)
Timer M0	TMSR0	TMIR0	TMIF0 (IF2H)	TMMK0 (MK2H)
Timer M1	TMSR1	TMIR1	TMIF1 (IF2H)	TMMK1 (MK2H)

Figure 10-63 Block diagram of timer M interrupt



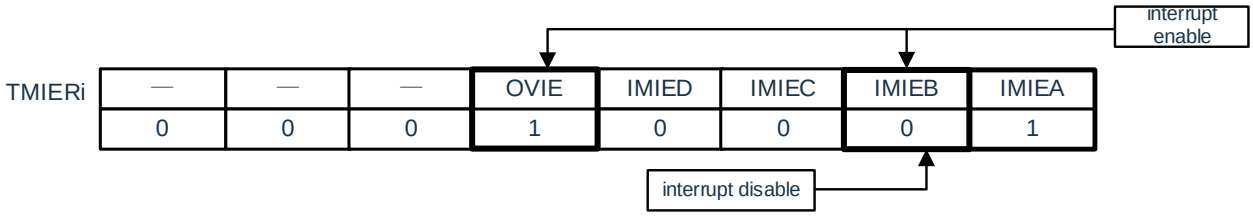
i = 0、1
IMFA、IMFB、IMFC、IMFD、OVF、UDF：TMSri register bits
IMIEA、IMIEB、IMIEC、IMIED、OVIE： TMIERi register bits

Since timer M generate 1 interrupt requests (interrupt timer M) from multiple interrupt request sources, there are following differences with respect to other maskable interrupts:

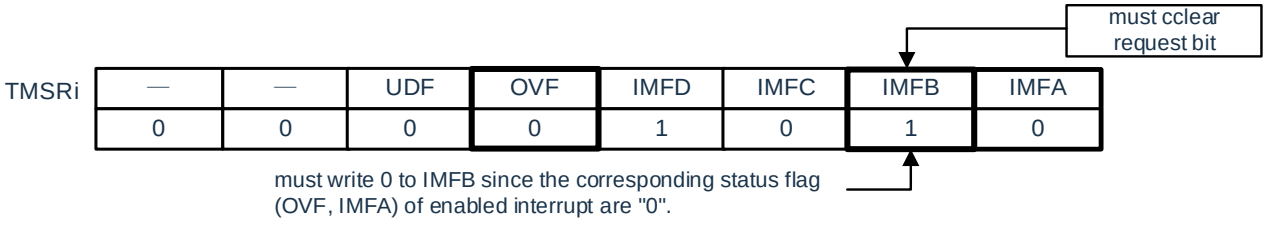
- When the TMSRi register has bit '1' and the corresponding TMIERi register has bit '1' (interrupt allowed), the TMIFi bit becomes '1'.
- When multiple bits of the TMIERi register are '1', it is necessary to determine which interrupt is generated by which request source through the TMSRi register.
- Because even if you accept interrupts, the TMSRi register members do not automatically become "0", you must "0" these locations in the interrupt.
- When the status flag of an interrupt source of timer M (hereinafter referred to as object status flag) is set to 0, interrupt must be
  - The object status flag must be written "0" after setting timer M interrupt enable register i (TMIERi).
  - When timer M interrupt enable register i (TMIERi) has bit with" 1" (allowed) and the bit allow interrupt source state flag" 0.

(Example) Cleaning IMFB in a state where IMIEA and OVIE are interrupts allowed and IMIEB is interrupts prohibited

- Timer M interrupts the state of the allow register i(TMIERi)



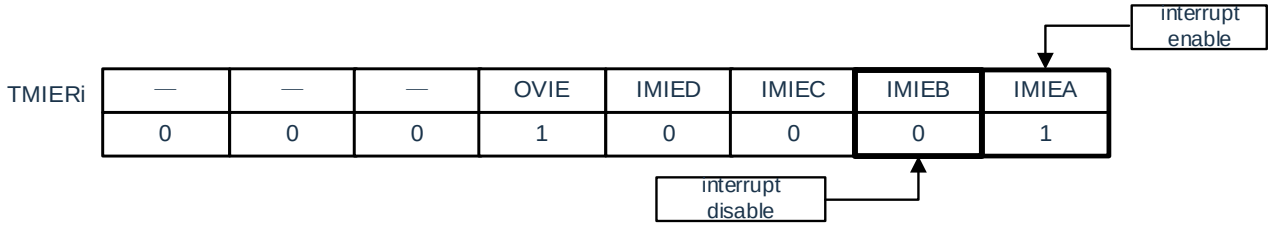
· Status of timer M status register i (TMSRi)



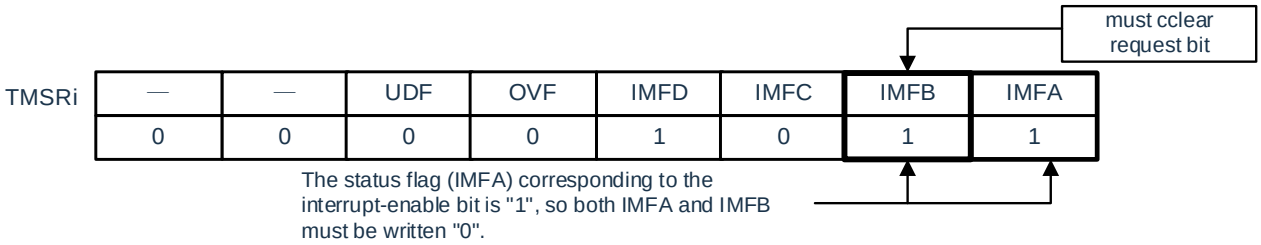
a) When timer M interrupt enable register i (TMIERi) has bit with '1' (allowed) and the bit allows interrupt source state flag '1'.

(Example) When IMIEA is Interrupt Allowed and IMIEB is Interrupt Disabled to clear IMFB

· Timer M interrupts the state of the allow register i(TMIERi)



· Status of timer M status register i (TMSRi)



10.7 Precautions when using timer M

10.7.1 Read and write access to SFR

To set timer M, the TMMEN location of the PER1 register must be "1" first. When the TMMEN bit is '0', the write operation of the control register of timer M is ignored and the read values are initial (except for port registers and port mode registers).

The following registers are not allowed to be overwritten during the count:

TMELC register, TMMR register, TMPMR register, TMFCR register, TMOER1 register, TMPTO bit of TMOER2 register, TMDFi register, TMCRi register, TMIORAi register, TMIORCi register, TMPOCRi register

(1) TMSTR register

- The TMSTR register can be set by an 8-bit memory operation instruction.
- When the TMSTR register has a CSELi bit (i=0,1) of '0' (stopping counting when comparing TMi register to TMGRAi register), the TSTARTi bit remains unchanged. The TSTARTi bit becomes '0' only when the TMGRAi register match is being compared.

When rewriting the TMSTR register, if the CSELi bit is changed to "1" without affecting the count with the CSELi bit "0".

If you write "1" to the TSTARTi bit while the counter stops counting, counting may begin. To stop the count through the program, you must write "0" to the TSTARTi bit after the CSELi position "1". The count cannot be stopped even if "1" and "0" are written to both the CSELi and TSTARTi bits simultaneously.

· The output level during the count stop period when the TMIOji pin (j=A, B, C, D) is used for the timer M output is shown in Table 10-19.

Table 10-19. Output level of TMIOji pins (j=A, B, C, D) when stopping counting

Method for stopping counting	Output level of the TMIOji pin when stopping the count
When the CSELi bit is "1", stop counting by writing "0" to the TSTARTi bit.	The output level before stopping counting is maintained (in the complementary PWM mode or reset synchronous PWM mode of the timer M, the OLS0 bit and OLS1 bit selected initial output level).
When the CSELi bit is "0", the count stops when the TMi register and the TMGRAi register match.	The level after which the comparison match causes the output change is maintained (the initial output level selected by the OLS0 bit and OLS1 bit of the TMFCR register).

Note: i=0,1,j=A,B,C,D

(2) TMDFi register (i=0,1)

The count must start after setting the DFCK0 and DFCK1 bits of the TMDFi register.

(3) TMi register (i=0,1)

If the value of the TMi register becomes the time sequence of the "0000H" and the time sequence of writing the TMi register overlaps, the register is preferentially written.

10.7.2 mode switching

- To switch modes during the run, you must switch after entering the count stop state (TSTART0 bit and TSTART1 position '0').
- Before changing the TSTART0 and TSTART1 bits from "0" to "1", the TMIF0 and TMIF1 positions "0" must be Refer to "Chapter 25 Interrupt Features" for details.

10.7.3 count source

- To switch the count source, you must switch after stopping the count.

  [Change Steps]

  1) Set the TSTARTi bit (i=0,1) of the TMSTR register to "0".
  2) Change the TCK0~TCK2 bit of the TMCRi register.
- To select fHOCO (64MHz or 48MHz) as the counter source for timer M, you must set fCLK to fIH before setting the bit4 (TMMEN) of the Peripheral Admission Register 1 (PER1). If you want to change fCLK to a clock other than fIH, you must change it after clearing the bit4 (TMMEN) of peripheral enable register 1 (PER1).

10.7.4 Input capture function

- The pulse width of the input capture signal must be at least 3 operating clock cycles of timer M.
- After the input capture signal is input from TMIOji pin (j=A, B, C, D), it is necessary to wait 2 to 3 runtime clock (fCLK) cycles of timer M and then transfer the value of TMi register to TMGRji register (without digital filter).
- In the input capture mode, if the TMTSTARTi bit of the TMSTR register is '0' (stop count) and the IOj0 bit of the TMIORki register is input to the TMIOji pin, an input capture interrupt request (i=0,1,j=A,B,C,k=A, C) is generated at the effective edge of the TMIOji input.

10.7.5 Set-up steps for TMIOAi, TMIOBi, TMIOCi, TMIODi pins (i=0,1)

- After reset, the TMIOAi, TMIOBi, TMIOCi, TMIODi pin's I/O port is used as the input port.
- To export from TMIOAi, TMIOBi, TMIOCi, TMIODi pins, you must follow these steps.

  [Change Steps]

  1) Sets the mode and initial value.
  2) Place the TMIOAi, TMIOBi, TMIOCi, TMIODi pins as the allowed output (TMOER1 register).
  3) Position "0" of port register corresponding to TMIOAi, TMIOBi, TMIOCi, TMIODi pin.
  4) Set the bit of the port mode register corresponding to the TMIOAi, TMIOBi, TMIOCi, TMIODi pin to the output mode (output from TMIOAi, TMIOBi,TMIOCi pin).
  5) Start counting (TSTART0 bits and TSTART1 position "1").

- To change the bit of the port mode register for the TMIOAi, TMIOBi, TMIOCi, TMIODi pins from output mode to input mode, you must follow these steps.
- [Change Steps]

  1) (1) Set the bit of the port mode register corresponding to TMIOAi, TMIOBi, TMIOCi, TMIODi pin to the input mode (from TMIOAi, TMIOBi, TMIOCi,TMIODi pin).
  2) (2) Set as input capture function.
  3) (3) Start counting (TSTART0 bits and TSTART1 position "1").

- When TMIOAi, TMIOBi, TMIOCi, TMIODi pins are switched from output mode to input mode, input capture may be run depending on the status of the pins. when a digital filter is not used, edge detection is performed at least after two CPU clock cycles; When digital filter is used, it needs 5 sampling clock cycles of digital filter to detect the edge.

10.7.6 external clock TMCLK

The external clock input by the TMCLK pin must have a pulse width of at least 3 operating clock cycles of timer M.

Reset synchronous PWM mode

- When this mode is used for motor control, it must be used under OLS0=OLS1 conditions.
- To be set to reset the synchronous PWM mode, you must follow the steps below.

  [Change Steps]

  1) Place the TSTART0 location of the TMSTR register "0" (stop count).
  2) Position "00B" (timer mode, PWM mode, and PWM3 mode) at the CMD1 bit and CMD0 of TMFCR registers.
  3) Reset the CMD1 bit and the CMD0 position "01B".
  4) Reset other relevant registers for timer M.

10.7.7 complementary PWM mode

- When this mode is used for motor control, it must be used under OLS0=OLS1 conditions.
- When you want to change the CMD0 and CMD1 bits of the TMFCR register, you must follow the steps below.

  [Change steps: set to complementary PWM mode (including reset), or changing buffer register to general register data transfer timing in complementary PWM mode]

  1) Place both the TSTART0 and TSTART1 bits of the TMSTR register at "0".
  2) Position "00B" (timer mode, PWM mode, and PWM3 mode) at the CMD1 bit and CMD0 of TMFCR registers.
  3) Position the CMD1 bit and the CMD0 position "10B" or "11B" (complementary PWM mode).
  4) Reset other relevant registers for timer M.

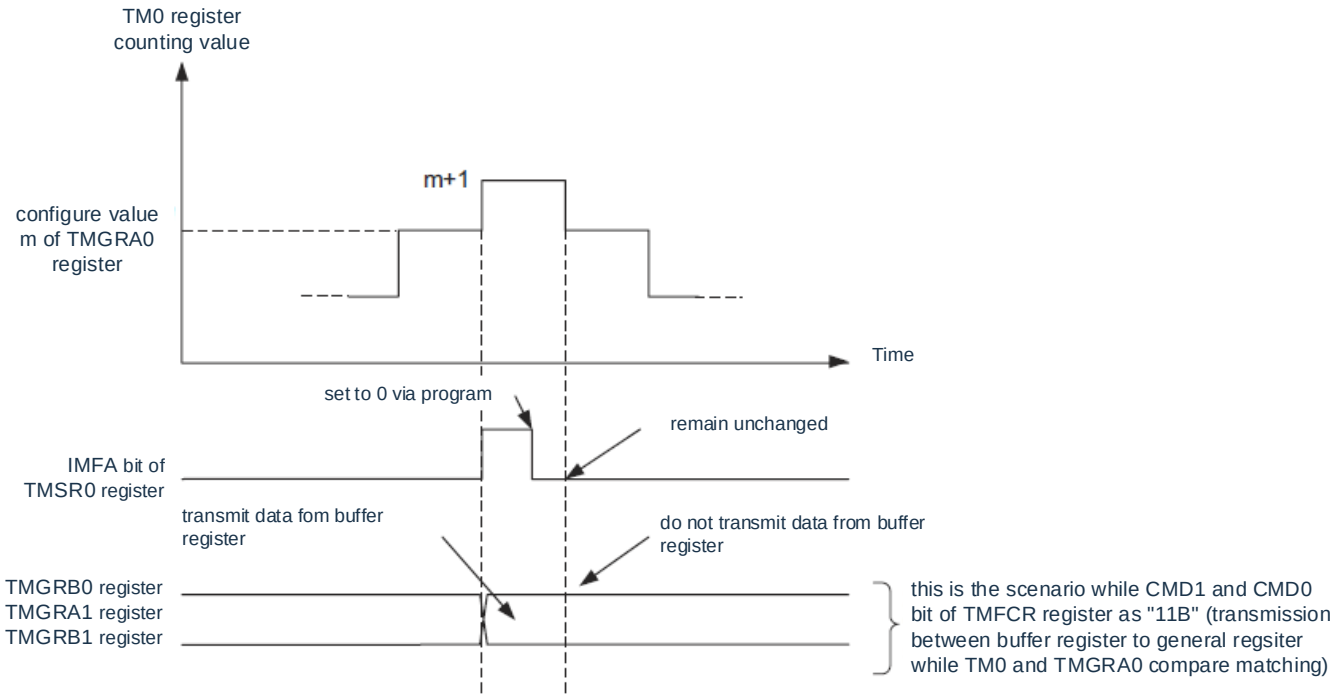
  [Change steps: Case of Terminating Complementary PWM Mode]
  (1) Place both the TSTART0 and TSTART1 bits of the TMSTR register at "0".
  (2) CMD1 bit and CMD0 position "00B" (timer mode, PWM mode and PWM3 mode)

- Cannot write TMGRA0, TMGRB0, TMGRA1, TMGRB1 registers during the run.
- To change the PWM waveform, the write values of TMGRD0, TMGRC1, TMGRD1 registers must be transferred to TMGRB0, TMGRA1,TMGRB1 registers. However, when writing TMGRD0, TMGRC1, and TMGRD1, TMBFD0 bits and TMBFD1 position "0" must be first written to these registers. Thereafter, the TMBFD0 bit, TMBFC1 bit and TMBFD1 position "1" (buffer register) are enabled. The PWM cycle cannot be changed.
- Assuming the TMGRA0 register is set to m, the TM0 register is m-1 m when it changes from an increasing count to a decreasing count

  Count of m+1 m m-1.
- The IMFA bit of the TMSRi register becomes '1' when the m m+1 increment count occurs. When the CMD1 bit and CMD0 bit of the TMFCR register are "11B" (complementary PWM mode, when TM0 and TMGRA0 registers are compared and matched), the contents of the buffer registers (TMGRD0, TMGRC1, TMGRD1) are transferred to the universal registers (TMGRB0, TMGRA1, TMGRB1).

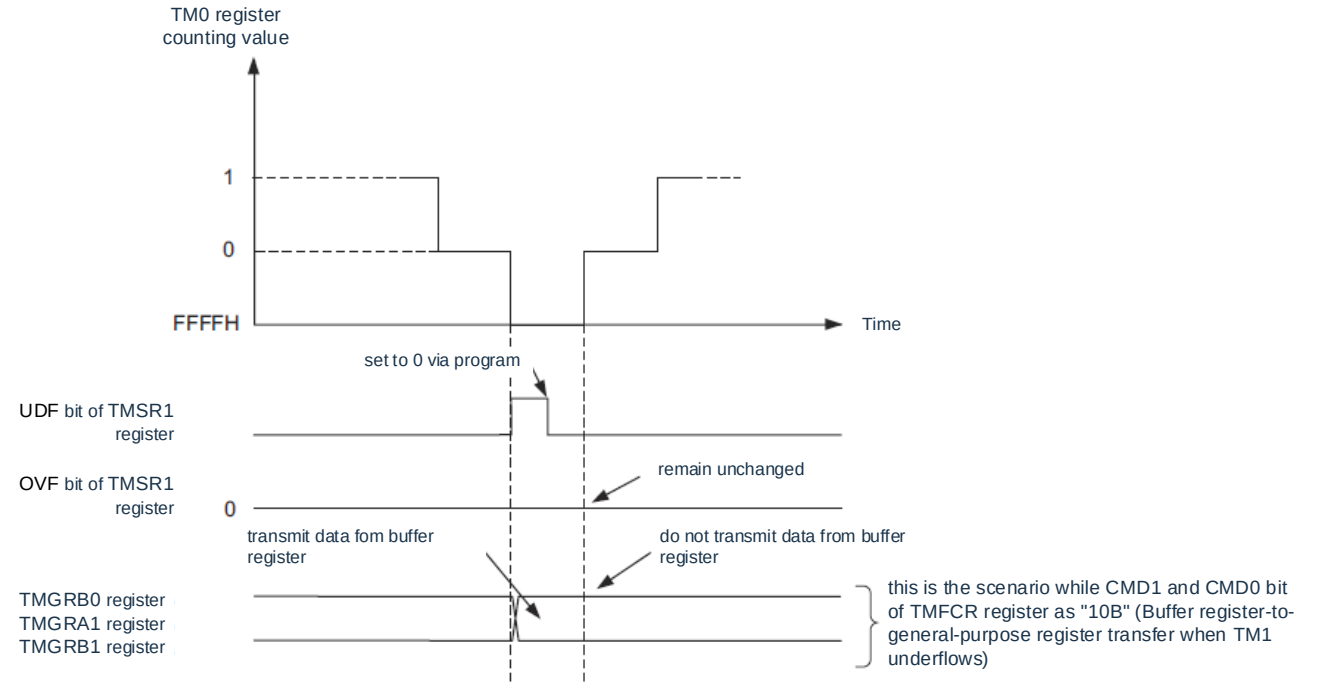
  The IMFA bit does not change when m +1m -1 decreasing count is performed, and the data is not transferred to a register such as TMGRA0.

Figure 10-64 Running example of TM0 and TMGRA0 register matching in complementary PWM mode



- When the count is changed from decreasing to increasing, TM1 counts the 1 0 FFFFH 0 1. The UDF bit of the TMSRi register becomes "1" when a 10FFFFH decreasing count is performed. When the TMFCR register has a CMD1 bit and a CMD0 bit of '10B' (complementary PWM mode, buffer data is transmitted when TM1 underflows), the contents of buffer registers (TMGRD0, TMGRC1, TMGRD1) are transmitted to universal registers (TMGRB0,TMGRA1,TMGRB1).
  When FFFFH 0 1 is counted incrementally, data is not transferred to registers such as TMGRB0, and the OVF bit of the TMSRi register is unchanged.

Figure 10-65 Running example of TM1 underflow in complementary PWM mode

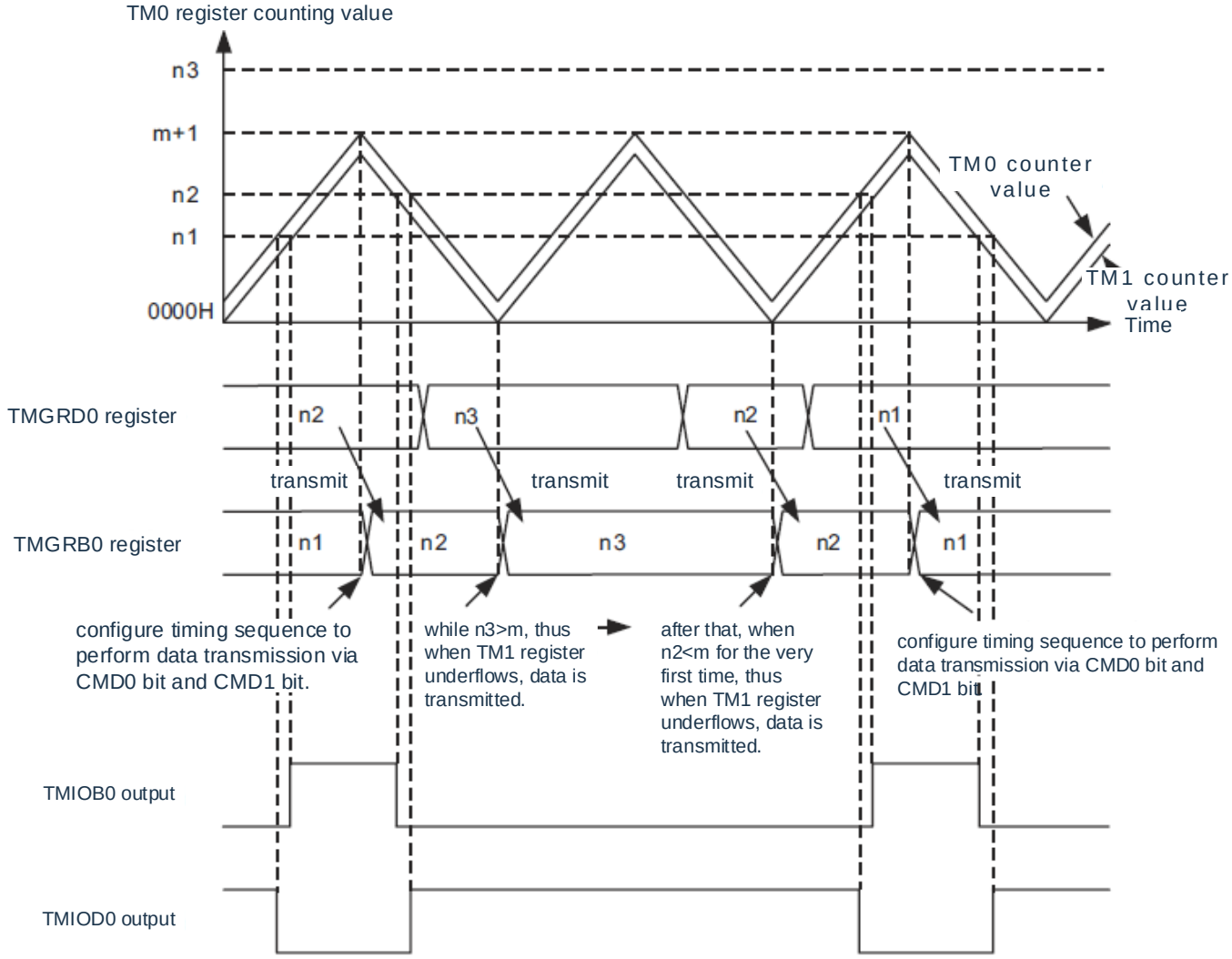


- The data transfer timing of the buffer register to the general register must be selected through the CMD0 and CMD1 bits of the TMFCR register. However, in the case of 0% duty cycle and 100% duty cycle, the values of the CMD0 bits are independent of the following transmission sequence.

  Buffer Register Value≥The case of the TMGRA0 register value (0% duty cycle) is transferred when the TM1 register underflows. Thereafter, if a value is set to the buffer register (0001H≤<TMGRA0 register value), data is transferred to the general register when the TM1 register underflows 10 times. Data is then transmitted through the timing selected by the CMD0 bit and the CMD1 bit.

  However, a waveform with a 0% duty cycle cannot be generated when the buffer register has an initial value of 'FFFFH'. When generating a 0% duty waveform, the value of the buffer register ≥the value of the TMGRA0 register must be made by writing the buffer register.

Figure 10-66 The Value of Buffer Register in Complementary PWM Mode≥An Example of Running When Value of TMGRA0 Register



If the buffer register is set value (value ≥TMGRA0 register value), the buffer register value is transferred to the general register when the TM1 counter overflows, and output level as fixed pattern as positive phase 100% duty cycle and negative phase 0% duty cycle, also not relevant to setting of CMD0 bit.

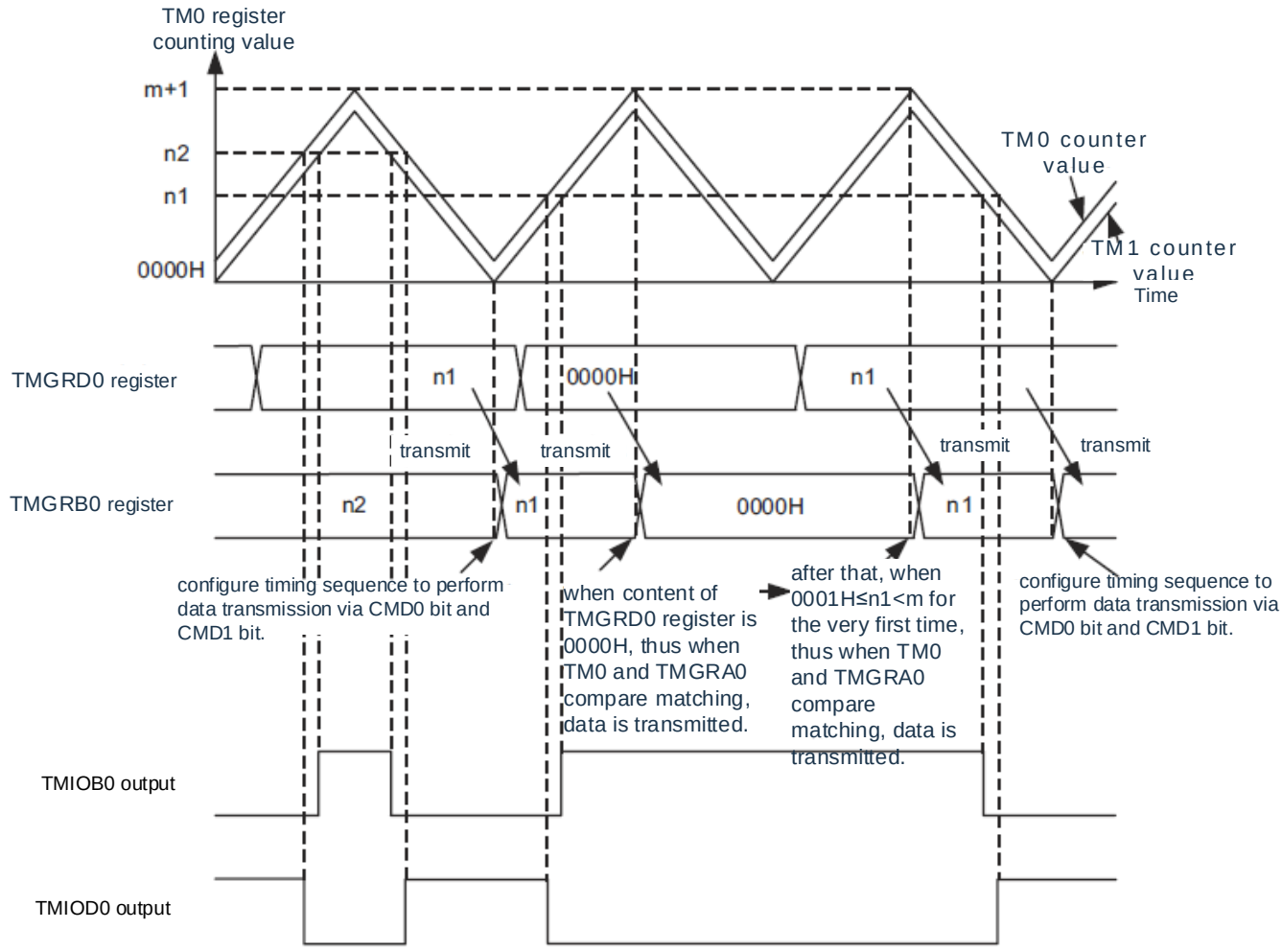
To unset the output level, the buffer register must be set a value (TM0 register value≤set value≤ (TMGRA0 value -TM0 register value). After writing the buffer, regardless of the setting of CMD0 bits, the buffer register value is transferred to the general register and the PWM waveform is output when the TM1 counter underflows. After outputting the PWM waveform, the value of the buffer register is transmitted to the general register through the timing set by the CMD0 bit.

However, you cannot set positive 100% duty cycle and negative 0% duty cycle output with the initial value 'FFFFH' of the buffer register. Also can not directly change from positive 100% duty ratio and negative 0% duty ratio to positive 100 duty ratio.

The buffer register value of "0000H" (100% duty cycle) is transmitted when the TM0 and TMGRA0 registers match.

Thereafter, if a value is set to the buffer register (0001H≤ set value <TMGRA0 register value), data is transferred to the general register when the TM0 and TMGRA0 registers have been set for the first comparison match. Data is then transmitted through the timing selected by the CMD0 bit and the CMD1 bit.

Figure 10-67 An example of running when the buffer register value in the complementary PWM mode is "0000H"



If the buffer register is written "0000H", the buffer register value is transferred to the general register when the TM0 register and the TMGRA0 register match.

To unset the output level, the buffer register must be set a value (TM0 register value≤set value≤ (TMGRA0 value -TM0 register value). After writing the buffer, regardless of the setting of CMD0 bits, the buffer register value is transferred to the general register and the PWM waveform is output when the TM1 counter underflows. After outputting the PWM waveform, the value of the buffer register is transmitted to the general register through the timing set by the CMD0 bit.

You cannot change the output from positive 0% and negative 100% duty directly to positive 100% duty.

10.8 PWMOP

The PWMOP unit can implement the output mandatory cut-off function of TimerM. Cutoff sources can be selected from CMP0, INTP0, and EVENT. This is different from the TimerM's own impulse-Forced cutoff.

Table 10-20 Compulsory Cut-off of Pulse Output and Functional Comparison of PWMOP

	Forced cut off of TimerM pulse output	Output Forced Cutoff for PWMOP
Mode to support forced cutoff	PWM function Reset synchronous PWM mode complementary PWM mode PWM3 mode	All mode port output that supports TimerM can also be forced to shut down
Forced Cutoff Source	Event Input for EVENTC INTP0 input	Event Input for EVENTC INTP0 input Output from comparator 0
Force Cutoff Relieved Source	Stop Counter, Software Undoes	hardware removal Software Undock (no need to stop counters)
Pin that can force a cutoff	Pxx/TMIOA0, Pxx/TMIOB0, Pxx/TMIOC0, Pxx/TMIOD0, Pxx/TMIOA1, Pxx/TMIOB1, Pxx/TMIOC1, Pxx/TMIOD1. (based on TimerM output settings)	Pxx/TMIOA0, Pxx/TMIOB0, Pxx/TMIOC0, Pxx/TMIOD0, Pxx/TMIOA1, Pxx/TMIOB1, Pxx/TMIOC1, Pxx/TMIOD1. The output of the PORT can also be forced off
PORT Status at Forced Cutoff	HI- ZL level H level	The HI-ZL level H level can only output HI-Z if the forced cut off of the PORT output is selected

Note: You cannot select the same source when you use both the pulse force cutoff and the output force cutoff functions.

10.8.1 Features of PWMOP

PWMOP enables the following:

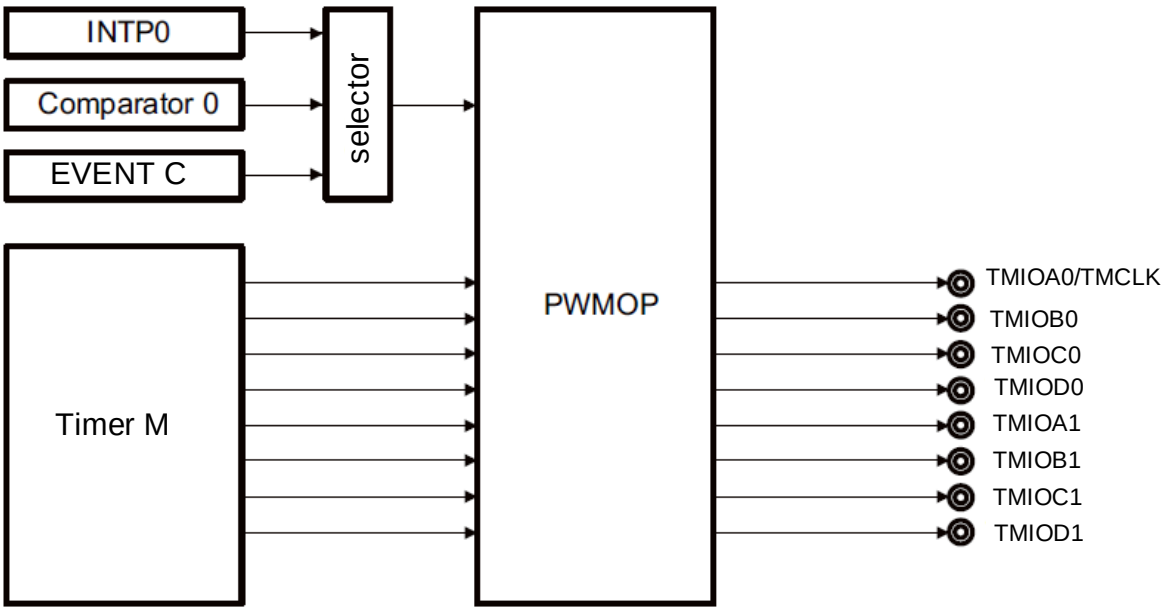
You can select the output of comparator 0, the INTP0 input, and the event input of EVENTC as the source of the output Forced cutoff.

When you select the output of comparator 0 and the INTP0 input as the source, you can choose to follow the checkout.

Output may be unforced by software or hardware.

When forced off, the output level may be selected from the "H" level, the "L" level and Hi-Z.

Figure 10-68 Block diagram of PWMOP



10.8.2 Register for PWMOP

The register for PWMOP is shown in Table 10-21.

Table 10-21 Control Registers for PWMOP

register name	symbol
PWMOP control register 0	OPCTL0
PWMOP Force Cutoff Control Register 0	PDF0
PWMOP Force Cutoff Control Register 1	PDF1
PWMOP along selection register	OPEDGE
PWMOP status register	OPSR

1) PWMOP control register 0 (OPCTL0)

Figure 10-69 Format of PWMOP control register 0

Address: 0x40043C58 After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
OPCTL0	0	HAZARD_SET	IN_EG	IN_SEL1	IN_SEL0	ACT	HZ_REL	HS_SEL

HAZARD_SET	Output Force Cutoff hazard Control Note 1
0	Ban hazard policies
1	Allow hazard countermeasures

IN_EG	Output Forced Cutoff Source/Output Forced Cutoff Relieved Source Edge Selection 2,3
0	Rising edge: Output Force Cutoff Descender Edge: Output Force Cutoff Relief
1	Rising edge: Output Force Cut Off Down Edge: output forced cut-off

IN_SEL1	IN_SEL0	Selection of Cutoff Sources Note 2, 4, 5
0	0	Do not select
0	1	Output from comparator 0
1	0	INTP0 input
1	1	Event input from EVENTC

ACT	Timing Selection of Software Release during Software Release
0	When the HZ_SEL is set to 1 by software, the forced cut-off is removed and the pulse output is recovered
1	When HZ_SEL is 1, the time sequence of pulse output recovery is as follows: Timer M complementary PWM mode: When the count value of TM0 is 0000H, the pulse output reverts to the mode below: when the count value of TM0 is 0000H, when the count value of TMIOj0 (j=A, B, C, D) is 0000H, TMIOj1 (j=A, B, C, D) forces the cut-off to cancel TM1

HZ_REL	When software is removed: Output Force Cutoff Release Control Note 7
0	Keep output forced cutoff (HZ_REL bit becomes 0 when output forced cutoff is lifted)
1	Force Cut Off, Pulse Output Recovery Note 8
The readout values of HZ_REL bit are different according to the state: Normal state: 1/0 write, 0 sense output forced off: Only write 1 and read 1	

HS_REL	Output Forced Cutoff Mode Selection
0	Hardware removal: When the hardware is used to release the forced cut-off of the output, the timing is different according to the different operation modes of the timer M. Timer M complementary PWM mode: Upon detection of the release source, the edge of the TMIOC0 selected according to the OPEDGE is forced off. When the timer M resets the synchronous PWM mode: When the count value of TM0 is 0000H, the pulse output is recovered TMIOj0 (j=A, B, C, D) is forced to terminate when the count of TM0 is 0000H TMIOj1 (j=A, B, C, D) Force Cutoff Release 9 for TM1 with a count of 0000H
1	software removal

Note 1: Timer M cannot be overridden during action

Note 2: Set IN_SEL1 and IN_SEL0 after at least three clocks after IN_EG.

Note 3: Enable when selecting comparator 0 output and INTP0 input.

Note 4: When you use EVENTC to unforce a cutoff, you must select Software Undoing (HS_SEL is set to 1). There are no restrictions when using comparator 0 output and INTP0 input.

Note 5: The effective width of the select comparator 0 output and the INTP0 input must be greater than one clock cycle.

NOTE 6: Count values of TM0 and TM1 = 0000H means the time when the counters bit15~bit0 are all 0.

Note 7: If the timer M works in the output comparison function, PWM function or PWM3 mode, using two channels and using one channel, the termination action of the output is different.

When using 2 channels:

If HZ_REL is set to 1 by the software, the output cutoff bit (HZOF0, HZOF1) becomes all 0 and the HZ_REL bit becomes 0.

When using 1 channel:

If HZ_REL is set to 1 by the software, the output cutoff bit (HZOF0 or HZOF1) for the channel in use becomes 0.

Note 8: Cannot be set to 1 if no forced cutoff occurs

Note 9: When the timer M works in the output comparison function, the PWM function or the PWM3 mode, the channel without action cannot release the forced output cutoff. (HZOF0 and HZOF1 do not become 0).

(1) PWMOP Enforce Cutoff Control Register 0 (OPDF0)

Figure 10-70 PWMOP Format of Forced Cutoff Control Register 0

Address: 0x40043C59 After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PDF0	DFD01	DFD00	DFC01	DFC00	DFB01	DFB00	DFA01	DFA00

DFD01	DFD00	TMIOD0 Pin Output Force Cut-off Control
0	0	Disable Mandatory Cutoff
0	1	Output Hi-Z
1	0	output L level
1	1	output H level

DFC01	DFC00	TMIOC0 Pin Output Force Cut-off Control
0	0	Disable Mandatory Cutoff
0	1	Output Hi-Z
1	0	output L level
1	1	output H level

DFB01	DFB00	TMIOB0 Pin Output Force Cut-off Control
0	0	Disable Mandatory Cutoff
0	1	Output Hi-Z
1	0	output L level
1	1	output H level

DFA01	DFA00	TMIOA0 Pin Output Force Cut-off Control
0	0	Disable Mandatory Cutoff
0	1	Output Hi-Z
1	0	output L level
1	1	output H level

Note 1: When the TMIOj0 (j=A,B, C,D) pin is used as the PORT output and enables mandatory cutoff functionality, the Hi-Z output must be selected.

Note 2: The value of the register cannot be changed in the force cutoff state.

Note 3: When using PIOR to redirect TMIOji (j=A,B,C,D;i=0,1) pins, only a single setting is allowed for the same pin.

Example: You must set DFA0n (n=0,1) to 0 when you use PIOR2 to select P17 as the TMIOD0 pin and allow TMIOD0 output.

2) PWMOP Force Cutoff Control Register 1 (OPDF1)

Figure 10-71 PWMOP Format of Forced Cutoff Control Register 1

Address: 0x40043C5A After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PDF1	DFD11	DFD10	DFC11	DFC10	DFB11	DFB10	DFA11	DFA10

DFD11	DFD10	TMIOD1 Pin Output Force Cut-off Control
0	0	Disable Mandatory Cutoff
0	1	Output Hi-Z
1	0	output L level
1	1	output H level

DFC11	DFC10	TMIOC1 Pin Output Force Cut-off Control
0	0	Disable Mandatory Cutoff
0	1	Output Hi-Z
1	0	output L level
1	1	output H level

DFB11	DFB10	TMIOB1 Pin Output Force Cut-off Control
0	0	Disable Mandatory Cutoff
0	1	Output Hi-Z
1	0	output L level
1	1	output H level

DFA11	DFA10	TMIOA1 Pin Output Force Cut-off Control
0	0	Disable Mandatory Cutoff
0	1	Output Hi-Z
1	0	output L level
1	1	output H level

Note 1: When the TMIOj1 (j=A,B, C,D) pin is used as the PORT output and enables mandatory cutoff functionality, the Hi-Z output must be selected.

Note 2: The value of the register cannot be changed in the force cutoff state.

Note 3: When using PIOR to redirect TMIOji (j=A,B,C,D;i=0,1) pins, only a single setting is allowed for the same pin.

Example: You must set DFC1n (n=0,1) to 0 when you use PIOR2 to select P16 as the TMIOA1 pin and allow TMIOA1 output.

3) PWMOP along selection register (OPEDGE)

When the timer M is in complementary PWM mode and the output is forced off by hardware release, the release time point can be set by OPEDGE register.

Figure 10-72 PWMOP along select register format

Address: 0x40043C5B After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
OPEDGE	-	-	-	-	-	-	EG1	EG0

EG1	EG0	Output Force Cutoff Relief Edge Selection
0	0	The rising edge of TMIOC0 is released.
0	1	The descent to TMIOC0 is lifted.
1	0	The rising or falling edge of the TMIOC0 dissolves.
1	1	Do not enable selection, keep off

4) PWMOP Status Register (OPSR)

Figure 10-73 Format of PWMOP Status Register

Address: 0x40043C5C After reset: 00H R

symbol	7	6	5	4	3	2	1	0
OPSR	0	0	0	0	0	HZOF1	HZOF0	HZIF0

HZOF1	Enforce Cutoff Status
0	Typically Timer output (TMIOA1, TMIOB1, TMIOC1, TMIOD1)
1	Force Cutoff State (TMIOA1, TMIOB1, TMIOC1, TMIOD1)

HZOF0	Enforce Cutoff Status
0	Typically Timer output (TMIOA0, TMIOB0, TMIOC0, TMIOD0)
1	Force Cutoff State (TMIOA0, TMIOB0, TMIOC0, TMIOD0)

HZIF0	Output Force Cutoff Source Status Notes 1, 2
0	Output force cutoff source within threshold
1	Output force cutoff source is out of threshold range

Note 1: After setting IN_SEL1 and IN_SEL0, the HZIF0 bit is set to 1, and HZOF0 and HZOF1 are not set.

Note 2: Valid only when INTP0 input and comparator 0 output are selected.

10.8.3 Running of PWMOP

You can select the output of comparator 0, the INTP0 input, and the event input of EVENTC as the source of the output Forced cutoff.

When you select the output of comparator 0 and the INTP0 input as the source, you can choose to follow the checkout.

10.8.3.1 output forced cut-off

You can select the output of comparator 0, the INTP0 input, and the event input of EVENTC as the trigger event to force the output of TMIOji (j=A,B,C,D; i=0,1) to be stopped.

When the trigger event is detected, the output of the timer M is forced off and the set value of the register OPDF0/OPDF1 is output. See Figure 10-75 for details.

The software or hardware is selected to remove the forced cutoff function via HS_SELbit of the OPCTL0 register.

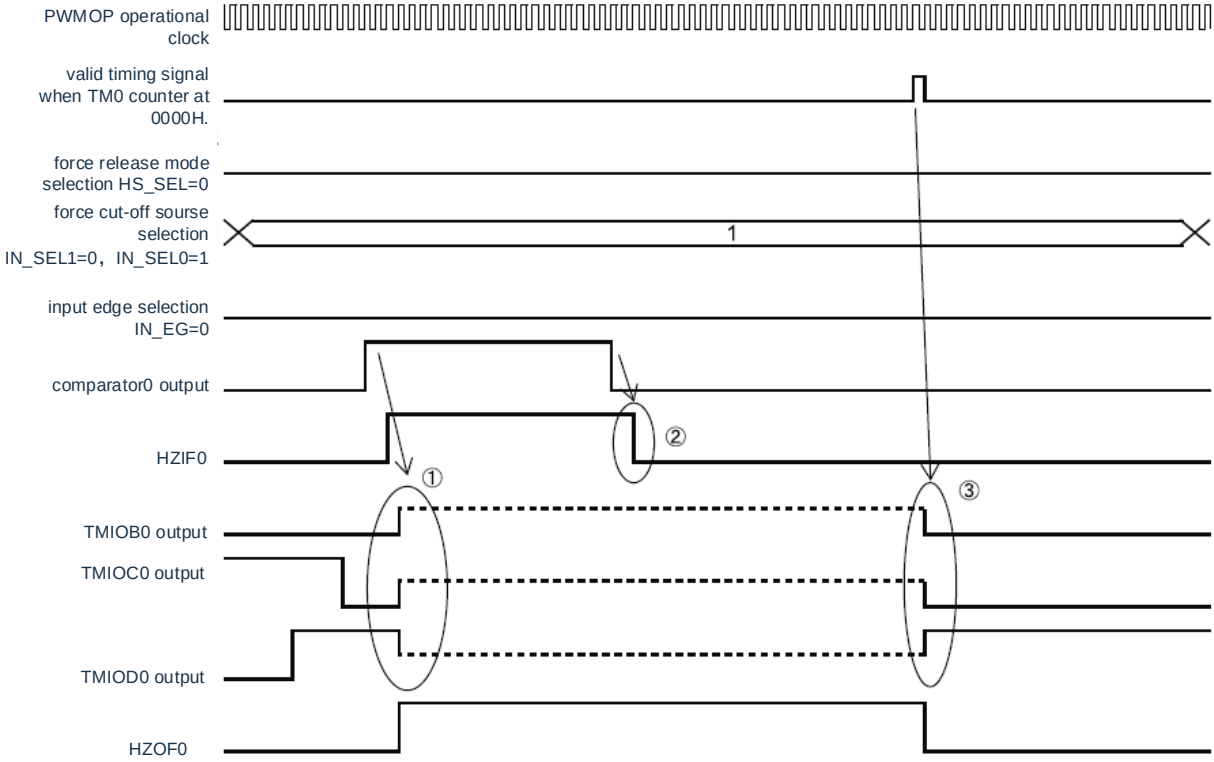
10.8.3.2 Hardware Undoing (HS_SEL=0)

When the timer M works in different modes, the timing of the release is different:

1) The Output Occasions Other Than Complementary PWM Function

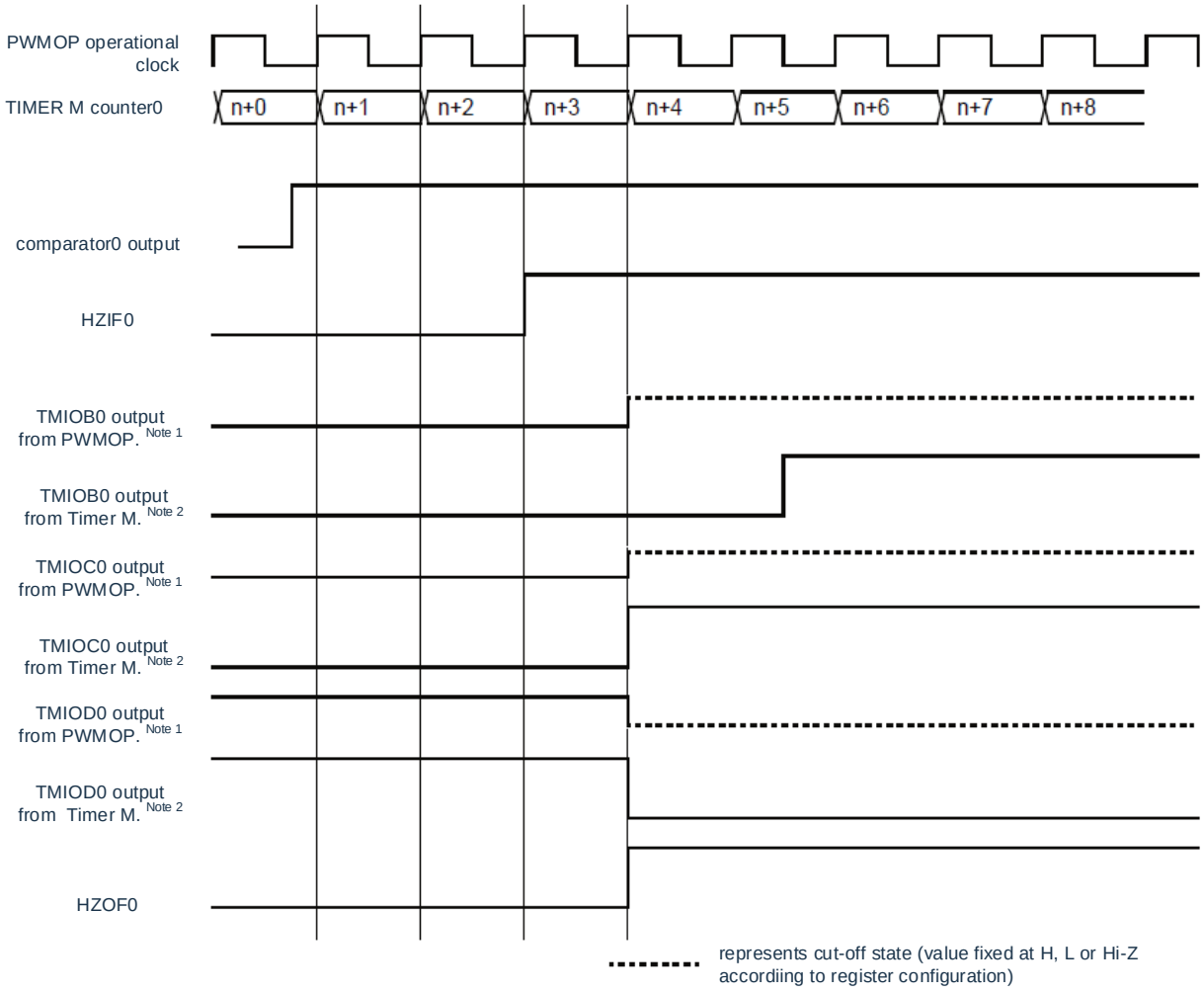
- When the timer M works in the output comparison function, PWM function or PWM3 mode, the forced cut-off of TMIOj0 (j=A, B, C, D) is removed when the count value of TM0 is 0000H. When the count of TM0 is 0000H, the forced cutoff of TMIOj1 (j=A, B, C, D) is removed.
- When the timer M is reset in the same period PWM mode, when the count value of TM0 is 0000H, all the TMIO pins are stopped.

Figure 10-74 Example of Output Forced Cutoff / Hardware Undoing Output Forced Cutoff
(TMIOB0, TMIOC0, example of TMIOD0 pin being forced off)



1) When the rising edge of the comparator output signal is detected, the output of the TMIOB0, TMIOC0, and TMIOD0 pins are in the forced cut off state.
2) When the falling edge of the comparator output signal is detected, the HZIF0 bit is cleared.
3) At a TDi count of 0,000 H, the forced cutoff state is released.

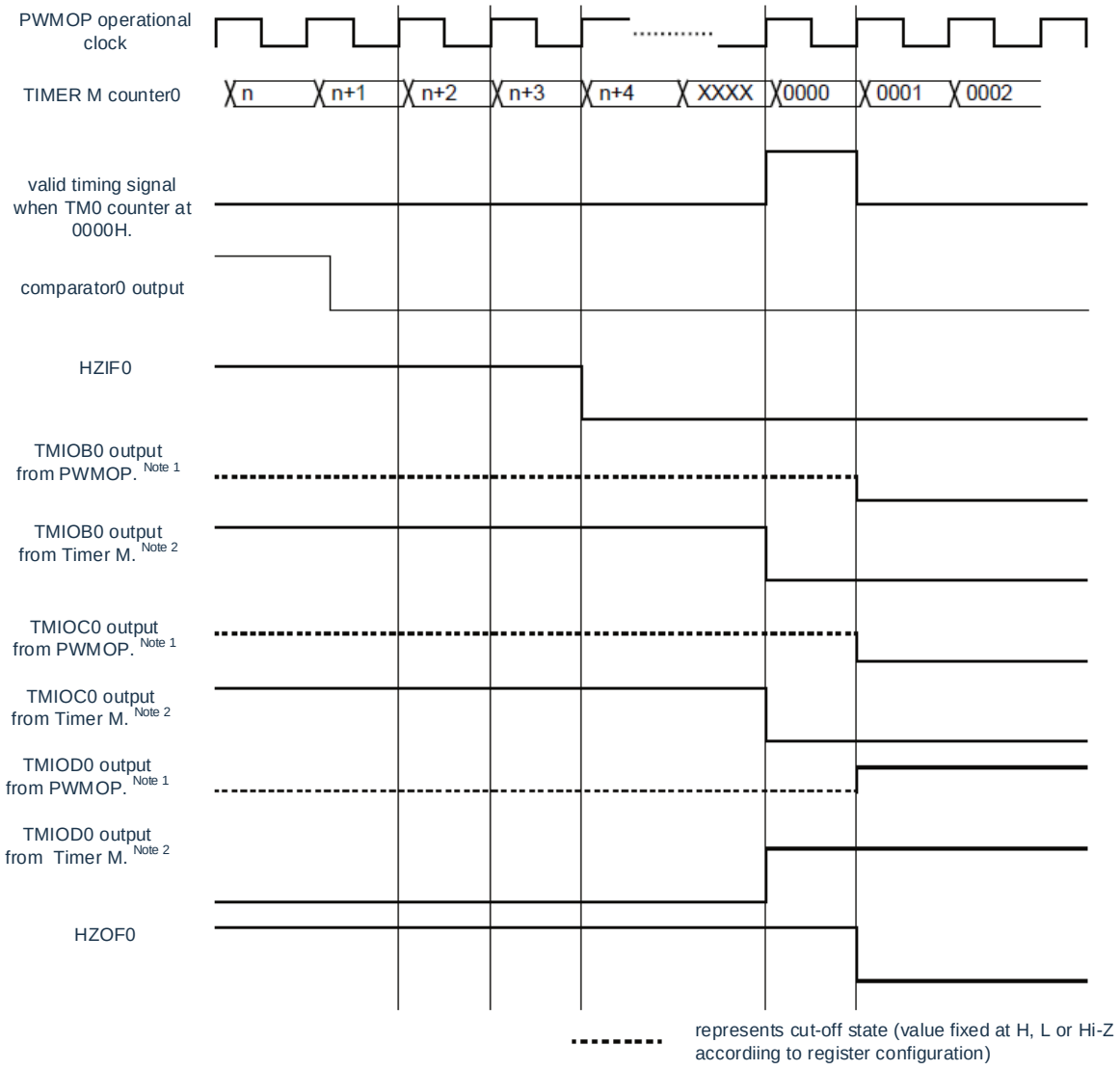
Figure 10-75 Detailed Time Series for Forced Cutoff



Note 1: The TMIO* (*=B~D) output from the PWMOP represents the state of the pins of the M function of the multiplexed timer.

Note 2: The TMIO* (*=B~D) output from Timer M indicates the input signal state output from timer M to PWMOP.

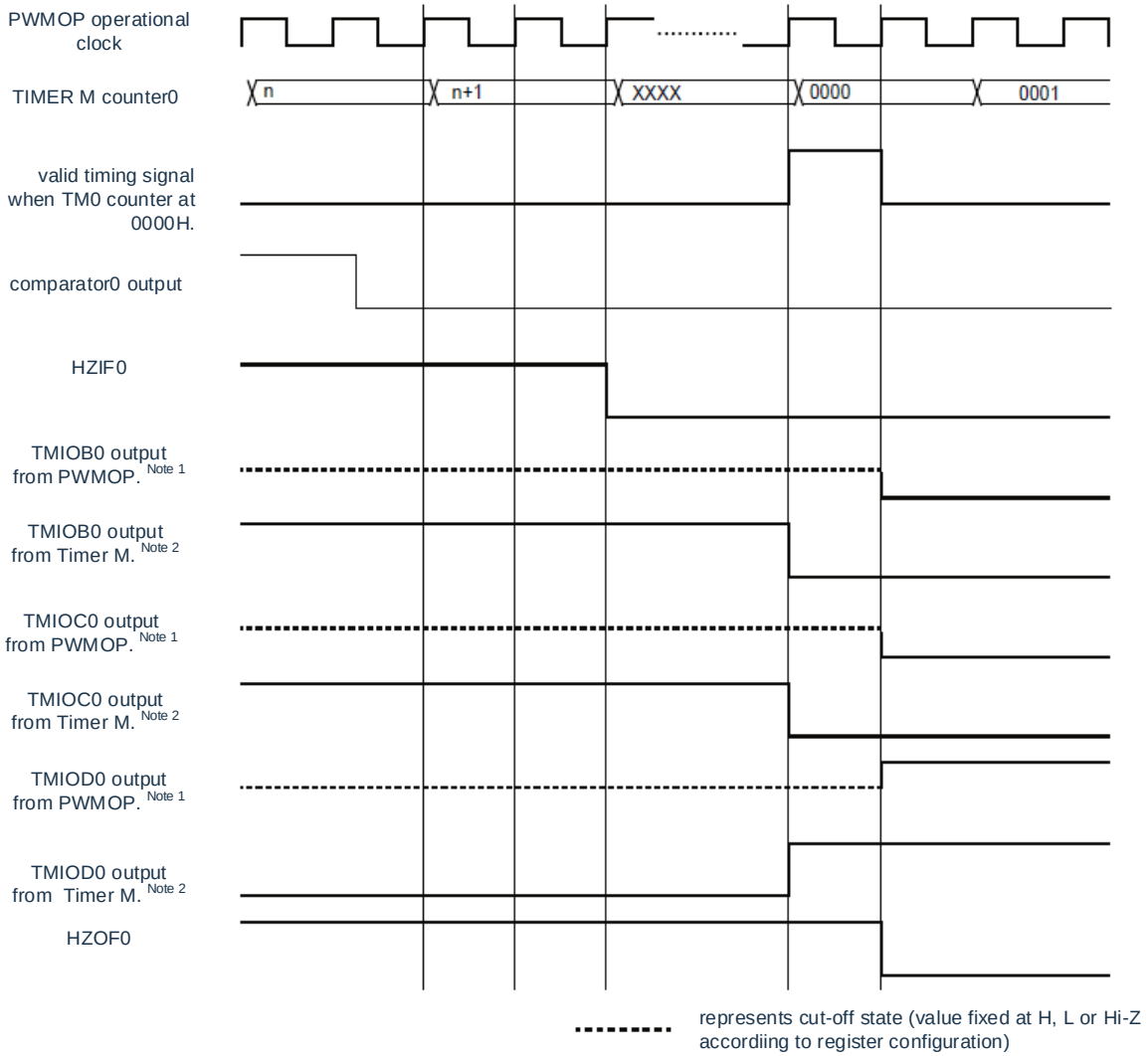
Figure 10-76 Detailed Time Series of Forced Cutoff Release (counter source for timer M=Fclk)



Note 1: The TMIO* (*=B~D) output from the PWMOP represents the state of the pins of the M function of the multiplexed timer.

Note 2: The TMIO* (*=B~D) output from Timer M indicates the input signal state output from timer M to PWMOP.

Figure 10-77 Detailed Time Series of Forced Cutoff Release (counter source for timer M=Fclk/2)



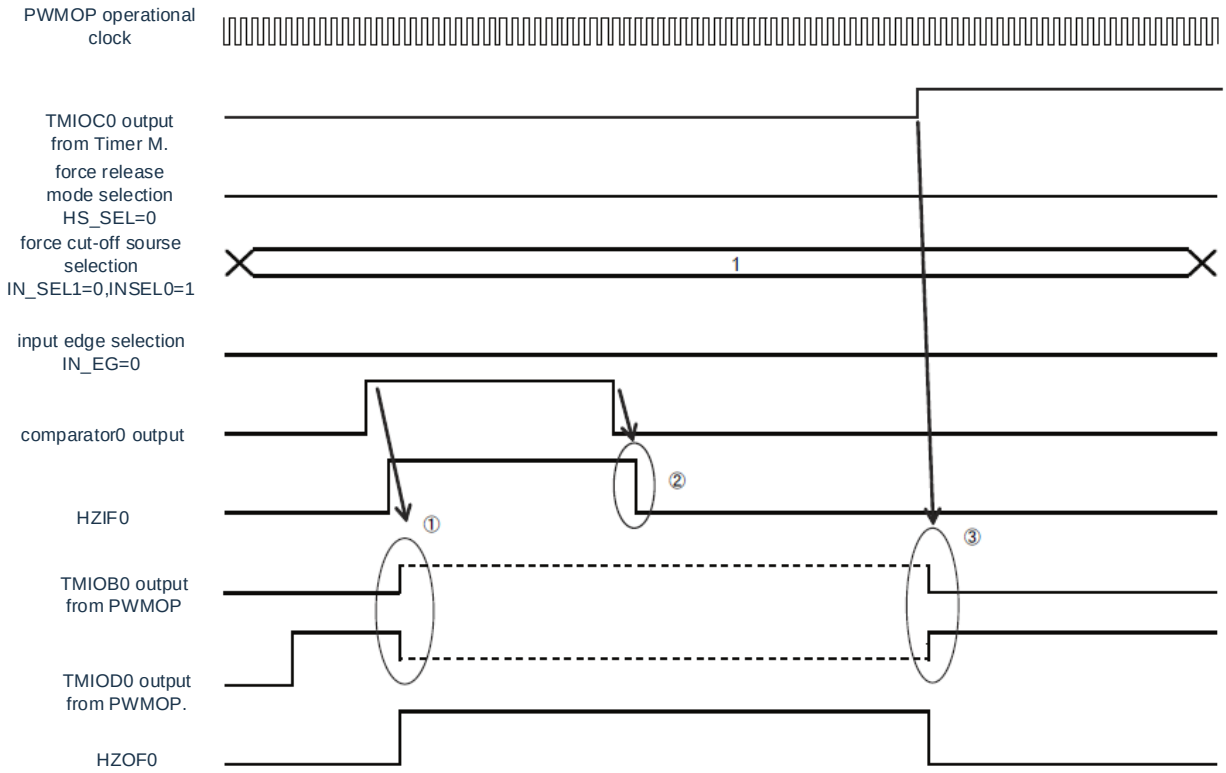
Note 1: The TMIO* (*=B~D) output from the PWMOP represents the state of the pins of the M function of the multiplexed timer.

Note 2: The TMIO* (*=B~D) output from Timer M indicates the input signal state output from timer M to PWMOP.

1) The Situation of Complementary PWM Function Output

Upon detection of the release source, the edge of the TMIOC0 selected according to the OPEDGE is forced off.

Figure 10-78 Example of Hardware Unforced Cutoff (Using TMIOB0,TMIOD0 as an example)



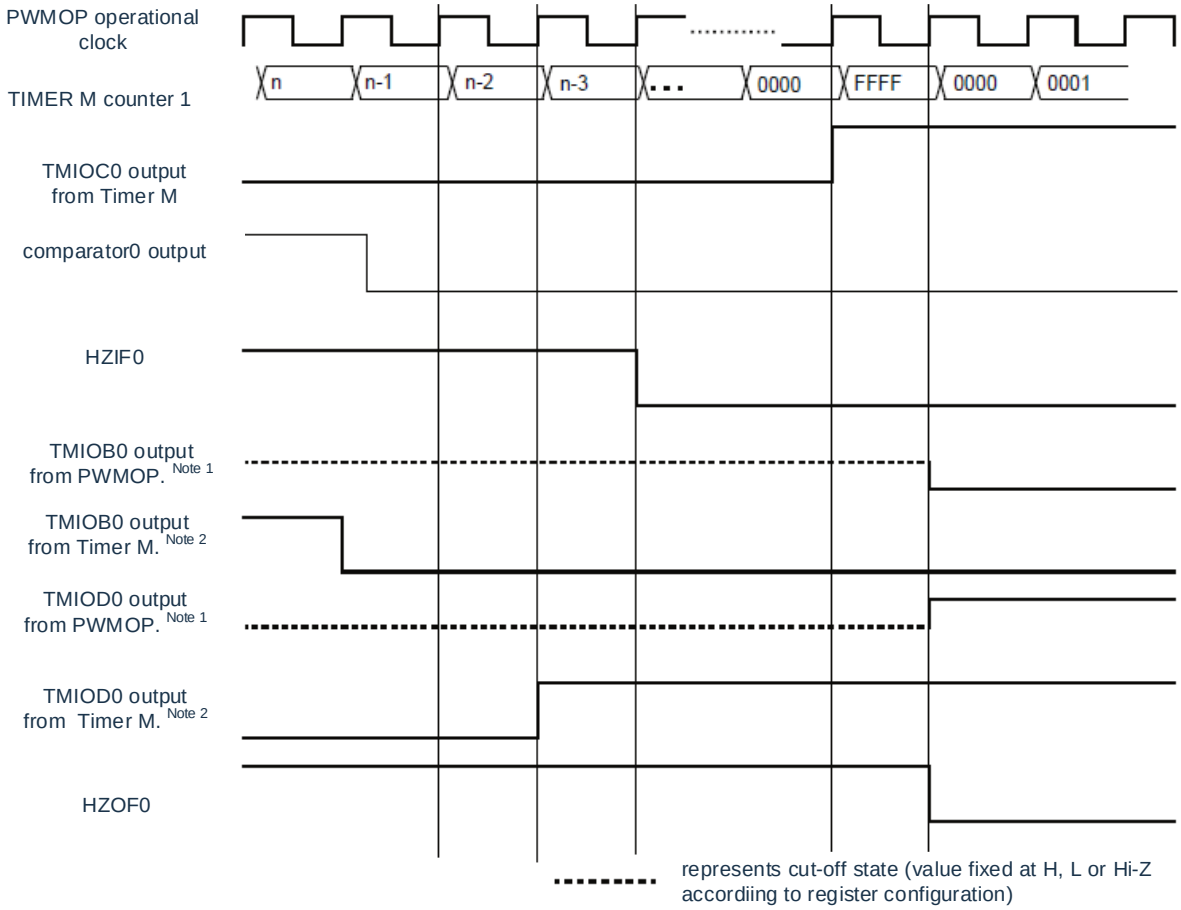
a) When the rising edge of the comparator 0 output signal is detected, the output of the TMIOB0, TMIOD0 pin is in a forced cut-off state.

b) The HZIF0 bit is cleared when a falling edge of the comparator 0 output signal is detected.

c) The forced cutoff state is released when the rising edge of the TMIOC0 occurs.

Please refer to Figure 10-75 for the detailed timing of the Forced cutoff.

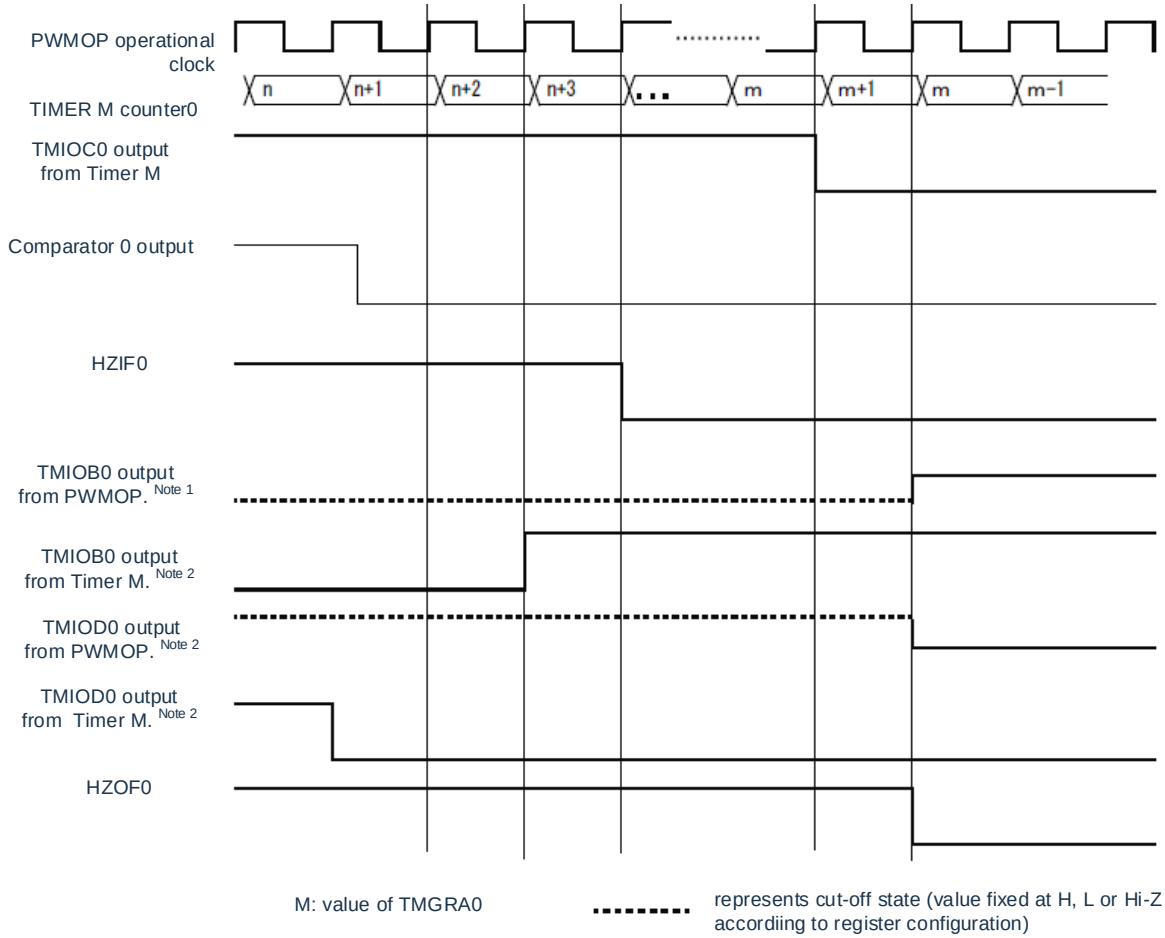
FIG. 10-79 Detailed timing diagram of forced cutoff release (counter source for timer M = Fclk, decremental count)



Note 1: The TMIO* (*=B~D) output from the PWMOP represents the state of the pins of the M function of the multiplexed timer.

Note 2: The TMIO* (*=B~D) output from Timer M indicates the input signal state output from timer M to PWMOP.

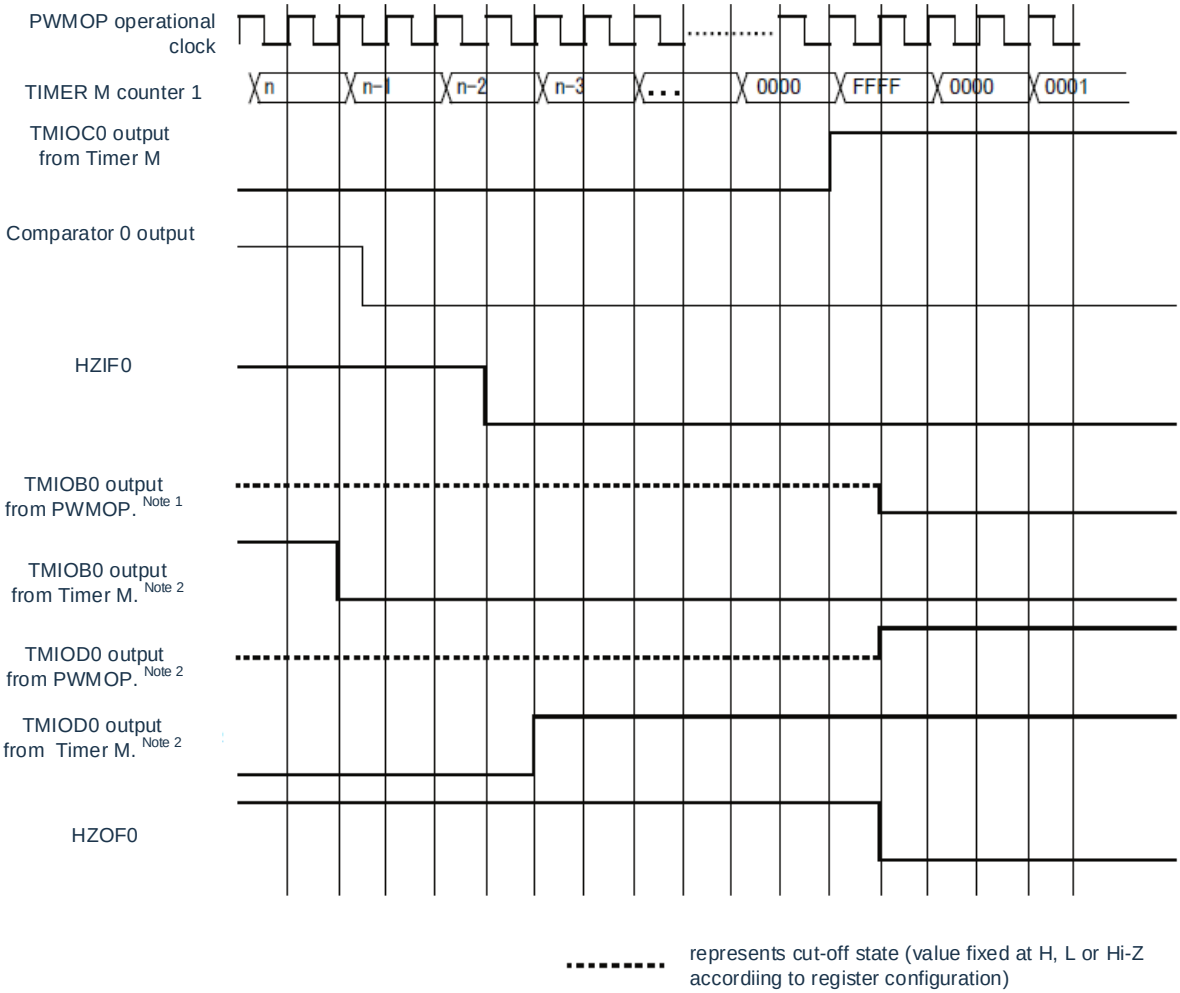
Figure 10-80 Detailed time series of forced cutoff release (counter source=Fclk, counter=TMGRA0)



Note 1: The TMIO* (*=B~D) output from the PWMOP represents the state of the pins of the M function of the multiplexed timer.

Note 2: The TMIO* (*=B~D) output from Timer M indicates the input signal state output from timer M to PWMOP.

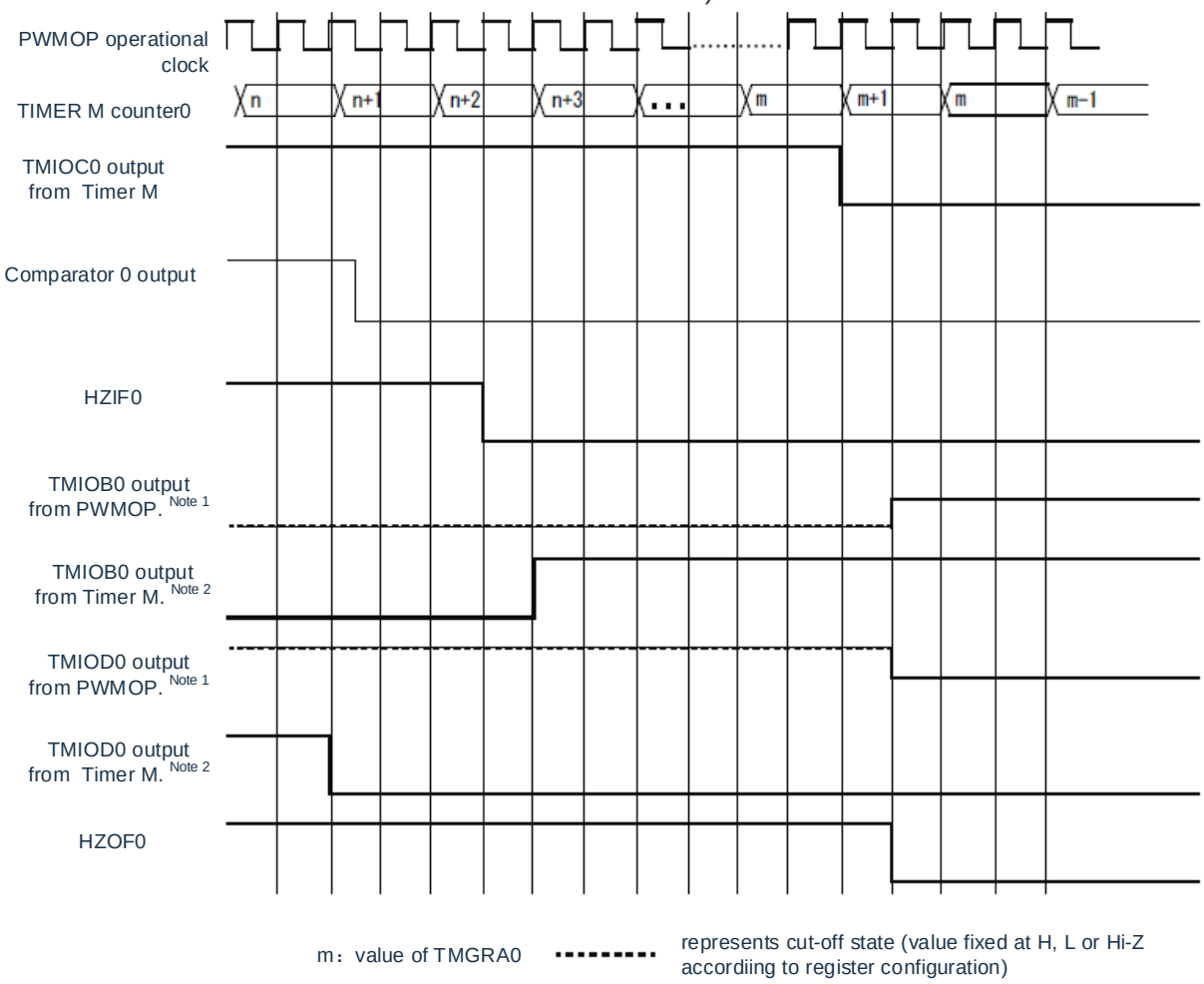
FIG. 10-81 Detailed timing diagram of forced cutoff release (counter source of timer M = Fclk/2, decremental count)



Note 1: The TMIO* (*=B~D) output from the PWMOP represents the state of the pins of the M function of the multiplexed timer.

Note 2: The TMIO* (*=B~D) output from Timer M indicates the input signal state output from timer M to PWMOP.

Figure 10-82 Detailed Time Series for Forced Cutoff Release (counter source=Fclk/2 for timer M, counter=TMGRA0)



Note 1: The TMIO* (*=B~D) output from the PWMOP represents the state of the pins of the M function of the multiplexed timer.

Note 2: The TMIO* (*=B~D) output from Timer M indicates the input signal state output from timer M to PWMOP.

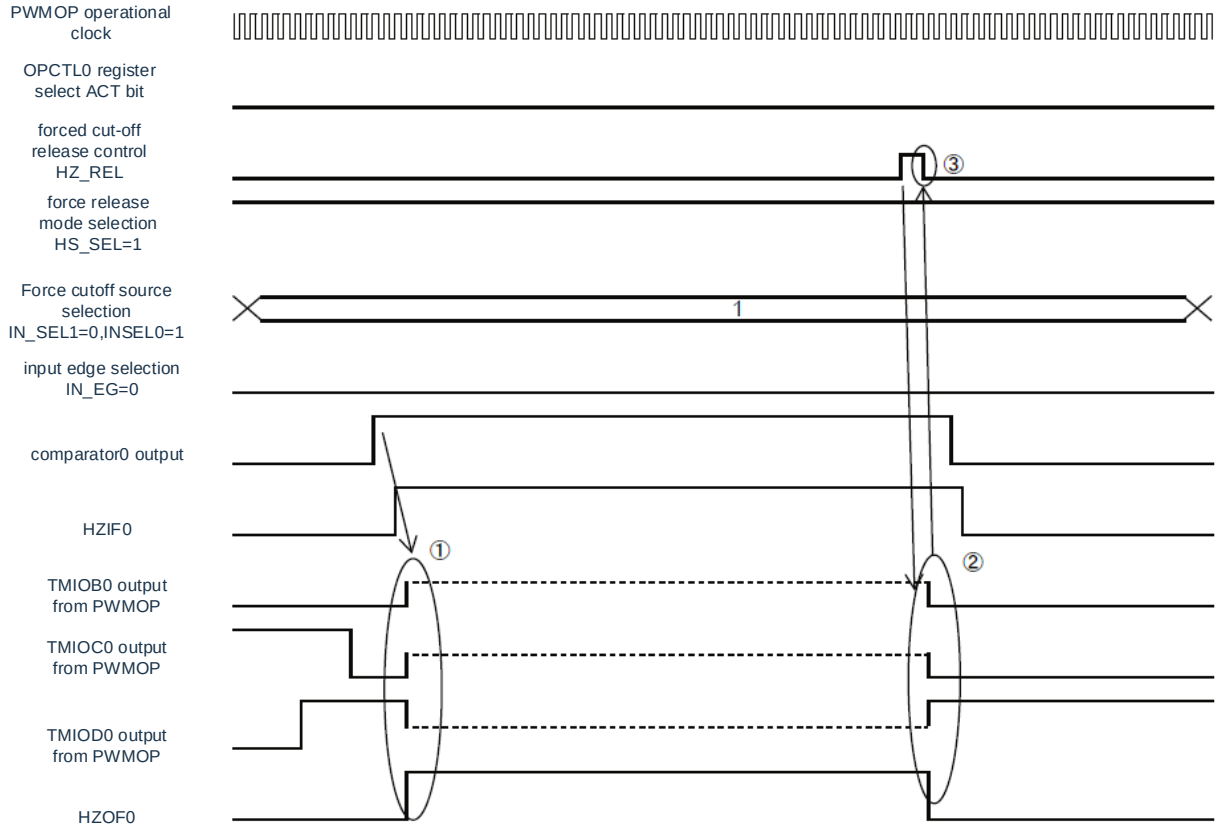
10.8.3.3 Software Undoing (HS_SEL=1)

The force cutoff release timing is different when the ACT settings of OPCTL0 are different.

1) Use software to immediately undock (ACT=0)

If ACT=0 is set, once the HZ_REL bit of the OPCTL0 register is set to 1, the force cutoff is immediately lifted. When the HZ_REL bit is released, it is automatically placed 0.

Figure 10-83 Examples of software de-Forced cutoff (in the example of TMIOB0, TMIOC0, TMIOD0)



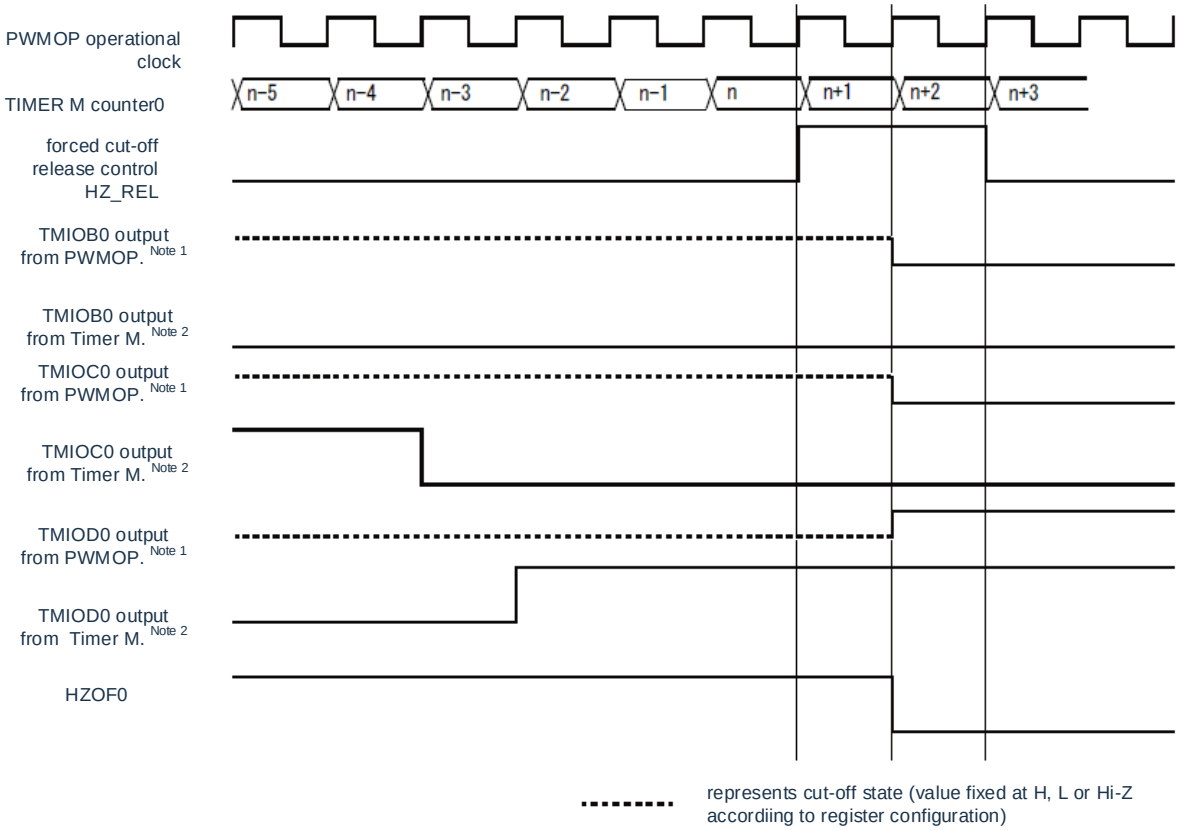
a) When the rising edge of the output signal of the comparator 0 is detected, the output of the TMIOB0, TMIOC0, TMIOD0 pins is in a forced cut-off state.

b) Set HZ_SEL to 1 to force the cutoff immediately to be lifted.

c) When the force cutoff is lifted, the HZ_REL bit becomes 0.

Please refer to Figure 10-75 for the detailed timing of the Forced cutoff.

Figure 10-84 Detailed Time Series for Forced Cutoff Release



2) Software Unrelease (ACT=1)

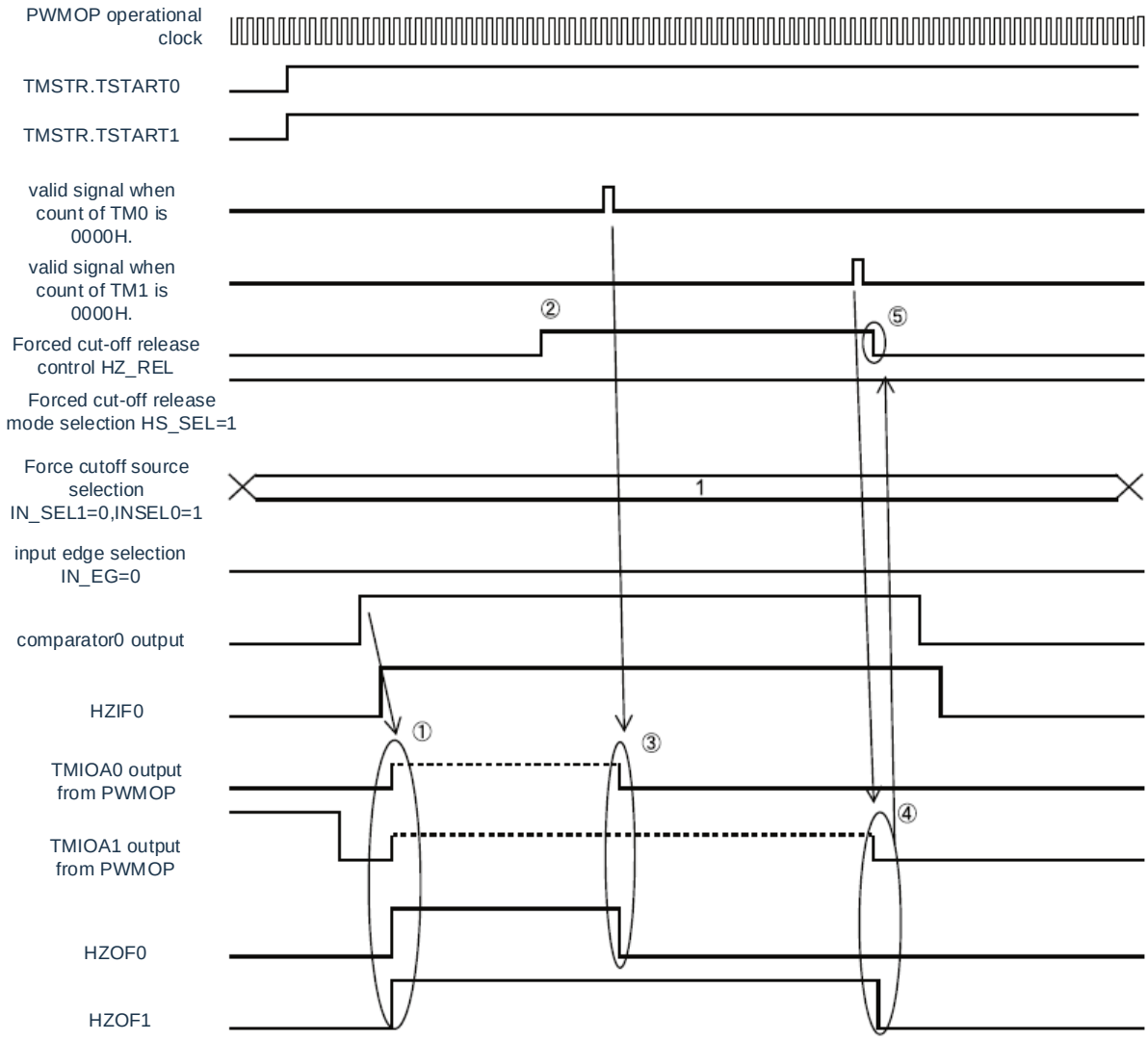
If the ACT is set to 1, the forced cutoff state can be removed by the signal of timer M after setting the HZ_REL of OPCTL0 to 1. HZ_REL is automatically set to 0 after release.

When the hardware deactivation is stopped, after the deactivation event is detected, the deactivation is triggered and the output is recovered by the signal of the timer M. When the software is released, setting HZ_REL to 1 triggers the release by the signal of the timer M and restores the output. The timing of the release is the same.

a) The timer M works in the output comparison function, PWM function, PWM3 mode occasions:

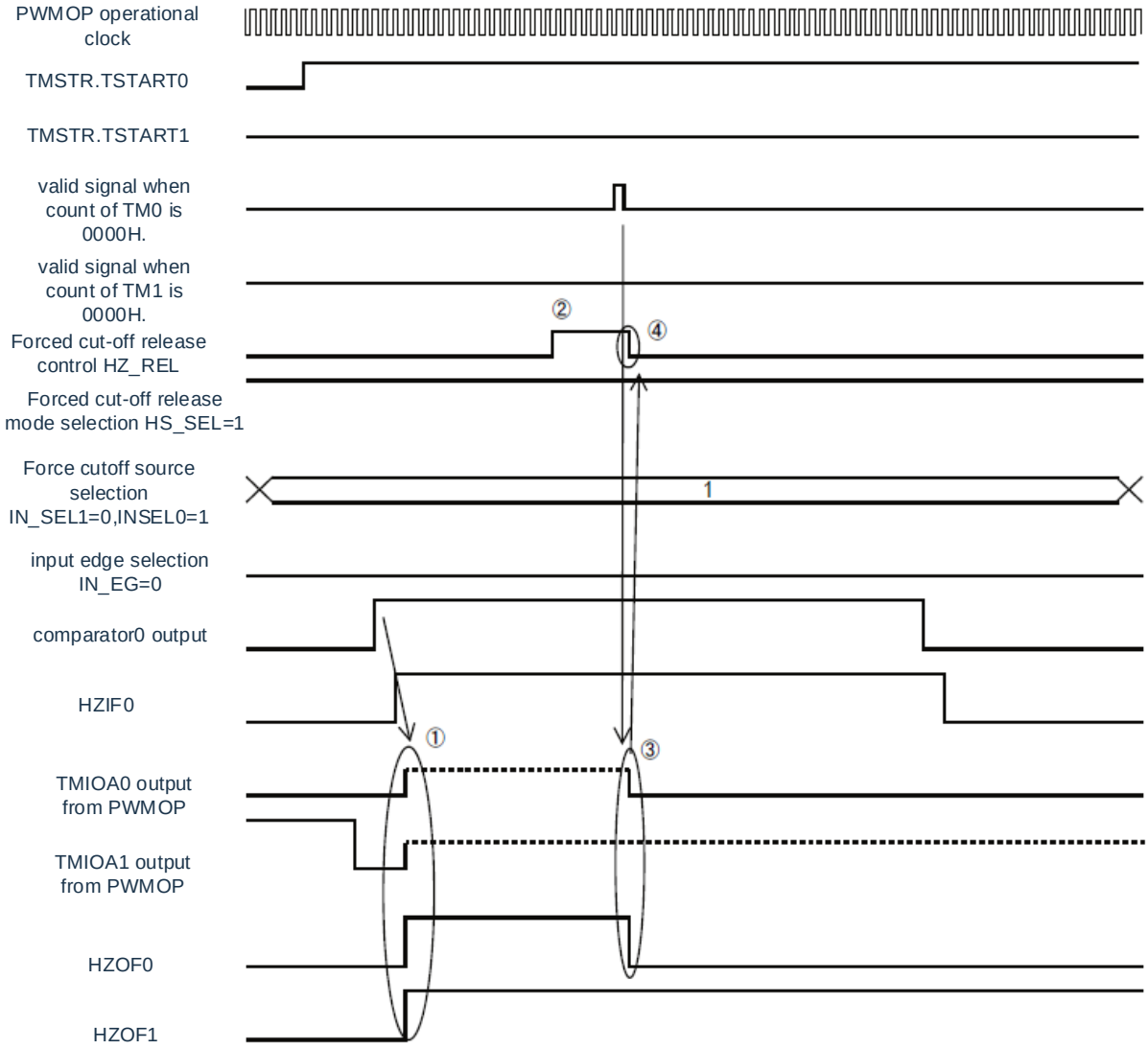
When TM0 is changed to 0000H after HZ_REL is set to 1, the output forced cut-off states of TMIOA0, TMIOB0, TMIOC0 and TMIOD0 are released. When the count of TM1 becomes 0000H, the output force cutoff state of TMIOA1, TMIOB1, TMIOC1,TMIOD1 is released.

Figure 10-85 Example of Software Unforced Cutoff (Timer M, 2 Channel Count)



1) When the rising edge of the comparator 0 output signal is detected, the output of the TMIOA0, TMIOA1 pin is in a forced cut-off state.
2) Set HZ_REL to 1 and wait for each count count value to 0000H.
3) The forced cutoff of TMIOA0 is lifted when the TM0 count reaches 0 000 H.
4) The forced cutoff of TMIOA1 is lifted when the TM1 count reaches 0000 H.
5) When the force cutoff is lifted, the HZ_REL bit becomes 0.

Figure 10-86 Example of Software Unforced Cutoff (Timer M, 1 Channel Count)



1) When the rising edge of the comparator 0 output signal is detected, the output of the TMIOA0, TMIOA1 pin is in a forced cut-off state.
2) Set HZ_REL to 1 and wait for each count count value to 0000H.
3) The forced cutoff of TMIOA0 is lifted when the TM0 count reaches 0 000 H.
4) When the force cutoff is lifted, the HZ_REL bit becomes 0.

Please refer to Figure 10-75 for a detailed timeline of the Forced cutoff.

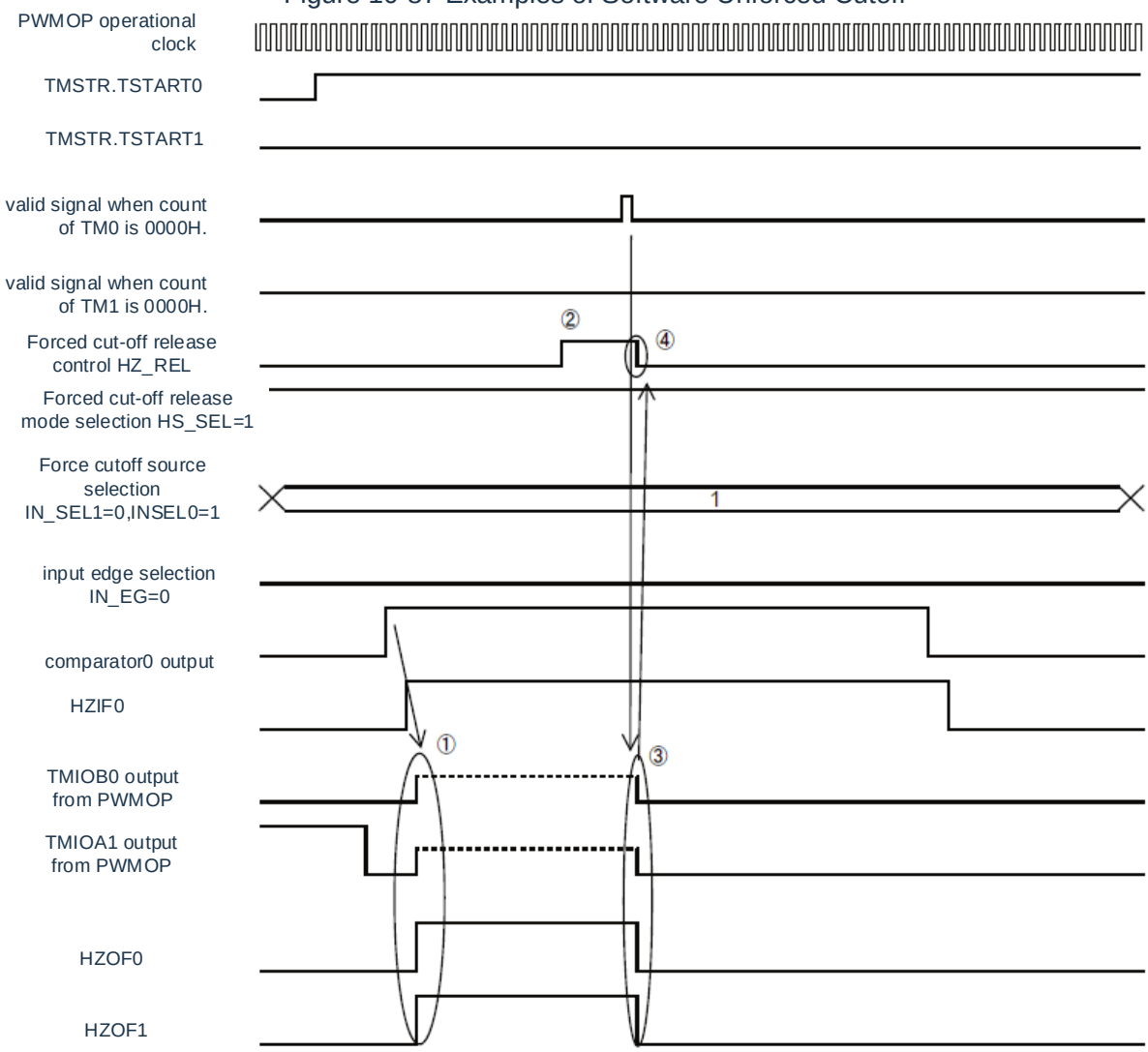
Please refer to Figure 10-76, Figure 10-77 for the detailed timing of force cutoff release.

For the HZ_REL auto-zero sequence, refer to Figure 10-84.

1) The timer M works in the case of resetting the synchronous PWM mode

Set HZ_REL to 1, and at 0,000H for TM0 count, all TMIO pins are forced off.

Figure 10-87 Examples of Software Unforced Cutoff



a) When the rising edge of the comparator 0 output signal is detected, the output of the TMIOA0, TMIOA1 pin is in a forced cut-off state.

b) When HZ_REL is set to 1, the channel 0count count value of the waiting timer M becomes 0000H.

c) When TM0 reaches 0000H, TMIOB0 and TMIOA1 are forced off (Channel 1 action is not affected).

d) When the force cutoff is lifted, the HZ_REL bit becomes 0.

Please refer to Figure 10-75 for a detailed timeline of the Forced cutoff.

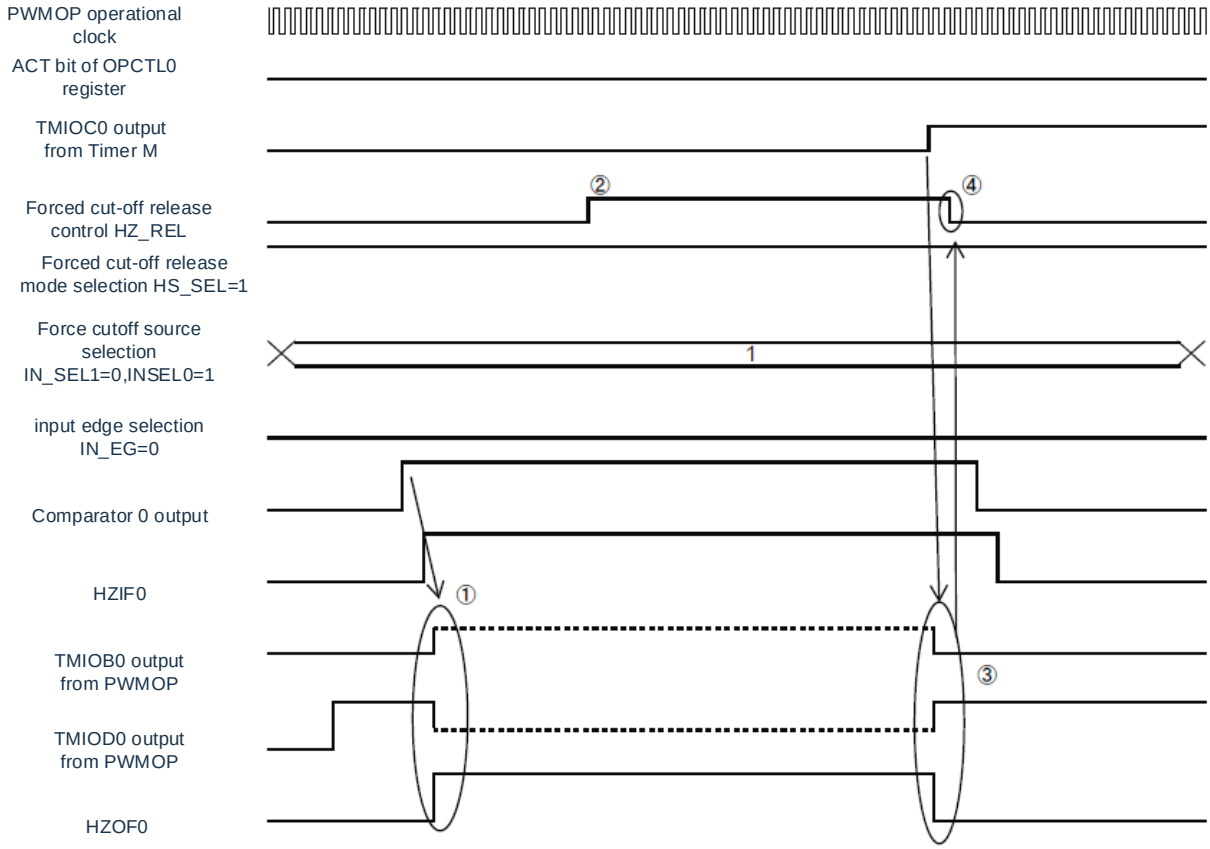
Please refer to Figure 10-76, Figure 10-77 for the detailed timing of force cutoff release.

For the HZ_REL auto-zero sequence, refer to Figure 10-84.

2) The timer works in complementary PWM mode

When HZ_REL is set to 1, the edge of the TMIOC0 selected according to the OPEDGE is unforced.

Figure 10-88 Examples of Software De-Forced Cutoff (TMIOB0, TMIOD0)



a) When the rising edge of the comparator 0 output signal is detected, the output of the TMIOB0, TMIOD0 pin is in a forced cut-off state.

b) Set HZ_REL to 1, wait for the TMIOC0 edge.

c) The forced cutoff state is released after the rising edge of TMIOC0 is checked out.

d) When the force cutoff is lifted, the HZ_REL bit becomes 0.

Please refer to Figure 10-75 for a detailed timeline of the Forced cutoff.

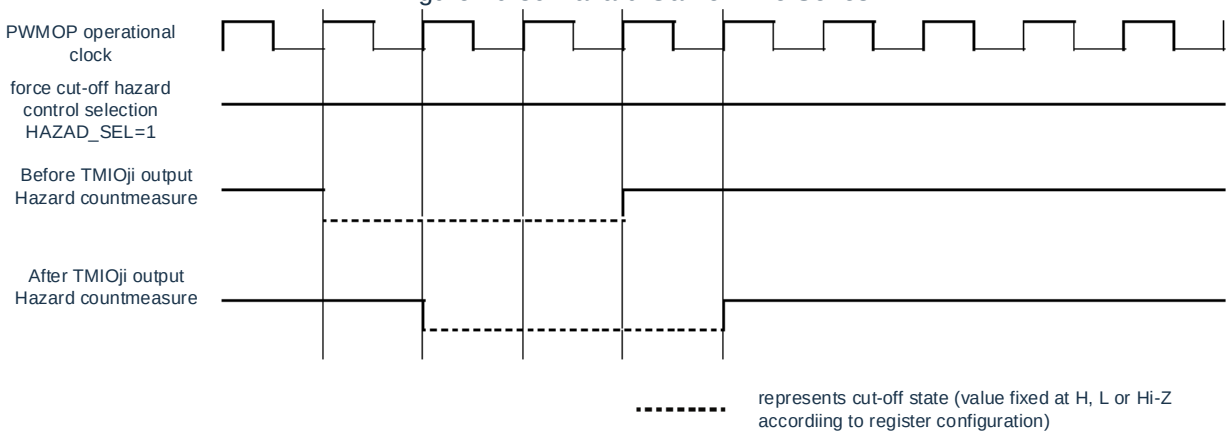
Please refer to Figures 10-79, 10-80, 10-81, and 10-82 for the detailed timing of force cutoff release.

For the HZ_REL auto-zero sequence, refer to Figure 10-84.

10.8.3.4 Hazard countermeasure

If in that act of force cut-off state/force cut-off release state/timer M, the TMIO pin is at risk of mis-acting while switching between the multiplexing function and the PORT function. You can set HAZAD_SET to 1 to allow hazard countermeasures to avoid this risk. Note that when using the Hazard countermeasure, the output of timer M is delayed by one clock cycle compared to when prohibiting the Hazard countermeasure.

Figure 10-89 Hazard Game Time Series



Note: j=A,B,C,D； i=0,1

10.8.3.5 Output Force Cutoff of Source Checked Out and Unchecked Out

The output force cutoff source checks out the level of the selected signal (INTP0, CMP0) by the cutoff source selection bit (OPCTL0.IN_SEL1, OPCTL0.IN_SEL0).

If OPCTL0.IN_EG is set to 0, the high level is the source detection state and the low level is the undetected state.

If OPCTL0.IN_EG is set to 1, the low level is the source detection state, and the high level is the undetected state.

Note: After setting IN_SEL1 and IN_SEL0 to select INTP0 input, before comparator 0 output as force cutoff element, if output force cutoff source exceeds threshold range, after IN_SEL1 and IN_SEL0 configured, the HZIF0 bit is set to 1, and HZOF0 and HZOF1 are not set.

10.8.3.6 The time sequence diagram when the value of the counter of timer M reaches 0000 H.

When the hardware cancels the mandatory cut-off of the output, the mandatory cut-off conditions are different according to the operation mode of the timer M.

1) The timer M operates in a timing sequence in which the count value reaches 0000H when the output comparison function
   - Count value=0000H, timer M count start: Cannot unforce deadline
   - The timer M is in the count, the counter is assigned a value of 0000H by software, and the forced cutoff state is removed
   - The timer overflows, the count value becomes 0000H, and the forced cut-off state is removed
   - Consistent with the value of TMGRA0, the count value becomes 0000H and the force cutoff state is lifted

2) When the timer M is working in the PWM function, the timing of the count value to 0,000 H
   - Count value=0000H, timer M count start: Cannot unforce deadline
   - The timer M is in the count, the counter is assigned a value of 0000H by software, and the forced cutoff state is removed
   - Consistent with the value of TMGRA0, the count value becomes 0000H and the force cutoff state is lifted

3) The timer M operates in a timing sequence in which the count value reaches 0000 H when the reset synchronous PWM mode is applied
   - Count value=0000H, timer M count start: Cannot unforce deadline
   - The timer M is in the count, the counter is assigned a value of 0000H by software, and the forced cutoff state is removed
   - Consistent with the value of TMGRA0, the count value becomes 0000H and the force cutoff state is lifted

4) The timing of the count value to 0000 H when the timer M is operating in PWM3 mode
   - Count value=0000H, timer M count start: Cannot unforce deadline
   - The timer M is in the count, the counter is assigned a value of 0000H by software, and the forced cutoff state is removed

- Consistent with the value of TMGRA0, the count value becomes 0000H and the force cutoff state is lifted

5) The timer M reaches a count value of 0000H, and the timer M stops working

If the timer M stops working while the count value reaches 0,000 H, the forced cut-off cannot be removed.

Figure 10-90 Timing sequence at count value = 0000 H

(Count source=Fclk/2, counter stops when timer M count reaches 000 H)

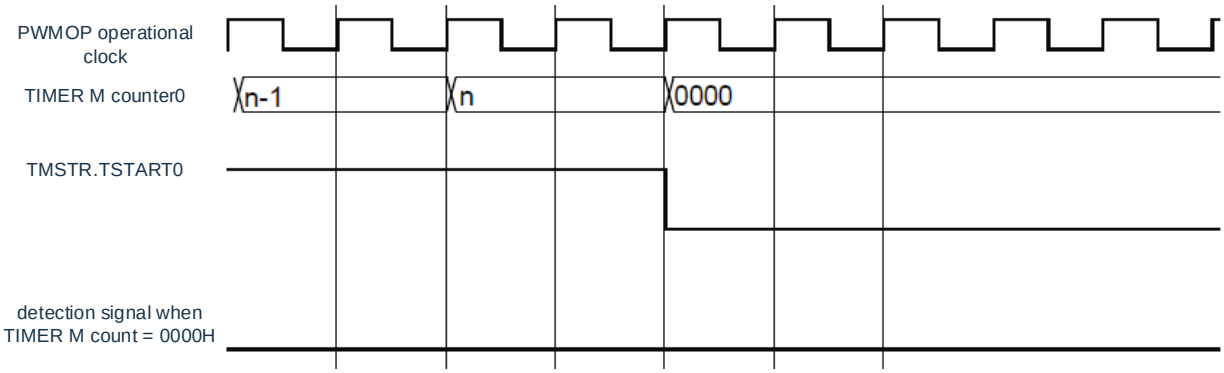
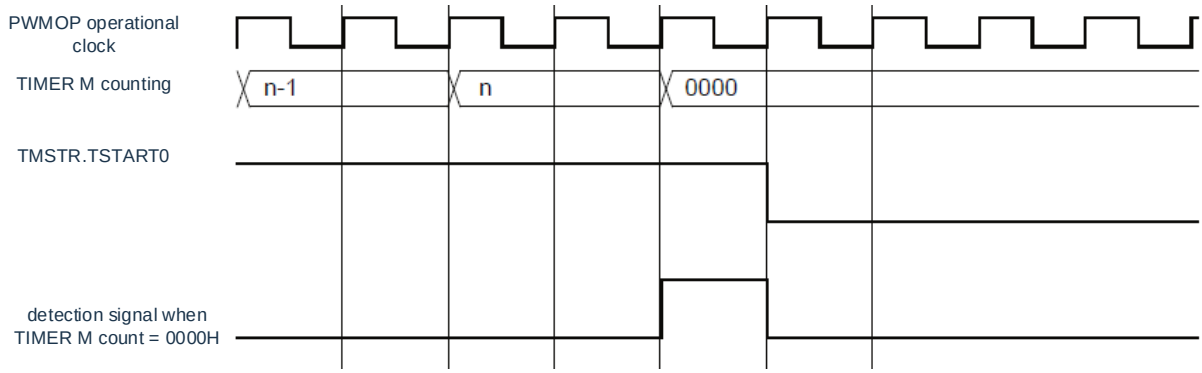


Figure. 10-91 Timing sequence at count value = 0000 H

(Count source=Fclk/2, timer M count value to 0000H next cycle counter stops)



10.8.3.7 Configuration Steps

The PWMOP may be associate with that timer M, and the setting of the PWMOP may be appended to the setting of the timer M. The steps are as follows:

After Timer M's clock and mode are set

1) Set PWMOPEN to 1
2) Set up OPCTL0 register
3) Set OPEDGE register
4) Set OPDF0, OPDF1 register, timer M action begins
5) Wait for the forced cutoff state shown by HZOF1, HZOF0
6) force termination

Note: The PWMOP implements the function of using comparator 0 output, external interrupt input INTP0 and Coordination Controllerto force cut off timer M output. Therefore, the PWMOP must be used when the timer M is active. If only timer M is used, the relevant registers of PWMOP need to be kept in reset state.

10.8.4 Precautions

1) When the output of the timer M and the output of the PWMOP are working simultaneously, the priority is as follows:

Table 10-22 Prioritization at Forced Cutoff

		Pin status when PWMOP forces a cutoff			
		prohibition	Hi-Z	L-level	H level
Pin status when timer M forces a cutoff	prohibition	prohibition	Hi-Z	L-level	H level
	Hi-Z	Hi-Z	Hi-Z	L-level	H level
	L-level	L-level	Hi-Z	L-level	H level
	H level	H level	Hi-Z	L-level	H level

2) In that complement PWM mode, if the PWMOP is in the output force cutoff state, the timer m also enter the pulse output force cutoff.
3) The state, which may produce an uncutoff edge for PWMOP.
4) When you use EVENTC as the cutoff source, you must use software to lift the cutoff state (HS_SEL=1)
5) When an output cutoff hazard countermeasure is used, the output via timer M of PWMOP delays a clock cycle.
6) When the output cut-off hazard countermeasure is used, the timer count operation can be switched via the output pin of the timer M of the PWMOP.
7) multiplexing and PORT functions.
8) The effective width of the select comparator 0 output and the INTP0 input must be greater than one clock cycle.

Chapter 11 real-time clock

11.1 The Function 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 months)
- Alarm clock interrupt (alarm clock: week, hour, minute)
- 1Hz Pin Out Capability

11.2 Structure of real-time clock

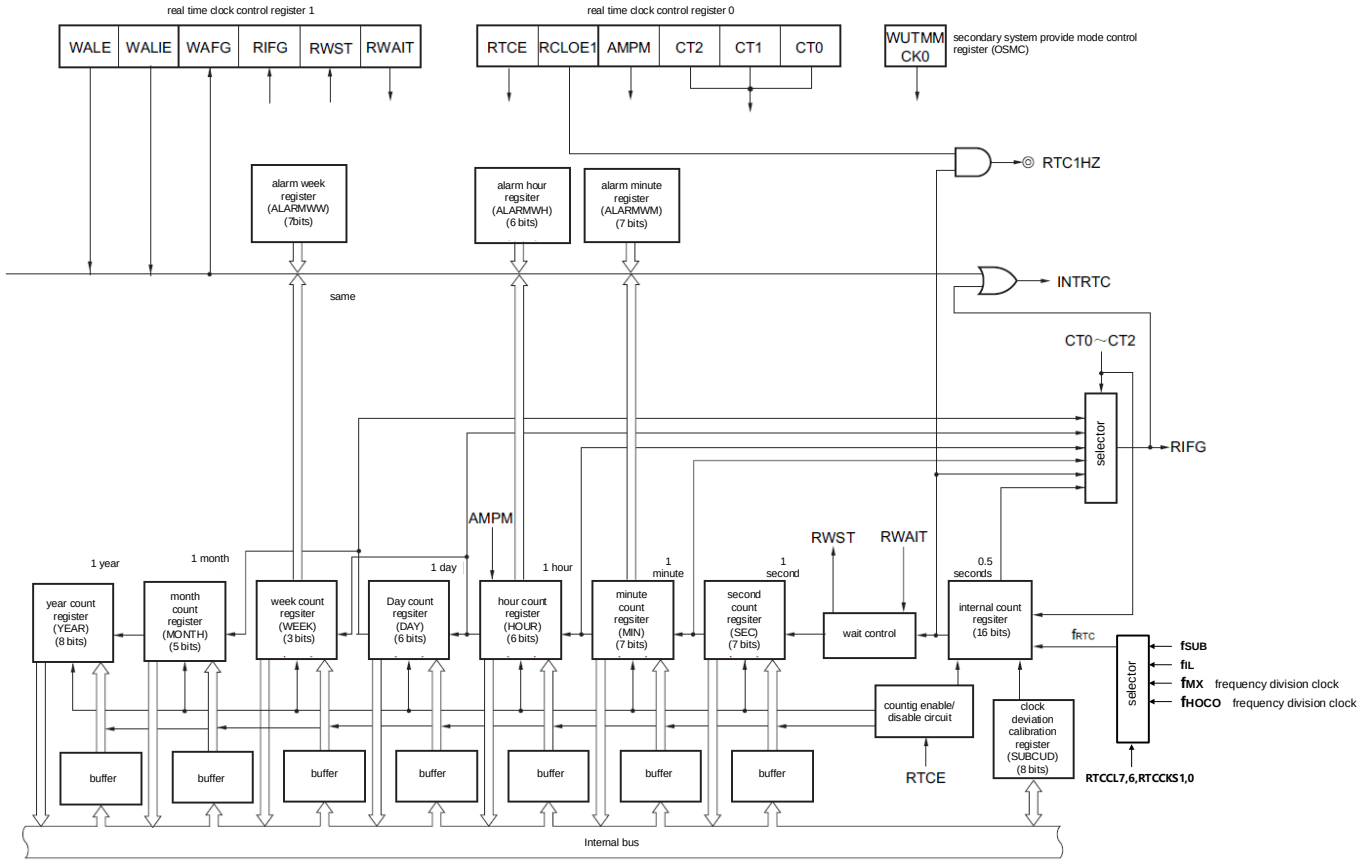
The real-time clock consists of the following hardware.

Table11-1 Structure of real-time clock

Project	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 register is only controlled by the POR reset.

Figure 11-1 Block diagram of real-time clock



Note: Count years, months, weeks, days, hours, minutes and seconds only if you select fmx/fhoco clock (≈32, 768KHZ after every week) or the secondary system clock (fSUB=32.768kHz) as the running clock for the real-time clock. When a low-speed internal oscillator clock (fIL=15kHz) is selected, only a fixed cycle interrupt function is used. The fixed cycle interrupt interval when selecting fIL is calculated using the following equation: Fixed period (value selected by the RTCC0 register) fSUB/fIL

11.3 Register 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 (PMx)
- Port register (Px)

11.3.1 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.

You must set bit7 (RTCEN) to '1' when you want to use real-time clocks. The PER0 register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 11-2 Format of Peripheral Admission Register 0 (PER0)

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

symbol	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDEN	ADCEN	IICA0EN	SCI1EN	SCI0EN	CAN0EN	TM40EN

RTCEN	Control of an input clock of a real-time clock (RTC) and a 15-bit interval timer
0	Stop provide an input clock. · You cannot write the SFR used by the real-time clock (RTC) and 15-bit interval timers. · The real-time clock (RTC) and the 15-bit interval timer are reset.
1	Provides an input clock. · SFRs that can read and write real-time clocks (RTCs) and 15-bit interval timers.

Note:1. If you want to use a real-time clock, you must first set the RTCEN position "1" in the fRTC oscillating steady state and then set the following register. 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)

2. The sub-system clock can be stopped in deep sleep mode or sleep mode running in the sub-system clock by providing the RTCLPC position of the OSMC.

11.3.2 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 11-3 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	RTCCS1	RTCCS0

RTCCL7	Selection of Clock Source for Real-time Clock and Counter Clock of 15-bit Interval Timer
0	Select a high speed system clock (f_{MX})
1	Select a high speed internal oscillator (fhoco)

RTCCS1	RTCCS0	RTCCL6	Selection of Running Clock of Real-time Clock and Counter Clock of 15-bit Interval Timer
0	0	0/1	Secondary System Clock (f_{SUB})
0	1		Low-speed internal oscillator clock (f_{IL}) (must set WUTMMCK0 to 1)
1	0	0	Main clock fmx/fhoco (via RTCCL7 selection)/1464
1	0	1	Main clock fmx/fhoco (via RTCCL7 selection)/976
1	1	0	Main clock fmx/fhoco (via RTCCL7 selection)/488
1	1	1	Main clock fmx/fhoco (via RTCCL7 selection)/244

11.3.3 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 operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 11-4 Format of real-time clock control register 0(RTCC0)

Address: 0x40044F5D After reset: 00H R/W

symbol	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 from running.
1	Start the counter running.

RCLOE1[note]	Output Control of RTC1HZ Pin
0	Disables the output of the RTC1HZ pin (1Hz).
1	Allow RTC1HZ pin output (1 Hz).

AMPM	12-hour system/24-hour system selection
0	12-hour system (for AM or PM)
1	24-hour system
· To change the value of the AMPM bit, the RWAIT bit (bit0 of Real-Time Clock Control Register 1 (RTCC1) must be overridden. If you change the value of the AMPM bit, the value of the HOUR register becomes the corresponding value of the time system you set. · Time frames are shown in Table 11-2.	

CT2	CT1	CT0	Fixed Cycle Interrupt (INTRTC) Selection
0	0	0	Do not use fixed cycle interrupts.
0	0	1	0.5 seconds once (in sync with second roll-up)
0	1	0	Once in 1 second (when accumulated in seconds)
0	1	1	Once in 1 minute (00 seconds per minute)
1	0	0	Once in 1 hour (00 minutes and 00 seconds per hour)
1	0	1	Once-a-day (00:00:00 seconds daily)
1	1	×	Once a month (Every month on the first day of the month at 00:00 seconds)
To change the CT2~CT0 bit value in counter run (RTCE=1), INTRTC must be set to disable interrupt handling by interrupt mask register, and RIFG and RTCIF flags must be cleared after override and then set to allow interrupt handling.			

Note: The RCLOE1 location "0" must be used when the 32-pin product.

Note: 1. You cannot change the RTCE bit when the RCLOE1 bit is '1'.

2. When the RTCE bit is "0", the RCLOE1 position "1" does not output 1 Hz.

Note: ×: Ignore

11.3.4 Real-time clock control register 1 (RTCC1)

This is an 8-bit register that controls the alarm clock interrupt function and the counter wait. The RTCC1 register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 11-5 Format of real-time clock control register 1 (RTCC1) (1/2)

Address: 0x40044F5E After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
RTCC1	WALE	VALLE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	Operation control of alarm clock
0	The consistent run is invalid.
1	Consistent operation is valid.
When setting the WALE bit with the WALIE bit "1" in the counter running (RTCE=1), INTRTC must be set to suppress interrupt handling by the interrupt mask register and the WAFG and RTCIF flags must be cleared after the override. To set the WALIE flag of each alarm clock register (RTCC1 register, ALARMWM), alarm clock hour register (ALARMWH) and alarm clock week register (ALARMWW)) must have the WALE position "0" (invalid		

VALLE	Operation control of INTRTC (alarm clock interrupt) function
0	Interrupt consistently without an alarm clock.
1	Interrupt that alarm clock consistently occur.

WAFG	alarm clock detection status flag
0	The alarm clock is out of sync.
1	Consistent alarm clock detected.
This is a status flag indicating that a consistent alarm clock has been detected. Valid only if WALE bit is '1', becomes '1' after detecting that alarm clock is consistent and after 1 fRTC clock. Clear this flag by writing "0" to it. Invalid operation to write "1".	

Figure 11-5 Format of real-time clock control register 1 (RTCC1) (2/2)

RIFG	Fixed Cycle Interrupt Status Flag
0	No fixed cycle interrupt was generated.
1	Interrupt of a fixed cycle is generate.
This is a status flag indicating that a fixed cycle interrupt is generated. This flag is "1" when a fixed cycle interrupt is generated. Clear this flag by writing "0" to it. Invalid operation to write "1".	

RWST	Wait state flag for real-time clock
0	Counter is running.
1	In read-write mode for the counter.
This is the state indicating whether the setting for the RWAIT bit is valid. The count value must be read and written after confirming this flag as "1".	

RWAIT	Waiting control of real-time clock
0	Set to counter run.
1	Set SEC~YEAR counter to stop running and enter read-write mode of 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, you must end reading and writing in 1 second and return to "0". Up to 1 fRTC clock time is required from RWAIT position '1' to read and write count value (RWST=1). If the internal counter (16 bits) overflows when the RWAIT bit is "1", the overflows are maintained and incremented after the RWAIT bit becomes "0".	

Note: 1. Fixed cycle interrupts and alarm clock consistent interrupts use the same interrupt source (INTRTC). When INTRTC interrupt occurs, which interrupt occurs can be judged by confirming fixed period interrupt state flag RIFG and alarm clock detection state flag WAFG.

2. If you write a second count register (SEC), clear the internal counter (16 bits).

11.3.5 Clock error correction register (SUBCUD)

This is a register capable of correcting clock speed with high accuracy by changing the overflow value from the internal counter (16 bits) to the second counter (SEC) (reference value: 7FFFH).

The SUBCUD register is set by the 16-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00000H".

Figure 11-6 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 Time Sequence for Correcting Clock Error
0	The clock error correction is performed when the second bits are "00", "20", and "40".
1	Clock error correction is only performed when the second bit is "00" (every 60 seconds).
Disable writing SUBCUD registers for the period shown: · DVE=0: Period of SEC=00H, 20H, 40H · DVE=1: Period of SEC=00H	

F12	Setting of clock error correction value
0	{(F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0)-1}2 increase
1	{(/F11,/F10,/F9,/F8,/F7,/F6,/F5,/F4,/F3,/F2,/F1,/F0)+1}2 Reduction
When (F12,F11,F10,F9,F8,F7,F6,F5,F4,F3,F2,F1,F0)=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 Or (1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1), no clock error correction. Range of correction values: (F12=0)2,4,6,8,......,8186,8188 (F12=1)-2,-4,-6,-8,......,-8186,-8188	

Note: "/" denotes the inverse of each.

The range of correction that can be performed by the clock error correction register (SUBCUD) is as follows.

	DEV=0 (correction every 20 seconds)	DEV=1 (correction every 60 seconds)
correctable range	-12496.9 ppm~12496.9 ppm	-4165.6 ppmto 4165.6 ppm
maximum quantization error	±1.53ppm	±0.51ppm
minimum resolution	±3.05ppm	±1.02ppm

Note: The DEV position "0" must be set when the correction range exceeds -4165.6ppm~4165.6 ppm.

11.3.6 Second Count Register (SEC)

This is an 8-bit register that represents the value of the second meter in 0-59 decimal. An incremental count is performed by overflowing an internal counter (16 bits).

At write time, the data is first written to the buffer and then to the counter after passing up to 2 fRTC clocks. Decimal 00-59 must be set in BCD-code.

The SEC register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 11-7 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

Note: When you want to read and write this register in the counter run (RTCE=1),11.4.3 The real-time clock counter reads and writes "in steps.

Note: If you write a second count register (SEC), the internal counter (16 bits) is cleared.

11.3.7 Minute Count Register (MIN)

This is an 8-bit register that represents the minutes value in 0-59 (decimal). Incrementally counts by overflowing the second counter.

At write time, the data is first written to the buffer and then to the counter after passing up to 2 fRTC clocks. The overrun of the second count register is ignored during a write operation and set to a write value. Decimal 00-59 must be set in BCD-code.

The MIN register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 11-8 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

Note: When you want to read and write this register in the counter run (RTCE=1),11.4.3 Real-time clock counter reading and writing"The recorded steps are carried out.

11.3.8 Hour count register (HOUR)

This is an 8bit register that represents hourly values with 00-23 or 01-12, 21-32 decimal values. Incrementally count by overflowing the minutes counter.

At write time, the data is first written to the buffer and then to the counter after passing up to 2 fRTC clocks. The overflow of the minute count register is ignored during a write operation and set to a write value.

The decimal 00~23 or 01~12,21~32 must be set in BCD code according to the bit3(AMPM) setting of RTCC0.

If you change the value of the AMPM bit, the value of the HOUR register becomes the corresponding value of the time system that is set. The HOUR register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "12H".

However, if the AMPM position "1" is reset, the value of this register becomes "00H".

Figure 11-9 Format of the 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

Note: 1. When the AMPM bit is selected as "0" (12 hour system), the bit5 (HOUR20) of the HOUR register indicates AM (1).

2. When you want to read and write this register in the counter run (RTCE=1), you must follow the"11.4.3 Real-time clock counter reading and writing"The recorded steps are carried out.

The setting value of the AMPM bit, the value of the HOUR, and the time are as followsTable11-2in the

Table11-2 Representation of Time Frame

24-hour representation (AMPM=1)		12-hour representation (AMPM=0)	
Time	HOUR register	Time	HOUR register
0	00H	AM12	12H
1	01H	AM1	01H
2	02H	AM2	02H
3	03H	AM3	03H
4	04H	AM4	04H
5	05H	AM5	05H
6	06H	AM6	06H
7	07H	AM7	07H
8	08H	AM8	08H
9	09H	AM9	09H
10	10H	AM10	10H
11	11H	AM11	11H
12	12H	PM12	32H
13	13H	PM1	21H
14	14H	PM2	22H
15	15H	PM3	23H
16	16H	PM4	24H
17	17H	PM5	25H
18	18H	PM6	26H
19	19H	PM7	27H
20	20H	PM8	28H
21	21H	PM9	29H
22	22H	PM10	30H
23	23H	PM11	31H

When the AMPM bit is "0", the value of the HOUR register is 12 hours; When the AMPM bit is "1", the value of the HOUR register is 24 hours.

The bit5 of the HOUR register indicates AM/PM at the 12 hour representation. Morning (AM) is "0" and afternoon (PM) is "1.

11.3.9 Day count register (DAY)

This is an 8-bit register that represents the daily count value in 1-31 decimal. An incremental count is performed by overflowing the hour counter. The counter counts as follows.

- 01~31 (1,3,5,7,8,10,12 months)
- 01-30 (4,6,9,11 months)
- 01~29 (February leap year)
- 01~28 (February normal year)

At write time, the data is first written to the buffer and then to the counter after passing up to 2 fRTC clocks. The overflow of the hour count register is ignored during the write operation and set to the write value. Decimal 01-31 must be set in BCD-code.

The day register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "01H".

Figure 11-10 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

Note: When you want to read and write this register in the counter run (RTCE=1),11.4.3 Real-time clock counter reading and writing"The recorded steps are carried out.

11.3.10 Week count register (WEEK)

This is an 8-bit register that represents the day of the week value in 0-6 decimal. Increment counts in synchronization with the daily counter.

At write time, the data is first written to the buffer and then to the counter after passing up to 2 fRTC clocks. Must set decimal 00~06 with BCD code.

The WEEK register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 11-11 Format of the 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

week	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

Note: 1. The corresponding values of the MONTH and Day Count registers (DAY) are not automatically saved to the Day register (WEEK). The following settings must be made after the reset is removed:

2. When you want to read and write this register in the counter run (RTCE=1), you must follow the"11.4.3 Real-time clock counter reading and writing"The recorded steps are carried out.

11.3.11 Month count register (MONTH)

This is an 8-bit register that represents the monthly count value in 1-12 decimal. The incremental count is performed by overflowing the daily counter.

At write time, the data is first written to the buffer and then to the counter after passing up to 2 fRTC clocks. The daily count register overflow is ignored during the write operation and set to the write value. Decimal 01-12 must be set in BCD code format.

The MONTH register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "01H".

Figure 11-12 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

Note: When you want to read and write this register in the counter run (RTCE=1),11.4.3 Real-time clock counter reading and writing"The recorded steps are carried out.

11.3.12 Year count register (YEAR)

This is an 8-bit register that represents the annualized value in 0-99 decimal. Incrementing counts by overflowing the monthly counter (MONTH). 00,04,08,....., 92, and 96 are leap years.

At write time, the data is first written to the buffer and then to the counter after passing up to 2 fRTC clocks. The MONTH register overflow is ignored during the write operation and set to the write value. Decimal 00-99 must be set with BCD code. The YEAR register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 11-13 Format of Annual 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

Note: When you want to read and write this register in the counter run (RTCE=1),11.4.3 Real-time clock counter reading and writing"The recorded steps are carried out.

11.3.13 Alarm clock minute register (ALARMWM)

This is a register that sets alarm minutes.

The ALARMWM 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: Decimal 00-59 must be set in BCD-code. If you set a value outside of the range, the alarm clock is not detected.

Figure 11-14 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

11.3.14 Alarm clock hour register (ALARMWH)

This is a register that sets alarm clock hours.

The ALARMWH register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "12H".

However, if the AMPM position "1" is reset, the value of this register becomes "00H".

Note: Must set decimal 00~23 or 01~12,21~32 with the BCD code. If you set a value outside of the range, the alarm clock is not detected.

Figure 11-15 Format of Alarm Clock 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

Note: When the AMPM bit is selected as "0" (12-hour system), the bit5 (WH20) of the ALARMWH register indicates AM(0)/PM(1).

11.3.15 Alarm clock week register (ALARMWW)

This is the register that sets the alarm week.

The ALARMWW register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 11-16 Format of Alarm Clock Week Register (ALARMWW)

Address: 0x40044F5CH After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
ALARMW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

An example of setting an alarm clock time is shown below.

Alarm clock set time	week							12-hour note				Twenty-four hours			
	Sunday WW 0	Monday WW 1	Tuesday WW 2	Wednesday WW 3	Thursday WW 4	Friday WW 5	Saturday WW 6	10:00	1 hour	10 points	1 point	10:00	1 hour	10 points	1 point
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
Monday-Friday 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
Monday, Wednesday, Friday 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

11.3.16 port mode register and port register

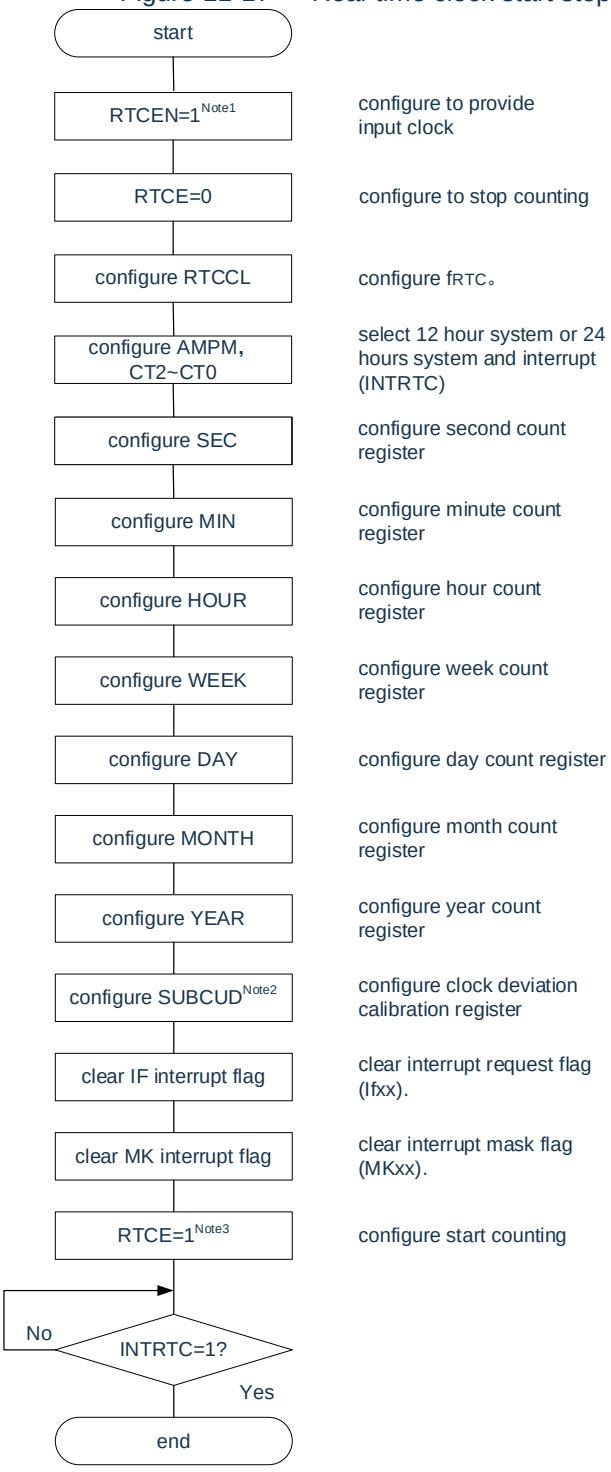
When the multiplexed port of the RTC1HZ output pin is outputted at 1 Hz, the bit of the port mode control register (PMCxx), the bit of the port mode register (PMxx) and the position

The set port mode registers (PMxx), port registers (Pxx), and port mode control registers (PMCxx) differ by product. For more information, refer to Register Settings when Multiplexing is Used in 2.5..

11.4 Operation of real-time clock

11.4.1 Real Time clock Operation start

Figure 11-17 Real-time clock start step



Note: 1. The RTCEN position "1" must be first set in a state where the count clock (fRTC) oscillates.

2. This is only a case where clock errors need to be corrected. Refer to the for how correction values are calculated.11.4.6Example of clock deviation calibration for a real-time clock".

3. Confirm that " when the RTCE bit is " 1 " and is transferred to sleep mode without waiting for the INTRTC bit to " 1 "11.4.2Transition to sleep mode after start of operationThe steps.

11.4.2 Transition to sleep mode after start of operation

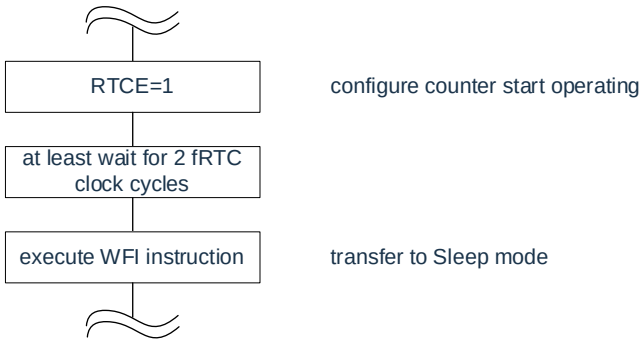
One of the following processes must be performed to transfer RTCE position "1" to sleep, including deep sleep. However, these proces are not required if you want to move to sleep mode after a INTRTC interrupt occurs, after that RTCE position" 1".

· Transfer to sleep mode after passing at least 2 count clocks (f_{RTC}) after RTCE position "1" (referenceFigure 11-18Example 1).

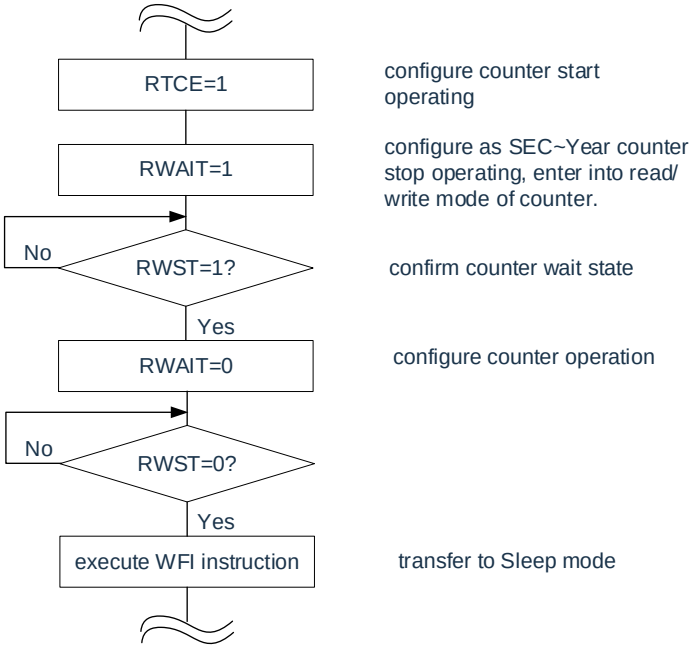
· After RTCE location "1", change the RWAIT location "1" to "1" by polling the RWST. The RWAIT position" 0" is then poll again to confirm that the RWST position is"0" and then transferred to sleep mode (referenceFigure 11-18Example 2).

Figure 11-18 Transfer step of sleep/deep sleep mode after RTCE position "1"

example1



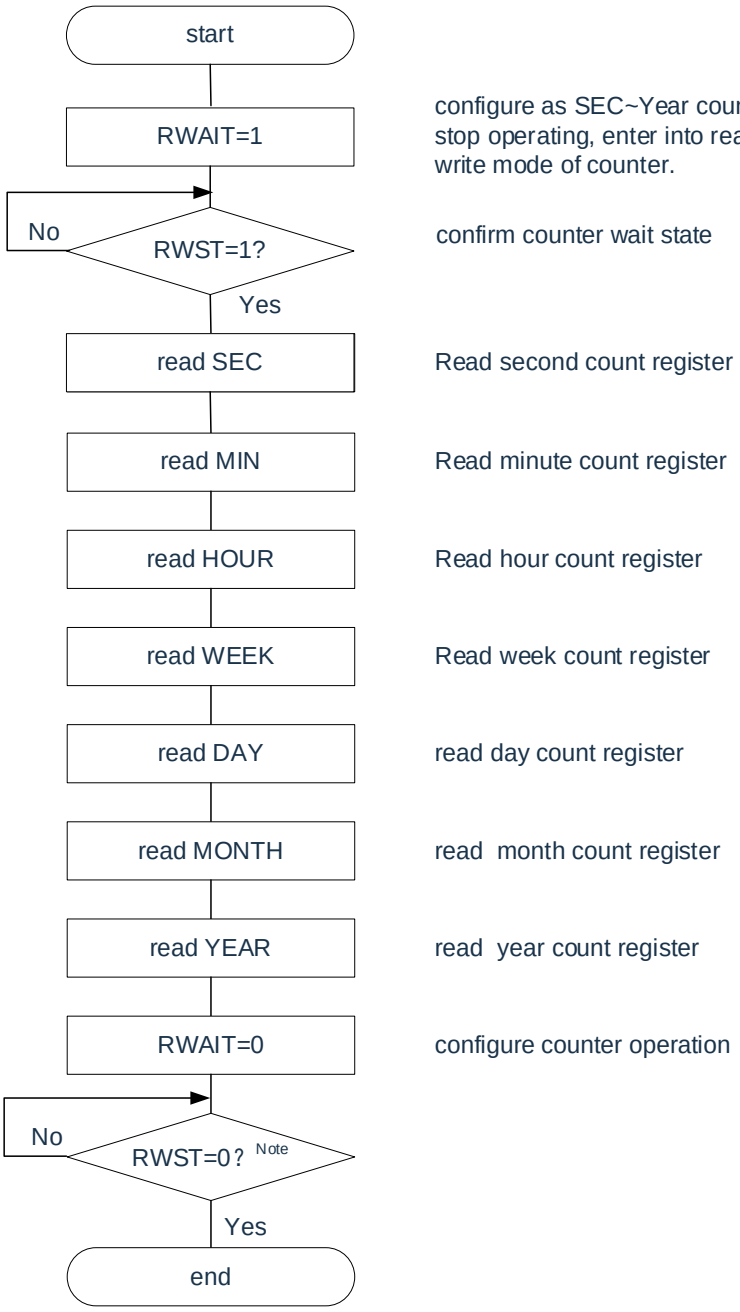
Example 2



11.4.3 Real-time clock counter reading and writing

The RWAIT location "1" must be read and written first. The RWAIT position "0" must be read and written to counters.

Figure 11-19 Read operation steps of real-time clock counter

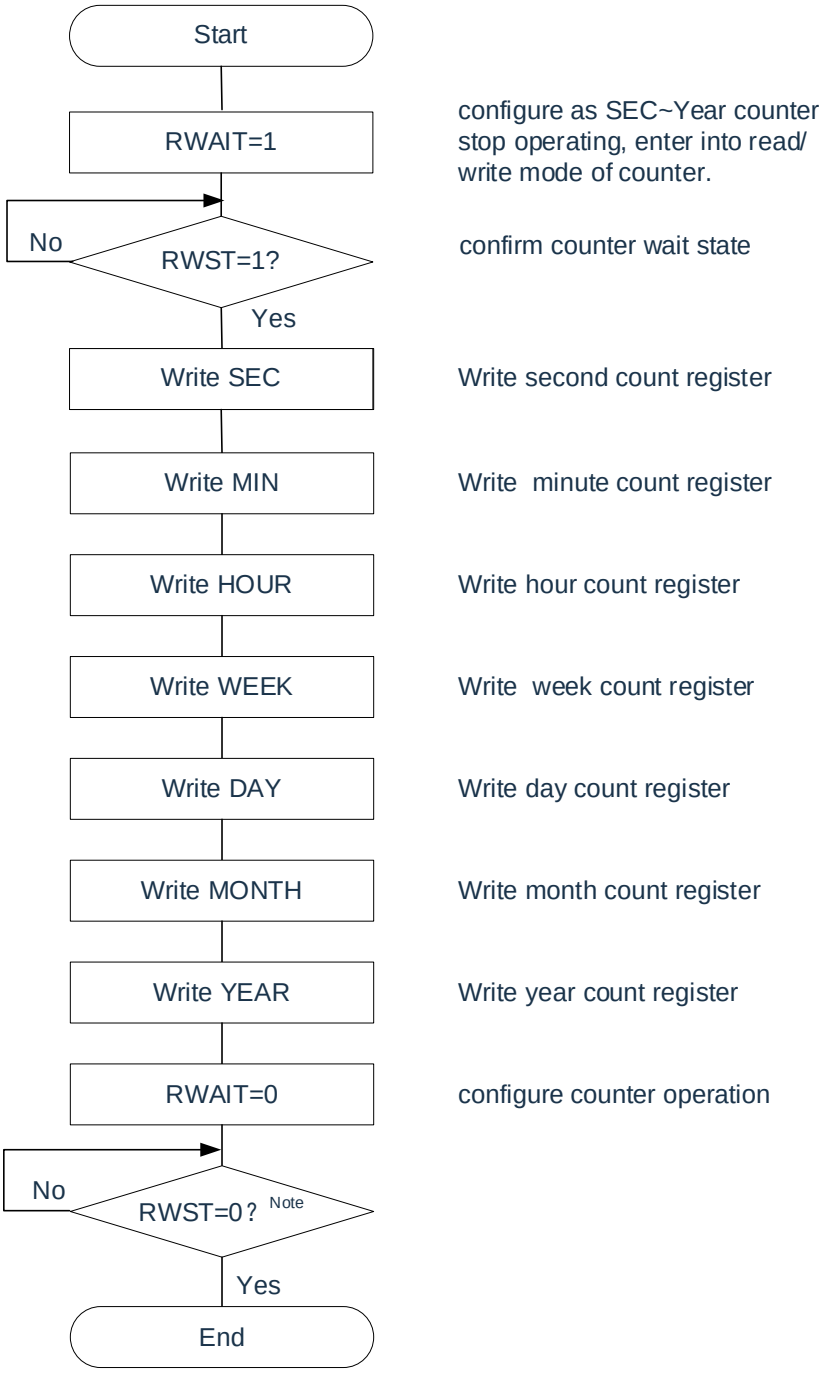


Note: You must verify that the RWST bit is "0" before moving to sleep mode.

Note: The RWAIT position'1' to the RWAIT position'0' must be processed within 1 second.

Note: Do not limit the read order of seconds/minutes/hours/week/day/month/and year count register/s. It is possible to read only part of a register without read all of that register.

Figure 11-20 Read operation steps of real-time clock counter



Note: The RWST bit must be confirmed as '0' before being transferred to SLEEP mode.

Note: 1. The RWAIT position "1" must be processed to the RWAIT position "0" within 1 second.

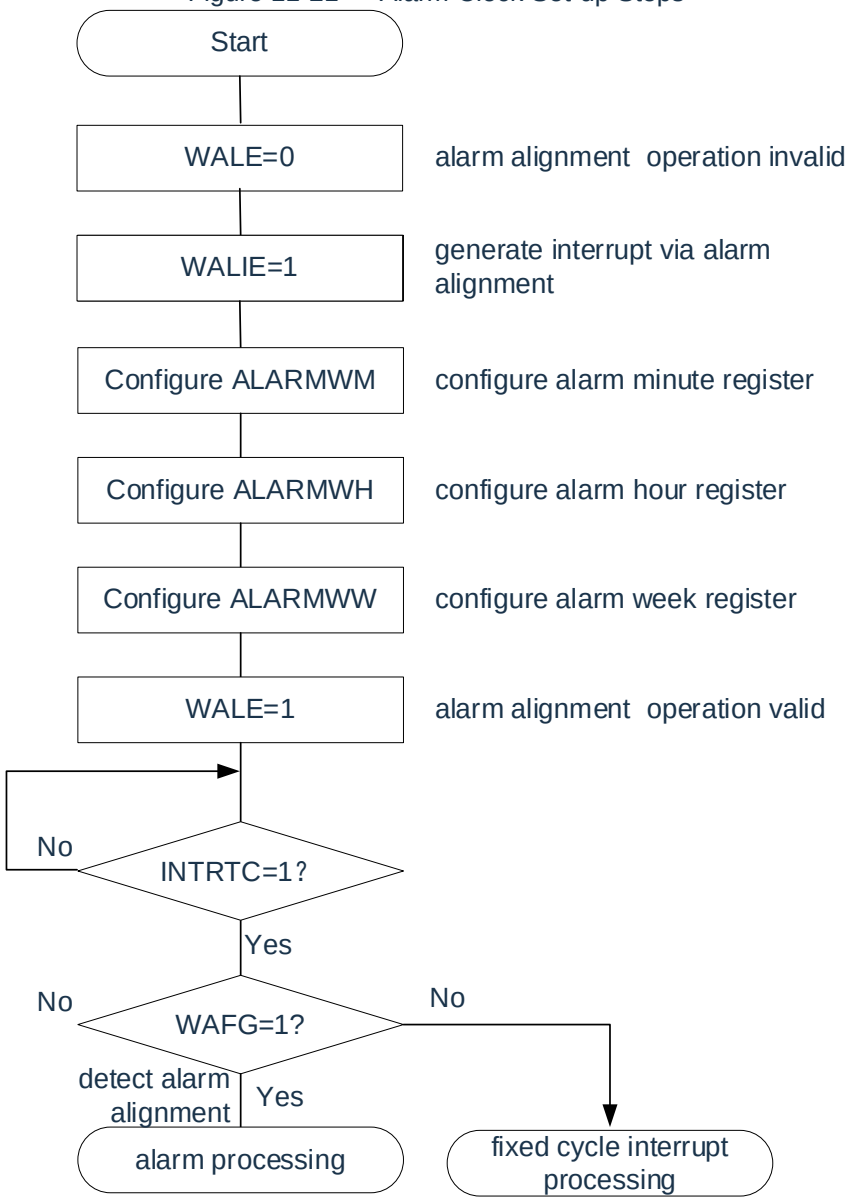
2. To override the SEC, MIN, HOUR, WEEK, DAY, MONTH, YEAR registers in the counter run (RTCE=1), INTRTC must be set by interrupt mask register to suppress interrupt handling for override, and the WAFG flag, RIFG flag, and RTCIF flag must be cleared after override.

Note: Do not limit the read order of seconds/minutes/hours/week/day/month/and year count register/s. It is possible to read only part of a register without read all of that register.

11.4.4 Alarm setting for real-time clock

The WALE location "0" must be set before the alarm clock is not running properly.

Figure 11-21 Alarm Clock Set-up Steps

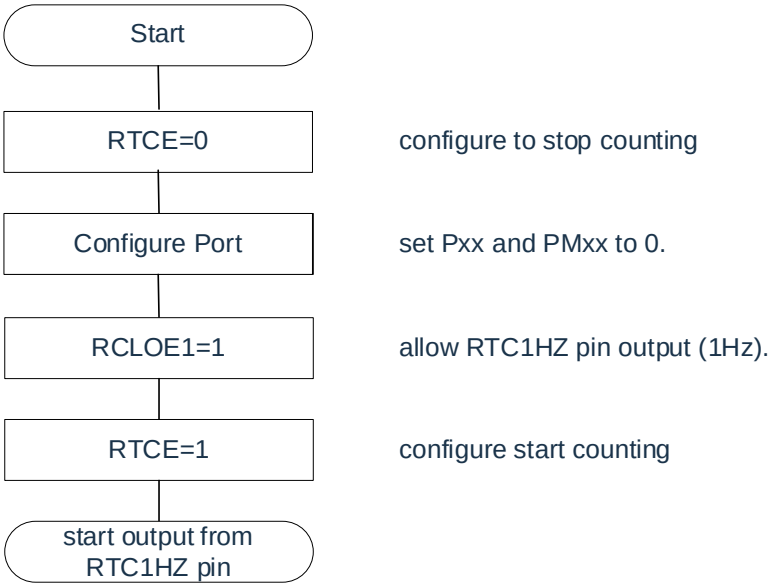


Note:1. Write operation order of ALARMWM, ALARMWH, and ALARMWW is not restricted.

2. Fixed cycle interrupts and alarm clock consistent interrupts use the same interrupt source (INTRTC). When INTRTC occurs, it is possible to determine which interrupt occurs by confirming a fixed period interrupt status flag (RIFG) and an alarm detection status flag (WAFG).

11.4.5 1 Hz output of real-time clock

Figure 11-22 Set-up steps for 1Hz output



Note: 1. The RTCEN position "1" $_{must}$ be first set in a state where the count clock (fSUB) is oscillatingly stable.

2. Some packages do not support the 1 Hz output of real-time clocks.

11.4.6 Example of clock deviation calibration for a real-time clock

A clock speed correction can be performed with high accuracy by setting a value to a clock error correction register.

Example of calculation method of correction value

The correction value for correcting the count value of the internal counter (16 bits) can be calculated using the following formula. The DEV position "0" must be used when the correction range exceeds the -4165.6ppm~4165.6 ppm range.

(The case of DEV=0)

Correction Value [Note] = 1 minute correction count value = (oscillation frequency target frequency -1) ÷32768 ×60× ÷3

(The case of DEV=1)

Correction Value [Note] = 1 minute correction count value = (Oscillation Frequency Target Frequency-1) ×32768×60

Note: The correction value is a clock error correction value calculated based on the value of bit12~0 of the clock error correction register (SUBCUD).

(Case of F12=0) Correction value = {(F11,F10,F9,F8,F7,F6,F5,F4,F3,F2,F1,F0)-1}×2

(Case of F12=1) Correction = -{(/F11,/F10,/F9,/F8,/F7,/F6,/F5,/F4,/F3,/F2,/F1,/F0)+1}×2

When (F12~F0)=(*,0,0,0,0,0,0,0,0,0,0,0,0,0,*), no correction of clock error is performed. * is "0" or "1".

/F12~/F0 is the inverse of each member (when "000000000011", "111111111100).

Note: 1. The correction value is 2,4,6,8,1.....,8186, 8188 or -2,-4,-6,-8,.....,-8186,-8188.

2. The oscillation frequency is the value of the counting clock (fRTC).

Output frequency of the RTC1HZ pin 032768 when clock error correction register is initial value (00H)

3. The target frequency is the frequency corrected using the clock error correction register.

correction example

Examples from 32767.4 Hz to 32768Hz (32767.4Hz+18.3ppm)

[Measurement of oscillation frequency]

The oscillating frequencies of the products are measured by outputting a signal of about 1Hz from the RTC1HZ pin when the clock error correction register (SUBCUD) is an initial value Note.

Note: Refer to the "1 Hz output of the 10.4.5 real-time clock" for RTC1Hz output.

[Calculation of correction values]

(Output frequency of the RTC1HZ pin is 0.999817 Hz)

Oscillation frequency=32768× 0.9999817≈32767.4Hz

Suppose the target frequency is 32768Hz (32767.4Hz+18.3ppm) and DEV=1.

A formula for calculating the correction value when the DEV bit is "1" is applied.

Correction value =1 minute correction count value= ÷(oscillation frequency target frequency-1) ×32768×60

= (32767.4÷32768-1) ×32768 ×60

=-36

[(Calculation of F12~F0) settings]

(Case of correction = -36)

F12=1 because the correction value is less than 0 (for faster cases). The correction values are calculated (F11 to F0).

-{(/F11~/F0)-1}×2=-36

(/F11~/F0)=17

(/F11~/F0)=(0,0,0,0,0,0,1,0,0,0,1)

(F11~F0)=(1,1,1,1,1,1,1,0,1,1,0)

Therefore, from 32767.4 Hz to 32768Hz (32767.4Hz+18.3ppm), the following is true:

If the correction register is set by DEV=1 and correction value =-36 (bit12~0:1,1,1,1,1,1,1,1,1,1,1,1,0 of the SUBCUD register, 32768Hz (0ppm).

Chapter 12 15-bit interval timer

12.1 The function of a 15-bit interval timer

An interrupt (INTIT) is generated at any time interval set in advance, which can be used for arousal from deep sleep mode.

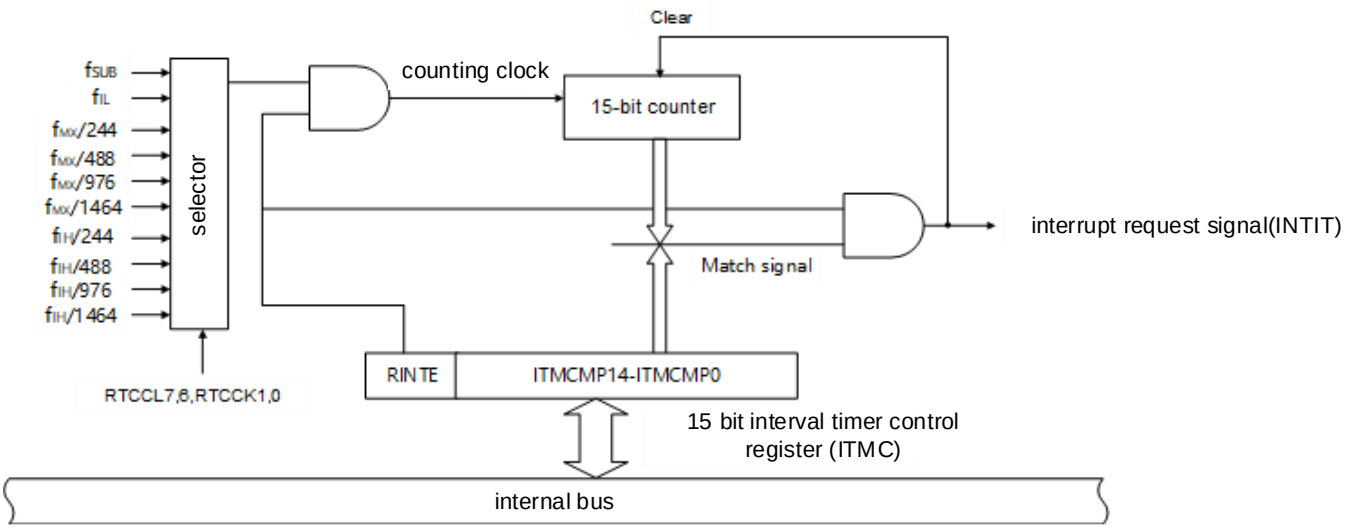
12.2 Structure of 15-bit interval timer

The 15-bit interval timer is composed of the following hardware.

Table12-1 Structure of 15-bit interval timer

Project	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 12-1 Block Diagram of the 15-bit Interval Timer



12.3 Register 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)

12.3.1 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 operation instruction. After the reset signal is generated, the value of this register changes to "00H".

graph12-2 Format of Peripheral Admission Register 0 (PER0)

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

symbol	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDEN	ADCEN	IICA0EN	SCI1EN	SCI0EN	CAN0EN	TM40EN

RTCEN	Control of an input clock of a real-time clock (RTC) and a 15-bit interval timer
0	Stop provide an input clock. · You cannot write the SFR used by the real-time clock (RTC) and 15-bit interval timers. · The real-time clock (RTC) and the 15-bit interval timer are reset.
1	Provides an input clock. · SFRs that can read and write real-time clocks (RTCs) and 15-bit interval timers.

12.3.2 Real-time clock selection register (RTCCL)

A real-time clock and a counter clock (fRTC) of a 15-bit interval timer can be selected through RTCCL.

Figure 12-3 Format of Real-Time Clock Selection Register (RTCCL)

Address: 0x4002047C After reset: 00H R/W

RTCCL

RTCCL7	RTCCL6	0	0	0	0	RTCCS1	RTCCS0

RTCCL7	Selection of Clock Source for Real-time Clock and Counter Clock of 15-bit Interval Timer
0	Select a high speed system clock (fMX)
1	Select a high speed internal oscillator (fhoco)

RTCCS1	RTCCS0	RTCCL6	Selection of Running Clock of Real-time Clock and Counter Clock of 15-bit Interval Timer
0	0	0/1	Secondary System Clock (fSUB)
0	1		Low-speed internal oscillator clock (fIL) (must set WUTMMCK0 to 1)
1	0	0	Main clock fmax/fhoco (via RTCCL7 selection)/1464
1	0	1	Main clock fmax/fhoco (via RTCCL7 selection)/976
1	1	0	Main clock fmax/fhoco (via RTCCL7 selection)/488
1	1	1	Main clock fmax/fhoco (via RTCCL7 selection)/244

12.3.3 Control register for 15-bit interval timer (ITMC)

This is a register that sets the start and stop of the 15-bit interval timer and compares the values. The ITMC register is set by the 16-bit memory operation instruction.

After the reset signal is generated, the value of this register changes to "7FFFH".

Figure 12-4 Format of the 15-bit interval timer control register (ITMC)

Address: 0x40044F50 After reset: 7FFFH R/W

symbol	15	14~0
ITMC	RINTER	ITCMP1 4~ITCMP0

RINTER	Operation Control of 15-bit Interval Timer
0	Stop the counter from running (clear count).
1	Start the counter running.

ITCMP1 4~ITCMP0	15bit interval timer comparison value setting
001H	These bits generate a fixed cycle interrupt for× the "count clock cycle (ITCMP set value +1)".
· The · The · The	
7 FFFH	
0000H	Disable from setting.
Example of interrupt period when ITCMP14~ITCMP0 is "0001H" or "7FFFH" ·ITCMP14~ITCMP0=0001H, count clock:fSUB=32.768kHz 1/32.768 [kHz]×(1+1)=0.06103515625 [ms]≈61.03 [μs] • ITCMP14~ITCMP0=7FFFH, count clock: fSUB=32.768kHz 1/32.768 [kHz]× (32767+1)=1000 [ms]	

Note:

1. When changing the RINTE bit from "1" to "0", you must override by setting INTIT to disable interrupt handling through the interrupt mask register. To restart (from "0" to "1"), you must set to allow interrupt handling after clearing the ITIF flag.
2. The read value of the RINTE bit is reflected after setting the 1 count clock of the RINTE bit.
3. After transferring from sleep mode to normal run mode, if ITMC register is to be set and transferred to sleep mode again, it must be transferred to sleep mode after confirming write value.
4. To change the setting of the ITCMP1 4~ITCMP0 bit, you must do it in the state with the RINTE bit "0". However, you can change the RINTE 4~ITCMP0 bit settings while changing the ITCMP1 bit from 0 to 1 or 1 from 1.

12.4 Operation of a 15-bit interval timer

12.4.1 Run-time sequence of 15-bit interval timer

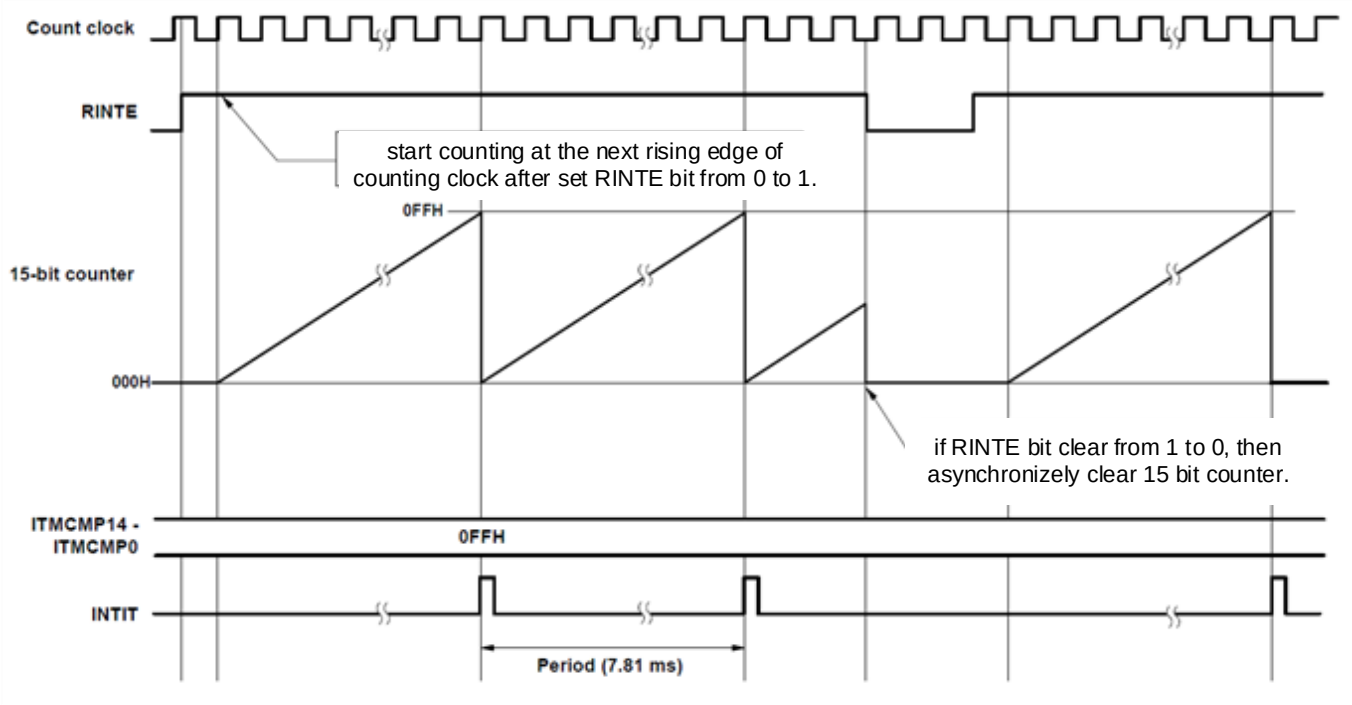
A 15-bit interval timer for repeated generation of interrupt requests (INTIT) is operated at intervals of ITCMP14~ITCMP0 bits. If the RINTE position is "1," the 15-bit counter starts counting.

When the 15-bit count value is equal to the set value of the ITCMP14~ITCMP0 bit, the 15-bit count value is cleared '0' and continues to count, and an interrupt request signal (INTIT) is generated.

The basic operation of the 15-bit interval timer is as followsFigure 12-5in the

Figure 12-5 Run-time sequence of 15-bit interval timer

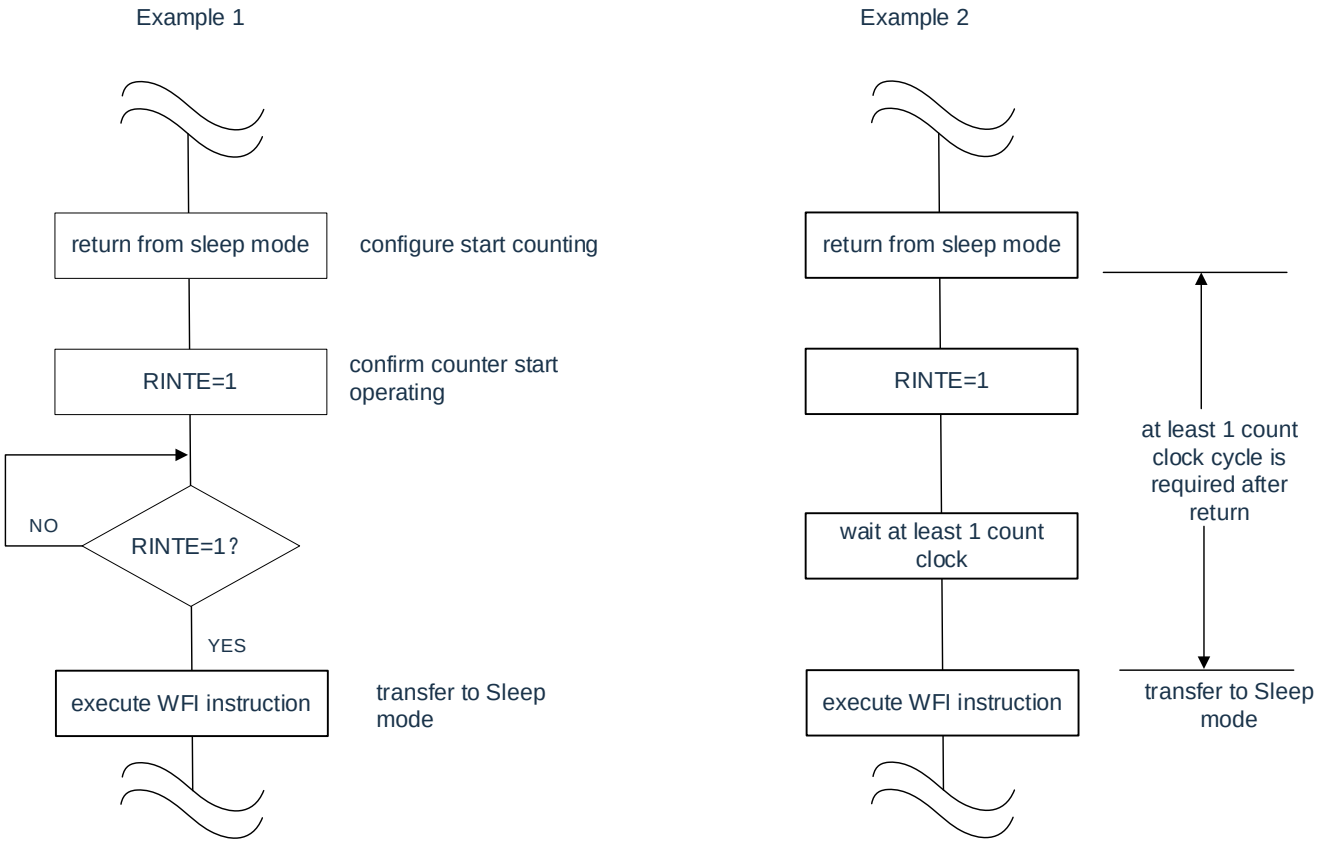
(ITCMP14~ITCMP0=0FFH, count clock: f_{SUB}=32.768kHz)



12.4.2 The operation of the counter is started after returning from the sleep mode and the transfer to the sleep mode is repeated

After returning from sleep mode, if RINTE position '1' is to be transferred to sleep mode again after at least 1 count clock.

- After RINTE position "1" is polled to confirm that the RINTE bit becomes "1" and then moves to sleep mode (see Example 1 of the following figure).
- Transfer to sleep mode after at least 1 count clock time has elapsed since that RINTE position" 1" (see example 2 of the following figure).



Chapter 13 Clock output/buzzer output control circuit

13.1 The Function of Clock Output/Buzzer Output Control Circuit

The output of the clock is the function of output to the peripheral IC clock, and the output of the buzzer is the function of output the frequency square wave of the buzzer.

It can be use as clock output or buzzer output with 1 pin selection.

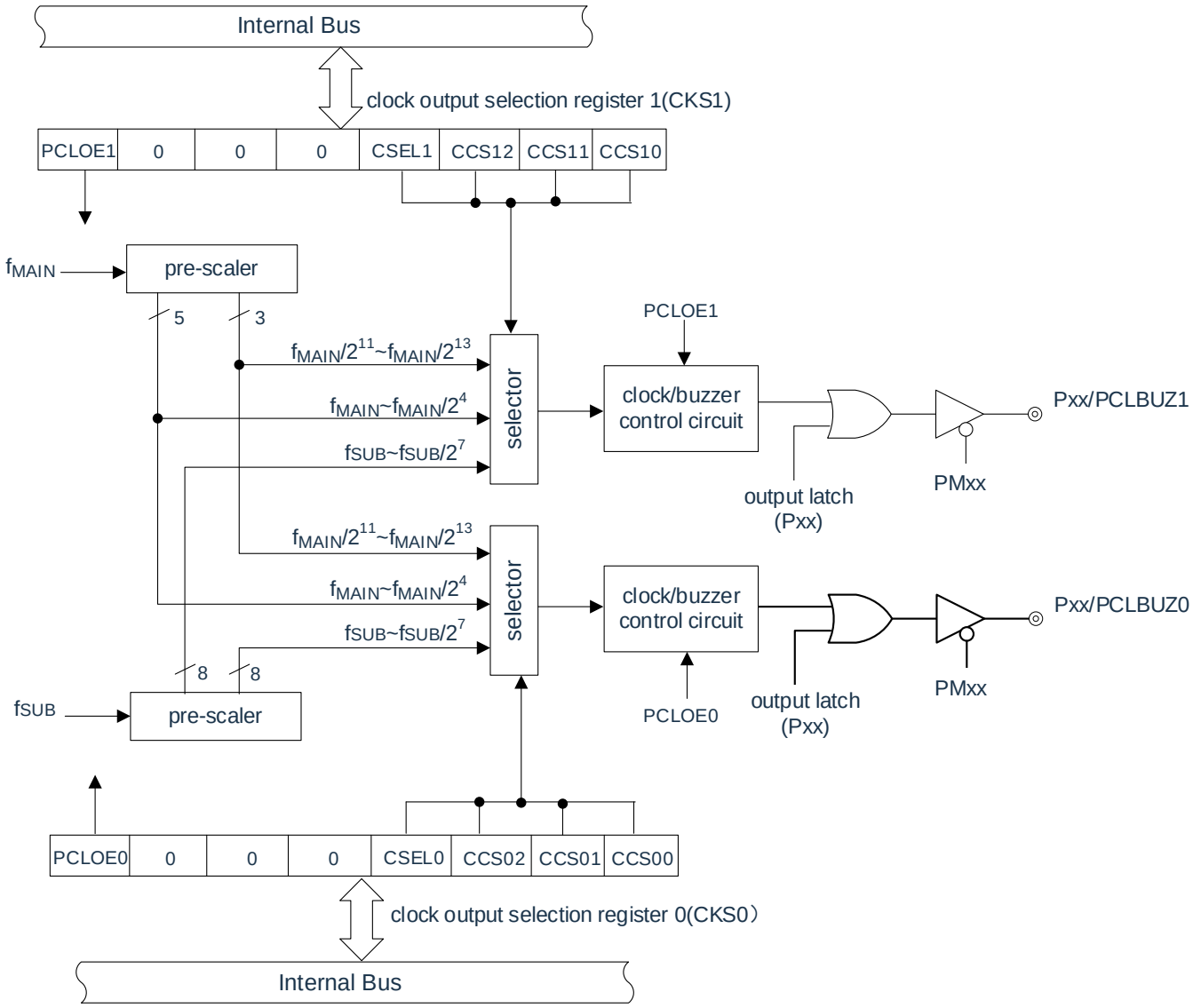
The CLKBUZn pin outputs a clock selected by the clock output selection register n (CKSn).

The block diagram of the output control circuit of the clock output buzzer is as follows in the Figure 13-1

Note: The secondary system clock (fSUB) cannot be output from the CLKBUZn pin in SLEEP mode where the RTCLPC bit of the OSMC is 1.

Notes: n=0,1.

Figure 13-1 Block diagram of clock output/buzzer output control circuit



Note: Refer to "AC Characteristics" for frequencies that can be exported from the CLKBUZ0 and CLKBUZ1 pins.

13.2 Structure of clock output/buzzer output control circuit

The clock output/buzzer output control circuit is composed of the following hardware.

Table13-1 Register for clock output/buzzer output control circuit

Project	register list
control register	Clock output selection register n (CKSn) Port Mode Register (PMmn) Port Register Pmn)

13.3 Register for control clock output/buzzer output control circuit

13.3.1 Clock output selection register n (CKSn)

This is a register that allows or disables the output of the clock output pin or CLKBUZn and sets the output clock.

Select the clock output from the CLKBUZn pin through the CKSn register. The CKSn register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

graph13-2 Format of Clock Output Selection Register n(CKSn)

Addresses: 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	CLKBUZn Pin Output Allowed/Forbidden Assignments
0	Suppress output (default).
1	Allow output.

CSELn	CCSn2	CCSn1	CCSn0		Selection of CLKBUZn Pin Output Clock			
					fMAIN =10M	fMAIN =20M	fMAIN =32M	fMAIN =48M
0	0	0	0	fMAIN	10MHz Notes	Disable from setting Notes	Disable from setting Notes	Disable from setting Notes
0	0	0	1	fMAIN/2	5MHz	10MHz Notes	16MHz Notes	Disable from
0	0	1	0	fMAIN/22	2.5MHz	5MHz	8MHz	12MHz
0	0	1	1	fMAIN/23	1.25MHz	2.5MHz	4MHz	6MHz
0	1	0	0	fMAIN/24	625kHz	1.25MHz	2MHz	3MHz
0	1	0	1	fMAIN/211	4.88kHz	9.77kHz	15.63kHz	23.44kHz
0	1	1	0	fMAIN/212	2.44kHz	4.88kHz	7.81kHz	11.72kHz
0	1	1	1	fMAIN/213	1.22kHz	2.44kHz	3.91kHz	5.86kHz
1	0	0	0	fSUB	32.768kHz			
1	0	0	1	fSUB/2	16.384kHz			
1	0	1	0	fSUB/22	8.192kHz			
1	0	1	1	fSUB/23	4.096kHz			
1	1	0	0	fSUB/24	2.048kHz			
1	1	0	1	fSUB/25	1.024kHz			
1	1	1	0	fSUB/26	512Hz			
1	1	1	1	fSUB/27	256Hz			

Note: The output clock must be used in a range of less than 16 MHz. Refer to "AC Characteristics" for details.

Note:

1. The output clock must be switched after it is set to disable output (PCLOEn=0).
2. When selecting the main system clock (CSELn=0), if you want to transfer to deep sleep mode, you must set the PCLOEn to "0" before executing WFI; When selecting a sub-system clock (CSELn=1), the PCLOEn can be set to '1' because the RTCLPC bit of the OSMC is '0'.
3. The secondary system clock (fSUB) cannot be output from the CLKBUZn pin in a sleep mode in which the RTCLPC bit of the OSMC is 1.

Note:

1. n=0,1
2. f_{MAIN}: main system clock frequency
   f_{SUB}: sub-system clock frequency

13.3.2 Register for controlling clock output/buzzer output pin port function

When used as a clock output/buzzer output function, you must set the control registers (port mode registers (PMxx) and port registers (Pxx)) for the port function multiplexing with the object channel. Refer to "2.3.1 Port Mode Register (PMxx)" and 2.3.2 Port Register (Pxx)" for details.

When using multiplexed ports of clock output/buzzer output pins as clock output/buzzer outputs, the bits of port mode register (PMxx) and the position of port register (Pxx) "0" must correspond

(Example) Using P140/INTP6/CLKBUZ0 as a Clock Output/Buzzer Output

Position PM140 of port mode register 14 at "0.

Position P140 of port register 14 at "0"

13.4 Operation of clock output/buzzer output control circuit

It can be use as clock output or buzzer output with 1 pin selection.

The CLKBUZ0 pin outputs a clock/buzzer selected by the clock output selection register 0 (CKS0).

The CLKBUZ1 pin outputs a clock/buzzer selected by the clock output selection register 1 (CKS1).

13.4.1 Operation of the Output Pin

The CLKBUZn pin follows the steps below to output:

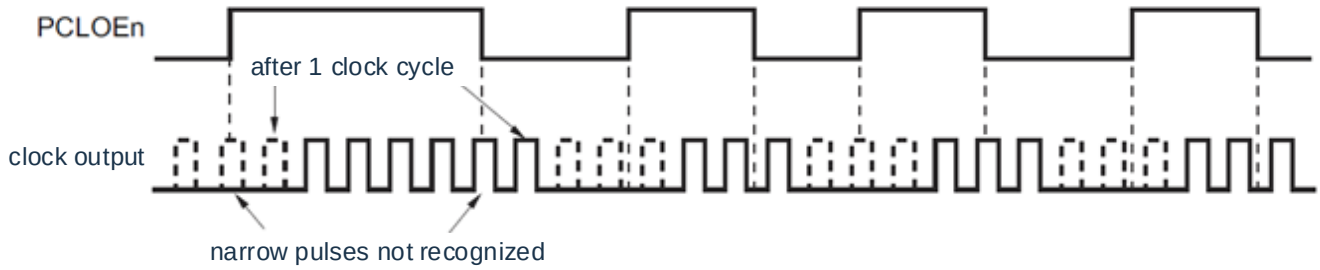
1) Position "0" of port mode register (PMxx) and port register (Pxx) corresponding to the port used as CLKBUZ0 pin.
2) The output frequency (output is forbidden) is selected by bit0~3(CCSn0~CCSn2, CSELn) of clock output selection register (CKSn) of CLKBUZn pin.
3) Set the bit7 (PCLOEn) of the CKSn register to "1" to allow the clock/buzzer output.

Note:

① A control circuit for clock output starts or stops clock output after allowing or disabling 1 clock output (PCLOEn bits). At this time, pulses of narrow width are not output. The timing of the output and the clock output by the PCLOEn bit is as followsgraph13-3in the

② n=0,1

graph13-3 Clock output timing for CLKBUZn pins



13.5 Precautions for Clock Output/Buzzer Output Control Circuit

When the main system clock is selected as the CLKBUZn output (CSELn=0), the output width of the CLKBUZn becomes narrower if the output clock of 1.5 CLKBUZn pins is shifted to the deep sleep mode after setting the stop output (PCLOEn=0).

Chapter 14 watchdog timer

14.1 Function of watchdog timer

The watchdog timer runs with the option byte (000C0H) setting count. The watchdog timer operates at a low speed internal oscillator clock (fIL). The watchdog timer is used to detect program runaway. An internal reset signal is generated when it is detected that the program is out of control.

The following circumstances were determined as out of control of the procedure.

· When the watchdog timer counter overflows

· When the bit operation instruction is executed on the WDTE of the watchdog timer

· When writing data other than "ACH" to the WDTE register

· Write data to the WDTE register during window closure

When a reset occurs due to the watchdog timer, the bit4(WDTRF) of the reset control flag register (RESF) is set "1". For more information on RESF registers, refer to Chapter 28 Reset Features.. Interval interruptions can be generated when the overflow time of 75%+1/2 fIL is reached.

14.2 Structure of watchdog timer

The watchdog timer consists of the following hardware.

Table14-1 Structure of watchdog timer

Project	structure
counter	Internal Counter (17-bit)
control register	Watchdog timer enable register (WDTE)

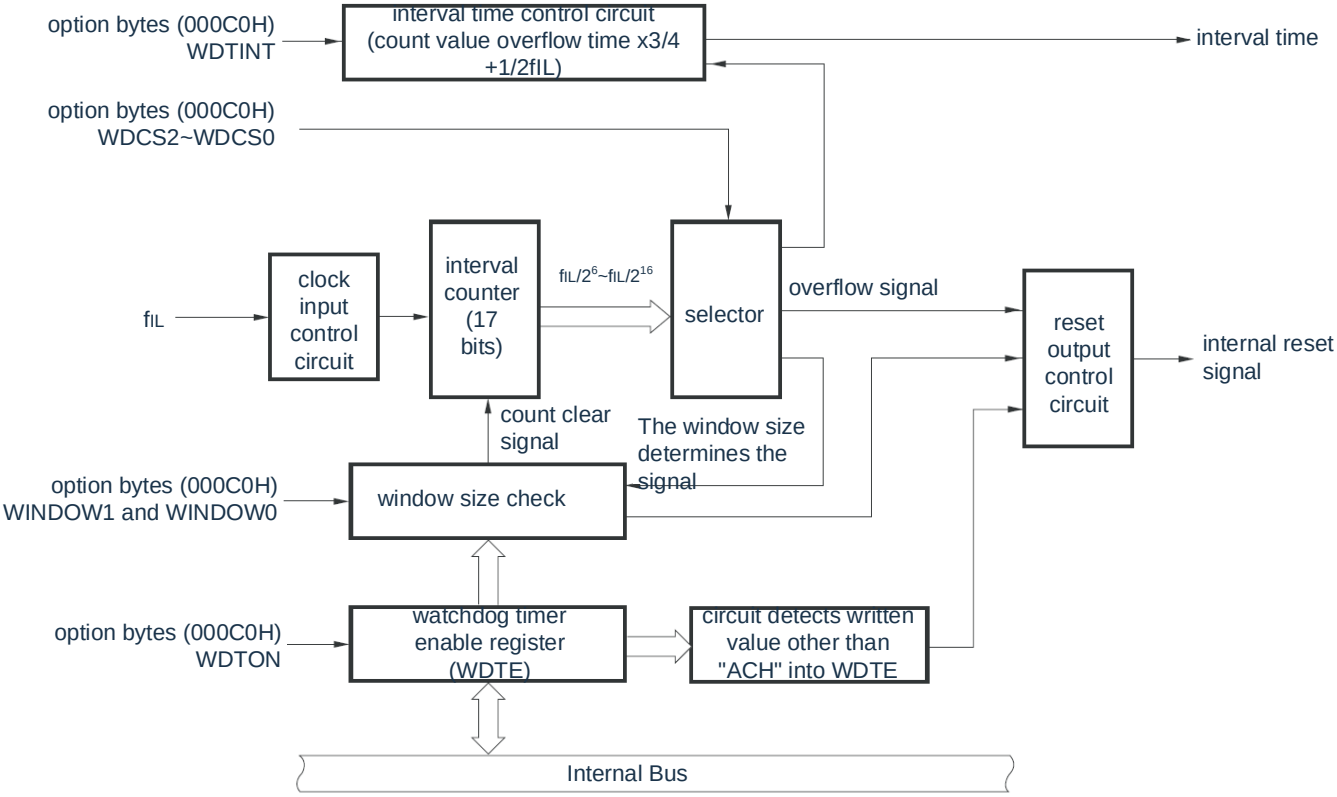
The option bytes control the operation of the counter and the setting of the overflow time, window open period and interval interrupts.

Table14-2 Option bytes and the settings for the watchdog timer

Setting content of watchdog timer	Option Bytes (000C0H)
Setting of interval interrupt for watchdog timer	bit7 (WDTINT)
Settings during window opening	bit6 and bit5 (WINDOW1, WINDOW0)
Counter Operation Control of Watchdog Timer	bit4 (WDTON)
Setting of overflow time of watchdog timer	bit3~1 (WDCS2~WDCS0)
Counter operation control of watchdog timer (during sleep)	bit0 (WDSTBYON)

Note: For option bytes, refer to "Chapter 33 Option Bytes".

Figure 14-1 Watchdog timer block diagram



Note: f_{IL}: Clock frequency of low speed internal oscillator

14.3 Register for controlling watchdog timer

The watchdog timer is controlled by an allowable register (WDTE) of the watchdog timer.

14.3.1 Watchdog timer enable register (WDTE)

By writing "ACH" to the WDTE register, the watchdog timer's counter is cleared and counting restarts. The WDTE register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to '9AH' or '1AH' note.

Figure 14-2 The format of the allowed register (WDTE) for watchdog timer

Address: 0x40021001 After Reset: 9AH/1AH Note R/W

symbol	7	6	5	4	3	2	1	0
WDTE								

Note: The reset value for the WDTE register varies depending on the setting value of the WDTON bit of the option byte (000C0H). The WDTON position "1" must be used for the watchdog timer to run.

Set value for WDTON bit	Reset value for WDTE register
0 (Disable counting of watchdog timer)	1AH
1 (allow the watchdog timer to count)	9AH

Note:

1. An internal reset signal is generated when a value other than "ACH" is written to the WDTE register.
2. An internal reset signal is generated when a bit operation instruction is executed on the WDTE register.
3. The read value of the WDTE register is "9AH/1AH" (different from "ACH").

14.3.2 LOCKUP Control Register (LOCKCTL) and Its Protection Register (PRCR)

The LOCKCTL register is the configuration register of whether the Cortex-M0+ LockUp function causes the watchdog timer to run, and PRCR is its write-protected register.

Set LOCKCTL, PRCR registers via 8-bit memory operation instructions.

After generating a reset signal, the value of the LOCKCTL, PRCR register becomes "00H".

Figure 14-3 LOCKUP Control Register (LOCKCTL). and its protection register (PRCR) format (1/2)

Address：40020405H After reset : 01H R/W

Symbol	7	6	5	4	3	2	1	0
LOCKCTL	0	0	0	0	0	0	0	lockup_rst

lockup_rst	CONFIGURATION OF THE LOCKUP FEATURE
0	• LOCKUP does not cause WDT reset
1	• LOCKUP causes WDT to reset

Figure 14-3 LOCKUP Control Register (LOCKCTL) and its protection register (PRCR) format (2/2)

Address : 40020406H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRCR	PRTKEY[7:1]							PRCR

PRCR	LOCKUP controls register write protection
0	• The LOCKCTL register is not writable
1	• LOCKCTL registers are writable

PRTKEY[7:1]	Write protection of PRCR
78H	• PRCR is writable
other	• PRCR is not writable

14.3.3 WDTCFG Configuration Register (WDTCFG0/1/2/3)

The WDTCFG configuration register is the register that forces the watchdog timer to run.

The WDTCFG registers are set via 8-bit memory operation instructions.

After generating a reset signal, the value of the WDTCFG register becomes "00H".

Figure 14-4 WDTCFG configuration registers (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	• The operation of the watchdog timer after reset is determined by the option byte Note 1Note 2
other				• Force the watchdog timer to run after reset

Note:

1. Refer to the 3.2 User Options Bytes chapter 3.2 for detailed configuration.

14.4 Operation of watchdog timer

14.4.1 Operation control of watchdog timer

1) When you use the watchdog timer, set the following by 000C0H:
   - The bit4 (WDTON) of option bytes (000C0H) must be set to "1" to allow the watchdog timer to count (after reset, the counter starts) (see Chapter 33 option bytes for details).

WDTON	Counter of watchdog timer
0	Disable counting (stop counting after reset).
1	Allow count to run (count begins after reset is removed).

   - You must set the overrun time by bit3~1 (WDCS2~WDCS0) for option bytes (000C0H) (see for details)14.4.2and chapter 33).
   - You must set the window opening period (WINDOW1, WINDOW0) through bit6 and bit5 (,) of option bytes (000C0H) (see for details)14.4.2and chapter 33).

2) After the reset is removed, the watchdog timer starts counting.
3) After counting is started and before the overset time set in the option byte, the watchdog timer is cleared and counting starts again if the WDTE of the watchdog timer is written ACH.
4) Thereafter, the write operation of the WDTE register after the second reset must be performed during the window opening period. If that WDTE register is write during window close, an internal reset signal is generate.
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 when:
   - When a bit operation instruction is executed on a WDTE register
   - When writing data other than "ACH" to the WDTE register

Note

1. Only when writing the WDTE of the watchdog timer for the first time after the reset is released, regardless of window opening.
2. It is possible to generate up to 2 fIL clock errors from writing "ACH" to the WDTE register to clearing the watchdog timer counter.
3. The watchdog timer can be cleared before the count value overflows.
4. As shown below, the watchdog timer runs in sleep or deep sleep mode differently depending on the setting value of bit0 (WDSTBYON) for option bytes (000C0H).

	WDSTBYON=0	WDSTBYON=1
sleep mode	Stop the watchdog timer from running.	Keep the watchdog timer running.
deep sleep mode		

When the WDSTBYON bit is '0', the watchdog timer is counted again after the sleep or deep sleep mode is canceled. At this point, clear the counter "0" and start counting.

When the deep sleep mode is released and the CPU is operated with an X1 oscillating clock, the CPU starts to operate after an oscillating steady time.

If that time from the deepsleep mode to the overturn of the watchdog timer is short, the overturn of the

watchdog will occur within the oscillation stable time and reset. Therefore, after releasing the deep sleep mode through interval interruption, the watchdog timer is to be operated and cleared with the X1 oscillating clock.

14.4.2 Setting of overflow time of watchdog timer

Set the overflow time of the watchdog timer by bit3~1(WDCS2~WDCS0) of the option byte (000C0H).

An internal reset signal is generated when an overturn occurs. If the "ACH" is written to the WDTE of the watchdog timer during the window opening before the overage time, the count is cleared and counting restarts. The overflow times that can be set are shown below.

Table14-3 Setting of overflow time of watchdog timer

WDCS2	WDCS1	WDCS0	Overflow time of watchdog timer ($_{Case}$ of fIL=20kHz(MAX.))
0	0	0	$2^6/f_{IL}$（3.2ms）
0	0	1	$2^7/f_{IL}$（6.4ms）
0	1	0	$2^8/f_{IL}$（12.8ms）
0	1	1	$2^9/f_{IL}$（25.6ms）
1	0	0	$2^{11}/f_{IL}$（102.4ms）
1	0	1	$2^{13}/f_{IL}$（409.6ms）
1	1	0	$2^{14}/f_{IL}$（819.2ms）
1	1	1	$2^{16}/f_{IL}$（3276.8ms）

Note: $_{fIL}$: The clock frequency of a low-speed internal oscillator.

14.4.3 Watchdog timer window settings during opening

Set the window opening period of the watchdog timer by bit6 and bit5 (WINDOW1, WINDOW0) of option bytes (000C0H). The window summary is as follows:

- If a watchdog timer's allow register (WDTE) is written "ACH" during window opening, the watchdog timer is cleared and counting restarts.
- During window closing, even if the WDTE register is written "ACH", an exception is detected and an internal reset signal is generated.

Note: Only when writing WDTE register for the first time after reset is released, the watchdog timer is cleared and counting is resumed, regardless of window opening.

The window opening period that can be set is shown below.

Table14-4 Watchdog timer window settings during opening

WINDOW1	WINDOW0	Watchdog timer window opening period
0	-	Disable from setting
1	0	75%
1	1	100%

Note: When the option byte (000C0H) has bit0 (WDSTBYON) of "0", it is independent of the WINDOW1 bit and WINDOW0 bit values, and the window is 100%.

Note: When setting the overspill time to 2^9/fIL, the window close time and open time are shown below.

	Settings during window opening	
	75%	100%
Window Close Time	0~12.8 ms	None
Window Open Time	12.8~25.6ms	0~25.6 ms

<When the window is open at 75%>

- Overflow time:
- $2^9/f_{IL}$(MAX.)=2^9/20kHz(MAX.)=25.6ms
- Window closed:
- 0～$2^9/f_{IL}$(MIN.)×(1–0.75)=0～2^9/10kHz×0.25=0～12.8ms
- Window opened:
- $2^9/f_{IL}$(MIN.)×(1–0.75)～$2^9/f_{IL}$(MAX.)=12.8～25.6ms

14.4.4 Setting of watchdog timer interval interrupt

Interval interrupts (INTWDTI) can be generated when 75%+1/2fIL is reached by setting bit7 (WDTINT) of option bytes (000C0H).

Table14-5 Setting of watchdog timer interval interrupt

WDTINT	Watchdog timer interval interrupt use/no use
0	Interrupt without interval.
1	Interval interruptions occur when the overset time reaches 75%+1/2 fIL.

Note: When the deep sleep mode is released and the CPU is operated with an X1 oscillating clock, the CPU starts to operate after an oscillating steady time. If that time from the deepsleep mode to the overturn of the watchdog timer is short, the overturn of the watchdog will occur within the oscillation stable time and reset. Therefore, when the deep sleep mode is released by interval interruption, the watchdog timer is to be operated and cleaned by X1 oscillating clock.

Note: Counting continues even after the generation of the INTWDTI (continues until the "ACH" is written to the WDTE of the watchdog timer). If that" ACH" is not write to the WDTE register before the overrun time, an internal reset signal is generate.

14.4.5 WDTCFG is not configured when the watchdog timer is running

When WDTCFG is not configured, the watchdog timer timer starts running automatically, and the overflow time is determined by the overflow time control bit (WDCS2~WDCS0) in the option byte.

Chapter 15 A/D converter

The number of analog input channels of the A/D converter varies depending on the product.

Pin Number	32 pin	48 pins	64 pins
analog transmission access channel	10ch	15ch	16ch
	(ANI0~ANI3, ANI8~ANI12, ANI14)	(ANI0~ANI12) (ANI14~ANI15)	(ANI0~ANI15)

15.1 Functions of A/D Converter

A/D converter is a converter that converts analog input to digital values and can control A/D conversion of up to 19 analog channels.

The A/D converter has the following functions.

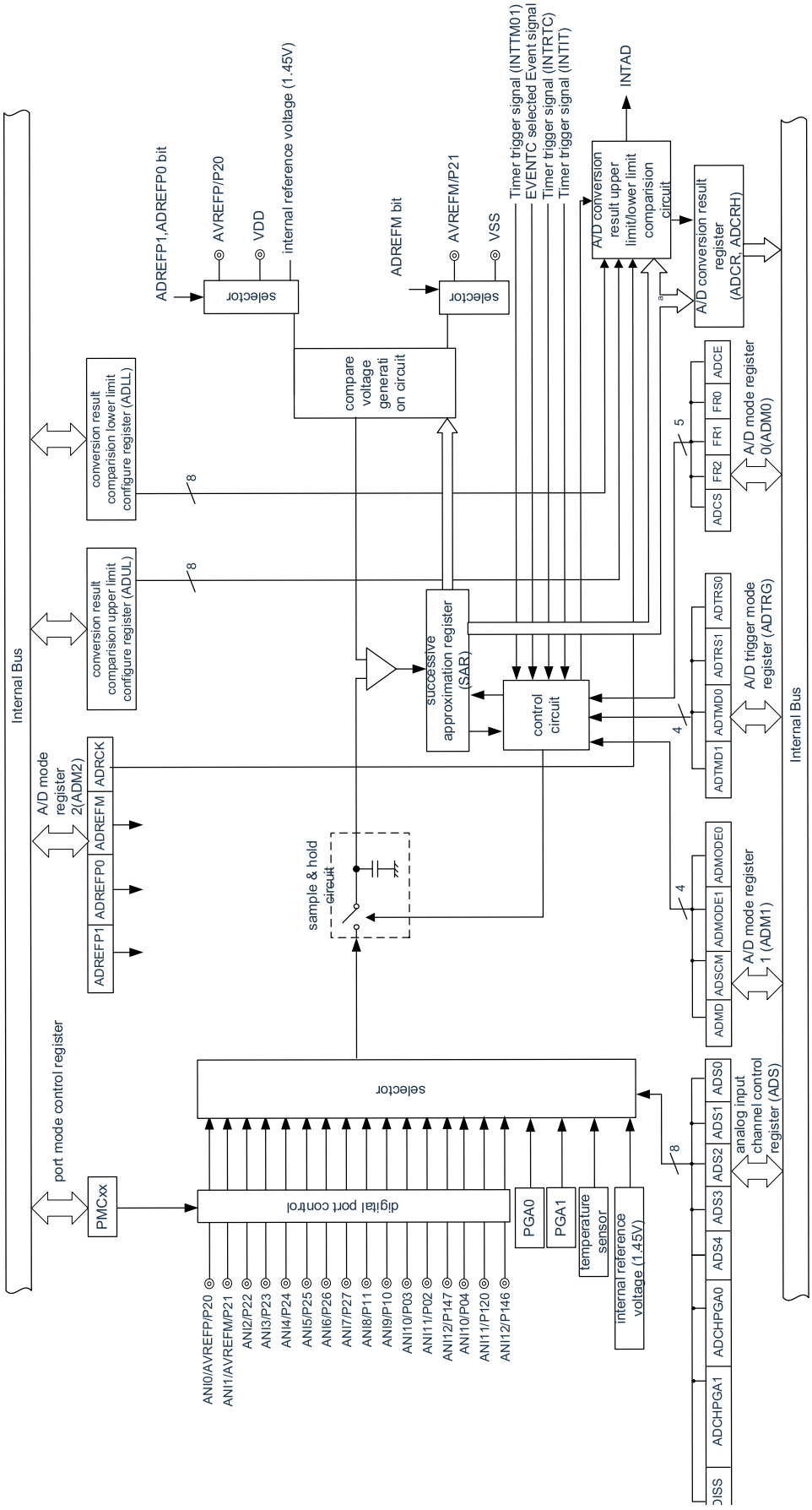
· A/D Conversion of 12-bit Resolution

The 12-bit resolution A/D conversion is repeated by selecting 1 channel analog input from ANI0~ANI15, PGA0, PGA1 and temperature sensors. Each time an A/D transition is completed, an interrupt request (INTAD) is generated (a case of mode selection).

Various A/D conversion modes can be set by the following mode combination.

trigger mode	software trigger	The conversion is started by software operation.
	Hardware triggered no-wait mode	The conversion is started by detecting a hardware trigger.
	Hardware Trigger Wait Mode	In the state of switching off power supply, the switching on power supply is detected by hardware trigger, and the switching is started automatically.
channel selection mode	selection mode	Select the analog input for 1 channel for A/D conversion.
	Scan mode	Analog inputs for the four channels are A/D converted sequentially. It can select 4 channels in ANI0~ANI15 as the analog input.
Conversion Mode	Single Conversion Mode	Make 1 A/D conversion on the selected channel.
	continuous conversion mode	Performs a continuous A/D conversion on the selected channel until it is stopped by the software.
sampling time	Sampling Clock 5.5~255 ADCLK	The sampling time can be selected by the ADNSMP register, using 13.5 conversion clocks ($_{fAD}$).

Figure 15-1 A/D Converter Block Diagram



15.2 Register for controlling A/D converter

The registers that control the A/D converter are as follows:

Register Base Address: CSC_BASE=4002_0420H; ADC_BASE=4004_5000H; PORT_BASE=4004_000H

Register Name	Register Description	R/W	Reset Value	Register Address
PER0	Peripheral Enable Register 0	R/W	00H	CSC_BASE+20H
ADM0	Mode register 0 for A/D converter	R/W	00H	ADC_BASE+00H
ADM1	Mode register 1 for A/D converter	R/W	00H	ADC_BASE+02H
ADM2	Mode register 2 for A/D converter	R/W	00H	ADC_BASE+04H
ADTRG	Trigger mode register of A/D converter	R/W	00H	ADC_BASE+06H
ADS	analog input channel assignment register	R/W	00H	ADC_BASE+08H
ADLL	Conversion Result Compare Lower Limit Value Setting Register	R/W	00H	ADC_BASE+0AH
ADUL	Conversion Result Compare Upper Value 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	Charge and discharge control register of A/D converter	R/W	00H	ADC_BASE+11H
ADSMTWIT	A/D converter sampling time prolonging control register	R/W	00H	ADC_BASE+15H
ADFLG	A/D hard module status register	R	00H	ADC_BASE+16H
PMC0	Port Mode Control Register 0	R/W	FFH	PORT_BASE+60H
PMC1	Port Mode Control Register 1	R/W	FFH	PORT_BASE+61H
PMC2	Port Mode Control Register 2	R/W	FFH	PORT_BASE+62H
PMC12	Port Mode Control Register 12	R/W	FFH	PORT_BASE+6CH
PMC14	Port mode control register 14	R/W	FFH	PORT_BASE+6EH

R:read only, W:write only, R/W:both read and write

15.2.1 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 you want to use an A/D converter, you must set the bit5 (ADCEN).

The PER0 register is set by an 8-bit memory operation instruction.

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

Figure 15-2 Format of Peripheral Admission Register 0 (PER0)

Reset value: 00H
R/W

	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDEN	ADCEN	IICA0EN	SCI 1EN	SCI0EN	CAN0	TM40EN

ADCEN	Control of Input Clock of A/D Converter
0	Stop provide an input clock. · Cannot write the SFR used by the A/D converter. · The A/D converter is in a reset state.
1	Provides an input clock. · Read and write SFRs used by A/D converters.

Note 1.: When you want to set up an A/D converter, you must first set the following register in the state with the ADCEN bit "1". When the ADCEN bit is '0', the control register value of the A/D converter is the initial value, ignoring write operations (except for the port mode control register (PMCxx)).

- Mode register 0 (ADM0) for A/D converter
- Mode register 1 for A/D converter (ADM1)
- Mode register 2 for A/D converter (ADM2)
- A/D converter trigger mode register (ADTRG)
- Analog input channel assignment register (ADS)
- Translation Results Compare Lower Limit Settings Register (ADLL)
- Conversion Result Compare Upper Value Setting Register (ADUL)
- A/D 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 charge/discharge control register (ADNDIS)
- A/D Sample Time Prolongation Control Register (ADSMPWAIT)
- A/D Hard Module Status Register (ADFLG)

15.2.2 Mode register 0 (ADM0) for A/D converter

Register used to set the A/D conversion clock, start of conversion, or stop. The ADM0 register is set by an 8-bit memory operation instruction.

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

Figure 15-3 Format of Mode Register 0 (ADM0) for A/D Converter

Reset value: 00H
R/W

	7	6	5	4	3	2	1	0
ADM0	ADCS	0	FR2	FR1	FR0	0	0	ADCE

ADCS	Control of A/D Conversion Operation
0	Stop the conversion run. [Read] Stop Transition Run/Standby State
1	Allow the transformation to run. [Read] When software triggers mode: Transition Health When hardware triggers wait mode: A/D power supply waits for steady state + transition running state

ADCE	A/D Voltage Comparator Operation Control Note 2
0	Stop the A/D voltage comparator from running.
1	Allows the A/D voltage comparator to run.

Note: 1. For details on FR2-FR0, SHT1-SHT0 and A/D conversion, refer to Table 14-3A/D Conversion.

2. A/D converters take 1s to start running. In that software trigger mode or hardware trigger no wait mode, the conversion result is valid if at least 1s of the ADCE position"1 is pass. If that wait time is less than 1μs and the ADCS position" 1", the conversion result must be ignored. In the hardware triggered wait mode, the design guarantees the wait time of 1μs.

Note 1. You must change the FR2~FR0 bit under the transition stop state ADCS=0.

2. Prohibit ADCS=1, ADCE=0 settings.

3. Disable from setting ADCS=0, ADCE=0 status to ADCS=1, ADCE=1 through 8-bit operation instructions. You must follow the steps of the 15.7 A/D Converter Set-up Flowchart.

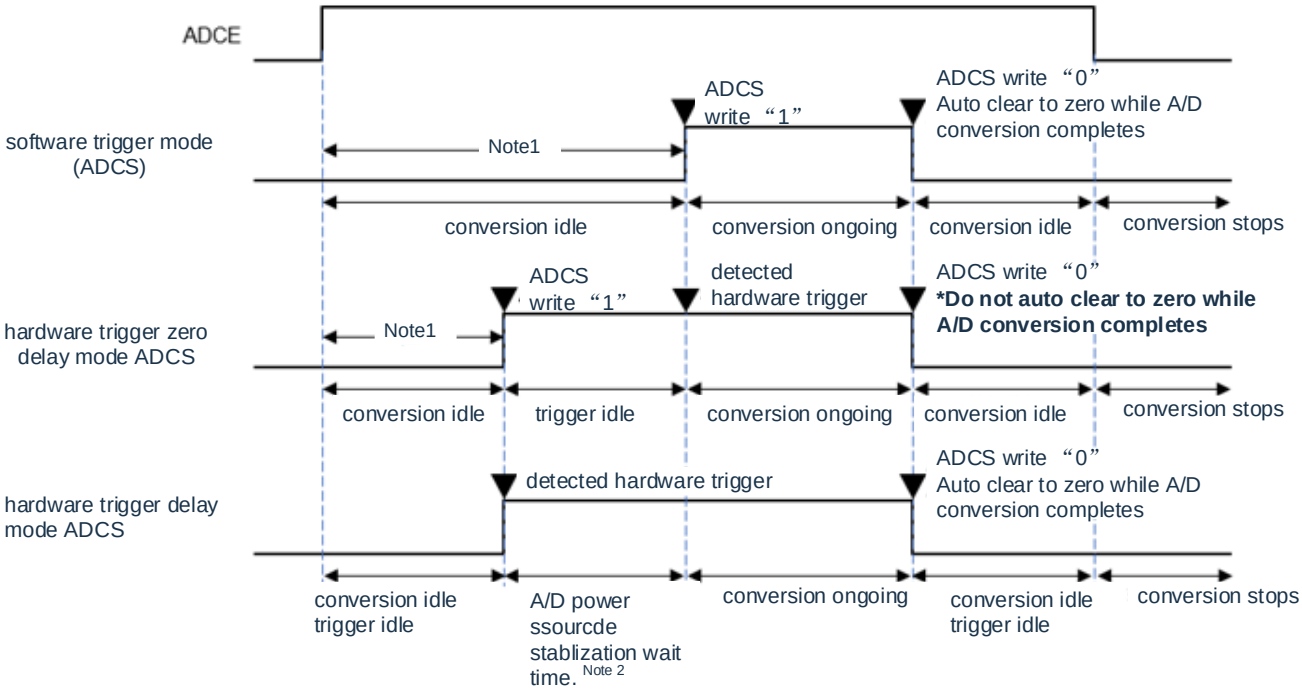
Table 15-1 Settings for ADCS and ADCE bits

ADCS	ADCE	A/D Conversion Run
0	0	Transition Stop Status
0	1	Transition Standby
1	0	Disable from setting.
1	1	Transition Health

Table 15-2 Placement and Purge Conditions for ADCS Bits

A/D conversion mode			placement condition	Clear Condition
software trigger	selection mode	continuous conversion mode	When writing "1" to ADCS bit	When writing "0" to ADCS bit
		Single Conversion Mode		· When writing "0" to ADCS bits · Automatically clears "0" at the end of the A/D conversion.
	Scan mode	continuous conversion mode		When writing "0" to ADCS bit
		Single Conversion Mode		· When writing "0" to ADCS bits · Automatically clears "0" when 4 channel transitions are set up.
Hardware triggered no-wait mode	selection mode	continuous conversion mode	When I give ADCS a bit "1" hour	When writing "0" to ADCS bit
		Single Conversion Mode		· When writing "0" to ADCS bits
	Scan mode	continuous conversion mode		When writing "0" to ADCS bit
		Single Conversion Mode		· When writing "0" to ADCS bits
Hardware Trigger Wait Mode	selection mode	continuous conversion mode	When the input hardware triggers	When writing "0" to ADCS bit
		Single Conversion Mode		· When writing "0" to ADCS bits · Automatically clears "0" at the end of the A/D conversion.
	Scan mode	continuous conversion mode		When writing "0" to ADCS bit
		Single Conversion Mode		· When writing "0" to ADCS bits · Automatically clears "0" when 4 channel transitions are set up.

Figure 15-4 Action State Diagram with A/D Modes



Note: 1. In the software trigger mode or hardware trigger no wait mode, in order to stabilize the internal circuit, the rise time from ADCE bit to ADCS bit needs at least 1s (TBD).

2. In hardware triggered wait mode, A/D power supply stability time 1μs is guaranteed by design.

Note:

1. Disable ADCS position '1' when using hardware trigger wait mode (automatically switches to '1' when a hardware trigger signal is detected). However, to set to A/D to convert standby, the ADCS position "0".
2. The ADCE bit must be overridden when the ADCS bit is "0" (Stop Switching/Switching Standby).
3. To end an A/D conversion, you must set at least the hardware trigger interval to:
   Hardware triggered no-wait mode: 2 fCLK clock + A/D transition times
   Hardware triggered wait mode: 2 fCLK clocks +A/D power supply steady wait time +A/D transition time

Note: $_{fCLK}$: Clock Rate for CPU/Peripheral Hardware

Table 15-3 A/D Conversion Time Selection (1/2)

(1) No A/D power steady wait time (Software Trigger Mode/Hardware Trigger No Wait Mode)

Mode of A/D Converter Register 0 (ADM0)			Mode of A/D Converter Register 1 (ADM1)		mode	Frequency of the conversion clock ADCLK (f_{AD})	12-bit resolution conversion time	
FR2	FR1	FR0	ADMODE [1]	ADMOD [0]			Converted Clocks	Transition Time
0	0	0	0	0	high speed transform mode	f_{CLK}/32	45 ADCLK (Number of sampled clocks: 13.5 ADCLK)	1440/f_{CLK}
0	0	1				f_{CLK}/16		720/f_{CLK}
0	1	0				f_{CLK}/8		360/f_{CLK}
0	1	1				f_{CLK}/4		180/f_{CLK}
1	0	0				f_{CLK}/2		90/f_{CLK}
1	0	1				f_{CLK}/1		45/f_{CLK}
0	0	0	1	1	low current mode	f_{CLK}/32	54 ADCLK (Number of sampled clocks: 13.5 ADCLK)	1728/f_{CLK}
0	0	1				f_{CLK}/16		864/f_{CLK}
0	1	0				f_{CLK}/8		432/f_{CLK}
0	1	1				f_{CLK}/4		216/f_{CLK}
1	0	0				f_{CLK}/2		108/f_{CLK}
1	0	1				f_{CLK}/1		54/f_{CLK}

Note: 1. To override FR2~FR0 bits, ADMODE[1:0] bits to different data, it must be done in the transition stop state (ADCS=0).

Note: f_{CLK}: Clock Rate for CPU/Peripheral Hardware

Table 15-3 A/D Conversion Time Selection (2/2).

(2) Has A/D power steady wait time (hardware triggered wait mode note 1).

Mode of A/D Converter Register 0 (ADM0)			Mode of A/D Converter Register 1 (ADM1)		mode	Frequency of the conversion clock ADCLK (fAD)	A/D power supply stable waiting Time	Converted Clocks	A/D Stable waiting time + Transition Time
FR2	FR1	FR0	ADMODE [1]	ADMOD [0]					
0	0	0	0	0	high speed transform mode	fCLK/32	1μs	45 ADCLK (Number of sampled clocks: 13.5 ADCLK)	1μs +1440/fCLK
0	0	1				fCLK/16			1μs +720/fCLK
0	1	0				fCLK/8			1μs +360/fCLK
0	1	1				fCLK/4			1μs +180/fCLK
1	0	0				fCLK/2			1μs +90/fCLK
1	0	1				fCLK/1			1μs +45/fCLK
0	0	0	1	1	low current mode	fCLK/32	1μs	54 ADCLK (Number of sampled clocks: 13.5 ADCLK)	1μs +1728/fCLK
0	0	1				fCLK/16			1μs +864/fCLK
0	1	0				fCLK/8			1μs +432/fCLK
0	1	1				fCLK/4			1μs +216/fCLK
1	0	0				fCLK/2			1μs +108/fCLK
1	0	1				fCLK/1			1μs +54/fCLK

Note 1: In continuous conversion mode, an A/D power supply steady wait time occurs only after the first hardware trigger is detected (see Table 15-3).

Note: 1. To override FR2~FR0 bits, ADMODE[1:0] bits to different data, it must be done in the transition stop state (ADCS=0).

2. The transition time in Hardware Triggered Wait Mode contains the A/D Power Supply Steady Wait time after the hardware trigger is detected.

Note: $_{fCLK}$: Clock frequency of the CPU/peripheral hardware.

15.2.3 Mode register 1 for A/D converter (ADM1)

This is a register that sets the A/D conversion mode.

The ADM1 register is set by an 8-bit memory operation instruction.

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

Figure 15-6 Format of Mode Register 1 (ADM1) for A/D Converter

Reset value: 00H
R/W

	7	6	5	4	3	2	1	0
ADM1	ADMD	0	0	0	ADSCM	0	ADMODE1	ADMODE0

ADMODE1	ADMODE0	A/D conversion mode
0	0	high speed transform mode
1	1	low current mode
Others		Disable from setting

ADSCM	Settings for A/D Conversion Mode
0	continuous conversion mode
1	Single Conversion Mode

ADMD	Set-up of A/D Conversion Channel Selection Mode
0	selection mode
1	Scan mode

Note:1. To override the ADM1 register, it must be done in the transition stop state (ADCS=0, ADCE=0).

2. In order to end A/D conversion normally, you must set the hardware trigger interval at least as follows:

Hardware triggered no-wait mode: 2 fCLK clock + A/D transition times
Hardware triggered wait mode: 2 fCLK clocks +A/D power supply steady wait time +A/D transition time

Note 1: $_{fCLK}$: Clock Rate for CPU/Peripheral Hardware

15.2.4 Mode register 2 for A/D converter (ADM2)

The ADM2 register is set by an 8-bit memory operation instruction.

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

Figure 15-7 Format of Mode Register 2 (ADM2) for A/D Converter (1/3)

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 (+) Reference Voltage Source for A/D Converter
0	0	Provided by VDD.
0	1	Provided by P20/AVREFP/ANI0.
1	0	Provided by the internal reference voltage of the A/D converter.
1	1	Disable from setting

ADREFM	Selection of Negative (Negative) Reference Voltage Source for A/D Converter
0	Provided by VSS.
1	Courtesy of P21/AVREFM/ANI1.

ADRCK	Examination of the Upper and Lower Limits of Conversion
0	An interrupt signal (INTAD) is generated when the ADLL register ≤ADCR register ≤ ADUL
1	An interrupt signal (ADCR) is generated when the ADCR register <ADLL register (AREA2) or the ADUL register <INTAD register (AREA3).
The generation range of the interrupt signal (INTAD) of AREA1~AREA3 is shown in FIG. 15-8.	

CHRDE	Output enable of channel identification in A/D converter scan mode
0	When scanning mode, channel numbers are not identified in the translation results
1	When scanning mode, the fourth bit of the converted result (ADCR register [15:12]) is the channel number for this result

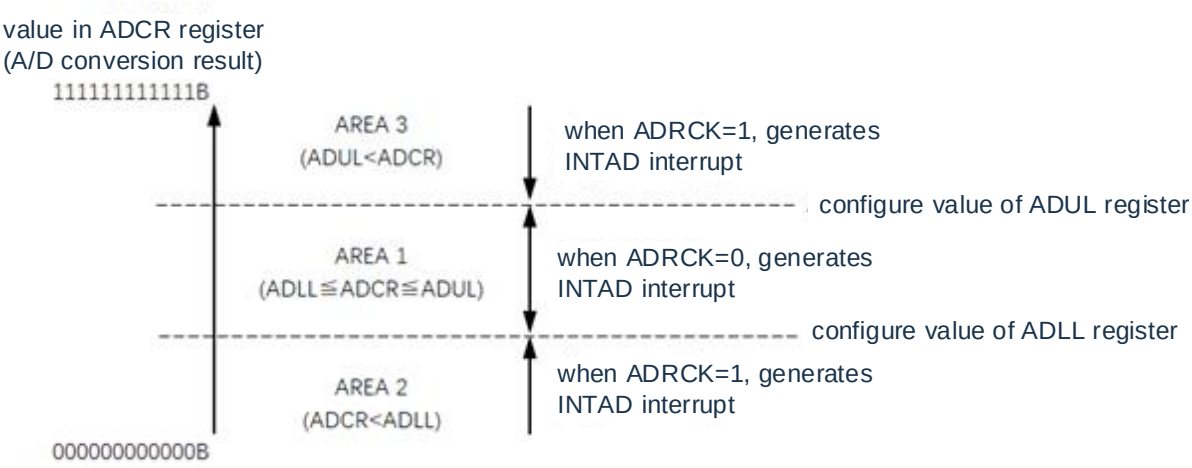


Figure 15-8 Interrupt Signal Generation Range for ADRCK Bits

Note 1. To override the ADM2 register, it must be done in the transition stop state (ADCS=0).

2. When using AVREFP and AVREFM, ANI0 and ANI1 must be set to analog input and set to input mode through the port mode register.

Note: When INTAD does not occur, A/D conversion results are not saved to ADCR registers and ADCRH registers.

15.2.5 A/D converter trigger mode register (ADTRG)

This is a register that sets the A/D conversion trigger mode and hardware trigger signal.

The ADTRG register is set by an 8-bit memory operation instruction.

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

Figure 15-6 Format of the Trigger Mode Register (ADTRG) for A/D Converter

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 Trigger Mode for A/D Conversion
0	0	software trigger mode
0	1	
1	0	Hardware triggered no-wait mode
1	1	Hardware Trigger Wait Mode

ADTRS1	ADTRS0	Selection of hardware trigger signal
0	0	Timer channel 1 counts end or captures end interrupt signal (INTTM01)
0	1	ELC selected event signal
1	0	Real-time clock interrupt (INTRTC)
1	1	Interval timer interrupt signal (INTIT)

Note 1. To override the ADTRG register, it must be done in the transition stop state (ADCS=0, ADCE=0).

2. In order to end A/D conversion normally, you must set the hardware trigger interval at least as follows:

Hardware triggered no-wait mode: 2 fCLK clock + A/D transition times
Hardware triggered wait mode: 2 fCLK clocks +A/D power supply steady wait time +A/D transition time

Note 1. $_{fCLK}$: Clock frequency of the CPU/peripheral hardware

15.2.6 Analog input channel assignment register (ADS)

This is a register that specifies the analog voltage input channel to be A/D converted.

The ADS register is set by an 8-bit memory operation instruction.

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

Figure 15-11 Format of Analog Input Channel Specified Register (ADS)

Reset value: 00H R/W

	7	6	5	4	3	2	1	0
ADS	ADISS	ADCHPGA1	ADCHPGA0	ADS4	ADS3	ADS2	ADS1	ADS0

◆ Select Mode (ADM1.ADMD=0)

ADS register setting value								CH Selection
ADISS	ADCHPGA1	ADCHPGA0	ADS4	ADS3	ADS2	ADS1	ADS0	
0	0	0	0	0	0	0	0	ANI0 (P20)
0	0	0	0	0	0	0	1	ANI1 (P21)
0	0	0	0	0	0	1	0	ANI2 (P22)
0	0	0	0	0	0	1	1	ANI3 (P23)
0	0	0	0	0	1	0	0	ANI4 (P24)
0	0	0	0	0	1	0	1	ANI5 (P25)
0	0	0	0	0	1	1	0	ANI6 (P26)
0	0	0	0	0	1	1	1	ANI7 (P27)
0	0	0	0	1	0	0	0	ANI8 (P11)
0	0	0	0	1	0	0	1	ANI9 (P10)
0	0	0	0	1	0	1	0	ANI10 (P03)
0	0	0	0	1	0	1	1	ANI11 (P02)
0	0	0	0	1	1	0	0	ANI12 (P147)
0	0	0	0	1	1	0	1	ANI13 (P04)
0	0	0	0	1	1	1	0	ANI14 (P120)
0	0	0	0	1	1	1	1	ANI15 (P146)
0	0	0	1	1	1	1	1	ANI0~ANI15 Channel Full Switch
1	0	0	0	0	0	0	0	Output Voltage Note 1 for Temperature Sensor
1	0	0	0	0	0	0	1	Internal Reference Voltage (1.45V)
0	1	0	0	0	0	0	0	PGA1
0	0	1	0	0	0	0	0	PGA0
Others disable from setting								

Note: 1. If the internal reference voltage (1.45V) is selected as the comparator 0 or the reference voltage of 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
0	1	1	0	0	ANI12	ANI1 3	ANI14	ANI15
Other than above					Disable from setting.			

Note 1. bit4, bit5 and bit6 must be set to 0.

2 A/D conversion can be performed by ADS for ports that are set by the PMCx register as analog inputs.

3. The pin set by the port mode control register (PMCxx) as the digital input/output cannot be set by the ADS register.

4. To override the ADISS bit, it must be done in the transition stop state (ADCS=0, ADCE=0).

5. You cannot select AVREFP as the A/D conversion channel when you use ANI0 as the positive (+) reference voltage for A/D converters.

6. When using AVREFM as the negative (-) reference voltage of an A/D converter, you cannot select ANI1 as the A/D converter channel.

7. You cannot use the first transition result after the ADISS position "1" is used. For detailed set-up procedures, refer to "Settings when 15.5.5 Select the Output Voltage/Internal Reference Voltage of the Temperature Sensor."

8. The ADISS position "1" cannot be transferred to deep sleep mode or to sleep mode when the CPU is in sub-system clock operation.

15.2.7 12-bit A/D conversion result register (ADCR)

This is a 16-bit register that holds the A/D conversion result, which is readable only. Each time the A/D conversion ends, the Translation Result Note is loaded from the Successive Approximation Register (SAR) Note.

The high 4-bit readout value of the register is fixed to '0' when selecting mode, and the channel number of this conversion result can be configured by ADM2.CHRDE=1.

The ADCR register is read by a 16-bit memory operation instruction.

After the reset signal is generated, the value of this register changes to "0000H".

Note: A/D conversion results are not saved if their values are not within the set values of the A/D conversion results comparison function (set through the ADRCK bit and the ADUL/ADLL register (FIG. 15-8).

Figure 15-9 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]											

Note 1. If only 8-bit resolution A/D conversion results are required, the high 8-bit conversion results can be read through the ADCRH register.

2. When 16-bit access to the ADCR register, the high 12 bits of the translation result can be read in sequence from bit11.

- Select Mode (ADM1.ADMD=0)

  The readout value of ADCH0~3 is fixed to 4‘b0000

- Scan mode (ADM1.ADMD=1) and ADM2.CHRDE=1,ADCH0~3 read-out values are as follows:

ADCH3	ADCH2	ADCH1	ADCH0	Conversion Channel Identification
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 (P27)
1	0	0	0	ANI8 (P11)
1	0	0	1	ANI9 (P10)
1	0	1	0	ANI10 (P03)
1	0	1	1	ANI11 (P02)
1	1	0	0	ANI12 (P147)
1	1	0	1	ANI13 (P04)
1	1	1	0	ANI14 (P120)
1	1	1	1	ANI15 (P146)

15.2.8 8-bit A/D conversion result register (ADCRH)

This is an 8-bit register that holds the A/D conversion result, holding a high 8-bit note with 12-bit resolution Note.

The ADCRH register is read by an 8-bit memory operation instruction.

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

Note: A/D conversion results are not saved if their values are not within the set values of the A/D conversion results comparison function (set through the ADRCK bit and the ADUL/ADLL register (FIG. 15-8).

Figure 15-10 Format of 8-bit A/D Conversion Result Register (ADCRH)

Reset value: 00HR

	7	6	5	4	3	2	1	0
ADCRH								

Note: The result of the conversion must be read after the conversion is completed and before the ADM0, ADS registers are configured. Otherwise, you may not be able to read the correct conversion results.

15.2.9 Conversion Result Compare Upper Value Setting Register (ADUL)

This is the setting register for checking the upper limit value of the A/D conversion result.

The A/D conversion result is compared with the value of the ADUL register, and the ADRCK in the mode register 2 (ADM2) of the A/D converter

The generation of the interrupt signal (INTAD) is controlled within the setting range of the bit (refer to Figure 15-8). The ADUL register is set by an 8-bit memory manipulation instruction.

After the reset signal is generated, the value of this register becomes "FFH".

Note 1. Only the high 81-bit and ADLL registers of the 12-bit A/D conversion result register (ADCR) are compared to the ADUL register.

2. To override the ADUL and ADLL registers, it must be done in the transition stop state (ADCS=0).

3. When setting the ADUL register and the ADLL register, you must make ADUL>ADLL.

Figure 15-12 Conversion results compare the format of the Upper Value 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

15.2.10 Translation Results Compare Lower Limit Settings Register (ADLL)

This is the set-up register used to check the lower limit of the A/D conversion result.

The A/D conversion result is compared with the value of the ADLL register, and the ADRCK of the A/D converter's mode register 2 (ADM2) is

The generation of an interrupt signal (INTAD) is controlled within a set range of bits (reference FIGS. 15-8). The ADLL register is set by an 8-bit memory operation instruction.

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

Figure 15-13 Conversion Results Compare Format of 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

Note:1. Only the 12-bit A/D conversion result register (ADCR) is compared to the high 8-bit and ADLL registers.

2. To override the ADUL and ADLL registers, it must be done in the transition stop state (ADCS=0).

3. When setting the ADUL register and the ADLL register, you must make ADUL>ADLL.

15.2.11 A/D Sample Time Control Register (ADNSMP)

This register controls the A/D sampling time.
The ADNSMP register is set by an 8-bit memory operation instruction.
After the reset signal is generated, the value of this register changes to '0dH'.

Figure 15-14 A/D Sampling Time Control Register (ADNSMP) Format

Reset value: 0dH R/W

	7	6	5	4	3	2	1	0
ADNSMP	ADNSMP[7:0]							

Sample clock setting:

ADNSMP [7:0]	sampling 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
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	

Note: To override the ADNSMP register, it must be done in the transition stop state (ADCS=0).

Under different conditions, the sampling time of each channel should be guaranteed:

A/D conversion mode	AVDD[V]	ANI0~ANI15[ns]	PGA0/PGA1 [ns]
high speed transformation	4.5~5.5	211	633
	2.7~5.5	250	750
	2.4~5.5	422	1266
low current transformation	2.7~5.5	500	759
	2.4~5.5	844	1281
	1.8~5.5	1688	2563

15.2.12 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 operation instruction.

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

Figure 15-14 A/D Sampling Time Extension Register (ADSMPWAIT) Format

Reset Value: 00H R/W

	7	6	5	4	3	2	1	0
ADSMTWIT	0	0	0	0	0	0	0	ADSMTWIT

ADSMTWIT	A/D conversion object
0	When "0", the A/D sampling time is set directly by the ADNSMP register
1	The sampling time of A/D is arbitrarily prolonged for "1", and is controlled continuously by ADNSMP after changing from "1" to "0"

Note: Set ADSMPWAIT=1 in the transition stop state (ADCS=0) to override ADSMPWAIT to'0' at (ADCS=1).

15.2.13 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 operation instruction.
After the reset signal is generated, the value of this register changes to "00H".

Figure 15-14 A/D test register (ADTES) format

Reset Value: 00H R/W

	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	ADTES3	ADTES2	ADTES1	ADTES0

ADTES2	ADTES1	ADTES0	A/D operation mode
0	0	0	Normal Conversion
0	0	1	Self-Diagnostic Test for 0 Code
0	1	1	Self-diagnostic test of half code
1	0	1	Self-diagnostic testing of full codes
Other than above			Disable from setting.

15.2.14 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 operation instruction.
After the reset signal is generated, the value of this register changes to "00H".

Figure 15-14 A/D State Register (ADFLG) Format

Reset value: 00HR

	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	When a Single Conversion Mode is performed, the transition ends (2 ADCLK is automatically zeroed out). ADFLG4 remains 1'b0 in 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 (1 ADCLK auto-zeroing)

ADFLG2	A/D transition state
0	2 ADCLK before non-A/D conversion ends
1	2 ADCLK before A/D conversion ends (1 ADCLK auto-zeroing)

ADFLG1	A/D transition state
0	Non-sequential comparison period
1	Compare Period By Period

ADFLG0	A/D transition state
0	Non-A/D sampling period
1	A/D Sampling Period

15.2.15 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 operation instruction.

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

Figure 15-15 A/D Charge/Discharge Control Register (ADNDIS) Format

Reset value: 00HW

	7	6	5	4	3	2	1	0
ADNDIS	0	0	0	ADNDIS4	ADNDIS3	ADNDIS2	ADNDIS1	ADNDIS0

ADNDIS [4]	charge-discharge control
1'b0	discharge
1'b1	charging

ADNDIS [3:0]	charge and discharge time
4'b0000	No charging or discharging
4'b0010	2 x ADCLK
4'b0011	3 ADCLK
4'b0100	4 x ADCLK
4'b0101	5 ADCLK
4'b0110	6 x ADCLK
.....	.....
4'b1111	15 ADCLK

Note: Disable from setting charge/discharge time to 1 ADCLK, i.e. ADNDIS[3:0]=4 ‘b0001

15.2.16 Register for controlling analog input pin port function

You must set up a control register (port mode control register (PMCxx)) for the port function that is multiplexed with the analog input of the A/D converter. Refer to the "2.3.6 port mode control register (PMCxx)".

When using ANI0~ANI15 pins as analog input to an A/D converter, the position of the port mode control register (PMCxx) for each port must be "1".

15.3 Input voltage and conversion results

The analog input voltage of the analog input pin (ANI0~ANI15) and the theoretical A/D conversion result register (ADCR) are related.

$$ADCR=INT(\frac{V_{AIN}}{AV_{REF}}\times 4096+0.5) \text{ or } (ADCR-0.5)\times\frac{AV_{REF}}{4096}\le V_{AIN}<(ADCR+0.5)\times\frac{AV_{REF}}{4096}$$

INT(): Function that returns the integer part of the value in parentheses

V_{AIN}: analog input voltage

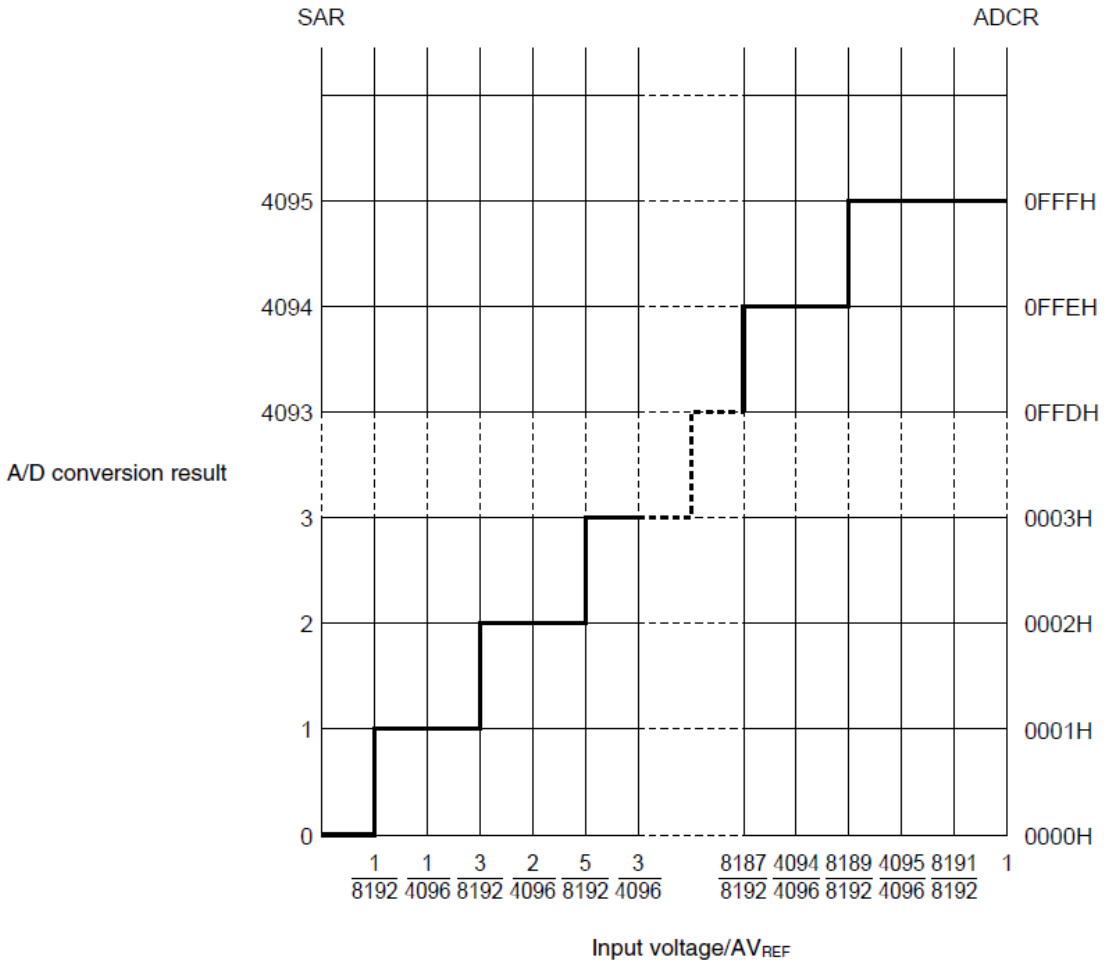
AV_{REF}:AVREF Pin Voltage

ADCR:The value of the A/D conversion result register (ADCR)

SAR: successive approximation register

The relationship between the simulated input voltage and the A/D conversion result is shown in Figure 15-16.

Figure 15-16 The Relationship between Analog Input Voltage and A/D Conversion Result



Note: AVREF is the positive (+) reference voltage of the A/D converter, either AVREFP or VDD can be selected.

15.4 Operation mode of the A/D Converter

The modes of the A/D converter are operated as follows. Refer to the "Set-up Flowchart for 15.5A/D Converter" for set-up procedures.

15.4.1 Software trigger mode (select mode, continuous conversion mode)

① In the stopped state, the ADCE position "1" of mode register 0 (ADM0) of the A/D converter is switched into standby.

② After the stable waiting time (1μs) is counted by software, the ADM0 register ADCS position "1" is A/D converted to the analog input specified by the analog input channel assignment register (ADS).

③ If the A/D conversion ends, the conversion result is saved to the A/D conversion result register (ADCR, ADCRH) and the A/D conversion end interrupt request signal (INTAD). The next A/D conversion begins immediately after the A/D conversion.

④ If the ADCS bit is overridden "1" during the conversion, the current A/D conversion immediately aborts and starts again.

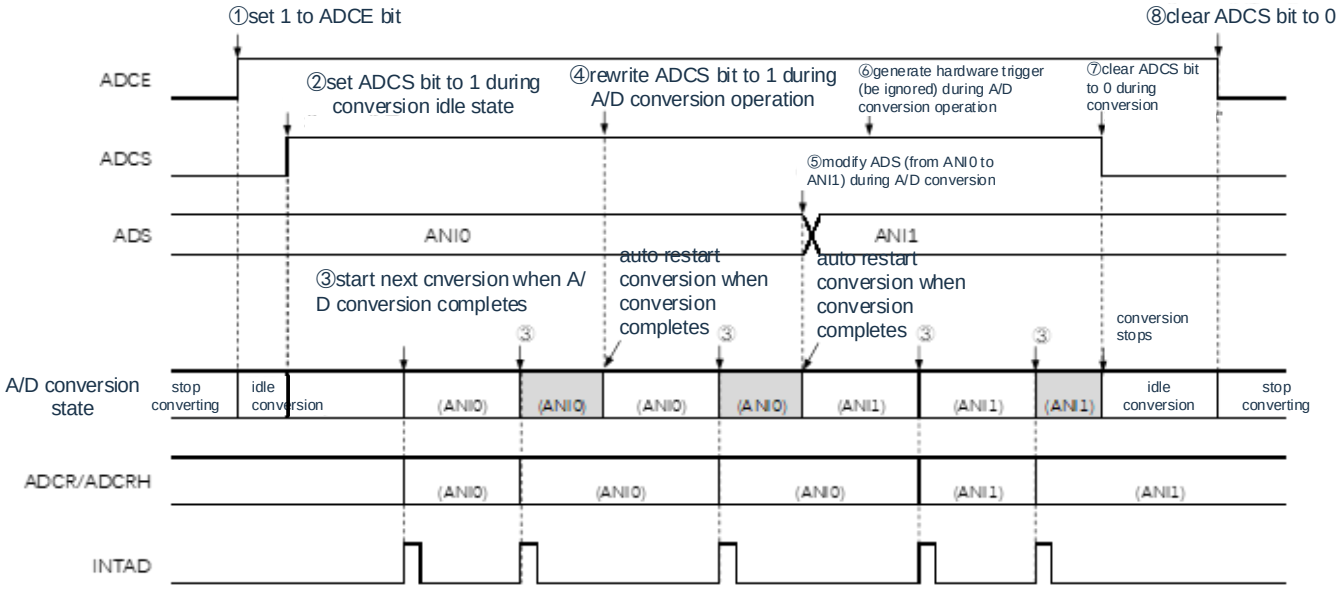
⑤ If the ADS register is rewritten or rewritten during the conversion, the current A/D conversion is immediately aborted and then A/D is converted.

⑥ An A/D conversion does not start even if a hardware trigger is entered during the conversion.

⑦ If the ADCS location "0" is used during the conversion, the current A/D conversion is immediately aborted and then A/D standby.

⑧ If the ADCE position is "0" in the A/D transition standby state, the A/D converter goes to stop. When the ADCE bit is '0', even the ADCS position '1' is ignored and no A/D conversion begins.

Figure 15-17 Example of runtime order for software trigger mode (select mode, continuous conversion mode)



15.4.2 Software Trigger Mode (Select Mode, Single Conversion Mode)

① In the stopped state, the ADCE position "1" of mode register 0 (ADM0) of the A/D converter is switched into standby.

② After the stable waiting time (1μs) is counted by software, the ADM0 register ADCS position "1" is A/D converted to the analog input specified by the analog input channel assignment register (ADS).

③ If the A/D conversion ends, the conversion result is saved to the A/D conversion result register (ADCR, ADCRH) and the A/D conversion end interrupt request signal (INTAD).

④ After the A/D conversion, the ADCS bit automatically clears "0" and enters the A/D conversion standby state.

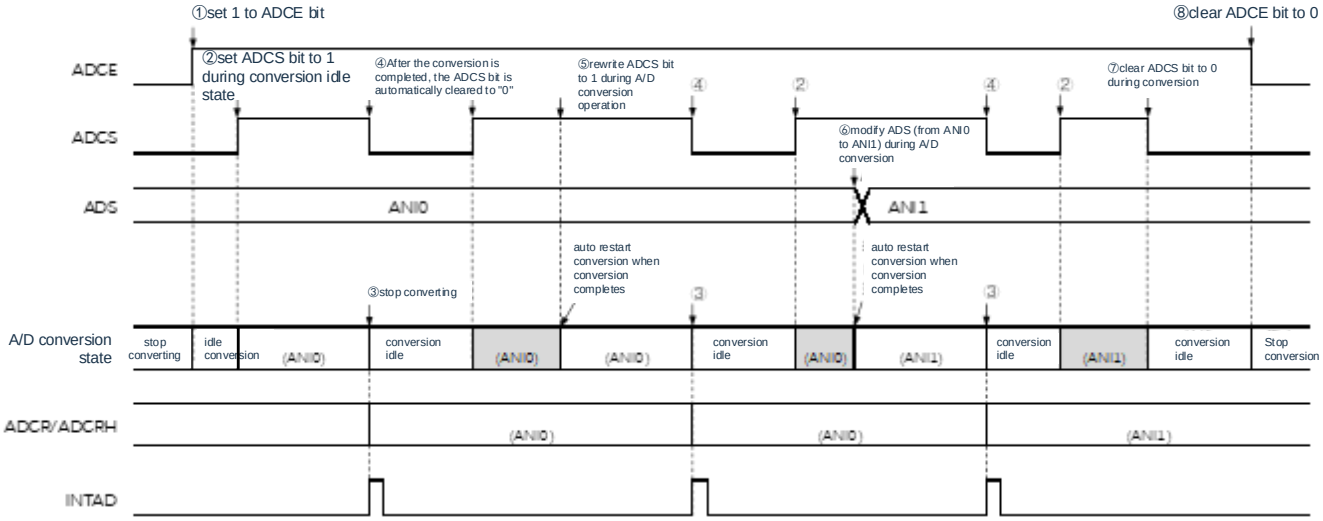
⑤ If the ADCS bit is overridden "1" during the conversion, the current A/D conversion immediately aborts and starts again.

⑥ If the ADS register is rewritten or rewritten during the conversion, the current A/D conversion is immediately aborted and then A/D is converted.

⑦ If the ADCS location "0" is used during the conversion, the current A/D conversion is immediately aborted and then A/D standby.

⑧ If the ADCE position is "0" in the A/D transition standby state, the A/D converter goes to stop. When the ADCE bit is '0', even the ADCS position '1' is ignored and no A/D conversion begins. The A/D conversion does not start even if the hardware trigger is entered while the A/D conversion is standby.

Figure 15-18 Runtime Sequence Example of Software Trigger Mode (Select Mode, Single Conversion Mode)



15.4.3 Software Trigger Mode (Scan Mode, Continuous Conversion Mode)

① In the stopped state, the ADCE position "1" of mode register 0 (ADM0) of the A/D converter is switched into standby.

② After the stable waiting time (1μs) is counted by software, the ADCS position of the ADM0 register is 1 to perform A/D conversion. The A/D conversion is performed from the analog input channel specified by the scan 0.

③ A/D conversion of 4 analog input channels is performed. Each time the A/D conversion ends, the conversion result is saved to the A/D conversion result register ADCR, ADCRH, and the A/D conversion end interrupt request signal is INTAD. The next A/D conversion (4 channels) automatically starts from the set channel immediately after the A/D conversion of the 4 channels is completed.

④ If the ADCS bit is overridden "1" during the conversion, the current A/D conversion immediately aborts and starts again.

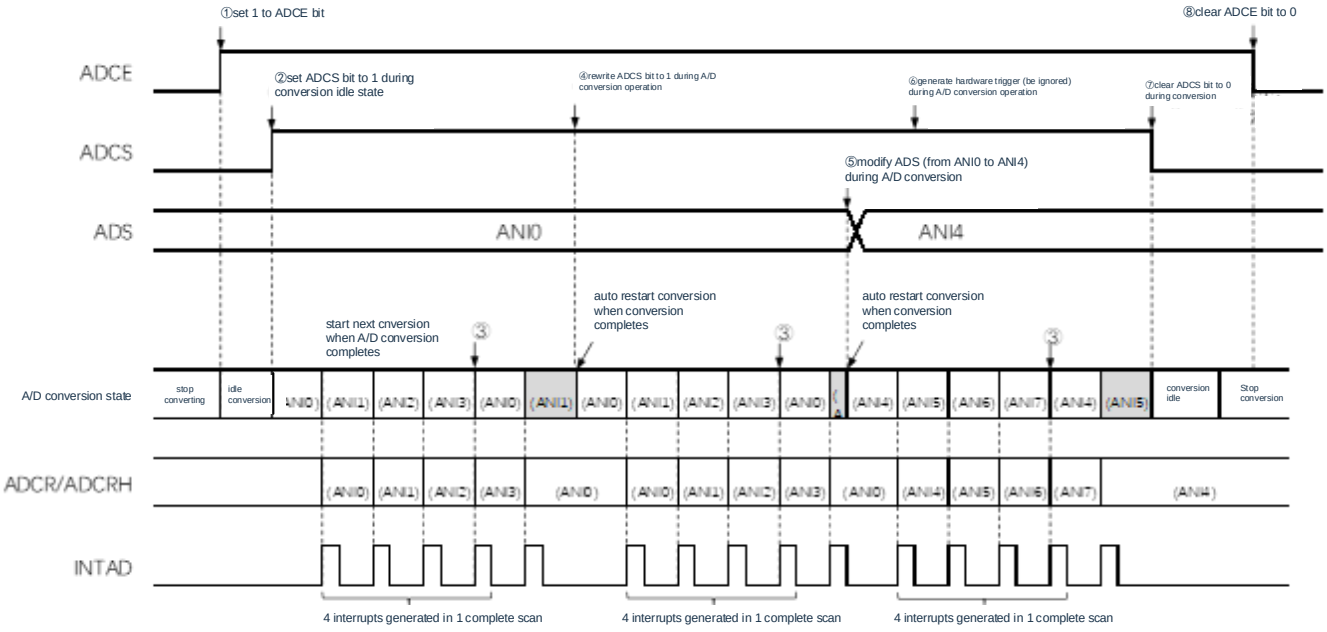
⑤ If the ADS register is rewritten or rewritten during the conversion, the current A/D conversion is immediately aborted and then A/D from the initial channel redesignated.

⑥ An A/D conversion does not start even if a hardware trigger is entered during the conversion.

⑦ If the ADCS location "0" is used during the conversion, the current A/D conversion is immediately aborted and then A/D standby.

⑧ If the ADCE position is "0" in the A/D transition standby state, the A/D converter goes to stop. When the ADCE bit is '0', even the ADCS position '1' is ignored and no A/D conversion begins.

Figure 15-19 Example of runtime order for software trigger mode (scan mode, continuous conversion mode)



15.4.4 Software Trigger Mode (Scan Mode, Single Conversion Mode)

① In the stopped state, the ADCE position "1" of mode register 0 (ADM0) of the A/D converter is switched into standby.

② After the stable waiting time (1μs) is counted by software, the ADM0 register ADCS position "1" is A/D for 4 analog input channels. The A/D conversion is performed from the analog input channel specified by the scan 0.

③ A/D conversion of 4 analog input channels is performed. Each time the A/D conversion ends, the conversion result is saved to the A/D conversion result register ADCR, ADCRH, and the A/D conversion end interrupt request signal is INTAD.

④ The ADCS bit automatically clears "0" after the A/D conversion of the 4 channels and enters the A/D conversion standby.

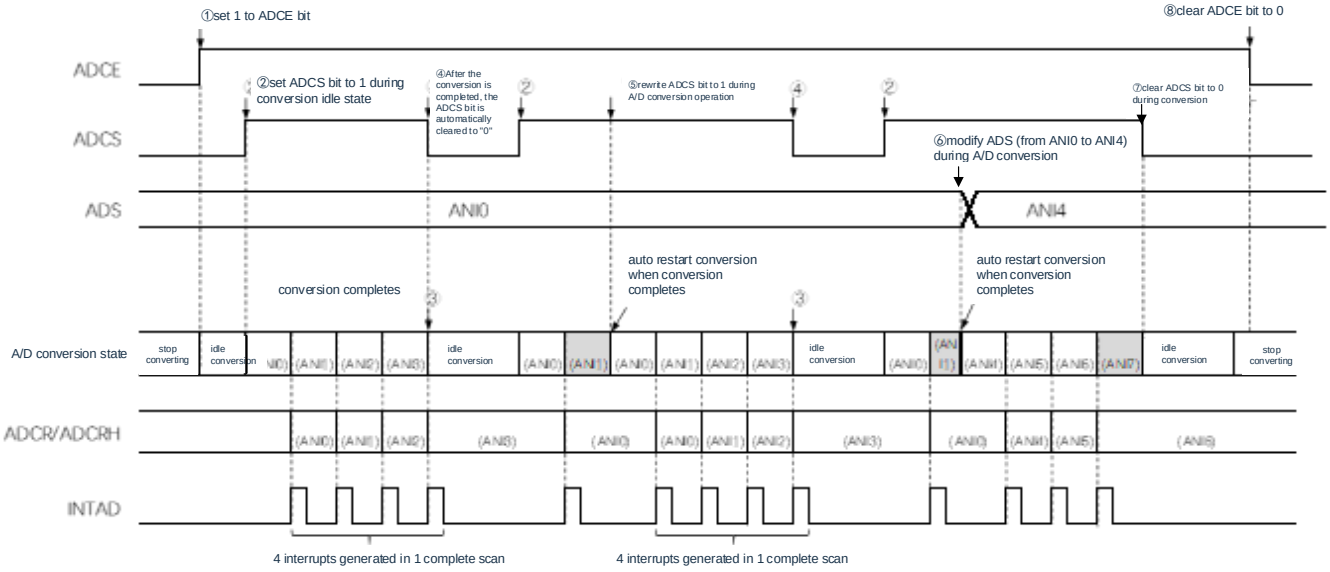
⑤ If the ADCS bit is overridden "1" during the conversion, the current A/D conversion immediately aborts and starts again.

⑥ If the ADS register is rewritten or rewritten during the conversion, the current A/D conversion is immediately aborted and then A/D from the initial channel redesignated.

⑦ If the ADCS location "0" is used during the conversion, the current A/D conversion is immediately aborted and then A/D standby.

⑧ If the ADCE position is "0" in the A/D transition standby state, the A/D converter goes to stop. When the ADCE bit is '0', even the ADCS position '1' is ignored and no A/D conversion begins. The A/D conversion does not start even if the hardware trigger is entered while the A/D conversion is standby.

Figure 15-20 Example of runtime order for software trigger mode (scan mode, Single Conversion Mode)



15.4.5 Hardware triggers no-wait mode (select mode, continuous conversion mode)

① In the stopped state, the ADCE position "1" of mode register 0 (ADM0) of the A/D converter is switched into standby.

② After the stable waiting time (1μs) is counted by the software, the ADCS position "1" of the ADM0 register enters the hardware triggered standby state (this phase does not start switching). The A/D conversion does not start even with the ADCS position "1" when the hardware triggers the standby state.

③ If a hardware trigger is entered with the ADCS bit '1', an A/D conversion is performed on analog inputs specified by analog input channel assignment registers.

④ If the A/D conversion ends, the conversion result is saved to the A/D conversion result register (ADCR, ADCRH) and the A/D conversion end interrupt request signal (INTAD). The next A/D conversion begins immediately after the A/D conversion.

⑤ If you enter a hardware trigger during the conversion, the current A/D conversion is immediately aborted and then restarted.

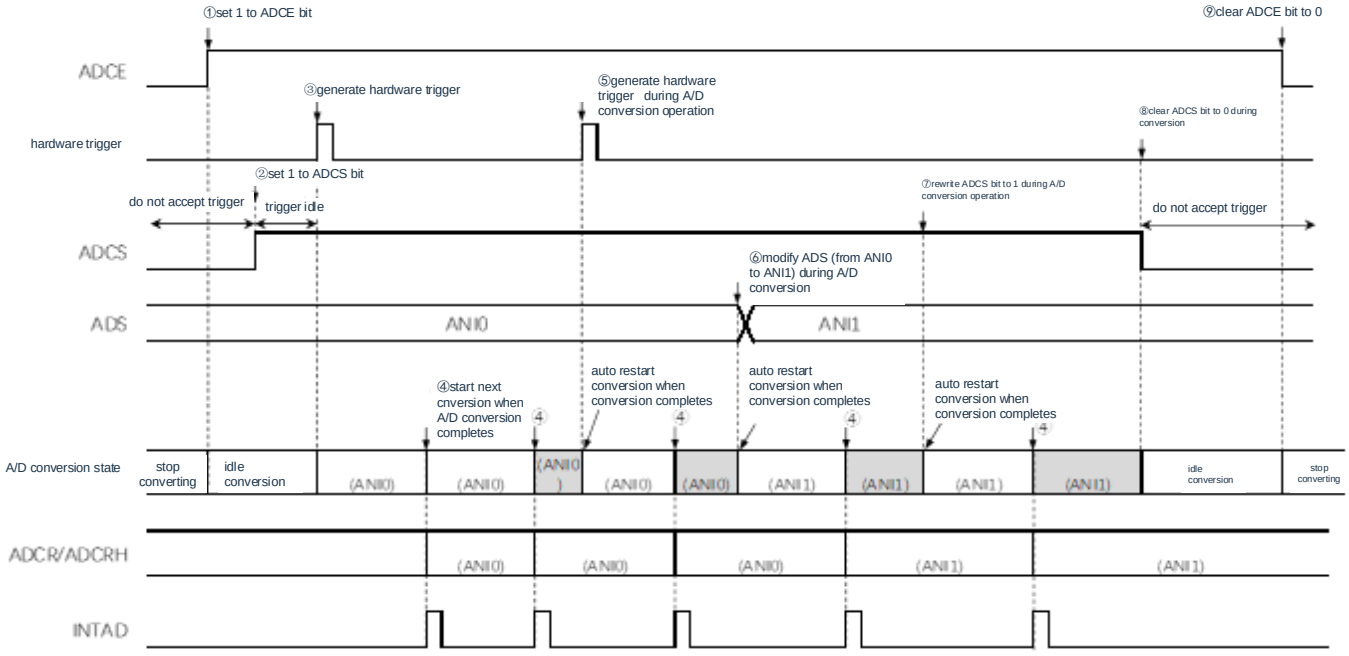
⑥ If the ADS register is rewritten or rewritten during the conversion, the current A/D conversion is immediately aborted and then A/D is converted.

⑦ If the ADCS bit is overridden "1" during the conversion, the current A/D conversion immediately aborts and starts again.

⑧ If the ADCS location "0" is used during the conversion, the current A/D conversion is immediately aborted and then A/D standby. However, in this state, the A/D converter does not enter a stop state.

⑨ If the ADCE position is "0" in the A/D transition standby state, the A/D converter goes to stop. When the ADCS bit is "0", even the input hardware trigger is ignored and A/D conversion does not start.

Figure 15-21 Hardware-triggered runtime sequence example of no-wait mode (select mode, continuous conversion mode)



15.4.6 Hardware triggers no-wait mode (select mode, Single Conversion Mode)

① In the stopped state, the ADCE position "1" of mode register 0 (ADM0) of the A/D converter is switched into standby.

② After the stable waiting time (1μs) is counted by the software, the ADCS position "1" of the ADM0 register enters the hardware triggered standby state (this phase does not start switching). The A/D conversion does not start even with the ADCS position "1" when the hardware triggers the standby state.

③ If a hardware trigger is entered with the ADCS bit '1', an A/D conversion is performed on analog inputs specified by analog input channel assignment registers.

④ If the A/D conversion ends, the conversion result is saved to the A/D conversion result register (ADCR, ADCRH) and the A/D conversion end interrupt request signal (INTAD).

⑤ After the A/D conversion is complete, the ADCS bit remains in the "1" state and enters A/D.

⑥ If you enter a hardware trigger during the conversion, the current A/D conversion is immediately aborted and then restarted.

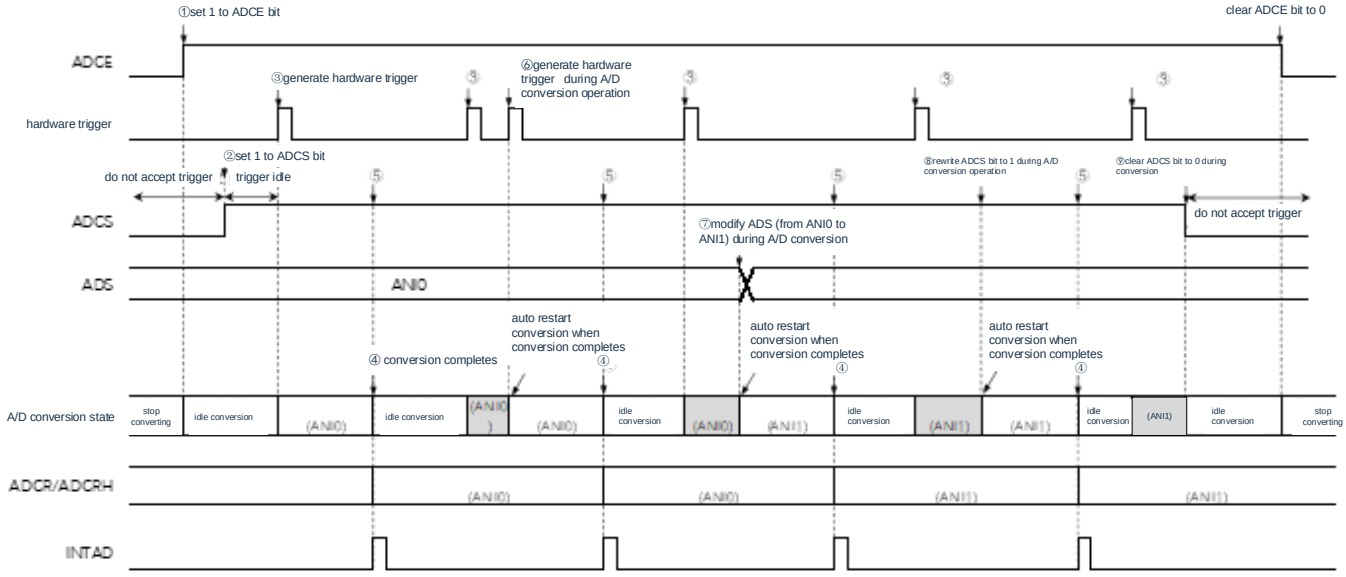
⑦ If the ADS register is rewritten or rewritten during the conversion, the current A/D conversion is immediately aborted and then A/D is converted.

⑧ If the ADCS bit is overridden "1" during the conversion, the current A/D conversion immediately aborts and starts again.

⑨ If the ADCS location "0" is used during the conversion, the current A/D conversion stops immediately and enters the A/D conversion standby. However, in this state, the A/D converter does not enter a stop state.

⑩ If the ADCE position is "0" in the A/D transition standby state, the A/D converter goes to stop. When the ADCS bit is "0", even the input hardware trigger is ignored and A/D conversion does not start.

Figure 15-22 Hardware-triggered runtime sequence example of no-wait mode (select mode, Single Conversion Mode)



15.4.7 Hardware triggered no-wait mode (scan mode, continuous conversion mode)

① In the stopped state, the ADCE position "1" of mode register 0 (ADM0) of the A/D converter is switched into standby.

② After the stable waiting time (1μs) is counted by the software, the ADCS position "1" of the ADM0 register enters the hardware triggered standby state (this phase does not start switching). The A/D conversion does not start even with the ADCS position "1" when the hardware triggers the standby state.

③ If a hardware trigger is entered in a state with ADCS bit '1', A/D conversion is performed on 4 analog input channels specified by analog input channel assignment register (ADS). The A/D conversion is performed from the analog input channel specified by the scan 0.

④ A/D conversion of 4 analog input channels is performed. Each time the A/D conversion ends, the conversion result is saved to the A/D conversion result register ADCR, ADCRH, and the A/D conversion end interrupt request signal is INTAD. The next A/D conversion is automatically started from the set channel immediately after the A/D conversion of the 4 channels is completed.

⑤ If a hardware trigger is entered during the conversion, the current A/D conversion is aborted immediately and the conversion restarts from the original channel.

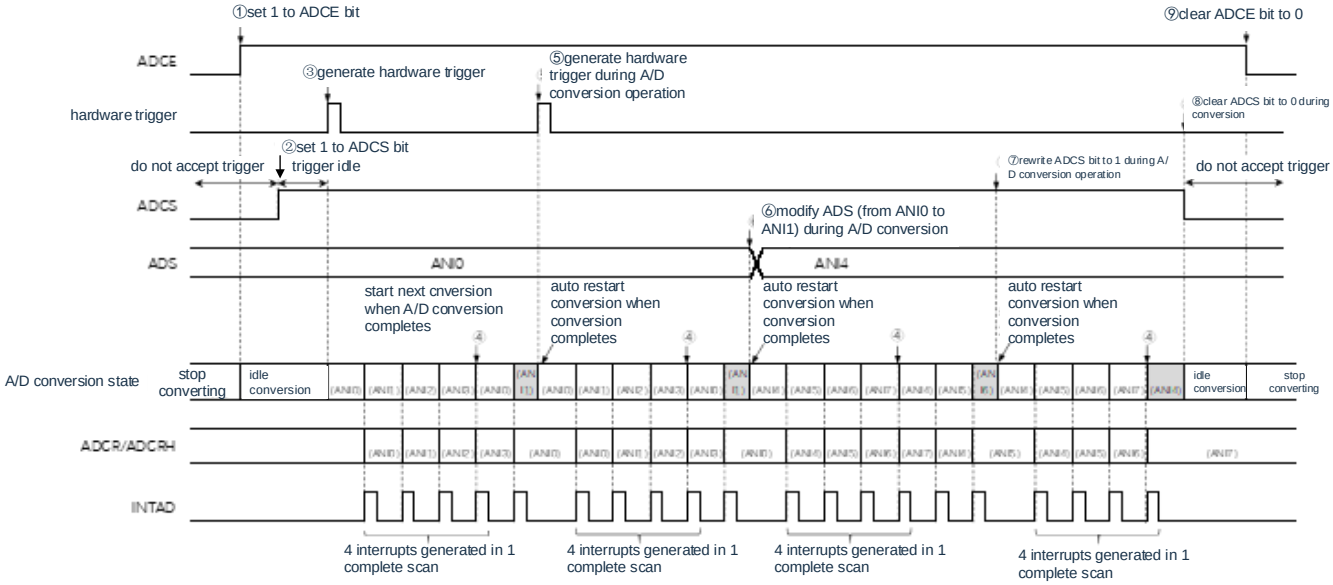
⑥ If the ADS register is rewritten or rewritten during the conversion, the current A/D conversion is immediately aborted and then A/D converted.

⑦ If the ADCS bit is overridden "1" during the conversion, the current A/D conversion immediately aborts and restarts from the original channel.

⑧ If the ADCS location "0" is used during the conversion, the current A/D conversion is immediately aborted and then A/D standby. However, in this state, the A/D converter does not enter a stop state.

⑨ If the ADCE position is "0" in the A/D transition standby state, the A/D converter goes to stop. When the ADCE bit is '0', even the ADCS position '1' is ignored and no A/D conversion begins.

Figure 15-23 Hardware-triggered runtime sequence example of no-wait mode (scan mode, continuous conversion mode)



15.4.8 Hardware triggered no-wait mode (scan mode, Single Conversion Mode)

① In the stopped state, the ADCE position "1" of mode register 0 (ADM0) of the A/D converter is switched into standby.

② After the stable waiting time (1μs) is counted by the software, the ADCS position "1" of the ADM0 register enters the hardware triggered standby state (this phase does not start switching). The A/D conversion does not start even with the ADCS position "1" when the hardware triggers the standby state.

③ If a hardware trigger is entered in a state with ADCS bit '1', A/D conversion is performed on 4 analog input channels specified by analog input channel assignment register (ADS). The A/D conversion is performed from the analog input channel specified by the scan 0.

④ A/D conversion of 4 analog input channels is performed. Each time the A/D conversion ends, the conversion result is saved to the A/D conversion result register ADCR, ADCRH, and the A/D conversion end interrupt request signal is INTAD.

⑤ The ADCS bit remains in the "1" state after the A/D conversion of the 4 channels is completed and enters the A/D conversion standby state.

⑥ If a hardware trigger is entered during the conversion, the current A/D conversion is aborted immediately and the conversion restarts from the original channel.

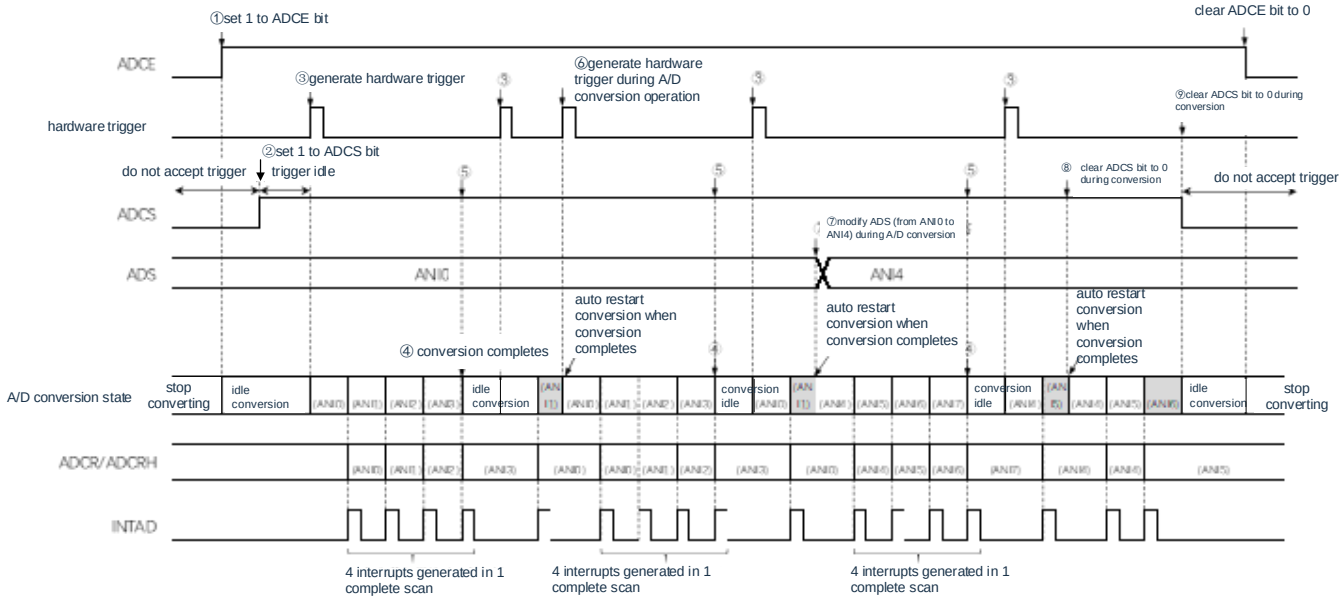
⑦ If the ADS register is rewritten or rewritten during the conversion, the current A/D conversion is immediately aborted and then A/D from the initial channel redesignated.

⑧ If the ADCS bit is overridden "1" during the conversion, the current A/D conversion is aborted immediately and then reconverted from the original channel.

⑨ If the ADCS location "0" is used during the conversion, the current A/D conversion is immediately aborted and then A/D standby. However, in this state, the A/D converter does not enter a stop state.

⑩ If the ADCE position is "0" in the A/D transition standby state, the A/D converter goes to stop. When the ADCS bit is "0", even the input hardware trigger is ignored and A/D conversion does not start.

Figure 15-24 Hardware-triggered runtime sequence example of no-wait mode (scan mode, single-pass conversion mode)



15.4.9 Hardware triggered wait mode (select mode, continuous conversion mode)

① In the stop state, the ADCE position of mode register 0 (ADM0) of the A/D converter enters the hardware triggered standby state.

② If a hardware trigger is input in a hardware trigger standby state, an A/D conversion is performed to the analog input specified by an analog input channel assignment register (ADS). Automatically "1" the ADM0 register's ADCS position while input hardware is triggered.

③ If the A/D conversion ends, the conversion result is saved to the A/D conversion result register (ADCR, ADCRH) and the A/D conversion end interrupt request signal (INTAD). The next A/D conversion starts immediately after the A/D conversion (at which time no hardware trigger is required).

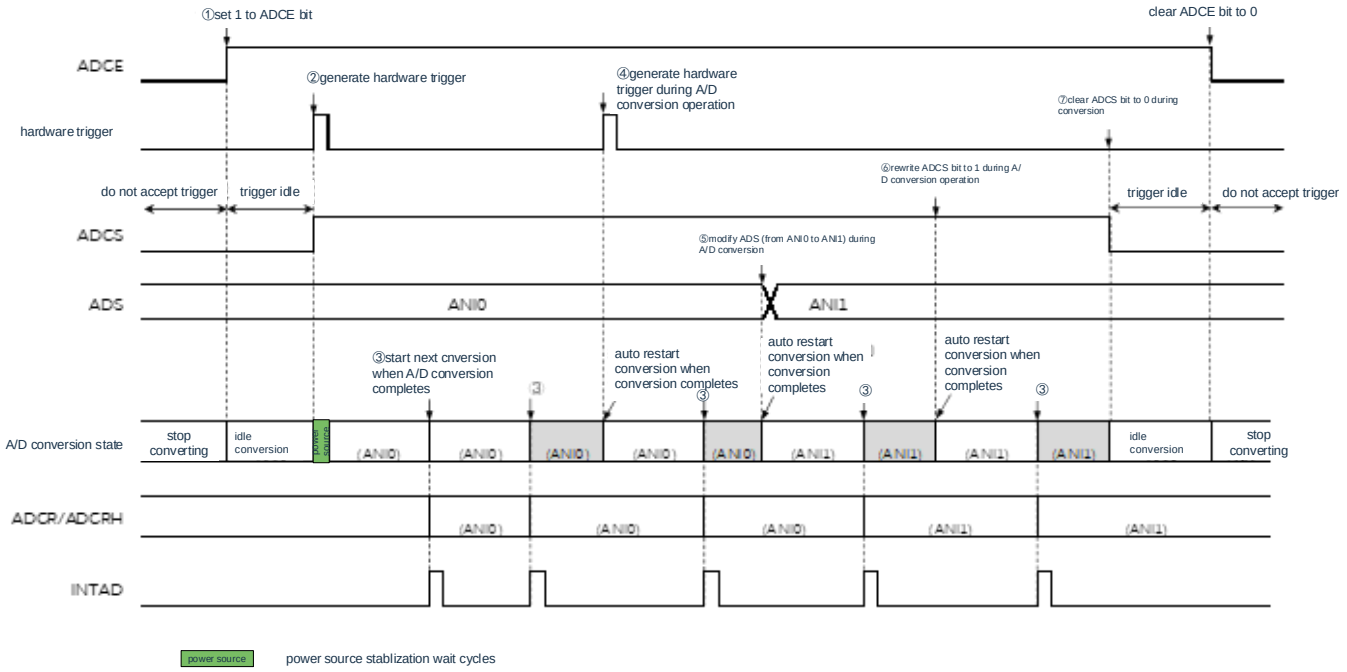
④ If you enter a hardware trigger during the conversion, the current A/D conversion is immediately aborted and then restarted.

⑤ If the ADS register is rewritten or rewritten during the conversion, the current A/D conversion is immediately aborted and then A/D is converted.

⑥ If the ADCS bit is overridden "1" during the conversion, the current A/D conversion immediately aborts and starts again.

⑦ If the ADCS position "0" is used during the conversion, the current A/D conversion is immediately aborted and then hardware triggered standby. When the ADCE bit is "0", even the input hardware trigger is ignored and A/D conversion does not start.

Figure 15-25 Runtime sequence example of hardware triggered wait mode (select mode, continuous conversion mode)



15.4.10 Hardware triggered wait mode (select mode, Single Conversion Mode)

① In the stop state, the ADCE position of mode register 0 (ADM0) of the A/D converter enters the hardware triggered standby state.

② If a hardware trigger is input in a hardware trigger standby state, an A/D conversion is performed to the analog input specified by an analog input channel assignment register (ADS). Automatically "1" the ADM0 register's ADCS position while input hardware is triggered.

③ If the A/D conversion ends, the conversion result is saved to the A/D conversion result register (ADCR, ADCRH) and the A/D conversion end interrupt request signal (INTAD).

④ After the A/D conversion is completed, the ADCS bit automatically clears "0", and the A/D converter goes to stop.

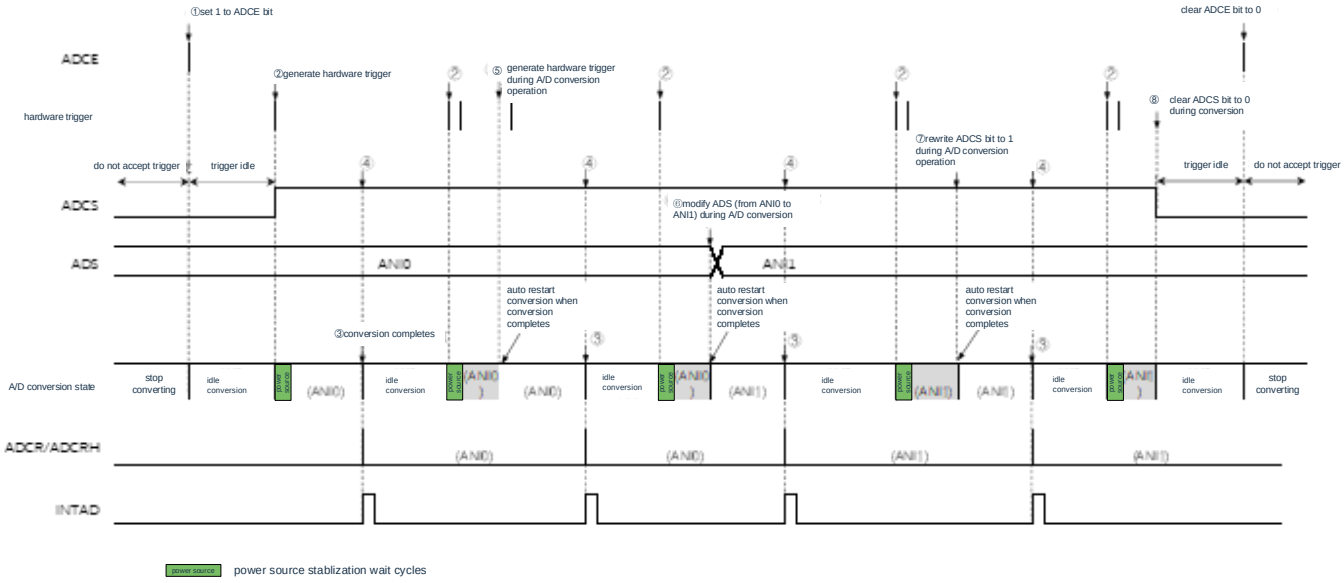
⑤ If you enter a hardware trigger during the conversion, the current A/D conversion is immediately aborted and then restarted.

⑥ If the ADS register is rewritten or rewritten during the conversion, the current A/D conversion is immediately aborted and then A/D is converted.

⑦ If the ADCS bit is overridden "1" during the conversion, the current A/D conversion immediately aborts and starts again.

⑧ If the ADCS position "0" is used during the conversion, the current A/D conversion is immediately aborted and then hardware triggered standby. When the ADCE bit is "0", even the input hardware trigger is ignored and A/D conversion does not start.

Figure 15-26 Runtime sequence example of hardware-triggered wait mode (select mode, Single Conversion Mode)



15.4.11 Hardware triggered wait mode (scan mode, continuous conversion mode)

① In the stop state, the ADCE position of mode register 0 (ADM0) of the A/D converter enters the hardware triggered standby state.

② If a hardware trigger is input in a hardware trigger standby state, an A/D conversion is performed on 4 analog input channels. Automatically "1" the ADM0 register's ADCS position while input hardware is triggered. The A/D conversion is performed from the analog input channel specified by the scan 0.

③ A/D conversion of 4 analog input channels is performed. Each time the A/D conversion ends, the conversion result is saved to the A/D conversion result register ADCR, ADCRH, and the A/D conversion end interrupt request signal is INTAD. The next A/D conversion is automatically started from the set channel immediately after the A/D conversion of the 4 channels is completed.

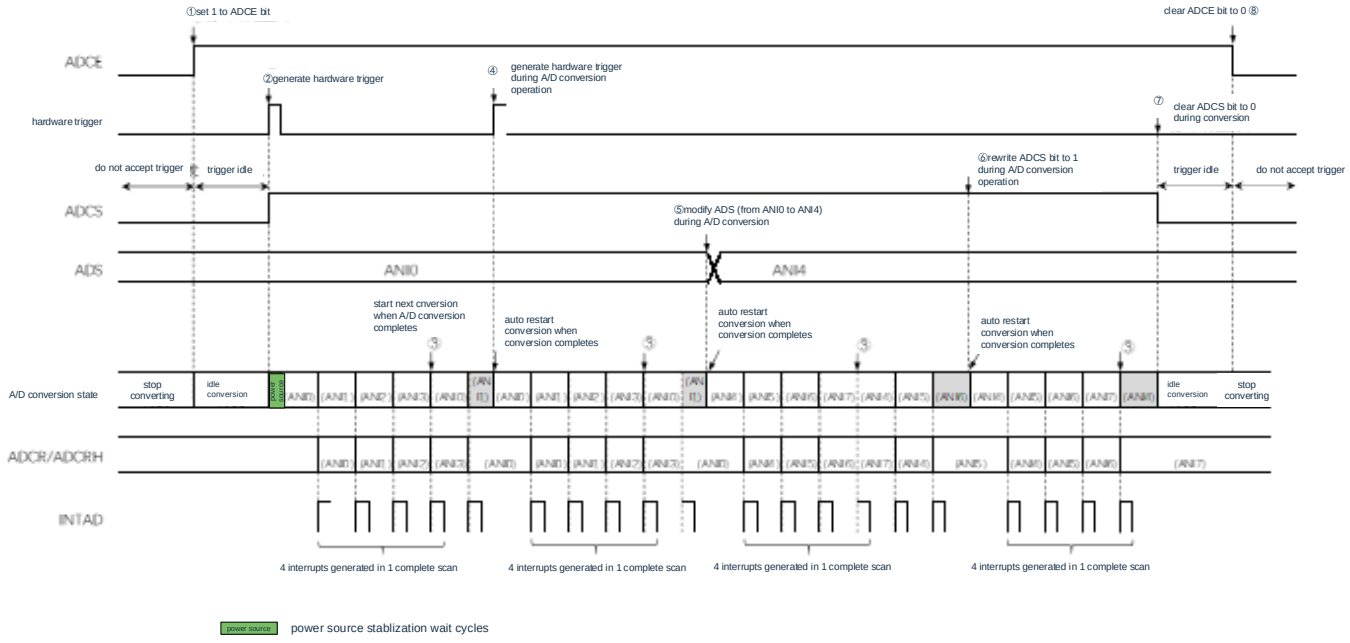
④ If a hardware trigger is entered during the conversion, the current A/D conversion is aborted immediately and the conversion restarts from the original channel.

⑤ If the ADS register is overridden or rewritten during the conversion, the current A/D conversion is immediately aborted and then scanned from the channel redesignated by the ADS register.

⑥ If the ADCS bit is overridden "1" during the conversion, the current A/D conversion immediately aborts and restarts from the original channel.

⑦ If the ADCS position "0" is used during the conversion, the current A/D conversion is immediately aborted and then hardware triggered standby. When the ADCE bit is "0", even the input hardware trigger is ignored and A/D conversion does not start.

Figure 15-27 Runtime sequence example of hardware triggered wait mode (scan mode, continuous conversion mode)



15.4.12 Hardware triggered wait mode (scan mode, Single Conversion Mode)

① In the stop state, the ADCE position of mode register 0 (ADM0) of the A/D converter enters the hardware triggered standby state.

② If a hardware trigger is input in a hardware trigger standby state, an A/D conversion is performed on 4 analog input channels. The ADCS location of the ADM0 register is "1" automatically after the input hardware trigger. The A/D conversion is performed from the analog input channel specified by the scan 0.

③ A/D conversion of 4 analog input channels is performed. Each time the A/D conversion ends, the conversion result is saved to the A/D conversion result register ADCR, ADCRH, and the A/D conversion end interrupt request signal is INTAD.

④ After the A/D conversion is completed, the ADCS bit automatically clears "0", and the A/D converter goes to stop.

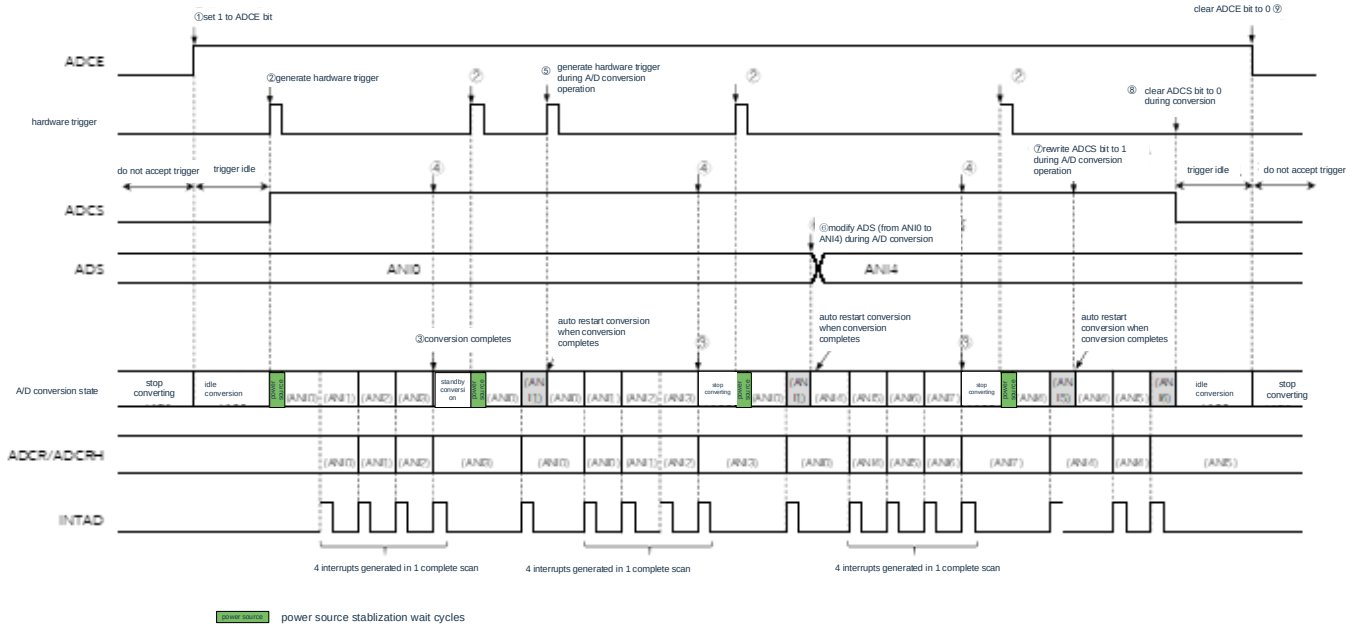
⑤ If a hardware trigger is entered during the conversion, the current A/D conversion is immediately aborted and the conversion is rescanned from the original channel.

⑥ If the ADS register is overridden or rewritten during the conversion, the current A/D conversion is immediately aborted and then scanned from the channel redesignated by the ADS register.

⑦ If the ADCS bit is rewritten "1" during the conversion, the current A/D conversion is aborted immediately and the conversion is scanned from the original channel.

⑧ If the ADCS position "0" is used during the conversion, the current A/D conversion is immediately aborted and then hardware triggered standby. When the ADCE bit is "0", even the input hardware trigger is ignored and A/D conversion does not start.

Figure 15-28 Runtime sequence example of hardware-triggered wait mode (scan mode, Single Conversion Mode)

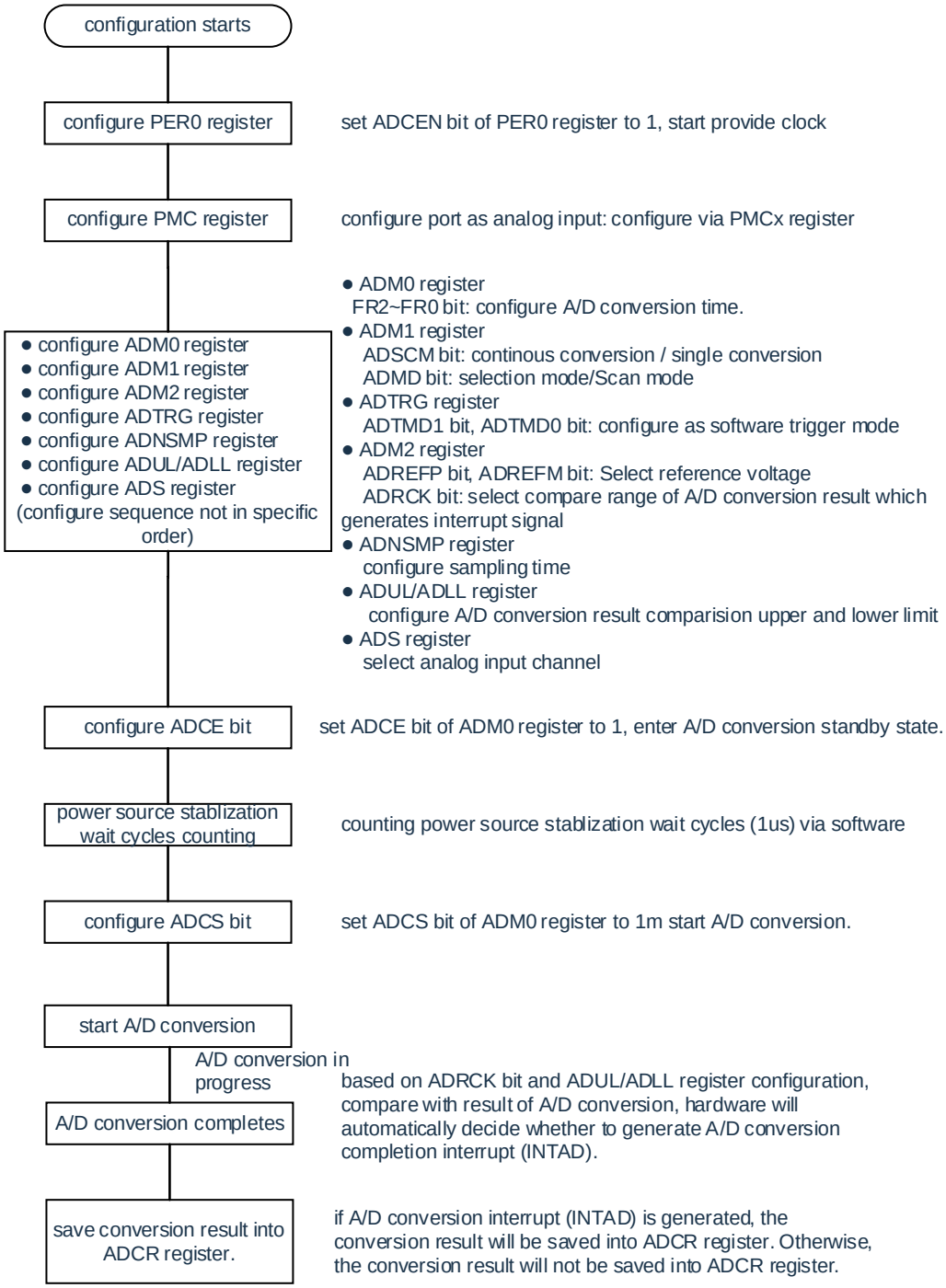


15.5 Converter Set-up Flowchart

The set-up flowchart of the A/D converters for each mode of operation is shown below.

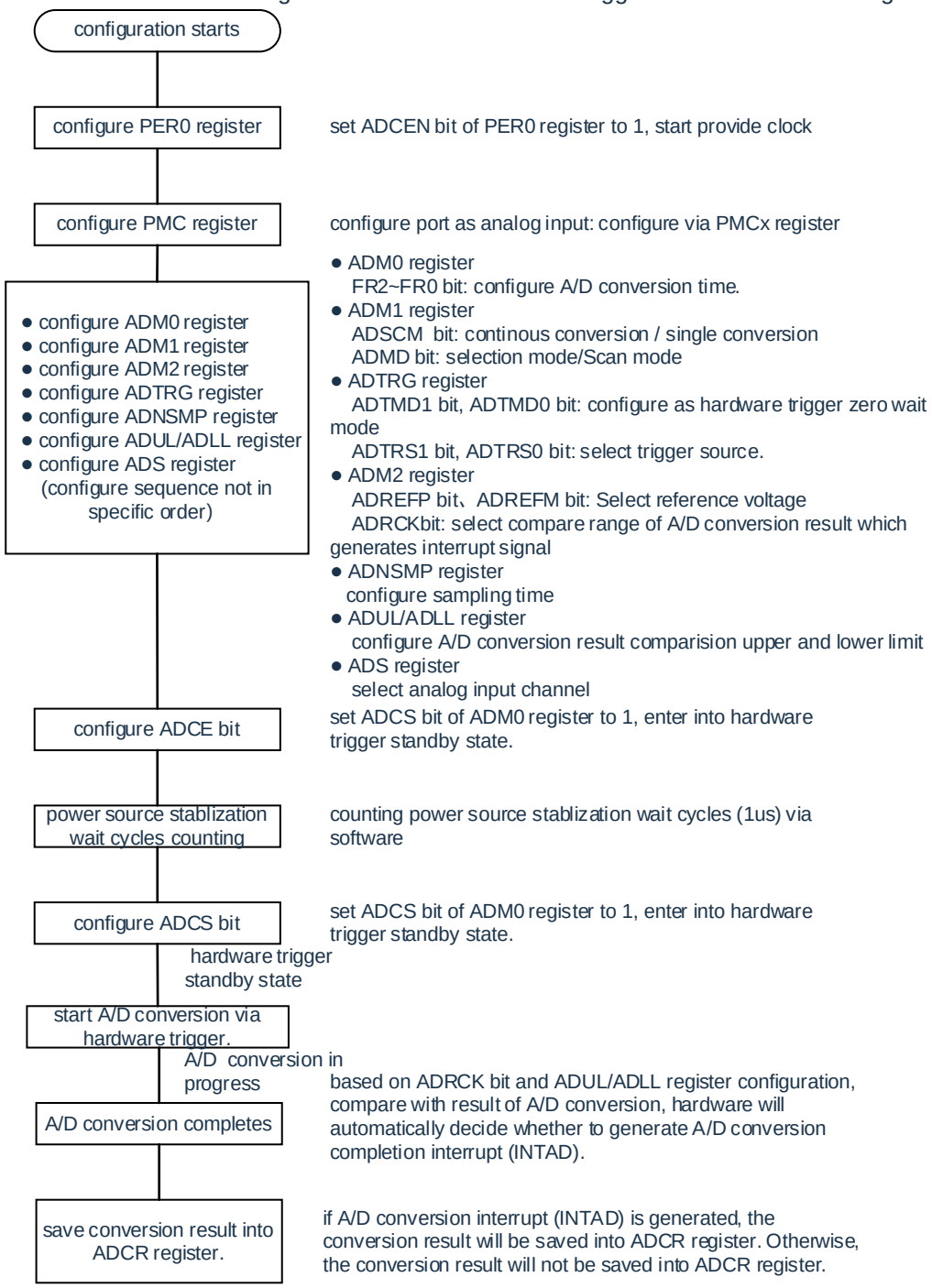
15.5.1 Settings for software trigger mode

Figure 15-30 Settings for software trigger mode



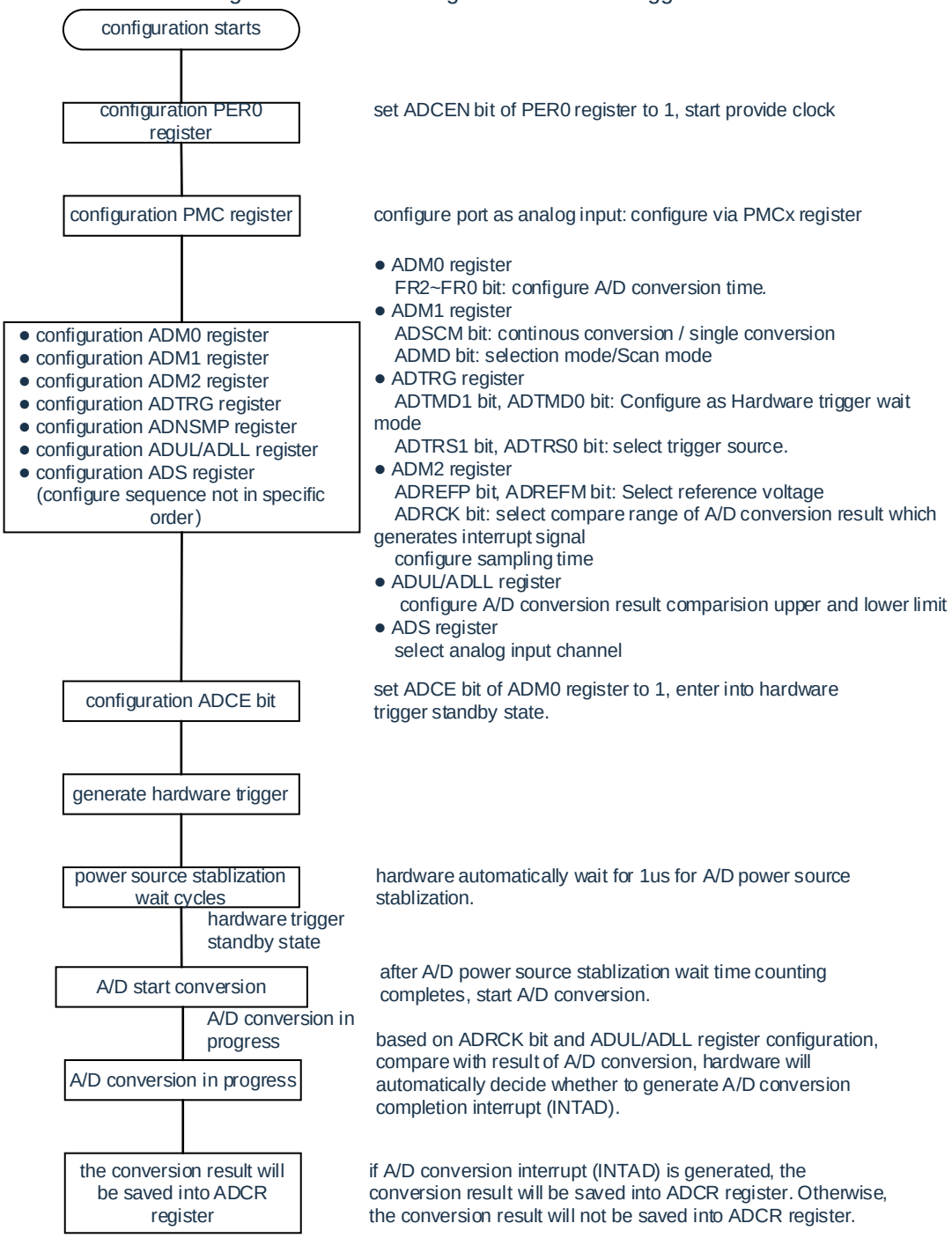
15.5.2 Hardware Trigger No Wait Mode Settings

Figure 15-31 Hardware Trigger No Wait Mode Settings



15.5.3 Settings for Hardware Trigger Wait Mode

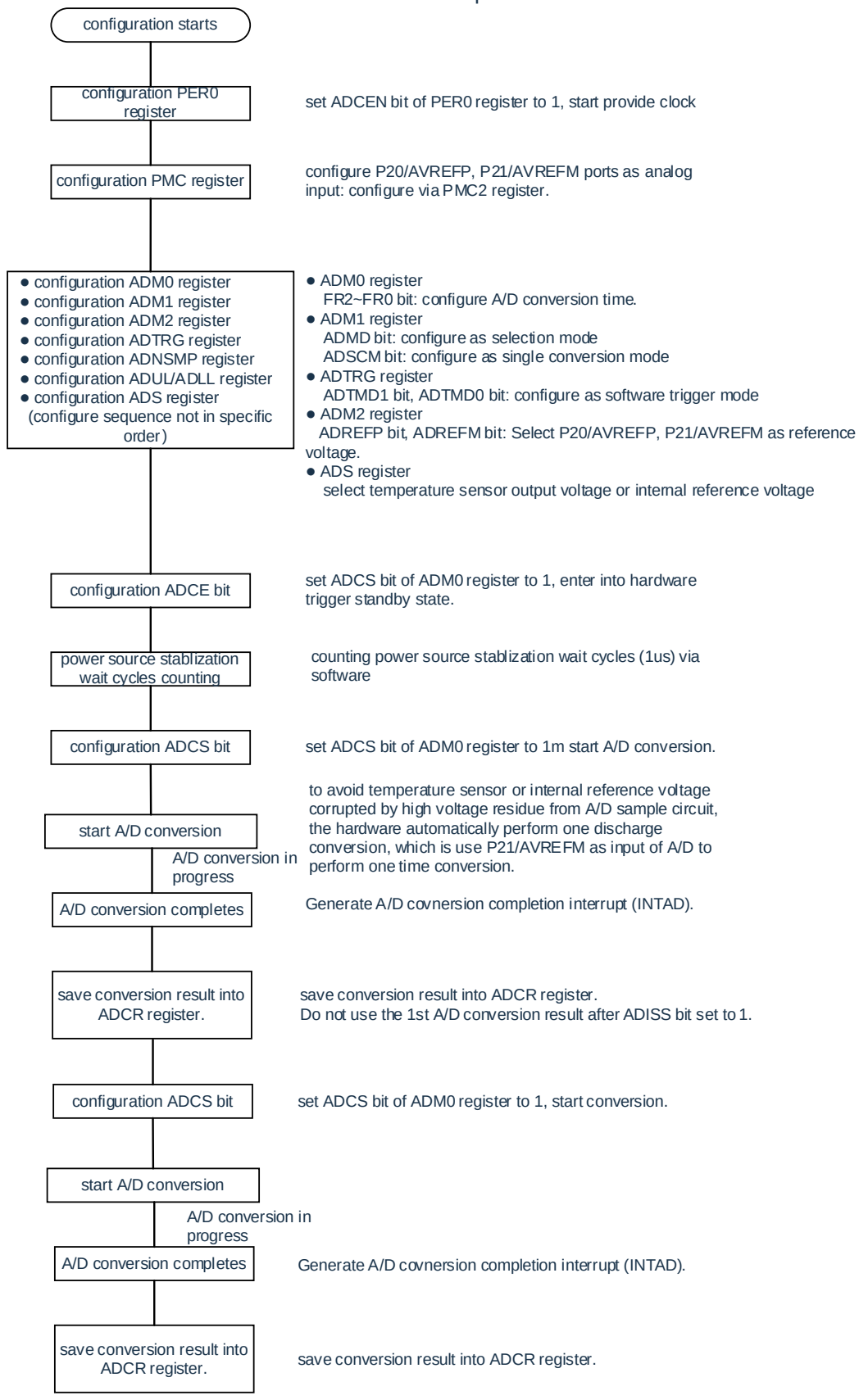
Figure 15-32 Settings for Hardware Trigger Wait Mode



15.5.4 Settings when selecting the output voltage/internal reference voltage of the temperature sensor

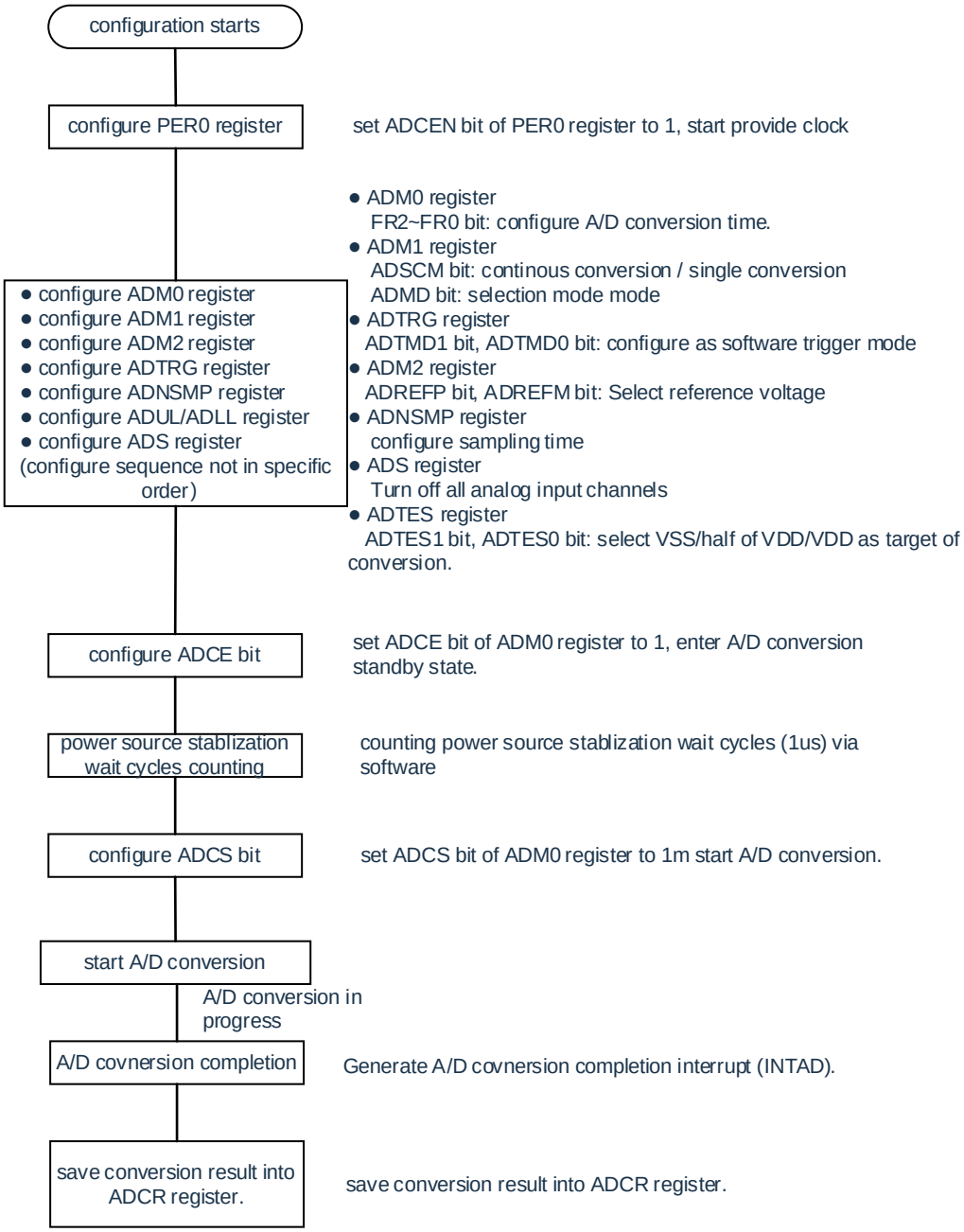
(Take software trigger mode, single conversion mode for example)

Figure 15-33 Settings when selecting the output voltage/internal reference voltage of the temperature sensor



15.5.5 Settings for test mode

Figure 15-34 Settings for the test mode (VSS/half_VDD/VDD as the translation object)



Chapter 16 D/A Converter

The channels of the D/A converter vary depending on the product.

Table 1 6-1 Output pin of D/A converter

D/A output pin	64PIN	52 PIN	48PIN	40 PIN	36PIN	32 PIN	24PIN
ANO0	○	○	○	○	○	○	○
ANO1	○	○	○	○	○	○	—

Note: The following sections in this chapter are intended for the 64 pin products.

16.1 Function of D/A Converter

D/A converter converts digital input into analog signal with 8-bit resolution, which controls analog output of two channels (ANO0).

The D/A converter has the following functions:

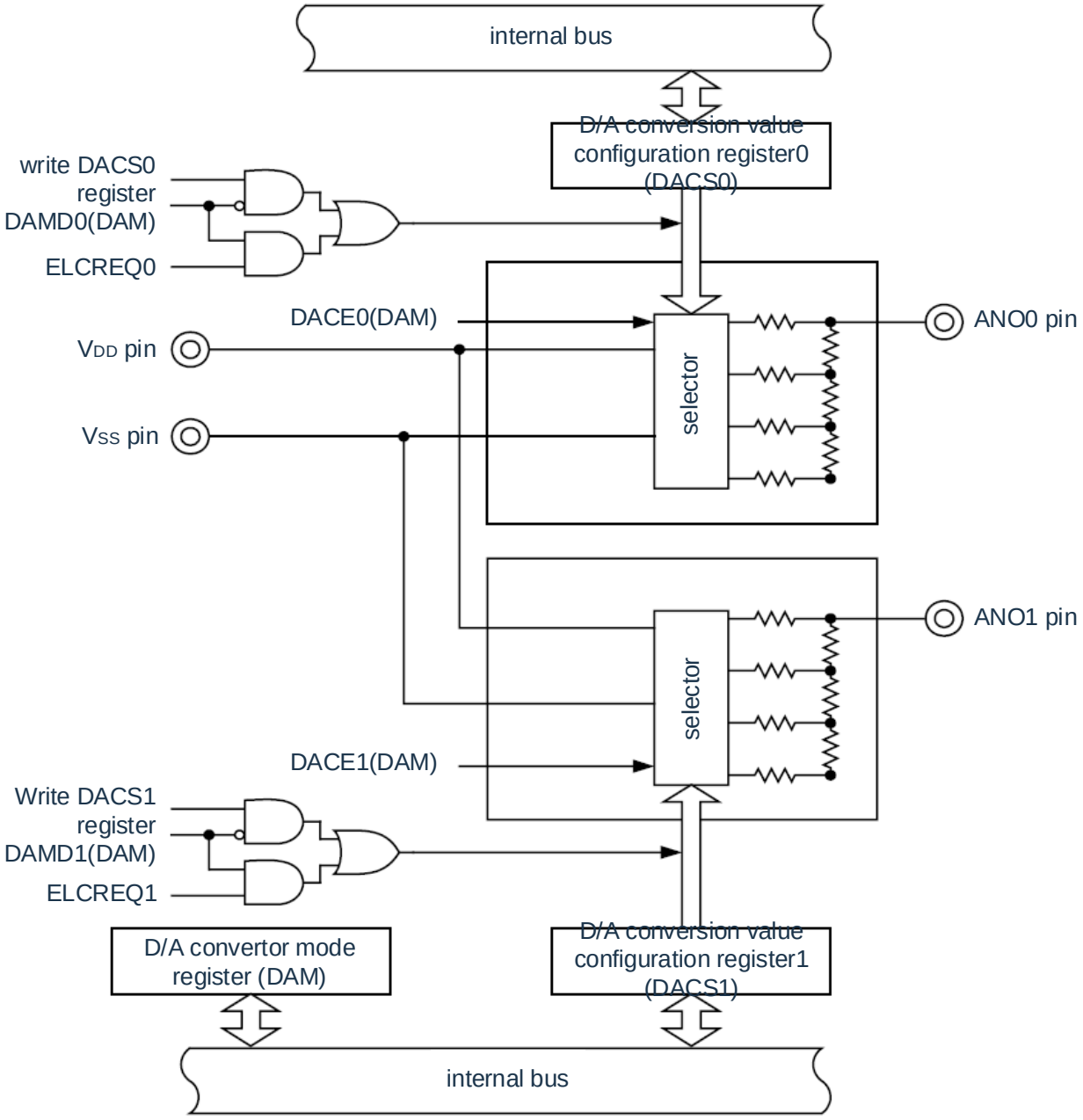
- 8-bit resolution 2ch
- R-2R ladder
- analog output voltage
  - 8-bit resolution:VDDm8/256 (m8: value set to DACSi register)
- operation mode
  - Normal Mode
  - real-time output mode

Notes: i=0,1

16.2 Structure of D/A converter

The D/A converter box is shown in Figure 16-1.

Figure 16-1 Block Diagram for D/A Converter



Note: ELCREQ0 and ELCREQ1 are trigger signals (event signals of EVENTC) for real-time output mode.

16.3 Register for controlling D/A converter

The D/A converter is controlled through the following registers.

- Peripheral Enable Register 1 (PER1)
- Mode register (DAM) for D/A converter.
- D/A conversion value setting registers 0,1 (DACS0,DACS1)
- event output target selection register n(ELSELRn), n=00~21

16.3.1 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.

When using the D/A converter, you must set bit7(DACEN) to "1.

The PER1 register is set by an 8-bit memory operation instruction.

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

Figure 16-2 Format of Peripheral Admission Register 1(PER1)

Address: 0x4002081A After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN	TMMEN	DMAEN	PWMPEN	TMCEN	TMAEN

DACEN	Control of Input Clock of D/A Converter
0	Stop provide an input clock. · You cannot write the SFR used by the D/A converter. · The D/A converter is in a reset state.
1	Provides an input clock. · Read and write SFRs used by D/A converters.

Note. : When you want to set up a D/A converter, you must first set the DACEN position "1".

When the DACEN bit is '0', writes to the control register of the D/A converter are ignored, except for the port mode register 2 (PM2P).

16.3.2 Mode register (DAM) for D/A converter.

This is the register that controls the D/A converter to run.
The DAM register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 16-3 Format of the Mode Register (DAM) for D/A Converter

Address: 40044736H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
DAM	0	0	DACE1	DACE0	0	0	DAMD1	DAMD0

DACEi	Control of Conversion Operation of D/A Converter
0	Stop the D/A conversion.
1	Allow D/A conversion.

DAMDi	The Choice of Running Mode of D/A Converter
0	Normal Run Mode
1	real-time output mode

Note: i=0,1

16.3.3 D/A conversion value setting register i(DACSi) (i=0,1)

This is a register that sets the analog voltage value that is output to the ANO0 pin and ANO1 pin when using the D/A converter. The DACSi register is set by an 8-bit memory operation instruction.
After the reset signal is generated, the values of these registers change to '00H'.

Figure 16-4 The D/A conversion value sets the format of register i(DACSi) (i=0,1)

Addresses: 40044734H (DACS0), 40044735H (DACS1) After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
DACSi	DACSi7	DACSi6	DACSi5	DACSi4	DACSi3	DACSi2	DACSi1	DACSi0

Note: The analog output voltage (VANOi) of the D/A converter is as follows:$VANOi=_{VDD}(DACSi)/256$

When the D/A converter is not used, the DACEi position "0" (output forbidden) and the DACSi register "00H" are required to reduce unnecessary consumption.

16.3.4 event output target selection register n(ELSELRn), n=00~21

When the real-time output mode of the D/A converter is used, the event signal of the event link controller is used as the start trigger for D/A conversion. Refer to " 24.3.1 Event Output Target Selection Register n (ELSELRn) (n=00~21)".

16.3.5 Register for controlling analog input pin port function

You must set up control registers (PMxx and PMC) for port functions that are multiplexed with the D/A converter analog output. Refer to "2.3.1 Port Mode Register (PMxx)" and "2.3.6 Port Mode Control Register (PMCxx)".

When using the ANO0 pin and the ANO1 pin as analog outputs for D/A converters, the port mode register (PMxx) location "1" for each port must be set to analog output via the port mode control register (PMC).

16.4 Operation of D/A Converter

16.4.1 Normal mode operation

The DACSi register is used as the start trigger to perform D/A conversion.

It is set as follows:

① Start supplying the D/A converter with the DACEN position "1" of the PER1 register (Peripheral Enable Register 1).

② Set the port to the analog pin through the PMC register (port mode control register).

③ Place the DAMDi position of the DAM register (mode register of D/A converter) in "0".

④ The DACSi register (D/A conversion value setting register i) sets the analog voltage value of the ANOi pin output.

The above (1) to (4) are the initial settings.

⑤ Place the DACEi location of the DAM register "1" (D/A conversion allowed).
The D/A conversion is started, and the analog voltage set by (4) is output to the ANOi pin after the settling time.

⑥ After that, the DACSi register is written when the D/A conversion occurs.

The results of the previous D/A conversion are maintained until the next D/A conversion is performed.
Stop the D/A conversion if the DACEi position of the DAM register is "0" (stop D/A conversion).

Note 1. Even if the setting value of the DACEi bit is set as '1', the analog voltage set by the DACSi register is output to the ANOi pin after the settling time after the last setting of '1'.

2. If the DACSi register is overridden within a stable time, the conversion is aborted

and the conversion restarts with the overridden value. Note: i=0,1

16.4.2 Operation of Real-time Output Mode

Each channel of the D/A converter uses the event signal of EVENTC as the start trigger to perform the D/A conversion.

It is set as follows:

① Start supplying the D/A converter with the DACEN position "1" of the PER1 register (Peripheral Enable Register 1).

② Set the port to the analog pin through the PMC register (port mode control register).

③ Place the DAMDi position of the DAM register (mode register of D/A converter) in "0".

④ The DACSi register (D/A conversion value setting register i) sets the analog voltage value of the ANOi pin output.

⑤ Place the DACEi location of the DAM register "1" (D/A conversion allowed).
The D/A conversion is started, and after a settling time, the analog voltage set by (3) is output to the ANOi pin.

⑥ A trigger signal for real-time output mode is set by an event output target selection register n (ELSELRn, n=00~21).

⑦ Place the DAMDi location of the DAM register "1" (Live Output Mode).

⑧ Start the run of the event occurrence source.

The above (1) to (8) are the initial settings.

⑨ Thereafter, the D/A conversion is started by generating a trigger signal for the real-time output mode, and the set analog voltage is output to the ANOi pin after the settling time.
Before the next D/A conversion (generating a trigger signal for the real-time output mode), the DACSi register is set with the analog voltage value of the ANOi pin output.

The DACSi register must be set to the analog voltage value of the ANOi pin output before the next D/A conversion (generating a trigger signal for the real-time output mode).

Stop the D/A conversion if the DACEi position of the DAM register is "0" (stop D/A conversion).

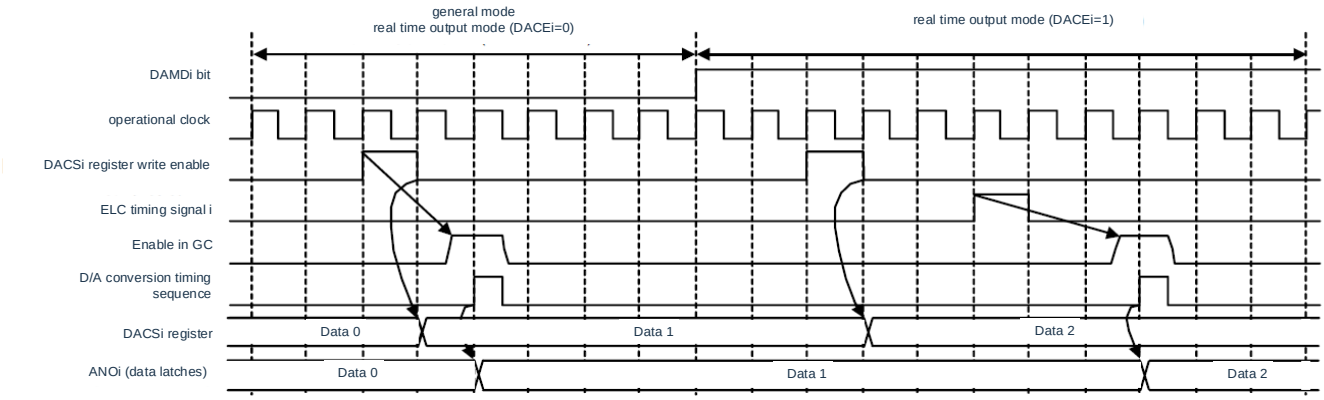
Note 1. Even if the setting value of the DACEi bit is set to "1" →"0"➞1", the DACSi register setting analog voltage is output to the ANOi pin.

2. The generation interval of the trigger signal for the same channel real-time output mode must be greater than the stability time. If a trigger signal for the real-time output mode is generated in a stable time, the D/A conversion is aborted and resumed.

3. The generation interval of the trigger signal for the same channel real-time output mode must be greater than 3 fCLK clocks. If a start trigger is continuously generate at intervals less than or equal to 3 fCLK clock, only D/A conversion is performed upon generation of that first trigger.

16.4.3 Output Timing of D/A Conversion Values

The output timing of the D/A conversion values is shown in Figure 16-5.



Note: i=0,1

- Normal Run Mode and Live Output Mode (situations where conversion is not allowed)
  Write the data latch (output from the ANOi pin) after 1 cycle of writing the DACSi register (runtime clock).
- Live Output Mode (if conversion is allowed)
  Writes the data latch (output from the ANOi pin) after 3 cycles of accepting the EVENTC event signal.

16.5 Precautions of using the D/A converter

The following precautions shall be taken taken when using the D/A converter.

1) When the port is set as an analog pin through a PMC register (port mode control register), the input/output function of the digital port multiplexed with ANO0 pin does not work. If a P2 register is read during port setting to an analog pin through the ADPC register, a value of P2 is read in input mode. In addition, the output data is not output to the pin even if the digital output mode is set.
2) In sleep mode and deep sleep mode, the D/A converter continues to operate. In order to reduce power consumption, the DACEi bit must be cleared "0" and the WFI instruction must be executed after stopping D/A conversion.

Notes: i=0,1

3) When you stop the real-time output mode, including when you change to the normal mode, follow these steps:
   · Wait at least 3 clocks after stopping triggering the output source, and then place the DACEi bit and DAMDi position "0".
   · After "0" the DACEi bit and DAMDi position, "0" the DACEN position of the PER1 register (stop DAC).
   If the DACEN location is "0," all registers inside the DAC are cleared.
   Therefore, each SFR needs to be set up when the restart runs.
4) When D/A conversion is allowed, A/D conversion cannot be performed on analog input pins that are multiplexed with ANO0 and ANO1.
5) In real-time output mode, the value of the DACSi register must be set before generating a trigger signal for the real-time output mode.
   You cannot change the setting value of the DACSi register while the trigger signal is active.
6) Because of the high output impedance of the D/A converter, the current cannot be obtained from the ANO0 pin and the ANO1 pin. In case of low input impedance of load, the tracking amplifier must be inserted between load and ANO0 pin and ANO1 pin. In addition, cabling between the tracking amplifier and the load must be minimized (due to the high output impedance).
   If the wiring is too long, grounding patterns or the like must be processed around the wiring.
7) If you want to enter deep sleep mode when live output mode is active, you must disable event linking for EVENTC before entering deep sleep mode.

Chapter 17 comparator

This product has a comparator with 2 channels built in.

17.1 Function 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.

· When the motor stops, the position of the motor can be detected by comparing the combinations of U, V and W to realize sensorless motor control.

· Three-phase zero cross detection can be realized by switching a comparator.

· The comparison result of comparator 0 and comparator 1 can be output by pins (VCOUT0, VCOUT1).

Table17-1 Comparator Feature Summary

Project	Content
CMP	· 2-channel comparators (CMP0 and CMP1) · The negative end of the comparator can select a reference voltage: Optional external pin input on the negative side of the CMP0, built-in reference voltage of CMP0 and internal reference voltage (1.45V) Optional external pin input (4) for the negative end of the CMP1, built-in reference voltage of CMP1 and internal reference voltage (1.45V) · The internal reference voltage of the negative end may be set (256steps) · The front end of the CMP0 selects the output of the PGA · The front end of the CMP1 can select external pin inputs (4) · When the input voltage of the positive end > the input voltage of the negative end, the output high level When the input voltage of the positive end is less than the input voltage of the negative end, the output level is low · Filter width of digital filter is optional · output inversion function · The comparison result can be output from the VCOUT0 (VCOUT1) · An effective edge of the comparator output can be detected and an interrupt signal generated · Combined with other functions, the initial position of the motor can be detected and high/low speed rotation can be controlled The 6-phase PWM output of the timer can be set/reset to high resistance state under the condition of overcurrent · Combined with Timer4 to output TIMER WINDOW

17.2 Structure of comparator

The box diagram of the comparator is as followsFigure 17-1in the

Figure 17-1 Block diagram for comparator 0

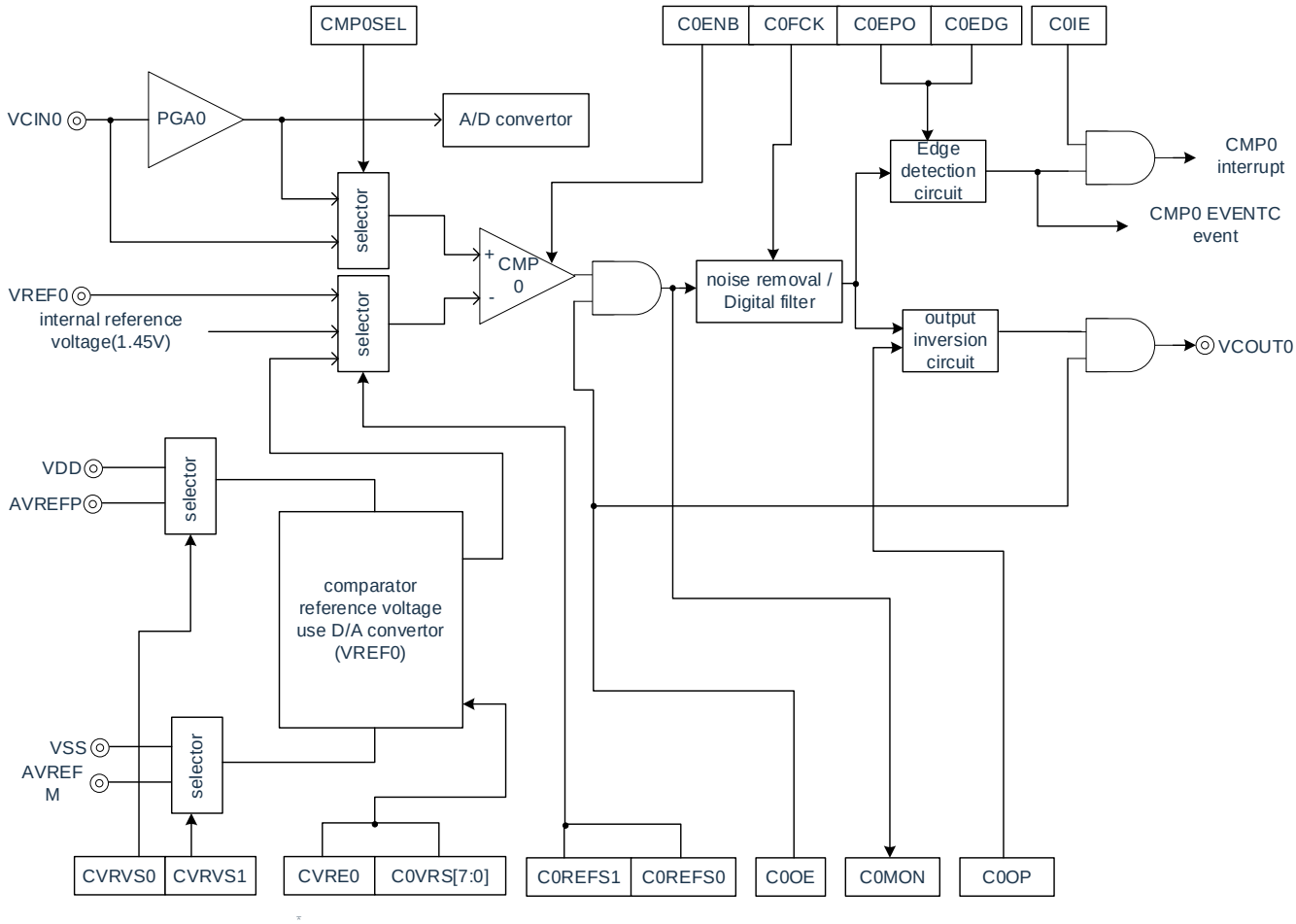
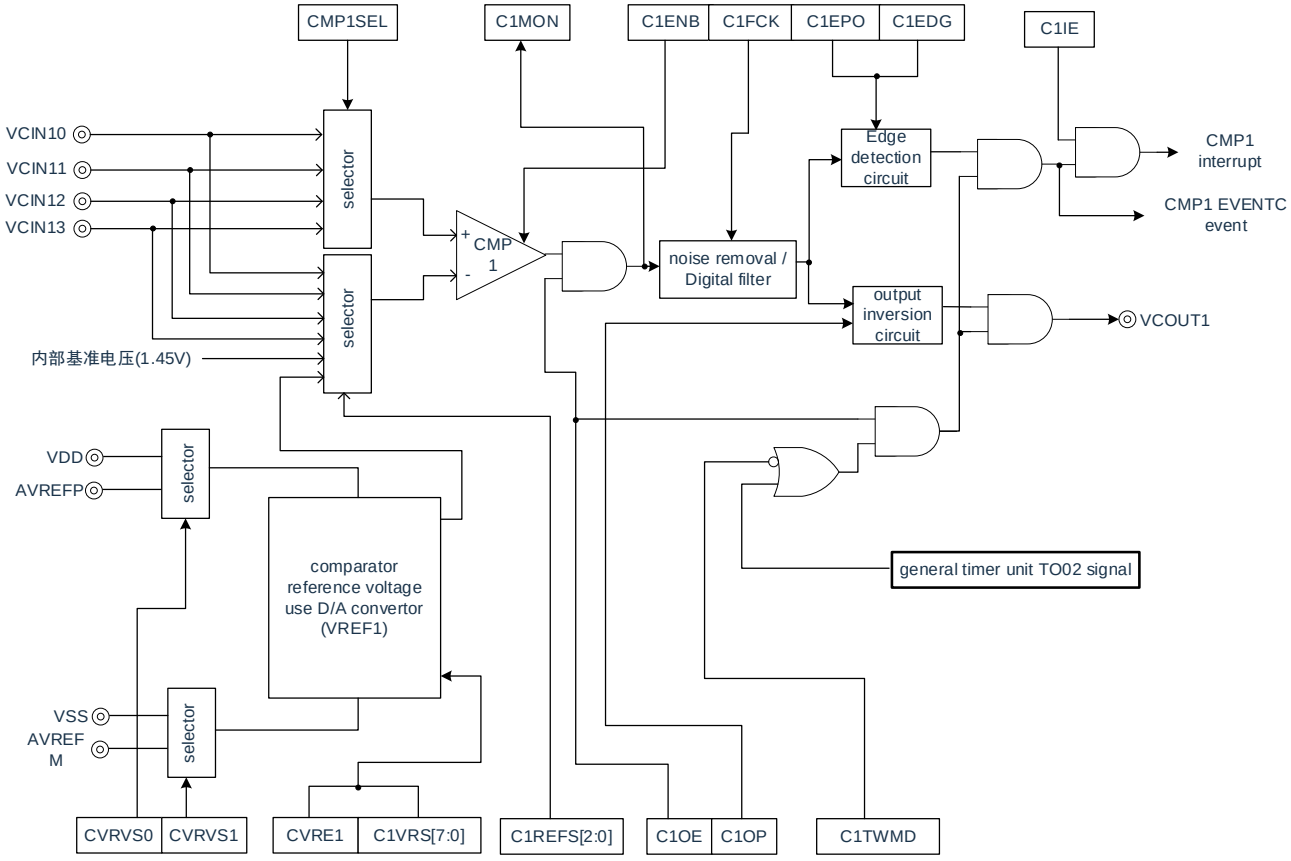


Figure 17-2 Block diagram for comparator 1



17.3 Register for control comparator

The registers controlling the comparator are as followsTable17-2in the

Table17-2 Register for control comparator

register name	symbol
Peripheral Enable Register 1	PER1
comparator mode setting 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 register	Pxx

17.3.1 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.

You must set bit5 (PGACMPEN) to '1' when you want to use comparators.

The PER1 register is set by an 8-bit memory operation instruction.

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

Figure 17-3 Format of Peripheral Admission Register 1(PER1)

Address: 0x4002081A After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN	TMMEN	DMAEN	PWMPEN	TMCEN	TMAEN

PGACMPEN	Control of comparator input clock
0	Stop provide an input clock. · Cannot write the SFR used by the comparator. · The comparator is in a reset state.
1	Provides an input clock. · SFR that can read and write to the comparator.

Note: The PGACMPEN location "1" must first be set for comparator.

When the PGACMPEN bit is "0", writes to the comparator's control register are ignored and the read values are initial (except Port Mode Register (PMxx) and Port Register (Pxx).

17.3.2 Comparator mode set-up register (COMPMDR)

The COMPMDR register is a register that sets the comparator action permission/prohibition and detects the comparator output.

The CiENB bit is not set to '0' when the comparator output license (CiOE position '1' of the COMPOCR register).

The CiENB position "1" (i=0,1) is prohibited:

- 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 operation instruction.

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

Figure 17-4 Format of Comparator Mode Set-up Register (COMPMDR)

Location: 40043840H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
COMPMDR.	C1MON	0	0	C1ENB	C0MON	0	0	C0ENB

C1MON	Comparator 1 Monitor Flag Notes 1, 2
0	VCIN1<Reference voltage of comparator 1, or comparator 1 is not running.
1	VCIN1> Reference voltage for comparator 1

C1ENB	Allow for comparator 1 to run
0	Disables comparator 1 operation.
1	Allow comparator 1 to run.

C0MON	Monitor Flag Note 1,2 for Comparator 0.
0	VCIN0<Reference voltage of comparator 0, or comparator 0 is not running.
1	VCIN0>Reference voltage of comparator 0

C0ENB	Allow for comparator 0 to run
0	Prevents comparator 0 from running.
1	Allow comparator 0 to run.

Note 1. Immediately after unreset becomes "0" (initial value), if both C0ENB and C1ENB bits are "0" after allowing comparator operation.

2. Ignore the write value for this bit.

17.3.3 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 operation instruction.

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

Figure 17-5 Format of comparator filter control register (COMPFIR)

Location: 400438 41H. 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 descent 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 through the fCLK.
1	0	Comparator 1 has a filter and samples it by fCLK/8.
1	1	Comparator 1 has a filter and samples through fCLK/32.

C0EDG	Comparator 0 Edge Detection Selection Note 2
0	Interrupt requests are generated by single edge detection of comparator 0.
1	Interrupt requests are 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 descent 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, through the fCLK sampling.
1	0	Comparator 0 has a filter, and samples through fCLK/8.
1	1	Comparator 0 has a filter, and samples through fCLK/32.

Note:

1. If C1FCK1~C1FCK0 bits, C1EPO bits and C1EDG bits are changed, interrupt request and event signals output to EVENTC may occur. These bits must be changed after setting the EVENTC ELSELR21 register (output of unlinked comparator 1) to "0". In addition, the IF of the interrupt request flag register must be cleared "0".
   If C1FCK1~C1FCK0 bit is changed from '00B' (comparator1 has filter), the comparator 1 must use the interrupt request or the event signal output to EVENTC after 4 sampling before updating the output of the filter.

2. If you change the C0FCK1~C0FCK0 bits, C0EPO bits, and C0EDG bits, you may generate a interrupt request for comparator 0 and event signals output to EVENTC. These bits must be changed after setting the EVENTC ELSELR20 register (output of unlinked comparator 0) to "0". In addition, the IF of the interrupt request flag register must be cleared "0".

   If C0FCK1~C0FCK0 bit is changed from '00B' (comparator 0 without filter) to other values (comparator 0 with filter).

17.3.4 Comparator output control register (COMPOCR)

The COMPOCR register is a control register that sets the polarity of the comparator output, the permission/prohibition of the output, and the permission/prohibition of the interrupt output.

In the following cases, the CiOE position "1" (output license) of the COMPOCR register is prohibited. (i=0,1)

- When the comparator action stops (the CiENB bit of the COMPMDR register is "0")
- 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 the PGA action stops (the CMPSEL0 bit of the CVRCTL register is "1" and the PGAEN bit of the PGAEN register is "0")

The COMPOCR register is set by an 8-bit memory operation instruction.

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

Figure 17-6 Format of comparator output control register (COMPOCR)

Location: 400438 42H. 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 1TIMER WINDOW output mode (co-controlled by TO02 and C1OE)

C1OP	Selection of Output Polarity of VCOUT1
0	Output of comparator 1 from VCOUT1.
1	Invert output of comparator 1 from VCOUT1.

C1OE	Allowable for VCOUT1 Pin Output Note 2
0	Disables the VCOUT1 pin output of comparator 1.
1	Allow VCOUT1 pin output for comparator 1.

C1IE	Comparator 1 Allow for Interrupt Request Note 3
0	Disables interrupt request for comparator 1.
1	Allow interrupt requests for comparator 1.

C0OP	Selection of Output Polarity of VCOUT0
0	Output comparator 0 from the VCOUT0.
1	Invert output of comparator 0 from VCOUT0.

C0OE	Allowable for VCOUT0 Pin Output Note 4
0	Disables VCOUT0 output to the pin for comparator 0.
1	Allow VCOUT0 output of comparator 0 to footnote 4, note 8.

C0IE	Comparator 0 Allow for Interrupt Request Note 5
0	Disable interrupt request for comparator 0.
1	Allow interrupt requests for comparator 0.

Note1. When comparator 1 uses the TIMER WINDOW mode, the bit7 (C1EDG) of register COMPFIR must be set to "1".
C1OE and C1OTWMD bits cannot be set at the same time. Set the C1OTWMD bit before setting the C1OE position.
Note2. When you override the C1OE bit, you may generate a comparator 1 interrupt request and a EVENTC event. Please override this bit after setting the EVENTC's ELSELR21 register to 0 (output not linked to comparator 1). In addition, initialize the IF bit of the interrupt request flag register (non-interrupt request) after overwriting the C1OE bit.
Note3. When the result of comparator 1 is output to the pin, the bit2 (PIOR32) of the PIOR3 register must be set to "1".
Note4. When you override the C0OE bit, a comparator 0 interrupt request and a EVENTC event may occur. Please override this bit after setting the ELSELR20 register for EVENTC to 0 (output not linked to comparator 0). In addition, initialize the IF bit of the interrupt request flag register (noninterrupt request) after overwriting the C0OE bit.
Note5. The C0OE bit, the C0OP bit, and the result of the comparator 9 are input to the PWM option unit to control the forced cut-off of the PWM output.
Note6. If C1IE is changed from 0 to 1, the IF of the interrupt request flag register may become 1, so interrupt must be used after clearing IFL.
Note7. When the result of comparator 0 is output to the pin, the bit1 (PIOR31) of the PIOR3 register must be set to "1"
Note8. If C0IE is changed from 0 to 1, the IF of the interrupt request flag register may become 1, so interrupt must be used after IF.

17.3.5 Comparator built-in reference voltage control register (CVRCTL)

The CVRCTL register is a register that sets the built-in reference voltage permit/stop action of 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: The CVRVSi bit of the CVRCTL register is overridden when the built-in reference voltage stop action (CVREi=0).

Figure 17-7 Format of comparator built-in reference voltage control register (CVRCTL)

Location: 400438 43H. 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	Allow operation of built-in reference voltage 1

CVRVS1	ground side selection bit with built-in reference voltage
0	Internal Reference Voltage ground Side Select Vss
1	Internal Reference Voltage ground Side Select AVREFM Note 1

CVRE0	Control bit with built-in reference voltage 0
0	Disable operation of built-in reference voltage 0
1	Allow operation of built-in reference voltage 0

CVRVS0	Power supply side selection bit with built-in reference voltage
0	Power Supply Side Selection VDD with Internal Reference Voltage
1	Internal Reference Voltage Power Supply Side Selection AVREFP Note 2

Note The 1:P21 pin uses both AVREFM and VCIN13, so the CVRVS1 position "1" is not allowed when the P21 pin is the input signal for CMP1.

Note 2:P20 pins use both AVREFP and VCIN12, so the CVRVS0 position "1" is not allowed when the P20 pin is used as the input signal for CMP1.

17.3.6 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 operation instruction.

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

Figure 17-8 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

CiRVS7	CiRVS6	CiRVS5	CiRVS4	CiRVS3	CiRVS2	CiRVS1	CiRVS0	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

17.3.7 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 operation instruction.

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

Figure 17-9 Format of the input signal selection control register (CMPSEL0) for comparator 0

Location: 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 Internal Reference Voltage (1.45V)
1	0	Select an external pin (IVREF0 pin)
1	1	Disable from setting

17.3.8 Comparator 1 input signal selection control register (CMPSEL1)

The CMPSEL1 register is a selection register for the input signal of the positive end 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 operation instruction.

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

Figure 17-10 The format of the input signal selection control register (CMPSEL1) of comparator 1

Location: 4004384BH After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
CMPSEL1	CMP1SEL1	CMP1SEL0	0	0	0	C0REFS1	C0REFS1	C0REFS0

CMP1SEL1	CMP1SEL0	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)

C0REFS2	C0REFS1	C0REFS0	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 from setting
1	1	1	Disable from setting

Note: When switching the analog input of the CMP1, the switching interval must be more than 3 us in order to prevent the through current before the two input signals.

17.3.9 Register for controlling analog input pin port function

When using the VCIN0 pin, VCIN10-VCIN1 3 pin and VREF0 pin as analog inputs to the comparator, the bits of port mode register (PMxx) and PMCxx.

When using VCOUT0 and VCOUT1 functions, you must set the control registers (PMxx and Pxx) and peripheral IO redirection registers (PIOR2, PIOR3). Please refer to the "2.3.1Port Mode Register (PMxx)","2.3.2Port Register (Pxx)","2.3.9Peripheral I/O Redirection Register 2 (PIOR2)"and"2.3.10Peripheral I/O Redirection Register 3 (PIOR3)"

17.4 Operation Instructions

Comparator 0 and Comparator 1 can be run independently. The set-up method is the same as the run.

The CMP0 and PGA0 can be combined together.

The set-up steps for the comparator's independent operation and interactivity are as followsTable17-3in the

Table17-3 Set-up steps for comparator-related registers

Step	register	bit	Set Values
1	PGACTL	PGAVG0/1/2	Select Gain Note 3
2	PGACTL	PVRVS	0 (Vss Pin Selection) Note 3
3	PGACTL	PGAEN	1 (Allow to run) Note 3
4	Wait for PGA stability time (minimum 10 μ s)		
5	COMPSELi	CMP0SEL/CMP1SELi 1	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 20 μ s)		
11	Set VCIN0, VCIN1x, IVREF0 pin (input), PGAI Note 3 to the analog input function. PMCxx The VCIN0 pin, VCi, and IVREFi pin feature selection places the PMCxx position "1" (analog input). Position PMxx "1" (input mode).		
12	COMPMDR.	CiENB	1 (Allowed to run)
13	Wait for the stability time of the comparator (minimum 3 μ s)		
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	Sets the output of the VCOUTi (select Polarity, set Allow or Disable output). Refer to "Output of 17.4.4 comparator i (i=0,1)".
		CiIE	Sets the output that allows or disables the interrupt request. Refer to the "17.4.4 comparator i output (i=0,1)
		C1OTWMD	Set TIMER WINDOW Output Licensing/Disallowed for Comparator 1
16	MKxx Note 1	MKL	When using interrupts: Select Mask Interrupt.
17	IFxx Note 1	IFL	When using interrupts: 0 (non-disruptive request: initialization) Note 2

Note 1. MKxx,IFxx is the comparator interrupt control register, refer to "Chapter 25 Interrupt Features" for details.

Note 2. After the comparator is set, an undesired interrupt request may occur during stable operation and the interrupt request flag must be initialized.

Note3. Comparator 0 must be set when interacting with PGA

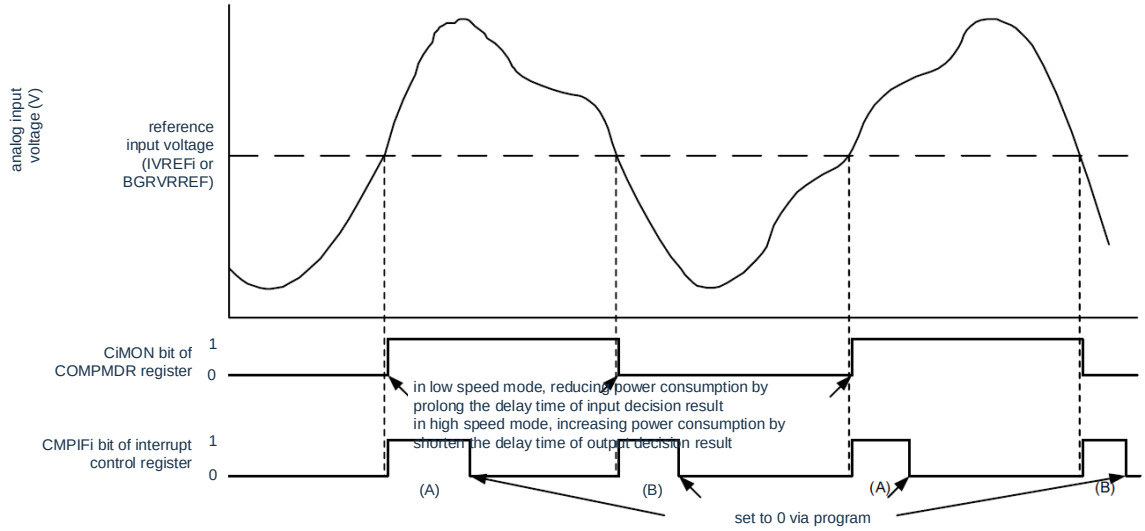
Notes: i=0,1,n=0-7,x=0-3

An example of operation of the comparator i (i=0,1) is as followsFigure 17-11in the In the basic mode, when the analog input voltage is higher than the reference input voltage, the CiMON bit of the COMPMDR register is "1"; When the analog input voltage is lower than the reference input voltage, the CiMON bit is "0".

To interrupt comparator i, you must have the CiIE location "1" of the COMPOCR register (allow interrupt requests). 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 "17.4.2 Comparator i interrupt (i=0,1)".

Figure 17-11　Running example of comparator i (i=0,1) (base mode)

- basic mode operation example



Note: The COMPFIR register has CiFCK1~CiFCK0 bit of '00B' (no filter) and CiEDG bit of '1' (two-sided edge) (CiEDG bit of '0' and CiEPO bit of '0' (rising edge) only CMPIFi changes in (A), CiEDG bit of '0' and CiEPO bit of '1' (falling edge).

17.4.1 Digital filter for comparator i (i=0,1)

The comparator i has a built-in digital filter, which can select the sampling clock through the CiFCK1~CiFCK0 bit of the COMPFIR register. The output signal of comparator i is sampled according to each sampling clock, and the digital filter outputs the sampling value.

Figure 17-12is a result of the digital filter of the comparator i,Figure 17-13Is an example of a digital filter and interrupt operation of a digital comparator i (i=0,1) of comparator i.

Figure 17-12 Digital filter of comparator i (i=0,1) and along detection structure

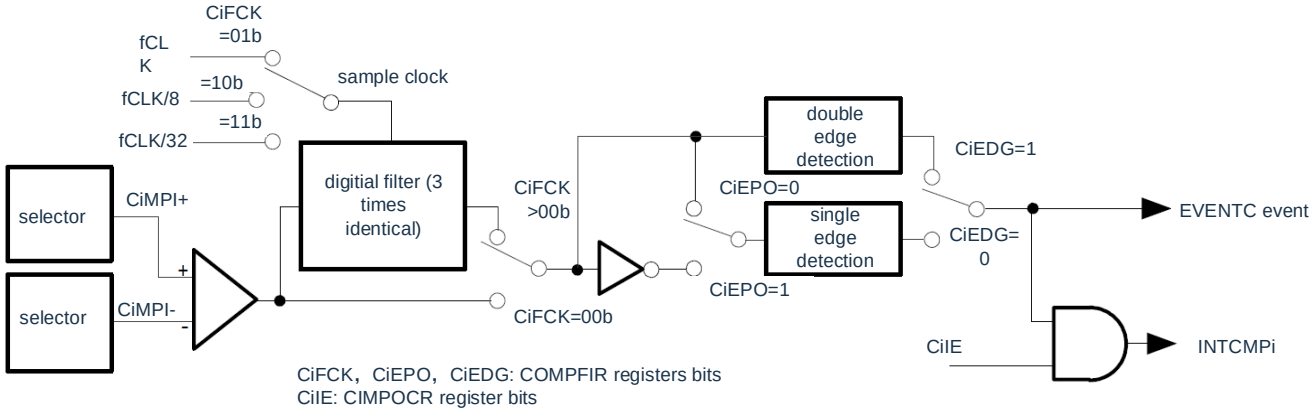
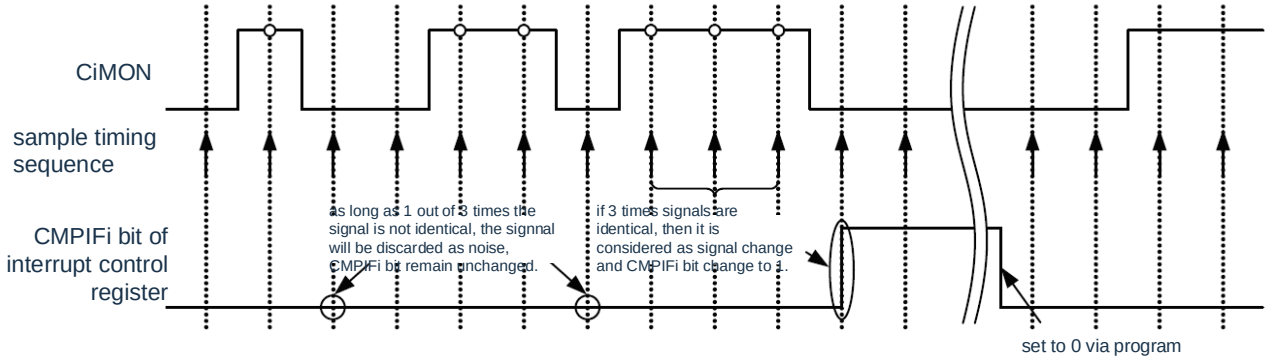


Figure 17-13 Examples of digital filters and interrupt operation of comparator i (i=0,1)



Note: The above figure is an example of the COMPFIR register running when the CiFCK1~CiFCK0 bit is "01B" or "11B" (with a digital filter).

17.4.2 comparator i interrupt (i=0,1)

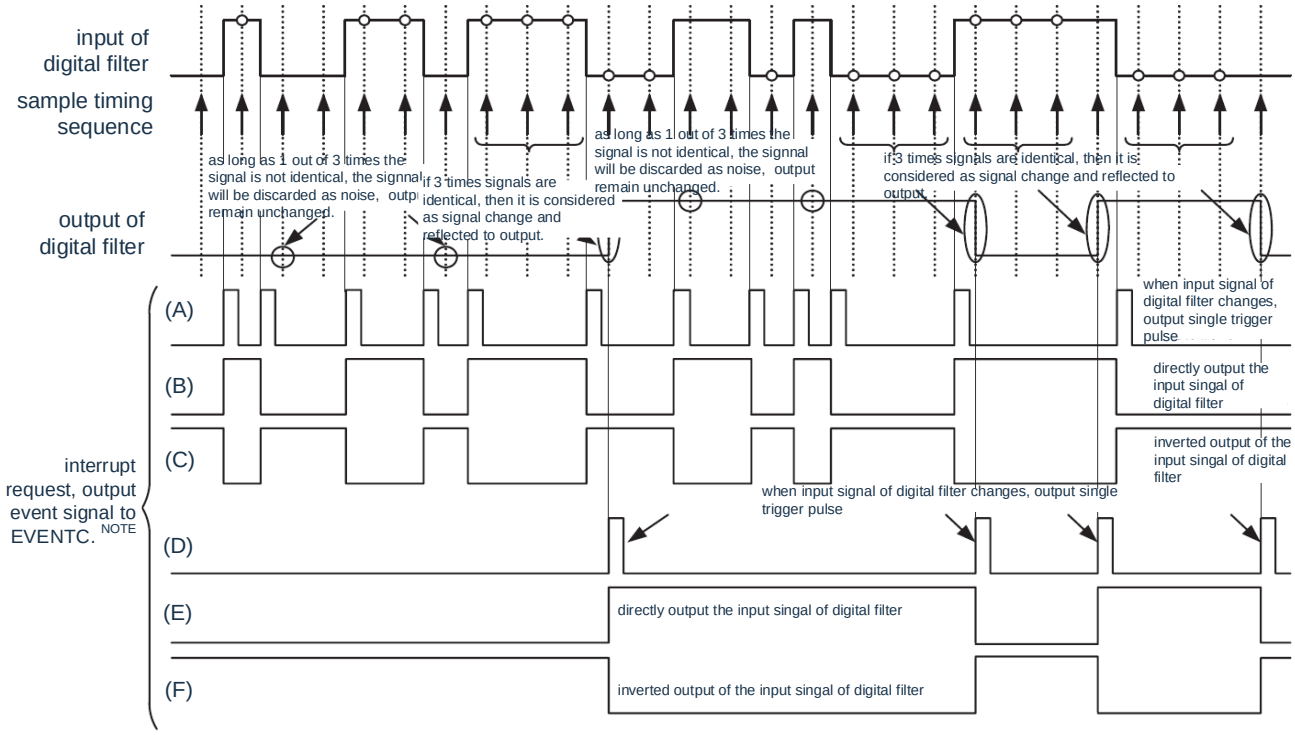
The comparator generates two interrupt requests for the comparator 0 and comparator 1. The comparator i interrupt has 1 priority specify flag, interrupt mask flag, interrupt request flag and interrupt vector respectively.

To interrupt comparator i, you must have the CiIE location "1" of the COMPOCR register to allow output of the interrupt request. The generation condition of the interrupt request is set through the COMPFIR register, and a digital filter can be added to the output of the comparator. The digital filter can select three kinds of sampling clock. Refer to the for register settings and to generate a response to interrupt requests.17.3.3 Comparator Filter Control Register (COMPFIR)" and"17.3.4 Comparator Output Control Register (COMPOCR)".

17.4.3 Event signal output to the Coordination 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 ELSELR20 register and ELSELR21 register.

Figure 17-14 Digital filter, interrupt request, and operation of outputting event signals to EVENTC



Note: When the CiIE bit (i=0, 1) is '1', the interrupt request and the event signal output to the EVENTC are identical waveforms.

When the CiIE bit (i=0,1) is "0", only interrupt requests are fixed "0".

(A), (B), (C) waveforms are the case that the CiFCK bit (i=0,1) of the COMPFIR register is '00B' (without a digital filter), and the waveforms of (D), (E), and (F) are the case that the CiFCK bit (i=0,1) of the COMPFIR register is '01B', '10B', or '11B' (with a digital filter). (A), (D) are cases where the CiEDG bit is 1 (bilaterally along), (B), (E) is the CiEDG bit '0' and the CiEPO bit '0' (rising along), (C), (F) is cases where the CiEDG bit is '0' and the CiEPO bit is '1' (falling along).

17.4.4 Output of comparator i (i=0,1)

The comparison result of the comparator can be output to external pin, and output polarity can be set through CiOP bit and CiOE bit of COMPOCR register. Refer to " for register settings and comparator output.17.3.4 Comparator output control register (COMPOCR)".

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 ("**错误!未找到引用源。**Steps 2-5 of the step).

② Sets the VCOUTi output of the comparator (sets the COMPOCR register, selects polarity, and allows output).

③ The port mode control register position "0" corresponding to the output pin of the VCOUTi.

④ Position "0" of the port register corresponding to the output pin of the VCOUTi.

⑤ Set the port direction register corresponding to the output pin of the VCOUTi to output (from the pin).

17.4.5 Stop and provision of comparator clock

In the case of stopping the comparator clock by setting the peripheral admission register 1 (PER1), you must follow these steps:

① Place the CiENB location of the COMPMDR register "0" (stop the comparator from running).

② The IF position of the interrupt request flag register is "0" (clear the interrupts that are not required before the comparator stops running).

③ Position "0" PGACMPEN for the PER1 register.

If you stop the clock by setting the PER1 register, all internal registers for the comparator are initialized, so you must follow theTable17-3Step Set-up Register for.

Note:

1. If the comparator n reference voltage selection bit (CnVRF) of the comparator mode setting register (COMPMDR) is set to '1', the output of the temperature sensor cannot be A/D converted.
2. If DMA start is permitted in one of the following states, DMA transfer is started and an interrupt is generated after transfer. Therefore, the monitor flag (CnMON) of the comparator must be confirmed as necessary to allow the DMA to start.
   - An interrupt request (CnEDG=0) is generated by unilateral edge detection of the comparator and an interrupt request (CnEPO=0) and VCIN>VREF (or internal reference voltage 1.45V).
   - An interrupt request (CnEDG=0) is generated by unilateral edge detection of the comparator and an interrupt request (CnEPO=1) and VCIN<VREF (or internal reference voltage 1.45V).

   (n=0, 1)

Chapter 18 programmable gain amplifier (PGA)

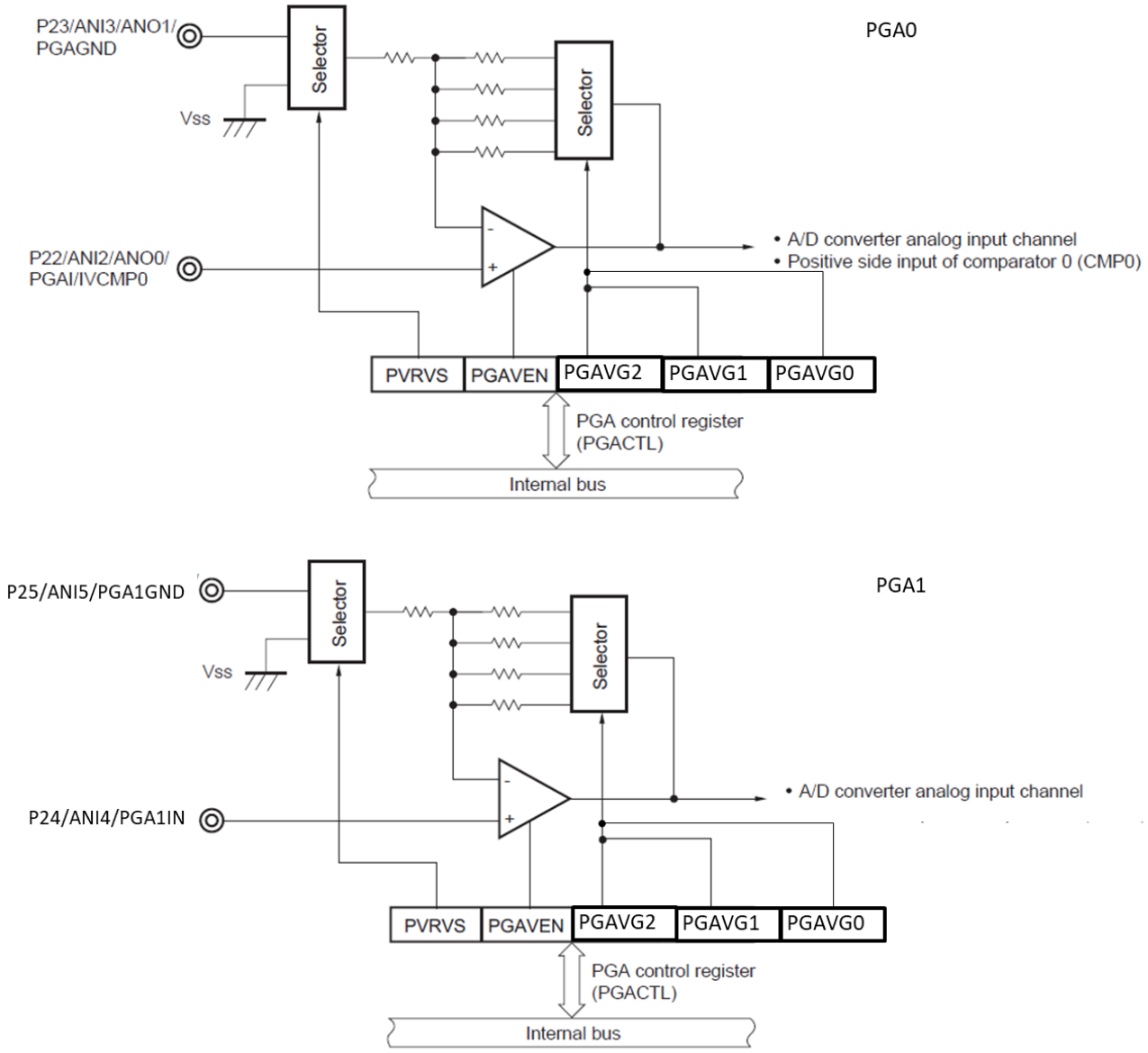
18.1 Function of Programmable Gain Amplifier

This product has two programmable gain amplifiers (PGA0 and PGA1) built in.

- Each PGA has 7 choices for amplification: 4x, 8x, 10x, 12x, 14x, 16x, 32x
- The external pin can be selected as the ground of the negative end feedback resistance of the PGA
- The output of the PGA 0 may be selected as the analog input for the A/D converter or the analog input for the positive end of the comparator 0 (CMP0)
- The output of the PGA1 can be selected as the analog input for the A/D converter

18.2 Structure of programmable gain amplifier

Figure 18-1: Block diagram of programmable gain amplifier



18.3 Register of programmable gain amplifier

Table18-1 Register for controlling programmable gain amplifier

Peripheral Admission Register 1	PER1
programmable gain amplifier control register	PGACTL
Port Mode Control Register 2	PMC2
Port Mode Register 2	PM2

18.3.1 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 register must be set to '1' when using a programmable gain amplifier.

The PER1 register is set by 1-bit or 8-bit memory operation instructions.

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

Figure 18-2 Format of Peripheral Admission Register 1 (PER1)

Reset Value: 00H R/W

symbol	7	6	5	4	3	2	1	0
PER1	DACEN	TIMERBEN	PGACMPEN	TIMERMEN	DTCEN	PWMPEN	TIMERCEN	TIMERAEN

PGACMPEN	Control of input clock of comparator/programmable gain amplifier
0	Stop provide an input clock. The comparator or the programmable gain amplifier is in a reset state
1	Provides an input clock. Register readable and writable of comparator or programmable gain amplifier

Note:

Before configuring the comparator or the register of the programmable gain amplifier, confirm that the bit bit of the PGACMPEN is set to 1.

If PGACMPEN=0, writing to the comparator or the programmable gain amplifier control register is invalid, and all read-out values are default values. (except for Port Mode Register 2 (PM 2) and Port Register P2)

18.3.2 programmable gain amplifier control register (PGAnCTL)

The PGA0CTL and the PGA1CTL register are used to control the programmable gain amplifier to start working, stop working, and amplify.

The PGA0CTL and PGA1CTL registers can be set by 1-bit or 8-bit memory instructions. After the reset signal is generated, the reset value of this register is 00H.

Figure 18-3 Format of PGA Control Register (PGAnCTL)

Reset Value: 00H R/W

	7	6	5	4	3	2	1	0
PGAnCTL	PGAnEN	-	-	-	PVVS[Note]	PGAnVG2	PGAnVG1	PGAnVG0

n=0,1

PGAnEN	programmable gain amplifier operation control
0	Amplifier Down
1	Allow amplifier operation

PVVSNote	Selection of Feedback Resistance Ground
0	Select Vss
1	Select PGAnGND Pin

PGAnVG2	PGAnVG1	PGAnVG0	PGAn gain
0	0	0	4x
0	0	1	8x
0	1	0	10x
0	1	1	12x
1	0	0	14x
1	0	1	16x
1	1	0	32x
Others			Disable from setting

This bit is set to 0 when Note 24 pin products.

Note: When the PGAnEN is set to 1, the programmable gain amplifier requires 10us steady-state time.

18.3.3 Register for controlling analog input pin port function

When the PGA0IN pin, PGA 1IN pin, PGA0GND pin, and PGA1GND pin are used as analog inputs to a programmable gain amplifier, the bits of the port mode register (PMxx) and the position of the port digital/analog control register (PMCxx) must correspond to the ports "1".

18.4 Operation of programmable gain amplifier

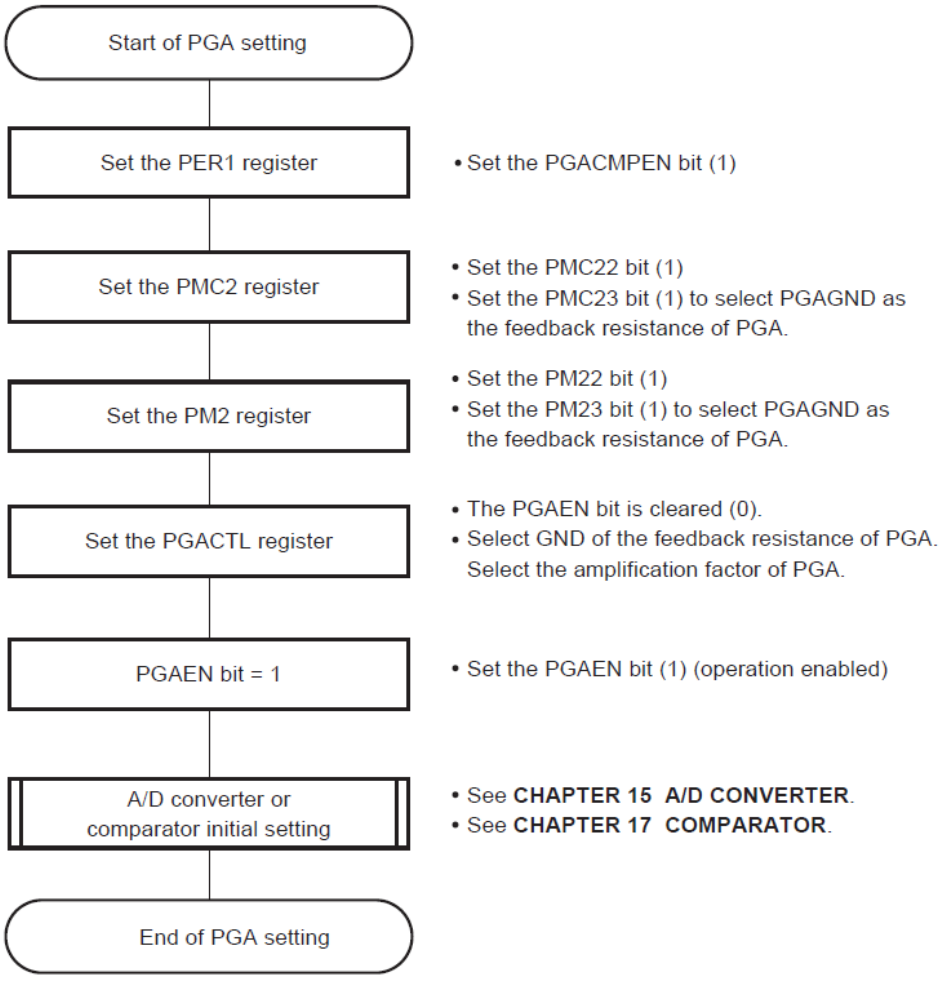
The analog voltage input by the PGAIN pin is amplified, and the amplification gain has seven choices: 4 times, 8 times, 10 times, 12 times, 14 times, 16 times and 32 times.

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.

18.4.1 Starting operation steps of programmable gain amplifier

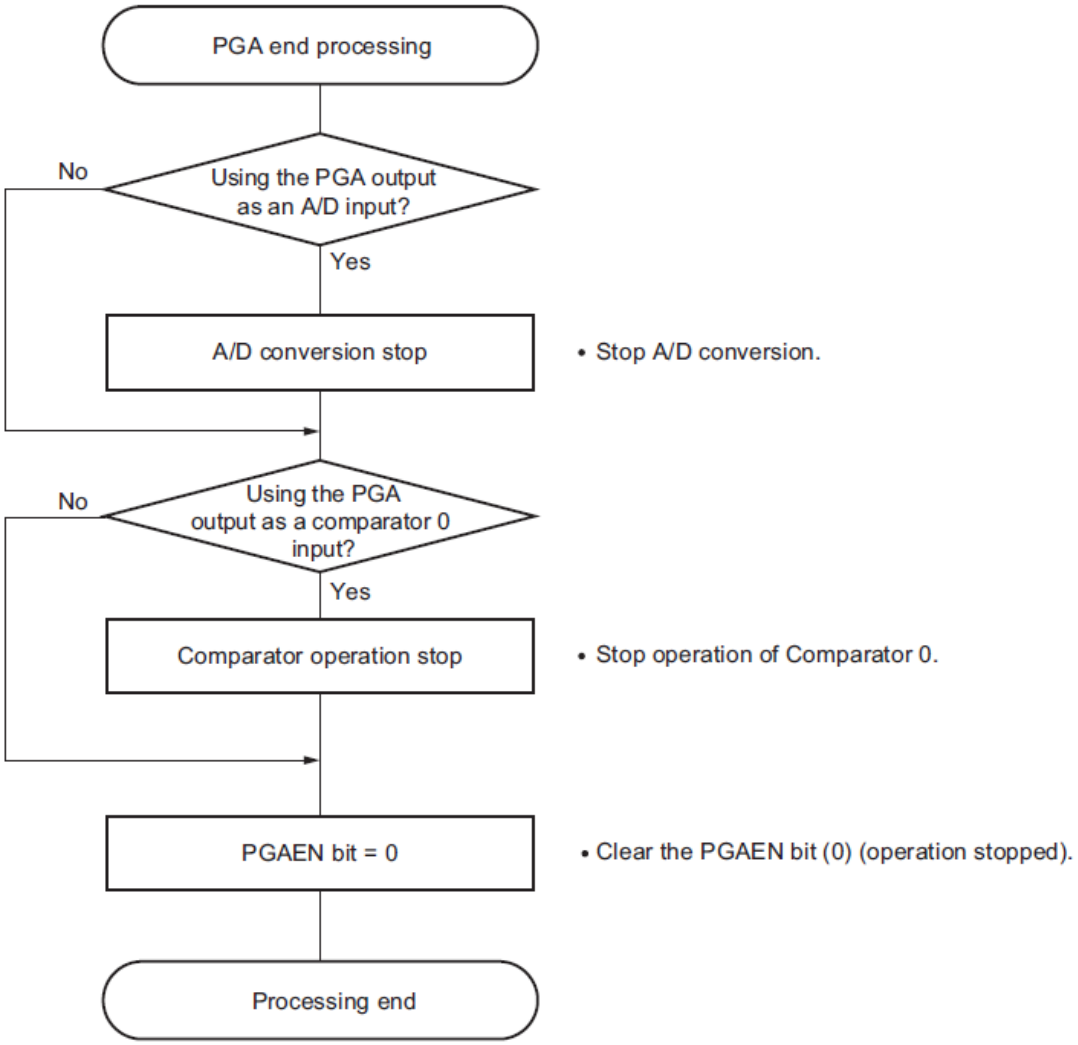
Take PGA0 as an example, set up as follows:



Note 1. A PGA stabilization time of 10us is required after setting the PGAEN bit to 1. The A/D conversion is then initiated.

18.4.2 Stopping operation step of programmable gain amplifier

Take PGA0 as an example, set up as follows:



Note 1. When restarting PGA and A/D conversion or amplifier, 10us PGA stabilization time is required after setting the PGAEN bit to 1.

2. Even if the PGA operation is stopped, straight-through pins can be used for A/D conversion and comparator action.

Chapter 19 universal serial communication unit

Unit 0 has 4 serial channels and unit 1 has 2 serial channels, each channel can realize 3-line SSPI, UART and simple I2C.

The functionality of each channel supported by BAT32A237 is distributed as follows:

○32 pin products

unit	channel	Use as SSPI	Use as UART	Used as Simple I2C
0	0	SSPI00 (Slave selectionInput Supported)	UART0 (LIN-bus support)	IIC00
	1	—		—
	2	—	UART1	—
	3	SSPI11		IIC11
1	0	SSPI20	UART2	IIC20
	1	—		—

○40 pin products

unit	channel	Use as SSPI	Use as UART	Used as Simple I2C
0	0	SSPI00 (Slave selectionInput Supported)	UART0 (LIN-bus support)	IIC00
	1	SSPI01		IIC01
	2	—	UART1	—
	3	SSPI11		IIC11
1	0	SSPI20	UART2	IIC20
	1	—		—

○48 pin products

unit	channel	Use as SSPI	Use as UART	Used as Simple I2C
0	0	SSPI00 (Slave selectionInput Supported)	UART0 (LIN-bus support)	IIC00
	1	SSPI01		IIC01
	2	—	UART1	—
	3	SSPI11		IIC11
1	0	SSPI20	UART2	IIC20
	1	SSPI21		IIC21

○64 pin products

unit	channel	Use as SSPI	Use as UART	Used as Simple I²C
0	0	SSPI00 (Slave selectionInput Supported)	UART0 (LIN-bus support)	IIC00
	1	SSPI01		IIC01
	2	SSPI10	UART1	IIC10
	3	SSPI11		IIC11
1	0	SSPI20	UART2	IIC20
	1	SSPI21		IIC21

SSPI00 and SSPI01 cannot be used when UART0 is used for channel 0 and channel 1 of unit 0, but SSPI10, UART1 and IIC10 of channel 2 and 3.

Note The following sections in this chapter describe the cell and channel structure of the 64 pin product.

19.1 Functions of Universal Serial Communication Unit

The features of each serial interface supported by BAT32A237 are shown below.

19.1.1 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 with 3 communication lines.

For specific set-up examples, refer to "Operation of 19.5 3-wire Serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21) communication".

[Sending and Receiving Data]

- 7-bit or 8-bit data length
- Phase control for transmitting and receiving data
- MSB/LSB First

Choice

[Clock Control]

- Master or subordinate selection
- Phase control of input/output clock
- A transmission period generated by a pre-divider and an intra-channel counter is set.
- maximum transfer rate note

Master Communications:

Max.fCLK/2

Slave Communication:

Max.fMCK/6

[Interrupt Capabilities]

- interrupt delivery end, buffer interrupt

[Error Detection Flag]

- overflow error

Note Must be used within a range that satisfies the $_{SCLK}$ Cycle Time (tKCY) characteristic. Refer to the data guide for details.

19.1.2 UART (UART0~UART2)

This is the ability to communicate asynchronously over a total of two lines, Serial Data Send (TxD) and Serial Data Receive (RxD). The two communication lines are used to transmit and receive data asynchronously (using internal baud rate) with other communication parties in data frames (consisting of start bit, data, parity bit and stop bit). Full duplex UART communication can be achieved by using two channels dedicated to transmit (even channel) and receive (odd channel), and LIN-bus can be supported by combining universal timer units and INTP0.

For specific set-up examples, refer to "19.7 UART (UART0~UART2) Communication Run."

[Sending and Receiving Data]

· Data length notes for 7, 8, or 9 bits

• MSB/LSB First Choice

· Level setting for transmitting and receiving data, selection of inversion

· Additional, parity functions for parity bits

· Additional stop bit

[Interrupt Capabilities]

· interrupt delivery end, buffer interrupt

· Error interrupt due to frame error, parity error, or overflow error

[Error Detection Flag]

· frame error, parity error, overflow error

UART0 (Channel 0 and Channel 1 of Unit 0)

supports LIN-bus. [LIN-bus Features]

· Detection of wake-up signal
· Detection of Interval Segments (BF)
· Measurement of Synchronous Section and Calculation of Baud Rate

} An external interrupt (INTP0) and universal timer unit are used.

19.1.3 Simple I2C (IIC00, IIC01, IIC10, IIC11, IIC20,IIC21)

This is the capability of clock synchronization with multiple devices through two lines of serial clock (SCL) and serial data (SDA). Since this simple I2C is designed for single communication with EEPROM, flash memory, A/D converter, etc., it is only used as master device.

For the start condition and stop condition, the AC specification must be observed, and the control register must be handled by software. For specific set-up examples, refer to "19.9 Easy I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) Communication Run".

[Sending and Receiving Data]

· Master Send, Master Receive (only for single master master master master functions)
· ACK output function Note, ACK detection function
· 8-bit data length (when sending an address, specifying the address with 7 bits high and R/W control with lowest bits)
· Manual generation of start and stop conditions

[Interrupt Capabilities]

· End of Transfer Interrupt

[Error Detection Flag]

· ACK error, overflow error

※[Features not supported by Simple I2C]

· Slave send, Dependent Receive
· Quorum Failure Detection
· Waiting for detection

Note When receiving the last data, the ACK is not output if writing "0" to the SOEmn bit (Serial Output Allowed Register m(SOEm). Refer to " 19.9.3 (2) Process Flow " for details.

Remark Refer to Chapter 20 Serial Interface IICA when using the full-function I2C bus.

19.2 Structure of universal serial communication unit

The universal serial communication unit consists of the following hardware.

Table 1 9-1 Structure of universal serial communication unit

Project	structure
shift register	8-bit or 9-bit note 1
buffer register	Low 8-bit or 9-bit note 1,2 for serial data register mn (SDRmn)
Serial clock input/output	SCLK00, SCLK01, SCLK10, SCLK11, SCLK20, SCLK21 pin (for 3-wire serial I/O), SCL00, SCL01, SCL10, SCL11, SCL20, SCL21 pin (for simple I2C)
serial data input	SDI00, SDI01, SDI10, SDI11, SDI20, SDI21 pin (for 3-wire serial I/O), RxD0 (for LIN-bus-enabled UART), RxD1, RxD2 pin (for UART)
serial data output	SDO00, SDO01, SDO10, SDO11, SDO20, SDO21 pin (for 3-wire serial I/O), TxD0 (for LIN-bus-enabled UART), TxD1, TxD2 pin (for UART)
Serial Data Input/Output	SDA00, SDA01, SDA10, SDA11, SDA20, SDA21 pin (for simple I2C)
slave selectioninput	SS00 Pin (for the slave selectioninput function)
control register	<Register for Cell Set-up> · Peripheral Enable Register 0 (PER0) · Serial clock selection register m (SPSm). · Serial Channel Allow State Storage m (SEm) · Serial channel start register m (SSm). · Serial channel stop register m (STm). · Serial output allows register m (SOEm). · Serial output register m (SOm). · Serial output level register m (SOLm). · Serial standby control register m (SSCm). · Input Switch Control Register (ISC) · Noise filter allows register 0 (NFEN0).
	<registers for each channel section> · Serial data register mn (SDRmn) · Serial mode register mn (SMRmn) · Serial communication run set-up register mn (SCRmn). · Serial state register mn (SSRmn). · Serial flag clear trigger register mn (SIRmn).
	· Port input mode register (PIMxx) · Port output mode register (POMxx) · Port mode register (PMxx) · Port register (Pxx)

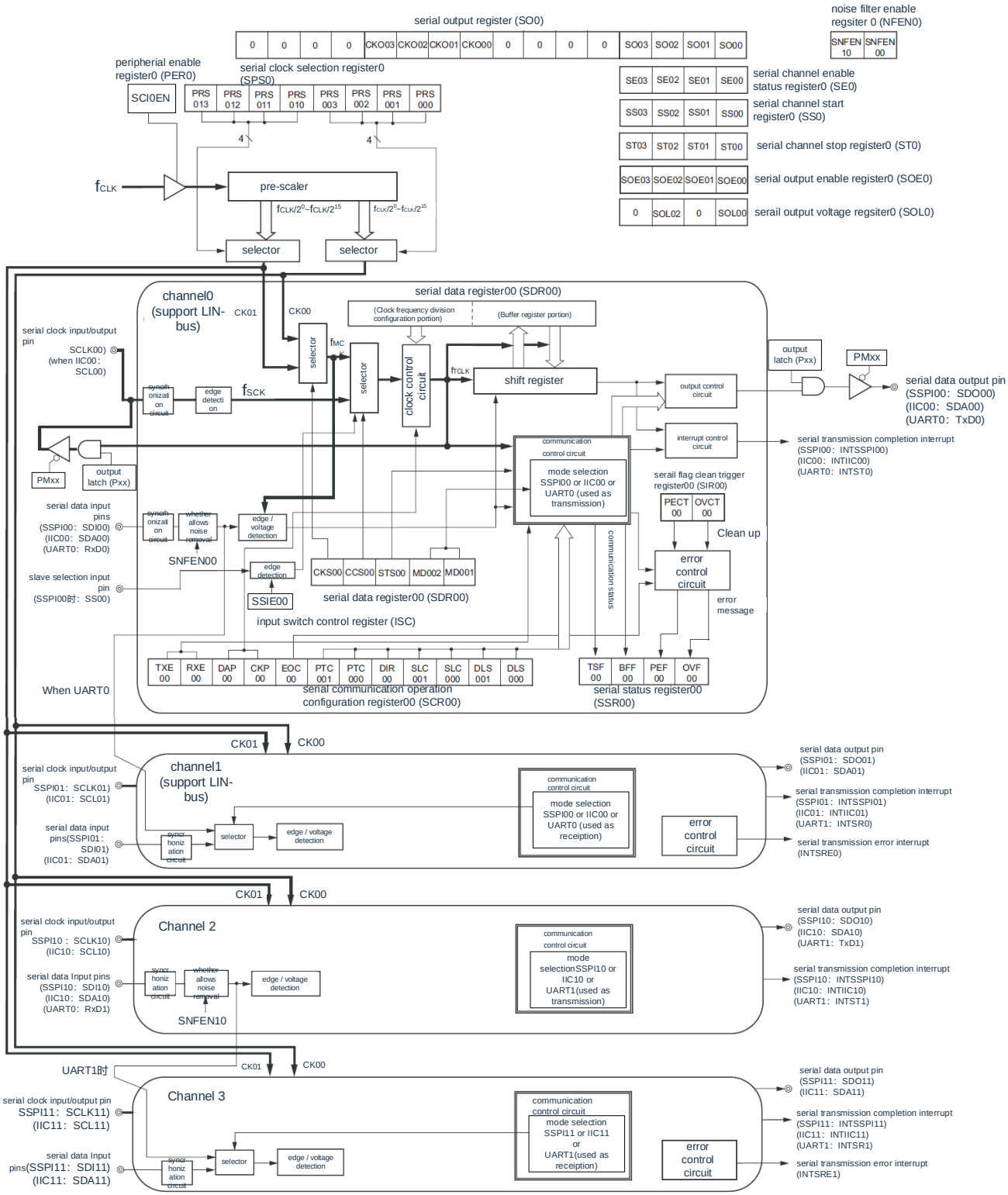
Note1. The bits used as shift registers and buffer registers vary depending on the cells and channels.
·24~64 pin products and mn=00,01: Low 9 bit
· Beyond the above: Low 8 bit

2. According to the communication mode, the low 8 bits of the serial data register mn (SDRmn) can be read and written.
- SSPIp Communications.....SIOp (SSPIp Data Register)
- UARTq Receive.....RXDq (UARTq Receive Data Register)
- UARTq Send.....TXDq (UARTq send data register)
- IICr Communications.....SIOr (IICr Data Register)

Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) p: SSPI Number (p=00,01,10,11,20,21)
q:UART (q=0~2)r:IIC (r=00,01,10,11,20,21)

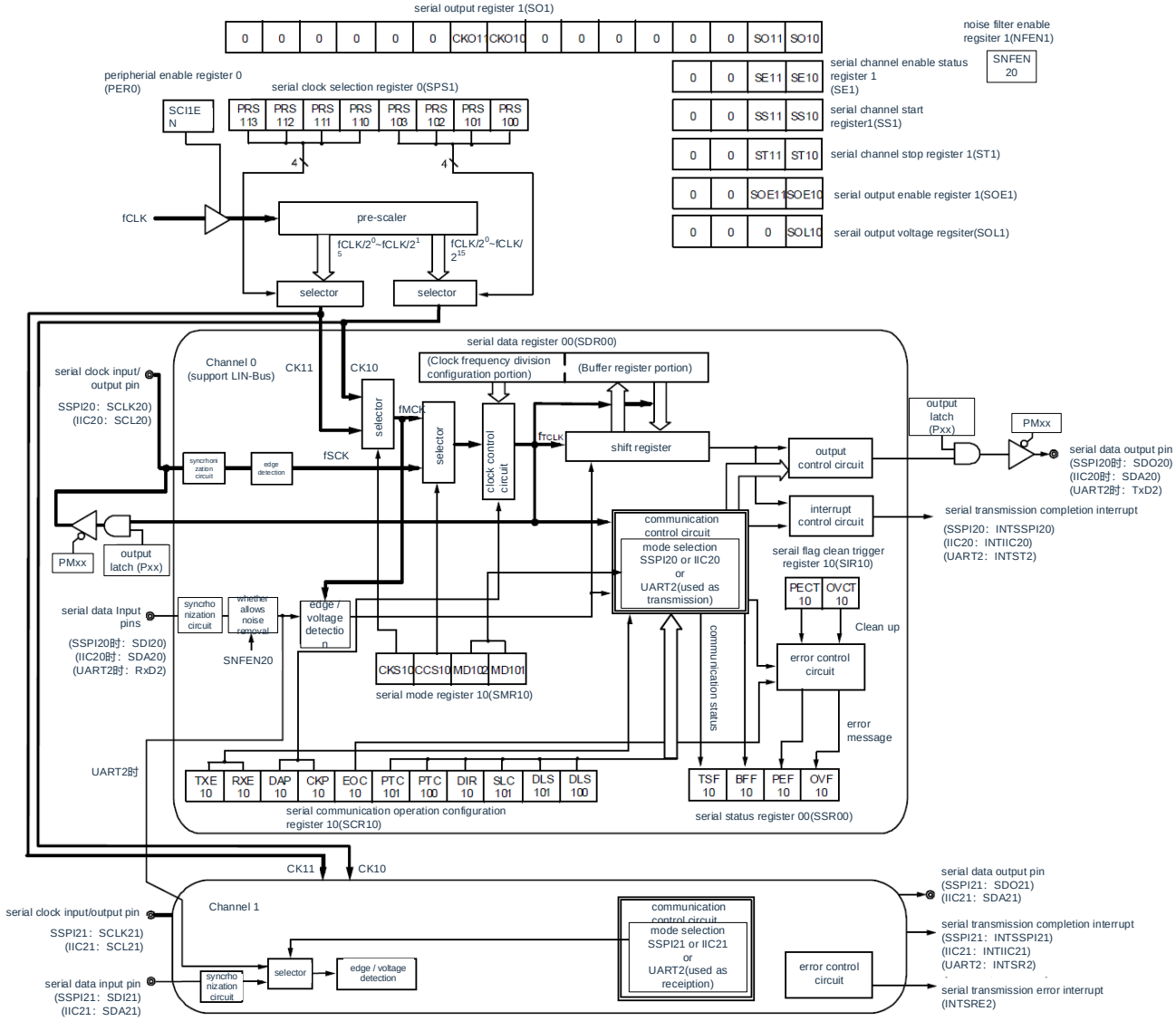
The block diagram of the universal serial communication unit 0 is shown in Figure 19-1.

Figure 19-1 Block diagram of the universal serial communication unit 0



The block diagram of the universal serial communication unit 1 is shown in Figure 19-2.

Figure 19-2 Block diagram of universal serial communication unit 1



19.2.1 shift register

This is a 9-bit register that performs parallel and serial interconversion.

When UART communication is performed with a 9-bit data length, a 9-bit (bit0~8) note 1. when receiving data, the input data of the serial input pin is converted into parallel data; When sending data, the value to be transferred to this register outputs note 1 from the serial output pin as serial data. The shift register cannot be operated directly through the program.

To read and write data from the shift register, use the low 8 bits or low 9 bits of the serial data register mn (SDRmn).

	8	7	6	5	4	3	2	1	0
shift register									

19.2.2 Low 8-bit or low 9-bit for serial data register mn (SDRmn)

The SDRmn register is a send and receive data register (16-bit) for channel n.

bit8~0 (Low 9 bits) or bit7~0 (Low 8 bits) serve as transmit and receive buffer registers, and bit15~9 serve as a fractional setting register for the runtime clock ($_{fMCK}$).

store that parallel data converted by the shift register to a low 8-bit or a low 9-bit when receiving the data; When transmitting data, the transmission data to be transmitted to the shift register is set to a low 8-bit or a low 9-bit.

Regardless of the output order of the data, the data saved to low 8 bits or low 9 bits is as follows according to bit0 and bit1 (DLSmn0,DLSmn1) of serial communication run set-up register mn (SCRmn):

- 7-bit data length (bit0~6 stored in the SDRmn register)
- 8-bit data length (bit0~7 stored in SDRmn register).
- 9-bit data length (bit0~8 stored in the SDRmn register) Note 1

The SDRmn register can be read and written in units of 16 bits.

According to the communication mode, it can read and write the low 8-bit or low 9-bit note of the SDRmn.

- SSPIp Communications.....SDIOp (SSPIp Data Register)
- UARTq Receive.....RXDq (UARTq Receive Data Register)
- UARTq Send.....TXDq (UARTq send data register)
- IICr Communications.....SDIOr (IICr Data Register)

After the reset signal is generated, the value of the SDRmn register changes to "0000H".

Note 1. Only UART0 supports 9-bit data length.

2. When running stop (SEmn=0), disable the SDRmn[7:0] rewrite via 8-bit memory operation instructions (otherwise SDRmn[15:9] is all cleared).

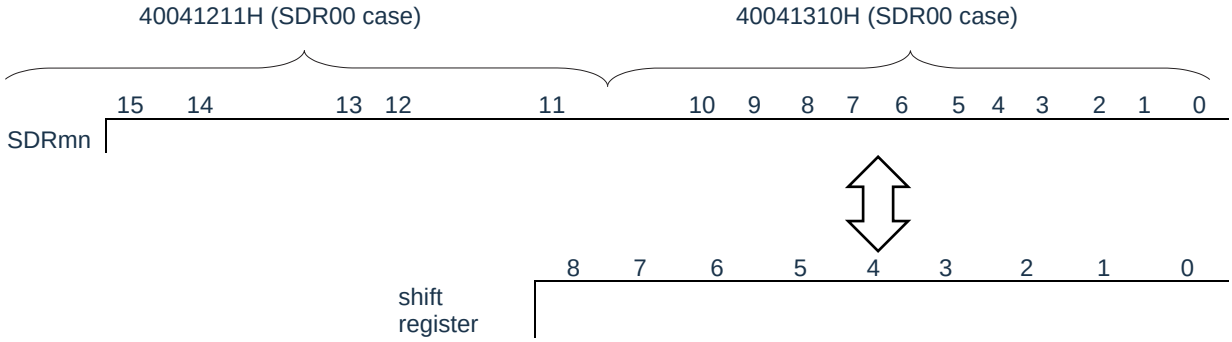
Note 1. After receiving, the bits in bit0~8 that exceed the data length are "0".

2.m: Cell number (m=0,1)n: Channel Number (n=0~3) p: SSPI Number (p=00,01,10,11,20,21)

q:UART (q=0~2)r:IIC (r=00,01,10,11,20,21)

Figure 19-3 Format of serial data register mn(SDRmn) (mn=00,01,10,11)

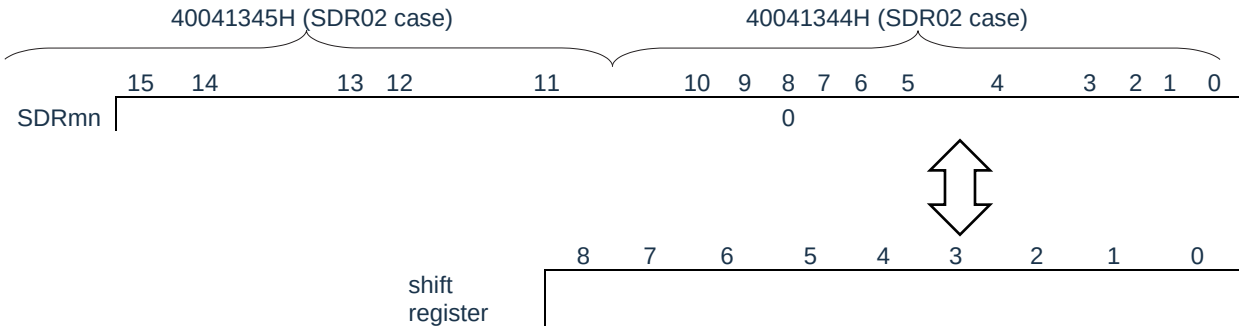
Address: 40041310H (SDR00), 40041312H (SDR01) Reset:0000H R/W
40041748H (SDR.10), 4004174AH(SDR.11)



Remark Refer to "19.3 Register to Control Universal Serial Communication Units" for the SDRmn register's high 7 bits.

Figure 19-4 Format of serial data register mn(SDRmn) (mn=02,03,10,11,12,13)

Address: 4004134H (SDR02), 40041346H (SDR03) Reset:0000H R/W



Note bit8 must be '0'.

Refer to "19.3 Register to Control Universal Serial Communication Units" for the SDRmn register's high 7 bits.

19.3 Register 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 Run Set-up Register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SDIRmn)
- Serial state register mn (SSRmn)
- Serial channel start register m (SSm).
- Serial channel stop register m (STm).
- Serial channel allows state register m (SEm).
- Serial output allows register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOm).
- Input switch control register (ISC)
- Noise filter allows register 0 (NFEN0).
- Port input mode register (PIMx)
- Port output mode register (POMx)
- Port Mode Register (PMx)
- Port register (Px)

Remark m: Cell number (m=0,1)n: Channel number (n=0~3)

19.3.1 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.

bit2 (SCI 0EN) must be set to '1' when using Universal Serial Communication Unit.

bit3 (SCI 1EN) must be set to "1" when using Universal Serial Communication Unit.

The PER0 register is set by an 8-bit memory operation instruction.

After the reset signal is generated, the value of the PER0 register changes to '00H'.

Figure 19-5 Format of Peripheral Admission Register 0 (PER0)

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

symbol	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDEN	ADCEN	IICA0EN	SCI 1EN	SCI0 EN	CAN0EN	TM40EN

SCIMEN	Control of an input clock of a universal serial communication unit m is provided
0	Stop provide an input clock. · Cannot write the SFR used by the universal serial communication unit m. · The universal serial communication unit m is in a reset state.
1	Allows providing an input clock. · SFR capable of reading and writing the UIC unit m.

Note 1. To set up the universal serial communication unit m, the following register must be set in the SCImEN bit "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 values (except input switch control register (ISC), noise filter permit register 0 (NFEN0), port input mode register (PIMx), port output mode register (POMx), port mode register (PMx), port mode control register (PMCx) and port register (Px)).

· Serial clock selection register m (SPSm).
· Serial mode register mn (SMRmn)
· Serial communication run set-up register mn (SCRmn).
· Serial data register mn (SDRmn)
· Serial flag clear trigger register mn (SIRmn).
· Serial state register mn (SSRmn).
· Serial channel start register m (SSm).
· Serial channel stop register m (STm).
· Serial channel allows status register m (SEm).
· Serial output allows register m (SOEm).
· Serial output level register m (SOLm).
· Serial output register m (SOm).
· Serial standby control register m (SSCm).

19.3.2 Serial clock selection register m (SPSm)

The SPSm register is a 16-bit register that selects two common run-time clocks (CKm0, CKm1). CKm1 is selected by bit7~4 of the SPSm register, and CKm0 is selected by bit3~0.

Prevents the SPSm register from being overridden during (SEmn=1).

The SPSm register is set by the 16-bit memory operation instruction.

A lower 8-bit of the SPSm register can be set with the SPSmL and through an 8-bit memory operation instruction.

After the reset signal is generated, the value of the SPSm register changes to "0000H".

Figure 19-6 Format of Serial Clock Selection Register m(SPSm)

Addresses: 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	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection Note for Runtime Clock (CKmk)					
					f_{CLK}=2MHz	f_{CLK}=5MHz	f_{CLK}=10MHz	f_{CLK}=20MHz	f_{CLK}=48MHz
0	0	0	0	f_{CLK}	2MHz	5MHz	10MHz	20MHz	48MHz
0	0	0	1	$f_{CLK}/2$	1MHz	2.5MHz	5MHz	10MHz	24MHz
0	0	1	0	$f_{CLK}/2^2$	500kHz	1.25MHz	2.5MHz	5MHz	12MHz
0	0	1	1	$f_{CLK}/2^3$	250kHz	625kHz	1.25MHz	2.5MHz	6MHz
0	1	0	0	$f_{CLK}/2^4$	125kHz	313kHz	625kHz	1.25MHz	3MHz
0	1	0	1	$f_{CLK}/2^5$	62.5kHz	156kHz	313kHz	625kHz	1.5MHz
0	1	1	0	$f_{CLK}/2^6$	31.3kHz	78.1kHz	156kHz	313kHz	750kHz
0	1	1	1	$f_{CLK}/2^7$	15.6kHz	39.1kHz	78.1kHz	156kHz	375kHz
1	0	0	0	$f_{CLK}/2^8$	7.81kHz	19.5kHz	39.1kHz	78.1kHz	187.5kHz
1	0	0	1	$f_{CLK}/2^9$	3.91kHz	9.77kHz	19.5kHz	39.1kHz	93.75kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95kHz	4.88kHz	9.77kHz	19.5kHz	46.9kHz
1	0	1	1	$f_{CLK}/2^{11}$	977Hz	2.44kHz	4.88kHz	9.77kHz	23.4kHz
1	1	0	0	$f_{CLK}/2^{12}$	488Hz	1.22kHz	2.44kHz	4.88kHz	11.7kHz
1	1	0	1	$f_{CLK}/2^{13}$	244Hz	610Hz	1.22kHz	2.44kHz	5.85kHz
1	1	1	0	$f_{CLK}/2^{14}$	122Hz	305Hz	610Hz	1.22kHz	2.92kHz
1	1	1	1	$f_{CLK}/2^{15}$	61Hz	153Hz	305Hz	610Hz	1.46Hz

Note To change the clock selected as fCLK during the Universal Serial Communication Unit (SCI) run (change the value of the System Clock Control Register (STm) =000FH).

Note bit15~8 must be placed at 0.

Note 1.f_{CLK}: Clock frequency of the CPU/peripheral hardware

2.m: Cell Number (m=0,1)

3.k=0,1

19.3.3 Serial mode register mn (SMRmn)

The SMRmn register is a register for setting the channel n operation mode, selecting the fMCK, specifying whether the serial clock can be used for ^{fSCLK} input. In addition, an inversion level of the received data is set only in the UART mode.

Prevents the SMRmn register from being overwritten during the run (SEmn=1), but MDmn0 bits can be overwritten during the run.

The SMRmn register is set by the 16-bit memory operation instruction.

After the reset signal is generated, the value of the SMRmn register changes to "0020H".

Figure 19-7 Format of serial mode register mn (SMRmn) (1/2)

Address: 40041110H(SMR00)~40041116H(SMR03) Reset:0020H R/W
40041550H (SMR.10)~40041552H (SMR.11)

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmangane	CCSmangan	0	0	0	0	0	STSm Note 1	0	SISmn0 Note 1	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	Channel n Runtime Clock (fMCK) Selection
0	Runtime Clock CKm0 for SPSm Register Settings
1	Runtime Clock CKm1 for SPSm Register Settings
The runtime clock (fMCK) is used for edge detection circuits. A transmission clock (fTCLK) is generated by setting the CCSmn bit and the SDRmn register's high 7 bits.	

CCSmn	Selection of channel n transfer clock (fTCLK)
0	The CKSmn bit-specified runtime clock fMCK's split clock
1	Input clock fSCLK from SCLKp pin (slave transfer of SSPI mode)
The transfer clock fTCLK is used for a shift register, a communication control circuit, an output controller, an interrupt control circuit, and an error control circuit. When the CCSmn bit is	

STSmn Note 1	Start triggering source selection
0	Only software triggers are valid (selected when SSPI, UART send, simple I2C).
1	Valid edge of RxDq pin (selected when UART receives)
When the above condition is satisfied after the SSm register is set "1", the transfer is started.	

Note1. Limited to SMR01, SMR03, SMR11 registers only.

Note The bit13~9,7,4,3 (SMR00, SMR02, SMR10 registers must be bit13~6,4,3) set "0" and the bit5 set "1".

Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) p: SSPI Number (p=00,01,10,11,20,21) q:UART (q=0~2)r:IIC (r=00,01,10,11,20,21)

Figure 19-7 Format of serial mode register mn (SMRmn) (2/2)

Address: 40041110H(SMR00)~40041116H(SMR03) Reset:0020H R/W
40041550H (SMR.10)~40041552H (SMR.11)

symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mangane se	CCS mangan ese	0	0	0	0	0	STS mn Note 1	0	SISmn0 Note 1	1	0	0	MD mn2	MD mn1	MD mn0

SISmn0 Note 1	Level inversion control of channel n receiving data in UART mode
0	The descent edge is detected as the starting position.
1	Detect the rising edge as the starting position. The input communication

MDmn2	MDmn1	Settings for Channel n Running Mode
0	0	SSPI mode
0	1	UART mode
1	0	Simple I²C mode
1	1	Disable from setting.

MDmn0	Selection of Channel n Interrupt Source
0	End of Transfer Interrupt
1	buffer air break (Occurs when data is transferred from a SDRmn register to a shift register)
When sending continuously, if the MDmn0 bit is "1" and SDRmn's data is empty, write the next sending data.	

Note1. Only SMR01, SMR03, SMR11, register.

Note The bit13~9,7,4,3 (SMR00, SMR02, SMR10 registers must be bit13~6,4,3) set "0" and the bit5 set "1".

Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) p: SSPI Number (p=00,01,10,11,20,21) q:UART (q=0~2)r:IIC (r=00,01,10,11,20,21)

19.3.4 Serial Communication Run Set-up Register mn (SCRmn)

The SCRmn register is a communication operation setting register of channel n, setting data sending and receiving mode, data and clock phase, whether shielding error signal, parity check bit, start bit, stop bit and data length.

Prevents the SCRmn register from being overridden during (SEmn=1).

The SCRmn register is set by the 16-bit memory operation instruction.

After the reset signal is generated, the value of the SCRmn register changes to "0087H".

Figure 19-8 Serial Communication Run Set-up Register mn (SCRmn) Format (1/2)

Address: 40041118H(SCR00)~4004111EH(SCR03) 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 mangan ese	RXE mangan ese	DAP mangan ese	CKP mangane se	0	EOC mangan ese	PTC mn1	PTC mn0	DIR mangan ese	0	SLCmn1 Note 1	SLC mn0	0	1	DLSmn1 Note 2	DLS mn0

TXEmn	RXEmn	Settings for Channel n Running Mode
0	0	No communication.
0	1	Enable receipt.
1	0	Enable to send.
1	1	Enables sending and receiving.

DAPmn	CKPmn	data and clock phase selection in SSPI mode	Type
0	0	SCLKp SDOp D7 D6 D5 D4 D3 D2 D1 D0 SDIp input timing sequence	1
0	1	SCLKp SDOp D7 D6 D5 D4 D3 D2 D1 D0 SDIp input timing sequence	2
1	0	SCLKp SDOp D7 D6 D5 D4 D3 D2 D1 D0 SDIp input timing sequence	3
1	1	SCLKp SDOp D7 D6 D5 D4 D3 D2 D1 D0 SDIp input timing sequence	4
in UART mode and simple I2C mode, must set DAPmn bit and CKPmn bit both to 0.			

EOCmn	Masking control for error interrupt signals (INTSREx (x=0~3))
0	Prevents the generation of an error interrupt INTSREx (INTSRx generation).
1	Allow error-generated interrupt INTSREx (INTSRx is not generated when an error occurs).
In SSPI mode and simple I^2C mode or when UART is sent, the EOCmn position "0" note 3.	

Note 1. Limited to SCR00, SCR02, SCR10 registers only.

2. Limited to SCR00 and SCR01 registers, other fixed to "1".

3. An error interrupt INTSREn may occur when the EOCmn bit is "0" and SSPImn is not used.

Note bit3, 6, 11 must be set to '0' (bit5 of the SCR01, SCR03, SCR11 register must also be set to '0') and bit2 to '1'.

Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) p: SSPI Number (p=00,01,10,11,20,21)

Figure 19-8 Serial Communication Run Set-up Register mn (SCRmn) Format (2/2)

Address: 40041118H(SCR00)~4004111EH(SCR03) 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	TXEman	RXEman	DAPman	CKPman	0	EOCman	PTCmn1	PTCmn0	DIRman	0	SLCmn1 Note 1	SLCmn0	0	1	DLSmn1 Note 2	DLSmn0

PTCmn1	PTCmn0	Settings for parity bits in UART mode	
		Send	Receive
0	0	No parity bits are output.	No parity is received.
0	1	Output parity note 3.	Parity is not judged.
1	0	Output even check.	Parity check.
1	1	Output odd check.	Judge odd check.
In SSPI mode and simple I2C mode, both PTCmn1 bits and PTCmn0 bits must be set to "0".			

DIRmn	Selection of data transfer order in SSPI and UART mode
0	MSB-first input/output.
1	LSB-preferred input/output.
In simple I2C mode, the DIRmn position must be "0".	

SLCmn1 Note 1	SLCmn0	Settings for stop bits in UART mode
0	0	No stop bit
0	1	stop bit length=1 bit
1	0	Stop bit length=2 bits (mn=00, 02,10 only).
1	1	Disable from setting.
If an end-of-transfer interrupt is selecte, an interrupt is generated aft all stop bits have been transfer. Must be set to 1 stop bit (SLCmn1, SLCmn0=0, 11) at UART receive or in easy I2C mode. In SSPI mode, must be set to no stop bit (SLCmn1, SLCmn0=0,0). When UART is sent, it must be set to 1 bit (SLCmn1, SLCmn0=0, 1) or 2 bit (SLCmn1, SLCmn0=1, 0).		

DLSmn1 Note 2	DLSmn0	Data length settings in SSPI and UART mode
0	1	9-bit data length (bit0~8 stored in the SDRmn register) (optional only in UART mode)
1	0	7-bit data length (bit0~6 stored in the SDRmn register)
1	1	8-bit data length (bit0~7 stored in SDRmn register).
Others		Disable from setting.
In simple I2C mode, you must set both the DLSmn1 bit and the DLSmn0 bit to 1.		

Note 1. Limited to SCR00, SCR02, SCR10 registers only.
2. Limited to SCR00 and SCR01 registers, other fixed to "1".
3. Always attach a "0" regardless of the content of the data.

Note bit3, 6, 11 must be set to '0' (bit5 of the SCR01, SCR03, SCR11 register must also be set to '0') and bit2 to '1'.

Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) p: SSPI Number (p=00,01,10,11,20,21)

19.3.5 Serial data register mn (SDRmn)

The SDRmn register is a data register (16-bit) sent and received by the channel n.

bit8~0 (Low 9 bits) for SDR00, SDR01 or bit7~0 (Low 8 bits) for SDR02, SDR03, SDR10, SDR11 serve as transmit and receive buffer registers, and bit15~9 (High 7 bits) serve as a fractional setting register for runtime clock (fMCK).

If the CCSmn position "0" of the serial mode register mn (SMRmn), the frequency division clock of the running clock set by bit15~9 (High 7 bits) of the SDRmn register is used as the transfer clock.

If CCSmn position" 1", then bit15~9 (7th higher) of SDRmn must be set to "0000000B. The input clock fSCLK of the SCLKp pin is the transfer clock.

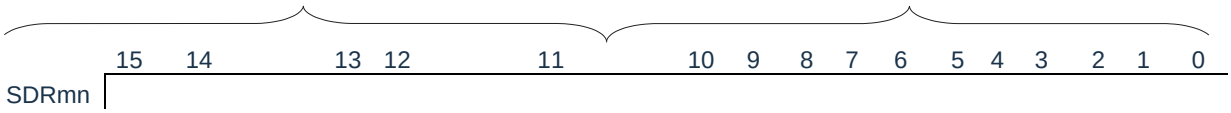
The low 8-bit or low 9-bit of the SDRmn register is used as a transmit and receive buffer register. store that parallel data converted by the shift register to a low 8-bit or a low 9-bit when receiving the data; When transmitting data, the transmission data to be transmitted to the shift register is set to a low 8-bit or a low 9-bit.

The SDRmn register can be read and written in units of 16 bits. However, you can read and write 7-bits high only if the idles (SEmn=0). In operation (SEmn=1), only low 8 bits or low 9 bits of the SDRmn register can be written, and the read value of high 7 bits of the SDRmn register is always "0"

After the reset signal is generated, the value of the SDRmn register changes to "0000H".

Figure 19-9 Format of Serial Data Register mn(SDRmn)

Address: 40041310H (SDR00), 40041312H (SDR01) Reset:0000H R/W
40041748H (SDR.10), 4004174AH(SDR.11)



Address: 4004134H (SDR02), 40041346H (SDR03) Reset:0000H R/W

SDRmn [15:9]							Transmit clock setting for operating clock
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

Note 1. The bit8 of the SDR02, SDR03, SDR10, SDR11 register must be set to "0".

2. When using UART, Disable from setting SDRmn[15:9] to "0000000B" and "0000001B".

3. When using Simple I2C, Disable from setting SDRmn[15:9] to "0000000B", setting value for SDRmn[15:9] must be greater than or equal to "0000001B.

4. When running stop (SEmn=0), disable the SDRmn[7:0] rewrite via 8-bit memory operation instructions (otherwise SDRmn[15:9] is all cleared).

Note 1. Refer to "19.2 Structure of Universal Serial Communication Unit" for low 8-bit or low 9-bit functionality.

2.m: Cell number (m=0,1)n: Channel number (n=0~3)

19.3.6 Serial flag clear trigger register mn (SIRmn)

This is the trigger register used to clear the error flags of channel n.

If the positions (FECTmn, PECTmn, OVCTmn) are set to "1", the corresponding positions (FEFmn, PEFmn, OVFmn) of the serial state register mn (SSRmn) are cleared to "0". Because the SDIRmn register is a trigger register, if you clear the corresponding bits of the SSRmn register, the SDIRmn register is also cleared immediately.

The SIRmn register is set by the 16-bit memory operation instruction.

A lower 8-bit of the SIRmn register can be set with the SIRmnL and through an 8-bit memory operation instruction.

After the reset signal is generated, the value of the SIRmn register changes to "0000H".

Figure 19-10 Format of serial flag purge trigger register mn (SIRmn)

Address: 40041108H(SIR00)~4004110EH(SIR03) 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	FECTmn Note1	PECTmn	OVCTmn

FECTmn Note 1	Clear trigger for channel n frame error flag
0	Do not clear.
1	Clear the FEFmn bit of the SSRmn register "0".

PECTmn	Purge trigger for channel n parity error flag
0	Do not clear.
1	Clear the PEFmn bit of the SSRmn register "0".

OVCTmn	Clear trigger for channel n overflow error flag
0	Do not clear.
1	Clear the OVFmn bit of the SSRmn register "0".

Note1. Limited to SIR01, SIR03, SIR11 registers only.

Note bit15~3 (SIR00, SIR02, SIR10 register bit15~2) must be set to"

Note 1.m: Cell number (m=0,1)n: Channel number (n=0~3)

2. The read value of the SIRmn register is always "0000H".

19.3.7 Serial state register mn (SSRmn)

The SSRmn register indicates the communication state of the channel n and the occurrence of an error. The errors represented are frame errors, parity errors, and overflow errors. The SSRmn register is read by a 16-bit memory operation instruction.
The lower 8 bits of the SSRmn register can be read with the SSRmnL and through the 8-bit memory operation instruction.
After the reset signal is generated, the value of the SSRmn register changes to "0000H".

Figure 19-11 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 note 1	PEF mn	OVF mn

TSFmn	Flag for channel n communication state
0	Communication Stop State or Communication Standby State
1	Communication Operational Status

[Purge Criteria]
· When the STmn position "1" of the STm register or the SSmn position "1" of the SSm register
· When communication ends
[Placement Criteria]
· When communication begins

BFFmn	Status Indication Flag for Channel n Buffer Register
0	The SDRmn register does not hold valid data.
1	The SDRmn register store valid data.

[Purge Criteria]
· Transfer the transmission data from the SDRmn register to the shift register during transmission
· Read out received data from the SDRmn register during the receiving process
· When the STmn location "1" of the STm register or the SSmn location "1" of the SSm register
[Placement Criteria]
· Write transmit data to the SCRmn register with the TXEmn bit of the SDRmn register "1"
· When saving received data to the SCRmn register with the RXEmn bit of the SDRmn register "1" (Receive, Send and Receive modes in each communication mode)
· When a receive error occurs

Note1. Limited to SSR01, SSR03, SSR11 registers only.

Remark m: Cell number (m=0,1)n: Channel number (n=0~3)

Figure 19-11 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 mangan	BFF mangan	0	0	FEF mn note 1	PEF mangan	OVF mangan

FEFmn Note 1	Detection flag for channel n frame errors
0	No error occurred.
1	An error occurred (when UART received).
[Purge Criteria] · When writing "1" to the FECTmn bit of the SIRmn register [Placement Criteria] · When no stop bit is detected at the end of UART reception	

PEFmn	Detection flag for channel n parity error
0	No error occurred.
1	An error occurred (when UART received) or ACK was not detected (when I^2C sent).
[Purge Criteria] · When writing "1" to the PECTmn bit of the SIRmn register [Placement Criteria] · When sending data with different parity and parity bits (parity errors) at the end of UART reception · At the time of I2C transmission and at the time of ACK reception the slave did not return an ACK signal (no ACK was detected)	

OVFmn	Detection flag for channel n overflow error
0	No error occurred.
1	An error has occurred.
[Purge Criteria] · When writing "1" to the OVCTmn bit of the SIRmn register [Placement Criteria] · In the SCRmn register, the RXEmn bit is "1" (reception mode, transmission and reception mode in each communication mode) · Data is not ready to be sent during a slave send or slave send and receive in SSPI mode	

Note1. Limited to SSR01, SSR03, SSR11 registers only.

Note 1. If you write the SDRmn register when the BFFmn bit is "1", the saved send or receive data is corrupted and an overflow error is detected (OVEmn=1).

Remark m: Cell number (m=0,1)n: Channel number (n=0~3)

19.3.8 Serial channel start register m (SSm).

The SSm register is a trigger register that sets a communication/start count for each channel.

If you write "1" to each (SSmn), set "1" to the corresponding bit (SEmn) of the serial channel allowed register m (SEm). Because the SSmn bit is a trigger bit, if the SEmn bit is "1", clear the SSmn bit immediately.

The SSm register is set by a 16-bit memory operation instruction.

A lower 8-bit of the SSm register can be set with the SSmL and through an 8-bit memory operation instruction.

After the reset signal is generated, the value of the SSm register changes to "0000H".

Figure 19-12 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

SSm	Trigger for channel n run start
0	No trigger.
1	Move the SEmn location "1" to the Communication Standby State Note.

Note If the SSmn position "1" is used during the communication, the communication is stopped and goes into standby. At this time, the value of the control register and the shift register, the SCLKmn pin and the SDOmn pin, the FEFmn flag, the PEFmn flag, and the OVFmn flag remain in state.

Note 1. bit15~4 of SS0 register and bit15~2 of SS1 register must be set to '0'.

2. At the time UART is received, at least 4 fMCK clocks must be separated after RXEmn position "1" of the SCRmn register and then SSmn.

Note 1.m: Cell number (m=0,1)n: Channel number

(n=0~3)

2.SSm register always reads '0000H'.

19.3.9 Serial channel stop register m (STm).

The STm register is a trigger register that sets a communication/stop count that allows each channel.

If you write "1" to each (STmn), clear "0" (SEmn) of the Serial Channel Allowed Status Register m (SEm). Because the STmn bit is a trigger bit, if the SEmn bit is "0", clear the STmn bit immediately.

The STm register is set by a 16-bit memory operation instruction.

A lower 8-bit of the STm register can be set with the STmL and through an 8-bit memory operation instruction.

After the reset signal is generated, the value of the STm register changes to "0000H".

Figure 19-13 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 trigger.
1	Clear the SEmn bit "0" to stop the communication run note.

Note The value of the control and shift registers, the SCLKmn and SDOmn pins, and the FEFmn, PEFmn, and OVFmn flags remain in state.

Note bit15~4 of ST0 register and bit15~2 of ST1 register must be set to'0'.

Note 1.m: Cell number (m=0,1)n: Channel number

(n=0~3)

The read value of the 2.STm register is always "0000H".

19.3.10 Serial channel allows state register m (SEm).

The SEm register is used to confirm the allowed or stopped states of serial transmission and reception of each channel.

If the bits of register m (SSm) are given permission to start a serial, their corresponding position 1. If the bits of the serial channel stop register m (STm) are written "1", their corresponding bits are "0".

For a channel n, the value of the CKOmn bit (serial clock output of channel n) of the subsequent serial output register m (SOm) cannot be overridden by software.

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, any waveform such as a start condition or a stop condition can be generated by software.

The SEm register is read by a 16-bit memory operation instruction.

The lower 8 bits of the SEm register can be read with the SEmL and through the 8-bit memory operation instruction.

The value of the SEm register changes to "0000H" after the reset signal is generated.

Figure 19-14 Format of Serial Channel Allowed State 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

SEm	A representation of the allowed or stopped state of operation of channel n
0	Idle Status
1	Run Allowed Status

Remark m: Cell number (m=0,1)n: Channel number (n=0~3)

19.3.11 Serial output allows register m (SOEm)

The SOEm register setting allows or stops 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) cannot be overridden by software.

For a channel n that stops 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, any waveform such as a start condition or a stop condition can be generated by software.

The SOEm register is set by the 16-bit memory operation instruction.

A lower 8-bit of the SOEm register can be set with the SOEmL and through an 8-bit memory operation instruction.

After the reset signal is generated, the value of the SOEm register changes to "0000H".

Figure 19-15 Serial output allows the format of 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	SOE03	SOE02	SOE01	SOE0

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	SOE11	SOE10

SOEman	Allow or stop of channel n serial output
0	Stop the output of serial communication.
1	The output that allows serial communication.

Note bit15~4 of SOE0 register and bit15~2 of SOE1 register must be set to'0'.

Remark m: Cell number (m=0,1)n: Channel number (n=0~3)

19.3.12 Serial output register m (SOm).

The SOm register is a buffer register for serial output of each channel.

The value of the SOmn bit of this register is output from the serial data output pin of channel n.

The value of the CKOmn bit of this register is output from the serial clock output pin of channel n.

The SOmn bit of this register can be overwritten by software only if serial output is prohibited (SOEmn=0). When serial output (SOEmn=1) is allowed, the value of the SOmn bit of this register can only be changed by serial communication, ignoring the rewriting of the software.

The CKOmn bit of this register can be overridden by software only if the channel is stopped running (SEmn=0). When the channel is allowed to run (SEmn=1), the value of the CKOmn bit of this register can only be changed by serial communication, ignoring the rewriting of the software.

To use the pin of a serial interface as a non-serial interface function such as a port function, you must have the corresponding CKOmn bit and SOmn position.

The SOm register is set by a 16-bit memory operation instruction.

After the reset signal is generated, the value of the SOm register changes to "0F0FH".

Figure 19-16 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	CKO03	CKO02	CKO01	CKO00	0	0	0	0	SO03	SO02	SO01	SO00

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	CKO11	CKO10	0	0	0	0	0	0	SO11	SO10

CKOman	Serial clock output of channel n
0	The output value of the serial clock is "0".
1	The output value of the serial clock is "1".

SOman	Serial data output of channel n
0	The output value of serial data is "0".
1	The output value of serial data is "1".

Note bit15~12 and bit7~4 of the SO0 register.
bit15~10 and bit7~2 of the SO1 register must be set to 0.

Remark m: Cell number (m=0,1)n: Channel number (n=0~3)

19.3.13 Serial output level register m (SOLm)

The SOLm register is a register that sets the reverse phase of the data output level of each channel.

This register can only be set in UART mode. In SSPI mode and simple I^2C mode, the corresponding position must be '0'. Only when serial output is allowed (SOEmn=1) is the channel n inversion setting of this register reflected to the pin output. When you disable serial output (SOEmn=0), the value of the SOmn bit is output directly. Prevents the SOLm register from being overridden during (SEmn=1).

The SOLm register is set by the 16-bit memory operation instruction.

A lower 8-bit of the SDOLm register can be set with the SOLmL and through an 8-bit memory operation instruction.

After the reset signal is generated, the value of the SOLm register changes to "0000H".

Figure 19-17 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	SOL02	0	SOL00

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	SOL10

SOLman	Selection of channel n transmission data level inversion in UART mode
0	The communication data is output directly.
1	Invert the communication data.

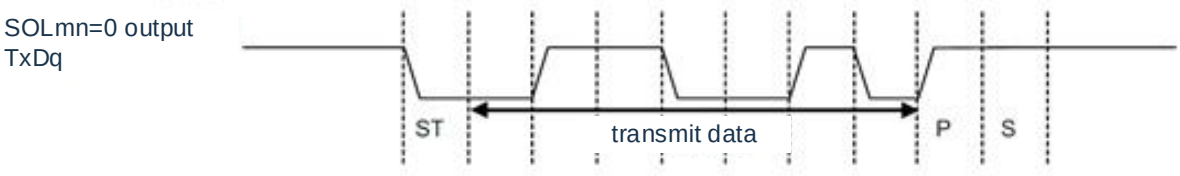
Note bit15~3 of SOL0 register and bit15~1 of bit1 and SOL1 register must be set to'0'.

Remark m: Cell number (m=0,1)n: Channel number (n=0,2)

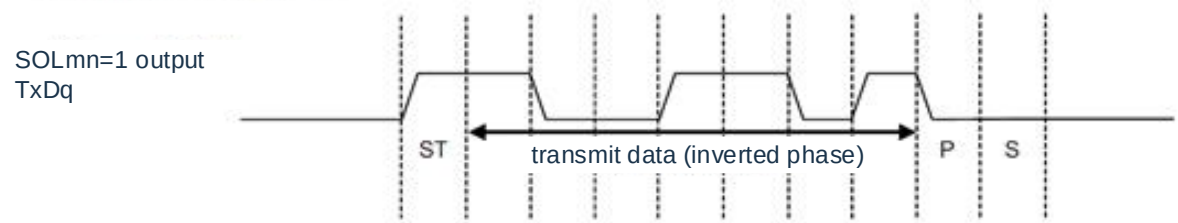
When UART transmission is performed, a level inversion example of the transmitted data is shown in FIG. 19-18.

Figure 19-18 Example of Level Inversion for Transmitting Data

(a)positive phase output (SOLmn=0)



(b)inverted phase output (SOLmn=0)



Remark m: Cell number (m=0,1)n: Channel number (n=0,2)

19.3.14 Input switch control register (ISC)

When LIN-bus communication is implemented by UART0, the ISC1 bit and ISC0 bit of the ISC register are used for external interrupts and coordination of timer array units. If bit0 is set to '1', the input signal of the serial data input (RxD0) pin is selected as the input of the external interrupt (INTP0), thereby detecting the wake-up signal by INTP0 interrupt.

If bit1 is set to '1', the input signal of the RxD0 pin is selected as the input of the timer.

The SS1E00 bit controls the SS00 pin input for channel 0 in the dependent mode of SSPI00 communication. During the period of inputting a high level to the SS00 pin, no transmission and reception are performed even if a serial clock is inputted; During the period of inputting a low level to the SS00 pin, if a serial clock is input, transmission and reception are performed according to the settings of each mode.

The ISC register is set by an 8-bit memory operation instruction.

After the reset signal is generated, the value of the ISC register changes to "00H".

Figure 19-21 Enter the format of the Switch Control Register (ISC)

Location: 40040473H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
ISC	SIE00	0	0	0	0	0	ISC1	ISC0

SDIE00	Settings for SS00 input for channel 0 in dependent mode of SSPI00 communication
0	Invalid SS00 pin input.
1	SS00 pin input is valid.

ISC1	Input switching of channel 3 of timer Timer4
0	Use the input signal of the TI03 pin as the input (usually running) of the timer.
1	The input signal of the RxD0 pin is used as the input of the timer (detecting the wake-up signal and measuring the low level width of the interval segment and the pulse width of the synchronization segment).

ISC0	External Interrupt (INTP0) input switch
0	Use the input signal of the INTP0 pin as the input for the external interrupt (usually run).
1	Use the input signal of the RxD0 pin as the input of the external interrupt (detect wake-up signal)

Note bit6~2 must be set to'0'.

19.3.15 Noise filter allows register 0 (NFEN0).

The NFEN0 register sets whether the noise filter is used for the input signal of the serial data input pin of each channel.

For pins for SSPI or simple I2C communications, the corresponding position must be "0" to disable the noise filter. For pins for UART communication, the corresponding position "1" must be made effective for the noise filter.

When the noise filter is effective, the two clocks are detected whether the two clocks are consistent after synchronization by the fMCK (object channel). When the noise filter is not valid, synchronization occurs only through the runtime clock (fMCK) of the object channel.

The NFEN0 register is set by an 8-bit memory operation instruction.

After the reset signal is generated, the value of the NFEN0 register changes to '00H'.

Figure 19-22 Noise filter allows the format of register 0(NFEN0).

Location: 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	Whether the Noise Filter of the RxD2 Pin Is Used or Not
0	noise filter OFF
1	noise filter ON
When used as a RxD2 pin, the SNFEN20 position "1". When used as a function other than the RxD2 pin, the SNFEN20 position '0' must	

SNFEN10	Whether the Noise Filter of the RxD1 Pin Is Used or Not
0	noise filter OFF
1	noise filter ON
When used as a RxD1 pin, the SNFEN10 position "1". When used as a function other than the RxD1 pin, the SNFEN10 position '0' must	

SNFEN00	Whether the Noise Filter of the RxD0 Pin Is Used or Not
0	noise filter OFF
1	noise filter ON
When used as a RxD0 pin, the SNFEN00 position "1". When used as a function other than the RxD0 pin, the SNFEN00 position '0' must	

Note bit7~5,3,1 must be set.

19.3.16 Register for controlling serial input/output pin port function

When using a universal serial communication unit, you must set a port function control register (PMxx), Pxx, PIMxx, POMxx, and PMCxx).

Refer to "2.3.1 Port Mode Register (PMxx)"2.3.2 Port Register (Pxx)", "2.3.4 Port Input Mode Register (PIMxx)", "2.3.5 Port Output Mode Register (POMxx)" and "2.3.6 Port Mode Control Register (PMCxx)" for details.

When using a multiplexed port of a serial data output pin or a serial clock output pin as a serial data output or serial clock output, the bit of port mode control register (PMxx).

In addition, when used in an N-channel drain open output mode, the position of the port output mode register (POMxx) corresponding to each port must be "1".

(Example) When P02 is used as the serial data output, the PMC02 position of port mode control register 0 is "0". Position PM02 of port mode register 0 "0" Position P02 for port register 0 "1.

When using a multiplexed port of a serial data input pin or a serial clock input pin as a serial data input or serial clock input, the corresponding location of port mode register (PMCxx) shall be set to 1, and port mode control register (PMCxx) location set to 0. The bit of the port register (Pxx) can be "0" or "1".

In addition, when used as a TTL input buffer, the location of the port input mode register (PIMxx) corresponding to each port must be "1"

(Example) When P03 is used as a serial data input, the PMC03 position "0" of port mode control register 0. Position PM03 for port mode register 0 at "1. Position P03 of port register 0 or '1'.

19.4 Idle Mode

Each serial interface of the universal serial communication unit has a running stop mode. Serial communication is not possible in the operational stop mode, so power consumption can be reduced. In addition, pin used for serial interface can be used as port function in idle mode.

19.4.1 Unit Stop Operation

Sets the stop run in units by the 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 providing a clock to a hardware that is not in use.

To stop the Universal Serial Communication Unit 0, you must set the bit2(SCI0EN) to "0"; To stop Universal Serial Communication Unit 1, you must set bit3 (SCI1EN) to "0".

Figure 19-23 Setting of Peripheral Allowable Register 0 (PER0) when unit-based stop operation

(a) Peripheral Allowable Register 0 (PER0).....Only the corresponding location "0" of the SCIm will be stopped.

	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDEN	ADCEN	IICA0EN	SCI1EN	SCI0EN	CAN0	TM40EN
	×	×	×	×	0/1	0/1	×	×

Control of SCIm Input Clock
0: Stop providing input clock
1: Provide input clock

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 initial. However, the following registers are excluded:

- Input Switch Control Register (ISC)
- Noise filter allows register 0 (NFEN0).
- Port input mode register (PIMx)
- Port output mode register (POMx)
- Port mode register (PMx)
- Port register (Px)

Remark ×: This is the unused bit of the universal serial communication unit (depending on the setting of other peripheral functions).

0/1: The "0" or "1" is set according to the user.

19.4.2 Stop by Channel

Stop operation by channel via each of the following register settings.

Figure 19-24 The settings for each register when stopping operation by channel

(a) Serial channel stop register m (STm).....This is a register that sets the communication/stop count 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 SEmn bit '0' and stop communication running

※ Because the STmn bit is a trigger bit, clear the STmn bit immediately if the SEmn bit is "0".

(b) Serial channel allowed state register m (SEm).....The register indicates the running or stopping state of data transmission and reception of 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: Idle Status

The ※SEm register is a read-only state register, which stops running through the STm register.
For a channel that has stopped running, the value of the CKOmn bit of the SOm register can be set by software.

(c) Serial output allow register m(SOEm).....This is a register that sets the permission or stop of serial communication output for 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: Stop output by serial communication operation

※ For channels that have stopped serial output, the value of the SOmn 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: Serial clock output value is "1"

1: The output value of serial data is "1"

※ When using the pin corresponding to each channel as a port function, the corresponding CKOmn bit and SOmn position "1" must be used.

Note Is limit to that universal serial communication unit 0.

Note 1.m: Cell number (m=0,1)n: Channel number (n=0~3)
2. □ : Cannot set (set initial value). 0/1: The "0" or "1" is set according to the user.

19.5 Operation of Serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21) Communication with 3 Lines

This is clock synchronization communication through three lines of SCLK and SDO (SDI and SDO).

[Sending and Receiving Data]
· 7-bit or 8-bit data length
· Phase control for transmitting and receiving data
• MSB/LSB First
Choice
[Clock Control]
· Master or subordinate selection
· Phase control of input/output clock
· A transmission period generated by a pre-divider and an intra-channel counter is set.
· maximum transfer rate note

Master Communications: Max.fCLK/2 (SSPI00 only)

Master Communications: Max.fCLK/4
Slave Communication: Max.fMCK/6

[Interrupt Capabilities]
· interrupt delivery end, buffer interrupt
[Error Detection Flag]
· overflow error

Note Must be used within a range that satisfies the $_{SCLK}$ Cycle Time (tKCY) characteristic. Refer to the data guide for details.

Channels 0~3 for SCI0 and 0~1 for SCI1 are channels that support 3-line serial I/O (SSPI00,SSPI01,SSPI10,SSPI11,SSPI20,SSPI21).

Serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21) runs with the following 60 communications:

· master transmission (Reference 19.5.1)
· master receive (Reference 19.5.2)
· Master send and receive (Reference 19.5.3)
· Slave send (Reference 19.5.4)
· slave receive (Reference 19.5.5)
· Slave send and receive (Reference 19.5.6)

19.5.1 master transmission

Master Send is the operation in which BAT32A237 outputs a transfer clock and sends data to other devices.

3-Wire Serial I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
object channel	Channel 0 for SCI0	Channel 1 for SCI0	Channel 2 for SCI0	Channel 3 for SCI0	Channel 0 for SCI1	Channel 1 for SCI1
Pin Used	SLK00, SDO00	SLK01, SDO01	SLK10, SDO10	SLK11, SDO11	SLK20, SDO20	SLK21, SDO21
interrupt	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	Interrupt at that end of the transfer may be selecte (single transfer mode) or buffer air-discontinuity (continuous transfer mode).					
Error detection flag	None					
length of transmit data	7 or 8-bit					
Transfer Rate Note	Max.$_{fCLK}$/2[Hz] (SSPI00 only), $_{fCLK}$/4[Hz] Min.$_{fCLK}$/(2 2^{15} 128) [Hz] $_{fCLK}$: system clock frequency					
data phase	Can be selected by the DAPmn bit of the SCRmn register. ·DAPmn=0: Start the data output when the serial clock starts running. ·DAPmn=1: The data output is started half a clock before the serial clock starts running.					
clock phase	Can be selected by the CKPmn bit of the SCRmn register. • CKPmn=0: prime • CKPmn=1: inversion					
data orientation	MSB First or LSB First					

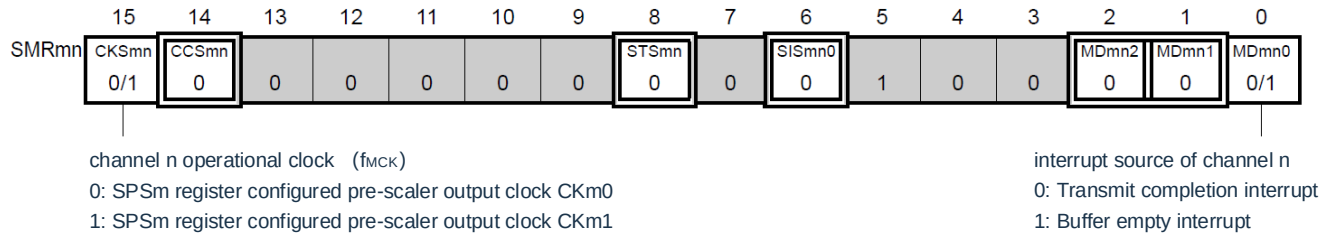
Note It must be used within the scope of peripheral functional characteristics (reference data manual) that meet this condition and electrical characteristics.

Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) mn=00~03,10~11.

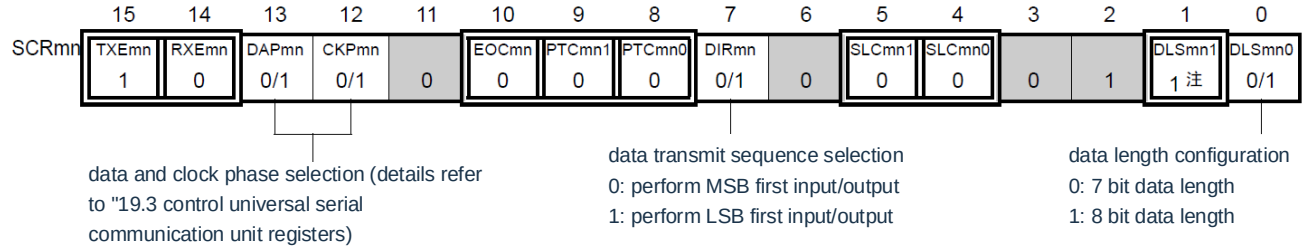
(1) Register settings

Figure 19-25 3-wire Serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20,SSPI21) Example of register setting content when master sending

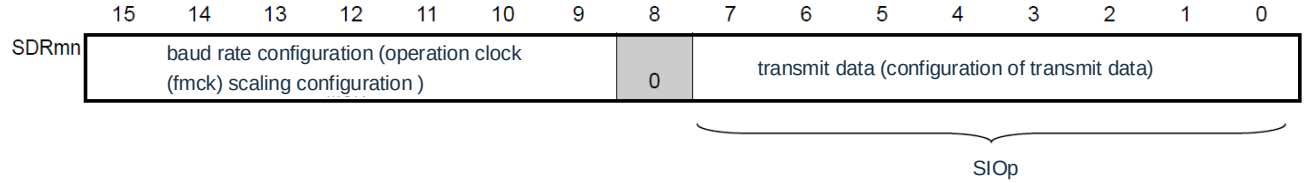
(a) serial mode register mn (SMRmn)



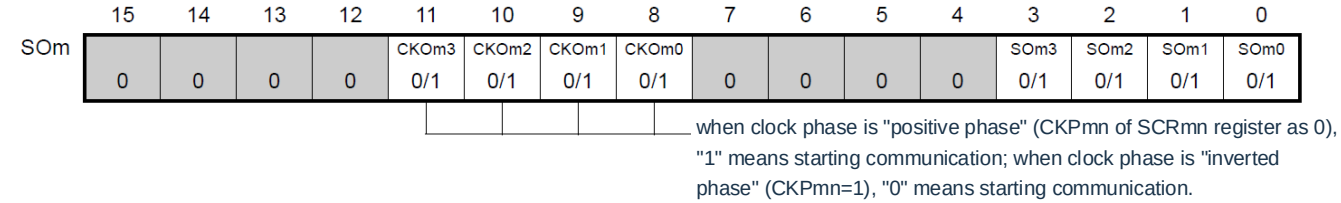
(b) serial communication operation configuration registermn mn(SCRmn)



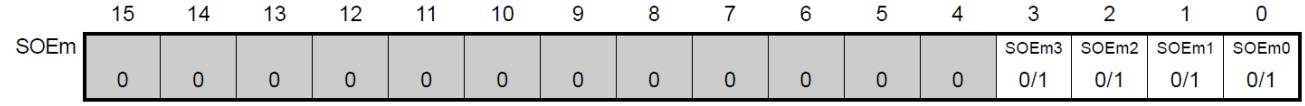
(c) serial data registermn mn(SDRmn)



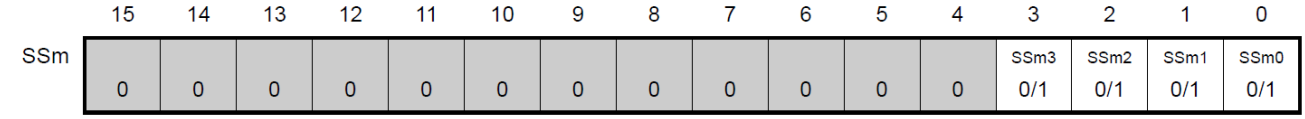
(d) serial output register m(SOm) .....Only configure bit of target channel



(e) serial output enable registerm (SOEm)….only set bit of target channel to 1.



(f) serial channel start registerm (SSm)….only set bit of target channel to 1.



Note: Limited to SCR00, SCR01 registers, other fixed to "1".

Note 1.m: Cell number (m=0,1)n: Channel Number (n=0~3) mn=00~03,10~11.

2. ▢ : Cannot set (set initial value). 0/1: The "0" or "1" is set according to the user.

(2) Procedure

Figure 19-26 Initial set-up steps sent by master

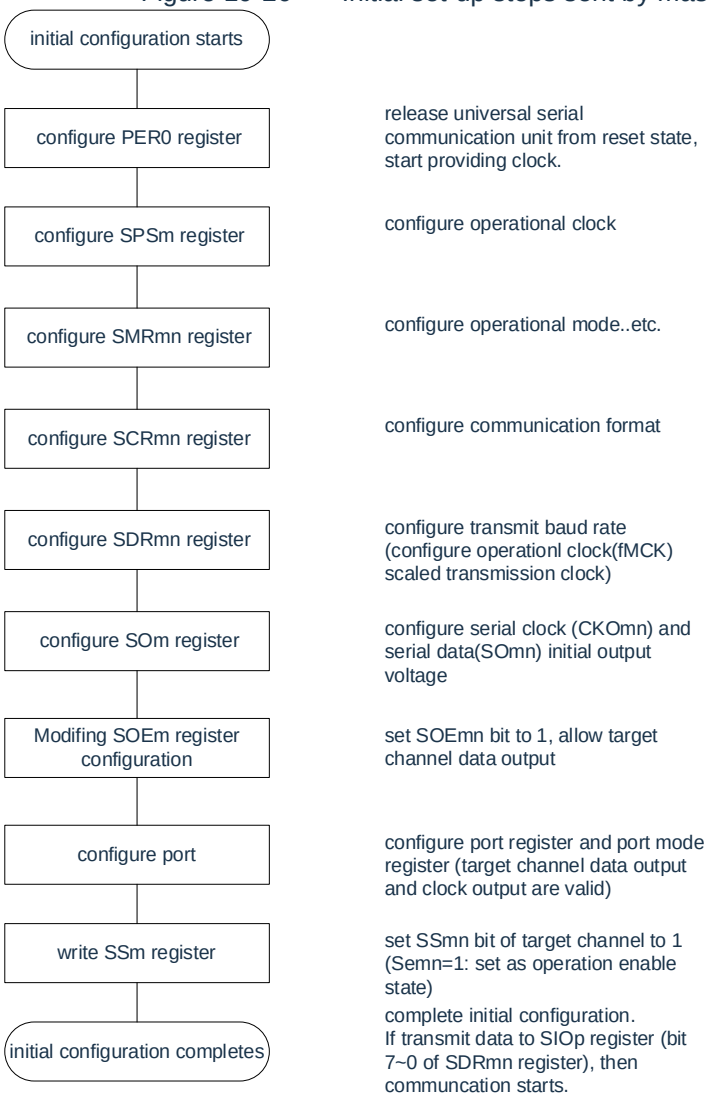


Figure 19-27 Stop step for master send

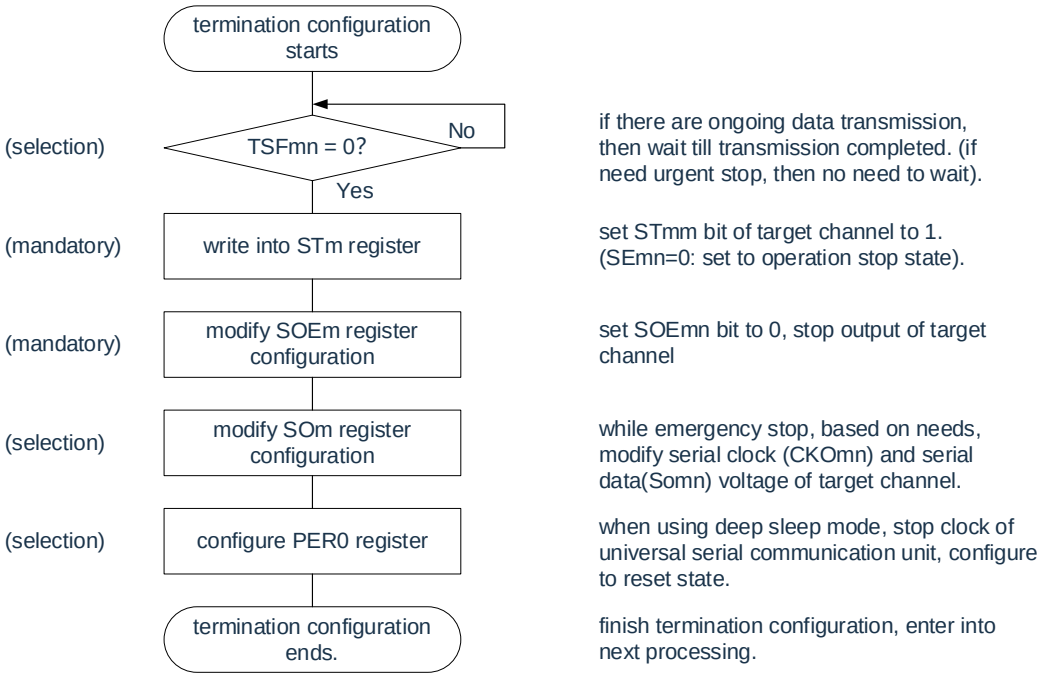
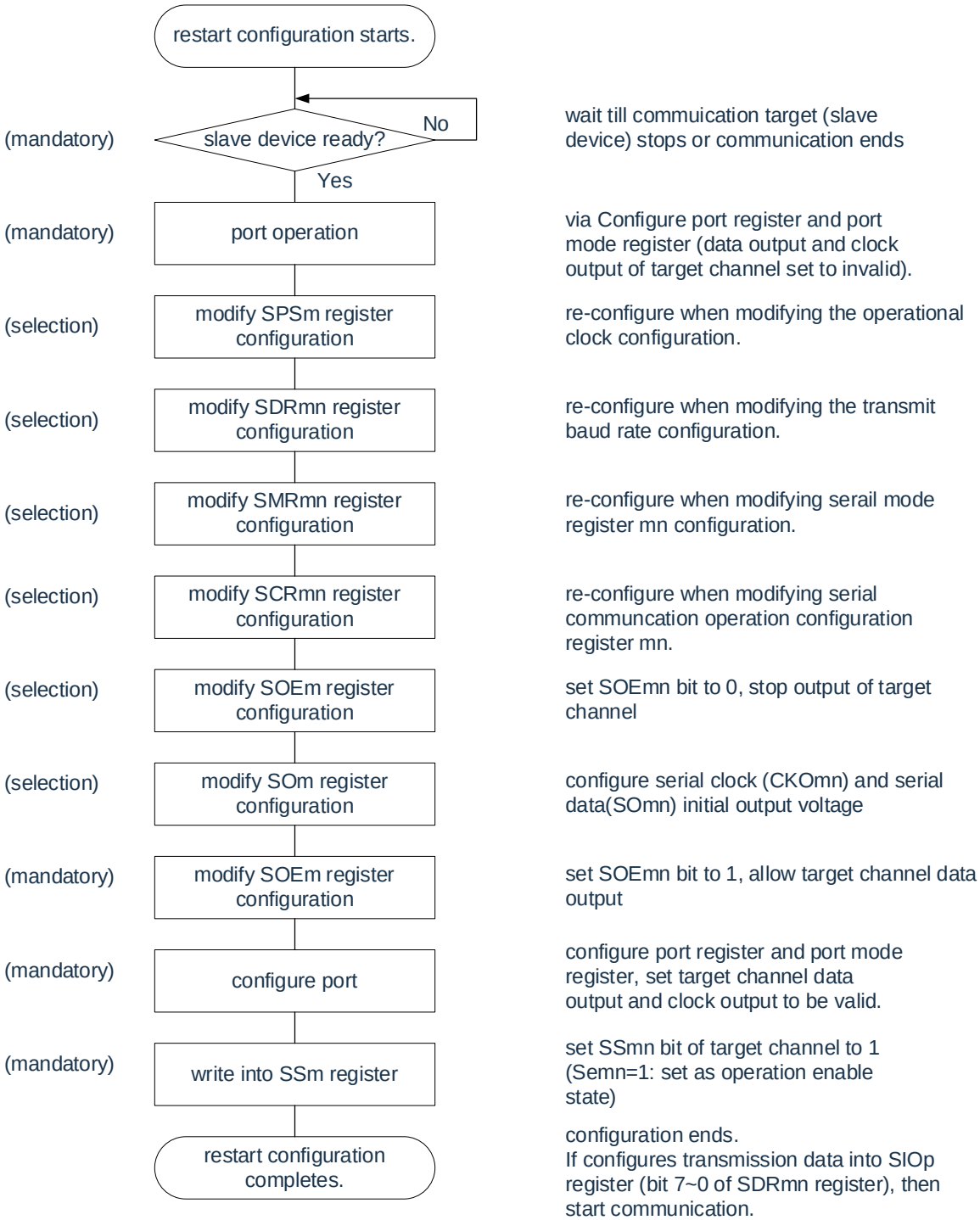


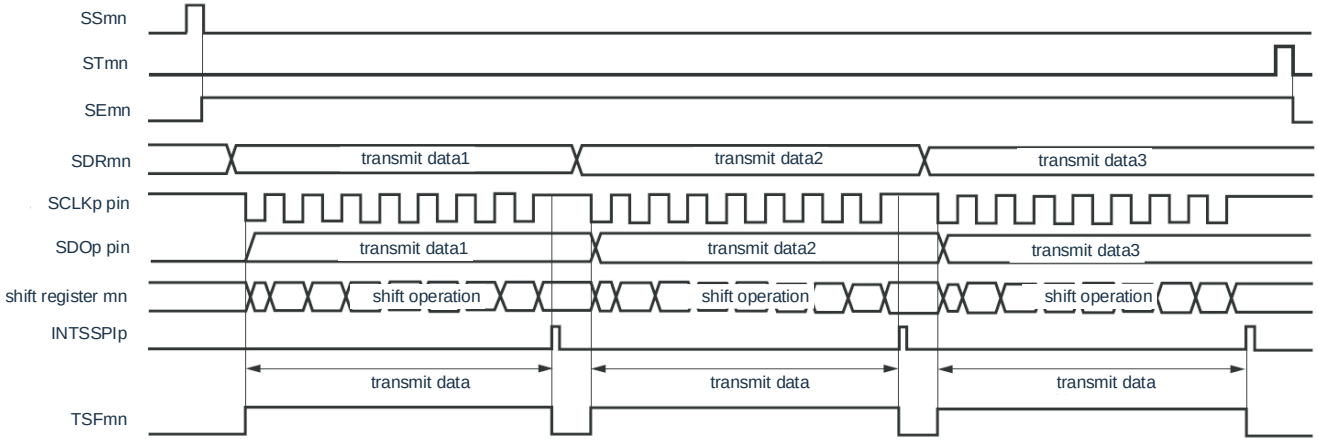
Figure 19-28 Restart set-up steps for master send



Remark If you override PER0 in the abort setting to stop providing the clock, you must wait until the communication object (slave) is stopped or the communication is over for the initial set-up instead of restarting it.

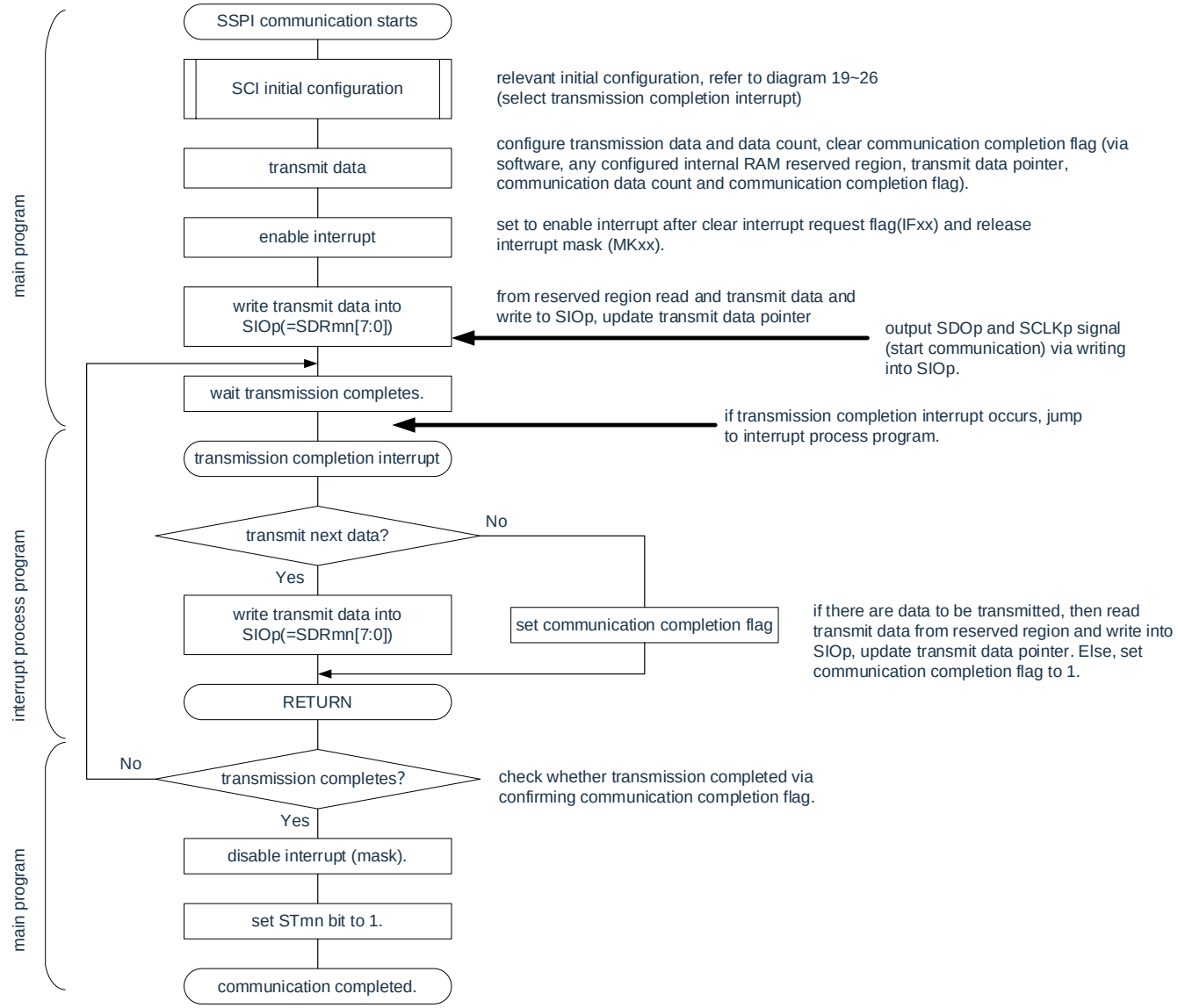
(3) Process flow (single send mode)

Figure 19-29 Timing diagram for master send (single send mode) (Type 1:DAPmn=0, CKPmn=0)



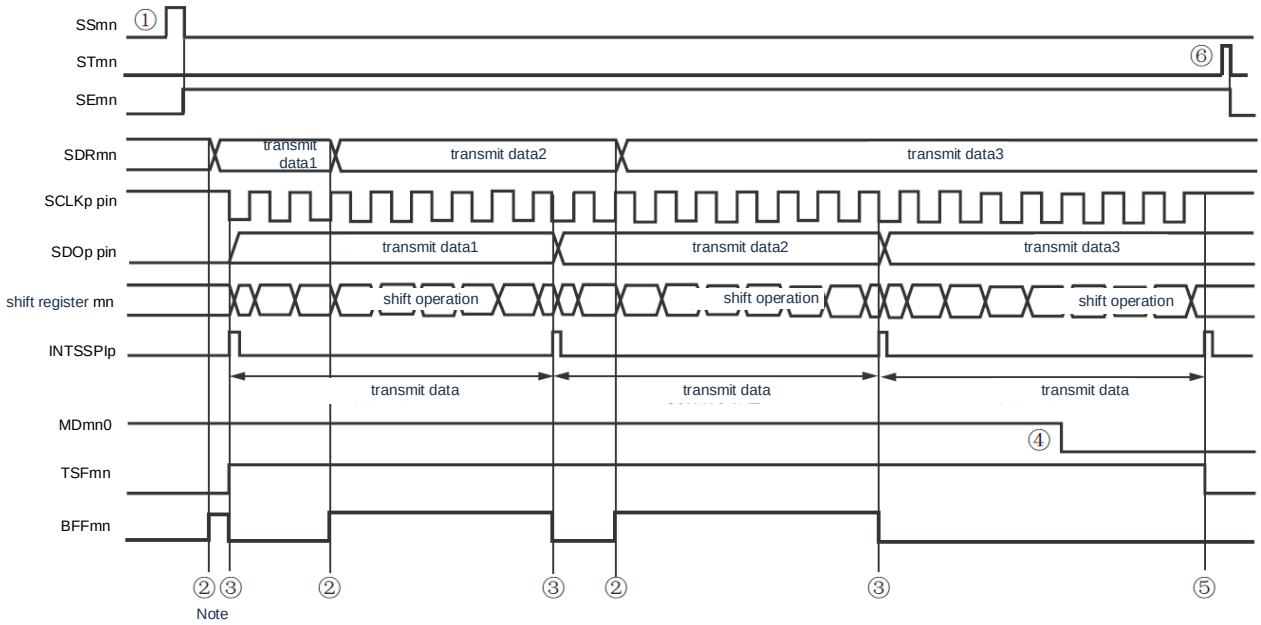
Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) p: SSPI Number (p=00,01,10,11,20,21) mn=00~03,10~11

Figure 19-30 Flowchart for Master Send (single send mode)



(4) Process flow (continuous send mode)

Figure 19-31 Timing diagram for master send (continuous send mode) (Type 1:DAPmn=0, CKPmn=0)

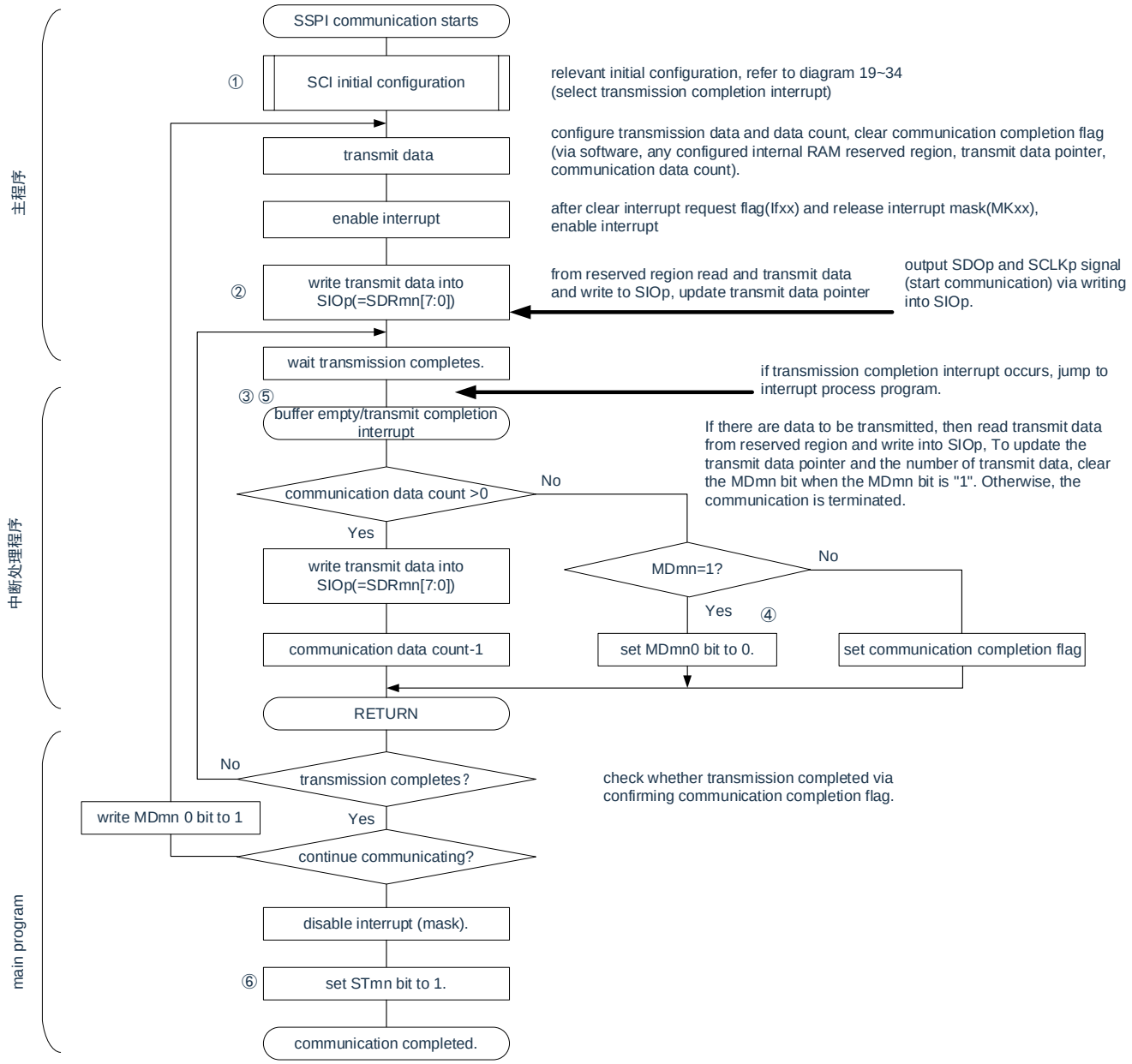


Note The transmission data is rewritten if the BFFmn bit of the serial status register mn (SSRmn) is "1" (SDRmn) when the valid data is stored in the serial data register mn (SDRmn).

Note The MDmn0 bit of the serial mode register mn(SMRmn) can be rewritten even in operation. However, in order to be able to catch the end of the transmission of the last transmitted data interrupt, it is necessary to override before starting the last transmission.

Remark m: Cell 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 19-32 Flow chart for master send (continuous send mode)



Remark (1) to (6) in the figure correspond to (1) to (6) in the figure.

19.5.2 master receive

Master reception refers to the operation of the BAT32A237 outputting a transmit clock and receiving data from other devices.

3-Wire Serial I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
object channel	Channel 0 for SCI0	Channel 1 for SCI0	Channel 2 for SCI0	Channel 3 for SCI0	Channel 0 for SCI1	Channel 1 for SCI1
Pin Used	SCLK00, SDO00	SLK01, SDO01	SLK10, SDO10	SLK11, SDO11	SLK20, SDO20	SLK21, SDO21
interrupt	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	Interrupt at that end of the transfer may be selecte (single transfer mode) or buffer air-discontinuity (continuous transfer mode).					
Error detection flag	Only the overflow error detection flag (OVFmn).					
length of transmit data	7 or 8-bit					
Transfer Rate Note	Max.fCLK/2[Hz] (SSPI00 only), fCLK/4[Hz] Min.fCLK/(2×2^{15}×128) [Hz] fCLK: system clock frequency					
data phase	Can be selected by the DAPmn bit of the SCRmn register. · DAPmn=0: Start the data output when the serial clock starts running. · DAPmn=1: The data output is started half a clock before the serial clock starts running.					
clock phase	Can be selected by the CKPmn bit of the SCRmn register. • CKPmn=0: prime CKPmn=1: inversion					
data orientation	MSB First or LSB First					

Note: It must be used within the scope of peripheral functional characteristics (reference data manual) that meet this condition and electrical characteristics.

Note: m: Cell 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 settings

Figure 19-33 3-wire Serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20,SSPI21) Example of Register Setting Content at Master Reception

(a) serial mode register mn(SMRmn)

	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	0	0	1	0	0	0	0	0/1

channel n operational clock (fMCK)
0: SPSm register configured pre-scaler output clock CKm0
1: SPSm register configured pre-scaler output clock CKm1

interrupt source of channel n
0: Transmit completion interrupt
1: Buffer empty interrupt

(b) serial communication operation configuration registermn mn(SCRmn)

	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	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1 注	0/1

data and clock phase selection (details refer to "19.3 control universal serial communication unit registers)

data transmit 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 bit: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	baud rate configuration (operation clock (fmck) scaling configuration)							0	received data write virtual data "FFH"							

SIOp

(d) serial output registerm (SOm)…. Only configure bit of target channel

	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	0/1	0/1	0/1	0	0	0	0	×	×	×	×

when clock phase is "positive phase" (CKPmn of SCRmn register as 0), "1" means starting communication; when clock phase is "inverted phase" (CKPmn=1), "0" means starting communication.

(e) serial output 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	×	×	×	×

(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	0/1	0/1	0/1

Note Limited to SCR00 and SCR01 registers, other fixed to "1".

Note 1.m: Cell 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 set (set initial value).
×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).
0/1: The "0" or "1" is set according to the user.

(2) Procedure

Figure 19-34 Initial set-up steps for master reception

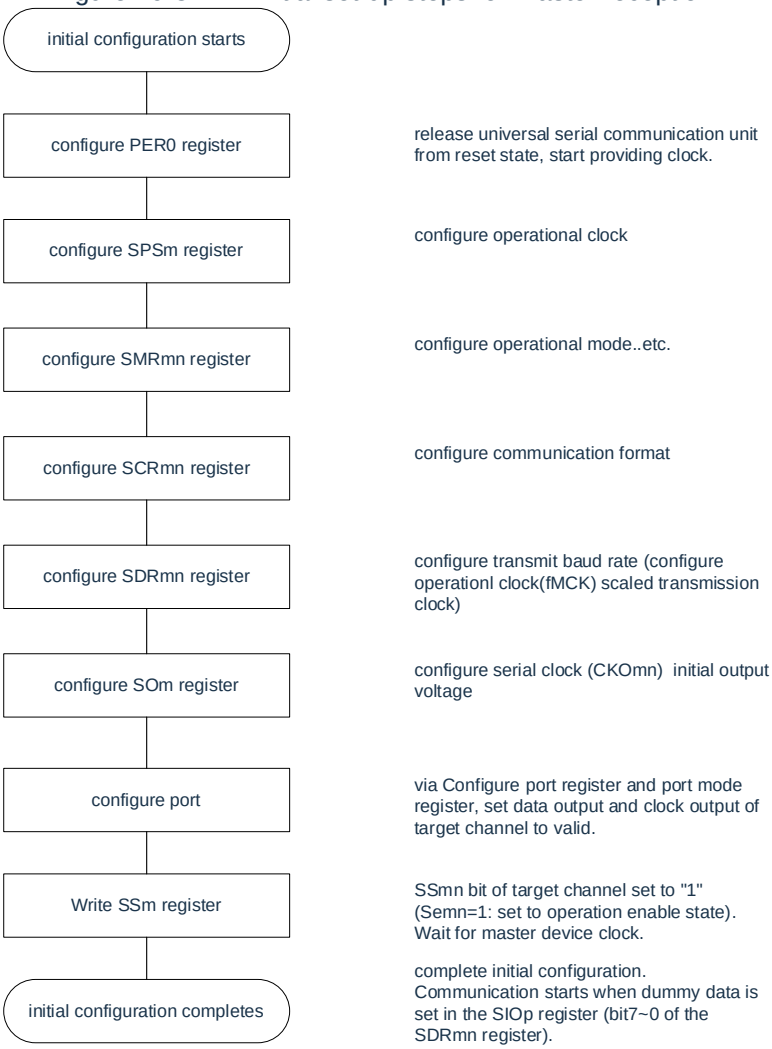


Figure 19-35 Stop step for master reception

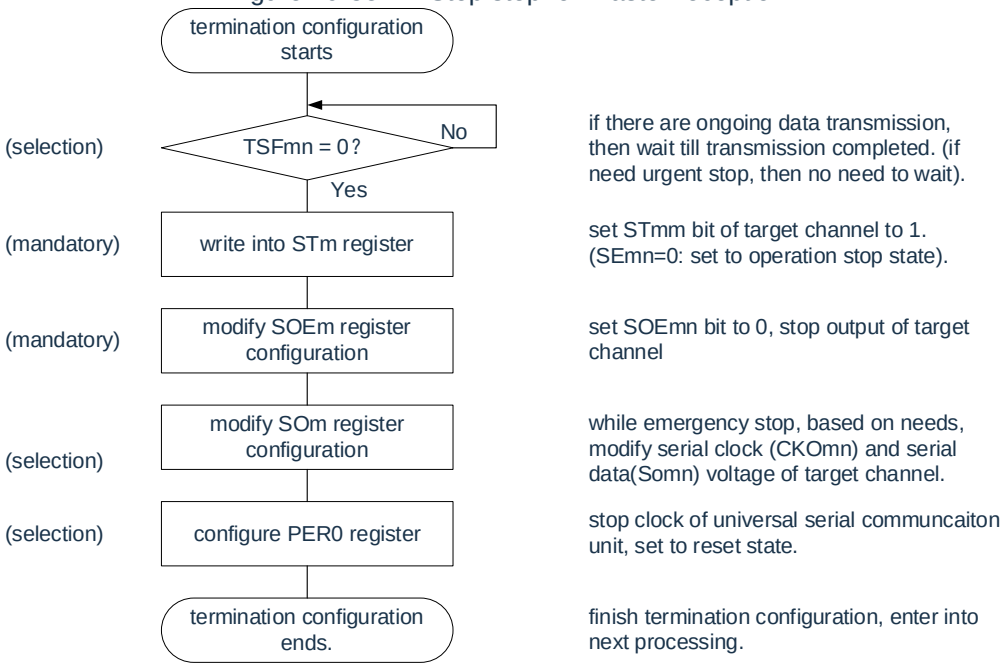
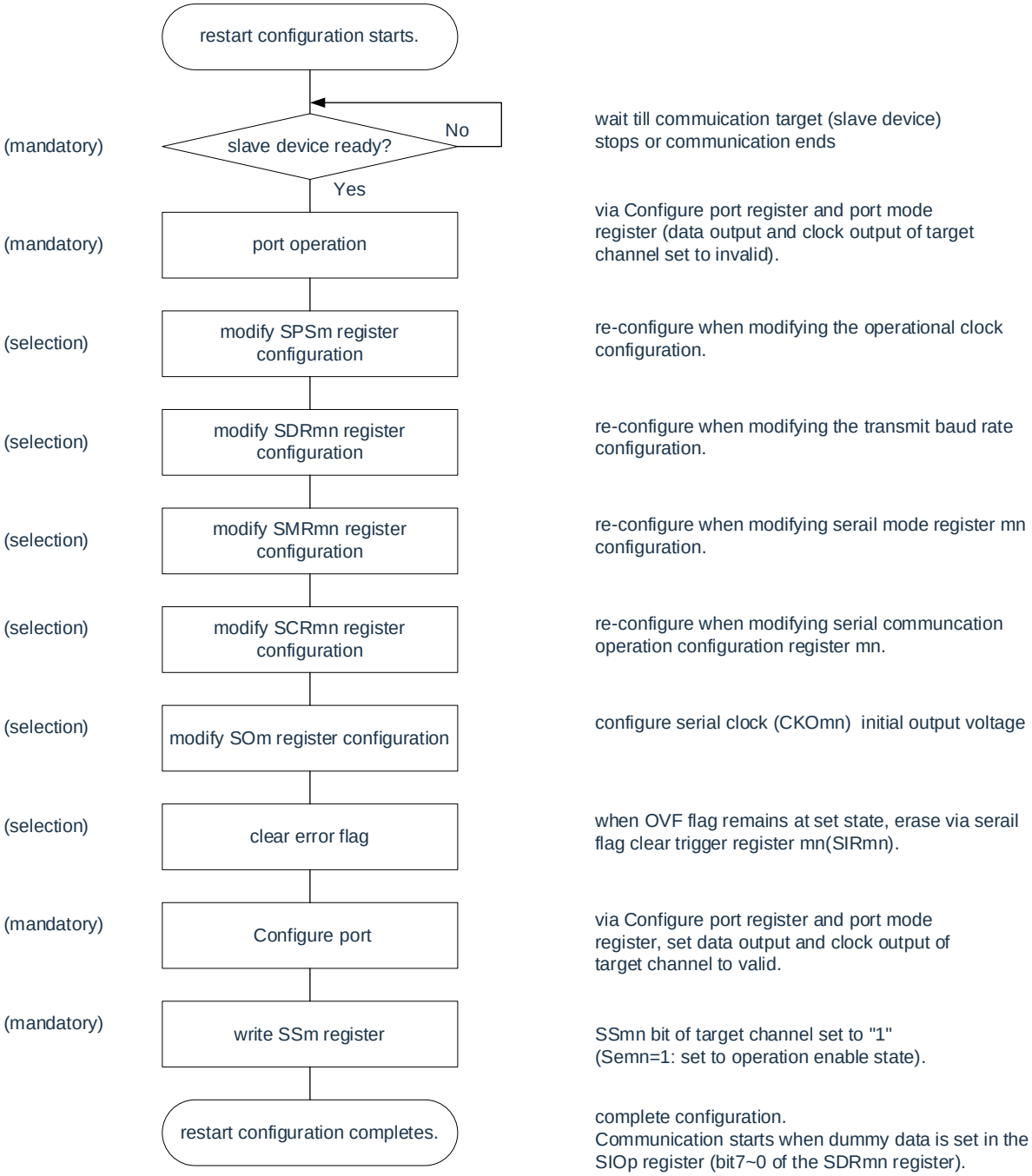


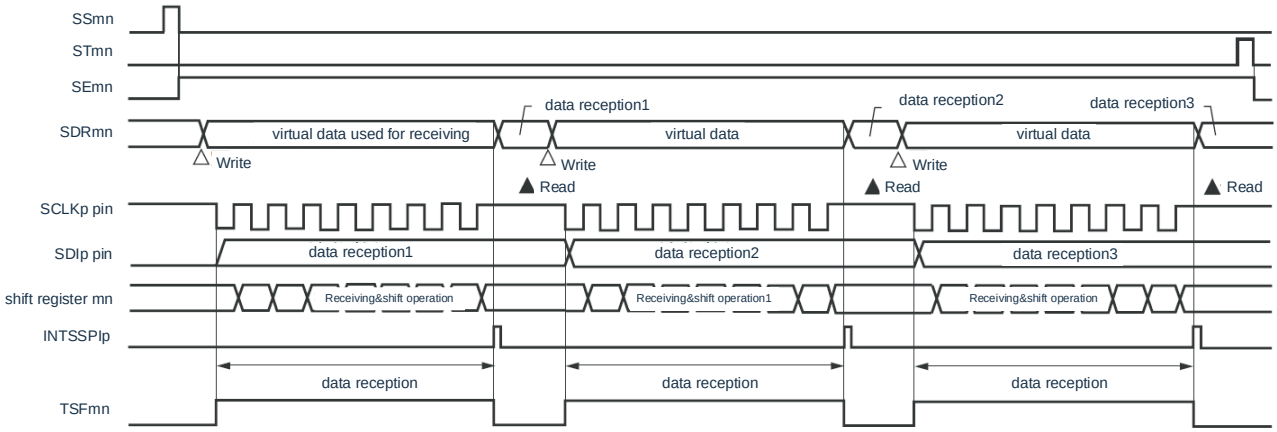
Figure 19-36 Restart set-up steps for master receipt



Remark If you override PER0 in the abort setting to stop providing the clock, you must wait until the communication object (slave) is stopped or the communication is over for the initial set-up instead of restarting it.

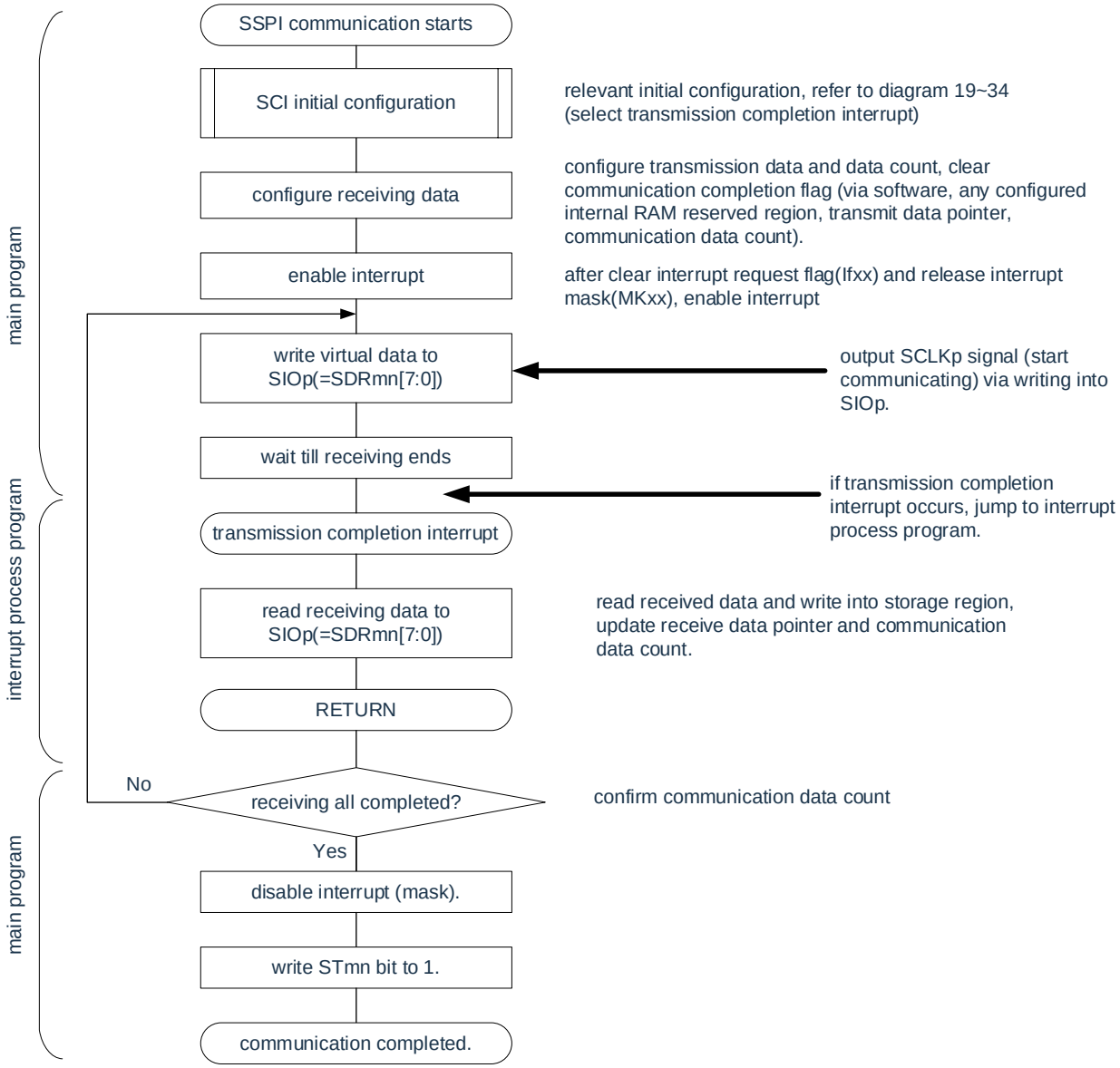
(3) Process flow (single receive mode)

Figure 19-37 Timing diagram of the master receive (single receive mode) (Type 1:DAPmn=0, CKPmn=0)



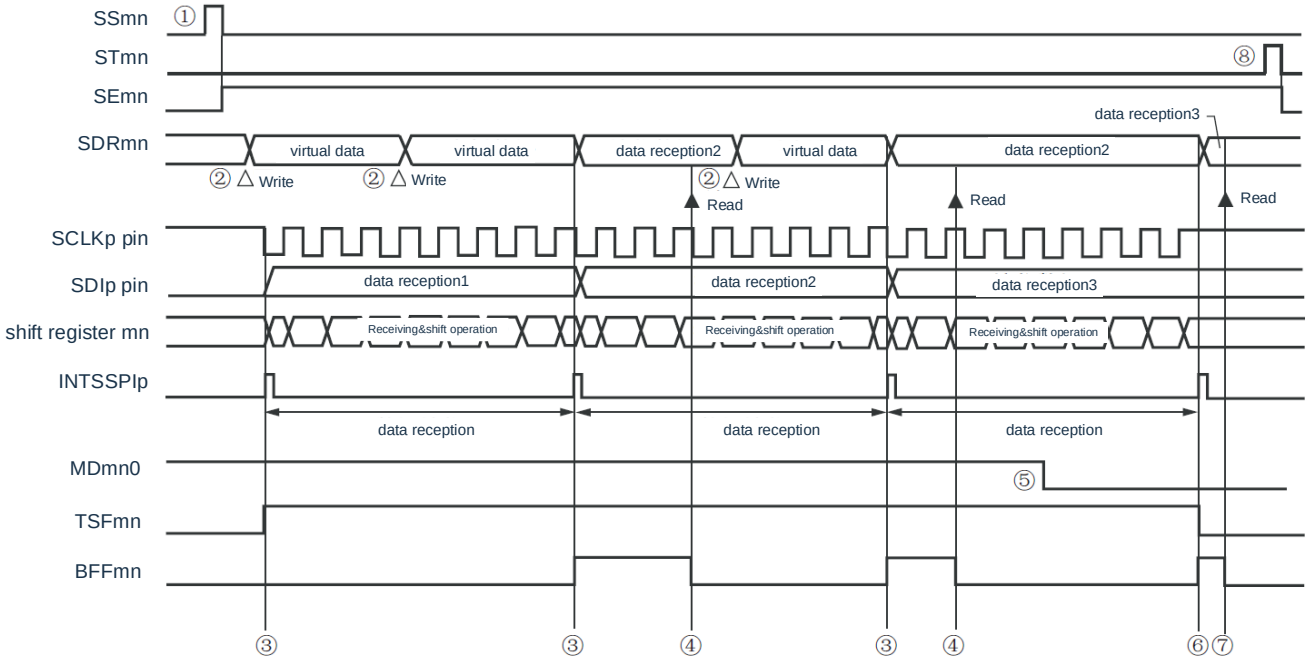
Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) p: SSPI Number (p=00,01,10,11,20,21) mn=00~03,10~11

Figure 19-38 Flowchart for Master Receive (Single Receive Mode)



(4) Process Flow (Continuous Receive Mode)

Figure 19-39 Timing diagram of the master receive (Continuous Receive Mode) (Type 1:DAPmn=0, CKPmn=0)



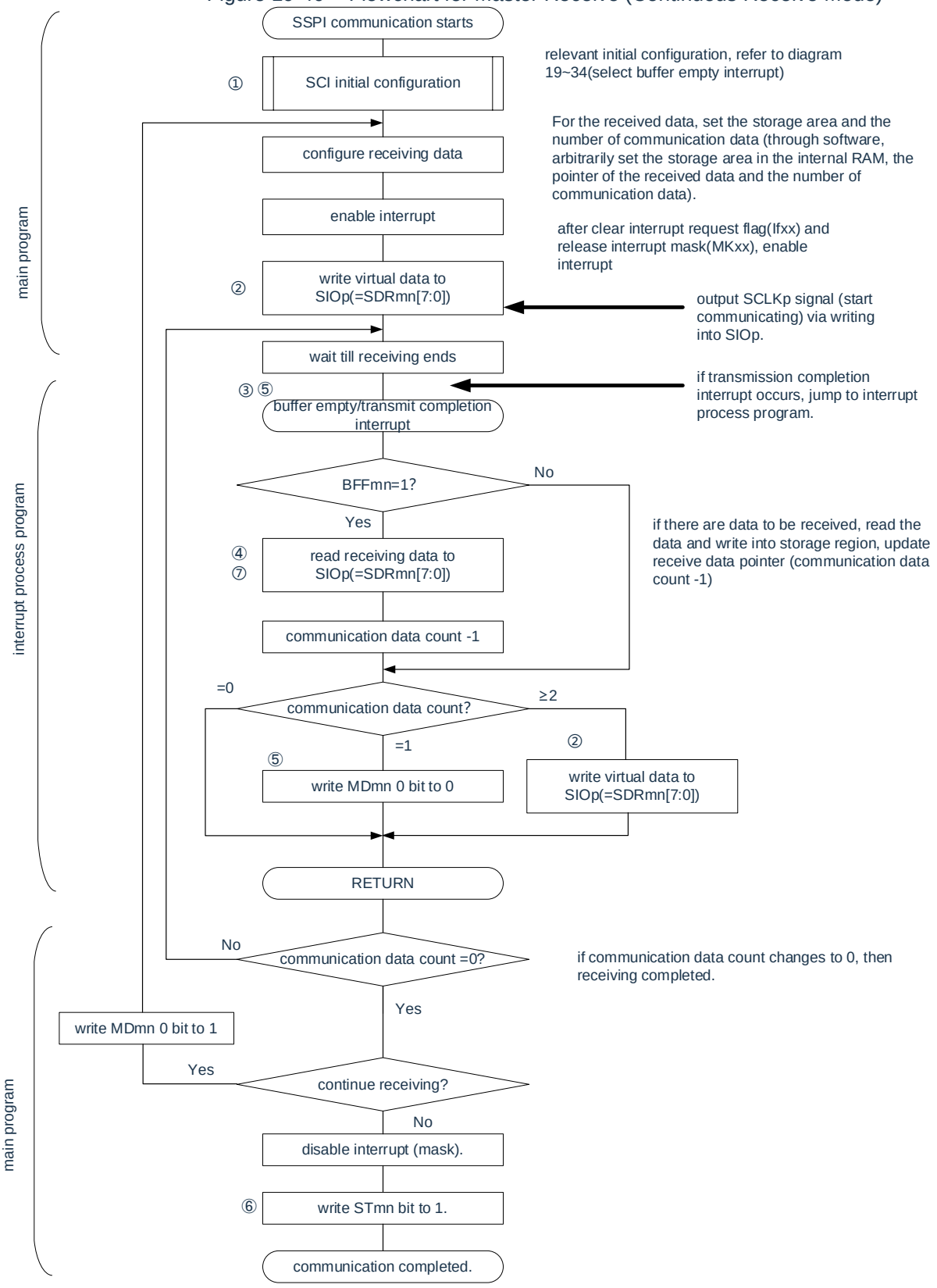
Note The MDmn0 bits can be overridden even during a run. However, in order to be able to catch that end interruption of the transmission of the last receive data, it is necessary to override before the last receive is started.

Note 1. (1) to (8) in the figure corresponds to (1) to (8) in Figure 19-40 Flow Chart.

2.m: Cell 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 19-40 Flowchart for Master Receive (Continuous Receive Mode)



Remark (1) to (8) in the figure correspond to (1) to (8) in the "Time Sequence Diagram of Main Control Receiving (Continuous Receiving Mode) Figure 19-39".

19.5.3 Master send and receive

The main control of sending and receiving refers to the BAT32A237 output transmission clock and other devices to send and receive data running.

3-Wire Serial I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
object channel	SCI0's Channel 0	SCI0's Channel 1	SCI0's Channel 2	SCI0's Channel 3	SCI1's Channel 0	SCI1's Channel 1
Pin Used	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
	Interrupt at that end of the transfer may be selecte (single transfer mode) or buffer air-discontinuity (continuous transfer mode).					
Error detection flag	Only the overflow error detection flag (OVFmn).					
length of transmit data	7 or 8-bit					
Transfer Rate Note	Max.fCLK/2[Hz] (SSPI00 only), fCLK/4[Hz] Min.fCLK/(2×2^{15}×128) [Hz] fCLK: system clock frequency					
data phase	Can be selected by the DAPmn bit of the SCRmn register. ·DAPmn=0: Start the data output when the serial clock starts running. ·DAPmn=1: The data output is started half a clock before the serial clock starts running.					
clock phase	Can be selected by the CKPmn bit of the SCRmn register. • CKPmn=0: prime					

Note: It must be used within the scope of peripheral functional characteristics (reference data manual) that meet this condition and electrical characteristics.

Note: m: Cell 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 settings

Figure 19-41 3-wire Serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20,SSPI21) Example of register setting content when master sends and receives

(a) serial mode register mn (SMRmn)

	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	0	0	1	0	0	0	0	0/1

channel n operational clock (fMCK)
0: SPSm register configured pre-scaler output clock CKm0
1: SPSm register configured pre-scaler output clock CKm1

interrupt source of channel n
0: Transmit completion interrupt
1: Buffer empty interrupt

(b) serial communication operation configuration registermn mn(SCRmn)

	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	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 "19.3 control universal serial communication unit registers)

data transmit 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 regsiter mn (SDRmn) (low 8 bit: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	baud rate configuration (operation clock (fmck) scaling configuration)							0	configuration of transmit data/received data register							

SIOp

(d) serial output register m(SOm) .....Only configure bit of target channel

	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	0/1	0/1	0/1	0	0	0	0	0/1	0/1	0/1	0/1

when clock phase is "positive phase" (CKPmn of SCRmn register as 0), "1" means starting communication;
when clock phase is "inverted phase" (CKPmn=1), "0" means starting communication.

(e) serial output enable registerm (SOEm)….only set bit of target channel to 1.

	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) …. 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	0/1	0/1	0/1

Note Limited to SCR00 and SCR01 registers, other fixed to "1".

Note 1.m: Cell 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 send and receive mode. ▩ : Cannot set (set initial value).
0/1: The "0" or "1" is set according to the user.

(2) Procedure

Figure 19-42 Initial set-up steps for master send and receive

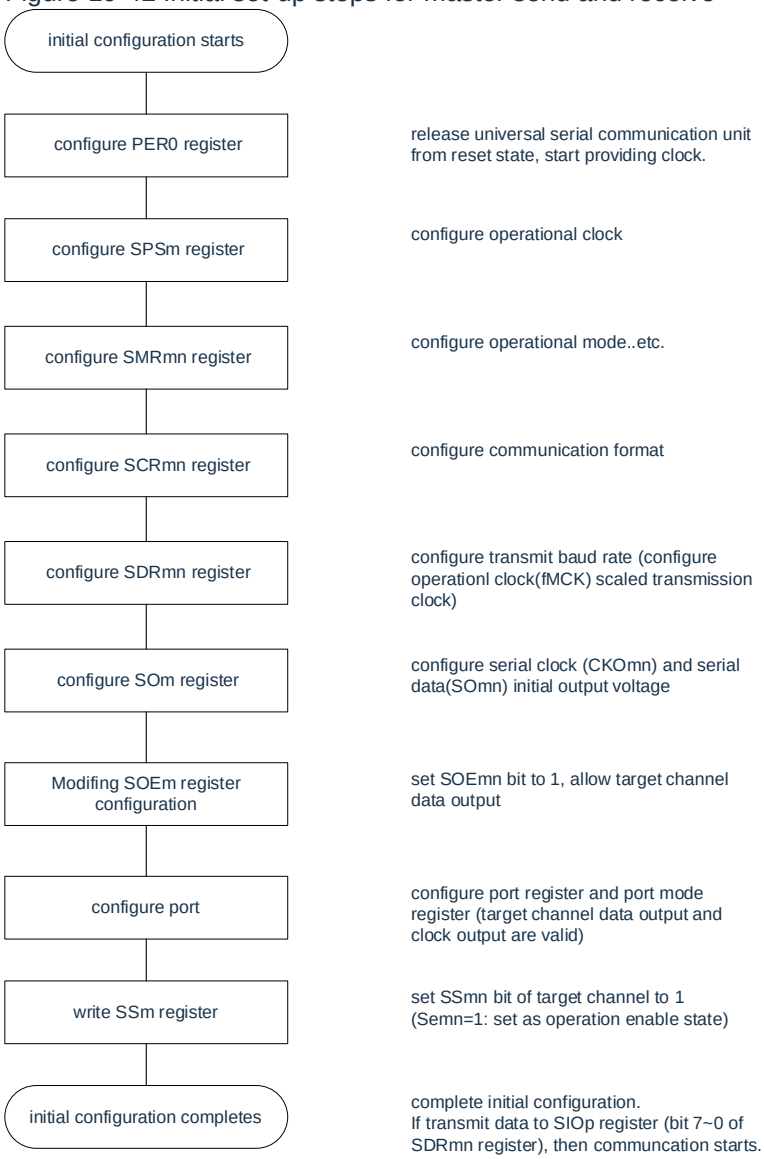


Figure 19-43 Stop step for master transmission and reception

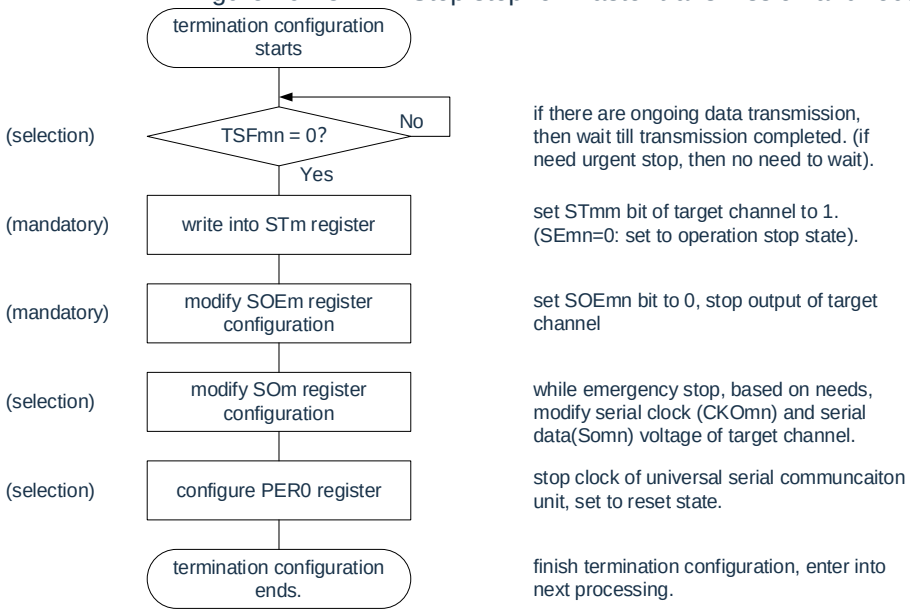
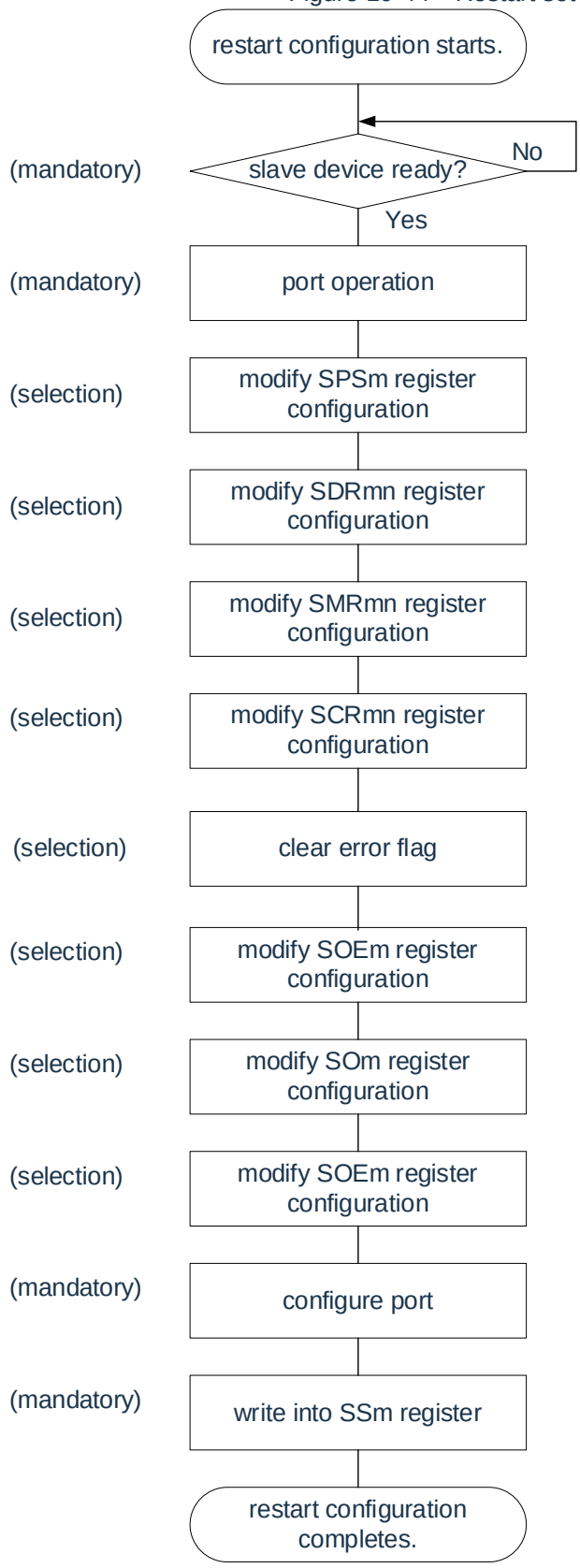
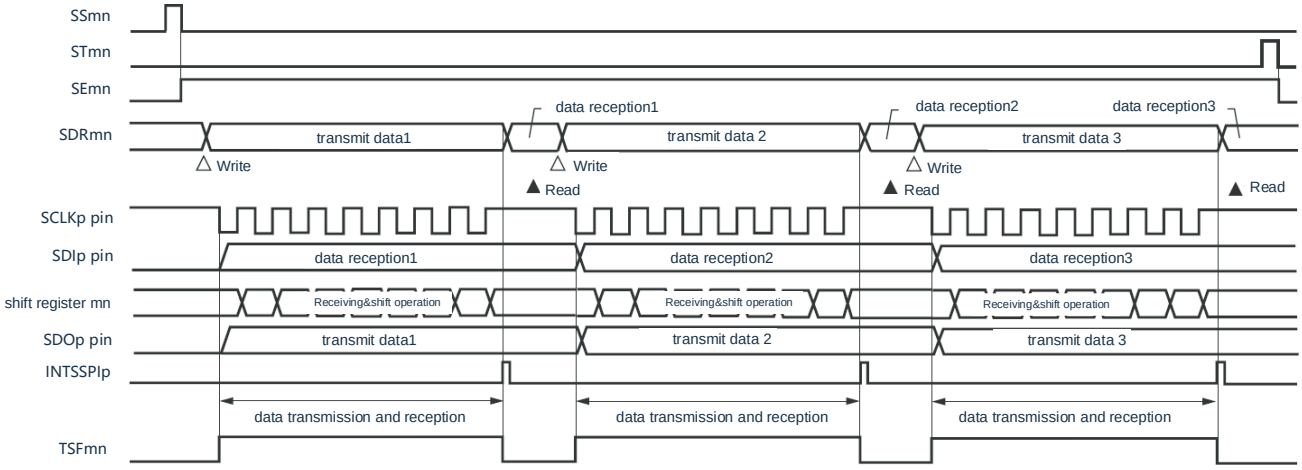


Figure 19-44 Restart set-up steps for master send and receive



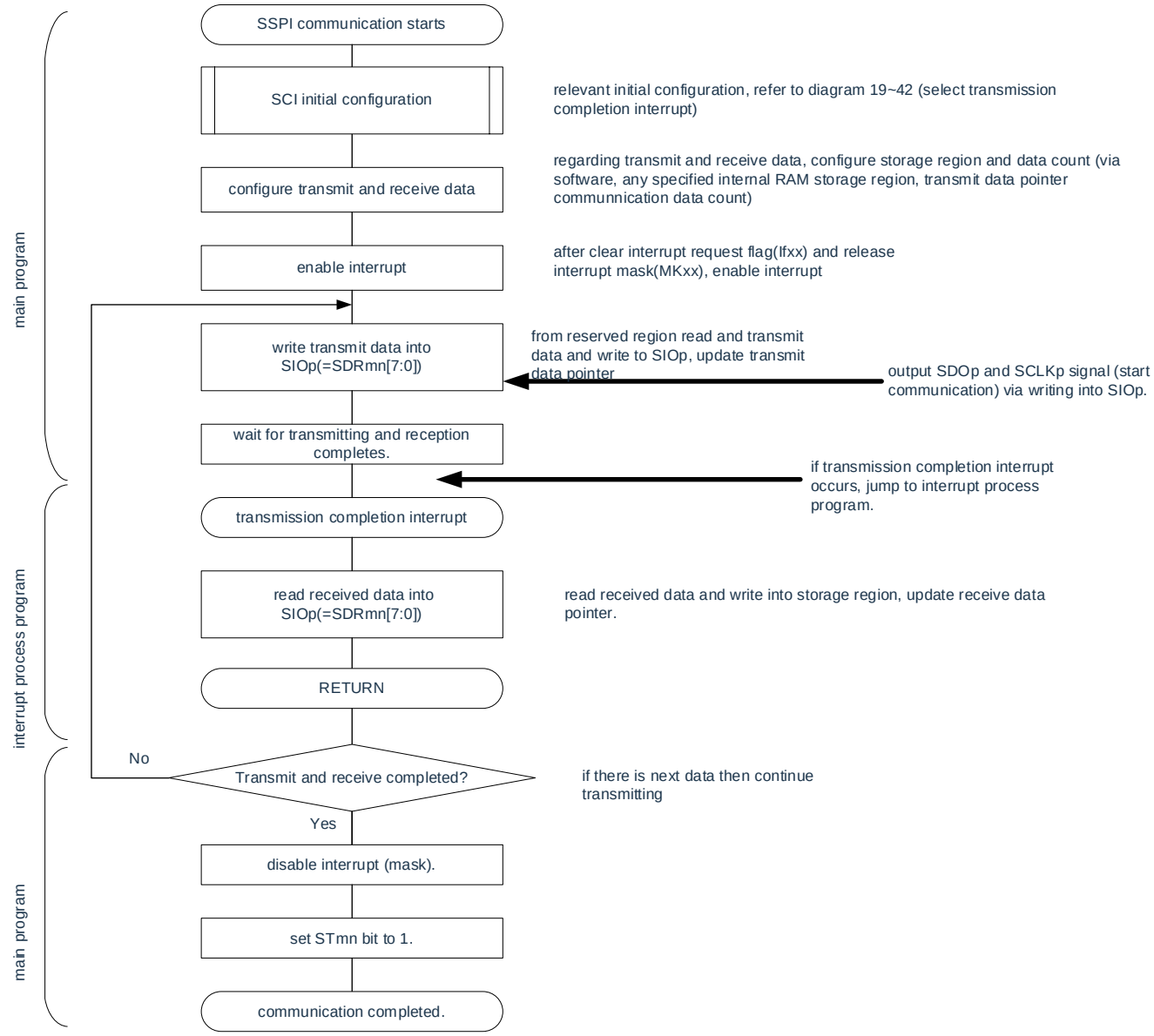
(3) Process flow (single send and receive mode)

Figure 19-45 Timing diagram for master send and receive (single send and receive mode) (Type 1:DAPmn=0, CKPmn=0)



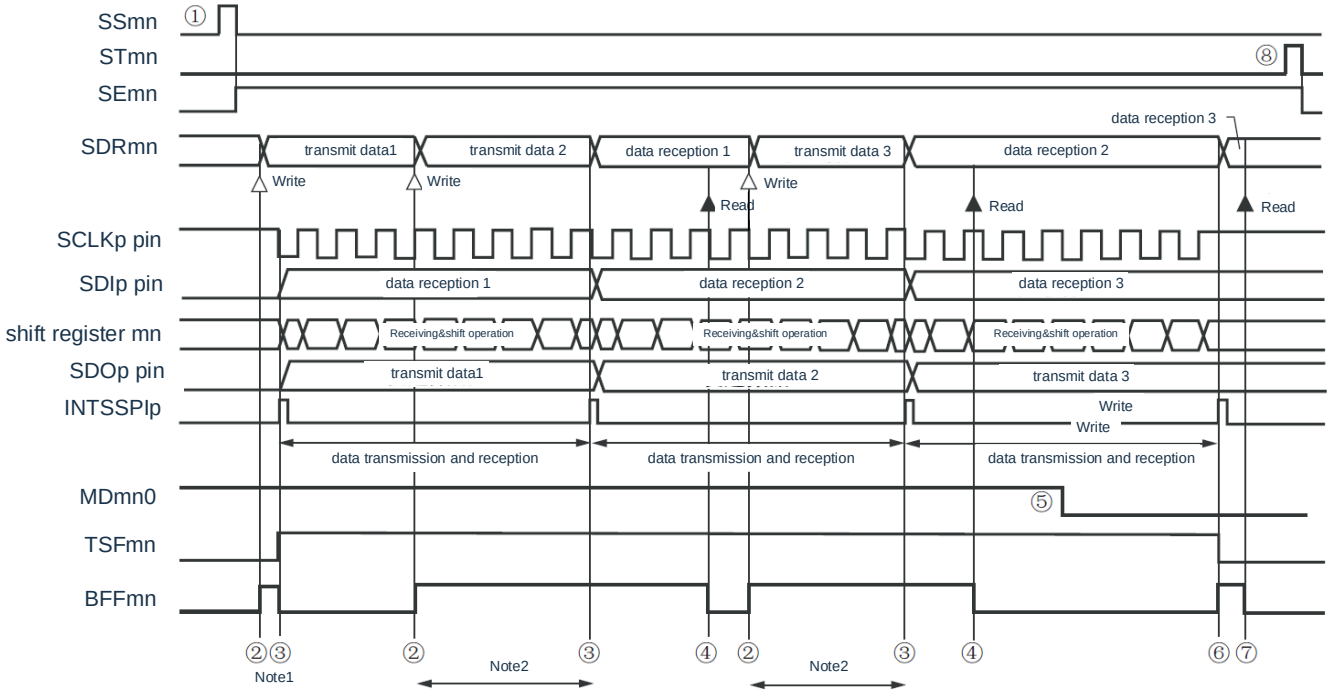
Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) p: SSPI Number (p=00,01,10,11,20,21) mn=00~03,10~11

Figure 19-46 Flow chart for master send and receive (single send and receive mode)



(4) Processing Flow (Continuous Send and Receive Mode)

Figure 19-47 Timing diagram for master send and receive (continuous send and receive mode) (Type 1:DAPmn=0, CKPmn=0).



Note 1. Rewrite the transmission data 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)).

2. Send data can be read if the SDRmn register is read during this period. At this point, the transfer operation is not affected.

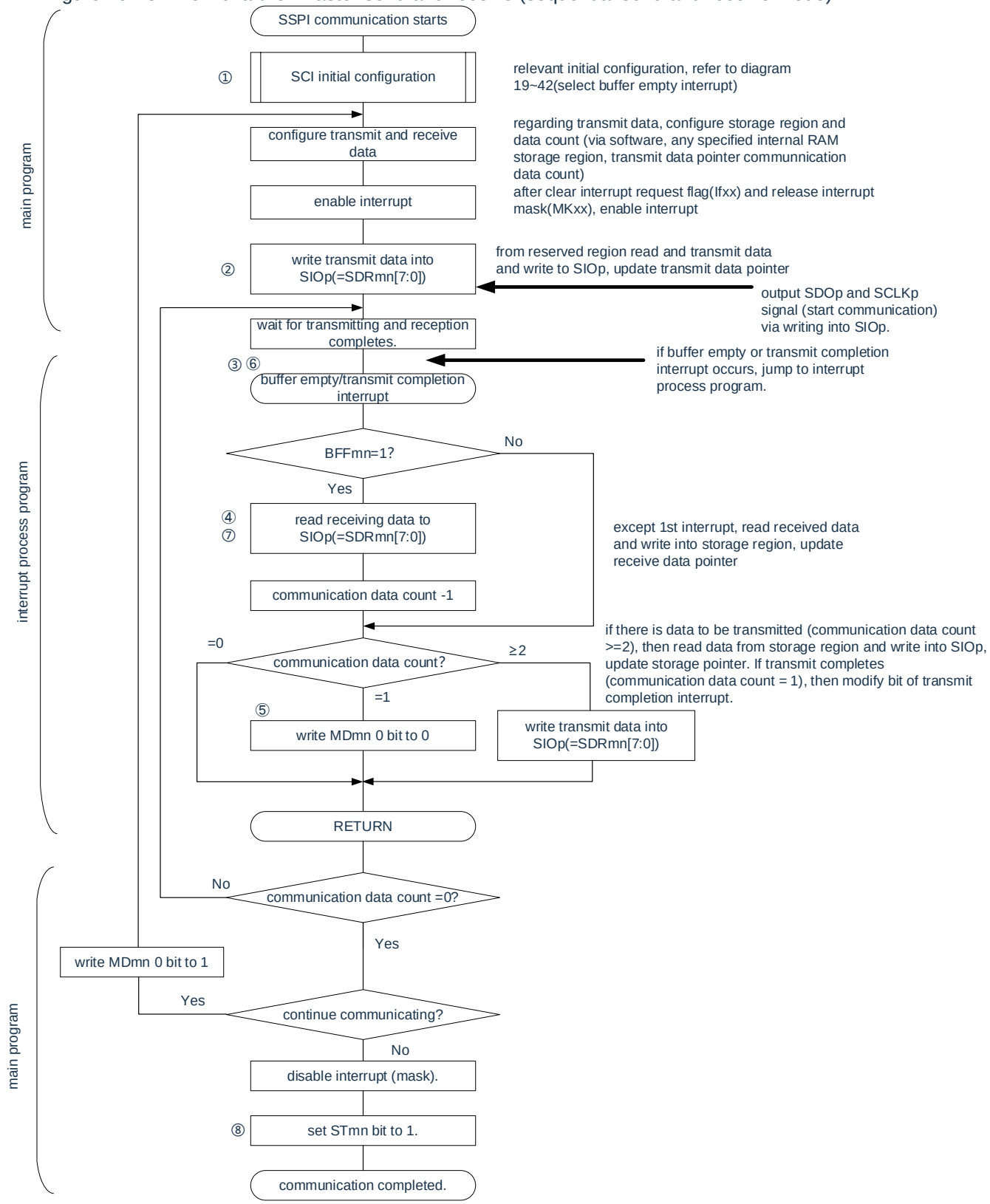
Note The MDmn0 bit of the serial mode register mn(SMRmn) can be rewritten even in operation. However, in order to be able to catch the end of the transmission of the last transmitted data interrupt, it is necessary to override before starting the last transmission.

Note 1. (1) to (8) in the figure corresponds to (1) to (8) in Figure 19-48 Flow Chart for Master Send and Receive Mode.

2.m: Cell 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 19-48 Flow chart for master send and receive (sequential send and receive mode)



Remark (1) to (8) in the figure correspond to (1) to (8) in "Figure 19-47 Timing Diagram for Main Control Sending and Receiving.

19.5.4 Slave send

Slave sending is the operation of the RL78 microcontroller sending data to other devices in a state where a transfer clock is input from other devices.

3-Wire Serial I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
object channel	Channel 0 for SCI0	Channel 1 for SCI0	Channel 2 for SCI0	Channel 3 for SCI0	Channel 0 for SCI1	Channel 1 for SCI1
Pin Used	SCLK00, SDO00	SCLK01, SDO01	SCLK10, SDO10	SCLK11, SDO11	SCLK20, SDO20	SCLK21, SDO21
interrupt	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	Interrupt at that end of the transfer may be selecte (single transfer mode) or buffer air-discontinuity (continuous transfer mode).					
Error detection flag	Only the overflow error detection flag (OVFmn).					
length of transmit data	7 or 8-bit					
transfer rate	Max.$_{fMCK}$/6[Hz]Note 1,2					
data phase	Can be selected by the DAPmn bit of the SCRmn register. ·DAPmn=0: Start the data output when the serial clock starts running. ·DAPmn=1: The data output is started half a clock before the serial clock starts running.					
clock phase	Can be selected by the CKPmn bit of the SCRmn register. • CKPmn=0: prime • CKPmn=1: inversion					
data orientation	MSB First or LSB First					

Note: 1. The maximum transfer rate is fMCK/6[Hz]since the external serial clock input by SCLK00, SCLK01, SCLK10, SCLK11, $_{SCLK20}$, $_{SCLK21\ pins\ is}$ used internally.

2. Must be used within the scope of peripheral functional characteristics (reference data manual) that satisfy this condition and electrical characteristics.

Note 1.$_{fMCK}$: The running clock frequency of the object channel

2.m: Cell number (m=0,1)n: Channel Number (n=0~3) mn=00~03,10~11

(1) Register settings

Figure 19-49 3-wire Serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20,SSPI21) Examples of register setting contents at slave transmission

(a) serial mode register mn (SMRmn)

	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	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

channel n operational clock (fMCK)
0: SPSm register configured pre-scaler output clock CKm0
1: SPSm register configured pre-scaler output clock CKm1

interrupt source of channel n
0: Transmit completion interrupt
1: Buffer empty interrupt

(b) serial communication operation configuration registermn mn(SCRmn)

	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	0/1	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 "19.3 control universal serial communication unit registers)

data transmit 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 regsiter mn (SDRmn) (low 8 bit: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 baud rate configuration							0	configuration of transmit data							

SIOp

(d) serial output register m(SOm) .....Only configure bit of target channel

	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	0/1	0/1	0/1	0/1

(e) serial output enable registerm (SOEm)….only set bit of target channel to 1.

	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) …. 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	0/1	0/1	0/1

Note Limited to SCR00 and SCR01 registers, other fixed to "1".

Note 1.m: Cell 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 set (set initial value).
×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).
0/1: The "0" or "1" is set according to the user.

(2) Procedure

Figure 19-50 Initial set-up steps for slave transmission

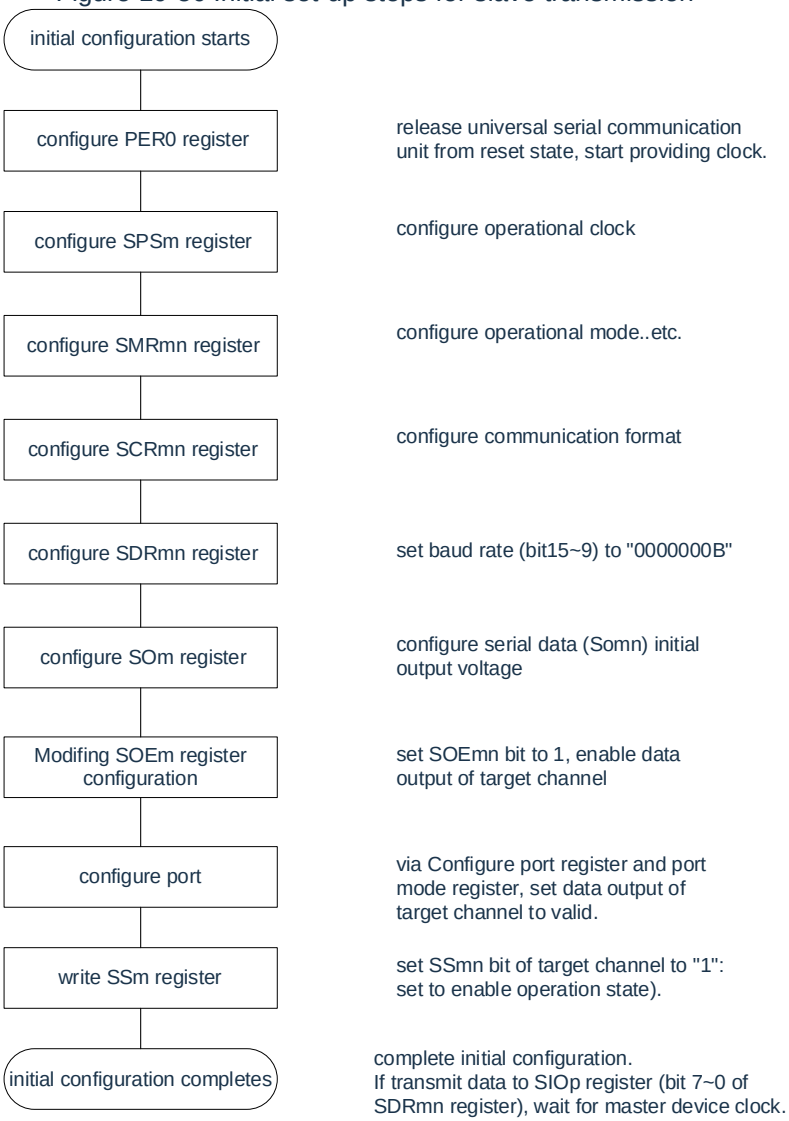


Figure 19-51 Abort Step for Slave transmission

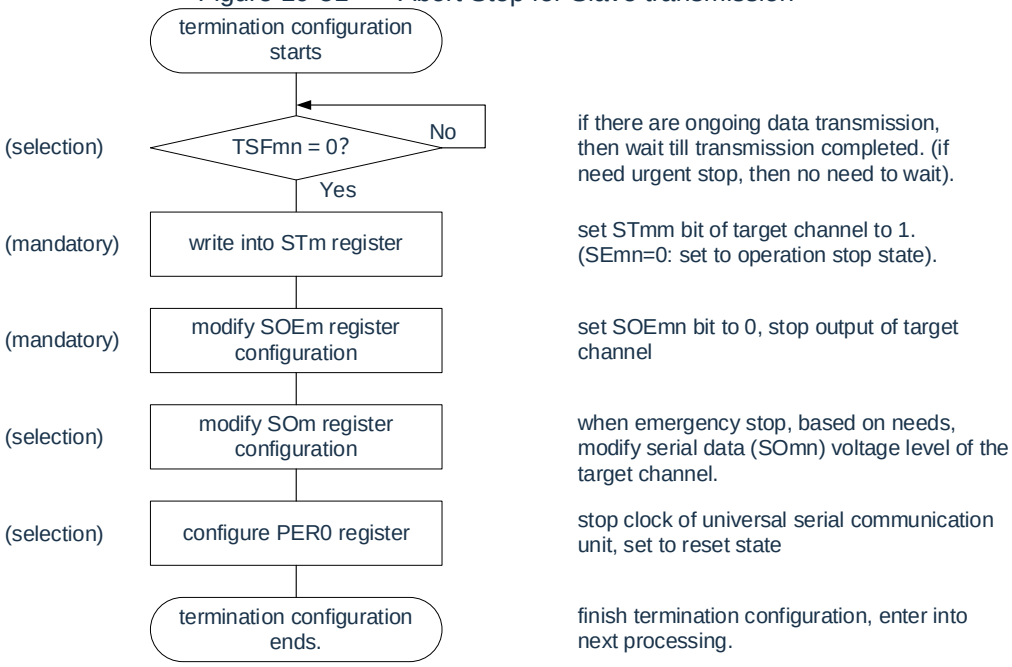
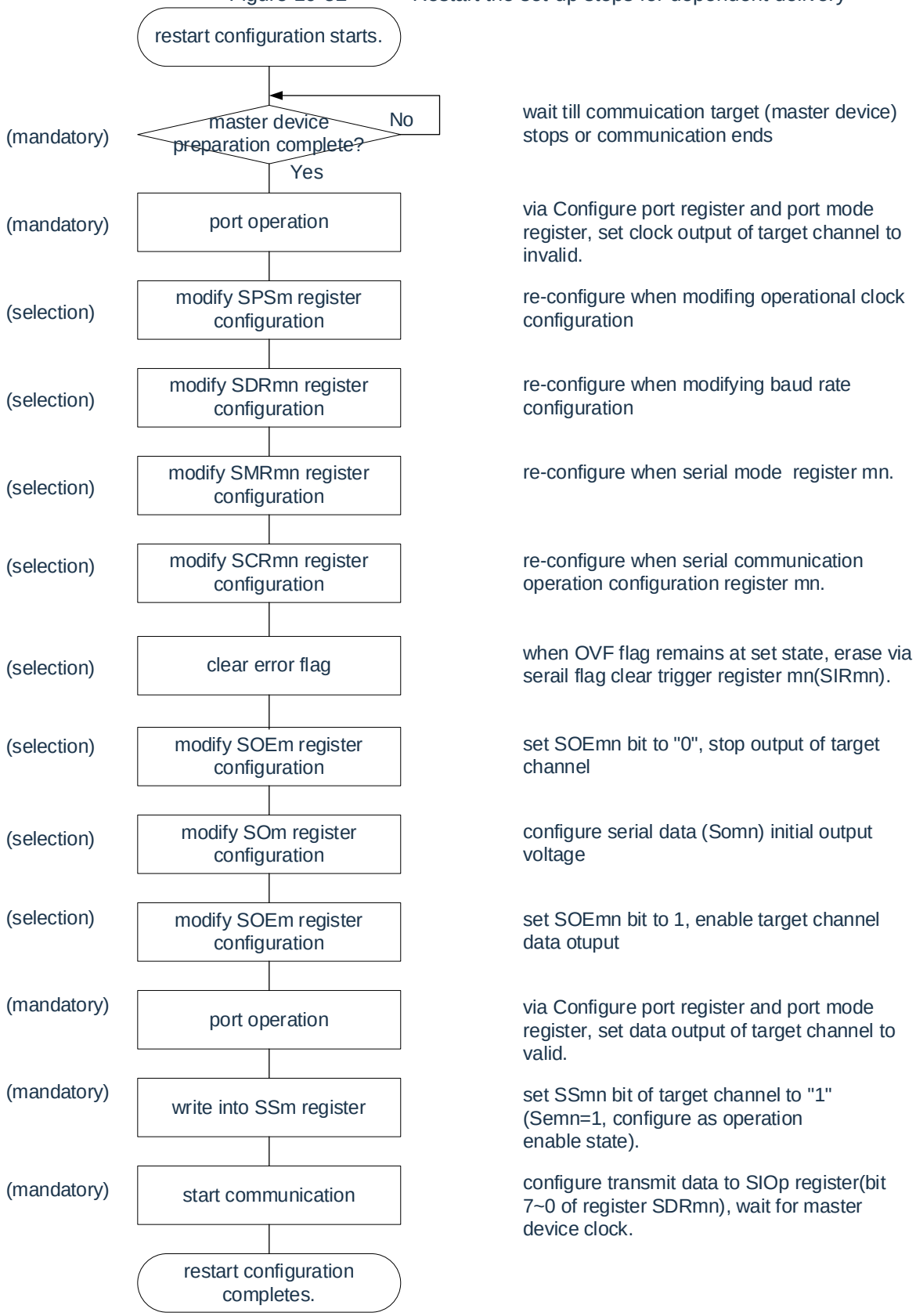


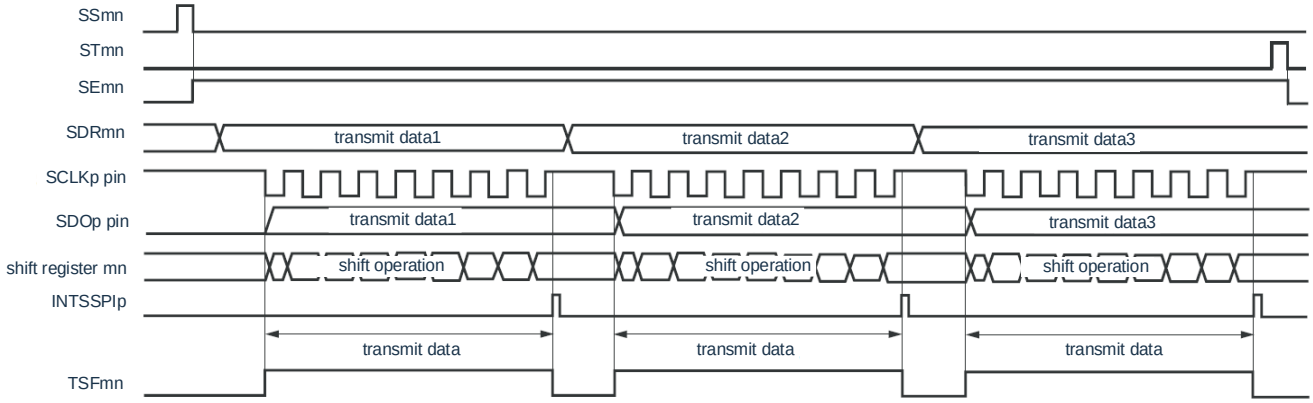
Figure 19-52 Restart the set-up steps for dependent delivery



Note: If you override PER0 in the abort setting to stop providing the clock, you must wait until the communication object (the master device) is stopped or the communication is over for the initial set-up instead of restarting.

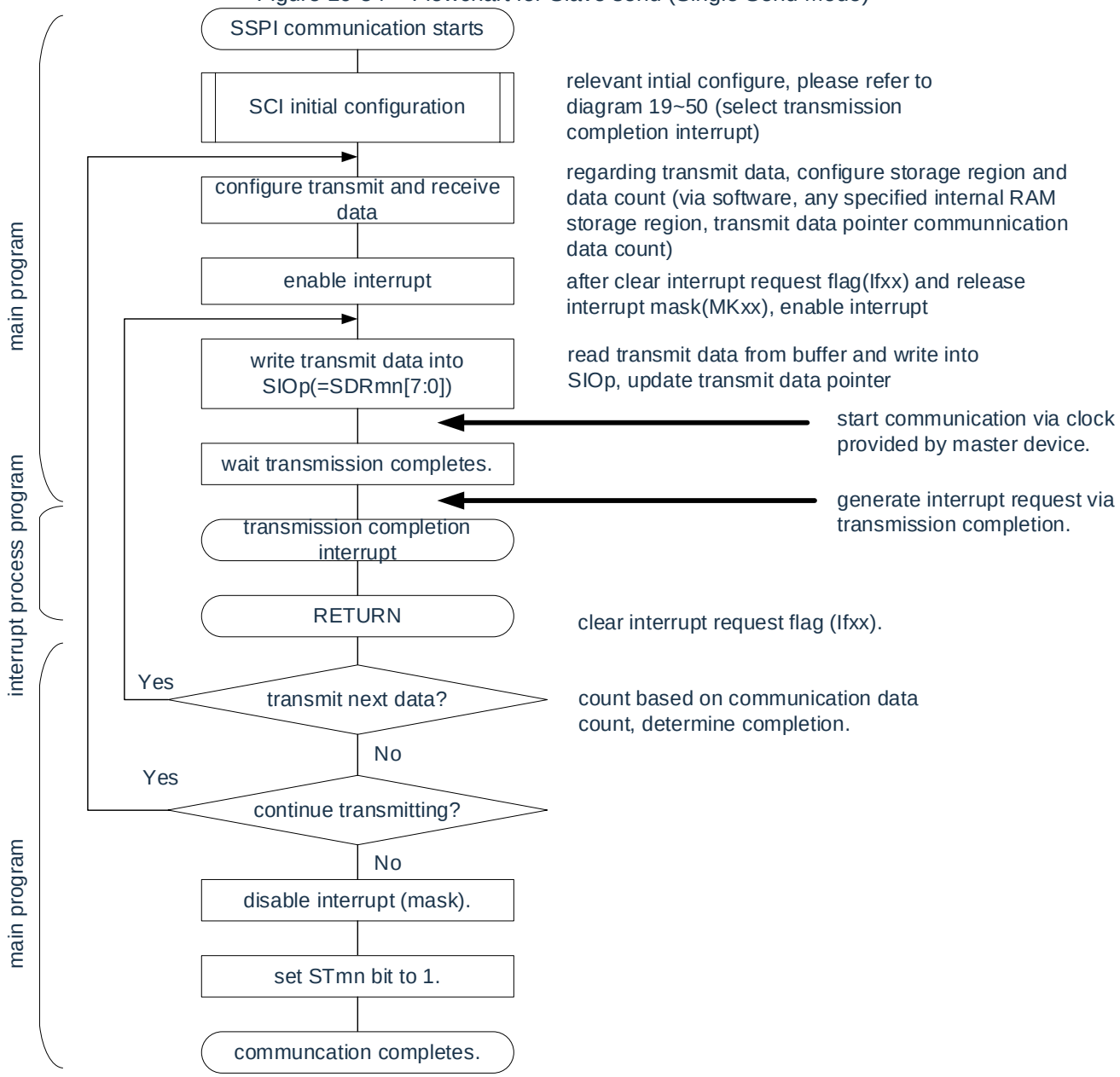
(3) Process flow (single send mode)

Figure 19-53 Time Series for Slave send (single send mode) (Type 1:DAPmn=0, CKPmn=0) Type 1



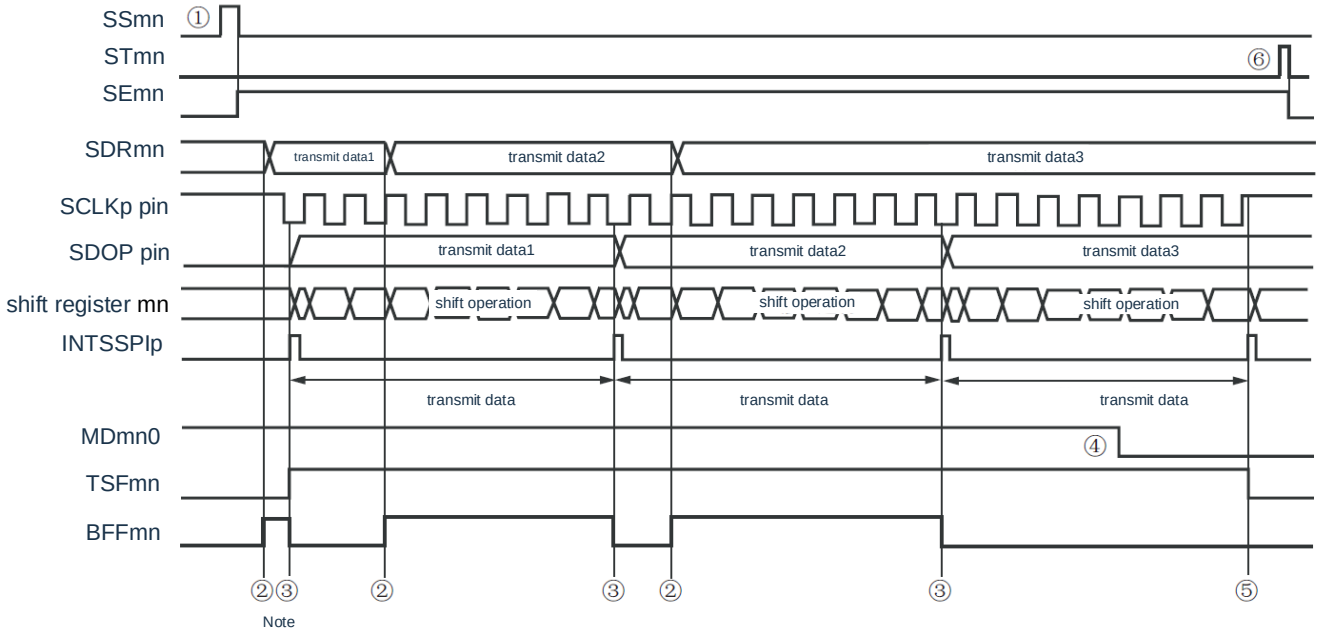
Note: m: Cell number (m=0,1)n: Channel Number (n=0~3) p: SSPI Number (p=00,01,10,11,20,21) mn=00~03,10~11

Figure 19-54 Flowchart for Slave send (Single Send Mode)



(4) Process flow (continuous send mode)

Figure 19-55 Time Series for Slave send (Continuous Send Mode) (Type 1:DAPmn=0, CKPmn=0).

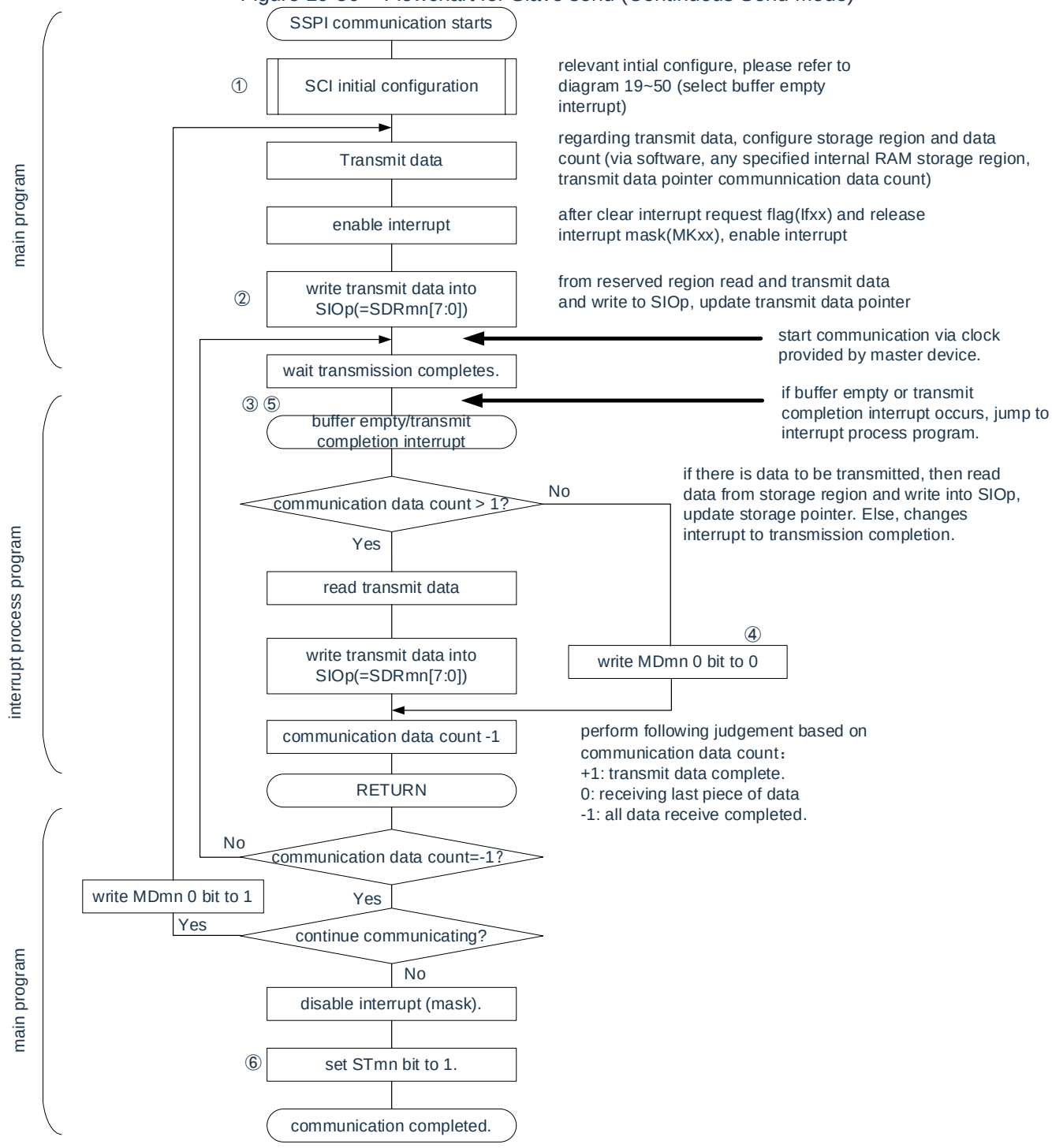


Note The transmission data is rewritten if the BFFmn bit of the serial status register mn (SSRmn) is "1" (SDRmn) when the valid data is stored in the serial data register mn (SDRmn).

Note The MDmn0 bit of the serial mode register mn(SMRmn) can be rewritten even in operation. However, you must override before you start transferring the last bit.

Remark m: Cell 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 19-56 Flowchart for Slave send (Continuous Send Mode)



Remark (1) to (6) in the figure corresponds to (1) to (6) in the "Chronograms of Slave transmission (Continuous Transmission Mode) in Figure 19-55".

19.5.5 slave receive

A slave receive is a run in which BAT32A237 receives data from other devices in a state in which a transfer clock is input from other devices.

3-Wire Serial I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
object channel	SCI0's Channel 0	SCI0's Channel 1	SCI0's Channel 2	SCI0's Channel 3	SCI1's Channel 0	SCI1's Channel 1
Pin Used	SCLK00, SDI00	SCLK01, SDI01	SCLK10, SDI10	SCLK11, SDI11	SCLK20, SDI20	SCLK21, SDI21
interrupt	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	Interrupt at that end of the transfer only (Disable from setting buffer air interrupt).					
Error detection flag	Only the overflow error detection flag (OVFmn).					
length of transmit data	7 or 8-bit					
transfer rate	Max.$_{fMCK}$/6[Hz]Note 1,2					
data phase	Can be selected by the DAPmn bit of the SCRmn register. ·DAPmn=0: Start the data output when the serial clock starts running. ·DAPmn=1: The data output is started half a clock before the serial clock starts running.					
clock phase	Can be selected by the CKPmn bit of the SCRmn register. • CKPmn=0: prime • CKPmn=1: inversion					
data orientation	MSB First or LSB First					

Note 1. The maximum transfer rate is fMCK/6[Hz]since the external serial clock input by SCLK00, SCLK01, SCLK10, SCLK11, $_{SCLK20}$, $_{SCLK21}$ pins is used internally.

2. Must be used within the scope of peripheral functional characteristics (reference data manual) that satisfy this condition and electrical characteristics.

Note 1.$_{fMCK}$: The running clock frequency of the object channel

2.m: Cell number (m=0,1)n: Channel Number (n=0~3) mn=00~03,10~11

(1) Register settings

Figure 19-57 3-wire Serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20,SSPI21) Example of register setting content at slave receive

(a) serial mode register mn (SMRmn)

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

channel n operational clock (fMCK)
0: SPSm register configured pre-scaler output clock CKm0
1: SPSm register configured pre-scaler output clock CKm1

interrupt source of channel n
0: Transmit completion interrupt
1: Buffer empty interrupt

(b) serial communication operation configuration registermn mn(SCRmn)

	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	0/1	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 "19.3 control universal serial communication unit registers)

data transmit 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 regsiter mn (SDRmn) (low 8 bit: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 baud rate configuration							0	received 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
SOm					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
SOEm													SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	×	×	×

(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	0/1	0/1	0/1

Note Limited to SCR00 and SCR01 registers, other fixed to "1".

Note 1.m: Cell 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

2. □ : Set in dependent receive mode for fixed. ■: Cannot set (set initial value).
×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).
0/1: The "0" or "1" is set according to the user.

2) Procedure

Figure 19-58 Initial set-up steps for dependent receipts

Figure 19-59 Dependent receive abort step

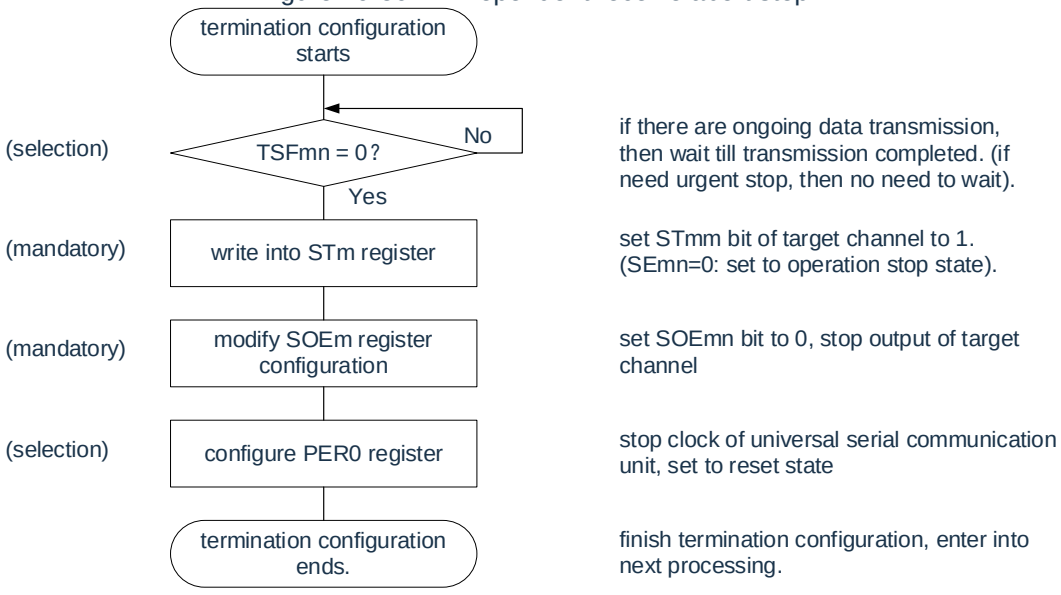
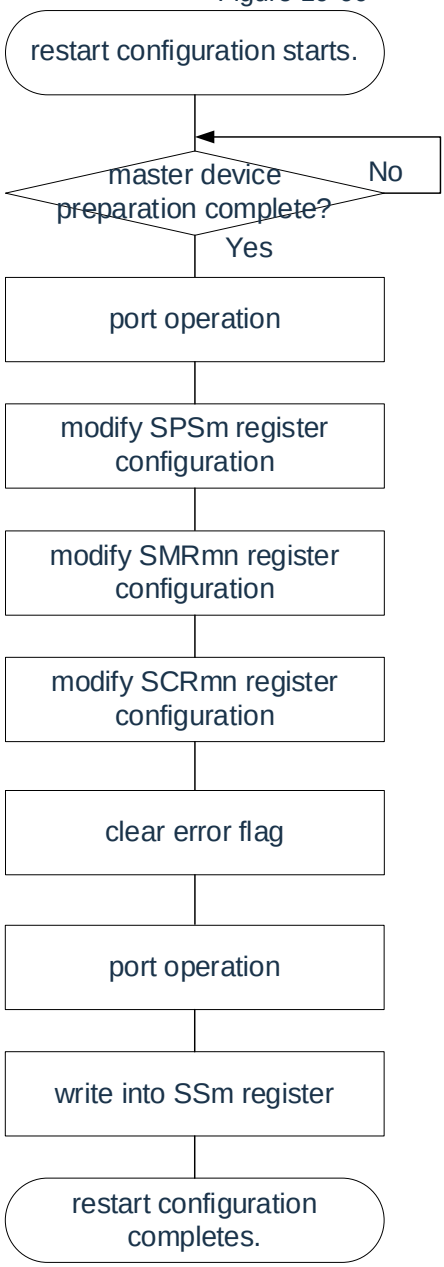


Figure 19-60 Restart set-up steps for dependent receipts

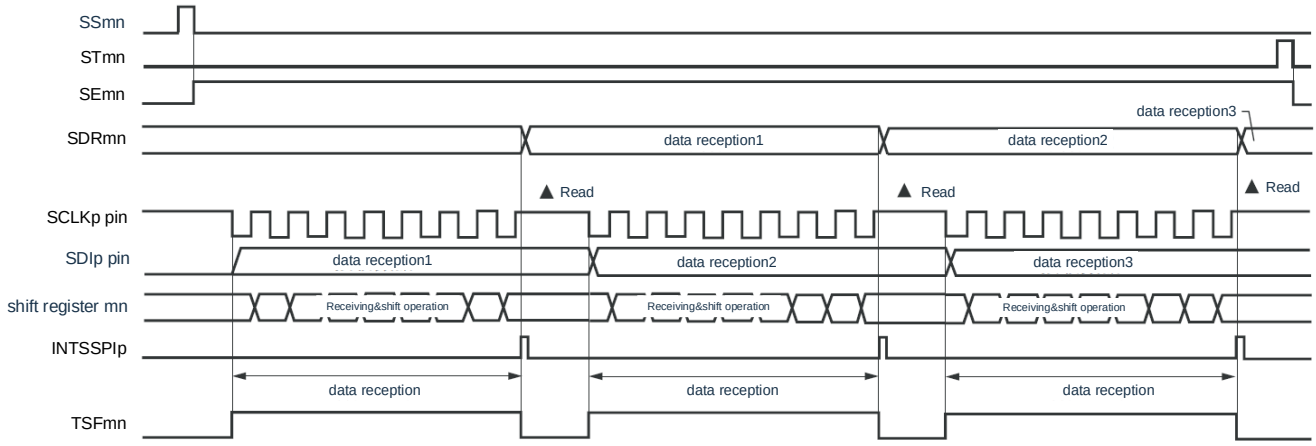


(mandatory)	master device preparation complete?	wait till commuication target (master device) stops or communication ends
(mandatory)	port operation	via Configure port register and port mode register, set clock output of target channel to invalid.
(selection)	modify SPSm register configuration	re-configure when modifing operational clock configuration
(selection)	modify SMRmn register configuration	re-configure when serial mode register mn.
(selection)	modify SCRmn register configuration	re-configure when serial communication operation configuration register mn.
(selection)	clear error flag	when OVF flag remains at set state, erase via serail flag clear trigger register mn(SIRmn).
(mandatory)	port operation	via Configure port register and port mode register, set data output of target channel to valid.
(mandatory)	write into SSm register	set SSmn bit of target channel to "1" (Semn=1, configure as operation enable state). wait for master device clock.

Remark If you override PER0 in the abort setting to stop providing the clock, you must wait until the communication object (the master device) is stopped or the communication is over for the initial set-up instead of restarting.

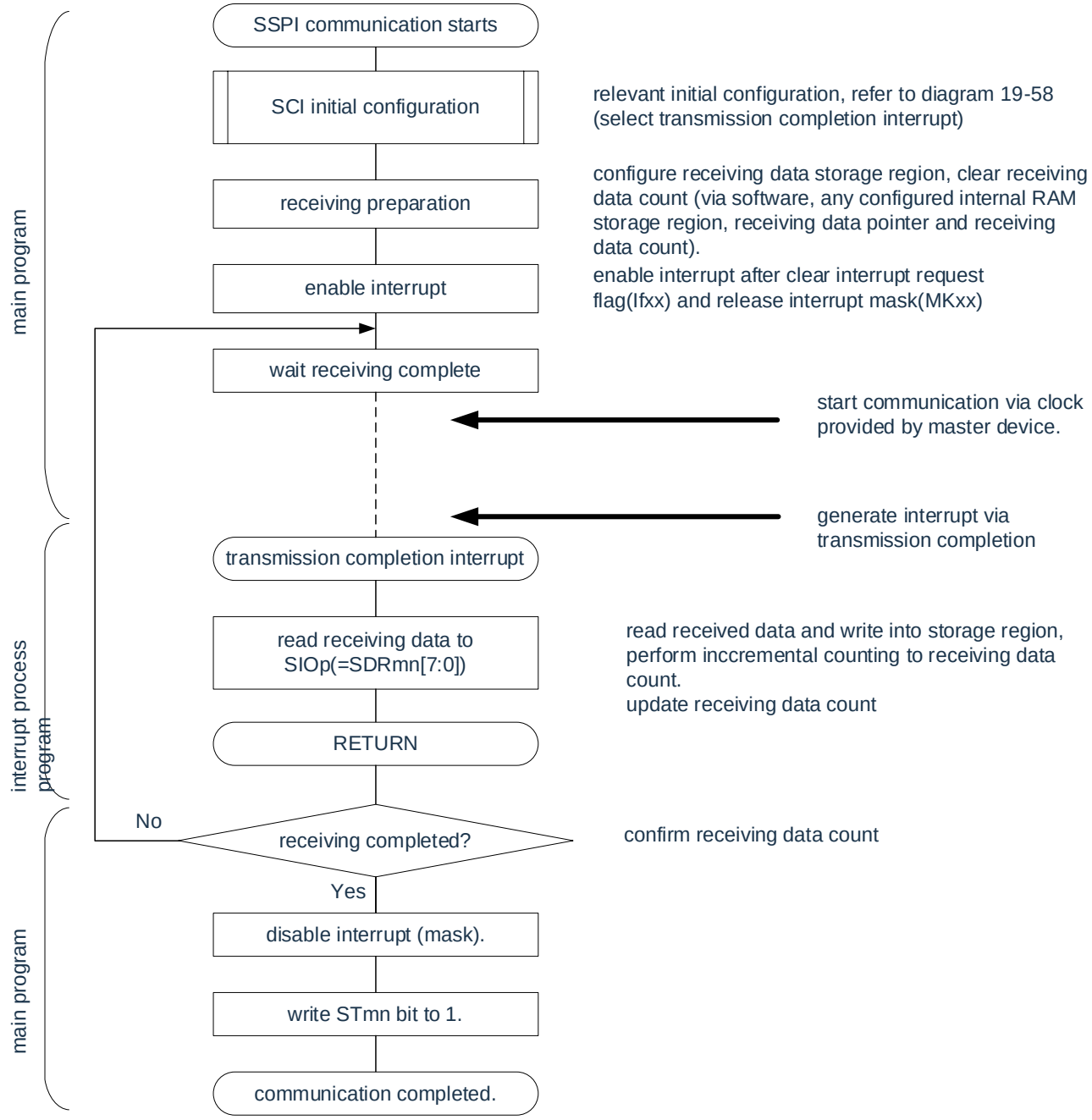
(3) Process flow (single receive mode)

Figure 19-61 Time Series of dependent receipts (single receive mode) (Type 1:DAPmn=0, CKPmn=0)



Remark m: Cell 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 19-62 Flow chart for dependent receipts (single receive mode)



19.5.6 Slave send and receive

Slave sending and receiving refers to the operation of BAT32A237 microcontroller and other devices to send and receive data in the state of input transfer clock.

3-Wire Serial I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
object channel	SCI0's Channel 0	SCI0's Channel 1	SCI0's Channel 2	SCI0's Channel 3	SCI1's Channel 0	SCI1's Channel 1
Pin Used	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
	Interrupt at that end of the transfer may be selecte (single transfer mode) or buffer air-discontinuity (continuous transfer mode).					
Error detection flag	Only the overflow error detection flag (OVFmn).					
length of transmit data	7 or 8-bit					
transfer rate	Max.$_{fMCK}$/6[Hz]Note 1,2					
data phase	Can be selected by the DAPmn bit of the SCRmn register. ·DAPmn=0: The data input/output is started when the serial clock is started. ·DAPmn=1: The data input/output is started half a clock before the serial clock starts running.					
clock phase	Can be selected by the CKPmn bit of the SCRmn register. • CKPmn=0: prime • CKPmn=1: inversion					
data orientation	MSB First or LSB First					

Note 1. The maximum transfer rate fMCK/6[Hz] is used because the external serial clock input by SCLK00, SCLK01, SCLK10, SCLK11, SCLK20, SCLK21.

2. Must be used within the scope of peripheral functional characteristics (reference data manual) that satisfy this condition and electrical characteristics.

Note 1.$_{fMCK}$: The running clock frequency of the object channel

2.m: Cell number (m=0,1)n: Channel Number (n=0~3) mn=00~03,10~11

(1) Register settings

Figure 19-63 3-wire Serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20,SSPI21) Example of Register Setting Content at Slave transmission and Reception

(a) serial mode register mn (SMRmn)

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 (fMCK)
0: SPSm register configured pre-scaler output clock CKm0
1: SPSm register configured pre-scaler output clock CKm1

interrupt source of channel n
0: Transmit completion interrupt
1: Buffer empty interrupt

(b) serial communication operation configuration registermn mn(SCRmn)

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 "19.3 control universal serial communication unit registers)

data transmit 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 regsiter mn (SDRmn) (low 8 bit: SIOp)

SDRmn	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0000000 baud rate configuration							0	configuration of transmit data/received data register							

SIOp

(d) serial output register m(SOm) .....Only configure bit of target channel

SOm	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 registerm (SOEm)….only set bit of target channel to 1.

SOEm	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 bit of target channel to 1.

SSm	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 and SCR01 registers, other fixed to "1".

Note Before the master device starts outputting the clock, the SIOp register must be set to send data.

Note 1.m: Cell 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

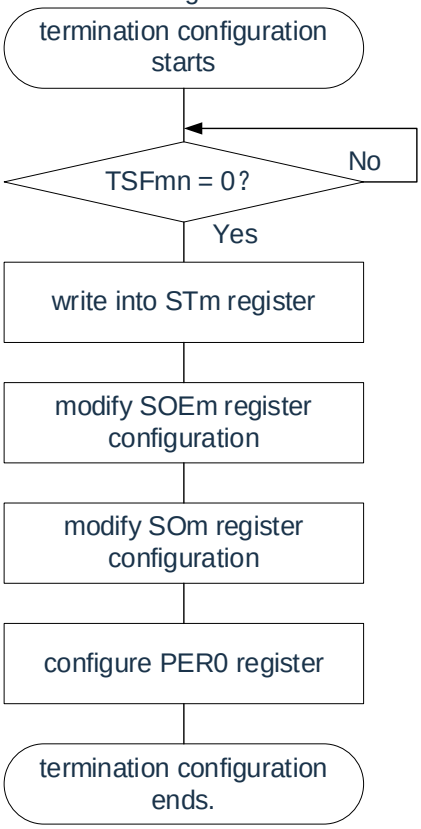
2. ☐ : Is fixed in the SSPI slave send and receive mode. ■ : Cannot set (set initial value).
×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).
0/1: The "0" or "1" is set according to the user.

(2) Procedure

Figure 19-64 Initial set-up steps for slave transmission and reception

Note Before the master device starts outputting the clock, the SIOp register must be set to send data.

Figure 19-65 Interrupt steps for slave transmission and reception



if there are ongoing data transmission, then wait till transmission completed. (if need urgent stop, then no need to wait).

set STmm bit of target channel to 1. (SEmn=0: set to operation stop state).

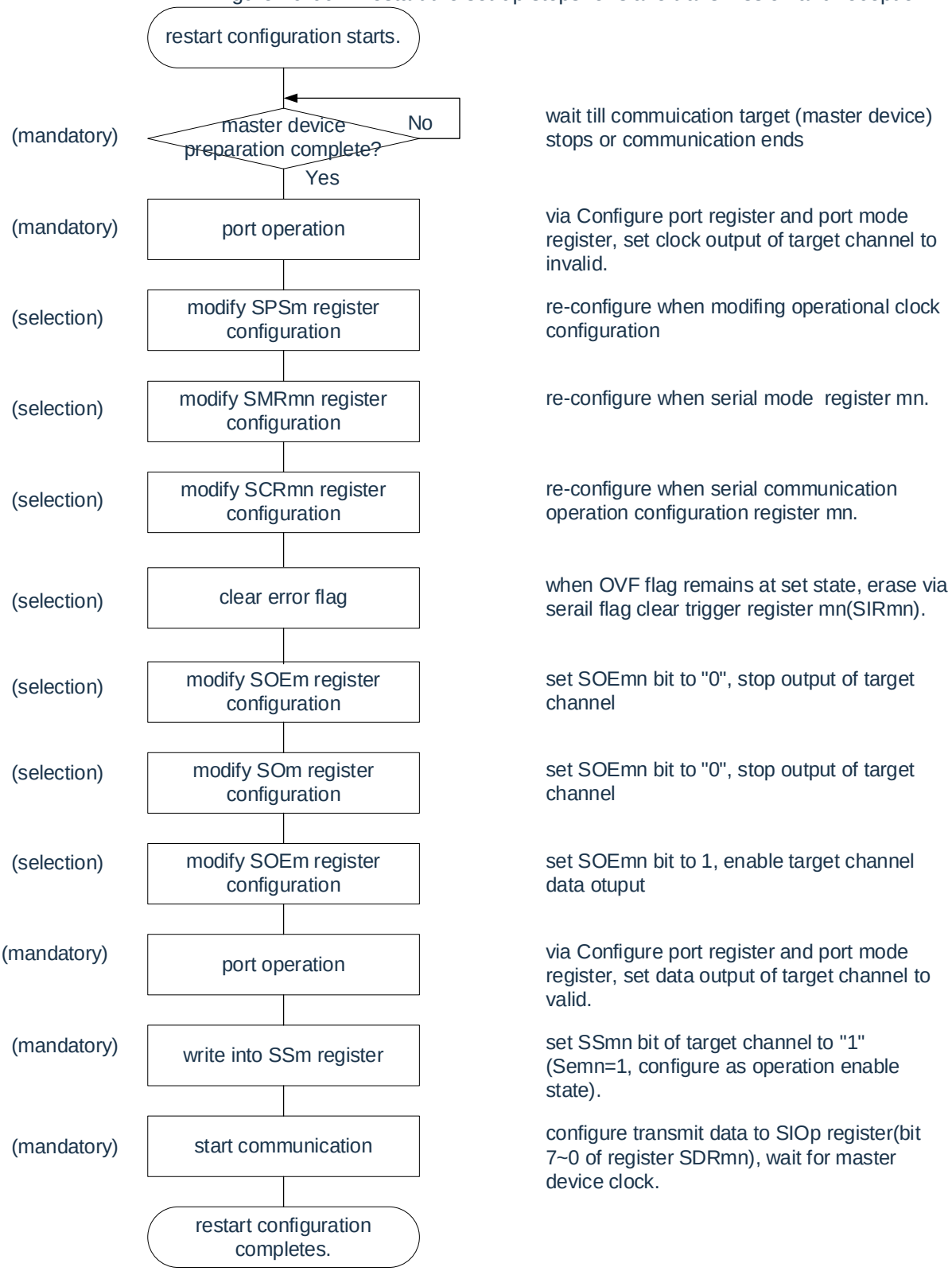
set SOEmn bit to 0, stop output of target channel

when emergency stop, based on needs, modify serial data (SOmn) voltage level of the target channel.

stop clock of universal serial communication unit, set to reset state

finish termination configuration, enter into next processing.

Figure 19-66 Restart the set-up steps for slave transmission and reception

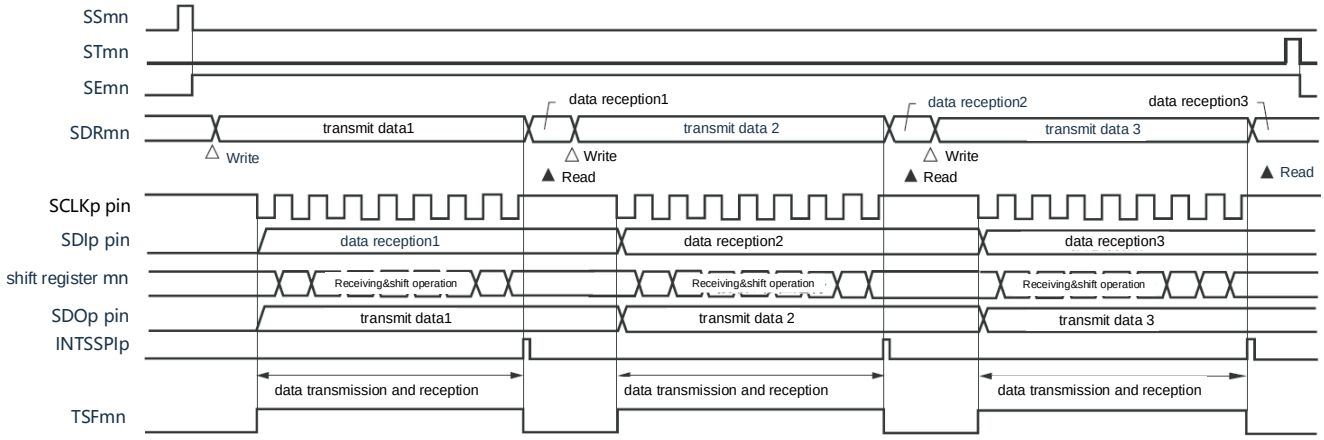


Note 1. The SIOp register setting must be sent before the master device starts outputting the clock.

2. If you override PER0 in the abort setting to stop providing the clock, you must wait until the communication object (master) stops or the communication is over for the initial set-up instead of restarting.

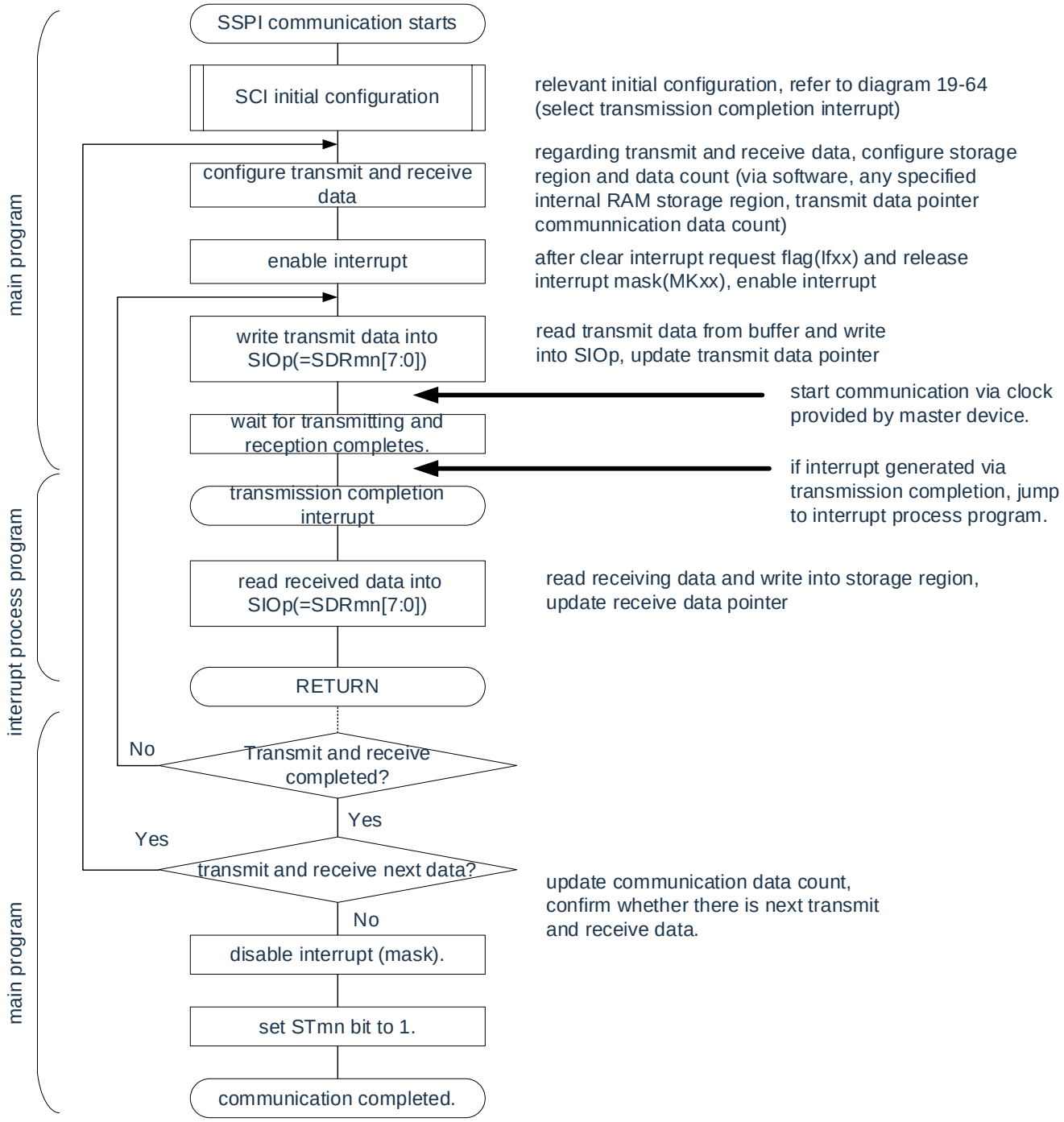
(3) Process flow (single send and receive mode)

Figure 19-67 Time Series for Slave send and receive (Single Send and Receive Mode) (Type 1: DAPmn=0, CKPmn=0) Figure.



Remark m: Cell 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

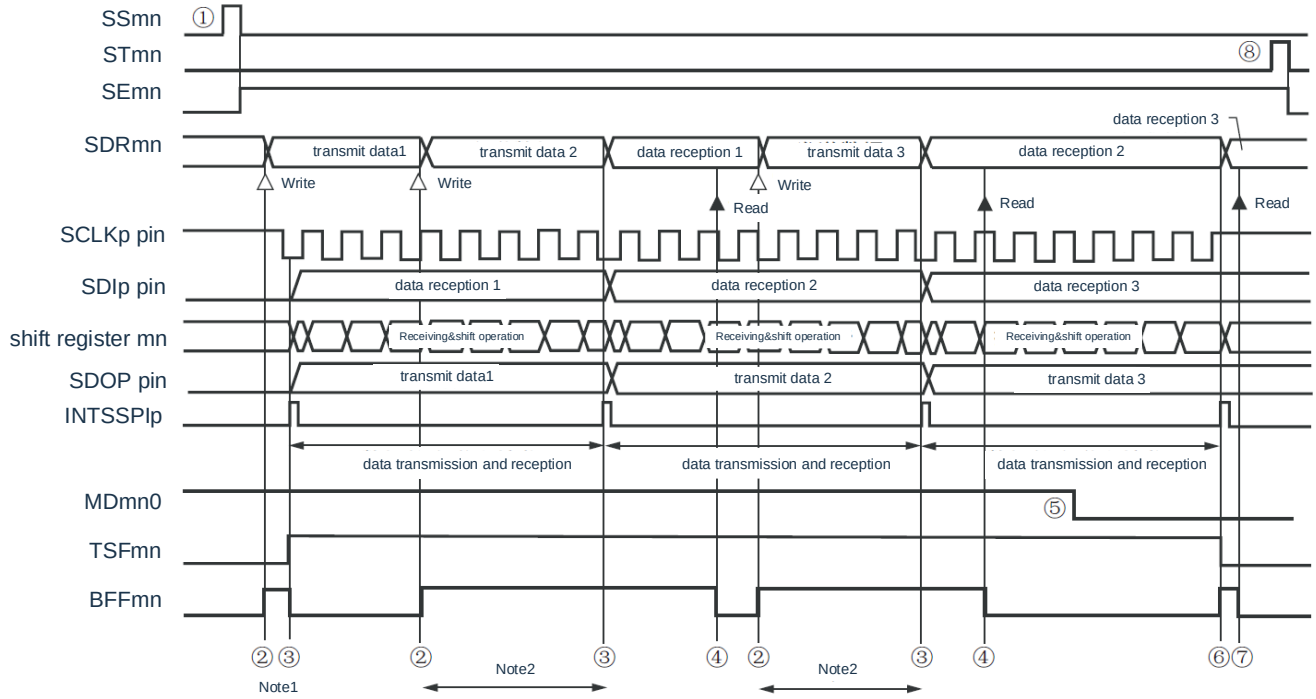
Figure 19-68 Flowchart for slave send and receive (single send and receive mode)



Note Before the master device starts outputting the clock, the SIOp register must be set to send data.

(4) Processing Flow (Continuous Send and Receive Mode)

Figure 19-69 Time Series for Slave send and receive (Continuous Send and Receive Mode) (Type 1: DAPmn=0, CKPmn=0) Figure.



Note1. Rewrite the transmission data if the BFFmn bit of the serial status register mn (SSRmn) is "1" (when valid data is stored in SDRmn).

2. Send data can be read if the SDRmn register is read during this period. At this point, the transfer operation is not affected.

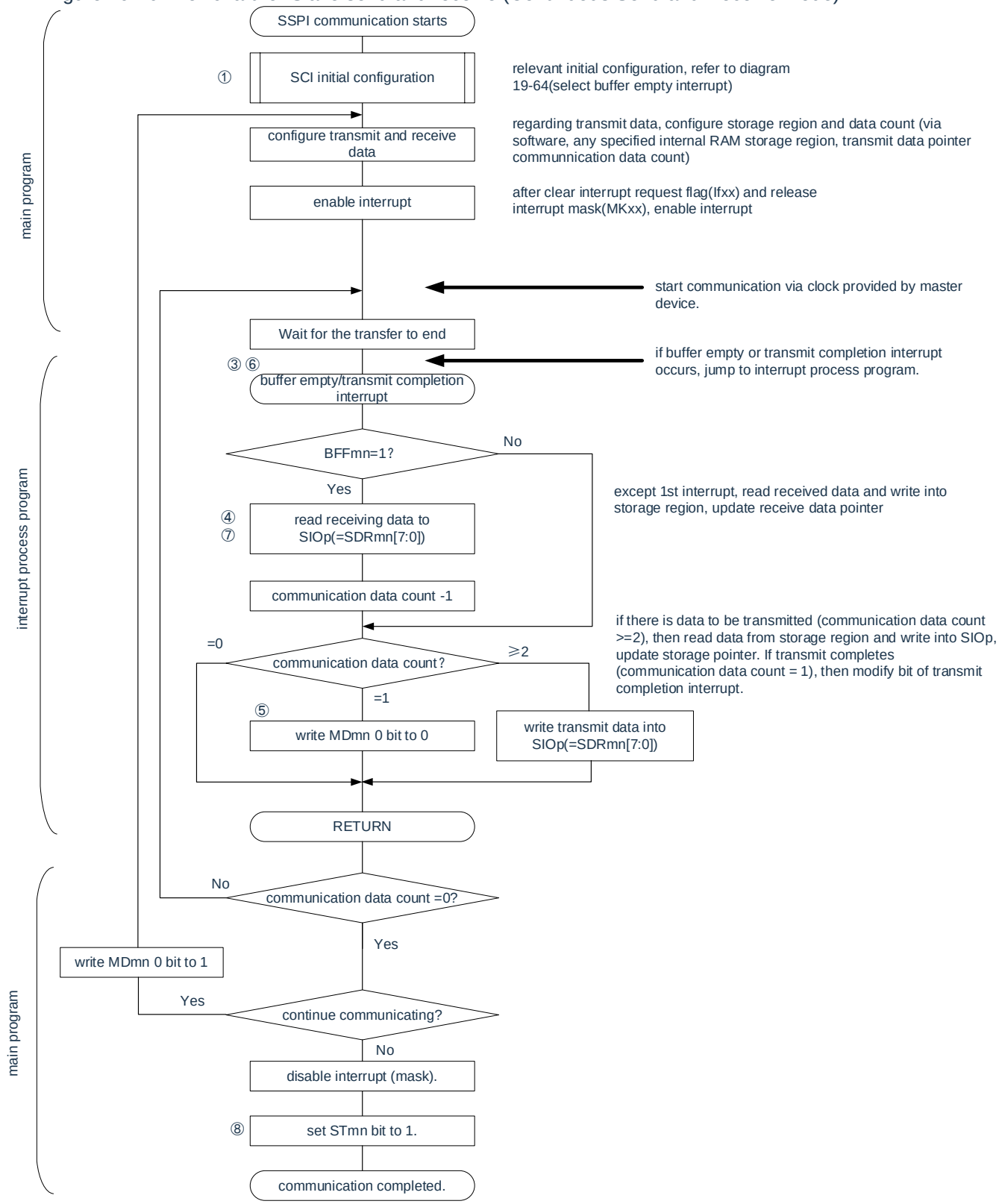
Note The MDmn0 bit of the serial mode register mn(SMRmn) can be rewritten even in operation. However, in order to be able to catch the end of the transmission of the last transmitted data interrupt, it is necessary to override before starting the last transmission.

Note 1. (1) to (8) in Figure corresponds to (1) to (8) in Figure 19-70 Slave send and receive.

2.m: Cell 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 19-70 Flowchart for Slave send and receive (Continuous Send and Receive Mode)



Note Before the master device starts outputting the clock, the SIOp register must be set to send data.

Remark (1) to (8) in the figure correspond to (1) to (8) in "Figure 19-69 Sequential Sending and Receiving Pattern".

19.5.7 Calculation of transmission clock frequency

The transfer clock frequency for 3-line serial I/O (SSPI00, SSPI01, SSPI10, SSPI20,SSPI21) communication can be calculated using the following formula.

(1) master device

(Transfer Clock Frequency)={Runtime Clock ($_{fMCK}$) Frequency}/(SDRmn[15:9]+1)÷2[Hz]

(2) slave

(Transfer Clock Frequency)={Serial Clock (SCLK) Frequency provided by the master device} Note [Hz]

Note The maximum allowable transfer clock frequency is fMCK/6.

Remark The value of SDRmn[15:9] is 0~127 because it is bit15~9 of serial data register mn(SDRmn).

The runtime clock (fMCK) depends on the bit15 (CKSmn) of the serial clock selection register m (SPSm) and the serial mode register mn (SMRmn).

Table 19-2 The Choice of 3-line Serial I/O Operating Clock

SMRmn register	SPSm register								Runtime Clock (f_{MCK}) Note	
CKSmn	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00		f_{CLK}=32 MHz Runtime
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 fCLK (change the value of the System Clock Control Register (CKC), you must change after stopping Universal Serial Communication Unit (SCI) =000FH.

Note 1.X: Ignore
2.m: Cell number (m=0,1)n: Channel Number (n=0~3) mn=00~03,10~11

19.5.8 Processing steps for errors occurring during 3-wire serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21) communication

The steps for handling errors occurring during 3-wire serial I/O (SSPI00, SSPI01, SSPI10, SSPI11, SSPI20, SSPI21) communication are shown in Figure 19-75.

Figure 19-75 Handling steps when overflow errors occur

software operation	Hardware Status	Comments
Read the serial data →	The BFFmn bit of the SSRmn register is "0" and the channel n is in a receiver state.	This is to prevent an overflow error from occurring to end the next receipt during error handling.
Read the serial status register mn (SSRmn)		The type of error is determined and the read value is used to clear the error flag.
Clear trigger register mn for serial flag →	Clear the error flag.	By writing the read value of the SSRmn register directly to the SDIRmn register, errors in the read operation can only be cleared.

Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) mn=00~03,10~11

19.6 Clock synchronization serial communication operation of slave selectioninput function

Channel 0 of the SCI0 is a channel for clock synchronous serial communication that supports the dependent select input function.

[Sending and Receiving Data]
- 7-bit or 8-bit data length
- Phase control for transmitting and receiving data
- MSB/LSB First Choice
- Level settings for sending and receiving data

[Clock Control]
- Phase control of input/output clock
- A transmission period generated by a pre-divider and an intra-channel counter is set.
- Maximum Transfer Rate Note Dependent
  Communication: Max.$_{fMCK}$/6

[Interrupt Capabilities]
- interrupt delivery end, buffer interrupt

[Error Detection Flag]
- overflow error

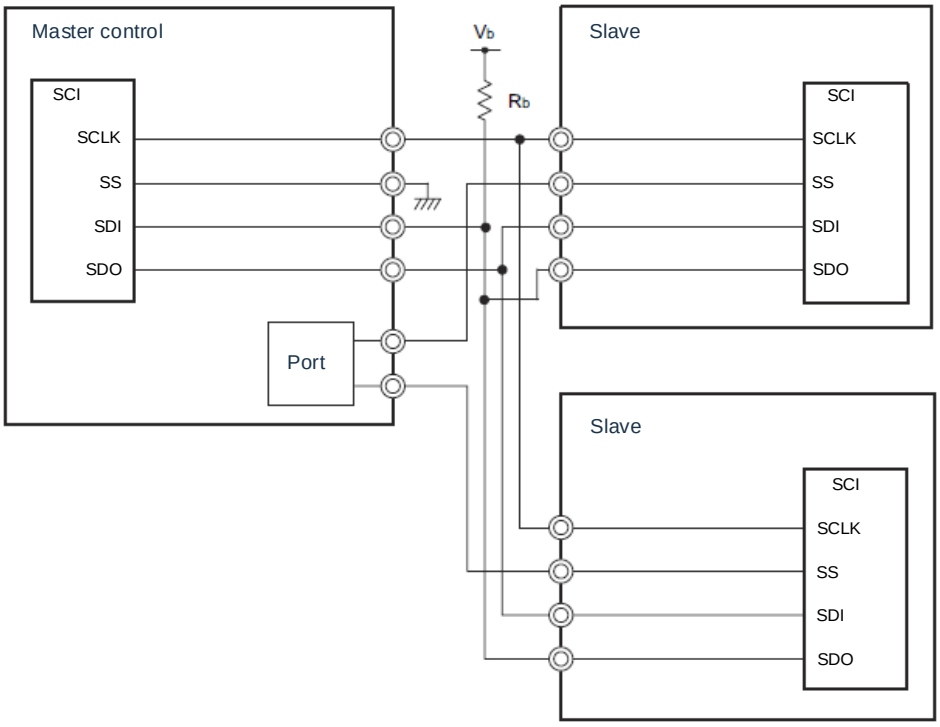
Note Must be used within a range that satisfies the $_{SCLK}$ Cycle Time (tKCY) characteristic. Refer to the data guide for details.

The Slave selectionInput function runs in three communications:
- Slave send (Reference 19.6.1)
- slave receive (Reference 19.6.2)
- Slave send and receive (Reference 19.6.3)

The invention can enable 1 main control device to connect a plurality of slave devices to communicate by using the slave selection input function. The master control device outputs a slave selection signal to the slave devices of the communication object, and each slave determines whether it is selected. When the slave device is selected as the communication object, the SDO pin can transmit data to the master device; The SDO pin is set to Nch-O.D and pulled on the node in an environment where multiple slave devices are connected. In addition, the serial clock of the main control device is not transmitted and received.

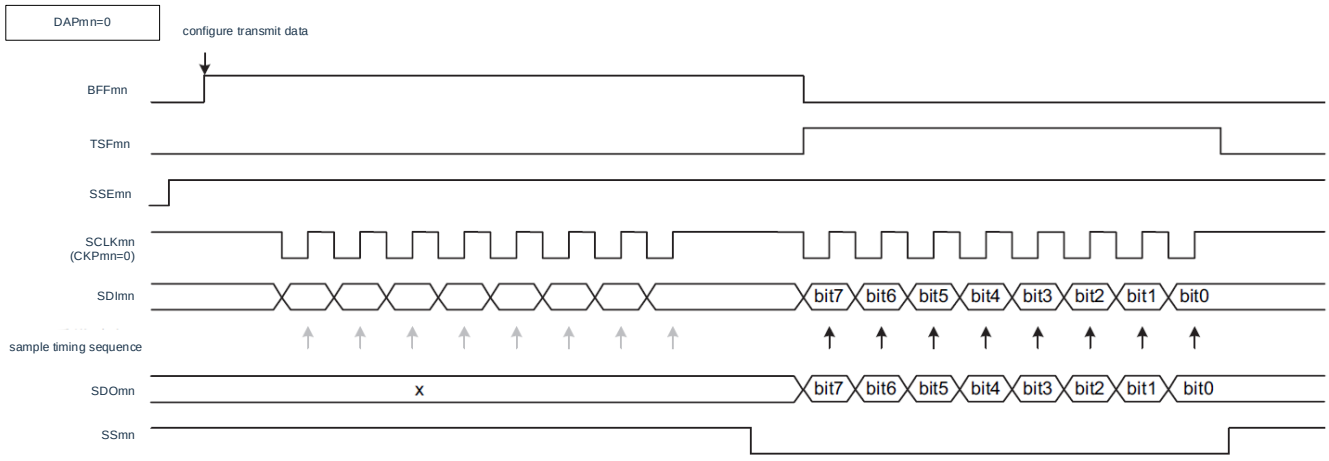
Note A slave selectionsignal must be output through the operation of the port.

Figure 19-76 Structural example of a slave selectioninput function

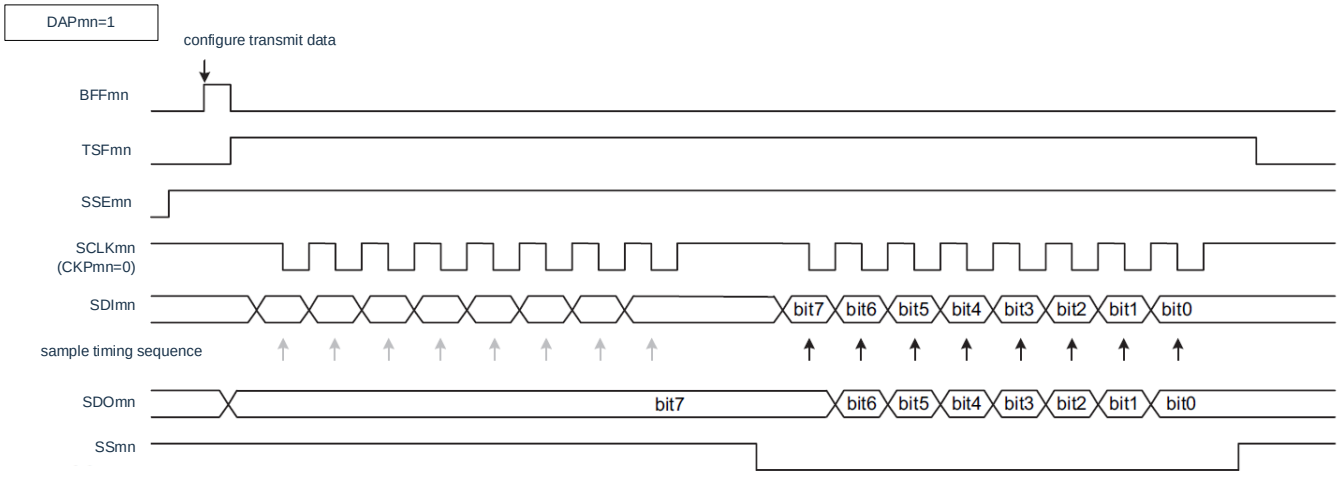


Note The SDO00 pin is selected as an N-channel drain open-circuit output mode.

Figure 19-77 Time Series of Slave selectionInput Function



During the period when SSmn is high, no transmission is performed even at the descending edge of SCKmn (serial clock) and no sampling of received data synchronized with the ascending edge is performed.
During a period when the SSmn is low, output data (shift) is synchronized with the falling edge of the serial clock and data is received synchronously with the rising edge.



When the DAPmn bit is '1', if the transmission data is set during the high level of the SSmn, the initial data (bit7) is supplied to the data output. However, even the rising edge of the SCLKmn (serial clock) does not shift, and does not sample the accepted data synchronized with the falling edge. If the SSmn becomes low, the output data (shift) is synchronized with the next rising edge and the data is received in synchronization with the falling edge.

Remark m: Cell number (m=0)n: Channel number (n=0)

19.6.1 Slave send

Slave sending is a run in which BAT32A237 sends data to other devices in a state where a transfer clock is entered from another device.

slave selectioninput function	SSPI00
object channel	Channel 0 for SCI0
Pin Used	SCLK00, SDO00, SS00
interrupt	INTSSPI00
	Interrupt at that end of the transfer may be selecte (single transfer mode) or buffer air-discontinuity (continuous transfer mode).
Error detection flag	Only the overflow error detection flag (OVFmn).
length of transmit data	7 or 8-bit
transfer rate	Max.$_{fMCK}$/6[Hz]Note 1,2
data phase	Can be selected by the DAPmn bit of the SCRmn register. ·DAPmn=0: Start the data output when the serial clock starts running. ·DAPmn=1: The data output is started half a clock before the serial clock starts running.
clock phase	Can be selected by the CKPmn bit of the SCRmn register. • CKPmn=0: prime • CKPmn=1: inversion
data orientation	MSB First or LSB First
slave selectioninput function	You can select the operation of the slave selectionfunction.

Note 1. Since the external serial clock input to the SCLK00 pin is sampled internally and used, the maximum transfer rate is $_{fMCK}$/6[Hz].

2. Must be used within the scope of peripheral functional characteristics (reference data manual) that satisfy this condition and electrical characteristics.

Note 1.$_{fMCK}$: The running clock frequency of the object channel

2.m: Cell number (m=0)n: Channel number (n=0)

(1) Register settings

Figure 19-78 Slave selectionInput Function (SSPI00) Example of register setting content upon slave transmission (1/2)

(a) serial mode register mn (SMRmn)

	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	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

channel n operational clock (fMCK)
0: SPSm register configured pre-scaler output clock CKm0
1: SPSm register configured pre-scaler output clock CKm1

interrupt source of channel n
0: Transmit completion interrupt
1: Buffer empty interrupt

(b) serial communication operation configuration registermn mn(SCRmn)

	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	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

data and clock phase selection (details refer to "19.3 control universal serial communication unit registers)

data transmit 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 regsiter mn (SDRmn) (low 8 bit: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 baud rate configuration							0	transmit data configuration							

SIOp

(d) serial output register m (SOm)…. Only configure bit of target channel

	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	×	×	×	0/1

(e) serial output enable register m (SOEm)…. Only set bit of target channel to "1".

	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

Note 1.m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

2. □ : Fixed in SSPI slave send mode. ■ : Cannot set (set initial value).

×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).

0/1: The "0" or "1" is set according to the user.

Figure 19-78 Slave selectionInput Function (SSPI00) Example of register setting content upon slave transmission (2/2)

(f) serial channel start register m (SSm) …. Only set bit of target channel to 1.

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

Note 1.m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

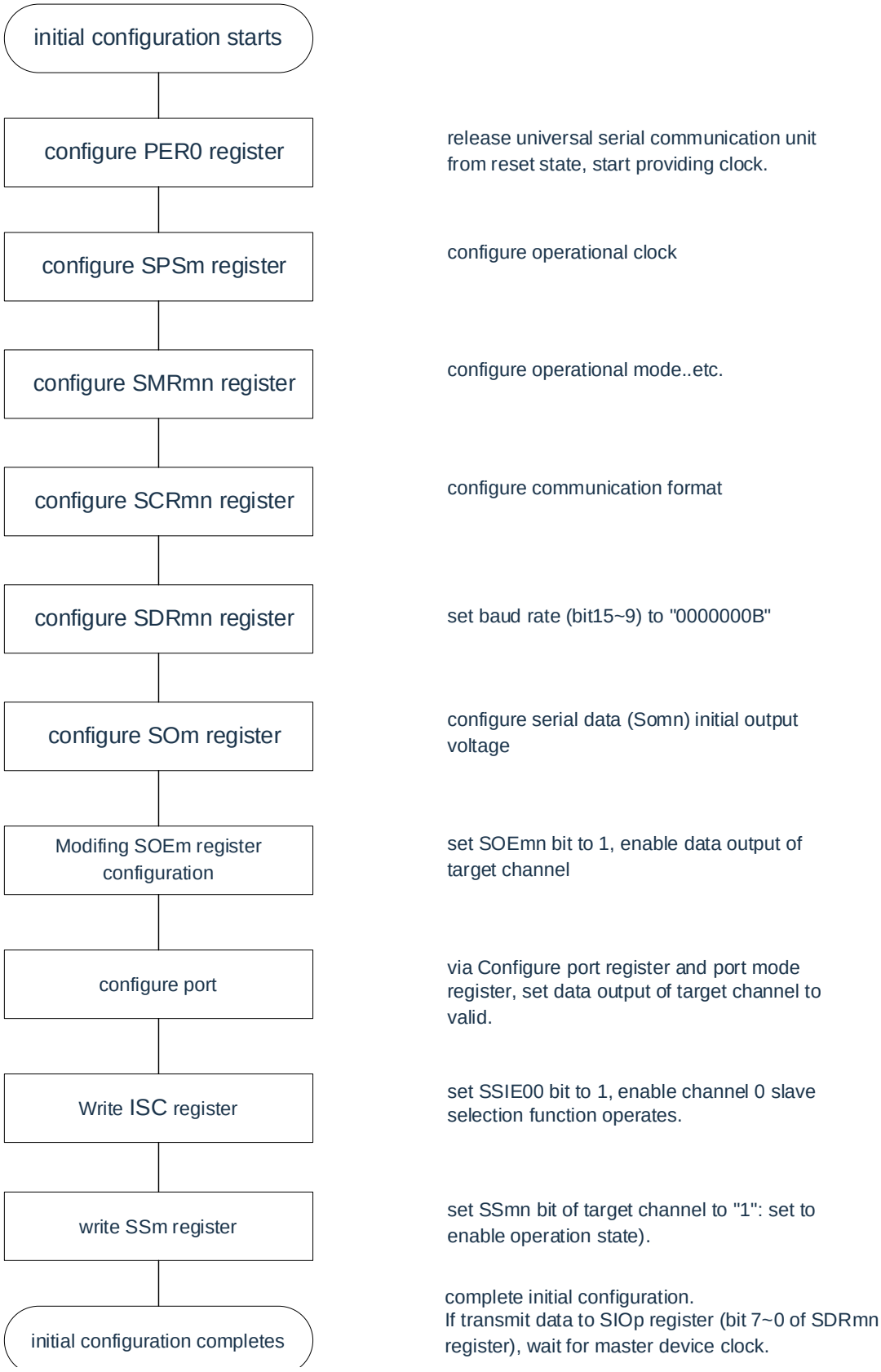
2. □ : Fixed in SSPI slave send mode. ■ : Cannot set (set initial value).

×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).

0/1: The "0" or "1" is set according to the user.

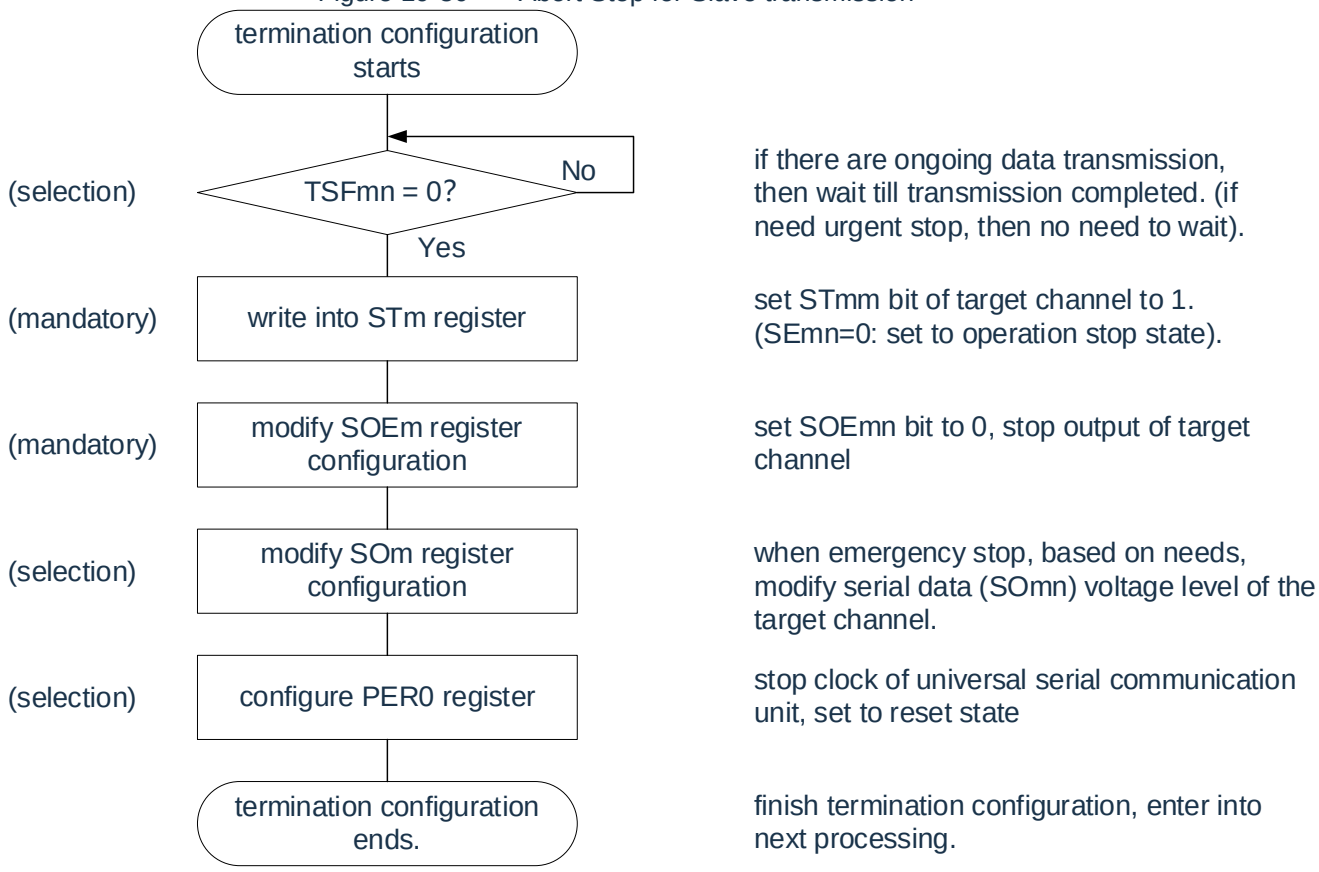
(2) Operation Procedure

Figure 19-79 Initial set-up steps for slave transmission



Remark m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

Figure 19-80 Abort Step for Slave transmission

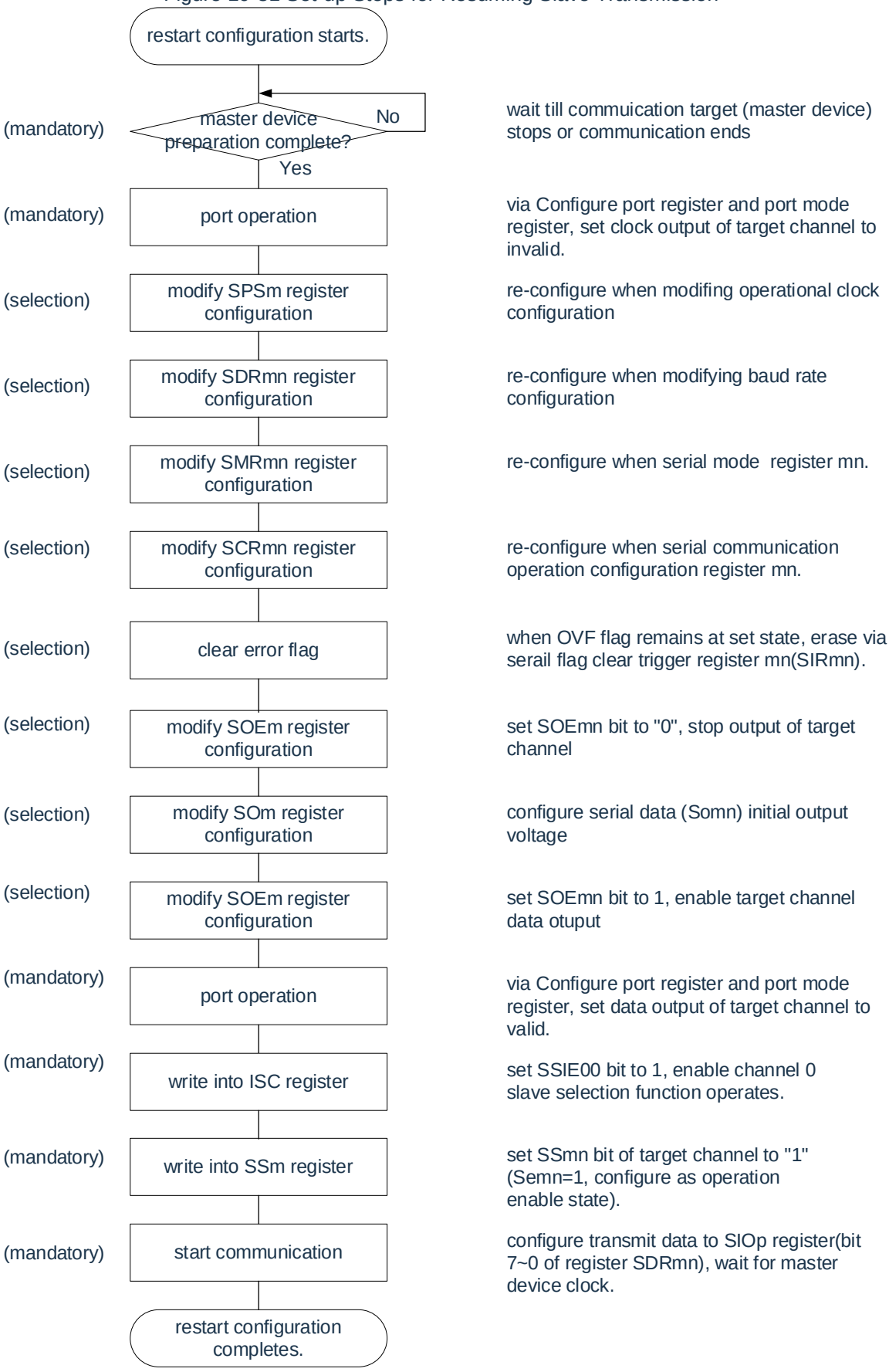


Remark m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

Note 1. If you override PER0 in the abort setting to stop providing the clock, you must wait until the communication object (master) is stopped or the communication is over for the initial set-up instead of restarting the set-up.

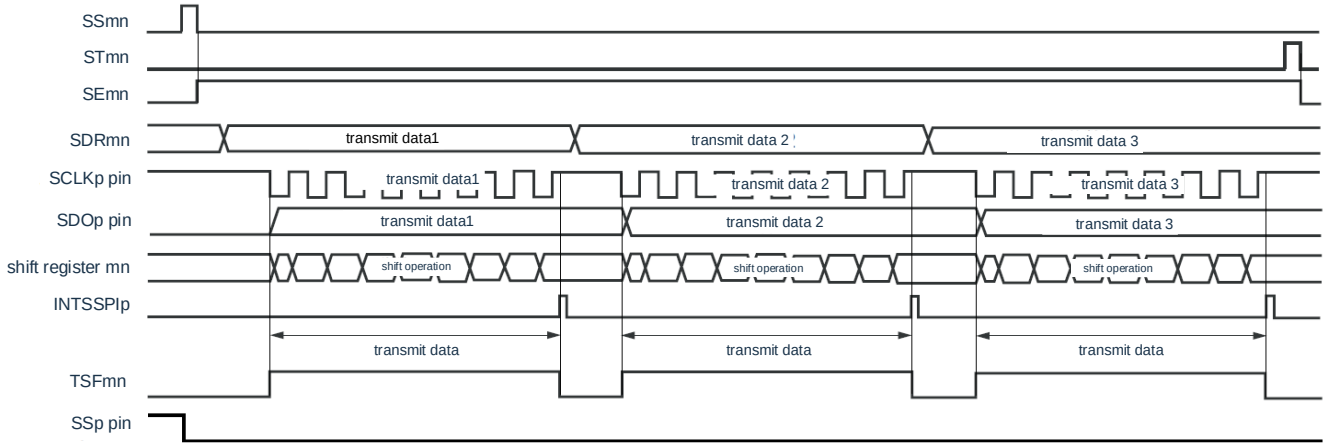
2.m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

Figure 19-81 Set-up Steps for Resuming Slave Transmission



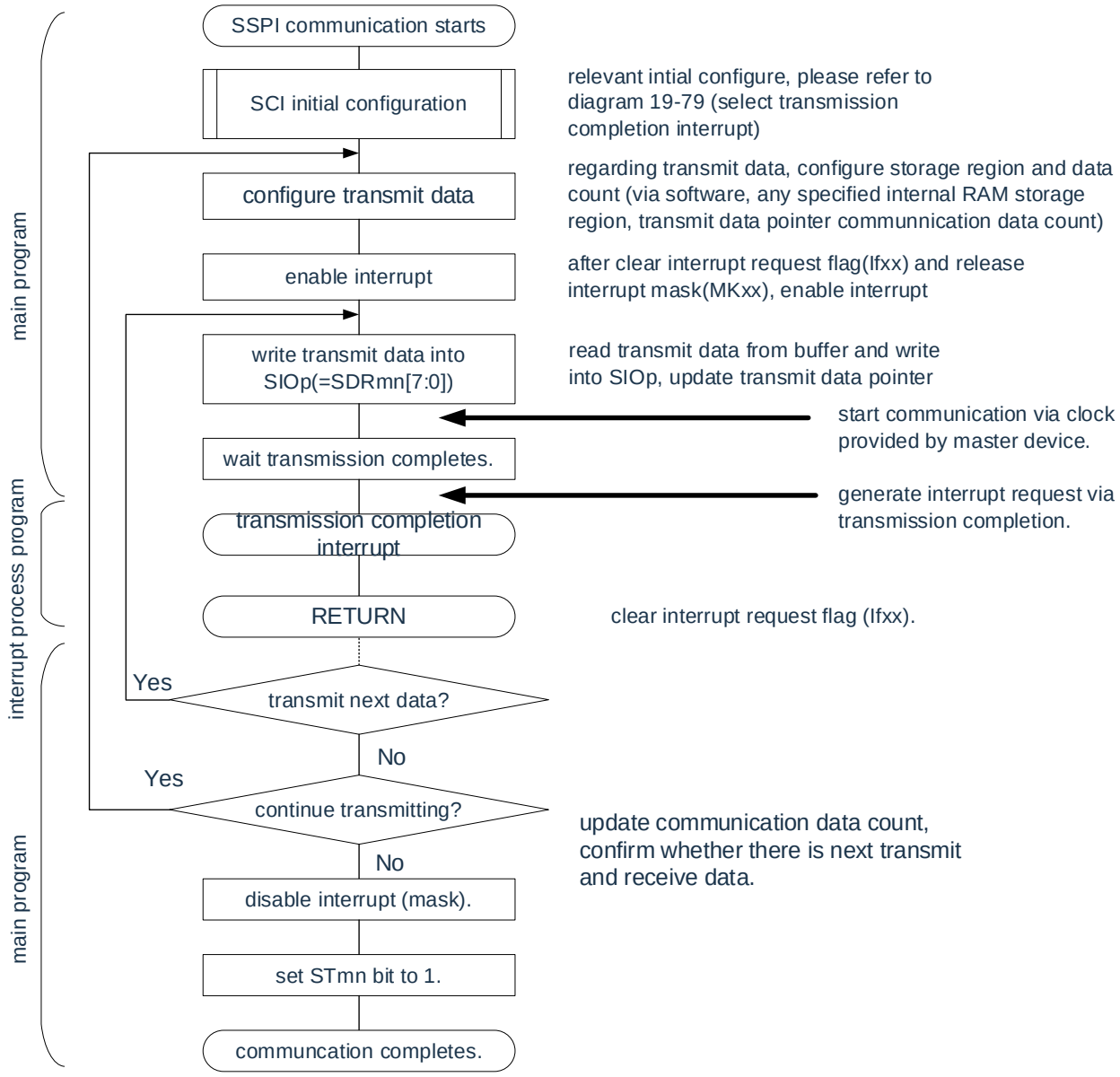
(3) Process flow (single send mode)

Figure 19-82 Time Series for Slave send (single send mode) (Type 1:DAPmn=0, CKPmn=0) Type 1



Remark m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

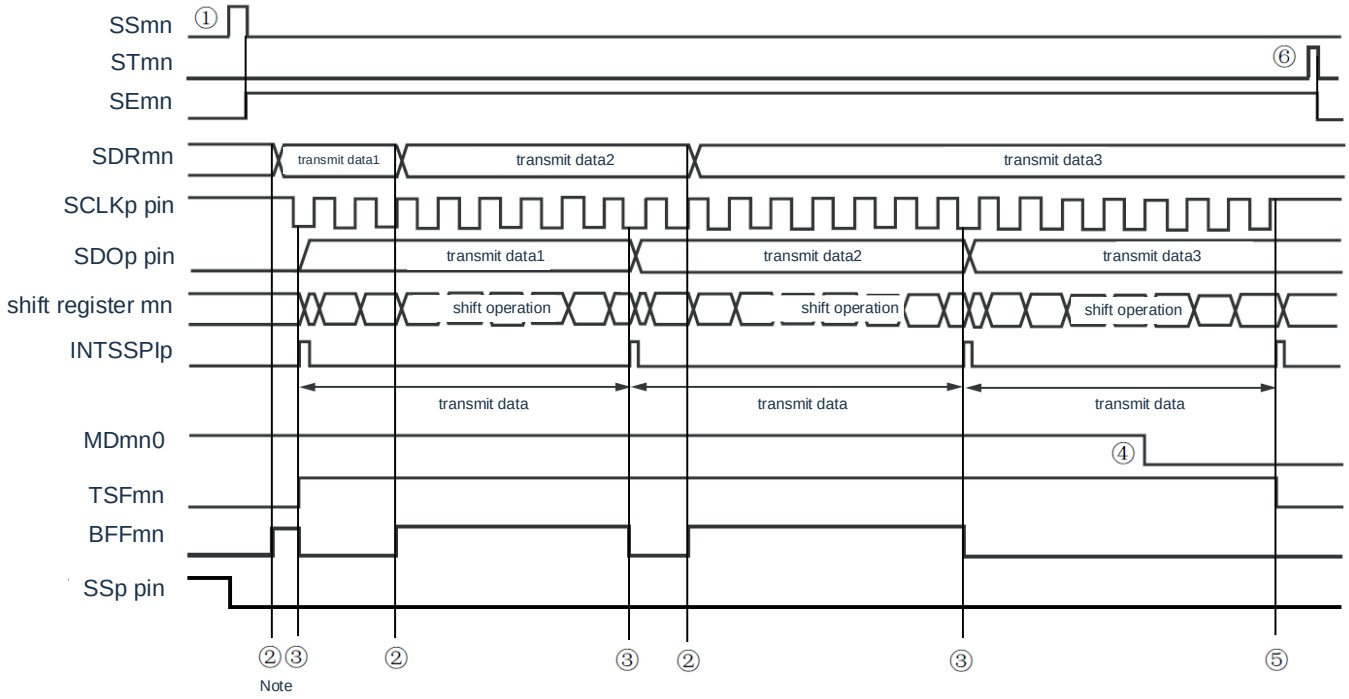
Figure 19-83 Flowchart for Slave send (Single Send Mode)



Remark m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

(4) Process flow (continuous send mode)

Figure 19-84 Time Series for Slave send (Continuous Send Mode) (Type 1:DAPmn=0, CKPmn=0).

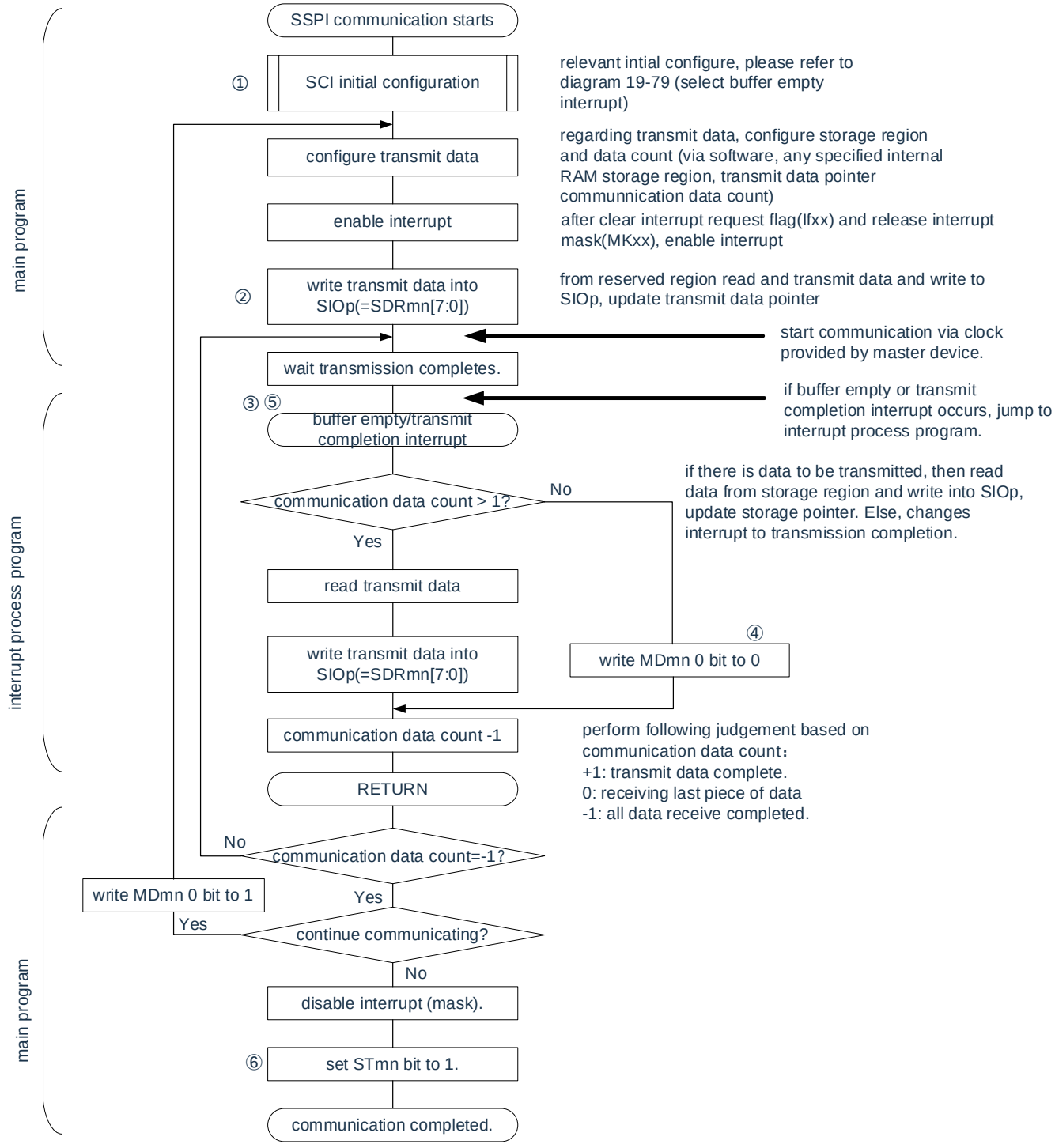


Note The transmission data is rewritten if the BFFmn bit of the serial status register mn (SSRmn) is "1" (SDRmn) when the valid data is stored in the serial data register mn (SDRmn).

Note The MDmn0 bit of the serial mode register mn(SMRmn) can be rewritten even in operation. However, you must override before you start transferring the last bit.

Remark m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

Figure 19-85 Flowchart for Slave send (Continuous Send Mode)



Note 1. (1) to (6) in the figure corresponds to (1) to (6) in Figure 19-84.

2.m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

19.6.2 slave receive

A slave receive is a run in which BAT32A237 receives data from other devices in a state in which a transfer clock is input from other devices.

slave selectioninput function	SSPI00
object channel	Channel 0 for SCI0
Pin Used	SLK00, SDI00, SS00
interrupt	INTSSPI00
	Interrupt at that end of the transfer only (Disable from setting buffer air interrupt).
Error detection flag	Only the overflow error detection flag (OVFmn).
length of transmit data	7 or 8-bit
transfer rate	$_{MaxfMCK}$/6[Hz] Notes 1,2
data phase	Can be selected by the DAPmn bit of the SCRmn register. ·DAPmn=0: Start the data output when the serial clock starts running. ·DAPmn=1: The data output is started half a clock before the serial clock starts running.
clock phase	Can be selected by the CKPmn bit of the SCRmn register. • CKPmn=0: prime • CKPmn=1: inversion
data orientation	MSB First or LSB First
slave selectioninput function	You can choose to run the slave selectioninput function.

Note1. Since the external serial clock input to the SCLK00 pin is sampled internally and used, the $_{maximum\ transfer\ rate\ is}$ $_{fMCK}$/6[Hz].

2. Must be used within the scope of peripheral functional characteristics (reference data manual) that satisfy this condition and electrical characteristics.

Note 1.$_{fMCK}$: The running clock frequency of the object channel

2.m: Cell number (m=0)n: Channel number (n=0)

(1) Register settings

Figure 19-86 Example of register setting contents (1/2) when a slave selectioninput function (SSPI00) is dependent

(a) serial mode register mn (SMRmn)

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

channel n operational clock (fMCK)
0: SPSm register configured pre-scaler output clock CKm0
1: SPSm register configured pre-scaler output clock CKm1

interrupt source of channel n
0: Transmit completion interrupt1: Buffer empty interrupt

(b) serial communication operation configuration registermn mn(SCRmn)

	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	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

data and clock phase selection (details refer to "19.3 control universal serial communication unit registers)

data transmit 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 regsiter mn (SDRmn) (low 8 bit: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 baud rate configuration							0	received 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
SOm					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
SOEm													SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	×	×	×

Note 1.m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

2. ☐ : Set in dependent receive mode for fixed. ▒: Cannot set (set initial value).
×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).
0/1: The "0" or "1" is set according to the user.

Figure 19-86 Example of register setting contents (2/2) when a slave selectioninput function (SSPI00) is dependent

(f) serial channel start register m (SSm) …. Only set bit of target channel to 1.

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

Note 1.m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

2. ▣ : Set in dependent receive mode for fixed. ■: Cannot set (set initial value).

×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).

0/1: The "0" or "1" is set according to the user.

(2) Procedure

Figure 19-87 Initial set-up steps for dependent receipts

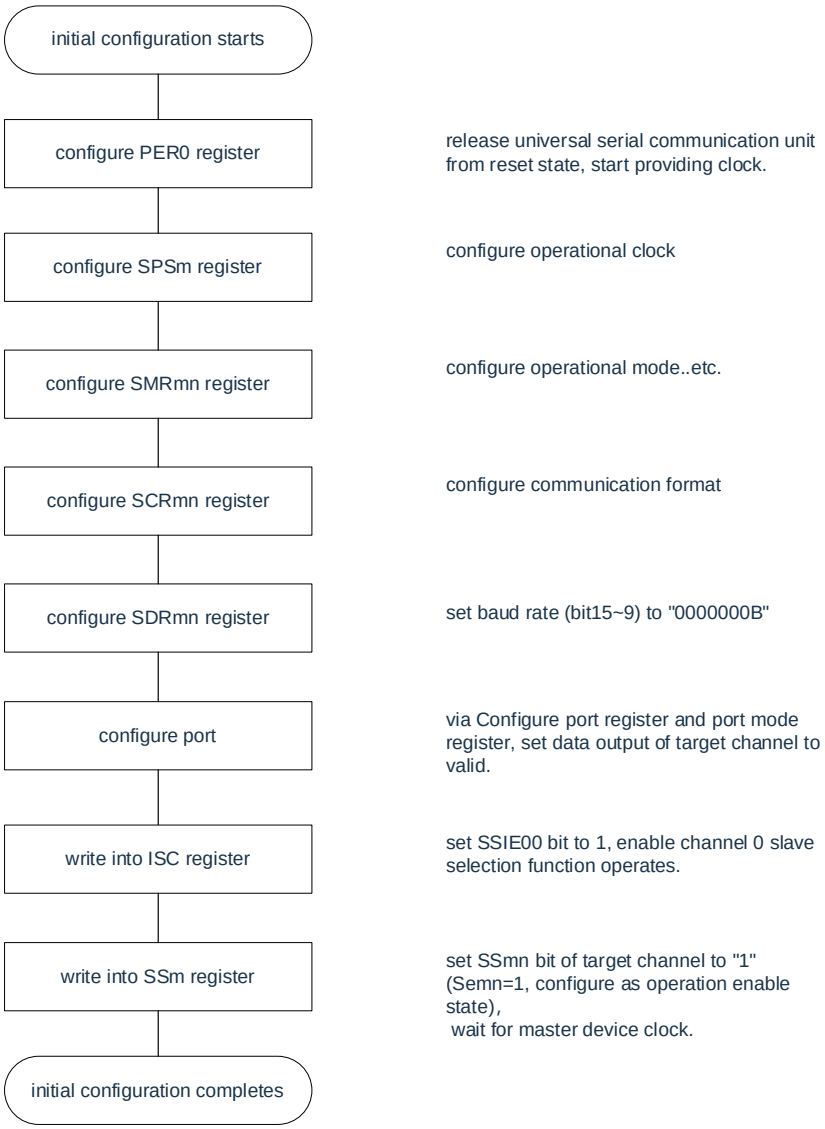
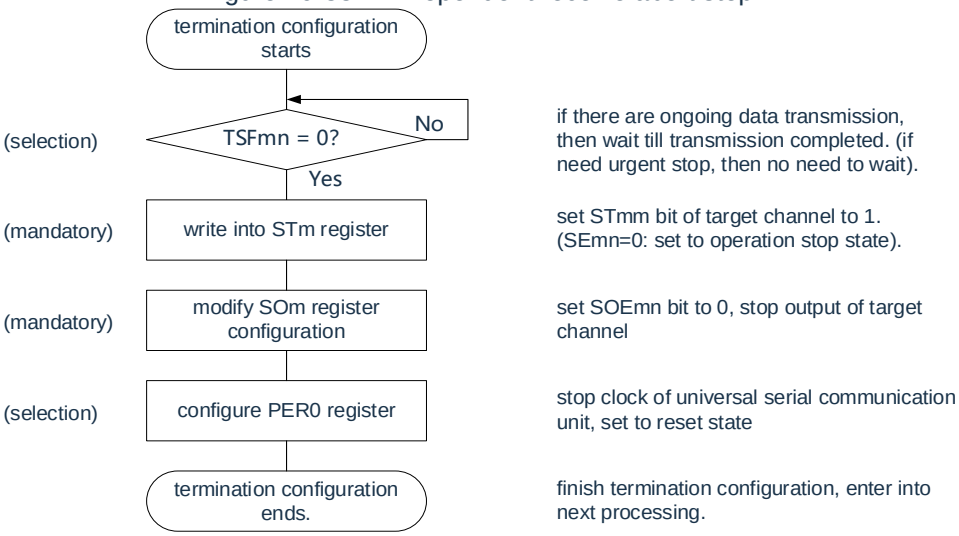
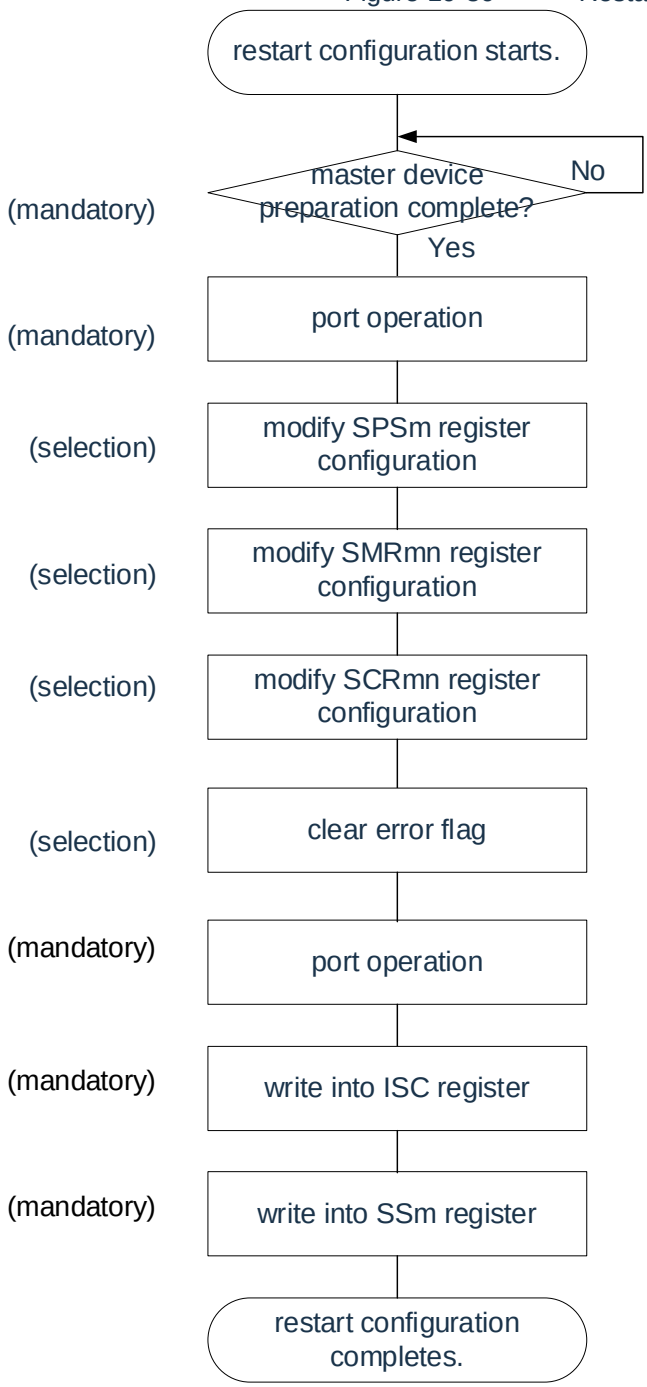


Figure 19-88 Dependent receive abort step



Remark m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

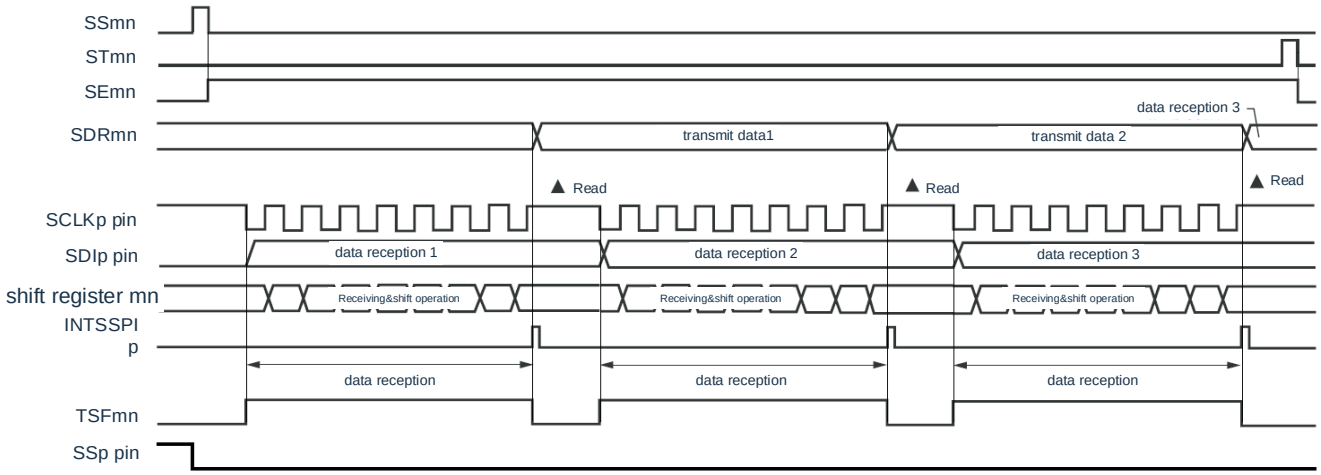
Figure 19-89 Restart set-up steps for dependent receipts



Remark m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

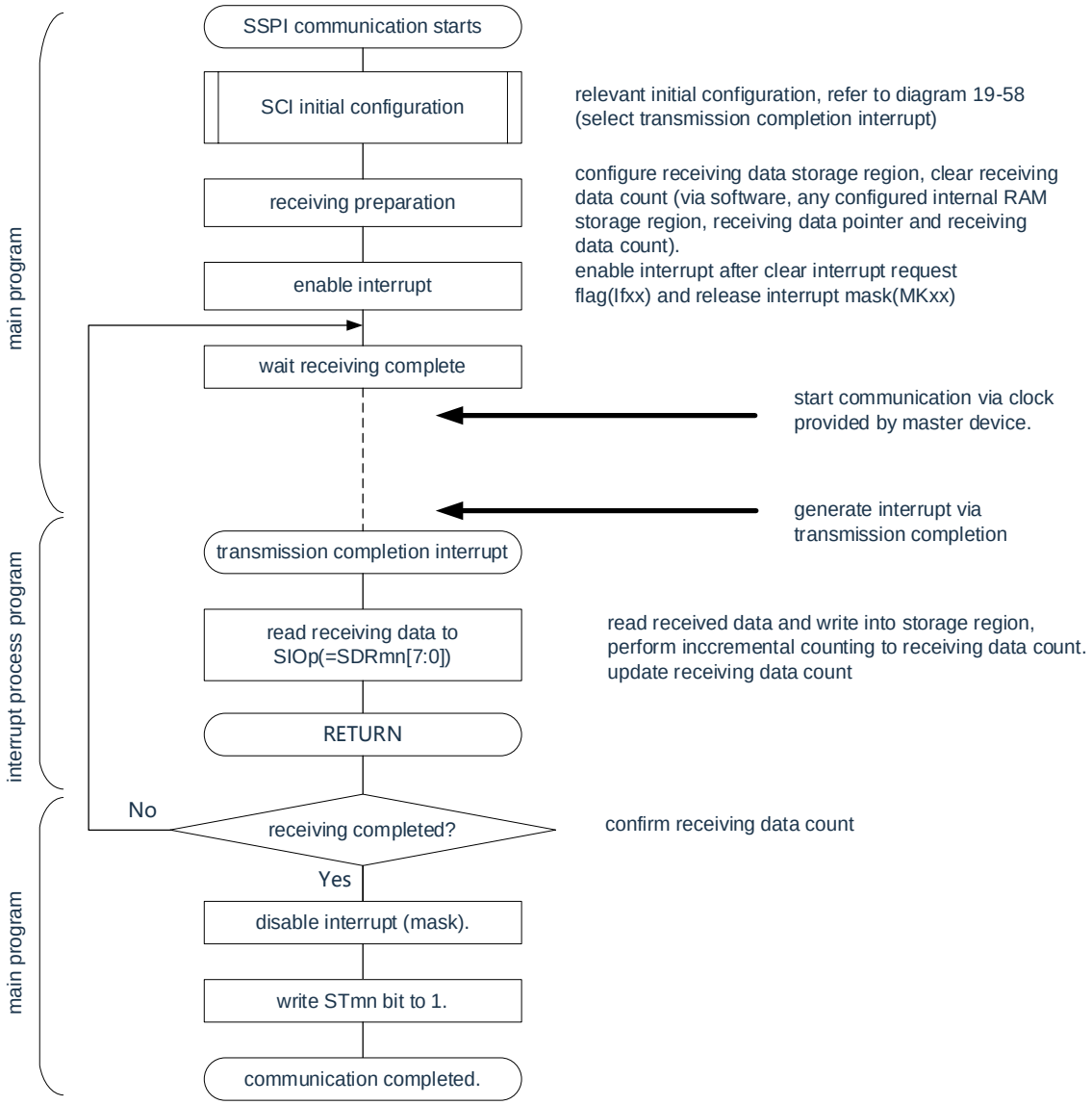
(3) Process flow (single receive mode)

Figure 19-90 Time Series of dependent receipts (single receive mode) (Type 1:DAPmn=0, CKPmn=0)



Remark m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

Figure 19-91 Flow chart for dependent receipts (single receive mode)



19.6.3 Slave send and Receive

Slave sending and receiving refers to the operation of sending and receiving data from BAT32A237 and other devices in the state of input transfer clocks from other devices.

slave selectioninput function	SSPI00
object channel	Channel 0 for SCI0
Pin Used	SCLK00, SDI00, SDO00, SS00
interrupt	INTSSPI00 Interrupt at that end of the transfer may be selecte (single transfer mode) or buffer air-discontinuity (continuous transfer mode).
Error detection flag	Only the overflow error detection flag (OVFmn).
length of transmit data	7 or 8-bit
transfer rate	Max.$_{fMCK}$/6[Hz]Note 1,2
data phase	Can be selected by the DAPmn bit of the SCRmn register. ·DAPmn=0: Start the data output when the serial clock starts running. ·DAPmn=1: The data output is started half a clock before the serial clock starts running.
clock phase	Can be selected by the CKPmn bit of the SCRmn register. • CKPmn=0: prime • CKPmn=1: inversion
data orientation	MSB First or LSB First
slave selectioninput function	You can choose to run the slave selectioninput function.

Note1. Since the external serial clock input to the SCLK00 pin is sampled internally and used, the maximum transfer rate is $_{fMCK}$/6[Hz].

2. Must be used within the scope of peripheral functional characteristics (reference data manual) that satisfy this condition and electrical characteristics.

Note 1.$_{fMCK}$: The running clock frequency of the object channel

2.m: Cell number (m=0)n: Channel number (n=0)

(1) Register settings

Figure 19-92 Slave selectionInput Function (SSPI00) Example of Register Setting Content at Slave transmission and 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
SMRmn	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 (fMCK)
0: SPSm register configured pre-scaler output clock CKm0
1: SPSm register configured pre-scaler output clock CKm1

interrupt source of channel n
0: Transmit completion interrupt
1: Buffer empty interrupt

(b) serial communication operation configuration registermn mn(SCRmn)

	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	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 "19.3 control universal serial communication unit registers)

data transmit 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 regsiter mn (SDRmn) (low 8 bit: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 baud rate configuration							0	configuration of transmit data/received data register							

SIOp

(d) serial output register m(SOm) .....Only configure bit of target channel

	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	×	×	×	0/1

(e) serial output enable registerm (SOEm)….only set bit of target channel to 1.

	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

Note Before the master device starts outputting the clock, the SIOp register must be set to send data.

Note 1.m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

2. ☐ : Set in dependent receive mode for fixed. ■: Cannot set (set initial value).

×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).

0/1: The "0" or "1" is set according to the user.

Figure 19-92 A slave selectioninput function (SSPI00) An example of register setting content upon slave transmission and reception (2/2)

(f) serial channel start register m (SSm) …. Only set bit of target channel to 1.

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

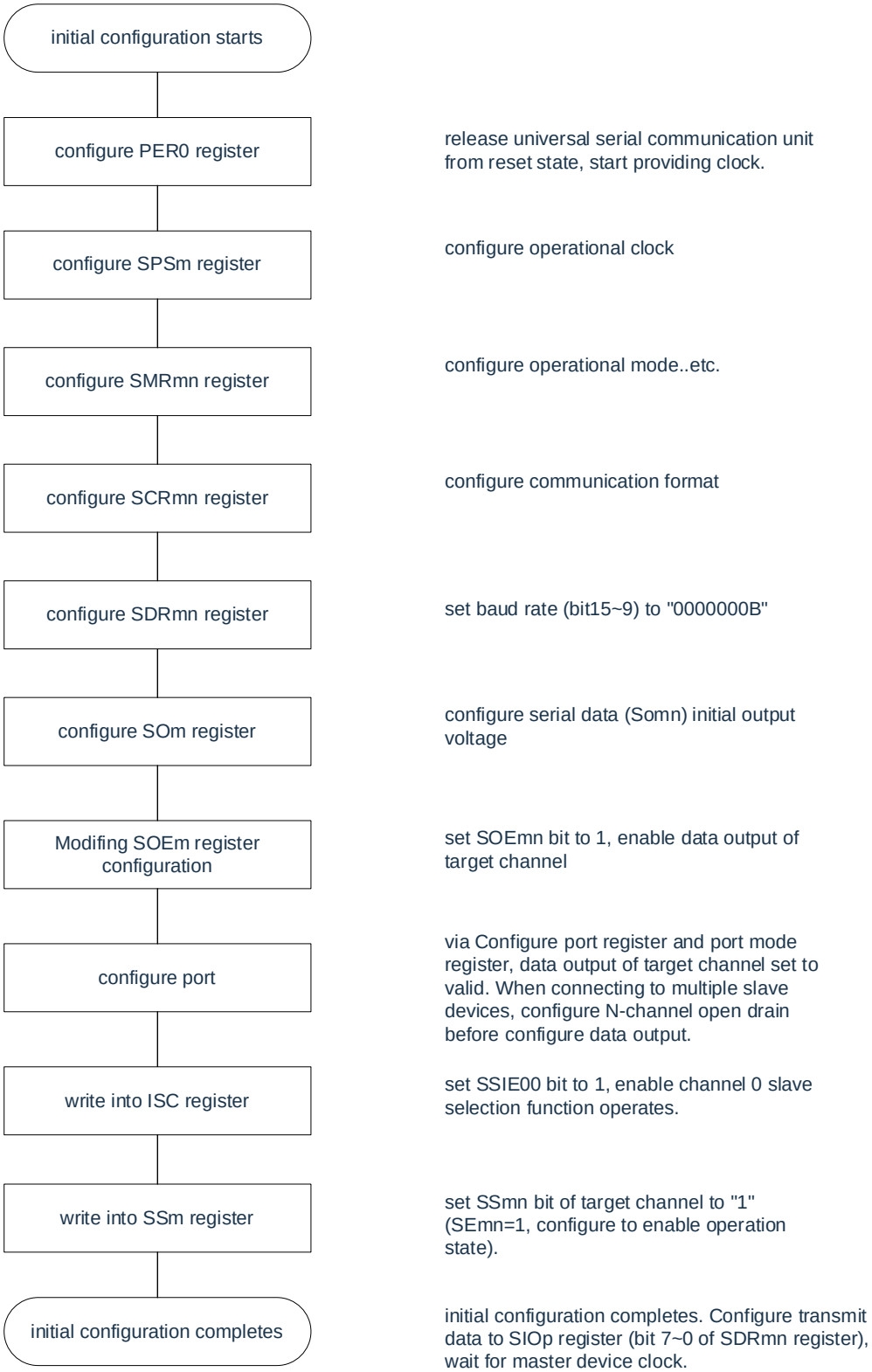
Note Before the master device starts outputting the clock, the SIOp register must be set to send data.

Note 1.m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

2. ☐ : Set in dependent receive mode for fixed. ▒: Cannot set (set initial value).
×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).
0/1: The "0" or "1" is set according to the user.

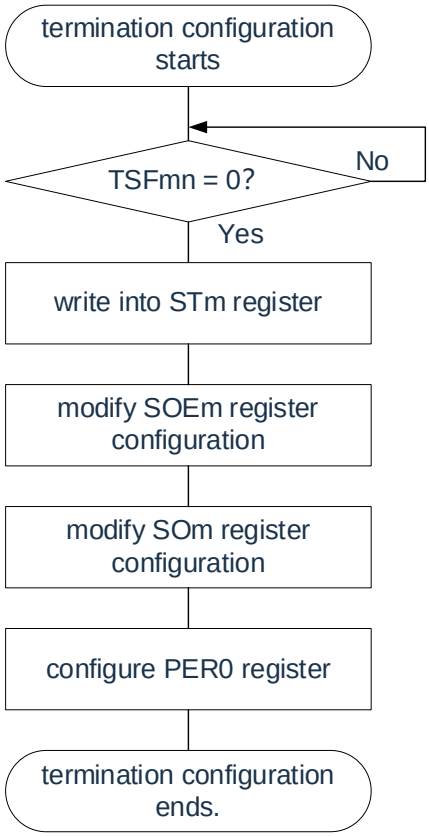
(2) Procedure

Figure 19-93 Initial set-up steps for slave transmission and reception



Note Before the master device starts outputting the clock, the SIOp register must be set to send data. Remark m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

Figure 19-94 Interrupt steps for slave transmission and reception



if there are ongoing data transmission, then wait till transmission completed. (if need urgent stop, then no need to wait).

set STmm bit of target channel to 1. (SEmn=0: set to operation stop state).

set SOEmn bit to 0, stop output of target channel

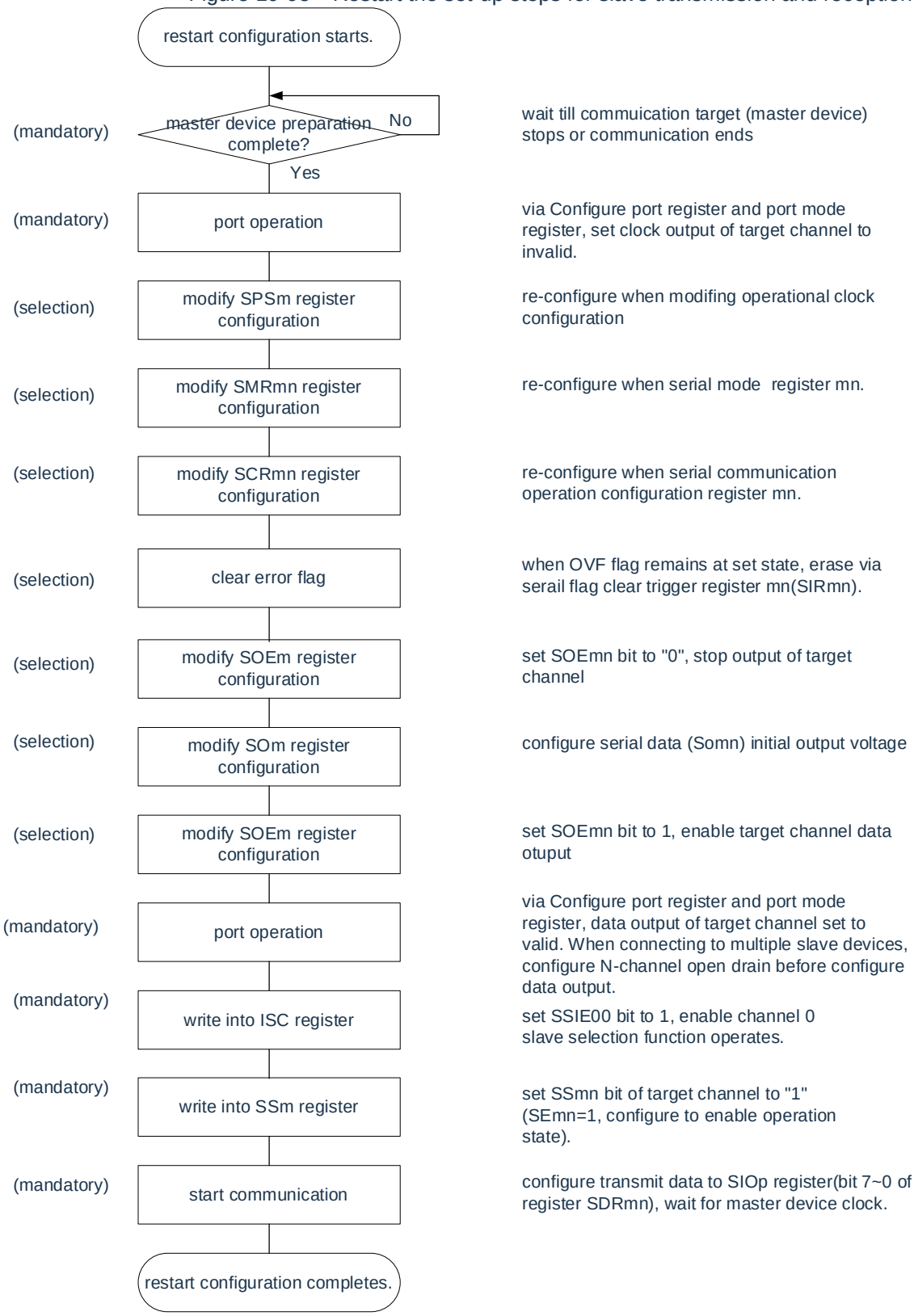
when emergency stop, based on needs, modify serial data (SOmn) voltage level of the target channel.

stop clock of universal serial communication unit, set to reset state

finish termination configuration, enter into next processing.

Note 1.m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

Figure 19-95 Restart the set-up steps for slave transmission and reception

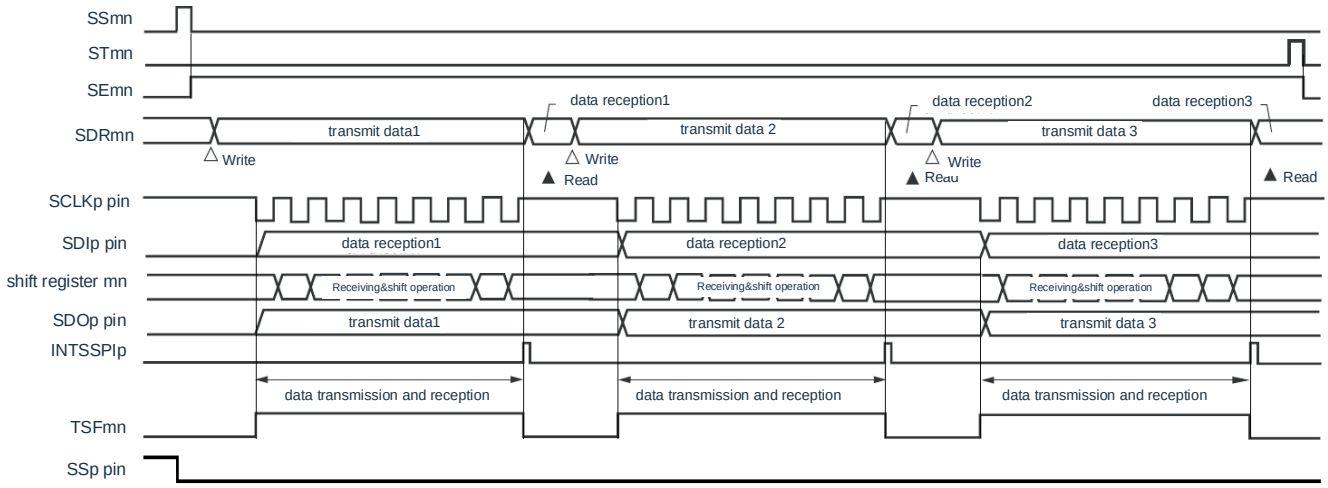


Note 1. The SIOp register setting must be sent before the master device starts outputting the clock.

2. If you override PER0 in the abort setting to stop providing the clock, you must wait until the communication object (master) stops or the communication is over for the initial set-up instead of restarting the set-up.

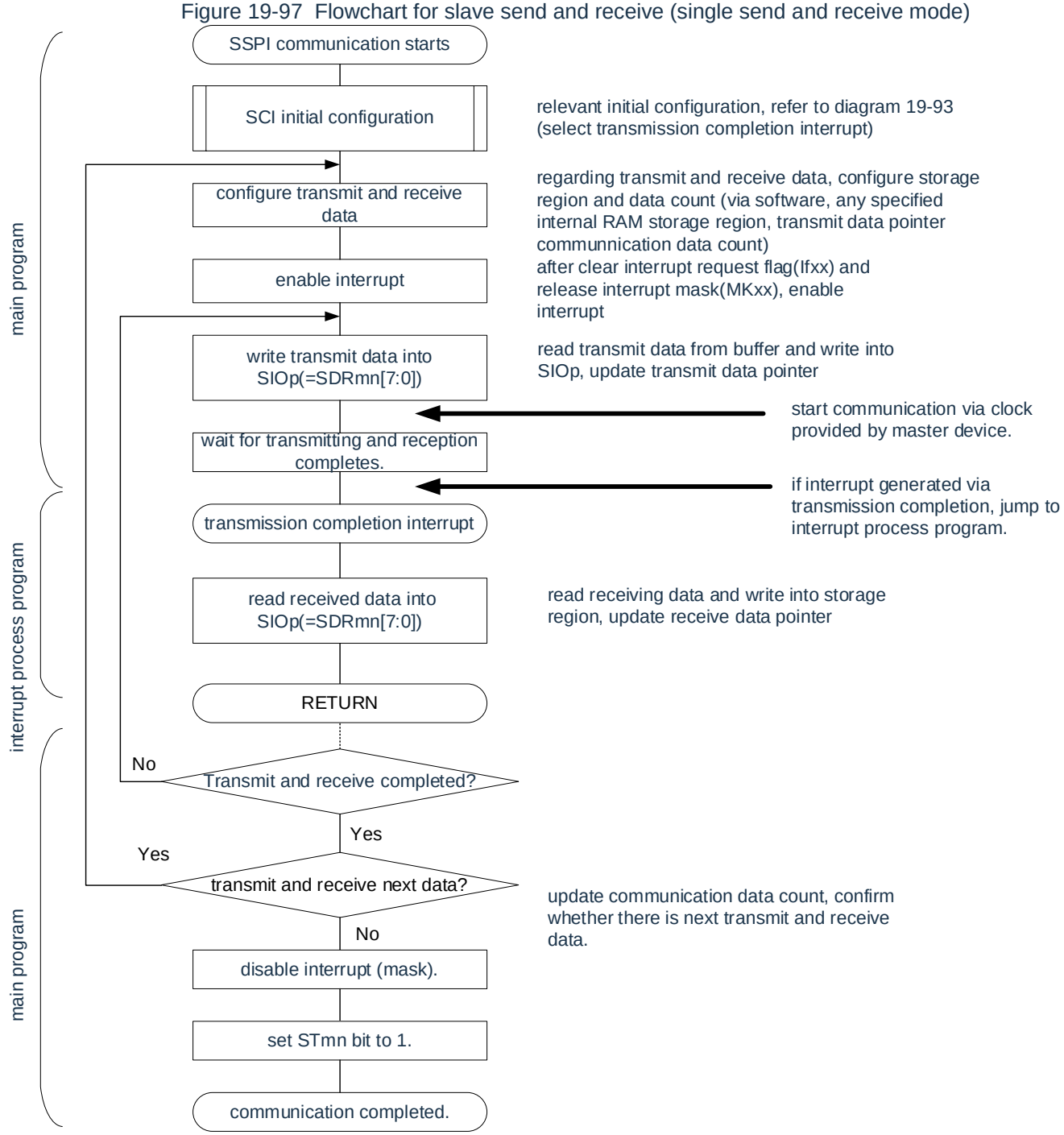
(3) Process flow (single send and receive mode)

Figure 19-96 Time Series for Slave send and Receive (Single Send and Receive Mode) (Type 1: DAPmn=0, CKPmn=0) Figure.



Remark m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

Figure 19-97 Flowchart for slave send and receive (single send and receive mode)

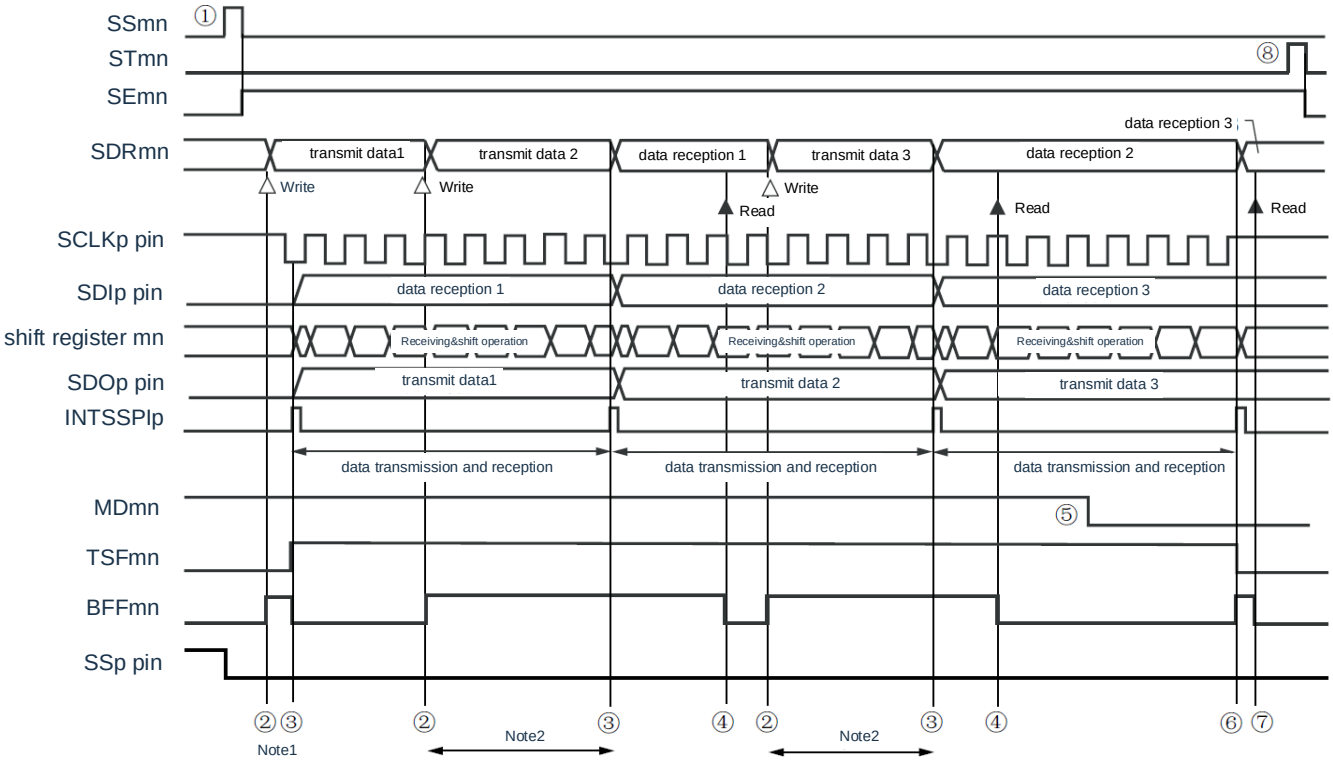


Note Before the master device starts outputting the clock, the SIOp register must be set to send data.

Remark m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

(4) Processing Flow (Continuous Send and Receive Mode)

Figure 19-98 Time Series for Slave send and Receive (Continuous Send and Receive Mode) (Type 1: DAPmn=0, CKPmn=0) Figure.



Note 1. Rewrite the transmission data 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)).

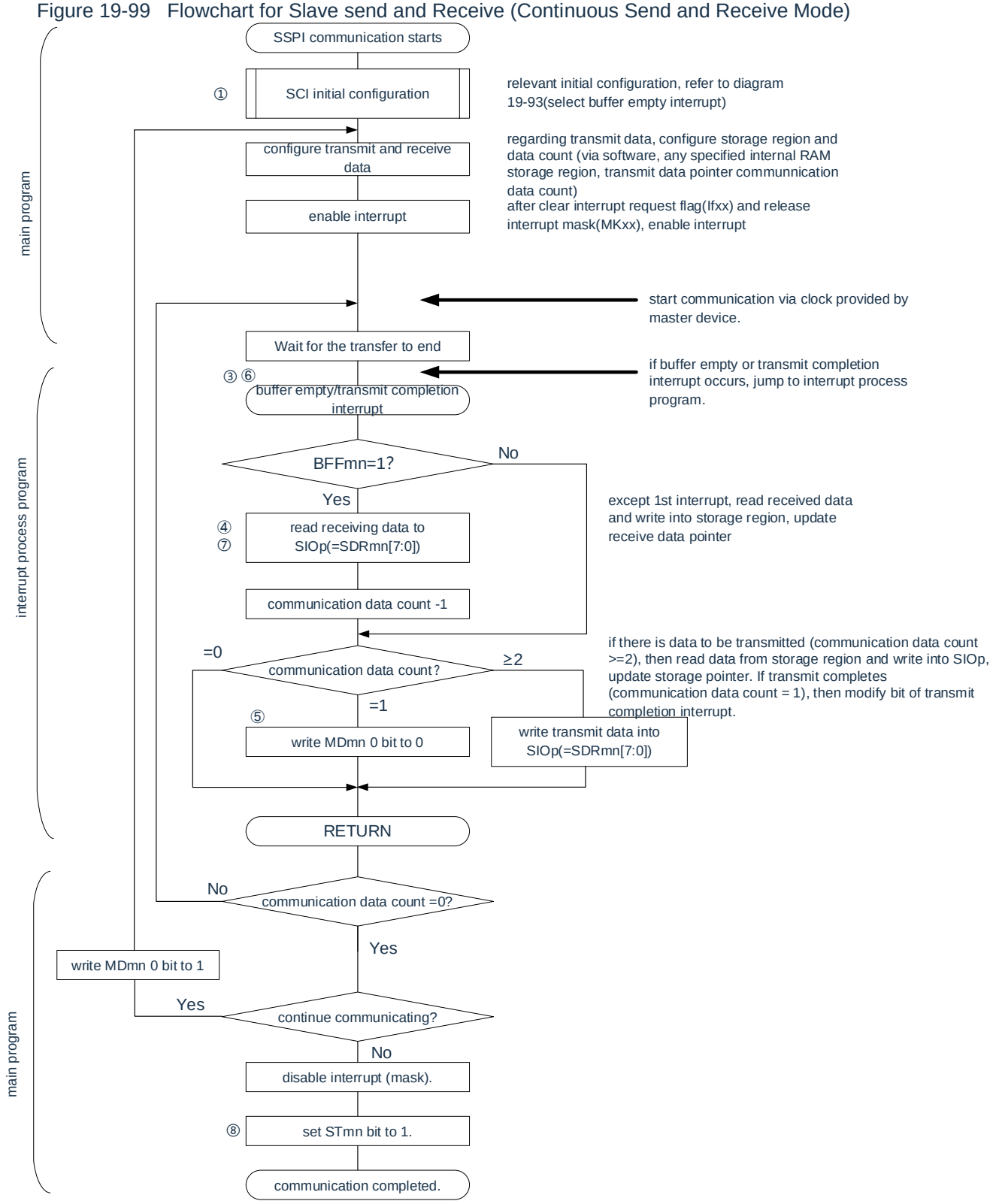
2. Send data can be read if the SDRmn register is read during this period. At this point, the transfer operation is not affected.

Note The MDmn0 bit of the serial mode register mn(SMRmn) can be rewritten even in operation. However, in order to be able to catch the end of the transmission of the last transmitted data interrupt, it is necessary to override before starting the last transmission.

Note 1. (1) to (8) in the figure corresponds to (1) to (8) in the "Flowchart for Slave send and Receive (Continuous Send and Receive Mode) Figure 19-99.

2.m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

Figure 19-99 Flowchart for Slave send and Receive (Continuous Send and Receive Mode)



Note Before the master device starts outputting the clock, the SIOp register must be set to send data.

Note 1. (1) to (8) in the figure corresponds to (1) to (8) in "Figure 19-98Sequential Send and Receive Patterns".

2.m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

19.6.4 Calculation of transmission clock frequency

The transfer clock frequency of the SSPI00 communication can be calculated using the following formula.

(1) slave

(Transfer Clock Frequency)={Serial Clock (SCLK) Frequency provided by the master device} Note ⌈Hz]

Note The maximum allowable transfer clock frequency is fMCK/6.

Remark m: Cell number (m=0)n: Channel Number (n=0)p:SSPI Number (p=00)

Table 19-3 Selection of the running clock of the slave selectioninput function

SMRmn register	SPSm register								Runtime Clock ($_{fMCK}$) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		$_{fCLK}$=32 MHz Runtime
0	X	X	X	X	0	0	0	0	$_{fCLK}$	32MHz
	X	X	X	X	0	0	0	1	$_{fCLK}/2$	16MHz
	X	X	X	X	0	0	1	0	$_{fCLK}/2^2$	8MHz
	X	X	X	X	0	0	1	1	$_{fCLK}/2^3$	4MHz
	X	X	X	X	0	1	0	0	$_{fCLK}/2^4$	2MHz
	X	X	X	X	0	1	0	1	$_{fCLK}/2^5$	1kHz
	X	X	X	X	0	1	1	0	$_{fCLK}/2^6$	500kHz
	X	X	X	X	0	1	1	1	$_{fCLK}/2^7$	250kHz
	X	X	X	X	1	0	0	0	$_{fCLK}/2^8$	125kHz
	X	X	X	X	1	0	0	1	$_{fCLK}/2^9$	62.5kHz
	X	X	X	X	1	0	1	0	$_{fCLK}/2^{10}$	31.25kHz
	X	X	X	X	1	0	1	1	$_{fCLK}/2^{11}$	15.63kHz
	X	X	X	X	1	1	0	0	$_{fCLK}/2^{12}$	7.81kHz
	X	X	X	X	1	1	0	1	$_{fCLK}/2^{13}$	3.91kHz
	X	X	X	X	1	1	1	0	$_{fCLK}/2^{14}$	1.95kHz
	X	X	X	X	1	1	1	1	$_{fCLK}/2^{15}$	977Hz

Note To change the clock selected as fCLK (change the value of the System Clock Control Register (CKC), you must change after stopping Universal Serial Communication Unit (SCI) =000FH.

Note 1.X: Ignore

2.m: Cell number (m=0)n: Channel number (n=0)

19.6.5 Processing steps when an error occurs during clock synchronization serial communication of a slave selection input function

The processing steps when an error occurs during clock synchronization serial communication of a slave selection input function are shown in Figure 19-100.

Figure 19-100 Handling steps when overflow errors occur

software operation	Hardware Status	Comments
Read the serial data register mn (SDRmn) →	The BFFmn bit of the SSRmn register is "0" and the channel n is in a receiver state.	This is to prevent an overflow error from occurring to end the next receipt during error handling. next receipt during error handling.
Read the serial status register mn (SSRmn).		The type of error is determined and the read value is used to clear the error flag.
Clear trigger register mn for serial flag (SDIRmn) Write "1". →	Clear the error flag.	By writing the read value of the SSRmn register directly to the SDIRmn register, errors in the read operation can only be cleared.

Remark m: Cell number (m=0)n: Channel number (n=0)

19.7 Operation of UART(UART0~UART2) Communication

This is the ability to communicate asynchronously over a total of two lines, Serial Data Send (TxD) and Serial Data Receive (RxD). The two communication lines are used to transmit and receive data asynchronously (using internal baud rate) with other communication parties in data frames (consisting of start bit, data, parity bit and stop bit). The invention can realize full-duplex asynchronous UART communication by using two channels, namely sending (even channel) and receiving (INTP0).

[Sending and Receiving Data]
· Data length notes for 7, 8, or 9 bits
• MSB/LSB First Choice
· Level setting for transmitting and receiving data (select whether level is inverted)
· Additional, parity functions for parity bits
· Additional Stop Bit and Detection Function of Stop Bit
[Interrupt Capabilities]
· interrupt delivery end, buffer interrupt
· Error interrupts due to frame errors, parity errors, and overflow errors
[Error Detection Flag]
· frame error, parity error, overflow error

UART0 (Channel 0 and Channel 1 of Unit 0)

supports LIN-bus. [LIN-bus Features]

· Detection of wake-up signal
· Detection of Interval Segments (BF) } Use external interrupts (INTP0) and Timer4 (Channel 3).
· Measurement of Synchronous Section and Calculation of Baud Rate

Note Only UART0 supports 9-bit data length.

UART0 uses SCI0's channel 0 and channel 1.
UART1 uses SCI0's channel 2 and channel 3.
UART2 uses SCI1's channel 0 and channel 1.

Each channel selects one of the functions to use, and other functions cannot run except the selected function.

For example, you cannot use SSPI00 and IIC01 when UART0 is used in Channel 0 and Channel 1 of Cell. However, while using UART0, channels 2 and 3 for different channels can use SSPI10, UART1, or IIC10.

Note When used as a UART, the sender (even channel) and receiver (odd channel) can only be used for UART.

UART has the following four communications runs:
· UART Send (Reference 19.7.1)
· UART Receive (Reference 19.7.2)
· LIN Send (UART0 only) (Reference 19.8.1)
· LIN Receive (UART0 only) (Reference 19.8.2)

19.7.1 UART Send

UART Send is an operation where the BAT32A237 microcontroller asynchronously sends data to other devices.

The even of the 2 channels used by UART are for UART transmission.

UART	UART0	UART1	UART2
object channel	Channel 0 for SCI0	Channel 2 for SCI0	Channel 0 for SCI1
Pin Used	TxD0	TxD1	TxD2
interrupt	INTST0	INTST1	INTST2
	Interrupt at that end of the transfer may be selecte (single transfer mode) or buffer air-discontinuity (continuous transfer mode).		
Error detection flag	None		
length of transmit data	7-bit, 8-bit, or 9-bit Note 1		
transfer rate	Max.fMCK/6[bps] (SDRmn[15:9]≥2), Min.fCLK/(2 215 128) [bps] Note2		
data phase	Positive output (default: High level). Invert output (default: Low level).		
parity bit	You can select the following: · No parity bits. · Additional zero check. · Additional even check. · Additional odd check.		
stop bit	You can select the following: · One additional digits. · Two additional digits.		
data orientation	MSB First or LSB First		

Note1. Only UART0 supports 9-bit data length.

2. Must be used within the scope of peripheral functional characteristics (reference data manual) that satisfy this condition and electrical characteristics.

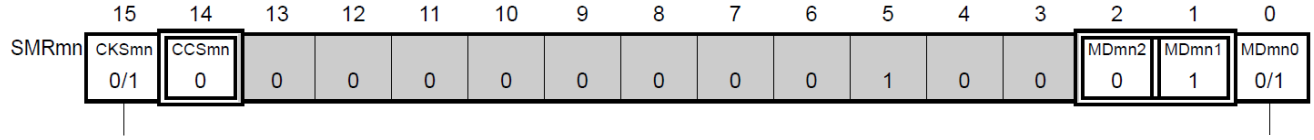
Note 1.fMCK: Object channel running clock frequency fCLK: system clock frequency

2.m: Cell number (m=0,1)n: Channel Number (n=0,2) mn=00,02,10

(1) Register settings

Figure 19-101 Example of register setting content (1/2) when UART of UART (UART0~UART2) is sent

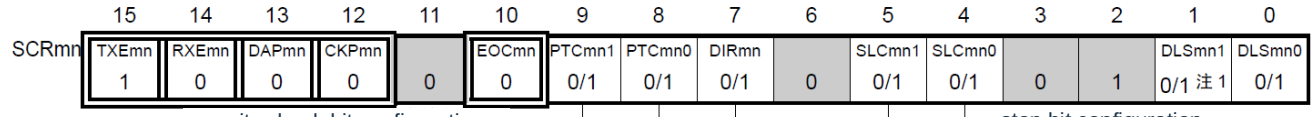
(a) serial mode register mn (SMRmn)



channel n operational clock (fMCK)
0: SPSm register configured pre-scaler output clock CKm0
1: SPSm register configured pre-scaler output clock CKm1

channel n interrupt source
0: Transmit completion interrupt
1: Buffer empty interrupt

(b) serial communication operation configuration register mn (SCRmn)

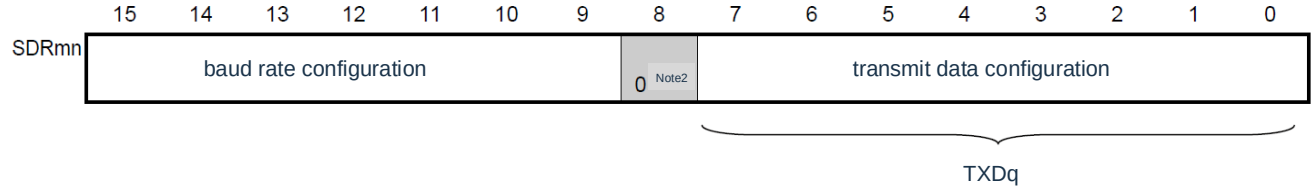


parity check bit configuration
00B: no parity check
01B: add zero parity
10B: add even parity
11B: add odd parity

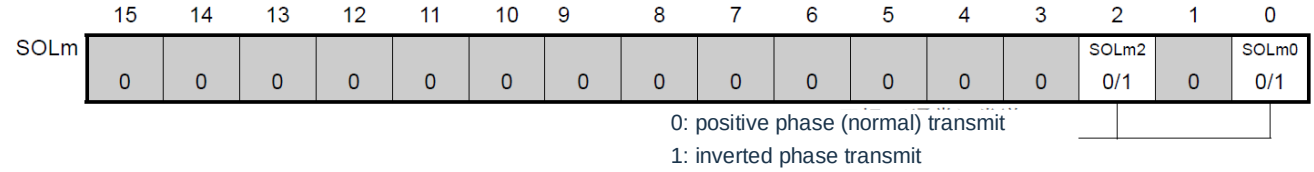
data transmit sequence selection
0: perform MSB first input/output
1: perform LSB first input/output

stop bit configuration
01B: add 1 bit
10B: add 2 bits

(c) serial data regsiter mn (SDRmn) (low 8 bit:TXDq)



(d) serial output voltage register m (SOLm) …. Only configure bit of target channel.



0: positive phase (normal) transmit
1: inverted phase transmit

Note1. Limited to SCR00 registers, other fixed to "1".

2. When 9-bit data length communication is carried out, bit0~8 of the SDRm0 register is set area. Only UART0 can communicate for 9-bit data length.

Note 1.m: Cell number (m=0,1)n: Channel Number (n=0,2)q:UART Number (q=0~2)mn=00,02,10

2.□ : Fixed in UART send mode. ■ : Cannot set (set initial value).

×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).

0/1: The "0" or "1" is set according to the user.

Figure 19-101 Example of register setting content (2/2) when UART of UART (UART0~UART2) is sent

(e) serial output register m (SOm)…. Only configure bit of target channel

	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 0/1 Note	SOm1 ×	SOm0 0/1 Note

0: serial data output value as "0"
1: serial data output value as "1"

(f) serial output enable register m (SOEm)…. Only set bit of 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 0/1	SOEm1 ×	SOEm0 0/1

(g) serial channel start register m (SSm) …. Only set bit of target channel to "1".

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Note Before starting sending, '1' must be set when the SOLmn bit of the corresponding channel is '0'; When the SOLmn bit of the corresponding channel is '1', '0' must be set. During communication, the value changes due to communication data.

Note 1.m: Cell number (m=0,1)n: Channel Number (n=0,2)q:UART Number (q=0~2)mn=00,02,10

2.□ : Fixed in UART send mode. ■ : Cannot set (set initial value).

×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).

0/1: The "0" or "1" is set according to the user.

(2) Procedure

Figure 19-102 Initial set-up steps sent by UART

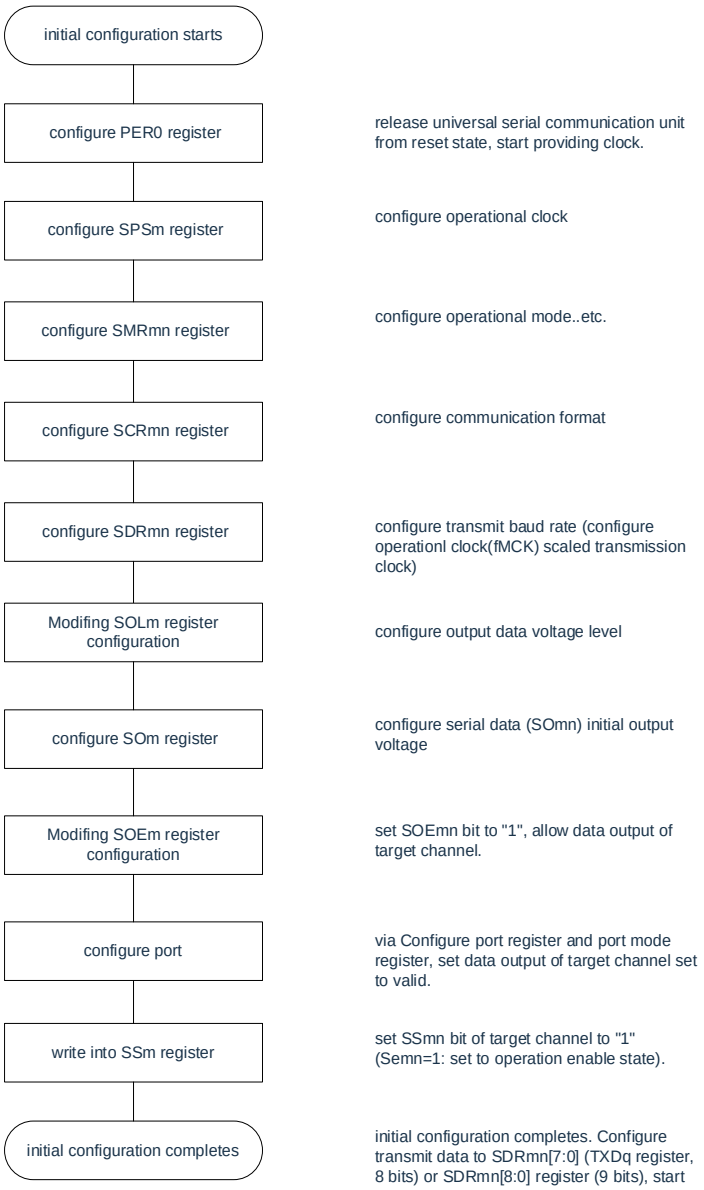


Figure 19-103 Abort Steps for UART Send

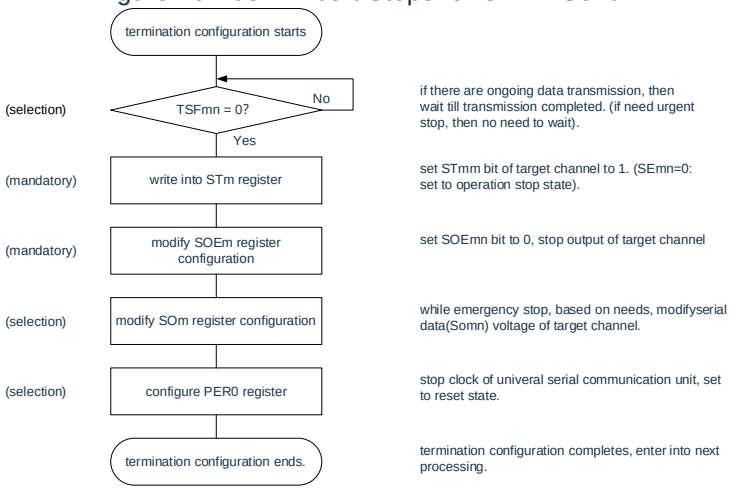
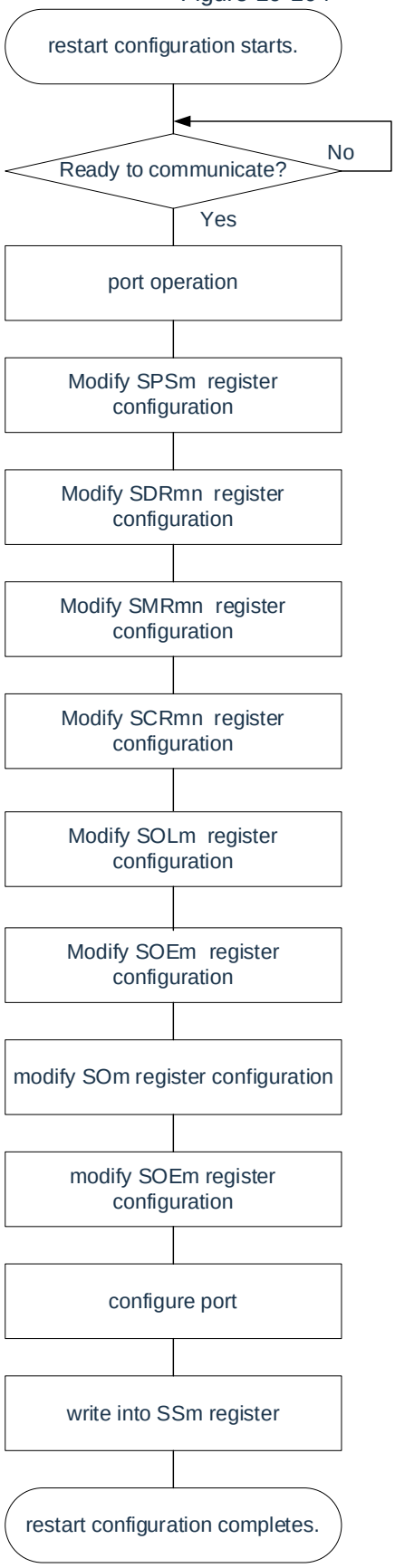


Figure 19-104 Restart set-up steps sent by UART

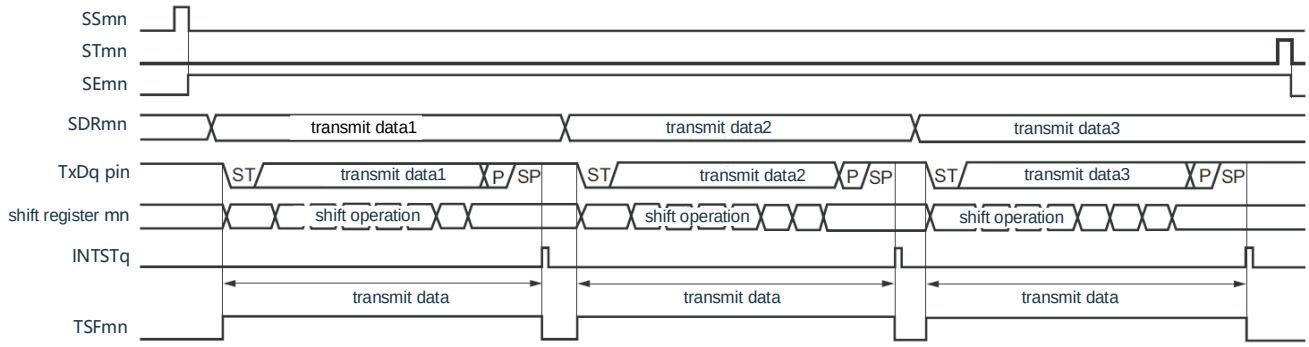


(mandatory)	Ready to communicate?	wait till commuication target (slave device) stops or communication ends
(mandatory)	port operation	The data output of the target channel is disabled by setting the port register and port mode register.
(selection)	Modify SPSm register configuration	re-configure when modifying the operational clock configuration.
(selection)	Modify SDRmn register configuration	re-configure when modifying the transmit baud rate configuration.
(selection)	Modify SMRmn register configuration	re-configure when modifying serail mode register mn configuration.
(selection)	Modify SCRmn register configuration	re-configure when modifying serial communcation operation configuration register mn.
(selection)	Modify SOLm register configuration	re-configure when modifying serail output voltage regsiter m
(selection)	Modify SOEm register configuration	set SOEmn bit to 0, stop output of target channel
(selection)	modify SOm register configuration	configure serial data(SOmn) initial output voltage
(selection)	modify SOEm register configuration	set SOEmn bit to 1, allow target channel data output
(mandatory)	configure port	configure port register and port mode register, set target channel data output to be valid.
(mandatory)	write into SSm register	set SSmn bit of target channel to 1 (Semn=1: set as operation enable state)
(mandatory)	restart configuration completes.	configuration ends. If configures transmission data into SIOp register (bit 7~0 of SDRmn register), then start communication.

Remark If that PER0 is override in the abort setting to stop the supply clock, the initial setting must wait until the communication object is stopped or communication is complete, rather than reset.

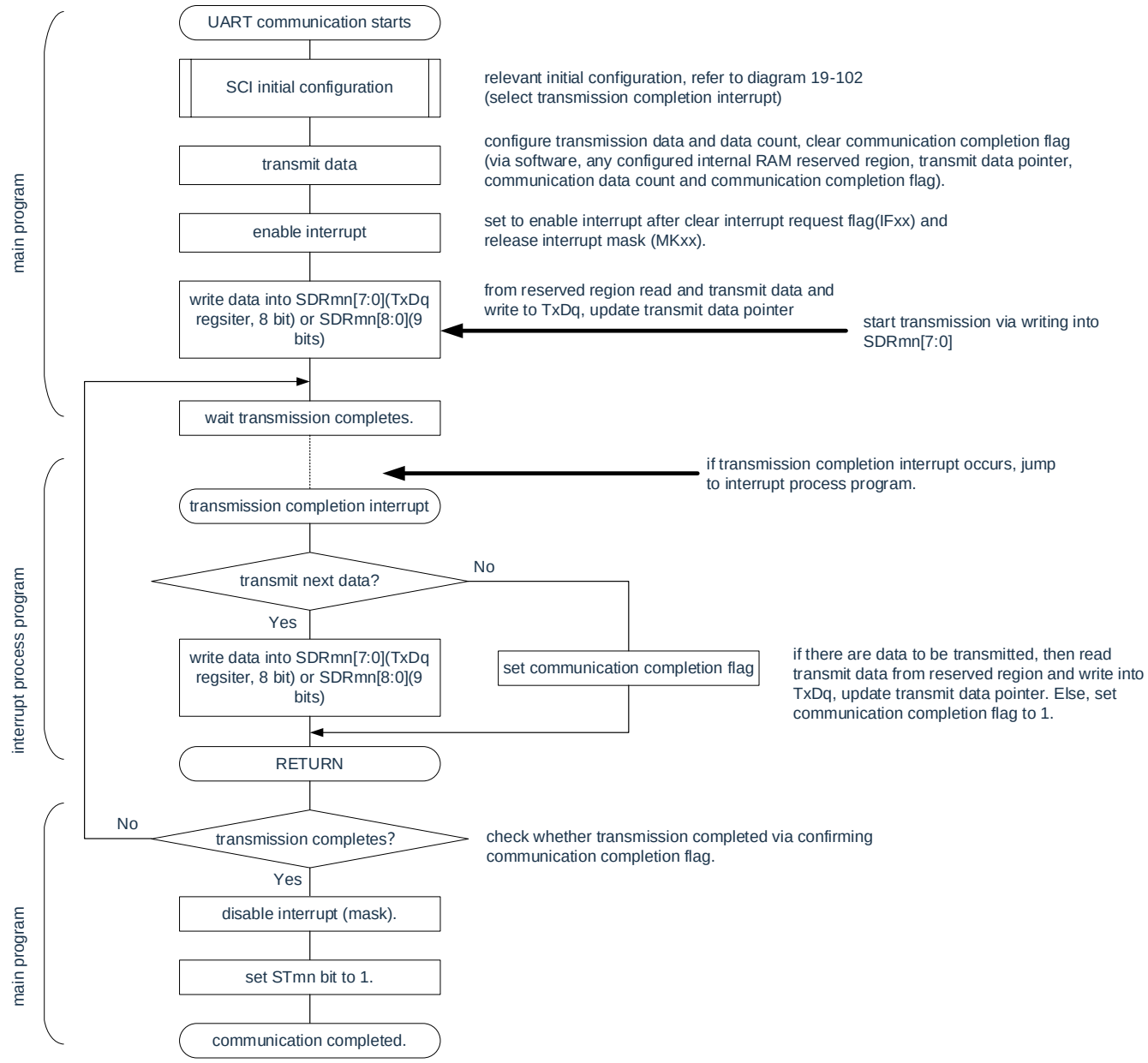
(3) Process flow (single send mode)

Figure 19-105 Time Series for UART Send (single send mode)



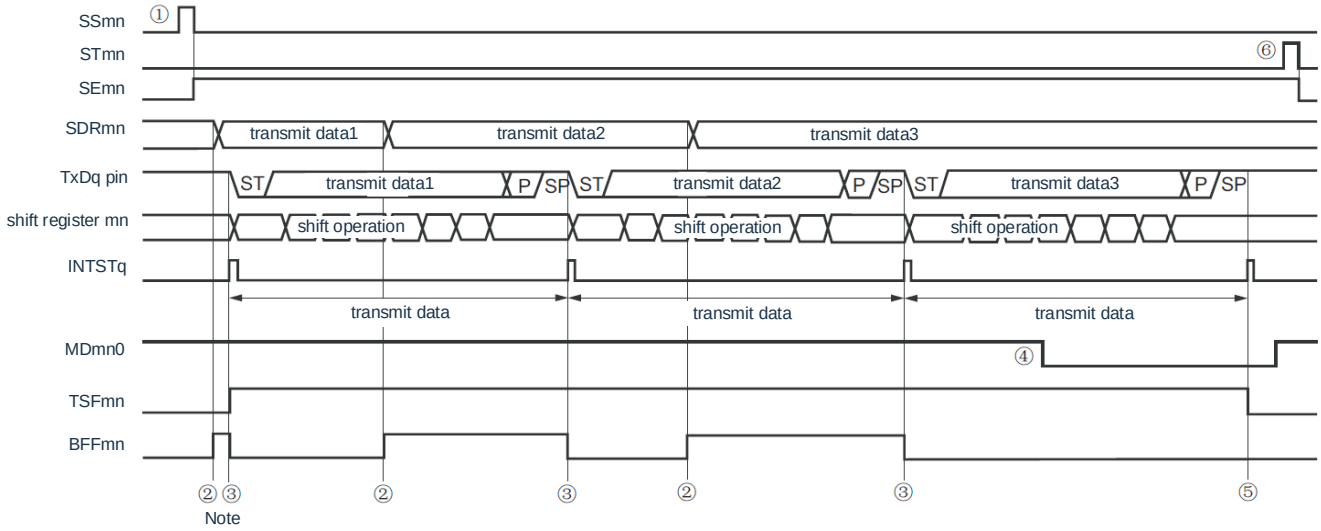
Remark m: Cell number (m=0,1)n: Channel Number (n=0,2)q:UART Number (q=0~2)mn=00,02,10

Figure 19-106 Flowchart for UART Send (single send mode)



(4) Process flow (continuous send mode)

Figure 19-107 Time Series for UART Send (Continuous Send Mode)

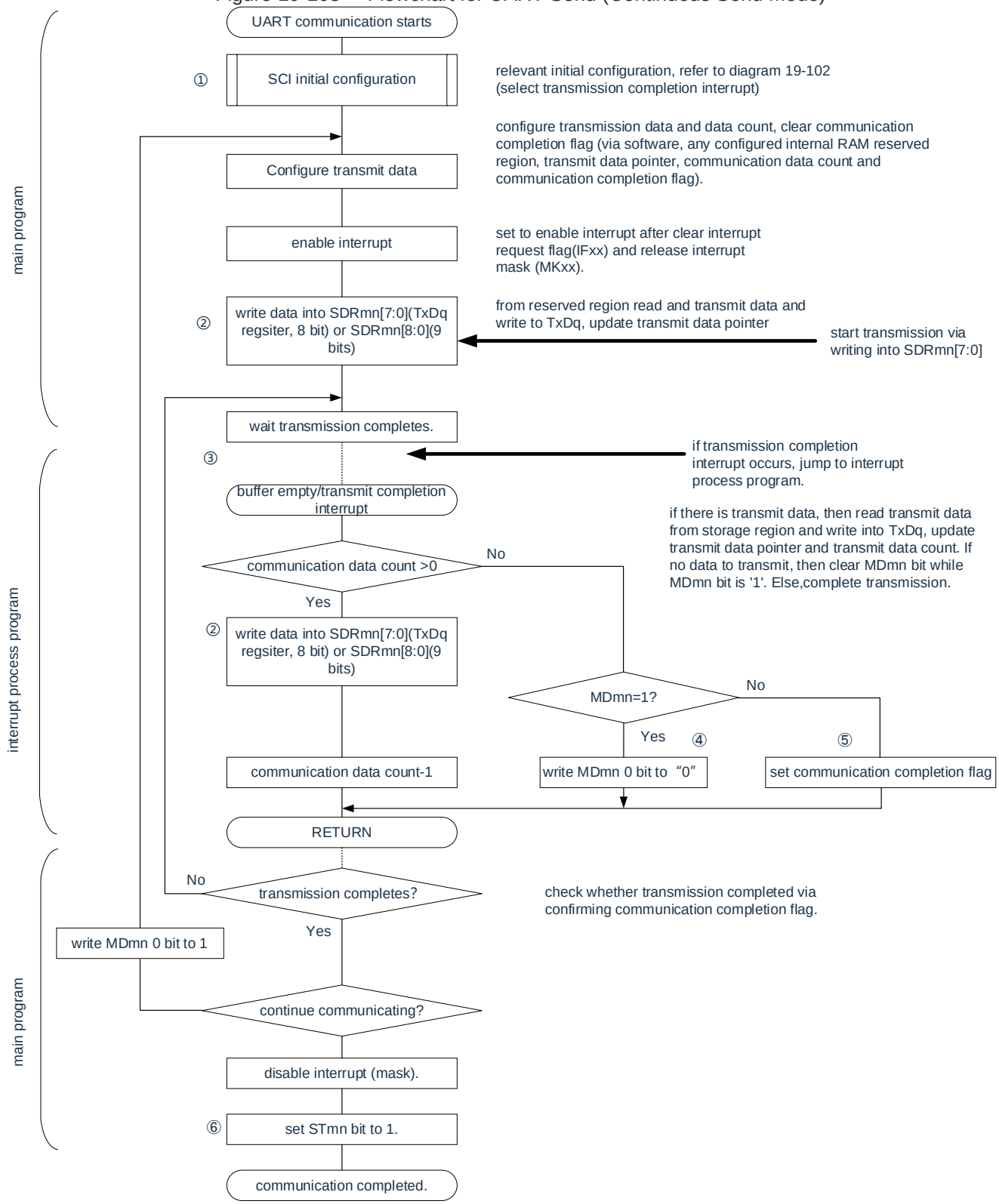


Note The transmission data is rewritten if the BFFmn bit of the serial status register mn (SSRmn) is "1" (SDRmn) when the valid data is stored in the serial data register mn (SDRmn).

Note The MDmn0 bit of the serial mode register mn(SMRmn) can be rewritten even in operation. However, in order to be able to catch the end of the transmission of the last transmitted data interrupt, it is necessary to override before starting the last transmission.

Remark m: Cell number (m=0,1)n: Channel Number (n=0,2)q:UART Number (q=0~2)mn=00,02,10

Figure 19-108 Flowchart for UART Send (Continuous Send Mode)



Remark (1) to (6) in the figure corresponds to (1) to (6) in the figure 19-107 UART.

19.7.2 UART Receive

UART Receive is an operation in which other devices of the BAT32A237 microcontroller receive data asynchronously.

The odd number of the 2 channels used by UART is used for UART receiving. However, you need to set up SMR. registers for odd and even channels.

UART	UART0	UART1	UART2
object channel	Channel 1 for SCI0	Channel 3 for SCI0	Channel 1 for SCI1
Pin Used	RxD0	RxD1	RxD2
interrupt	INTSR0	INTSR1	INTSR2
	Interrupt at that end of the transfer only (Disable from setting buffer air interrupt).		
error interrupt	INTSRE0	INTSRE1	INTSRE2
Error detection flag	· Frame Error Detection Flag (FEFmn) · Parity error detection flag (PEFmn) · Overflow Error Detection Flag (OVFmn)		
length of transmit data	7-bit, 8-bit, or 9-bit Note 1		
transfer rate	Max.$_{fMCK}$/6[bps] (SDRmn[15:9]≥2), Min.$_{fCLK}$/(2 2^{15} 128) [bps]		
data phase	Positive output (default: High level). Invert output (default: Low level).		
parity bit	You can select the following: · No parity bits (no parity). · Additional zero-check (no parity). · Even check · Odd check		
stop bit	Additional 1-bit.		
data orientation	MSB First or LSB First		

Note1. Only UART0 supports 9-bit data length.

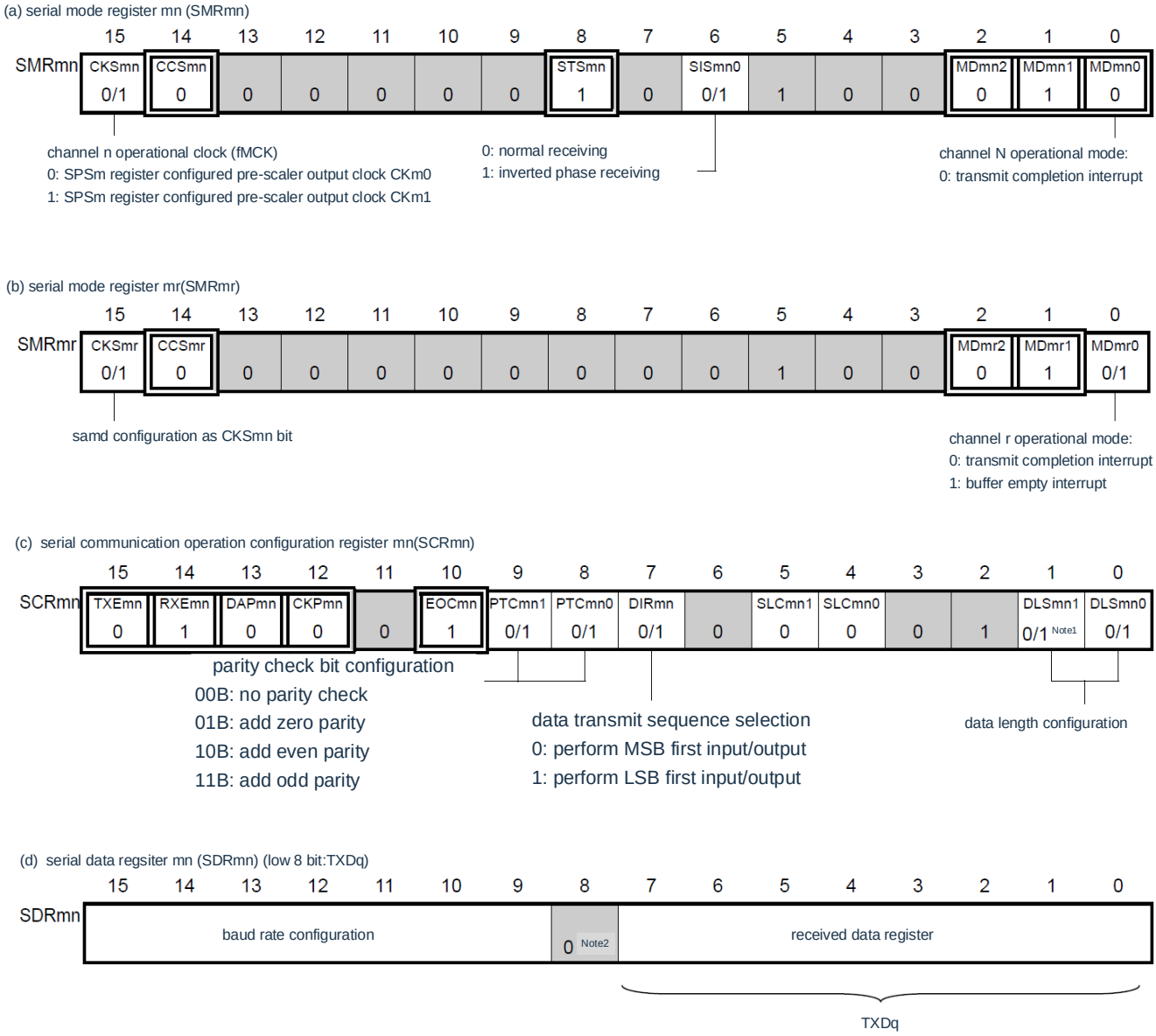
2. Must be used within the scope of peripheral functional characteristics (reference data manual) that satisfy this condition and electrical characteristics.

Note 1.fMCK: Object channel running clock
frequency fCLK: system clock
frequency

2.m: Cell number (m=0,1)n: Channel Number (n=1,3) mn=01,03,11

(1) Register settings

Figure 19-109 Example of register setting content (1/2) for UART reception of UART (UART0~UART2)



Note1. Limited to SCR01 registers, other fixed to "1".

2. When 9-bit data length communication is carried out, bit0~8 of the SDRm1 register is set area. Only UART0 can communicate for 9-bit data length.

Note When UART receives, you must also set the SMRmr register for channel r

that is paired with channel n.

Note 1.m: Cell 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 set (set initial value).

×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).

0/1: The "0" or "1" is set according to the user.

Figure 19-109 Example of register setting content (2/2) for UART reception of UART (UART0~UART2)

(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 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	×	0/1	×

Note 1.m: Cell Number (m=0,1)

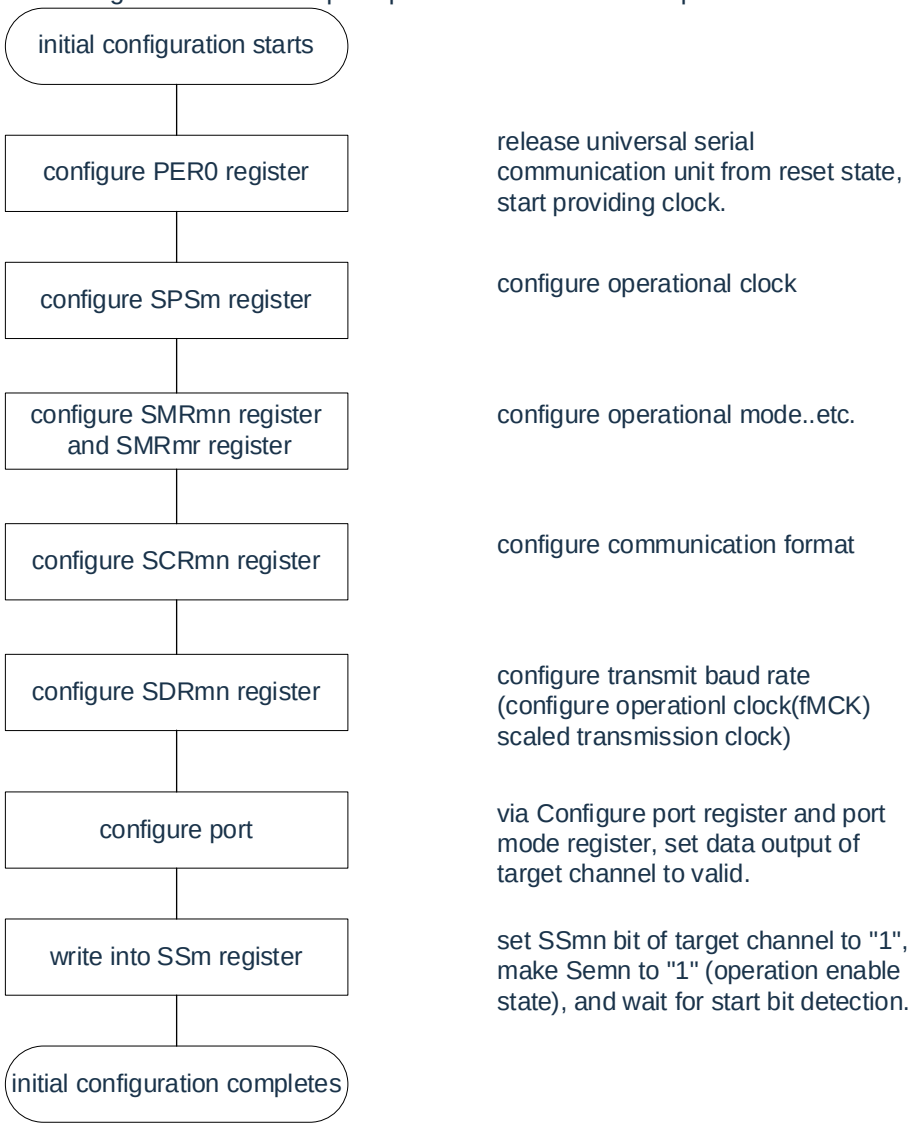
2. ☐ : Fixed in UART receive mode. ▒ : Cannot set (set initial value).

×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).

0/1: The "0" or "1" is set according to the user.

(2) Procedure

Figure 1 Initial Set-up Steps for 9-110 UART Reception



Note At least 4 fMCK clocks must be separated after "1" the RXEmn location of the SCRmn register, and then the SSmn location "1.

Figure 19-111 Termination Steps for UART Receipt

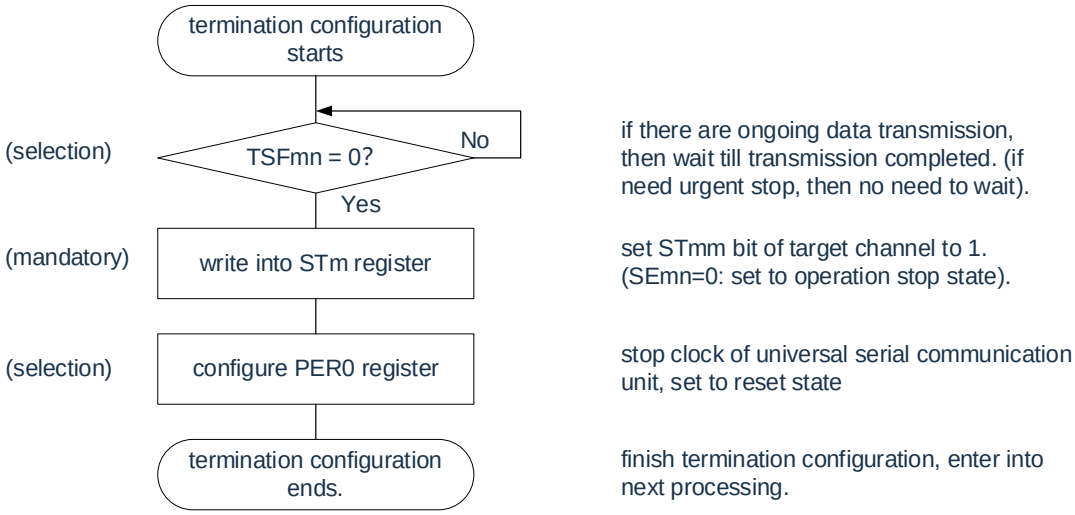
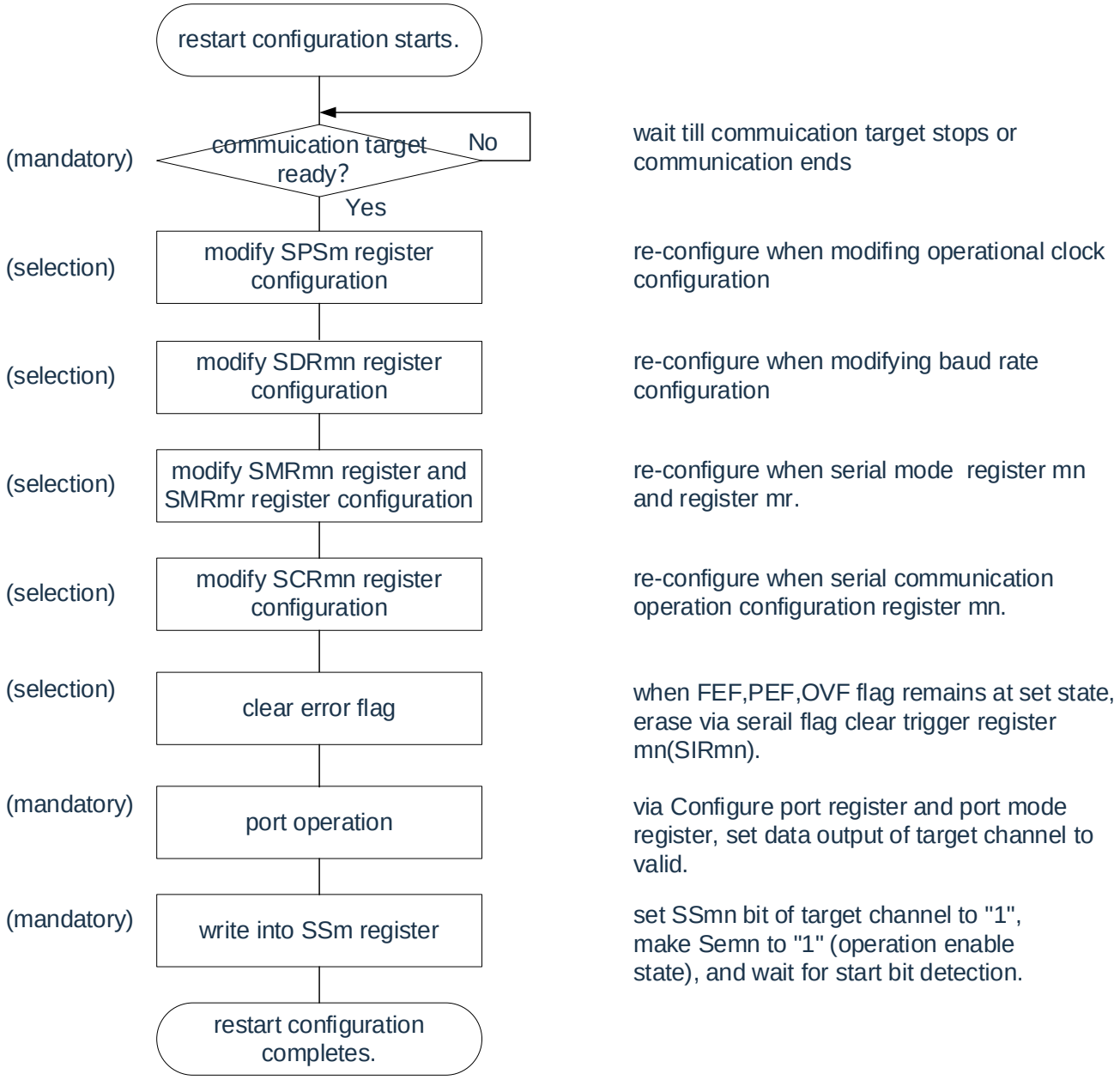


Figure 19-112 To Restart the Set-up Steps for UART Receipts

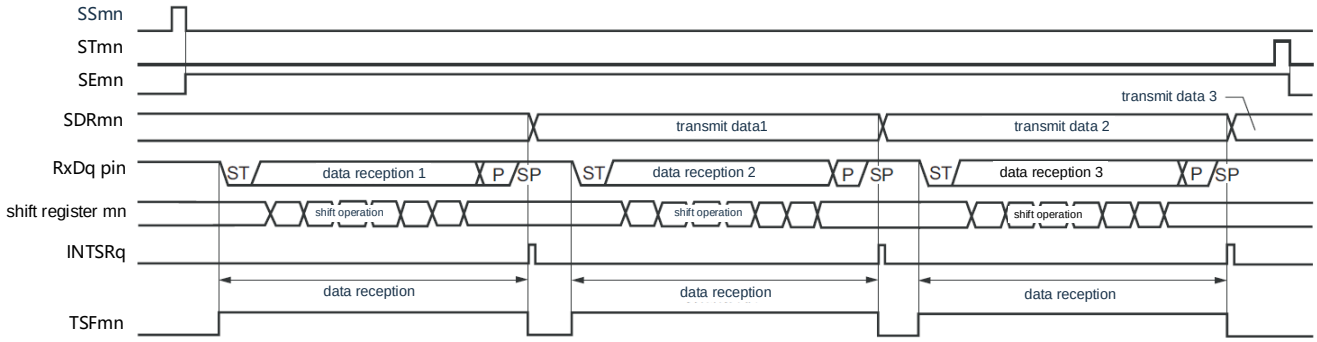


Note At least 4 fMCK clocks must be separated after "1" the RXEmn location of the SCRmn register, and then the SSmn location "1.

Remark If that PER0 is override in the abort setting to stop the supply clock, the initial setting must wait until the communication object is stopped or communication is complete, rather than reset.

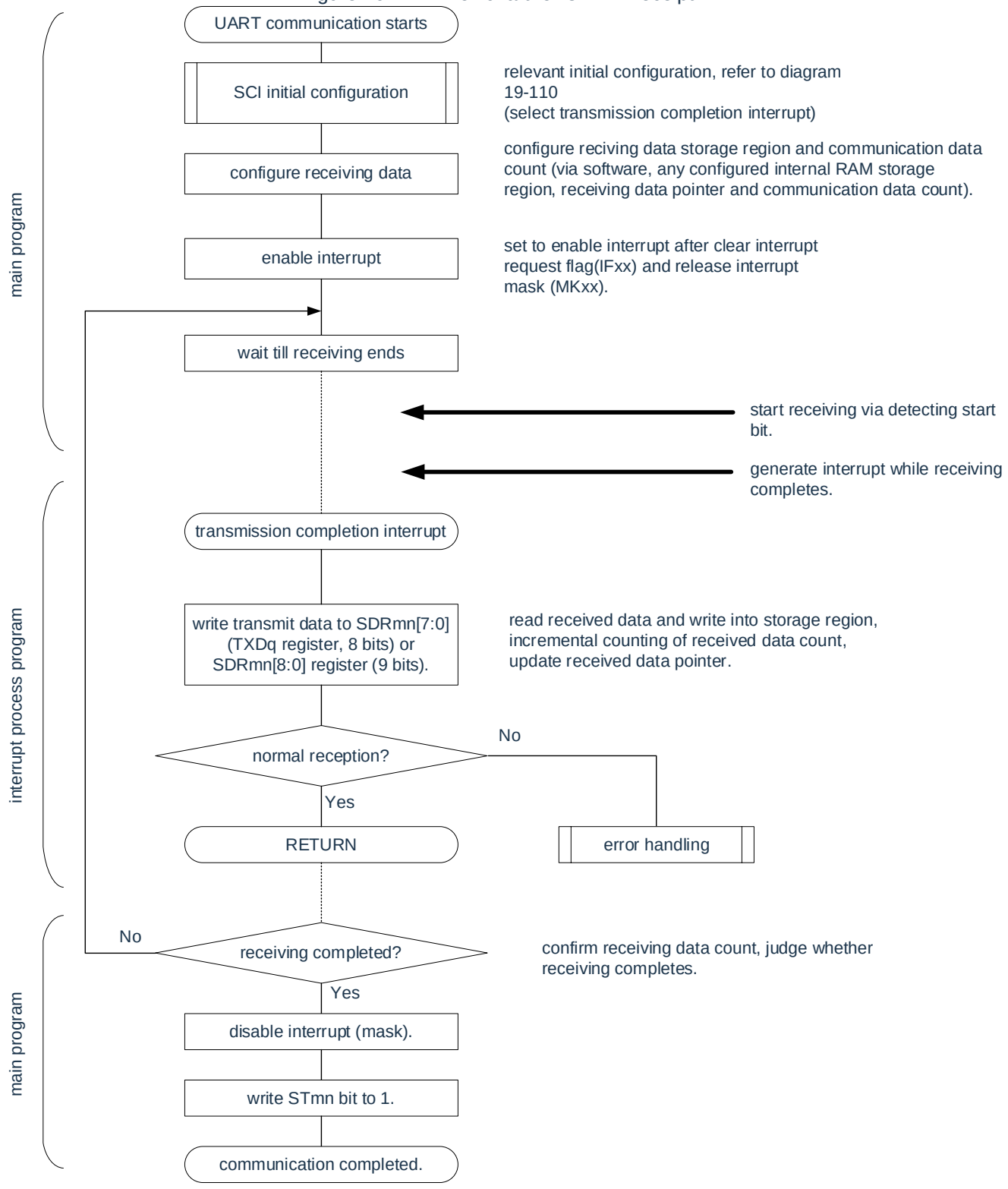
(3) process flow

Figure 19-113 Time Series Received by UART



Remark m: Cell 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 19-114 Flowchart for UART Receipt



19.7.3 Calculation of Baud Rate

(1) Formula for Baud Rate

The baud rate of UART(UART0~UART2) communication can be calculated by the following formula:

(baud rate)={Runtime clock (f_{MCK}) frequency} (SDRmn [15:9]+1) 2[bps]

Note Disable from setting SDRmn[15:9] of serial data register mn(SDRmn) to '0000000B' and '0000001B.

Note 1. Because the value of SDRmn[15:9] is bit15~9 of the SDRmn register (0000010B~1111111B) when using UART].

2.m: Cell number (m=0,1)n: Channel Number (n=0~2) mn=00~03,10~11.

The runtime clock (fMCK) depends on the bit15 (CKSmn bit) of the serial clock selection register m (SPSm) and the serial mode register mn (SMRmn).

Table 19-5. The Choice of UART Running Clock

SMRmn register	SPSm register								Runtime Clock (f_{MCK}) Note	
CKSmn	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00		f_{CLK}=32 MHz Runtime
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 fCLK (change the value of the System Clock Control Register (CKC), you must change after stopping Universal Serial Communication Unit (SCI) =000FH.

Note 1.X: Ignore
2.m: Cell number (m=0,1)n: Channel Number (n=0~2) mn=00~03,10~11.

(2) Baud Rate Error at Send

The baud rate error of UART (UART0~UART2) communication transmission can be calculated by the following formula, the baud rate of the sender must be set within the acceptable range of the baud rate of the receiver.

```
(baud rate error)=(calculated value of baud rate) ÷ (value of target baud rate) ×100-100[%]
```

An example of setting the UART baud rate at fCLK=32 MHz is shown below.

UART baud rate (target baud rate)	f_{CLK}=32MHz			
	Runtime Clock (f_{MCK})	SDRmn [15:9]	Calculated value of baud rate	Error with target baud rate
300bps	$f_{CLK}/2^9$	103	300.48bps	+0.16%
600bps	$f_{CLK}/2^8$	103	600.96bps	+0.16%
1200bps	$f_{CLK}/2^7$	103	1201.92bps	+0.16%
2400bps	$f_{CLK}/2^6$	103	2403.85bps	+0.16%
4800bps	$f_{CLK}/2^5$	103	4807.69bps	+0.16%
9600bps	$f_{CLK}/2^4$	103	9615.38bps	+0.16%
19200bps	$f_{CLK}/2^3$	103	19230.8bps	+0.16%
31250bps	$f_{CLK}/2^3$	63	31250.0bps	±0.0%
38400bps	$f_{CLK}/2^2$	103	38461.5bps	+0.16%
76800bps	$f_{CLK}/2$	103	76923.1bps	+0.16%
153600bps	f_{CLK}	103	153846bps	+0.16%
312500bps	f_{CLK}	50	313725bps	±0.39%

Remark m: Cell number (m=0,1)n: Channel Number (n=0,2) mn=00,02,10

(3) Acceptable range of baud rate at reception

The baud tolerance of UART (UART0~UART2) communication when receiving can be calculated by the following formula, the baud rate of the sender must be set within the baud tolerance of the receiver.

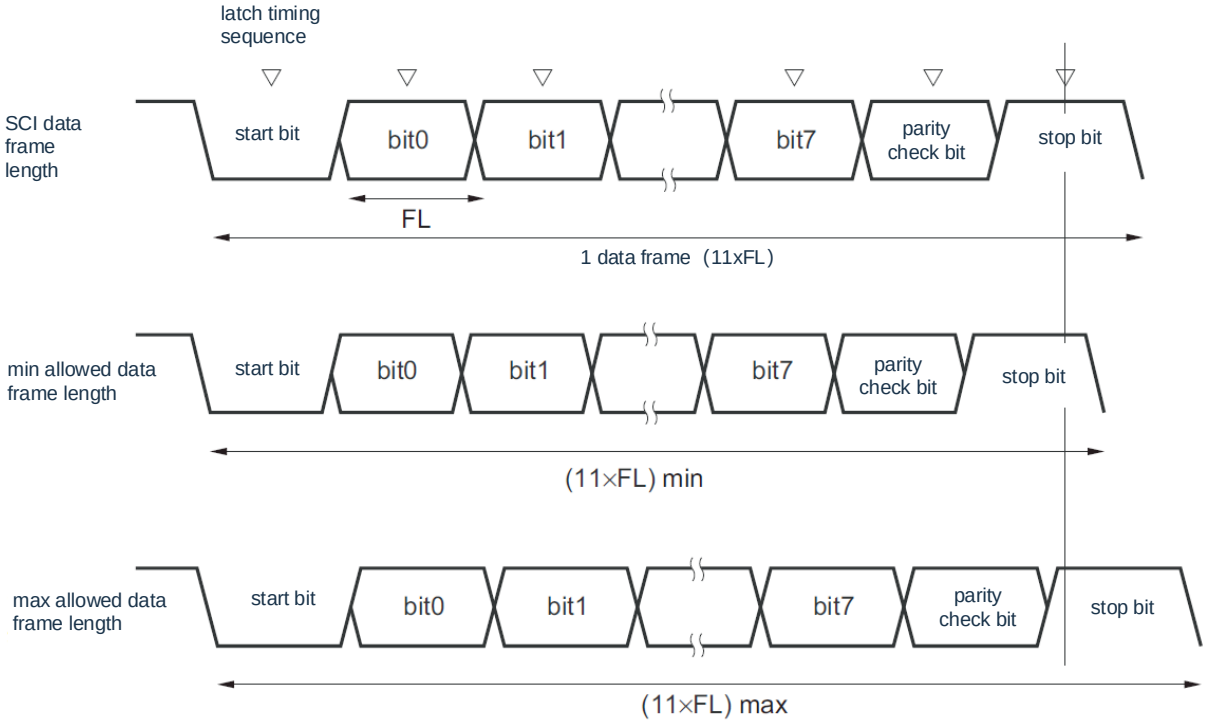
$$\text{(maximum acceptable baud rate)} = \frac{2 \times \times kNfr}{2 \times \times kNfr-} \times Brate$$

$$\text{(Minimum Acceptable Wavelet Rate)} = \frac{2 \times k \times (Nfr-1)}{2 \times \times kNfr-k-2} \times Brate$$

Brate: The calculated value of the baud rate for the receiver (reference to "19.7.4 (1) Baud rate formula")
k :SDRmn [15:9]+1
Nfr :Frame length of 1 data [bit]
= (start bit)+(data length)+(parity bit)+(stop bit)

Remark m: Cell number (m=0, 1n: Channel Number (n=1,3) mn=01,03,11

Figure 19-120 Acceptable range of baud rate at reception (in case of 1 data frame length=11 bits)



As shown in FIG. 19-120, after detecting the start bit, the latch timing of the received data depends on the bit15~9th set by serial data register mn (SDRmn). If the last data (the stop bit) can catch up with this latch sequence, it can be received normally.

19.7.4 Processing steps when an error occurs during UART (UART0~UART2) communication

The processing steps when an error occurs during UART (UART0~UART2) communication are shown in Figure 19-121 and Figure 19-122.

Figure 19-121 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 the channel n is in a receiver state.	This is to prevent an overflow error from occurring to end the next receipt during error handling.
Read the serial status register mn (SSRmn).		Determine the type of error and the read value is used to clear the error flag.
Clear trigger register mn for serial flag (SDIRmn) Write "1".	Clear the error flag.	By writing the read value of the SSRmn register directly to the SDIRmn register, errors in the read operation can only be cleared.

Figure 19122 Processing steps when frame errors occur

software operation	Hardware Status	Comments
Read the serial data register mn	The BFFmn bit of the SSRmn register is "0" and the channel n is in a receiver state.	This is to prevent an overflow error from occurring to end the next receipt during error handling.
Read the serial status register mn (SSRmn).		Determine the type of error and the read value is used to clear the error flag.
write serial flag clear trigger register mn (SIRmn).	Clear the error flag.	By writing the read value of the SSRmn register directly to the SDIRmn register, errors in the read operation can only be cleared.
Stop the Serial Channel from Register m (STm) The STmn location "1".	The serial channel allows the SEmn bit of the status register m (SEm) to be "0" and channel n to be running stopped.	
Synchronize with the communicating party.		A frame error can be considered to have occurred due to the start bit offset. It is therefore necessary to re-synchronize with that communicator and resume communication.
Start storing the serial channel for m (SSm) The SSmn location "1".	The serial channel allows the SEmn bit of the status register m (SEm) to be "1" and channel n to be operational.	

Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) mn=00~03,10~11.

19.8 Operation of LIN Communication

19.8.1 LIN Send

UART0 supports LIN communication in UART delivery.
LIN sends channel 0 of usage unit 0.

UART	UART0	UART1	UART2
LIN Communication Support	Yes	No	No
object channel	Channel 0 for SCI0	—	—
Pin Used	TxD0	—	—
interrupt	INTST0	—	—
	Interrupt at that end of the transfer may be selecte (single transfer mode) or buffer air-discontinuity (continuous transfer mode).		
Error detection flag	None		
length of transmit data	8-bit		
Transfer Rate Note	Max.fMCK/6[bps] (SDR.0[15:9]≥2), Min.fCLK/(2 2^{15}128) [bps]		
data phase	Positive output (default: High level). Invert output (default: Low level).		
parity bit	No parity bits.		
stop bit	Additional 1-bit.		
data orientation	LSB First		

Note It must be used within the context of peripheral functionality (reference data manual) that meets this condition and electrical characteristics, and 2.4/9.6/19.2kbps.

Remark fMCK: Object channel running clock frequency fCLK: system clock frequency

LIN is the abbreviation of Local Interconnect Network, which is a low-speed (1~20kbps) serial communication protocol to reduce automobile network cost. LIN communications are single-master communications, with up to 15 slave devices connected to a single master device.

The LIN slave is used for the control of switches, transmission devices, sensors, etc., which are connected to the main control device through the LIN.

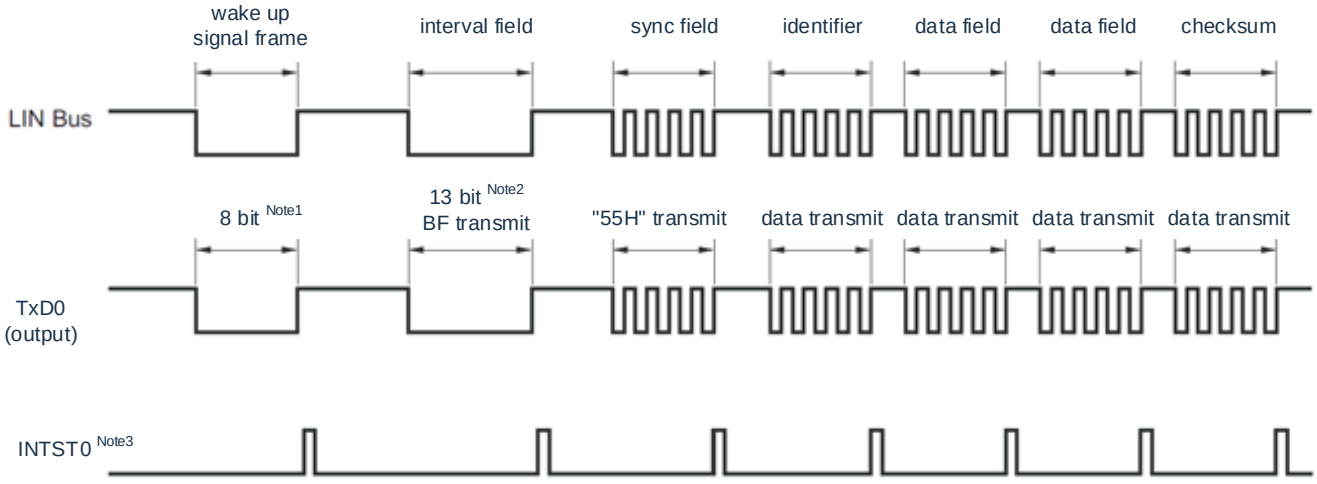
The LIN controls the network which connects CAN (Controller Area Network) and so on.

The LIN bus is a single-line bus, which connects nodes through ISDO9141-compliant transceivers.

According to the LIN protocol, the master device sends a frame with additional baud rate information, and the slave device receives the frame and corrects the baud rate error with the master device. Therefore, if the baud rate error of the slave device is not greater than ±15%, communication can be performed.

A summary of the LIN's send operations is shown in Figure 19-123.

Figure 19-123 LIN Send Operation



Note1. In order to meet the requirements of wake-up signal, set baud rate and send "80H" data to correspond.

2. The interval segment is specified as a 13-bit wide low-level output, so the baud rate used for the main transmission is N[bps]:

(Baud Rate for Interval) = 9/13 N

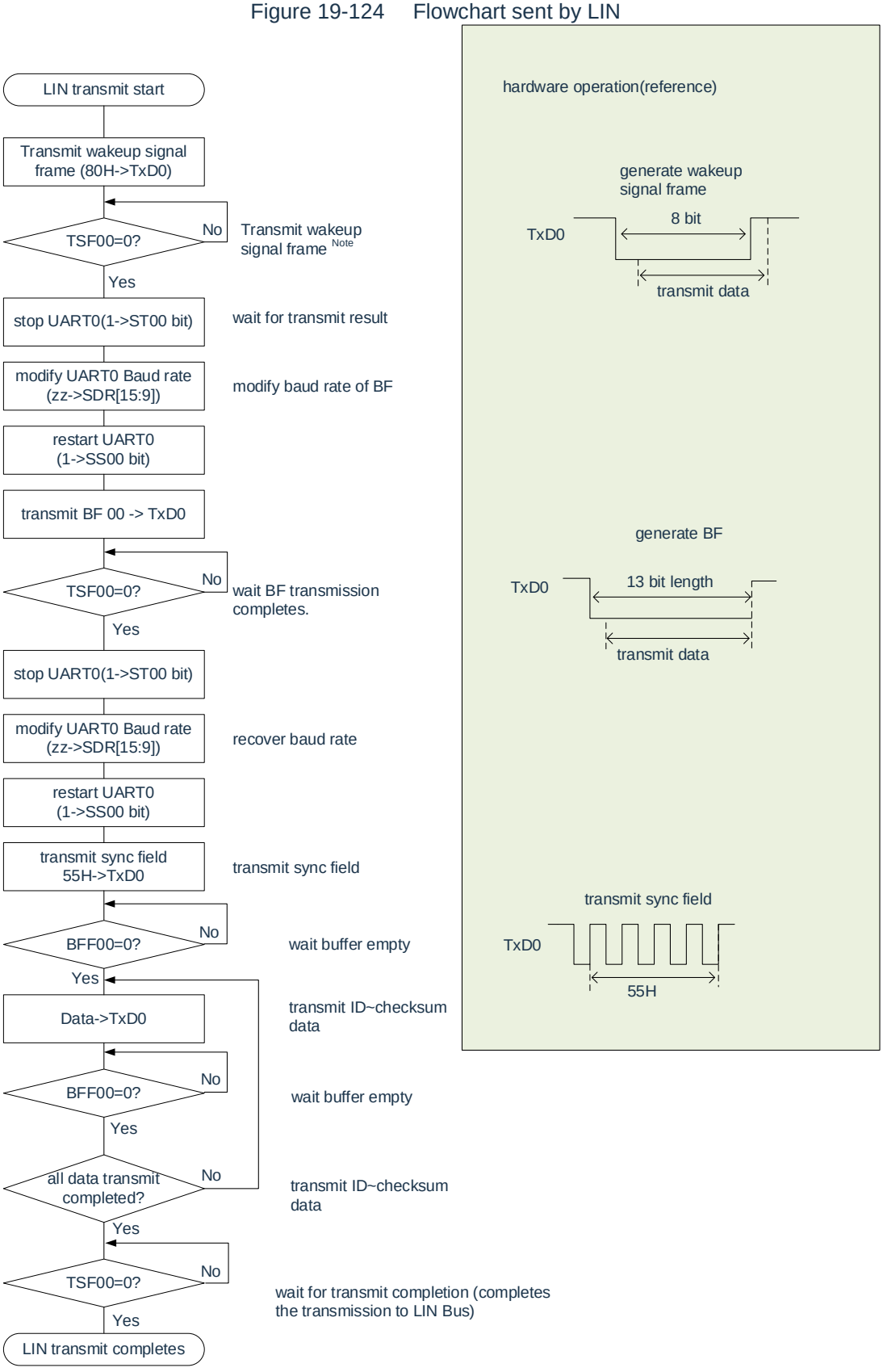
Send the data of "00H" through this baud rate to generate intervals.

3. Output INTST0 at the end of each data transmission, and also output

INTST0 at BF transmission.

Remark The software controls the interval between segments.

Figure 19-124 Flowchart sent by LIN



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 sending.

19.8.2 LIN receiving

In UART receiving, UART0 supports LIN communication.
The LIN receives the channel 1 of the UU.

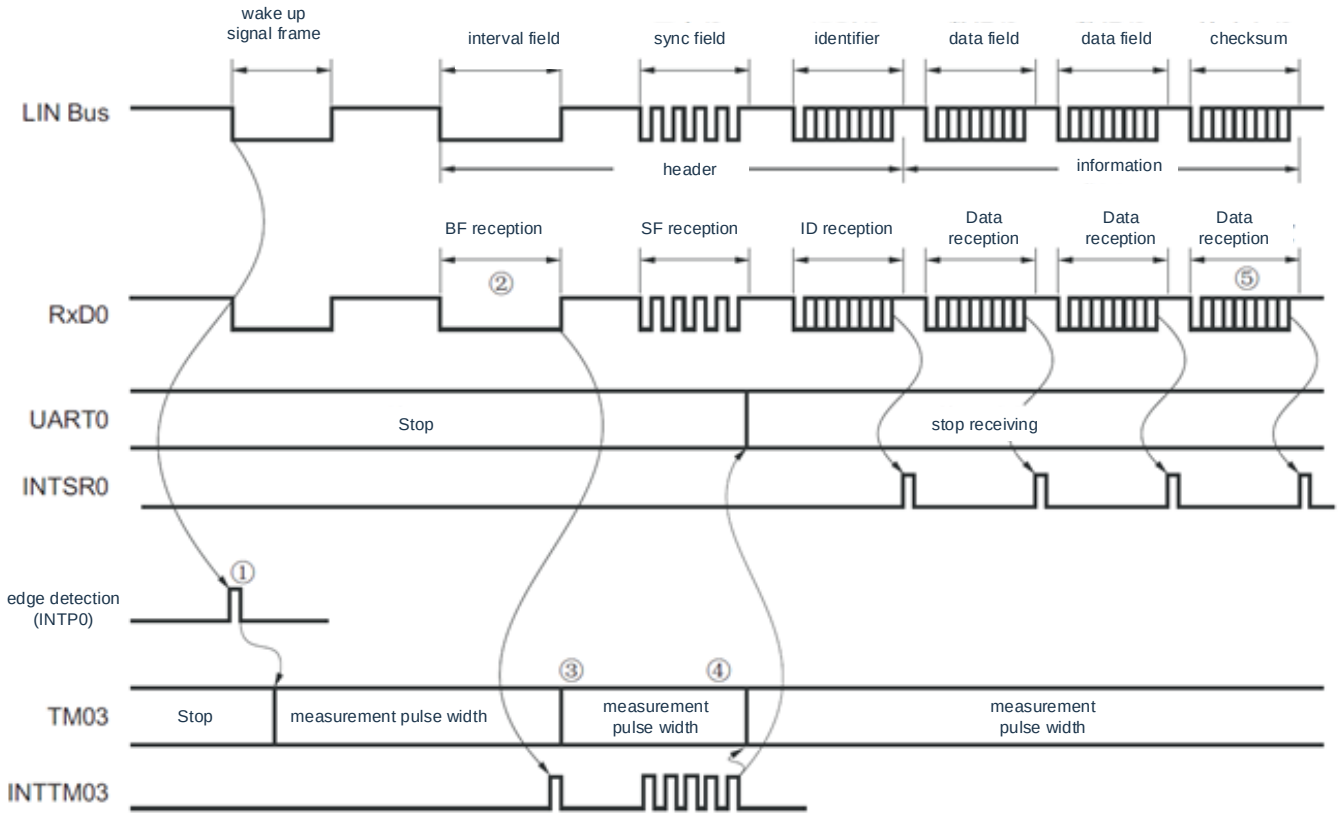
UART	UART0	UART1	UART2	UART3
LIN Communication Support	Yes	No	No	No
object channel	Channel 1 for SCI0	—	—	—
Pin Used	RxD0	—	—	—
interrupt	INTSR0	—	—	—
	Interrupt at that end of the transfer only (Disable from setting buffer air interrupt).			
error interrupt	INTSRE0	—	—	—
Error detection flag	· Frame Error Detection Flag (FEF01) · Overflow Error Detection Flag (OVF01)			
length of transmit data	8-bit			
Transfer Rate Note	Max.f_{MCK}/6[bps]（SDR01[15:9]≥2）、Min.f_{CLK}/(2 2^{15} 128)[bps]			
data phase	Positive output (default: High level). Invert output (default: Low level).			
parity bit	No parity bits (no parity).			
stop bit	Additional 1-bit.			
data orientation	LSB First			

Note It must be used within the scope of peripheral functional characteristics (reference data manual) that meet this condition and electrical characteristics.

Remark fMCK: Object channel running clock frequency fCLK: system clock frequency

A summary of the receive operations for the LIN is shown in Figure 19-125.

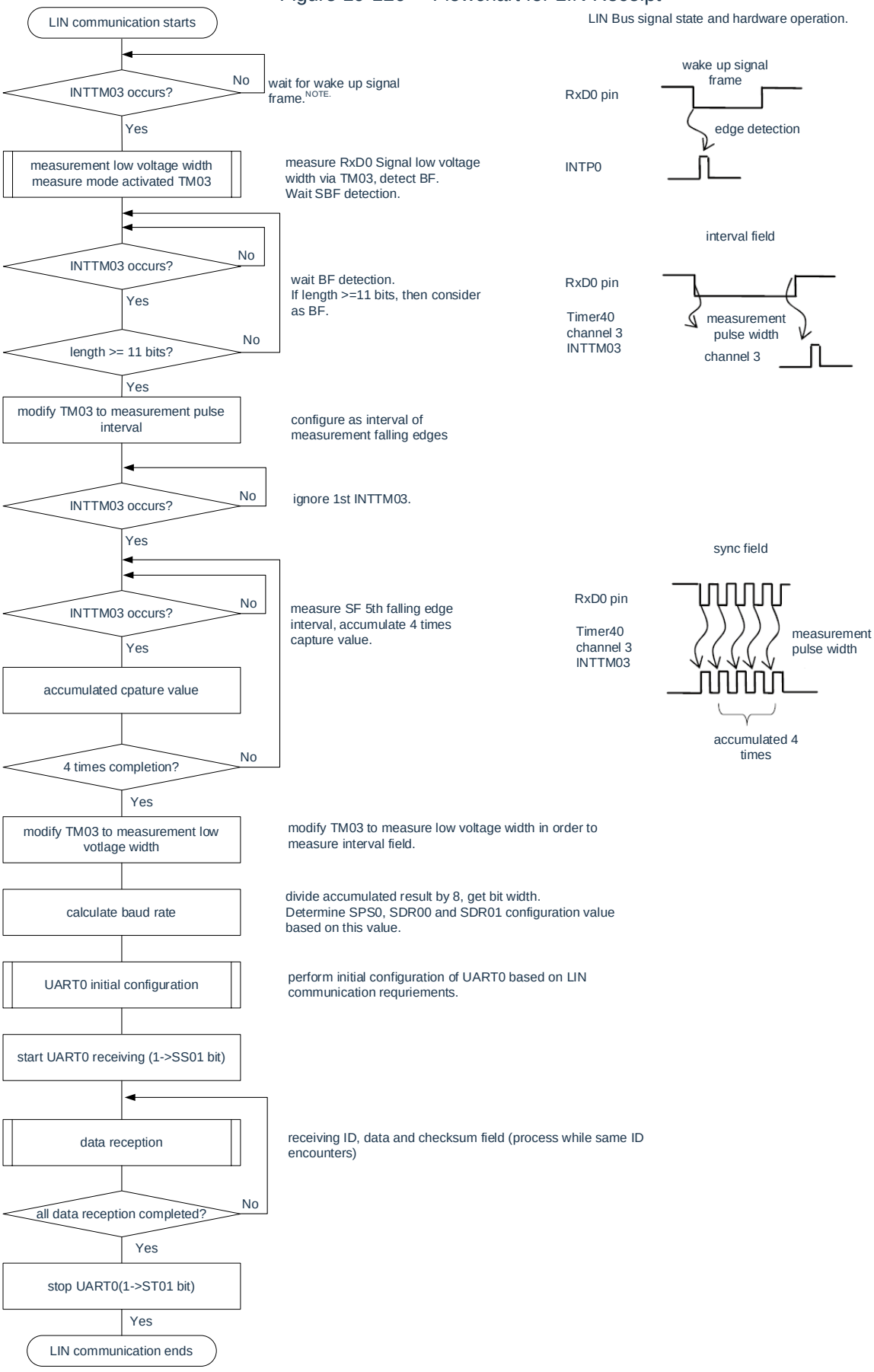
Figure 19-125 Receive operation for LIN



The signal processing flow is as follows:

(1) The wake-up signal is detected by detecting the INTP0 of the pin. When the wake signal is detected, the TM03 is set to measure the pulse width in order to measure the low level width of BF.
(2) If the falling edge of BF is detected, TM03 starts to measure the low level width and captures the rising edge of BF. The BF signal is judged according to the captured value.
(3) When BF reception ends normally, TM03 must be set as the measurement pulse interval, and the interval of RxD0 signal falling edge of 4 synchronizations.
(4) Calculating the baud rate error according to the bit interval of the synchronization section (SF). The baud rate must then be adjusted (reset) after the UART0 run has been paused.
(5) The checksum segment must be distinguished by software. You must also initialize the UART0 after receiving the checksum segment through the software and set it to the BF receive wait state again.

Figure 19-126 Flowchart for LIN Receipt



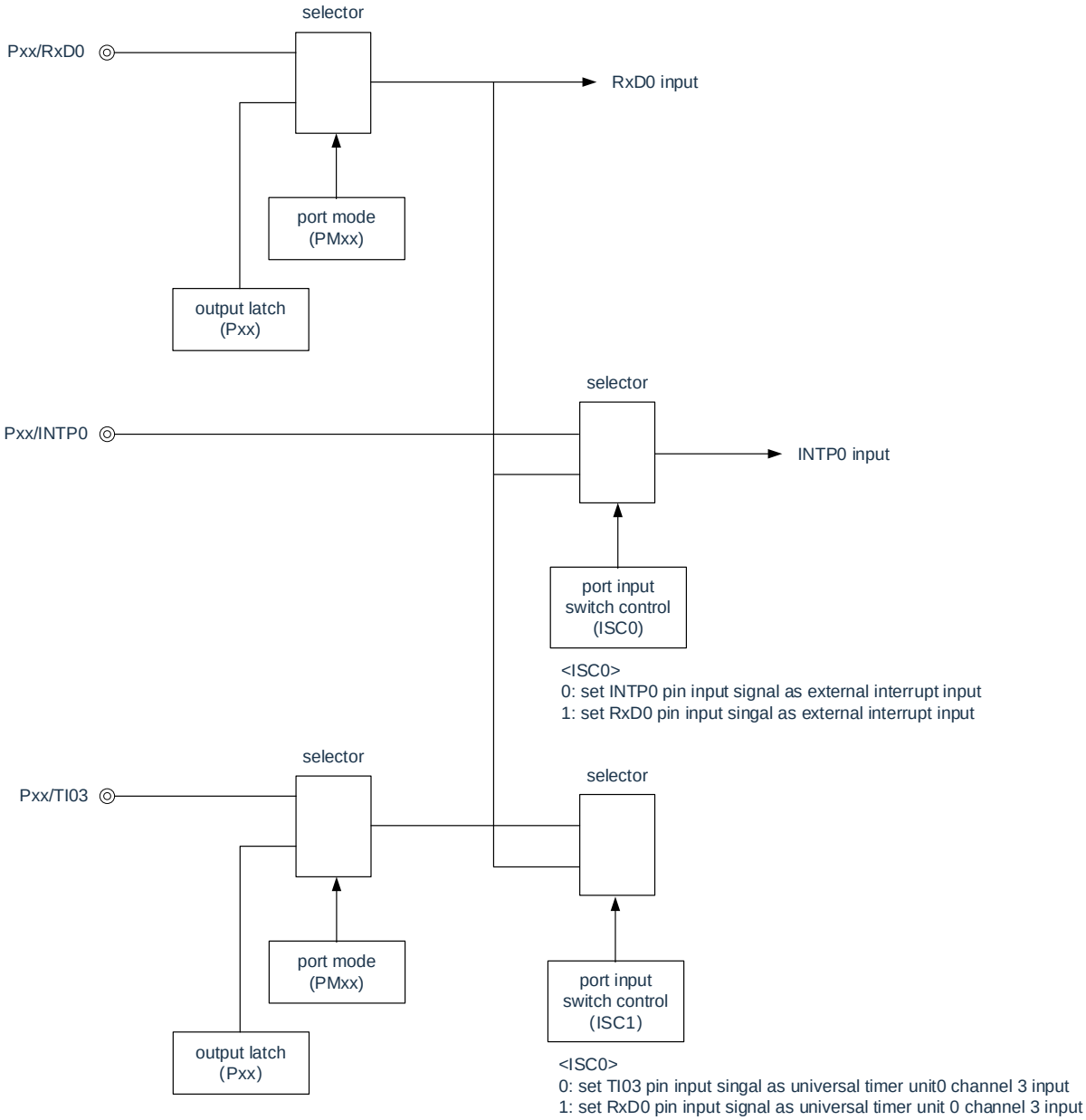
Note Only during sleep.

The port structure diagram for LIN receive operations is shown in Figure 19-127.

The wake-up signal sent by the LIN master is received through edge detection of the INTP0. The invention can measure the length of the synchronization segment sent by the LIN master and calculate the baud rate error through external event capture operation.

The input source for the received port input (RxD0) can be input to the external interrupt (INTP0) and timer array unit without external connection by port input switching control (ISC0/ISC1).

Figure 19-127 Port Map for LIN Receive Operations



Remark ISC0, ISC1: Enter the bit0 and bit1 for the Switch Control Register (ISC) (see Figure 19-21)

Peripheral features for LIN communication operations are summarized as follows:

<Peripheral Features Used>

· External interrupt (INTP0): Detection of wake-up signal

Purposes of use: Detects edges of wake-up signals and the start of communication.

· Channel 3 of the universal timer unit: Detection of Baud Rate Error and Detection of Interval (BF)

Purposes of use: The length of the synchronization section (SF) is detected and the baud rate error is detected by dividing its length by bits (the interval of the RxD0 input edge is measured by capture mode). A low level width is measured to determine whether it is a spacer (BF).

· Channel 0 and channel 1 (UART0) of universal serial communication unit 0 (SCI0)

19.9 Operation of Simple I2C (IIC00, IIC01, IIC10, IIC11, IIC20,IIC21) Communication

This is the capability of clock synchronization with multiple devices through two lines of serial clock (SCL) and serial data (SDA). Since this simple I2C is designed for single communication with EEPROM, flash memory, A/D converter, etc., it is only used as master device.

For the start condition and stop condition, the AC specification must be observed, and the control register must be handled by software.

[Sending and Receiving Data]

· Master Send, Master Receive (only for single master master master master functions)

· ACK output function Note, ACK detection function

· 8-bit data length (when sending an address, specifying the address with 7 bits high and R/W control with lowest bits)

· A start condition and a stop condition are generated by the software.

[Interrupt Capabilities]

· End of Transfer Interrupt

[Error Detection Flag]

• ACK Error

※[Features not supported by Simple I2C]

· Slave send, Dependent Receive

· Multi-Master (Quorum Failure Detection)

· Waiting for detection

Note When the last data is received, if "0" is written to the SDOEmn bit (SDOEm register) to stop the serial communication data output. Refer to "19.9.3 (2) Process Flow".

Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) mn=00~03,10~11.

The channels 0~3 of SCI0 and 0~1 of SCI1 are the channels supporting simple I2C (IIC00, IIC01,IIC10,IIC11,IIC20,IIC21).

Simple I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) has the following four kinds of communication running:

· address segment sending (Reference 19.9.1)

· data transmission (Reference 19.9.2)

· data receiving (Reference 19.9.3)

· Generation of Stop Condition (Reference 19.9.4)

19.9.1 address segment sending

Address segment sending is a send run that is first performed when I2C communication, specifically specifying a transfer object (slave). After the start condition is generated, the address (7 bits) and the transmission direction (1 bits) are transmitted as 1 frames.

Simple I2C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
object channel	SCI0's Channel 0	SCI0's Channel 1	SCI0's Channel 2	SCI0's Channel 3	SCI1's Channel 0	SCI1's Channel 1
Pin Used	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	INTEGER00	INTEGER01	INTEGER10	INTEGER11	INTEGER20	INTEGER21
	Only interrupt at that end of the transfer (no buffer interrupt can be selected).					
Error detection flag	ACK Error Detection Flag (PEFmn)					
length of transmit data	8-bit (High 7 bits as addresses and low 1 bits as R/W controls)					
Transfer Rate Note 2	Max.$_{fMCK}$/4[Hz](SDRmn[15:9]≥1)$_{fMCK}$: The operating clock frequency of the object channel, however, must meet the following conditions in each mode of I2C: · Max.1MHz · Max.400kHz (Quick Mode) · Max.100kHz (Standard Mode)					
data level	Positive output (default: High level).					
parity bit	No parity bits.					
stop bit	Additional 1-bit (for ACK reception).					
data orientation	MSB First					

Note 1. To communicate through simple I2C, the N-channel drain open output mode (POMxx=1) must be set through the port output mode register (POMxx). For details, refer to "Registers for 2.3 control port functionality" and "Register settings for 2.5 reuse functionality."

2. Must be used within the scope of peripheral functional characteristics (reference data manual) that satisfy this condition and electrical characteristics.

Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) mn=00~03,10~11.

(1) Register settings

Figure 19-128 Simple I2C (IIC00, IIC01, IIC10, IIC11, IIC20,IIC21) address segments sent Examples of register setting content for

(a) serial mode register mn (SMRmn)

	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	0	SISmn0	1	0	0	MDmn2	MDmn1	MDmn0
	0/1	0						0 Note1		0 Note1				1	0	0

channel n operational clock (fMCK)
0: SPSm register configured pre-scaler output clock CKm0
1: SPSm register configured pre-scaler output clock CKm1

Operation mode of channel n
0: Transmit completion interrupt

(b) serial communication operation configuration registermn mn(SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1	SLCmn0	0	1	DLSmn1	DLSmn0
	1	0	0	0		0	0	0	0		0 Note2	1			1 Note3	1

parity check bit configuration
00B: no parity check

stop bit configuration
01B: append 1 bit (ACK)

(c) serial data regsiter mn (SDRmn) (low 8 bit: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	baud rate configuration							0	configuration of transmit data(Address+R/W)							

SIOr

(d) serial output register m (Som)

	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
					0/1	0/1	0/1	0/1					0/1	0/1	0/1	0/1

generate start condition via operating Somn bit.

(e)serial otuput enable register m (SOEm)

	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
													0/1	0/1	0/1	0/1

before generating start condition, SOEmn bit is '0', after generating start condition, SOEmn bit is '1'.

(f) serial channel start register m (SSm) …. Only set bit of target channel to 1.

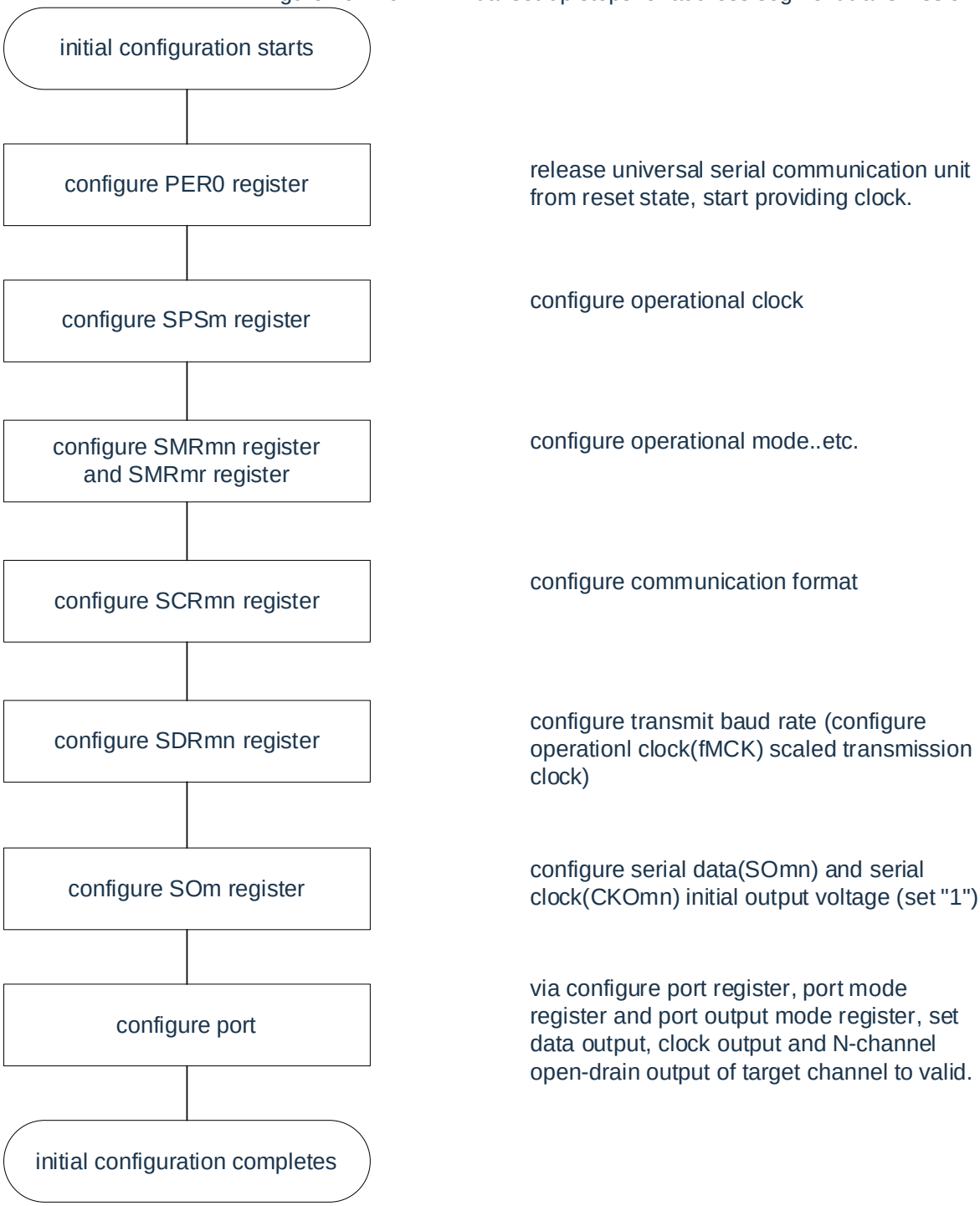
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0
													0/1	0/1	0/1	0/1

Note1. Limited to SMR00, SMR03, SMR11.
2. Limited to SCR00, SCR02, SCR10.
3. Limited to SCR00 and SCR01 registers, other fixed to "1".

Note 1.m: Cell 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. □ : Set in IIC mode for Fixed. ■ : Cannot set (set initial value).
×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).
0/1: The "0" or "1" is set according to the user.

(2) Procedure

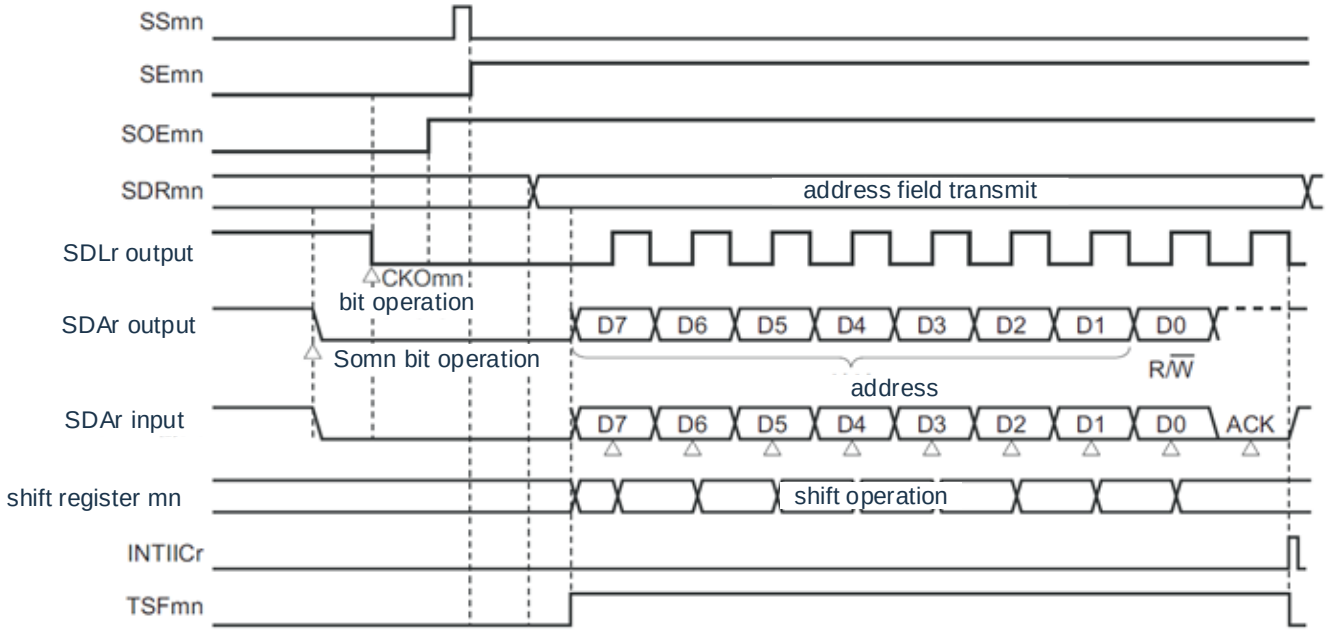
Figure 19-129 Initial set-up steps for address segment transmission



Remark At the end of the initial set-up, Simple I2C (IIC00, IIC01, IIC10, IIC11, IIC20,IIC21) is output-disabled and in a run-stop state.

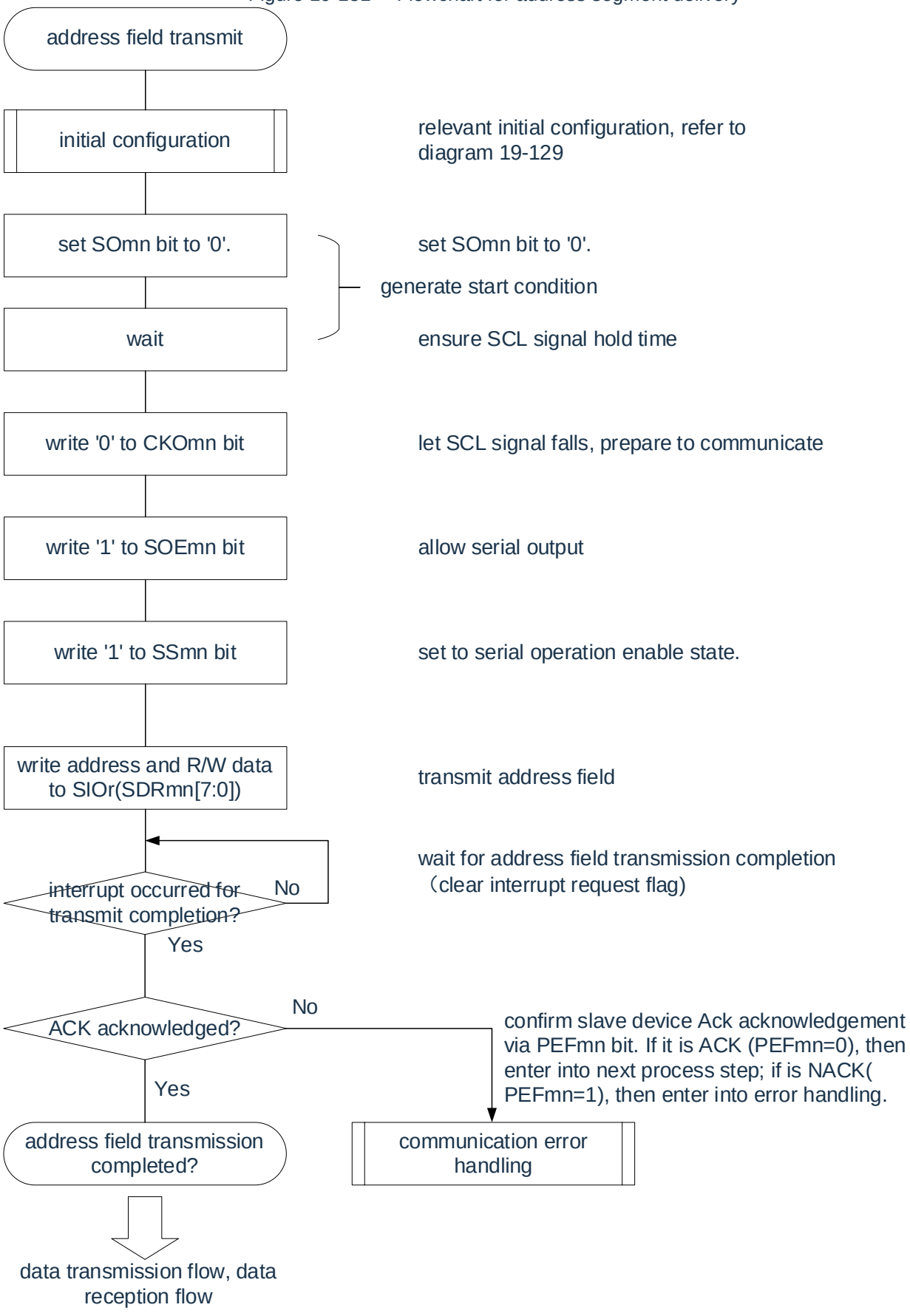
(3) process flow

Figure 19-130 Time Sequence Diagram for Address Segment Transmission



Remark m: Cell 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 19-131 Flowchart for address segment delivery



19.9.2 data transmission

Data transmission is the operation of transmitting data to the transmission object (slave device) after the address segment is transmitted. A stop condition is generated after all data is sent to the object slave and the bus is released.

Simple I²C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
object channel	SCI0's Channel 0	SCI0's Channel 1	SCI0's Channel 2	SCI0's Channel 3	SCI1's Channel 0	SCI1's Channel 1
Pin Used	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	INTEGER00	INTEGER01	INTEGER10	INTEGER11	INTEGER20	INTEGER21
	Only interrupt at that end of the transfer (no buffer interrupt can be selected).					
Error detection flag	ACK Error Flag (PEFmn)					
length of transmit data	8-bit					
Transfer Rate Note 2	Max.$_{fMCK}$/4 [Hz] (SDRmn [15:9]≥1) $_{fMCK}$: The operating clock frequency of the object channel, however, must meet the following conditions in each mode of I2C: · Max.1MHz · Max.400kHz (Quick Mode) · Max.100kHz (Standard Mode)					
data level	Positive output (default: High level).					
parity bit	No parity bits.					
stop bit	Additional 1-bit (for ACK reception).					
data orientation	MSB First					

Note 1. To communicate through simple I2C, the N-channel drain open output mode (POMxx=1) must be set through the port output mode register (POMxx). Refer to "Registers for 2.3 Control Port Functionality" and "Register Settings for 2.5 Reuse Functionality".

2. Must be used within the scope of peripheral functional characteristics (reference data manual) that satisfy this condition and electrical characteristics.

Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) mn=00~03,10~11.

(1) Register settings

Figure 19-132 Simple I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) when sending data

Examples of register setting content for

(a) serial mode register mn (SMRmn)……do not operate this register wihle 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 wihle 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 regsiter mn (SDRmn) (low 8 bit: SIOr) ……only lower 8 bits valid wihle 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							

SIOr

(d) serial output register m (Som) ……do not operate this register wihle 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 otuput enable register m (SOEm) ……do not operate this register wihle 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 wihle 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

Note 1. Limited to SMR01, SMR03, SMR11 registers only.

2. Limited to SCR00, SCR02, SCR10 registers only.
3. Limited to SCR00 and SCR01 registers, other fixed to "1".
4. No set-up is required because the address segment is already set when it is sent.
5. During a communication run, the value changes due to communication data.

Note 1.m: Cell 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. □ : Set in IIC mode for Fixed. ■ : Cannot set (set initial value).
×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).
0/1: The "0" or "1" is set according to the user.

(2) process flow

Figure 19-133 Time Series of Data Transmission

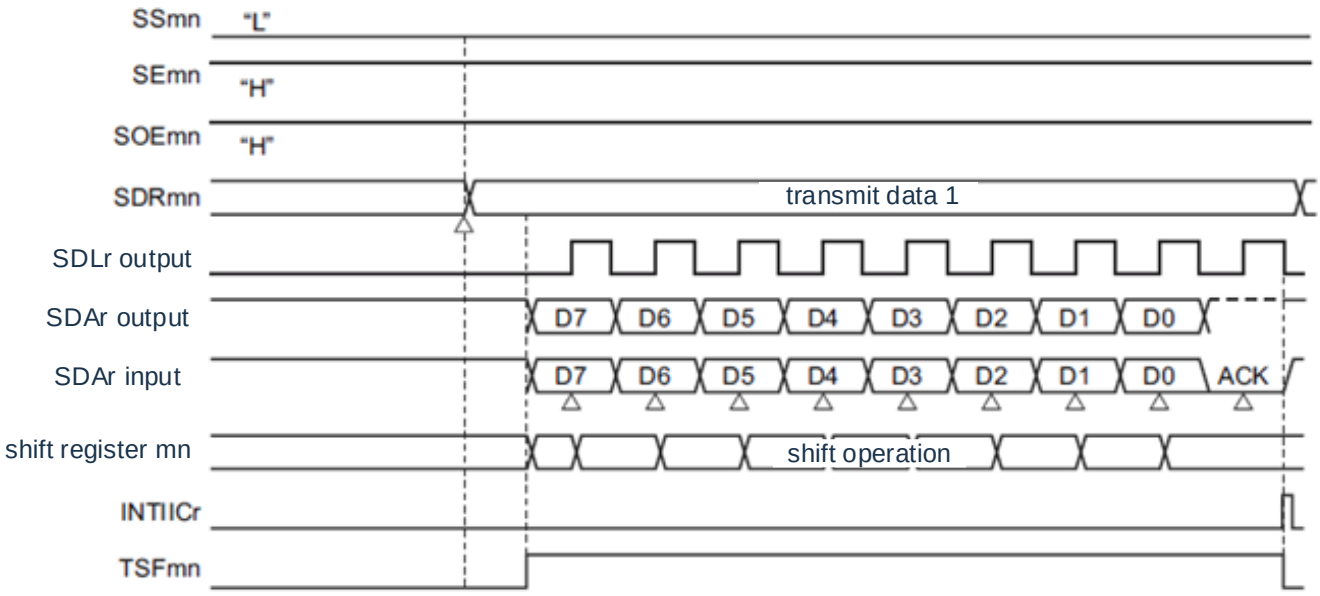
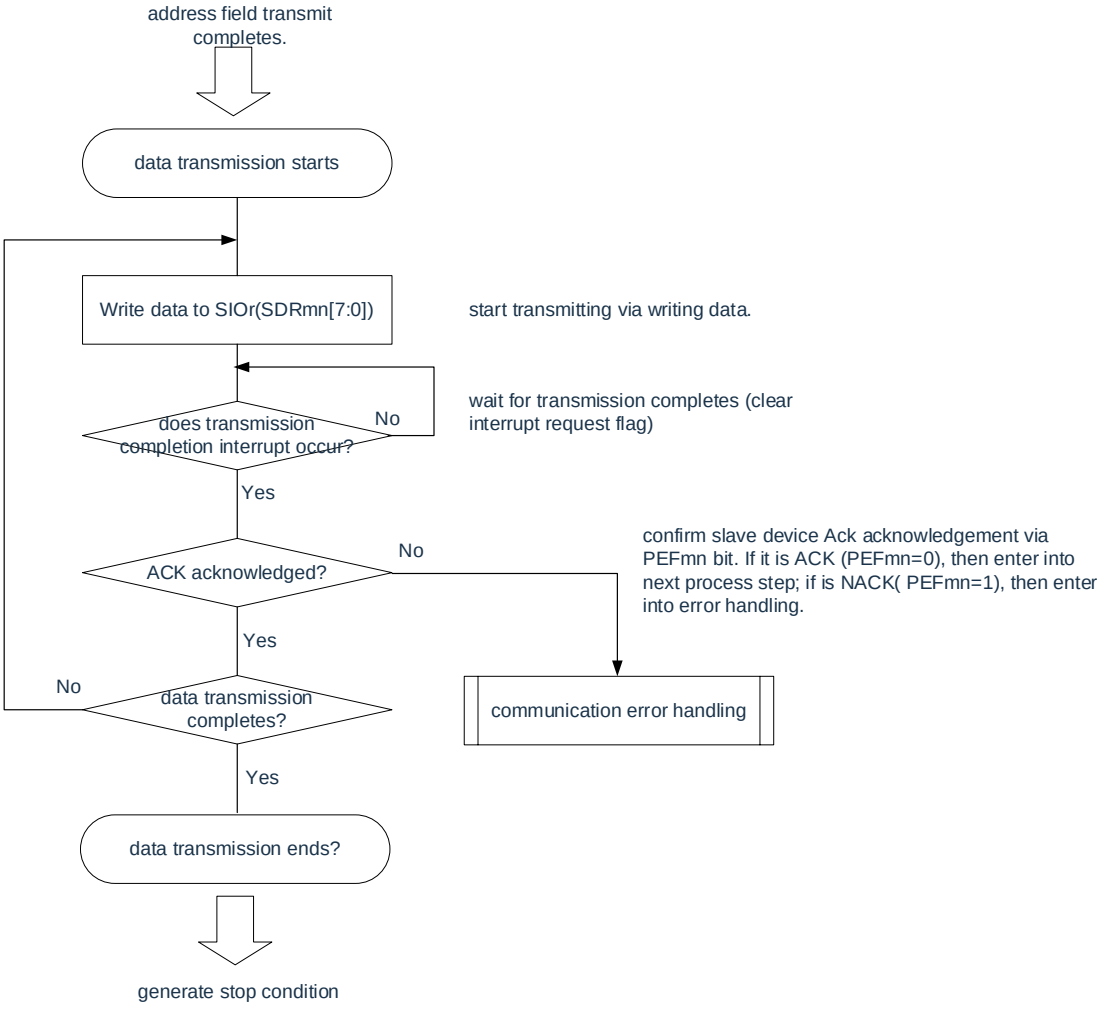


Figure 19-134 Flow chart for data delivery



19.9.3 data receiving

Data reception is a run that receives data from a transfer object (slave) after sending an address segment. A stop condition is generated and the bus is released after receiving all the data from the object slave.

Simple I²C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
object channel	SCI0's Channel 0	SCI0's Channel 1	SCI0's Channel 2	SCI0's Channel 3	SCI1's Channel 0	SCI1's Channel 1
Pin Used	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	INTEGER00	INTEGER01	INTEGER10	INTEGER11	INTEGER20	INTEGER21
	Only interrupt at that end of the transfer (no buffer interrupt can be selected).					
Error detection flag	Only the overflow error detection flag (OVFmn).					
length of transmit data	8-bit					
Transfer Rate Note 2	Max.$_{fMCK}$/4 [Hz] (SDRmn [15:9]≥1) $_{fMCK}$: The operating clock frequency of the object channel, however, must meet the following conditions in each mode of I2C: · Max.1MHz · Max.400kHz (Quick Mode) · Max.100kHz (Standard Mode)					
data level	Positive output (default: High level).					
parity bit	No parity bits.					
stop bit	Additional 1 bits (ACK send).					
data orientation	MSB First					

Note 1. To communicate through simple I2C, the N-channel drain open output mode (POMxx=1) must be set through the port output mode register (POMxx). Refer to "Registers for 2.3 Control Port Functionality" and "Register Settings for 2.5 Reuse Functionality".

2. Must be used within the scope of peripheral functional characteristics (reference data manual) that satisfy this condition and electrical characteristics.

Remark m: Cell number (m=0,1)n: Channel Number (n=0~3) mn=00~03,10~11.

(1) Register settings

Figure 19-135 Simple I2C (IIC00, IIC01, IIC10, IIC11, IIC20,IIC21) when receiving data

Examples of register setting content for

(a) serial mode register mn (SMRmn)……do not operate this register wihle data is transmitting or receiving.

	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	0	SISmn0	1	0	0	MDmn2	MDmn1	MDmn0
	0/1	0						0 Note1		0 Note1				1	0	0

(b) serial communication operation configuration register mn (SCRmn)……do not operate bits other than TXEmn and RXEmn of this register wihle 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	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1	SLCmn0	0	1	DLSmn1	DLSmn0
	0	1	0	0		0	0	0	0		0 Note2	1			1 Note3	1

(c) serial data regsiter mn (SDRmn) (low 8 bit: SIOr)

	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)							

SIOr

(d) serial output register m (Som) ……do not operate this register wihle data is transmitting or receiving.

	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
					0/1 Note5	0/1 Note5	0/1 Note5	0/1 Note5					0/1 Note5	0/1 Note5	0/1 Note5	0/1 Note5

(e) serial otuput enable register m (SOEm) ……do not operate this register wihle data is transmitting or receiving.

	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
													0/1	0/1	0/1	0/1

(f) serial channel start register m (SSm) ……do not operate this register wihle data is transmitting or receiving.

	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	0/1	0/1	0/1

Note1. Limited to SMR01, SMR03, SMR11 registers only.

2. Limited to SCR00, SCR02, SCR10 registers only.
3. Limited to SCR00 and SCR01 registers, other fixed to "1".
4. No set-up is required because the address segment is already set when it is sent.
5. During a communication run, the value changes due to communication data.

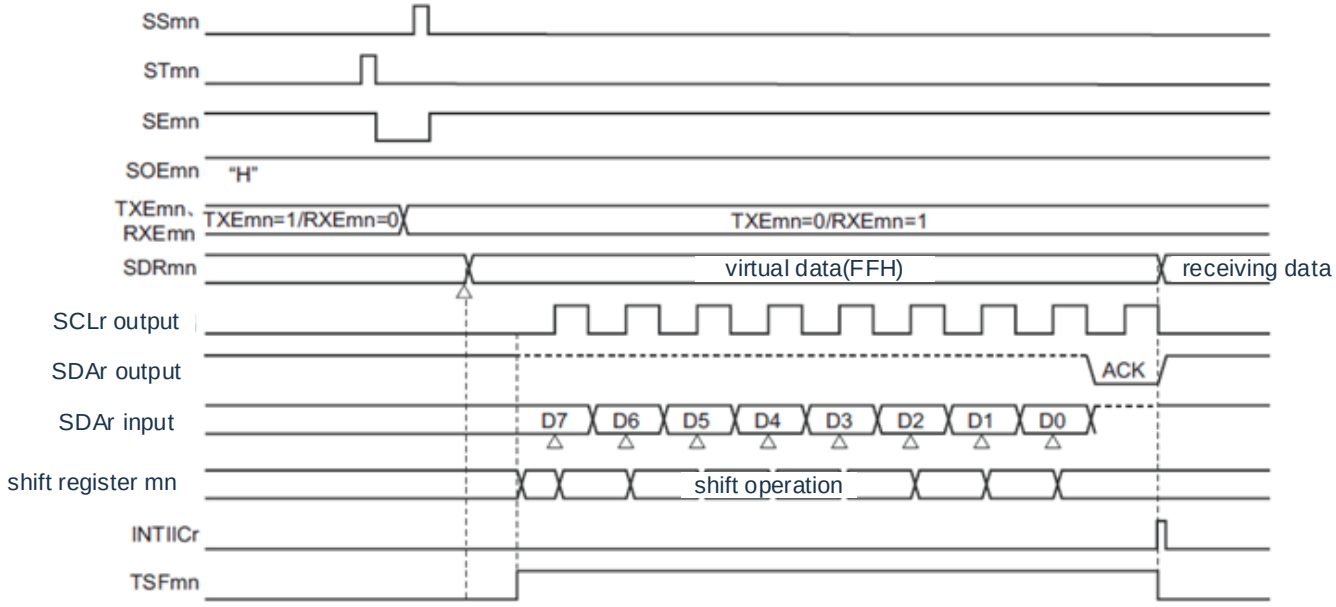
Note 1.m: Cell number (m=0,1)n: Channel number (n=0~3)r: IIC number (r=00,01,10,11,20,21)
mn=00~03,10~11

2. □ : Set in IIC mode for Fixed. ■ : Cannot set (set initial value).
×: This is a bit that cannot be used in this mode (set the initial value if not used in other modes).
0/1: The "0" or "1" is set according to the user.

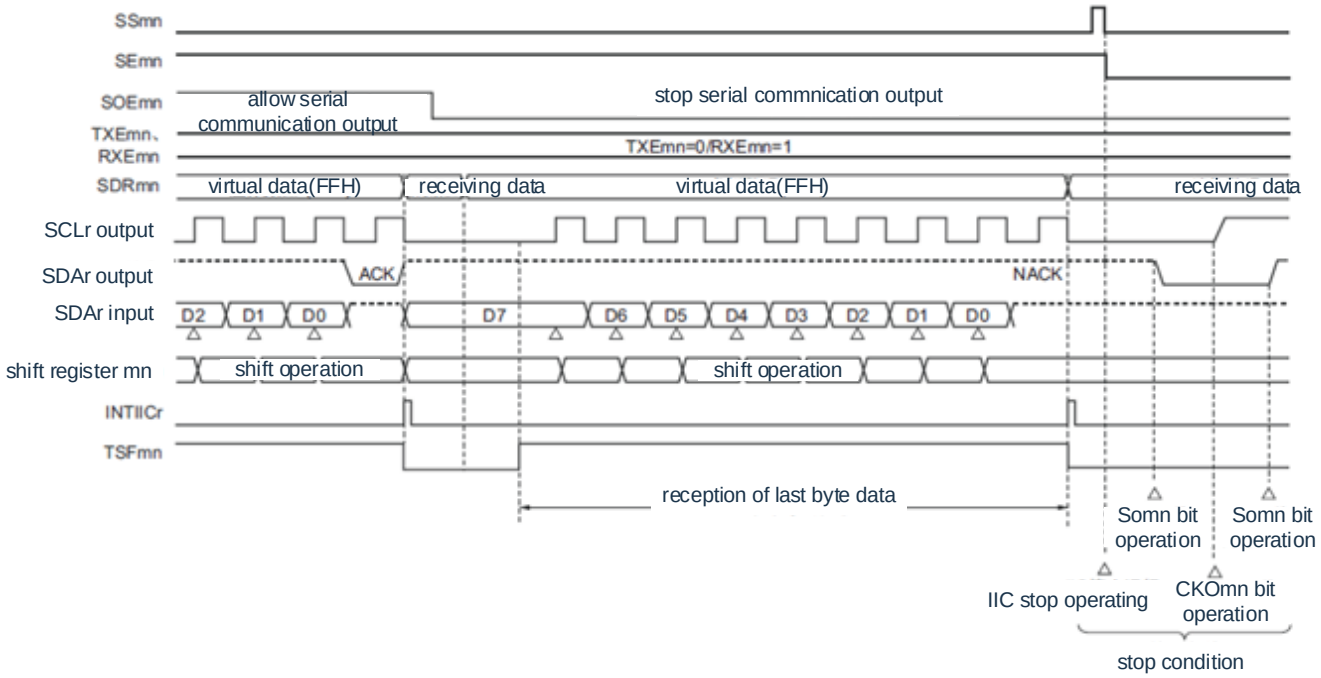
(2) process flow

Figure 19-136 Time Series for Data Reception

(a) Start of receiving data

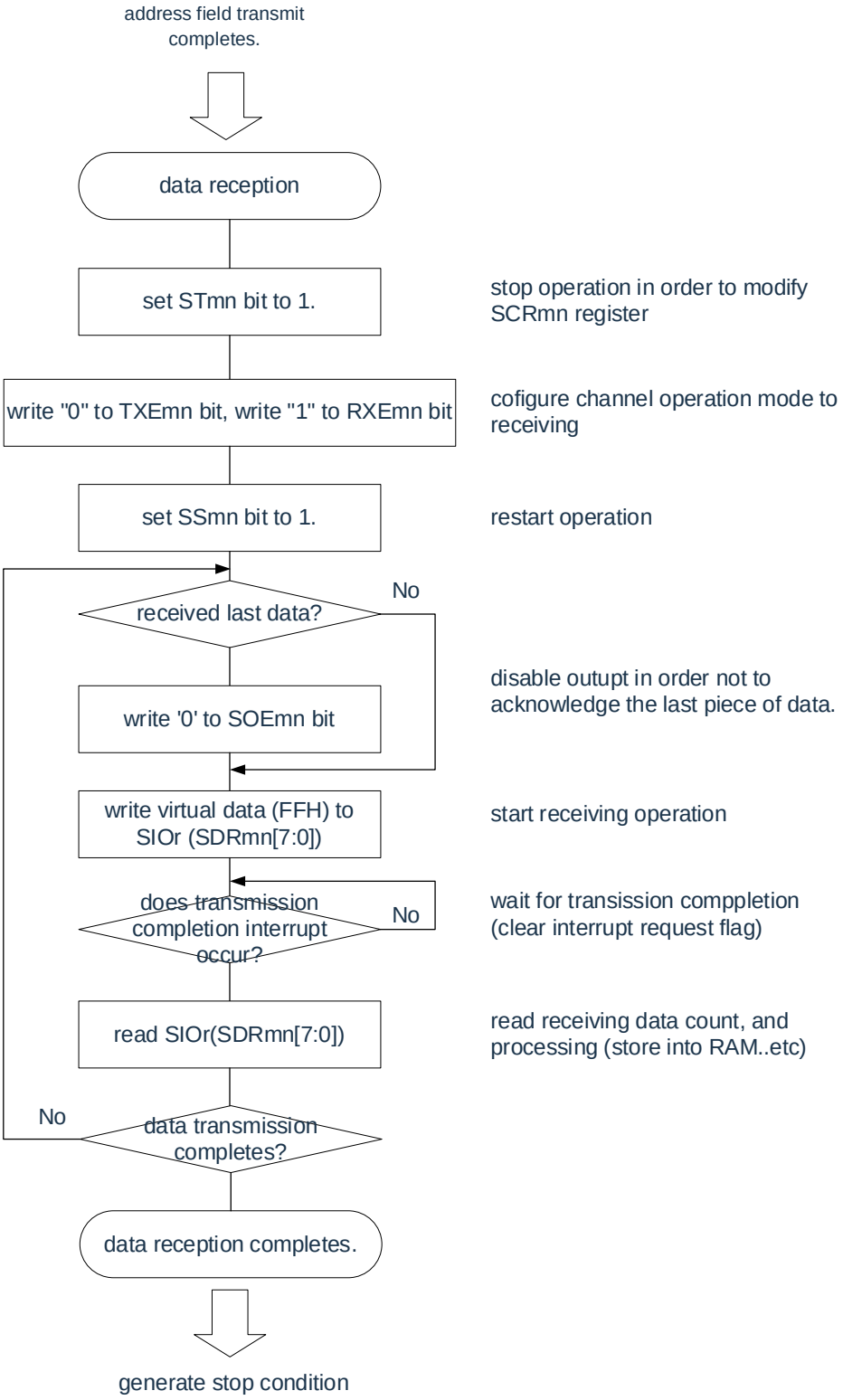


(b) Status of receipt of final data



Remark m: Cell 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 19-137 Flow chart for data reception



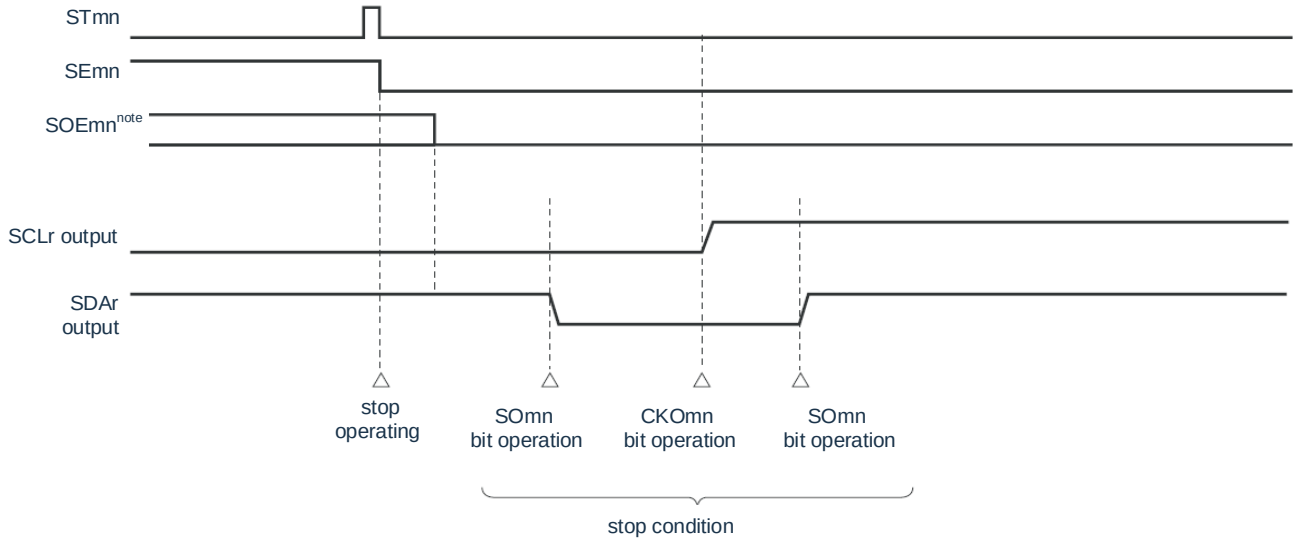
Note No ACK(NACK) is output when the last data is received. Thereafter, the communication is terminated by stopping the STmn position '1' of the serial channel stop register m (STm).

19.9.4 Generation of Stop Condition

After sending and receiving all the data with the object slave, a stop condition is generated and the bus is released.

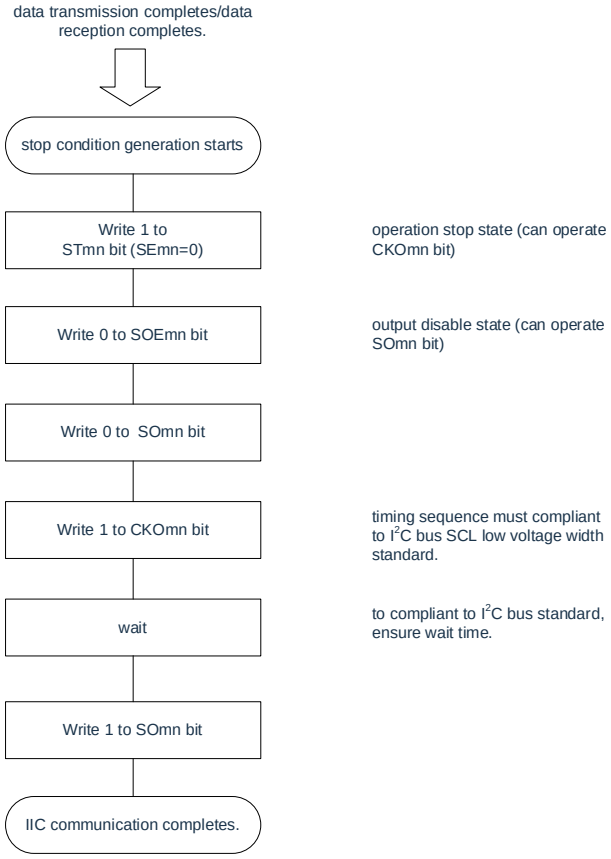
(1) process flow

Figure 19-138 Time sequence diagram for generating stop condition



Note At the time of receiving, the SOEmn position "0" of the permit register m (SOEm) was already output serially.

Figure 19-139 Flow chart for generating stop conditions



19.9.5 Calculation of transfer rate

The transfer rate for simple I2C (IIC00, IIC01, IIC10, IIC11, IIC20,IIC21) communication can be calculated using the following formula.

$$\text{(Transfer Rate)} = \{\text{Runtime Clock } (f_{MCK}) \text{ Frequency}\} \div (\text{SDRmn}[15:9]+1) \div 2$$

Note 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 simple 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.

Note 1. Because the value of SDRmn[15:9] is the value of bit15~9 of the SDRmn (0000001B~1111111B), it is 1~127.
2.m: Cell number (m=0,1)n: Channel Number (n=0~3) mn=00~03,10~11.

The runtime clock (fMCK) depends on the bit15 (CKSmn bit) of the serial clock selection register m (SPSm) and the serial mode register mn (SMRmn).

Table 19-6. Simple I2C Operating Clock Selection

SMRmn register	SPSm register								Runtime Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK}=32 MHz Runtime
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 above									Disable from setting.	

Note To change the clock selected as fCLK (change the value of the System Clock Control Register (CKC), you must change after stopping Universal Serial Communication Unit (SCI) =000FH.

Note 1.X: Ignore

2.m: Cell number (m=0,1)n: Channel Number (n=0~3) mn=00~03,10~11.

An example of setting the I2C transfer rate at fMCK=fCLK=32 MHz is shown below.

I^2C transfer mode (Expected Transfer Rate)	f_{CLK}=32MHz			
	Runtime 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% Notes
1MHz	f_{CLK}	18	0.84MHz	16.0% Notes

Note The error cannot be set to '0'% because the SCL signal has a 50% duty cycle.

19.9.6 Processing steps when an error occurs in a simple I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) communication process

The processing steps when an error occurs during a simple I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) communication are shown in Figures 19-140 and 19-141.

Figure 19-140 Handling steps when overflow errors occur

software operation	Hardware Status	Comments
Read the serial data register mn →	The BFFmn bit of the SSRmn register is "0" and the channel n is in a receiver state.	This is to prevent an overflow error from occurring to end the next receipt during error handling.
Read the serial status register mn (SSRmn).		The type of error is determined and the read value is used to clear the error flag.
Clear trigger register mn for serial flag →	Clear the error flag.	By writing the read value of the SSRmn register directly to the SDIRmn register, errors in the read operation can only be cleared.

Figure 19-141 Processing steps when an ACK error occurs in a simple I2C mode

software operation	Hardware Status	Comments
Read the serial status register mn (SSRmn).		Determine the type of error and the read value is used to clear the error flag.
write serial flag clear trigger register mn →	Clear the error flag.	By writing the read value of the SSRmn register directly to the SDIRmn register, errors in the read operation can only be cleared.
Stop the Serial Channel from Register m (STm) →	The serial channel allows the SEmn bit of the status register m (SEm) to be "0" and channel n to be running stopped.	The slave device is not ready to receive because no ACK is returned. Accordingly, a stop condition is generated and the bus is released, communication is started again from the start condition, or a restart condition can also be generated and restarted from the address transmission.
Generate a stop condition.		
Generate start condition.		
Start the serial channel for register m (SSm) The SSmn location "1". →	The serial channel allows the SEmn bit of the status register m (SEm) to be "1" and channel n to be operational.	

Remark m: Cell 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

Chapter 20 serial interface IICA

20.1 The Function of Serial Interface IICA

The serial interface IICA has the following 3 modes.

(1) IDLE Mode

This is a mode for non-serial transfer, which reduces power consumption.

(2) I^2C bus mode (multi-master capable)

This mode communicates 8-bit data with multiple devices through two lines of the serial clock (SCLAn) and the serial data bus (SDAAn). The master device can generate start condition, address, indication of transmission direction, data and stop condition for slave devices. The slave device automatically detects the received status and data through the hardware. This feature simplifies the I2C bus control section of the application.

Because the SCLAn pin and the SDAAn pin of the serial interface IICA are used as drain open output, the serial clock line and the serial data bus need pull-up resistors.

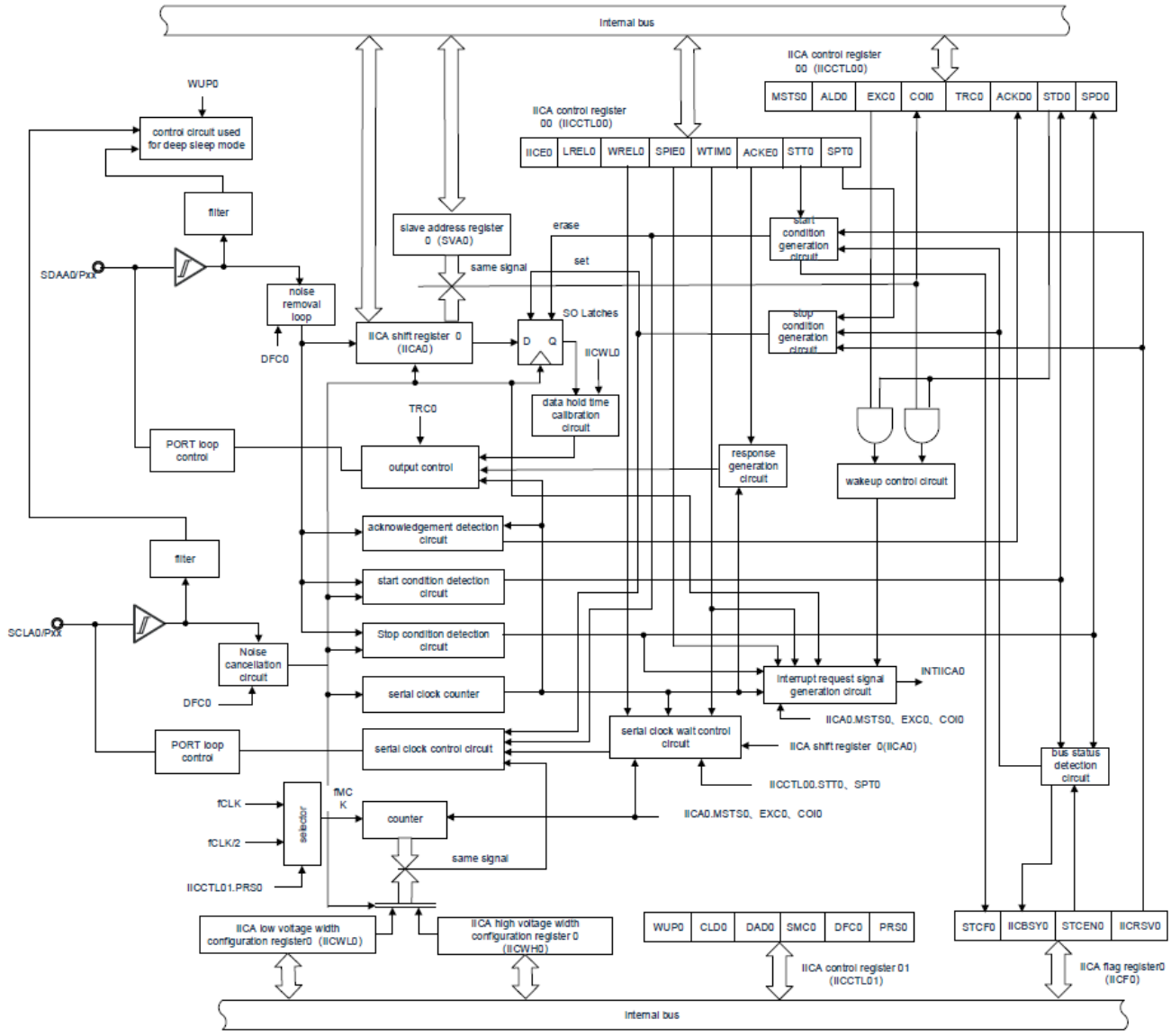
(3) wake-up mode

In a deep sleep mode, when an extension code or a local station address from a master control device is received, the deep sleep mode can be canceled by generating an interrupt request signal (INTIICAn). Set by the WUPn bit of the IICA control register n1 (IICCTLn1).

The block diagram of the serial interface IICA is shown in Figure 20-1.

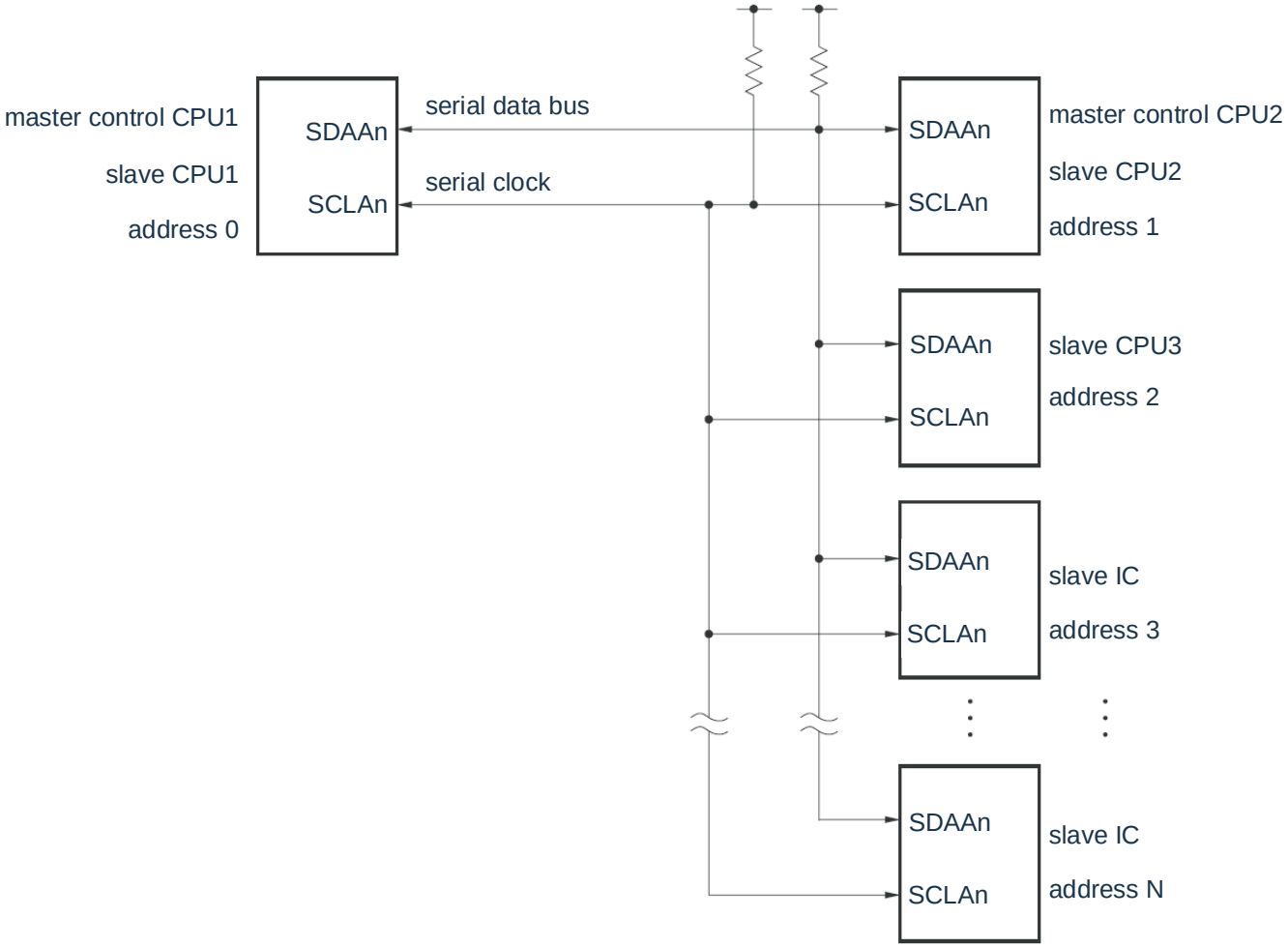
Remark n=0

Figure 20-1 Block diagram of serial interface IICA



Examples of serial bus structures are shown in Figure 20-2.

Figure 20-2 An Example of Serial Bus Structure for I2C Bus



Remark n=0

20.2 Structure of Serial Interface IICA

The serial interface IICA consists of the following hardware.

Table 20-1 Structure of Serial Interface IICA

Project	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 (IICFn) IICA control register n1 (IICCTLn1) IICA low level width set register n (IICWLn) IICA high level width set register n (IICWHn) port mode register x (PMx) Port register x (Px)

Remark n=0

(1) IICA shift register n (IICAn)

The IICAn register is a register 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 lifted by writing the IICAn register to start transferring data. The IICAn register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 20-3 Format of IICAn shift register n(IICAn)

Address: 0x40041B50 After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
ICAn								

Note 1. During data transfer, no data can be written to IICAn registers.

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".
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 operation instruction. However, this register is not allowed to be overwritten when the STDn bit is "1".

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

Figure 20-4 Format of the slave address register n(SVAn)

Location: 0x40041A34 After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
SVAn	A6	A5	A4	A3	A2	A1	A0	0 Notes

Note The bit0 is fixed "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 the sending and receiving process, the counter counts the output or input serial clock to check whether the 8-bit data is sent or received.

(6) interrupt request signal generate circuit

This circuit controls the generation of an interrupt request signal (INTIICAn).

The I2C interrupt request is generate by that following two trigger.

- Drop of the 8th or 9 th serial clock (set by the WTIMn bit)
- Interrupt request (set by SPIEn bit) due to detection of stop condition.

RemarkWTIMn bit: bit3 for IICA control register n0(IICCTLn0)
SPIEn bit :bit4 for IICA control register n0(IICCTLn0)

(7) serial clock control circuit

In the main control mode, the circuit generates a clock output to the SCLAn pin from the sampling clock.

(8) Serial clock waiting control circuit

This circuit controls the waiting sequence.

(9) response generate circuit, stop condition detection circuit, start condition detection circuit, response detection circuit

These circuits generate and detect various states.

(10) data hold time correction circuit

This circuit generates a data hold time for the serial clock drop.

(11) Start condition generating circuit

If the STTn position is "1", the circuit generates a start condition.

However, in a state where scheduled communication is disabled (IICRSVn bit=1) and the bus (IICBSYn bit=1) is not released, STCFn.

(12) Stop condition generating circuit

If the SPTn position is "1", the circuit generates a stop condition.

(13) Bus state detecting circuit

The circuit detects whether the bus is released by detecting a start condition and a stop condition. However, the bus state cannot be detected immediately after operation, so the initial state of the bus state detection circuit must be set through STCENn bits.

Note 1.STTn bit: bit1 for IICA control register n0 (IICCTLn0)
SPTn bit: bit0 for 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

20.3 Register for controlling serial interface IICA

The serial interface, IICA, is controlled through the following 8 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 (PMx)
- Port register (Px)

Remark n=0

20.3.1 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.

To use the serial interface IICAn, you must set bit4 (IICA0EN) to "1".

The PER0 register is set by an 8-bit memory operation instruction.

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

Figure 20-5 Format of Peripheral Admission Register 0 (PER0)

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

symbol	7	6	5	4	3	2	1	0
PER0	RTCEN Note	IRDEN	ADCEN	ICAHN	SCI1EN	SCI0EN	CAN0	TM40EN

IICAnEN	Provides control of input clock of serial interface IICAn
0	Stop provide an input clock. · SFR used by serial interface IICAn cannot be written. · The serial interface IICAn is in a reset state.
1	Allows providing an input clock. · SFR that can read and write to the serial interface IICAn.

Note 1: To set the serial interface IICA, you must first set the following register in the state with the IICAnEN bit "1". When the IICAnEN bit is "0", the control register value for serial interface IICA is the initial value, ignoring writes (except Port Mode Register (PMx) and Port Register (Px).

- 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

20.3.2 IICA control register n0 (IICCTLn0)

This is a register that allows or stops I2C running, sets wait times, and sets other I2C runs.

The IICCTLn0 register is set by an 8-bit memory operation instruction. However, SPIEn bits, WTIMn bits and ACKEn bits must be set when the IICEn bit is '0' or during waiting and IICEn bits can be set simultaneously.

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

Remarks: n=0

Figure 20-6 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	ICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

ICEn	I²C Run Allowed
0	Stop running. Reset the IICA state register n(IICSn) note 1, and stop the internal operation.
1	Allowed to run.
The position '1' must be held in a state where the SCLAn line and the SDAAn line are high level.	
Purge Criteria (IICEn=0)	Placement Criteria (IICEn=1)
· Clear by command. · When reset	· Set by command.

LRELn Note	Exit of communication
0	Run normally
1	Exit the current communication and go into standby. Automatically clears "0" after execution. It is use in that case of receive an extension code independent of the local station, etc. The SCLAn line and the SDAAn line become high resistance. The following flags in the IICA control register n0 (IICCTLn0) and the IICA state register n (IICSn) are cleared: ·STTn·SPTn·MSTSn·EXCn·COIn·TRCn·ACKDn·STDn
The standby state is changed to the standby state of the exiting communication and is maintained until the following communication participation conditions are met. · Starts as a master device after a stop condition is detected. · Address matches or receives an extension code after a start condition is detected.	
Purge Criteria (LRELn=0)	Placement Criteria (LRELn=1)
· Automatically clears after execution. · When reset	· Set by command.

WRELn	Pending Removal
0	The wait is not relieved.
1	Unwait. Automatically clears after the wait is removed.
If the WRELn bit (Unwait) is set during the 9th clock wait in the send state (TRCn=1), the SDAAn line becomes High impedance status (TRCn=0).	
Purge Criteria (WRELn=0)	Placement Criteria (WRELn=1)
· Automatically clears after execution. · When reset	· Set by command.

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).

2. The IICEn bit is not valid in the state of '0'.

3. The read values for the LRELn and WRELn bits are always "0".

Note that if I2C is allowed to run (IICEn=1) when the SCLAn line is high, the SDAAn line is low and the digital filter is ON. At this point, the LRELn position "1" must be continuously passed through the bit memory operation instruction after allowing I2C to run (IICEn=1).

Remark n=0

Figure 20-6 Format of IICA control register n0 (IICCTLn0) (2/4)

SPIEn Note 1	Interrupt request generated by allow or disabling stop condition detection
0	prohibition
1	Allow
The IICA control register n1(IICCTLn1) does not generate a stop condition interrupt even if the SPIEn position '1' is set.	
Purge Criteria (SPIEn=0)	Placement Criteria (SPIEn=1)
· Clear by command. · When reset	· Set by command.

WTIMn Note 1	Control of wait and interrupt requests
0	An interrupt request signal is generated at the descending edge of the eighth clock. Master device: After outputting 8 clocks, the clock output is set to a low level to wait. Slave: After 8 clocks are input, set the clock to a low level and wait for the master device.
1	An interrupt request signal is generated at the descending edge of the ninth clock. Master device: After outputting 9 clocks, the clock output is set to a low level to wait. Slave: After 9 clocks are input, set the clock to a low level and wait for the master device.
interrupt is generate at that descending edge of the 9th clock, regardless of the setting of this bit during address transfer; After the address transfer is completed, this bit is set to Effect. The master device enters a waiting state along the 9th clock descent during address transfer. The slave device receiving the local station address is responding A ninth clock down edge after (ACK) enters a waiting state, but a slave device receiving an extension code	
Purge Criteria (WTIMn=0)	Placement Criteria (WTIMn=1)
· Clear by command. · When reset	· Set by command.

ACKEn Notes 1, 2	response control
0	No replies.
1	Allow replies. The SDAAn line is set to a low level during the 9th clock.
Purge Criteria (ACKEn=0)	Placement Criteria (ACKEn=1)
· Clear by command. · When reset	· Set by command.

Note 1. The bit has an invalid signal in the state with the IICEn bit "0". This bit must be set during this time.

2. The setting value is invalid if the address is not an expander during address transfer. When the device is a slave and the address matches, a response is generated regardless of the set value.

Remark n=0

Figure 20-6 Format of IICA control register n0 (IICCTLn0) (3/4)

STTn Notes 1, 2	Trigger of Start Condition
0	Do not generate a start condition.
1	When the bus is released (standby, IICBSYn bit is "0"): If this position is "1", a start condition is generated (as the start of the master device). When a third party is communicating: · Circumstances in which communication reservation is allowed (IICRSVn=0) Use as start condition reservation flag. If this position is '1', the start condition is automatically generated after the bus is released. · IICRSVn=1, where communication reservation is prohibited. Even if this position is "1", the STTn bit is cleared and the STTn clear flag (STCFn) is set "1" without generating a start condition. Wait State (Master): A restart condition is generated after the wait is removed.

Precautions for Placement Timing:

· Master Receive: Prohibit this location "1" during the transfer. The ACKEn location "1" can only be used during a wait period when the slave has been notified to receive the "0".

· Master Send: During an acknowledgement, the start condition may not be generated properly. This position "1" must be used during the wait period after the ninth clock output.

· Disable from setting "1" at the same time as the trigger for the stop condition (SPTn

· After STTn position "1" is placed, it is prohibited to place "1" again before the purge condition is satisfied.

Purge Criteria (STTn=0)	Placement Criteria (STTn=1)
· Place the STTn position "1" in a state where communication reservation is disabled. · When arbitration fails · Master device generation start condition. · Purge due to LRELn bit '1' (Exit Communication) · When the IICEn bit is "0" · When reset	· Set by command.

Note 1. The bit has an invalid signal in the state with the IICEn bit "0".

2.The read value of the STTn bit is always "0".

Note 1. If you read bit1 (STTn) after setting the data, this bit becomes "0".

2.IICRSVn :bit0 for IICA flag register n(IICFn)

STCFn :bit7 for IICA flag register n(IICFn)

3.n=0

Figure 20-6 Format of IICA control register n0 (IICCTLn0) (4/4)

SPTn Note	Trigger of stop condition
0	No stop condition is generated.
1	Generate a stop condition (end of transfer as master device).

Precautions for Placement Timing:

· Master Receive: Prohibit this location "1" during the transfer. The ACKEn location "1" can only be used during a wait period when the slave has been notified to receive the "0".

· Master Send: During an acknowledgement, a stop condition may not be generated properly. This position "1" must be used during the wait period after the ninth clock output.

· Prevent "1" from being set at the same time as the trigger of the start condition (STTn

· The SPTn location "1" can only be used if the host device.

· When the WTIMn bit is "0", you must note: If the SPTn position "1" is taken during the wait period after 8 clock output, a stop condition is generated during the high level of the 9th clock after the wait. The WTIMn bit must be set from '0' to '1' during the wait period after output of the ninth clock to '1.

· After SPTn position "1" is taken, it is forbidden to take this position "1" again before the purge condition is satisfied.

Purge Criteria (SPTn=0)	Placement Criteria (SPTn=1)
· When arbitration fails · Automatically clears when a stop condition is detected. · Purge due to LRELn bit '1' (Exit Communication) · When the IICEn bit is "0" · When reset	· Set by command.

Note The read value of the SPTn bit is always "0".

Note that if the bit3 (TRCn) of the IICA state register n (IICSn) is "1" when the bit5 (WRELn) of the IICCTLn0 register is set to "1" at the ninth clock, the SDAAn line is set to a high impedance after clearing the TRCn bit (received state). The wait release when the TRCn bit is '1' (sending state) must be performed by writing IICA shift register n.

Remark n=0

20.3.3 IICA status register n (IICSn)

This is a register that represents the I2C status.

The 8-bit memory operation instruction can read the IICSn register only if the STTn bit is "1" and the wait period. After the reset signal is generated, the value of this register changes to "00H".

Note that the IICSn register is disabled from being read in the WUPn=1 state allowed in deep sleep mode. With WUPn bit '1', regardless of INTIICAn interrupt request, if WUPn bit is changed from '1' to '0', the change in state is not reflected until the next start or stop condition is detected. Therefore, to use wake-up functionality, you must allow (SPIEn=1) interrupts due to the detection of a stop condition and read the IICSn register after the interrupt is detected.

Remark STTn :bit1 for IICA control register n0(IICCTLn0)

WOPn :bit7 for IICA control register n1(IICCTLn1)

Figure 20-7 Format of IICA state register n(IICSn) (1/3)

Address: 0x40041B51 After reset: 00H R

symbol	7	6	5	4	3	2	1	0
IICSn	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

MSTSn	Acknowledgement flag for master status
0	Slave or Communication Standby
1	master communication status

Purge Criteria (MSTSn=0)	Placement Criteria (MSTSn=1)
· When a stop condition is detected · When the ALDn bit is "1" · Purge due to LRELn bit '1' (Exit Communication) · When the IICEn bit changes from "1" to "0" (stop running) · When reset	· When generating start conditions

ALDn	Detection of Arbitration Failure
0	indicating that no arbitration has occurred or the arbitration has been won.
1	Indicates arbitration failure. Clear MSTSn bits.

Purge Criteria (ALDn=0)	Placement Criteria (ALDn=1)
· Automatically clear note after reading the IICSn register. · When the IICEn bit changes from "1" to "0" (stop running)	· When arbitration fails

Note This bit is cleared even if a bit memory operation instruction is executed for a bit other than the IICSn register. Therefore, when using ALDn bits, you must read the data for the ALDn bits before reading the other bits.

Note 1.LRELn :bit6 for IICA control register n0(IICCTLn0)

ICEn :bit7 for IICA control register n0(IICCTLn0)

2.n=0

Figure 20-7 Format of IICA state register n(IICSn) (2/3)

EXCn	Receiving Detection of Spreading Codes
0	The extension code was not received.
1	An extension code was received.

Purge Criteria (EXCn=0)	Placement Criteria (EXCn=1)
· When a start condition is detected · When a stop condition is detected · Purge due to LRELn bit '1' (Exit Communication) · When the IICEn bit changes from "1" to "0" (stop running) · When reset	· When the 4-bit height of the received address data is "0000" or "1111" (Rising edge position at 8th clock)

COIn	Detection of address matching
0	Different address.
1	Same address.

Purge Criteria (COIn=0)	Placement Criteria (COIn=1)
· When a start condition is detected · When a stop condition is detected · Purge due to LRELn bit '1' (Exit Communication) · When the IICEn bit changes from "1" to "0" (stop running) · When reset	· When receiving an address and a local station address (slave address register n (SVAn) Same (Ascend position at 8th clock)

TRCn	Send/receive status detection
0	In receive state except send state. The SDAAn line is placed at a high impedance.
1	Is in a send state. Set to output the value of the SOn latch to the SDAAn line (valid after the descent of the ninth clock in byte 1).

Purge Criteria (TRCn=0)	Placement Criteria (TRCn=1)
<master and slave devices> · When a stop condition is detected · Purge due to LRELn bit '1' (Exit Communication) · When the IICEn bit changes from "1" to "0" (stop · Clear note due to WRELn bit of '1' · When the ALDn bit changes from "0" to "1" (Arbitration Failure) · When reset · Non-participation in communications (MSTSn, <Master device> · When the LSB (transmission direction indicator bit) <Slave> · When a start condition is detected · When the LSB (transmission direction indicator bit)	<Master device> · When generating start conditions · When the LSB (Direction of Transmission Bit) of the When exporting "0" (Master Send) <Slave> · When the host device's first byte (address transfer) LSB direction indicator bit) when you enter "1" (slave send)

Note When the bit3 (TRCn) of the IICA status register n (IICSn) is "1", if the bit5 (WRELn) of the IICA control register n0 (IICCTLn0) is set to "1" at the ninth clock, the SDAAn line is set to high impedance after the TRCn bit (reception state) is cleared. The wait release when the TRCn bit is '1' (sending state) must be performed by writing IICA shift register n.

Note 1.LRELn :bit6 for IICA control register n0(IICCTLn0)
ICEn :bit7 for IICA control register n0(IICCTLn0)
2.n=0

Figure 20-7 Format of IICA state register n(IICSn) (3/3)

ACKDn	Detection of Acknowledgements (ACKs)
0	No replies detected.
1	An acknowledgement was detected.

Purge Criteria (ACKDn=0)	Placement Criteria (ACKDn=1)
· When a stop condition is detected · When the first clock of the next byte goes up · Purge due to LRELn bit '1' (Exit Communication) · When the IICEn bit changes from "1" to "0" (stop running) · When reset	· When the SCLAn line is set to a low level by the 9th clock rising edge of the SDAAn line

STDn	Detection of Start Conditions
0	No start condition detected.
1	A start condition was detected, indicating that it is during address transfer.

Purge Criteria (STDn=0)	Placement Criteria (STDn=1)
· When a stop condition is detected · When the first clock of the next five-minute section after address transfer rises · Purge due to LRELn bit '1' (Exit Communication) · When the IICEn bit changes from "1" to "0" (stop running)	· When a start condition is detected

SPDn	Detection of Stop Conditions
0	No stop condition detected.
1	A stop condition is detected, the master device ends communication and the bus is released.

Purge Criteria (SPDn=0)	Placement Criteria (SPDn=1)
· When the first clock of the address transfer byte rises after the start condition is detected after this bit · When the WUPn bit changes from "1" to "0 · When the IICEn bit changes from "1" to "0" (stop running)	· When a stop condition is detected

Note 1.LRELn :bit6 for IICA control register n0(IICCTLn0)

ICEn :bit7 for IICA control register n0(IICCTLn0)

2.n=0

20.3.4 IICA flag register n (IICFn)

This is a register that sets the I2C run mode and represents the I2C bus status.

The IICFn register is set by an 8-bit memory operation instruction. However, only the STTn clear flag (STCFn) and I2C bus status flag (IICBSYn) can be read.

The communication reservation function is allowed or disabled by the IICRSVn bit setting, and the initial value of the IICBSYn bit is set by the STCENn bit. IICRSVn bits and STCENn bits can only be written if I2C is disabled (bit7(IICEn)=0 for IICA control register n0(IICCTLn0). Only IICFn registers can be read after a run is allowed. After the reset signal is generated, the value of this register changes to "00H".

Figure 20-8 Format of IICA flag register n(IICFn)

Address: 0x40041B52 After reset: 00H R/W Notes

symbol	7	6	5	4	3	2	1	0
IICFn	STCFn	ICBSYn	0	0	0	0	STCENn	IICRSVn

STCFn	STTn clear flag
0	Release Start Condition.
1	The STTn flag cannot be cleared by issuing a start condition.

Purge Criteria (STCFn=0)	Placement Criteria (STCFn=1)
· Purge due to STTn bit '1' · When the IICEn bit is "0" · When reset	· When the STTn bit is cleared to "0" because the start condition cannot be issued in a state set to disable IICRSVn=1

ICBSYn	I²C bus status flag
0	Bus Release Status (Initial Communication Status at STCENn=1)
1	Bus Communication Status (Initial Communication Status at STCENn=0)

Purge Criteria (IICBSYn=0)	Placement Criteria (IICBSYn=1)
· When a stop condition is detected · When the IICEn bit is "0" · When reset	· When a start condition is detected • Placement of IICEn Bits with STCENn Bit '0'

STCENn	Initial Allow Trigger
0	After allowing run (IICEn=1), the start condition is allowed to be generated by detecting the stop condition.
1	After allowing run (IICEn=1), the start condition is allowed to be generated without detecting the stop condition.

Purge Criteria (STCENn=0)	Placement Criteria (STCENn=1)
· Clear by command. · When a start condition is detected · When reset	· Set by command.

IICRSVn	communication reservation function prohibition bit
0	Allow communication appointments.
1	No communication appointments.

Purge Criteria (IICRSVn=0)	Placement Criteria (IICRSVn=1)
· Clear by command. · When reset	· Set by command.

Note bit6 and bit7 are read-only bits.

Note 1. STCENn bits can only be written when the (IICEn=0) is stopped.

2. If the STCENn bit is "1", the bus is considered as IICBSYn=0 regardless of the actual bus state, so to avoid breaking other traffic when issuing the first start condition (STTn=1), it is necessary to confirm that there is no third party communicating.

3. Write the IICRSVn only when (IICEn=0).

Note 1.STTn: bit1 for IICA control register n0 (IICCTLn0)

2.IICEn:bit7 for the IICA control register n0 (IICCTLn0)

20.3.5 IICA control register n1 (IICCTLn1)

This is a register used to set the I2C run mode and detect the status of the SCLAn and SDAAn pins.

The IICCTLn1 register is set by an 8-bit memory operation instruction. However, only CLDn and DADn bits can be read.

In addition to the WUPn bit, the IICCTLn1 register must be set when I2C is disabled (bit7(IICEn)=0 for IICA control register n0(IICCTLn0).

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

Figure 20-9 Format of IICA control register n1 (IICCTLn1) (1/2)

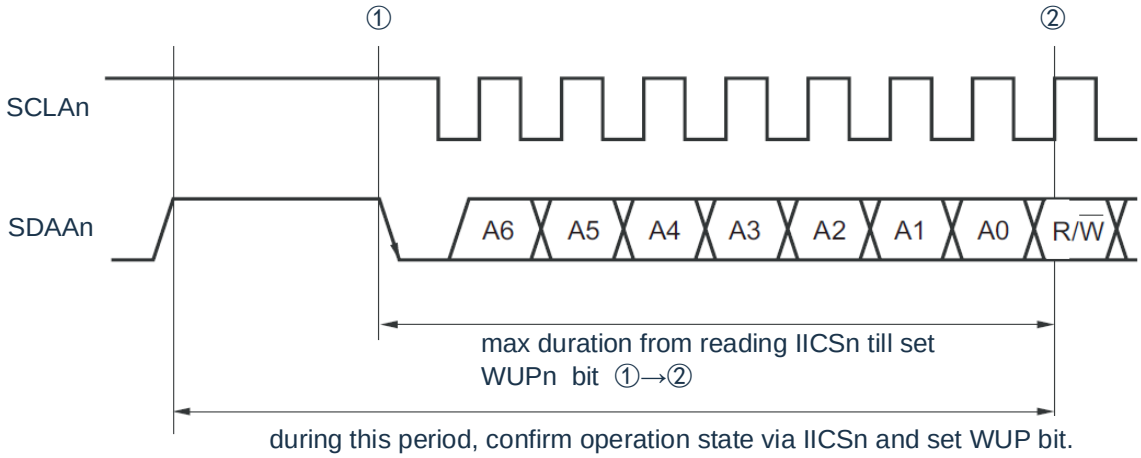
Location: 0x40041A31 After reset: 00H R/W Note 1

symbol	7	6	5	4	3	2	1	0
IICCTLn1	WOPn	0	CLDn	DADn	SMCn	DFCn	0	PRSn

WOPn	Control of address matching wake-up
0	In deep sleep mode, stop the address matching wake-up function from running.
1	In deep sleep mode, address matching wake-up functionality is allowed to run.
To transfer WUPn position "1" to deep sleep mode, you must pass at least 3 fMCK clocks after WUPn position "1" and execute deep sleep instructions (Fig. The WUPn bit must be cleared '0' after the address match or the extension is received. Can participate in subsequent communications by clearing WUPn bit '0' (requires unwaiting and writing to send data after clearing WUPn bit '0'). In the WUPn bit'1' state, the interrupt time when the address matches or receives the expansion code is the same as the WUPn bit'0 (The delay difference of the sampling error depending on the clock). In addition, when that WUPn bit is" 1", the stop condition interrupt does not occur even SPIEn position" 1".	
Purge Criteria (WUPn=0)	Placement Criteria (WUPn=1)
· Clear through instructions (after address matching or receive an extension code).	· Note 2 by instruction placement (MSTSn=0, EXCn=0, COIn=0, and STDn=0 (not participating in communication)

Note that 1.bit4 and bit5 are read-only bits.

2. During the period shown below, the IICA status register n (IICSn) needs to be acknowledged and placed.



Remark n=0

Figure 20-9 Format of IICA control register n1 (IICCTLn1) (2/2)

CLDn	SCLAn pin level detection (only valid if IICEn bit is "1")
0	The SCLAn pin is detected as low.
1	High SCLAn pin level detected.

Purge Criteria (CLDn=0)	Placement Criteria (CLDn=1)
· When the SCLAn pin is low · When the IICEn bit is "0" · When reset	· When the SCLAn pin is high level

DADn	SDAAn pin level detection (only valid if IICEn bit is "1")
0	The SDAAn pin is detected as low.
1	High SDAAn pin level detected.

Purge Criteria (DADn=0)	Placement Criteria (DADn=1)
· When the SDAAn pin is low · When the IICEn bit is "0" · When reset	· When the SDAAn pin is high level

SMCn	Switch of Operation mode
0	Run in standard mode (maximum transfer rate: 100kbps).
1	Run in Fast Mode (Maximum Transfer Rate: 400kbps) or Enhanced Fast Mode (Maximum Transfer Rate: 1Mbps).

DFCn	Operation Control of Digital Filter
0	digital filter OFF
1	Digital Filter ON
Digital filter must be used in fast mode or enhanced fast mode. A digital filter is used to eliminate noise. The transfer clock does not change whether the DFCn position is "1" or "0" clear.	

PRSn	Control of the runtime clock (fMCK)
0	Select fCLK (1MHz≤fCLK≤20MHz).
1	Select fCLK/2 (20MHz<fCLK).

Note 1. The maximum operating frequency of the $_{IICA}$ runtime clock (fMCK) is 20MHz (Max.) The bit0 (PRSn) of the IICA control register n1 (IICCTLn1) must be set to "1" only if fCLK exceeds 20 MHz.

2. In the case of setting the transfer clock, you must note the minimum operating frequency of the fCLK. The fCLK minimum operating frequency of the serial interface IICA depends on the operation mode.

Quick mode: $_{fCLK}$=3.5MHz (Min.)
Enhanced Fast Mode: fCLK=$_{10MHz}$(Min.)
Standard Mode: $_{fCLK}$=1MHz (Min.)

Note 1.IICEn: bit7 for IICA control register n0 (IICCTLn0)
2.n=0

20.3.6 IICA Low level width setting register n (IICWLn)

This register controls the SCLAn pin signal low level width (tLOW) and the SDAAn pin signal output by the serial interface IICA.

The IICWLn register is set by an 8-bit memory operation instruction.

IICWLn register must be set when I2C is disabled (bit7(IICEn)=0 for IICA control register n0(IICCTLn0).

After the reset signal is generated, the value of this register changes to FFH.

For IICWLn register set-up methods, refer to "20.4.2 Setting the Transfer Clock Method Through the IICWLn Register and IICWHn Register".

The data retention time is 1/4 of the time set by IICWLn.

Figure 20-10 Format of IICA low level width setting register n(IICWLn)

Location: 0x40041A32 After Reset: FFH R/W

symbol	7	6	5	4	3	2	1	0
IICWLn								

20.3.7 IICA High Level Width Setting Register n (IICWHn)

This register controls the SCLAn pin signal high level width and the SDAAn pin signal output by the serial interface IICA.

The IICWHn register is set by an 8-bit memory operation instruction.

IICWHn register must be set when I2C is disabled (bit7(IICEn)=0 for IICA control register n0(IICCTLn0).

After the reset signal is generated, the value of this register changes to FFH.

Figure 20-11 Format of IICA high level width setting register n(IICWHn)

Location: 0x40041A33 After Reset: FFH R/W

symbol	7	6	5	4	3	2	1	0
ICWHn								

Note 1. Refer to 20.4.2 (1) for the setting method of the master transmission clock; Refer to 20.4.2(2) for the dependent IICWLn register and IICWHn register set-up methods.

2.n=0

20.3.8 Port mode register x (PMx)

This register sets the input/output of the port.

Port mode register PMx and port output latch Px must be set to '0' when the Pxx/SCLA0 pin is used as clock input/output and Pxx/SDAA0 pin.

When the IICEn bit (bit7 of IICA control register n0 (IICCTLn0) is "0", the Pxx/SCLA0 pin and Pxx/SDAA0 pin are low level.

The PMx register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to FFH.

For details on port mode registers, refer to "2.3.1 Port Mode Registers (PMxx)".

20.4 Functions of I2C bus mode

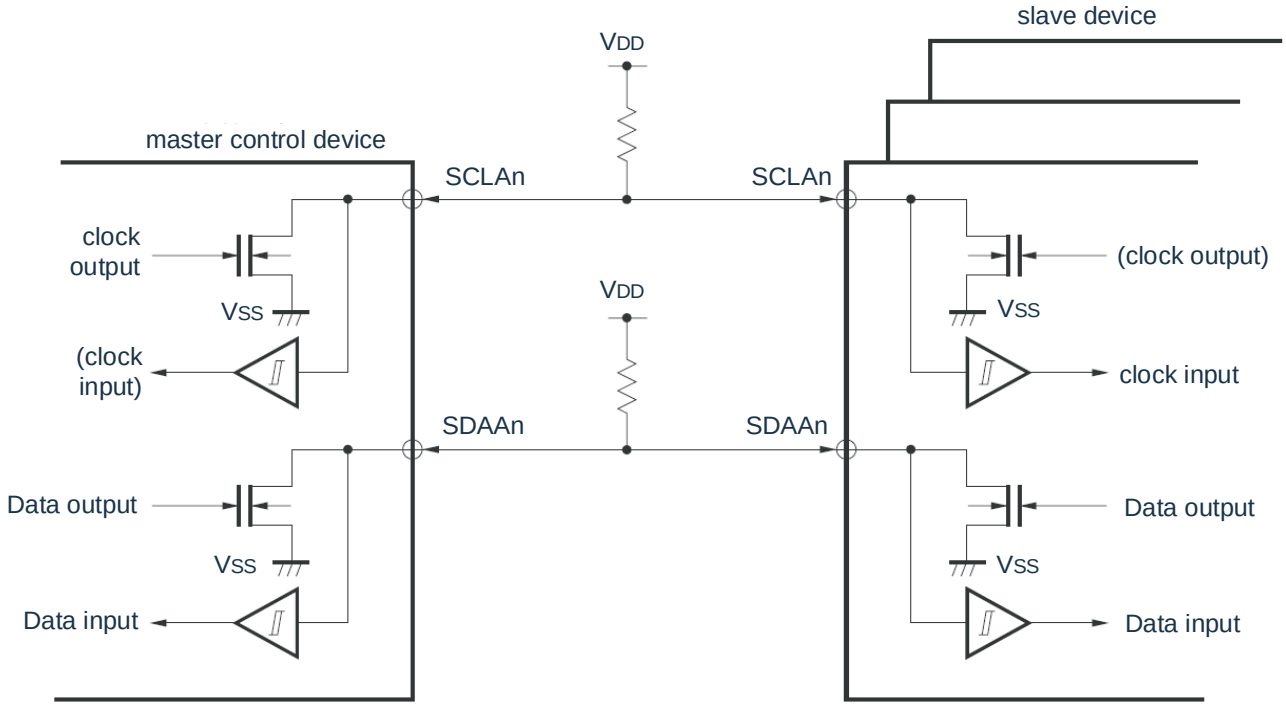
20.4.1 pin structure

The serial clock pin (SCLAn) and the serial data bus pin (SDAAn) are structured as follows.

(1) SCLAn.....Input/output pin for serial clock
The output of the master device and the slave device are N-channel drain open circuit output, and the input is Schmidt input.
(2) SDAAn.....Input/output multiplexing pin for serial data
The output of the master device and the slave device are N-channel drain open circuit output, and the input is Schmidt input.

Since the output of the serial clock line and the serial data bus is an open-circuit output of the N-channel drain, an external pull-up resistor is required.

Figure 20-13 Pin Diagram



Remark n=0

20.4.2 Method for setting transmission clock through IICWLn register and IICWHn register

(1) Setting method for main control party transmitting clock

$$\text{Transfer Clock}=\frac{f_{MCK}}{IICWL+IICWH+f_{MCK}(t_R+t_F)}$$

At this point, the best values for the IICWLn register and the IICWHn register are as follows:
(Decimal portion of all set values rounded)

· fast mode

$$IICWLn=\frac{0.52}{\text{transmission clock}}\times f_{MCK}$$

$$IICWHn=(\frac{0.48}{\text{transmission clock}}-t_R-t_F)\times f_{MCK}$$

· standard mode

$$IICWLn=\frac{0.47}{\text{transmission clock}}\times f_{MCK}$$

$$IICWHn=(\frac{0.53}{\text{transmission clock}}-t_R-t_F)\times f_{MCK}$$

· Enhanced Fast Mode

$$IICWLn=\frac{0.50}{\text{transmission clock}}\times f_{MCK}$$

$$IICWHn=(\frac{0.50}{\text{transmission clock}}-t_R-t_F)\times f_{MCK}$$

(2) Slave IICWLn register and IICWHn register setting method
(Decimal portion of all set values rounded)

· fast mode

IICWLn=1.3 s fMCK
IICWHn=(1.2 s-tR-tF) MCK

· standard mode

IICWLn=4.7 s fMCK
IICWHn=(5.3 s-tR-tF) MCK

· Enhanced Fast Mode

IICWLn=0.50 s MCK
IICWHn=(0.50 s-tR-tF) MCK

Note 1. The maximum operating frequency of the IICA runtime clock (fMCK) is 20MHz (Max.) The bit0 (PRSn) of the IICA control register n1 (IICCTLn1) must be set to "1" only if fCLK exceeds 20 MHz.

2. In the case of setting the transfer clock, you must note the minimum operating frequency of the fCLK. The fCLK minimum operating frequency of the serial interface IICA depends on the operation mode.

fast mode : f_{CLK}=3.5MHz (Min.)
Enhanced Fast Mode: fCLK=$_{10MHz}$(Min.)
standard mode : f_{CLK}=1MHz (Min.)

Note 1. Since the rising time (tR) and falling time (tF) of the SDAAn signal and SCLAn signal differ by pull up resistance and wiring capacitance.

2.IICWLn :IICA low level width set-up register n

ICWHn :IICA high level width set-up register n

tF Down time of SDAAn and SCLAn signals

tR rise time of SDAAn and SCLAn signals

fMCK : IICA Run Clock Frequency

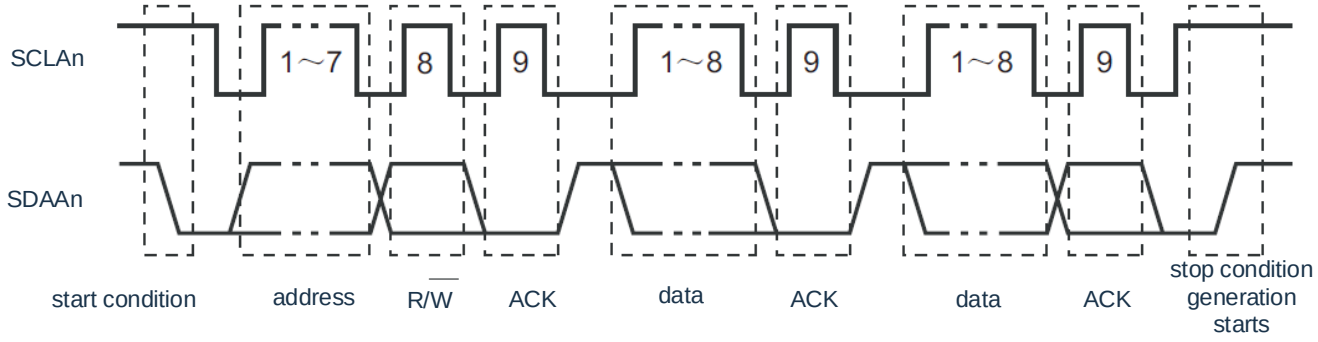
3.n=0

20.5 Definition and control method of I2C bus

The following describes the serial data communication format and the signal used for the I2C bus.

The Start Condition, Address, Data, and Stop Condition generated on the I2C bus serial data bus are shown in Fig. 20-14.

Figure 20-14 Serial Data Transfer Timing of I2C Bus



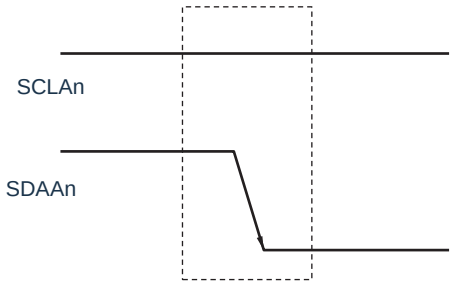
The master generates a start condition, a slave address, and a stop condition.

Both the master device and the slave device can generate an acknowledgement (ACK) (in general, the receiver outputs 8 bits). The master device continuously outputs a serial clock (SCLAn). However, the slave device can extend the low level period of the SCLAn pin and insert the wait.

20.5.1 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 condition is the signal generated when the master device starts serial transfer to the slave device. When used as a slave, a start condition is detected.

Figure 20-15 Start Condition



The start condition is output if the IICA control register n0 (bit0=1 of IICCTLn0) is set to '1'. If a start condition is detected, set the bit1 (STDn) of the IICSn register to "1".

Remark n=0

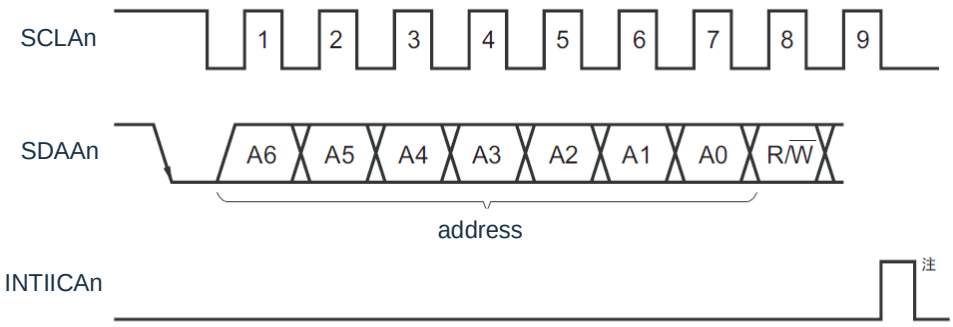
20.5.2 Address

Subsequent 7-bit data for the start condition is defined as an address.

The address is 7 bits of data output by the master control device for selecting a specific slave device from among a plurality of slave devices connected to 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 20-16 Address



Note If you receive data other than the local station address or the expansion code at the slave run-time, no INTIICAn is generated.

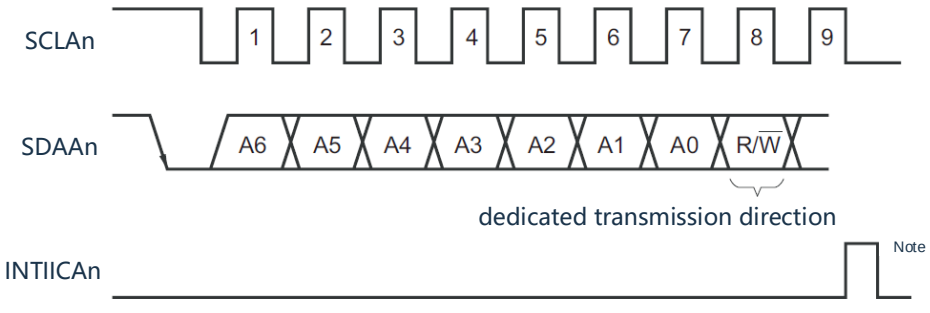
If 8 bits of data constituting the delivery direction described in 20.5.3 Assignment of Delivery Direction are written to IICA shift register n(IICAn). The received address is written to the IICAn register. The slave addresses are assigned to the IICAn register as high as 7 bits.

20.5.3 Assignment of transmission Direction

The master device transmits 1 bits of data specifying the transmission direction after the 7-bit address.

When the designated bit of the transmission direction is '0', the main control device transmits data to the slave device; When the specified bit of this transfer direction is "1", the master receives data from the slave.

Figure 20-17 Assignment of Routing Direction



Note If you receive data other than the local station address or the expansion code at the slave run-time, no INTIICAn is generated.

Remark n=0

20.5.4 Acknowledge (ACK)

The serial data status of the sender and receiver can be confirmed by an acknowledgement (ACK). The receiver returns a reply each time it receives 8 bits of data.

Typically, the sender receives a reply after sending 8-bit data. When the receiver returns a reply, it is considered that the reply has been received normally and the processing is continued. Detection of acknowledgements that can be confirmed by bit2(ACKDn) of the IICA status register n(IICSn). When the master control device receives the last data in the receiving state, the stop condition is generated without returning the response. When a response is not returned after receiving data from a secondary device, the main control device outputs a stop condition or a restart condition to abort transmission. The reason for not returning an acknowledgement is as follows:

(1) No normal reception.

(2) The receipt of final data has been ended.

(3) There is no receiver specified by address.

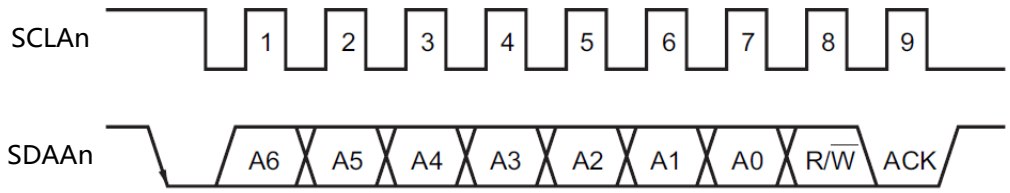
The receiver sets the SDAAn line at a low level at the 9th clock to generate a response (normal reception).

By setting the bit2(ACKEn) of IICA control register n0(IICCTLn0) to '1', the response is automatically generated. The bit3 (TRCn) of the IICSn register is set by the 8-bit data following the 7-bit address information. In the case of receiving (TRCn=0), the ACKEn location "1" is

When a dependent receive run (TRCn=0) cannot receive data or does not need the next data, the ACKEn bit must be cleared to "0".

When the next data is not needed in the main receive run (TRCn=0), the ACKEn bit must be cleared to "0".

Figure 20-18 Acknowledgement



When the address of the local station is received, the response is generated automatically regardless of the value of the ACKEn bit. When an address of a non-local station is received, no reply is generated (NACK).

When an expansion code is received, a response is generated by pre-positioning the ACKEn position '1'. The response generation method when receiving data varies depending on the wait time setting as shown below.

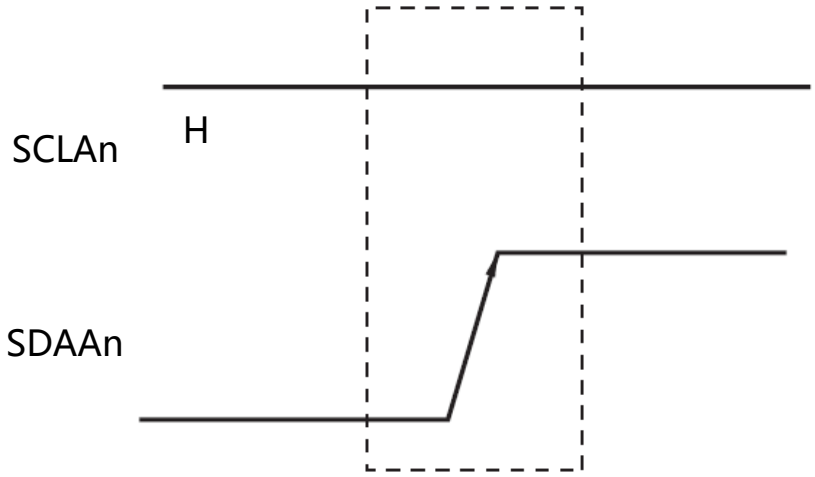
- When 8 clock waits are selected (bit3(WTIMn)=0 for the IICCTLn0 register): A response is generated in synchronization with the eighth clock descent of the SCLAn pin by placing the ACKEn position "1" before unwaiting.
- When 9 clock waits are selected (bit3(WTIMn)=1 for the IICCTLn0 register): Generate a response by pre-positioning the ACKEn position "1".

Remarks: n=0

20.5.5 stop condition

When the SCLAn pin is a high level, a stop condition is generated if the SDAAn pin changes from a low level to a high level. The stop condition is a signal generated when the master device finishes serial transfer to the slave device. A stop condition is detected when used as a slave device.

Figure 20-19 stop condition



A stop condition is generated if the bit0(SPTn) of the IICA control register n0(IICCTLn0) is set to '1'. If a stop condition is detected, the bit0 (SPDn) of IICA status register n (IICSn) is set "1" and a INTIICAn is generated when the bit4 (SPIEn) of the IICCTLn0 register is "1.

Remarks: n=0

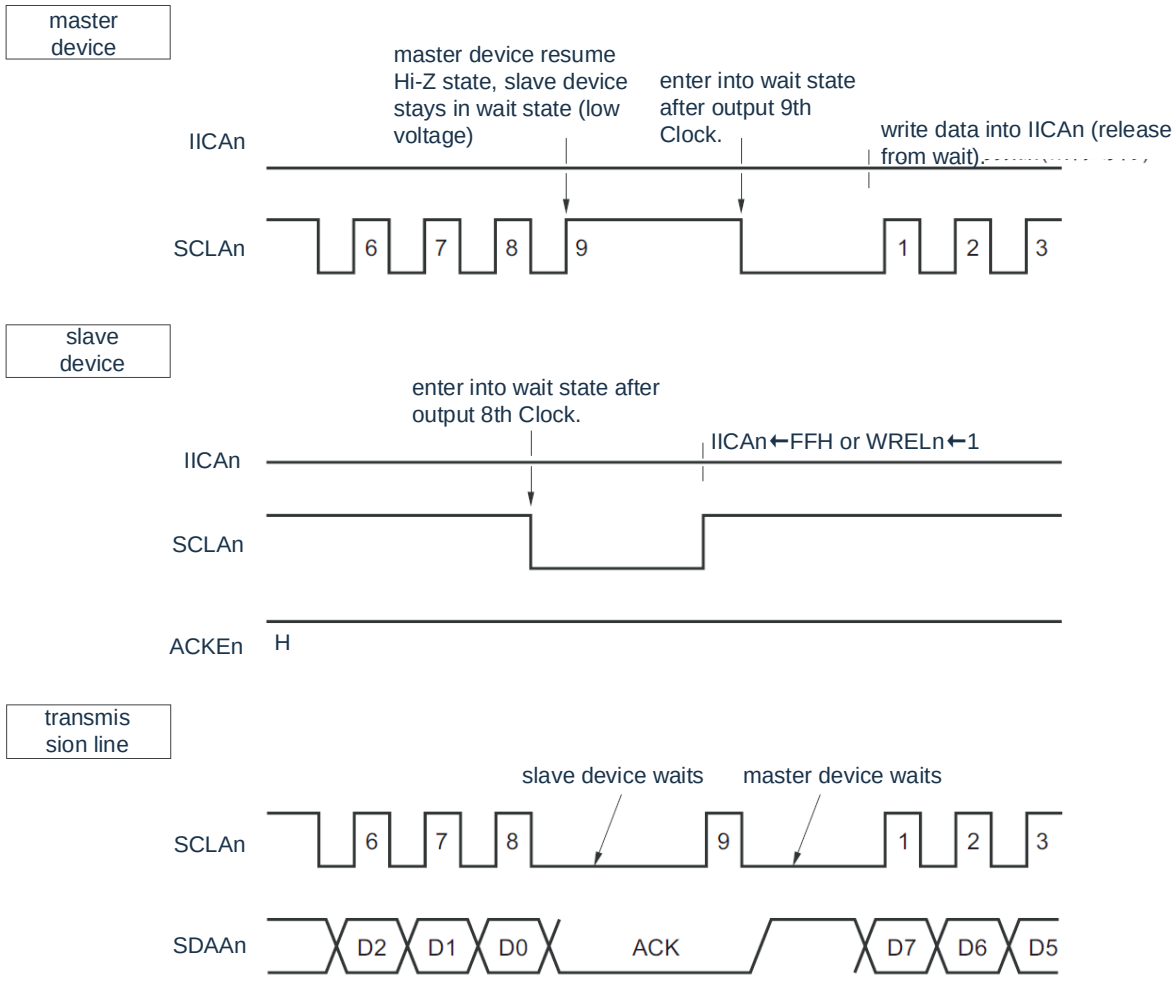
20.5.6 waiting

Notify the other master or slave to prepare the sending/receiving of data by waiting (waiting state).

Notify the other party that it is waiting by placing the SCLAn pin at a low level. If both the master device and the slave device wait states are released, the next transfer can begin.

Figure 20-20 Waiting (1/2)

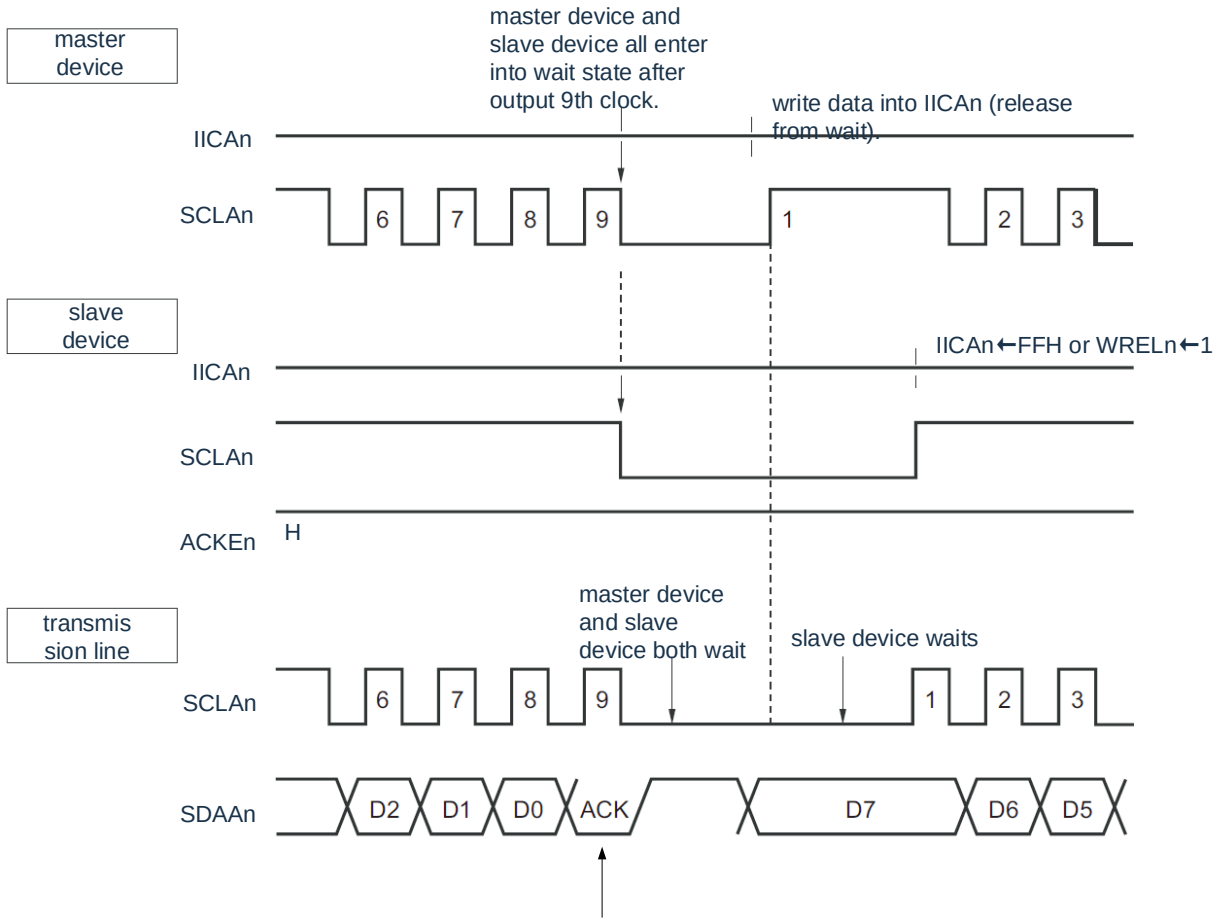
(1) Case where the master device waits for 9 clocks and the slave device waits for 8 clocks (Main control equipment: Send, slave: Receive, ACKEn=1)



Remark n=0

Figure 20-20 Waiting (2/2)

(2) When the master and slave are waiting for 9 clocks
(Main control equipment: Send, slave: Receive, ACKEn=1)



Remark ACKEn:bit2 for the IICA control register n0 (IICCTLn0)
WRELn:bit5 for the 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) At the sender, if data is written to the IICAn register, the wait is canceled. The master device also removes the wait by:

- Set the bit1 (STTn) of the IICCTLn0 register to 1.
- Set the bit0 (SPTn) of the IICCTLn0 register to 1.

Remark n=0

20.5.7 Waitinging Release Method

In general, that I²C can relieve the wait by the following treatment.

- Write data to IICA shift register n(IICAn).
- Set the bit5(WRELn) of the IICA control register n0(IICCTLn0) to Release Waiting.
- Place the bit1 (STTn) of the IICCTLn0 register (Generate Start Condition) Note.
- Place the bit0 (SPTn) of the IICCTLn0 register (generates a stop condition) Note.

Note: Limited to master devices.

If these waiting undoing processes are performed, I²C unwaits and restarts communication. To send data (including addresses) after the unwait, you must write data to the IICAn register.

To receive data after unwaiting or end sending data, the bit5 (WRELn) of the IICCTLn0 register must be set "1". To generate a restart condition after unwaiting, the bit1(STTn) of the IICCTLn0 register must be set "1". To generate a stop condition after unwaiting, the bit0(SPTn) of the IICCTLn0 register must be set "1". You can perform only one unprocess on a single wait.

For example, if data is written to the IICAn register after the WRELn position is removed by "1", the timing of changes to the SDAAn line may conflict with the IICAn register. In addition to these proces, in that case of interrupt communication, if IICEn bit is clear" 0", communication is stopped and waiting is relieved. In the case of I2C bus state being locked due to noise, if bit6 (LRELn) of IICCTLn0 register is set to "1", the communication is withdrawn, thus the waiting is relieved.

Note: If a wait undoing process is performed when the WUPn bit is '1', the wait is not unwound.

Remarks: n=0

20.5.8 Generation Timing and Waiting Control of Interrupt Request (INTIICAn)

By setting a bit3 (WTIMn) of the IICA control register n0 (IICCTLn0), INTIICAn is generated in the timing shown in Table 20-2 and is subjected to wait control.

Table 20-2 Generation Timing and Waiting Control of INTIICAn

WTIMn	slave operation			master run		
	Address	data receiving	data transmission	Address	data receiving	data transmission
0	9 Note 1, 2	8 Note 2	8 Note 2	9	8	8
1	9 Note 1, 2	9 Note 2	9 Note 2	9	9	9

Note 1. The slave device generates a INTIICAn signal and enters a waiting state at the descent edge of the 9th clock only if the received address and the set address of the slave address register n (SVAn) are identical.

At this point, a reply is generated regardless of the setting of the bit2 (ACKEn) of the IICCTLn0 register. The slave device receiving the spreading code generates a INTIICAn at the descending edge of the 8th clock. If that address is different aft the restart, INTIICAn is generate at the descending edge of the 9th clock, but does not enter the wait state.

2. If the received address and the content of the slave address register n (SVAn) are different and no extension code is received, INTIICAn is not generated.

Note: The numbers in the table represent the number of clocks for the serial clock. Both interrupt request and wait control are synchronized with the descent of the serial clock.

(1) Transmission and reception of addresses

· Slave Run: Regardless of the WTIMn bit, the interrupt and waiting timing is determined according to the conditions of note 1 and note 2 above.

· Master Run: Independently of the WTIMn bit, the descending edge of the 9th clock generates a sequence of interruptions and waits.

(2) Data reception

· Master Run/Slave Run: The WTIMn bits determine the timing of interrupts and waits.

(3) Data transmission

· Master Run/Slave Run: The WTIMn bits determine the timing of interrupts and waits.

Remarks: n=0

(4) Method of rescission of waiting

There are four methods for the release of the waiting:

- Write data to IICA shift register n(IICAn).
- Set the bit5(WRELn) of the IICA control register n0(IICCTLn0) to Release Waiting.
- Place the bit1 (STTn) of the IICCTLn0 register (Generate Start Condition) Note.
- Place the bit0 (SPTn) of the IICCTLn0 register (generates a stop condition) Note.

Note Limited to master devices.

When selecting a wait of 8 clocks (WTIMn=0), you need to decide whether to generate a response before canceling the wait.

(5) Test of stop condition

If a stop condition is detect, INTIICAn (SPIEn=1 only case) occurs.

20.5.9 Method for detecting address matching

In I2C bus mode, the master device can select a specific slave device by sending a slave address. It can automatically detect address matching through hardware. When the master device sends the same slave address and the same set address of the slave address register n (SVAn)

Or when only the spreading code is received, a INTIICAn interrupt request is generated.

20.5.10 error detection

In I2C bus mode, since the state of the serial data bus (SDAAn) during transmission is taken to the IICA shift register n(IICAn) of the transmitting device, it is possible to detect transmission errors by comparing the IICA data before starting transmission with the data after transmission. At this time, if two data are different, a transmission error is determined to have occurred.

Remark n=0

20.5.11 extension code

(1) When the 4th bit of the receiving address is '0000' or '1111', the expansion code receiving flag (EXCn) is set '1' and an interrupt request (INTIICAn) is generated at the descending edge of the eighth clock. Does not affect the local station address stored in the slave address register n (SVAn).

(2) When the setting value of the SVAn register is "11110xx0", the following setting occurs if "11110xx0" is sent from the master device. However, an interrupt request (INTIICAn) is generated along the descending edge of the eighth clock.

- Same 4-bit data: EXCn=1
- 7-bit data identical :COIn=1

Note EXCn: bit5 for IICA status register n(IICSn).
COIn: bit4 for IICA status register n(IICSn).

(3) The processing after the interrupt request is different because of the subsequent data of the extension code, and is processed by software. If an expansion code is received at a slave run, communication is attended even if the address is different. For example, if you do not want to run as a slave after receiving the extension code, you must set bit6(LRELn) of IICA control register n0(IICCTLn0) to "1".

Table 20-3 Bit Definition of Primary Expander

slave address	R/W bits	Description
0000000	0	Full Call Address
11110xx	0	10-bit Slave address Assignment (when address is authenticated)
11110xx	1	10-bit Slave Address assignment (when issuing read commands after the same address)

Note 1. Refer to the I2C bus specification issued by NXP for expansion codes other than those listed above.
2.n=0

20.5.12 arbitration

When a plurality of master devices generate start conditions at the same time (STTn position "1" before the STDn bit changes to "1"). This run is called arbitration.

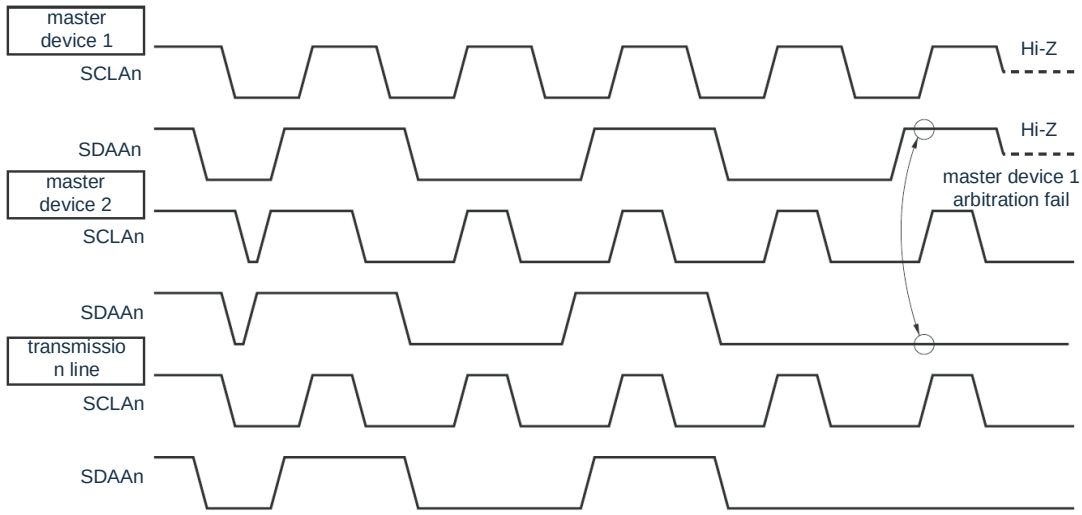
When arbitration failure occurs, the master control device sets ALDn of IICA state register n(IICSn) to "1", and sets SCLAn and SDAAn lines to high impedance state, releasing the bus.

When the next interrupt request occurs (for example: Stop condition is detected at 8 or 9 clock), and arbitration failure is detected by software with ALDn bit '1'.

For the generation sequence of interrupt requests, refer to "Generation Sequence and Latency Control of Interrupt Requests (INTIICAn)."

Remark STDn: bit1 for IICA status register n(IICSn).
STTn:bit1 for the IICA control register n0 (IICCTLn0)

Figure 20-21 example of arbitration sequence



Remark n=0

Table 20-4 The status at the time of arbitration and the sequence in which the interrupt request is generated

Status when arbitration occurs	Generation sequence of interrupt requests
address sending process	Note 1 for the descending edge of the 8th or 9th clock after byte transfer
Read and write information after sending address	
During the sending of the XD	
Read and write information after sending an extension code	
data transmission process	
During the response transmission process after sending data	
A restart condition was detected during data transfer.	
A stop condition was detected during data transfer.	When Generating a Stop Condition (SPIEn=1) Note 2
It is desirable to generate a restart condition, but that data is low level.	Note 1 for the descending edge of the 8th or 9th clock after byte transfer
A restart condition was intended to be generated, but a stop condition was detected.	When Generating a Stop Condition (SPIEn=1) Note 2
Want to generate stop condition, but data is low level.	Note 1 for the descending edge of the 8th or 9th clock after byte transfer
You want to generate a restart condition, but SCLAn is at a low level.	

Note 1. Interrupt requests are generated at the descending edge of the 9th clock when the WTIMn bit (bit3 of IICA control register n0(IICCTLn0) is "1"; When the WTIMn bit is '0' and a slave address of the extension code is received, an interrupt request is generated at the descending edge of the eighth clock.

2. When arbitrations are possible, you must have the SPIEn location "1" in the master run.

Note 1.SPIEn: bit4 for IICA control register n0 (IICCTLn0)

2.n=0

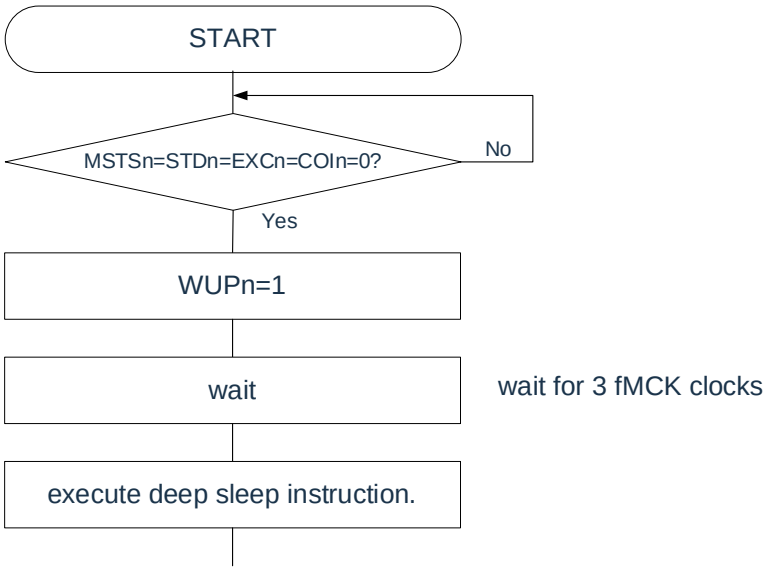
20.5.13 wake-up function

This is a dependent function of I2C, which is the function of generating an interrupt request signal (INTIICAn) upon receipt of the local station address and expansion code. In the case of different addresses, the unnecessary INTIICAn signal is not generated, thereby improving the processing efficiency. If a start condition is detected, a wake-up standby state is entered. Since the master device (which has generated a start condition) may also become a slave device due to arbitration failure, the master device enters a wake-up standby state while sending an address.

To use the wake feature in deep sleep mode, you must have the WUPn position "1". The address can be received regardless of the running clock. Even in this case, an interrupt request signal (INTIICAn) is generated upon receipt of the local station address and the expansion code. After this interrupt occurs, the WUPn bit is cleared '0' by the instruction, returning to the normal run.

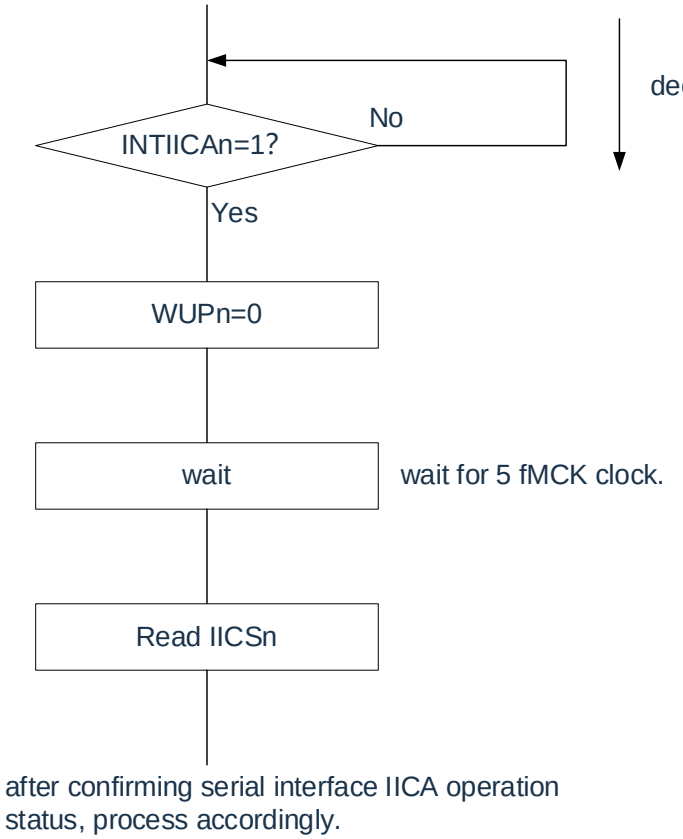
The flow at WUPn location "1" is shown in Figure 20-22, and the flow at WUPn location "0" is shown in Figure 20-23.

Figure 20-22 Process when WUPn position "1"



Remark n=0

Figure 20-23 The process of "0" WUPn by address matching, including receiving an extension code



In addition to the interrupt request (INTIICAn) from the serial interface IICA, deep sleep mode must be removed through the following process.

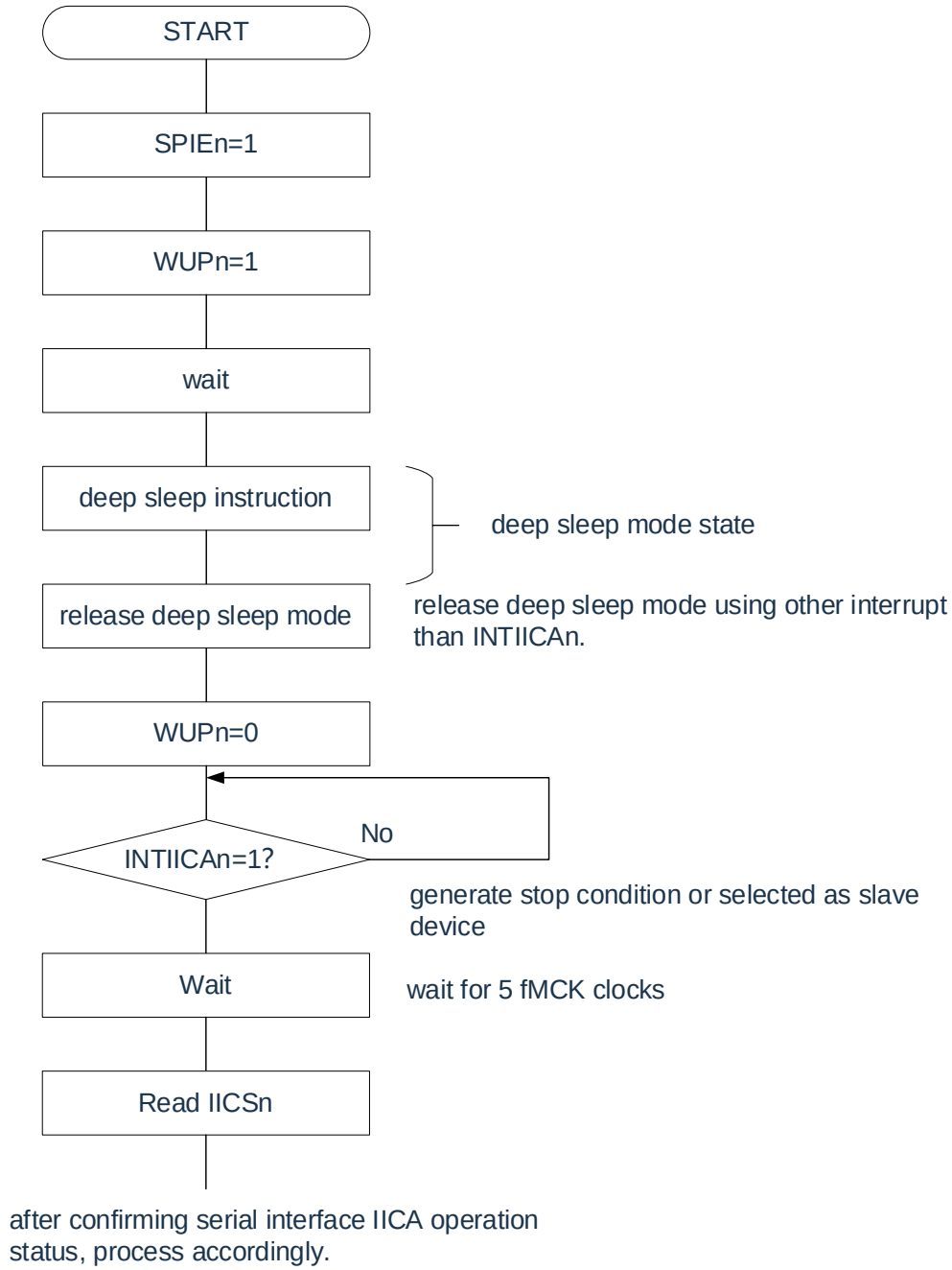
- The next IIC communication is the main control equipment running situation: Figure 20-24 Process
- The next IIC communication is when the slave is running:

Returns from INTIICAn Interrupt : The procedure is the same as in Figure 20-23.

Returns from an interrupt other than an INTIICAn interrupt: You must keep the WUPn bit "1" running before a INTIICAn interrupt is generated.

Remark n=0

Figure 20-24 Operating as a master device after deep sleep mode is removed by an interrupt other than INTIICAn



Remark n=0

20.5.14 communication reservation

(1) The case where the communication reservation function is allowed (IICA flag register n(IICFn) bit0(IICRSVn)=0)

When the next master control communication is performed in the state of not joining the bus, a start condition can be sent when the bus is released by a communication reservation. The non-join bus at this time includes the following two states:

- When the arbitration result is neither master nor slave
- bit6(LRELn) of IICA control register n0(IICCTLn0) is set '1'after receiving the spreading code and releasing the bus

If the bit1(STTn) of the IICCTLn0 register is set to "1" in the state of not joining the bus, the start condition is automatically generated after the bus is released (the stop condition is detected) and enters the waiting state.

The IICCTLn0 register's bit4 (SPIEn) is set to '1'. After detecting the release of the bus (stopping condition) by the generated interrupt request signal INTIICAn, communication is automatically initiated as the master. The data written to the IICAn register is invalid before stopping conditions are detected.

When the STTn position "1" is taken, it is decided whether to run as a start condition or as a communication reservation according to the bus status.

- Bus is on release.....Build Start Condition
- When the bus is not in a released state (standby state).....communication reservation

After the STTn position "1" and the waiting time, it is confirmed by the MSTSn bit (bit7 of IICA status register n(IICSn) whether to run as a communication reservation.

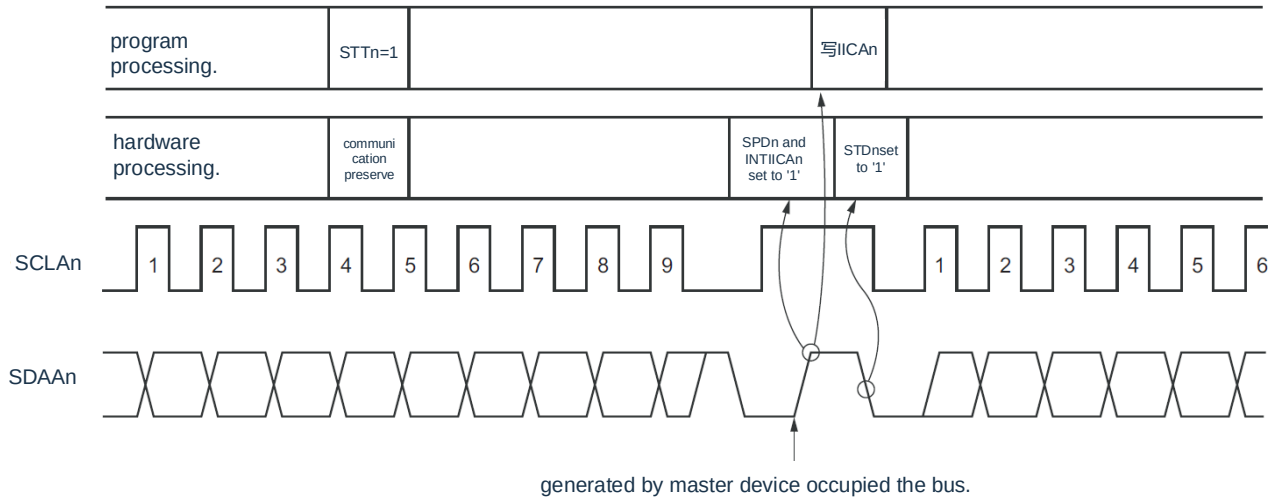
Waiting times calculated by the following equation must be ensured by software:

Wait time from STTn location "1" to confirming MSTSn flag: (Set value for IICWLn+Set value for IICWHn+4)/ f_{MCK}+t_F×2

Note 1.IICWLn :IICA low level width set-up register n
ICWHn :IICA high level width set-up register n
tF Down time of SDAAn and SCLAn signals
fMCK : IICA Run Clock Frequency
2.n=0

The timing of the communication reservation is shown in Figure 20-25.

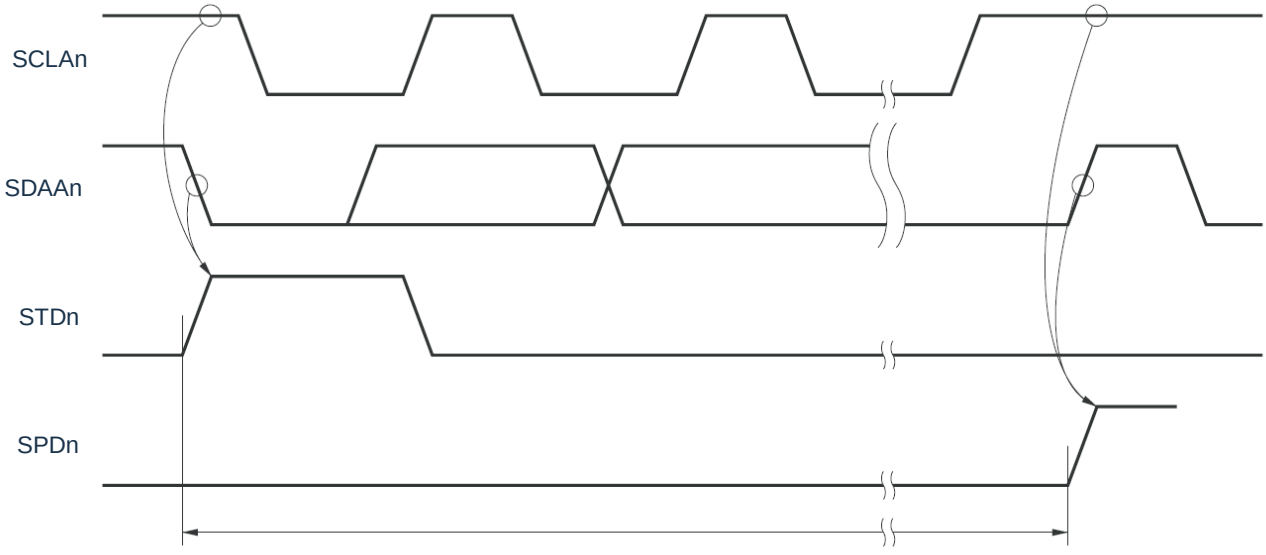
Figure 20-25 Timing of communication appointments



Remark ICAn :IICA shift register n
STTn :bit1 for IICA control register n0(IICCTLn0)
STDn :bit1 for IICA state register n(IICSn)
SPDn :bit0 for IICA state register n(IICSn)

The communication reservation is accepted by the timing shown in FIGS. 20-26. After the bit1(STDn) of the IICA status register n(IICSn) becomes '1' and before the stop condition is detected, the bit1(STTn) of the IICA control register n0(IICCTLn0)is set '1' for communication reservation.

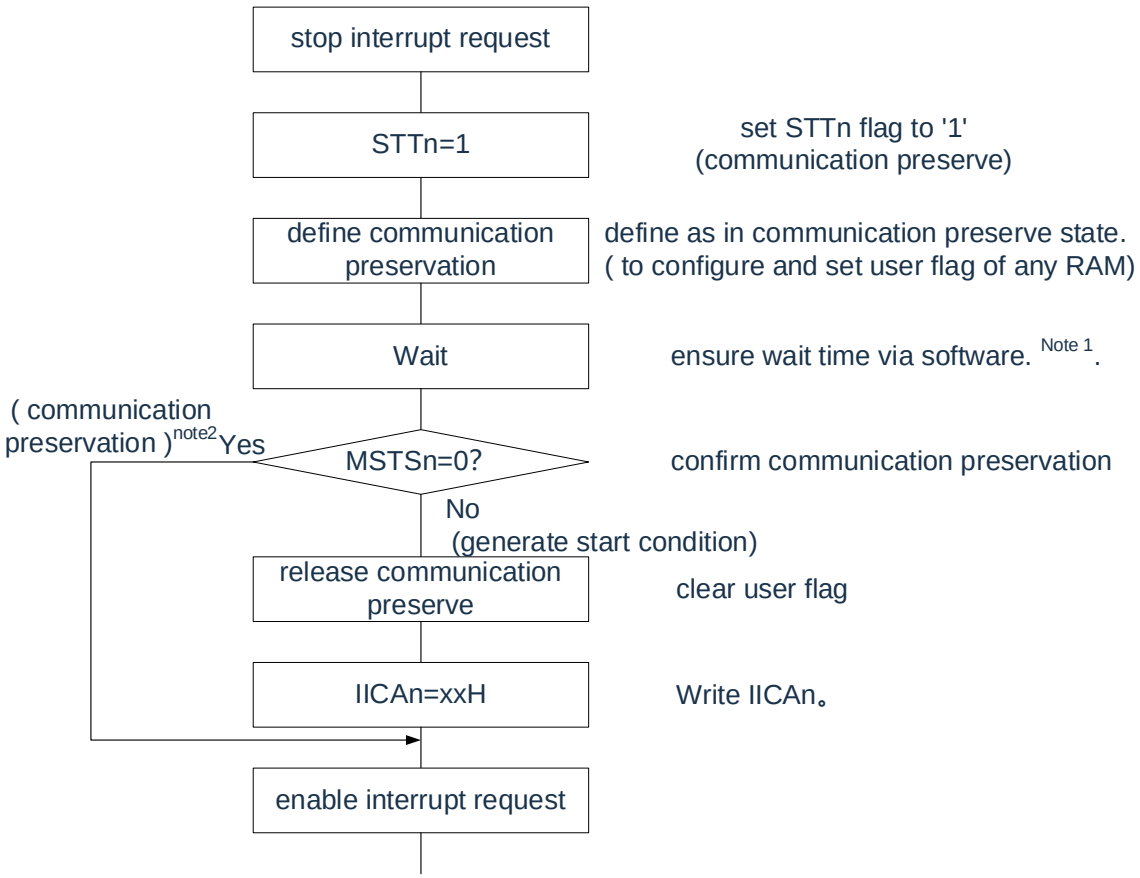
Figure 20-26 Timing of acceptance of communication appointments



The steps of the communication reservation are shown in FIG. 20-27.

Remark n=0

Figure 20-27 Steps for communication reservation



Note 1. The wait time is as follows: (Set value for IICWLn + Set value for IICWHn +4)/ f_{MCK}+t_F×2

2. Write the IICA shift register n (IICAn) by stopping the conditional interrupt request when the communication reservation is running.

Note 1.STTn :bit1 for IICA control register n0(IICCTLn0)
MSTSn :bit7 for IICA state register n(IICSn)
ICAn :IICA shift register n
IICWLn :IICA low level width set-up register n
ICWHn :IICA high level width set-up register n
tF Down time of SDAAn and SCLAn signals
fMCK : IICA Run Clock Frequency
2.n=0

(2) Case where communication reservation is disabled (IICA flag register n(IICFn) bit0(IICRSVn)=1)

During bus communication, if the bit1(STTn) of IICA control register n0(IICCTLn0) is set to "1" without taking part in this communication, this request is rejected and no start condition is generated. The non-join bus at this time includes the following two states:

- When the arbitration result is neither master nor slave
- Not run as slave after receiving extension code (bit6 (LRELn) of IICCTLn0 register is set "1" without returning a reply, releasing bus after exiting communication)

It is possible to confirm whether a start condition was generated or the request was rejected by STCFn (bit7 of IICFn register). Since it takes 5 fMCK clocks from the STTn bit '1' to the STCFn location '1', this time must be ensured by software.

Remark n=0

20.5.15 Other Precautions

(1) Cases with STCENn bit "0"

Immediately after I^2C is allowed to run (IICEn=1), the actual bus state is considered as the communication state (IICBSYn=1). In order to perform the master control communication in a state where no stop condition is detected, the stop condition must be generated, and the master control communication is performed after the bus is released. For multi-master, master communication cannot be performed in a state where the bus is not released (no stop condition is detected). Generate stop conditions in the following order:

(1) Set the IICA control register n1 (IICCTLn1).
(2) Set the bit7(IICEn) of IICA control register n0(IICCTLn0) to "1.
(3) Set bit0 (SPTn) of IICCTLn0 register to "1".

(2) Cases with STCENn bit'1'

Immediately after I^2C is allowed to run (IICEn=1), the actual bus state is considered as a released state (IICBSYn=0). Therefore, it is necessary to confirm that the bus has been released in order not to destroy other communications when generating the first start condition (STTn=1).

(3) I2C communications with other equipment in progress

When the SDAAn pin is low and the SCLAn pin is high, if I2C is allowed to run and participate in communication halfway, the macro of I2C is considered to be SDAAn pin changed from high to low (start condition detected). If the value on the bus is a value recognized as an expansion code at this time, a reply is returned that interferes with I2C communication. To avoid this, you must start I2C in the following order:

(1) The bit4 (SPIEn) of the IICCTLn0 register is cleared "0", and the interrupt request signal (INTIICAn) is prohibited.
(2) Set bit7(IICEn) of the IICCTLn0 register to '1' to allow I2C.
(3) Wait for the start condition to be detected.
(4) Setting bit6(LRELn) of IICCTLn0 register "1" before the return response (within 4 to 72 fMCK clocks after IICEn position "1".

(4) After the STTn bit and the SPTn bit (bit1 and bit0 of the IICCTLn0 register) are set, reset before "0" is prohibited.

(5) If a communication reservation is made, the SPIEn bit (bit4 of the IICCTLn0 register) must be set "1" to generate an interrupt request. After generating an interrupt request, the transmission is started by writing communication data to the IICA shift register n (IICAn). If an interrupt does not occur when a stop condition is detected, the wait state is stopped because an interrupt request is not generated at the start of communication. However, the SPIEn position" 1" is not required when detecting MSTSn bits (bit7 of IICA status register n (IICSn) through software.

Remark n=0

20.5.16 communication operation

The following 3 run steps are shown through the flowchart.

(1) Main Control Operation of Single Master Control System

The flow chart used as the master device in a single master system is shown below.

This process is broadly divided into Initial Settings and Communication Processing." The Initial Set-up section is executed at start-up, and the Communication Processing section is executed after preparation required for communication if communication with the slave device is required.

(2) Master Control Operation of Multi-Master Control System

In the multi-master system of I2C bus, it is impossible to judge whether the bus is released or in use. Here, if the data and the clock are at a high level within a certain period of time (1 frame), the bus is taken into communication as a released state. This process is broadly divided into Initial Set-up, Communication Waiting, and Communication Handling. A process that is designated as a slave device due to an arbitration failure is omitted here, only for use as the master device. After executing the "Initial Settings" section at start-up, add to the bus and wait for communication requests from the master or slave. The actual communication is the "communication processing" section, which supports arbitration with other master devices in addition to data transmission and reception with the slave devices.

(3) slave operation

An example of using as an I2C bus slave is shown below.

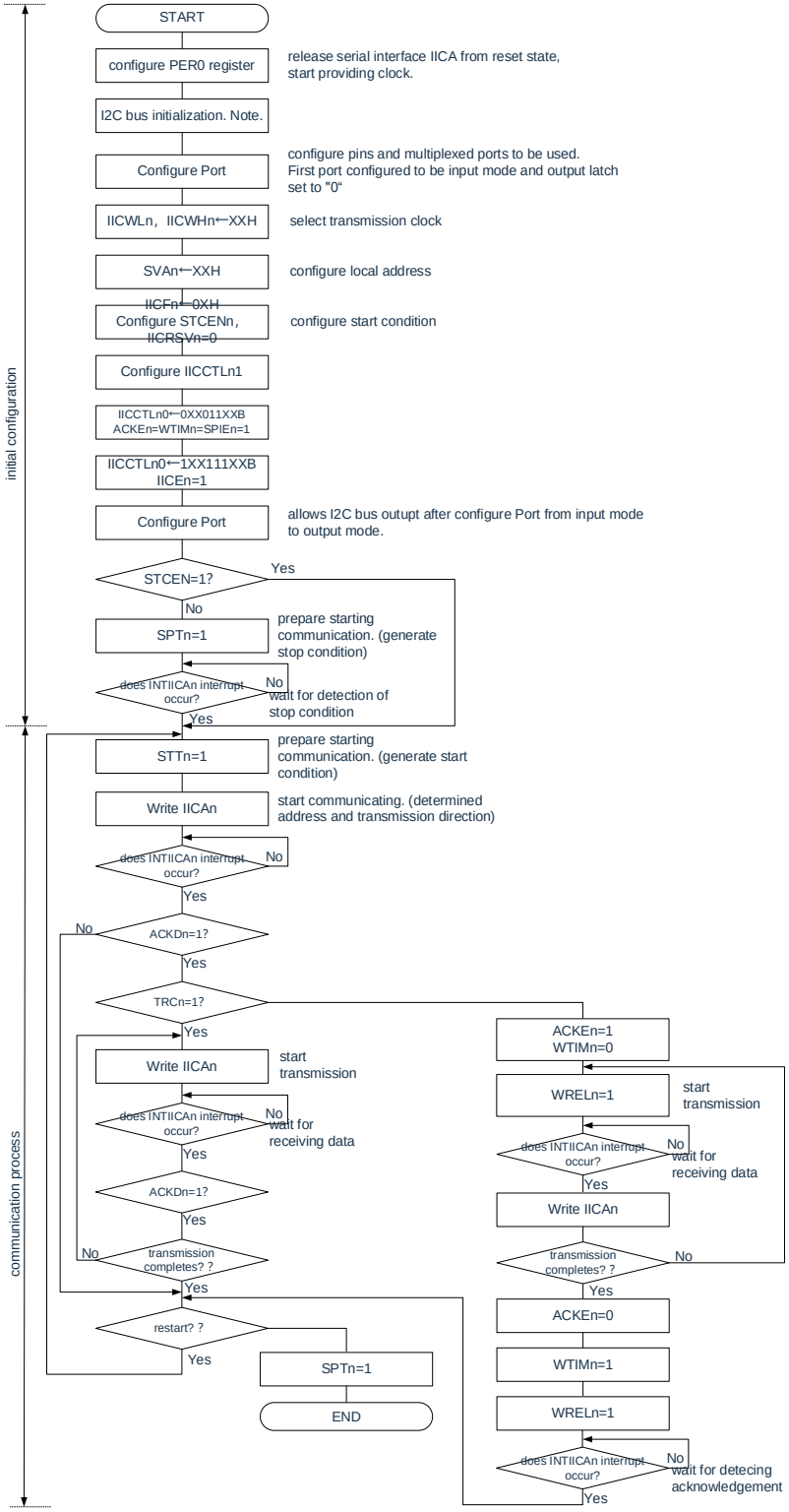
When used as a slave, the operation starts with an interrupt. Perform the Initial Settings section at start-up, and then wait for the INTIICAn interrupt by "Communication Waiting." If a INTIICAn interrupt occurs, a communication state is determined and a flag is transmitted to a main processing section.

Perform the required "communication processing" by examining the respective flags.

Remark n=0

(1) Main Control Operation of Single Master Control System

Figure 20-28 Main Control Operation of Single Master Control System

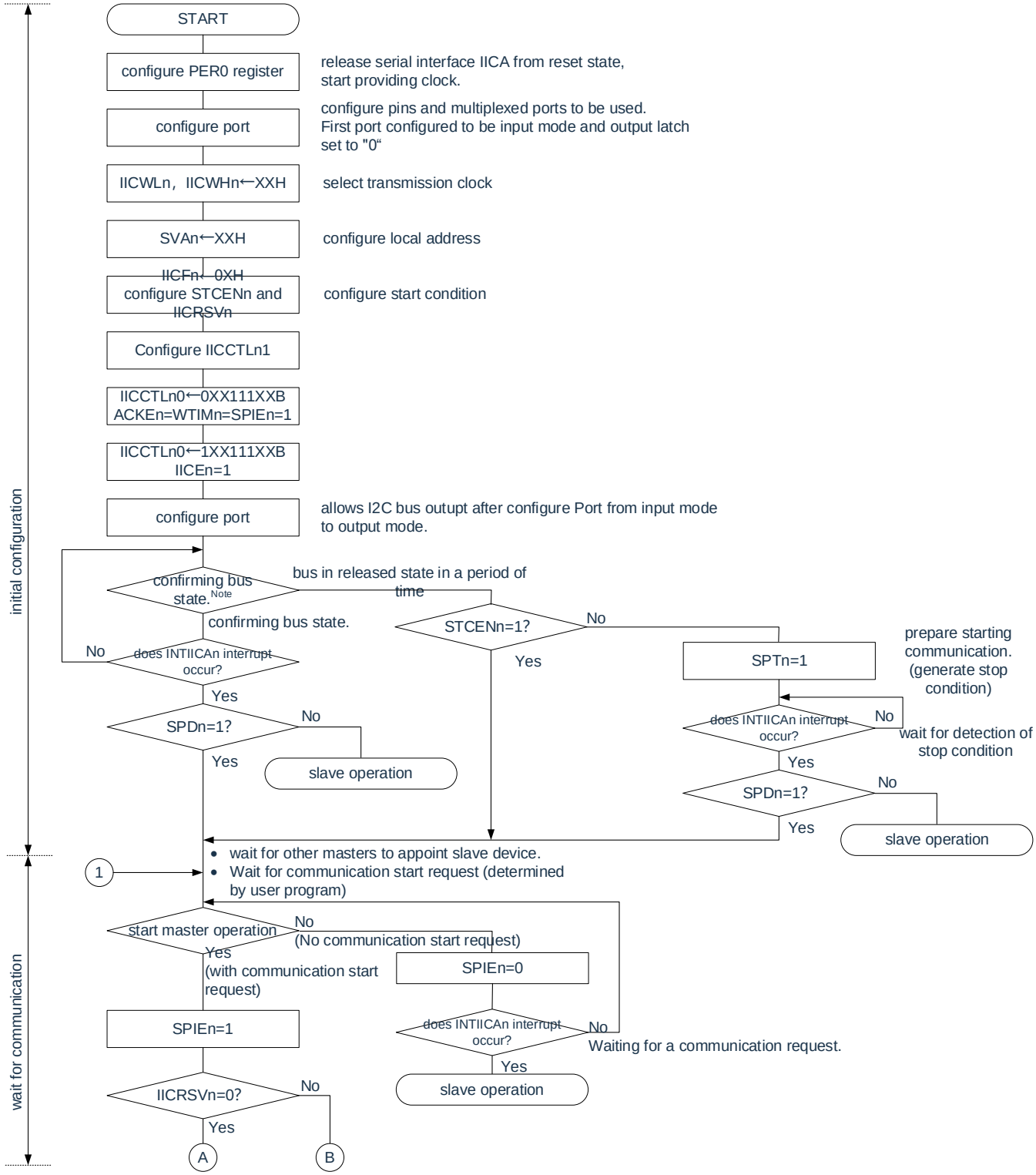


Note: The I2C bus (SCLAn and SDAAn pins are high level) must be released based on the specification of the product in communication. For example, if the EEPROM is in a low level output to the SDAAn pin, the SCLAn pin must be set as the output port.

Note 1. The format sent and received must conform to the specifications of the product in communication.
2.n=0

(2) Master Control Operation of Multi-Master Control System

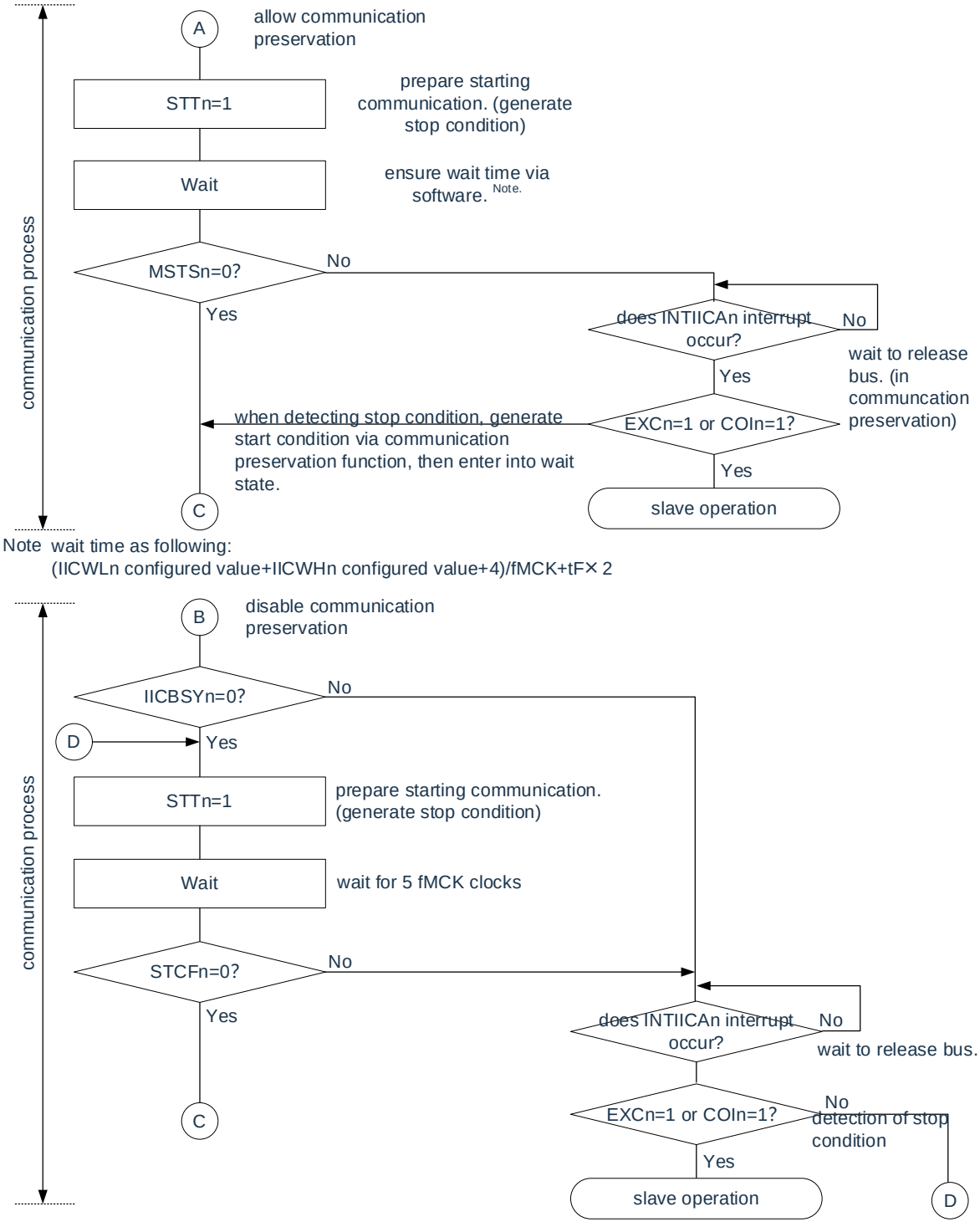
Figure 20-29 Main control operation of multi-master control system (1/3)



Note You must verify that the bus is in a released state for a certain time (for example, 1 frame) (CLDn bit=1, DADn bit=1). When the SDAAn pin is fixed at a low level, it must be determined whether to release the I²C bus (SCLAn and SDAAn pins are high).

Remark n=0

Figure 20-29 Main control operation of multi-master control system (2/3)



Note wait time as following:
(IICWLn configured value+IICWHn configured value+4)/fMCK+tF× 2

Note 1.IICWLn :IICA low level width set-up register n

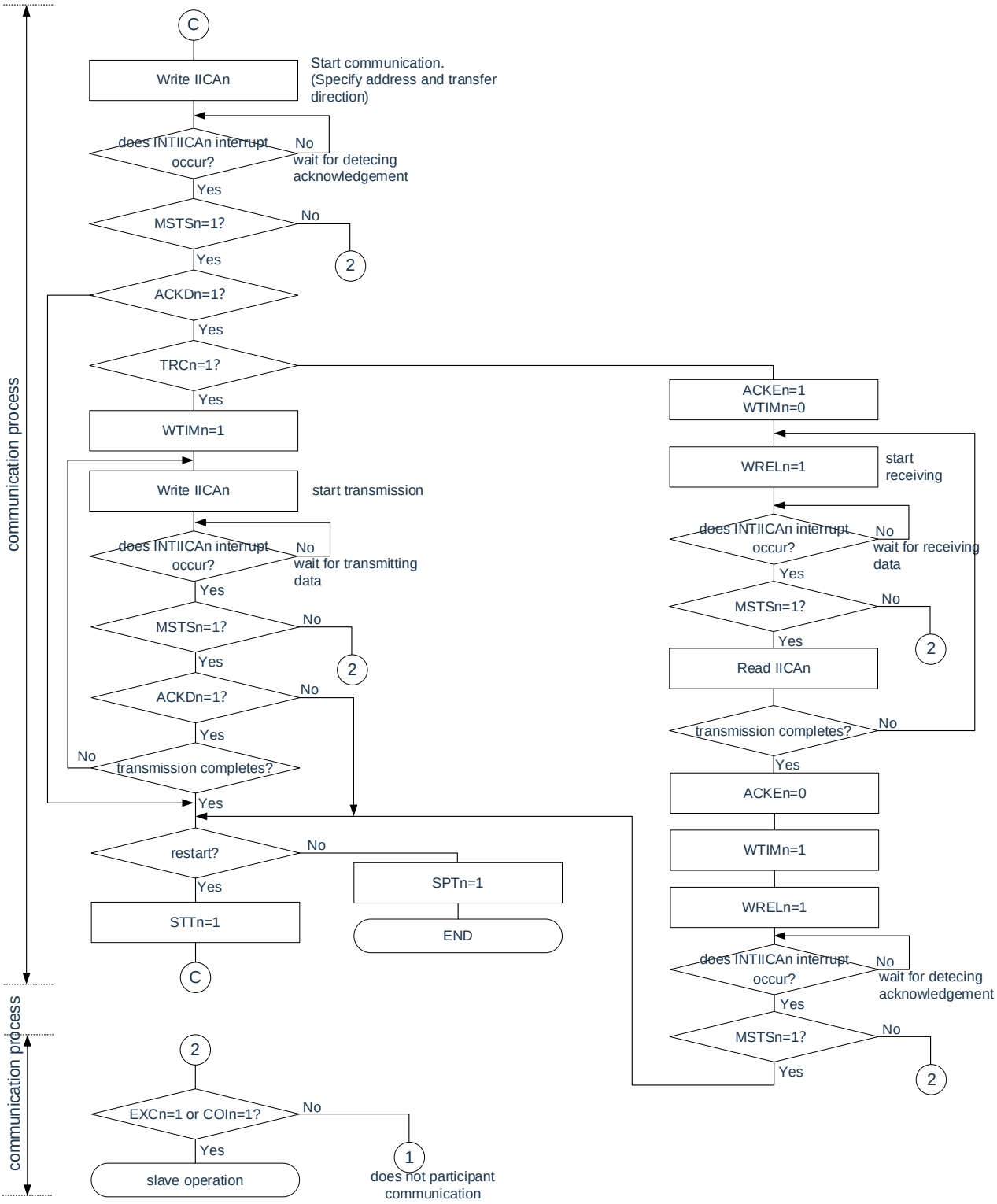
ICWHn :IICA high level width set-up register n

tF Down time of SDAAn and SCLAn signals

fMCK : IICA Run Clock Frequency

2.n=0

Figure 20-29 Main control operation of multi-master control system (3/3)



Note 1. The format of delivery and receipt must conform to the specifications of the product in communication.

2. In the case of multi-master system as master device, the MSTSn bit must be read each time INTIICAn interrupt occurs to confirm arbitration results.

3. In case d is used as slave in a multi-master system, the state must be confirmed by IICA state register n (IICFn) and IICA flag register n.

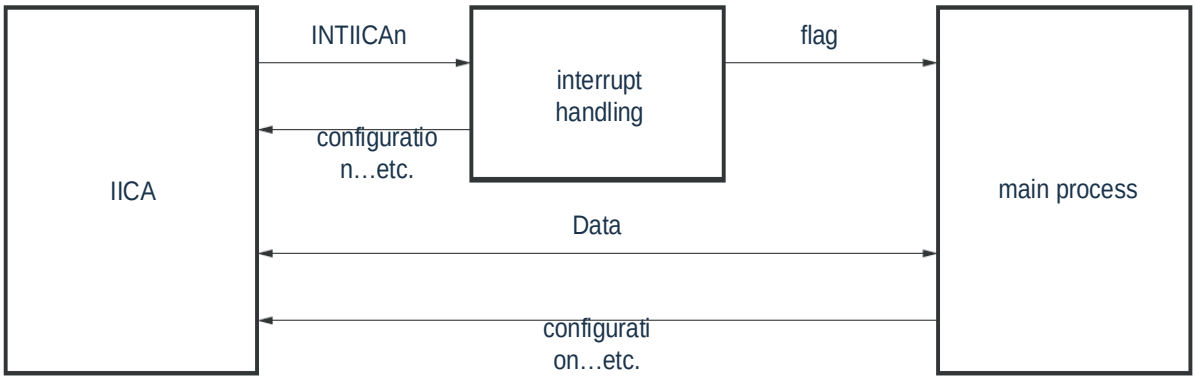
4.n=0

(3) slave operation

The processing steps for a Slave Run are as follows.

Dependent runs are essentially event driven and therefore require processing through INTIICAn interrupts (requiring significant change processing of operational states such as stop condition detection in communications).

In this specification, assuming that data communication does not support an expander code, the INTIICAn interrupt process only performs a state transfer process and the actual data communication is performed by the main process unit.



Therefore, the following three marks are prepared and the marks are transferred to the main processing unit instead of INTIICAn for data communication processing.

(1) Communication mode mark

This flag indicates the following two communication states:

· Purge mode: No status for data communication

· Communication mode: The state of data communication in progress (detection of valid addresses ~ detection of stop conditions, no response detected from master devices, different addresses)

(2) Ready Mark

This flag indicates that data communication is available. In a normal data communication, the same as INTIICAn interrupt, is set by an interrupt processing section and cleared by a main processing section. At the beginning of communication, the flag is cleared by the interrupt processing section. However, the interrupt processing section does not position the ready flag when the first data is transmitted, so the first data is transmitted without clearing the flag (address matching is interpreted as the next data request).

(3) Direction of Communication mark

This flag indicates the direction of communication and is the same value as the TRCn bit.

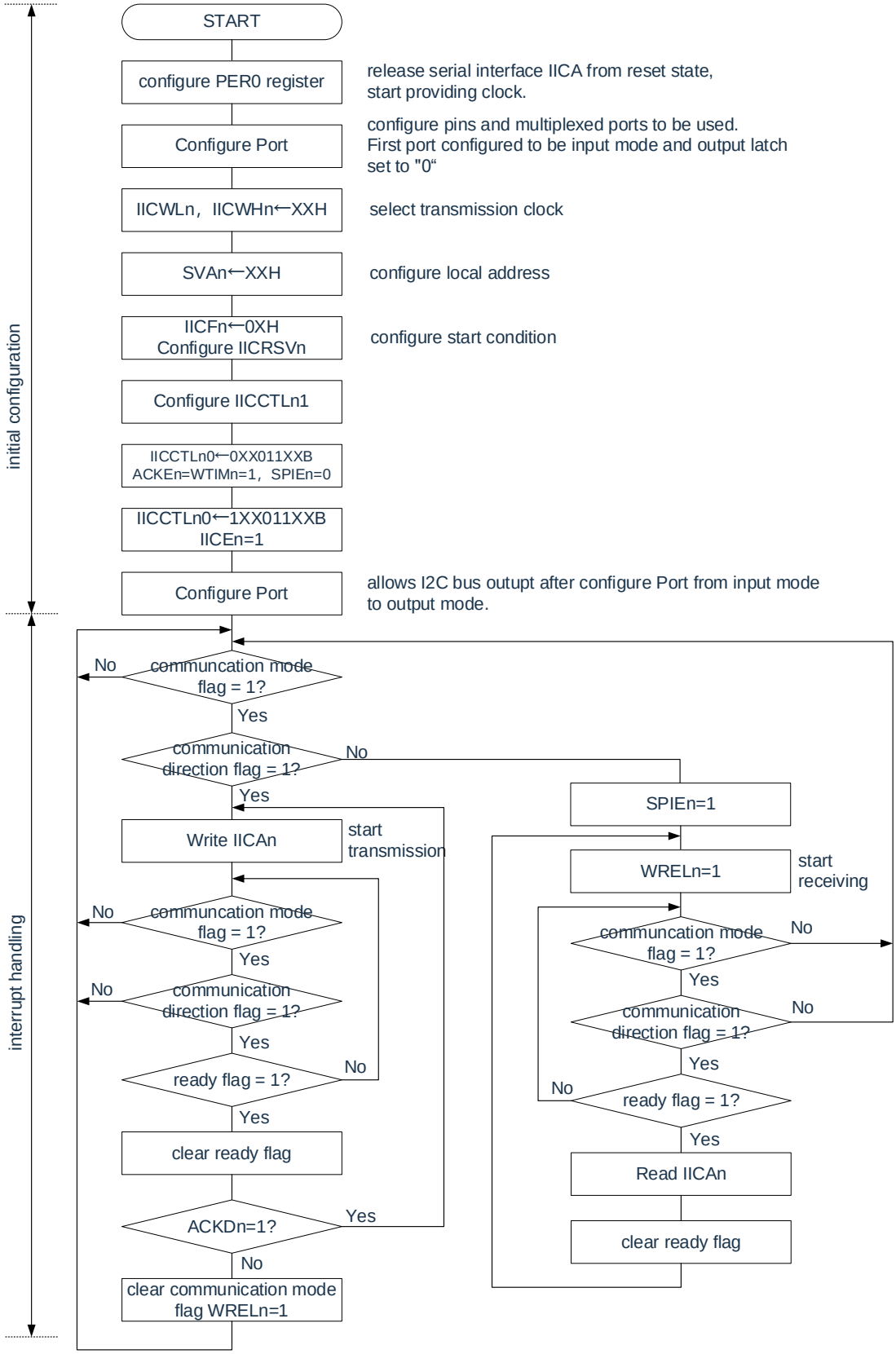
Remark n=0

The main process part of the slave run is run as follows.

Start the serial interface IICA, and wait to become communicable. If the state becomes communicable, the communication mode flag and the ready flag are used to communicate (the state is confirmed here by the flag because the stop condition and the start condition are processed by interrupt).

At the time of transmission, the transmission is repeated until the master device does not return an acknowledgement. If the master does not return a reply, the communication is terminated. Upon receipt, a desired amount of data is received. If the communication ends, no reply is returned on the next data. Thereafter, the master device generates a stop condition or a restart condition, thereby exiting the communication state.

Figure 20-30 Slave Run Step (1)



Note 1. The format of delivery and receipt must conform to the specifications of the product in communication.

2.n=0

An example of the steps for a slave device to process through INTIICAn interrupts is shown below (assuming no processing with an expander in this scenario). The status is interrupted by INTIICAn and the following processing is performed.

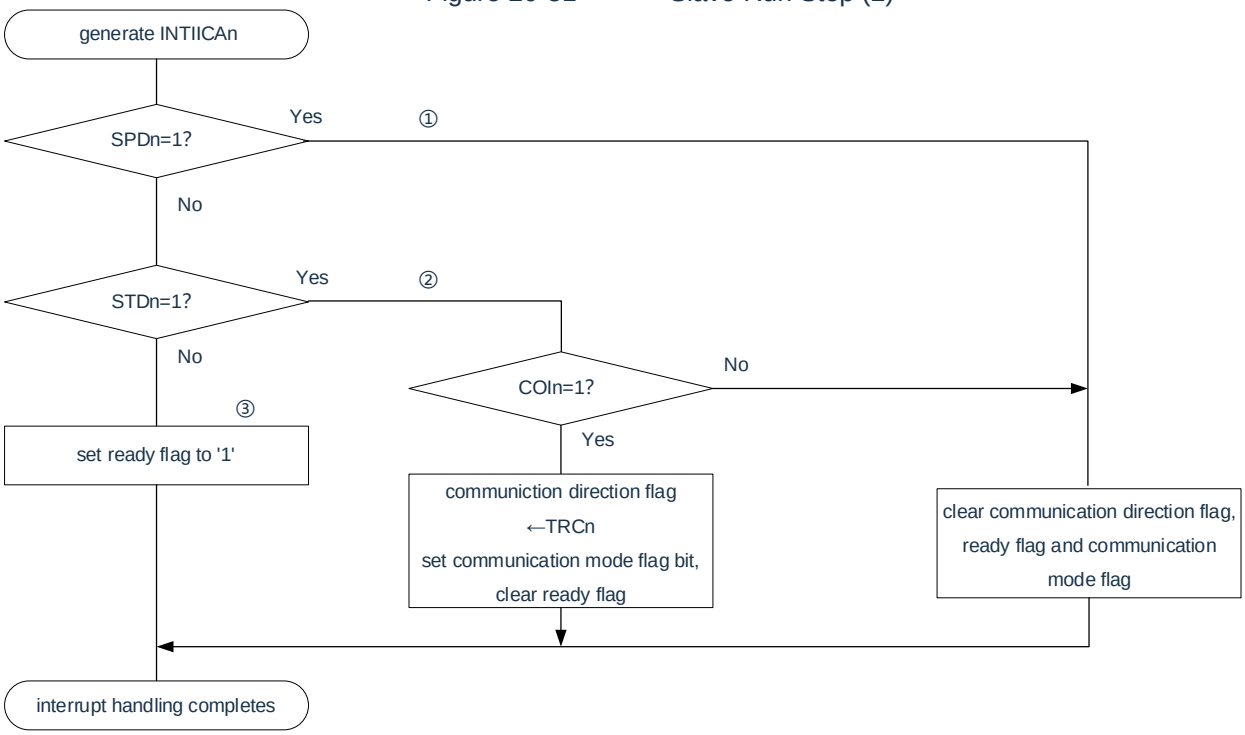
① If a stop condition is generated, the communication is ended.

② If you generate a start condition, confirm the address. If the addresses are different, the communication is terminated. If the addresses are the same, set to communication mode and unwaits, and returns from the interrupt (clear the readiness flag).

③ When sending and receiving data, the I2C bus remains in a waiting state and returns from an interrupt as long as the ready flag is set.

Remark The above (1) to (3) correspond to (1) to (3) of the sub-operation steps (2).

Figure 20-31 Slave Run Step (2)



Remark n=0

20.5.17 I2C Generation Sequence of Interrupt Requests (INTIICAn)

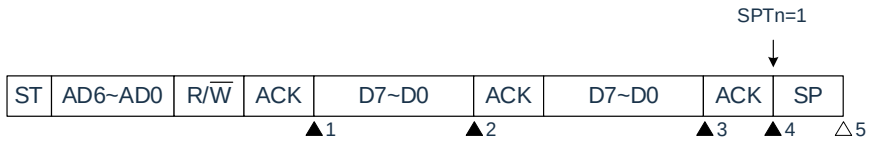
The values for the sending and receiving sequence of the data, the generation sequence of the INTIICAn interrupt request signal, and the IICA status register n (IICSn) when the INTIICAn signal is generated are shown below.

Note 1.ST : Start Condition
AD6~AD0: Address
R/$\overline{W}$: Assignment of Routing Direction
ACK : Acknowledgement
D7~D0 : data
SP : stop condition
2.n=0

(1) master run

(a) Start~Address~Data~Data~Stop (send and receive)

(i) WTIMn=0's situation

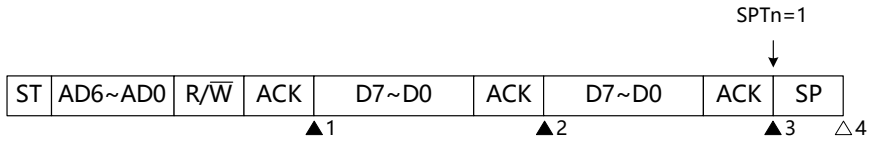


▲1:IICSn=1000X110B
▲2:IICSn=1000X000B
▲3:IICSn=1000X000B(set WTIMn bit to"1") Note
▲4:IICSn=1000XX00B(set SPTn bit to "1")
△5:IICSn=00000001B

Note: to generate stop condition, must set WTIMn bit to '1' and modify INTIICAn interrupt requet signal generation timing sequence.

Remark ▲ must generate
△ only generate while SPIEn bit is '1'
X any

(ii) WTIMn=1's situation



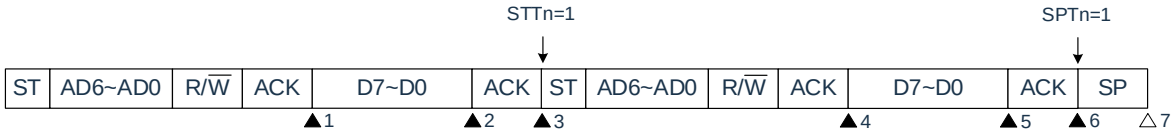
▲1:IICSn=1000X110B
▲2:IICSn=1000X100B
▲3:IICSn=1000XX00B(set SPTn bit to "1")
△4:IICSn=00000001B

Remark ▲ must generate
△ only generate while SPIEn bit is '1'
X any

Remark n=0

(b) Start~Address~Data~Start~Address~Data~Stop (Start Over)

(i) WTIMn=0's situation

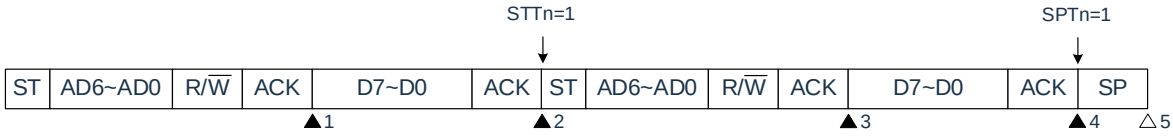


▲1:IICSn=1000X110B
▲2:IICSn=1000X000B(set WTIMn bit to"1") Note1
▲3:IICSn=1000XX00B(set WTIMn bit to "0". Note 2 and set STTn bit to"1")
▲4:IICSn=1000X110B
▲5:IICSn=1000X000B(set WTIMn bit to"1") Note3
▲6:IICSn=1000XX00B(set SPTn bit to "1")
△7:IICSn=00000001B

Note1. to generate start condition, must set WTIMn bit to '1' and modify INTIICAn interrtupt request signal generation timing sequence.
2. To recover original configuration, must set WTIMn bit to '0'.
3. to generate stop condition, must set WITIMn bit to '1' and modify INTIICAn interrupt request signal generation timing sequence.

Remark ▲ must generate
△ only generate while SPIEn bit is '1'
X Any

(ii) WTIMn=1's situation



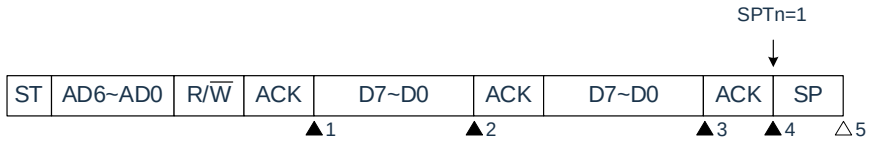
▲1:IICSn=1000X110B
▲2:IICSn=1000XX00B(set STTn bit to "1")
▲3:IICSn=1000X110B
▲4:IICSn=1000XX00B(set SPTn bit to "1")
△5:IICSn=00000001B

Remark ▲ must generate
△ only generate while SPIEn bit is '1'
X any

Remark n=0

(c) Start~Code~Data~Data~Stop (send extension code)

(i) WTIMn=0's situation

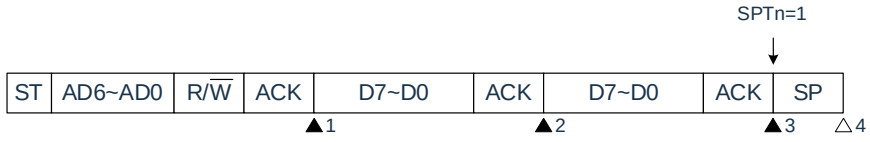


▲1:IICSn=1010X110B
▲2:IICSn=1010X000B
▲3:IICSn=1010X000B(set WTIMn bit to"1") Note
▲4:IICSn=1010XX00B(set SPTn bit to "1")
△5:IICSn=00000001B

Note: to generate stop condition, must set WITIMn bit to '1' and modify INTIICAn interrupt request signal generation timing sequence.

Remark ▲ must generate
△ only generate while SPIEn bit is '1'
X any

(ii) WTIMn=1's situation



▲1:IICSn=1010X110B
▲2:IICSn=1010X100B
▲3:IICSn=1010XX00B(set SPTn bit to "1")
△4:IICSn=00000001B

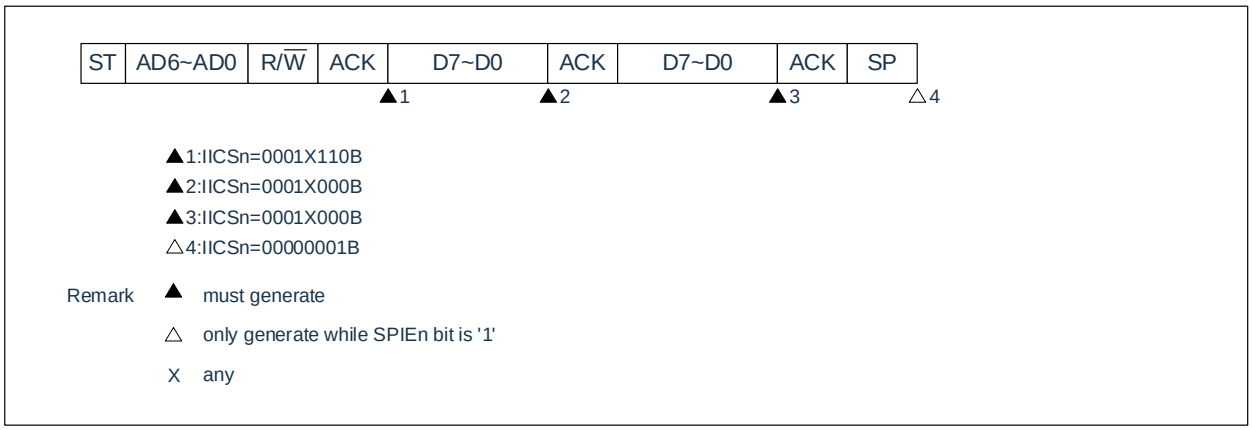
Remark ▲ must generate
△ only generate while SPIEn bit is '1'
X any

Remark n=0

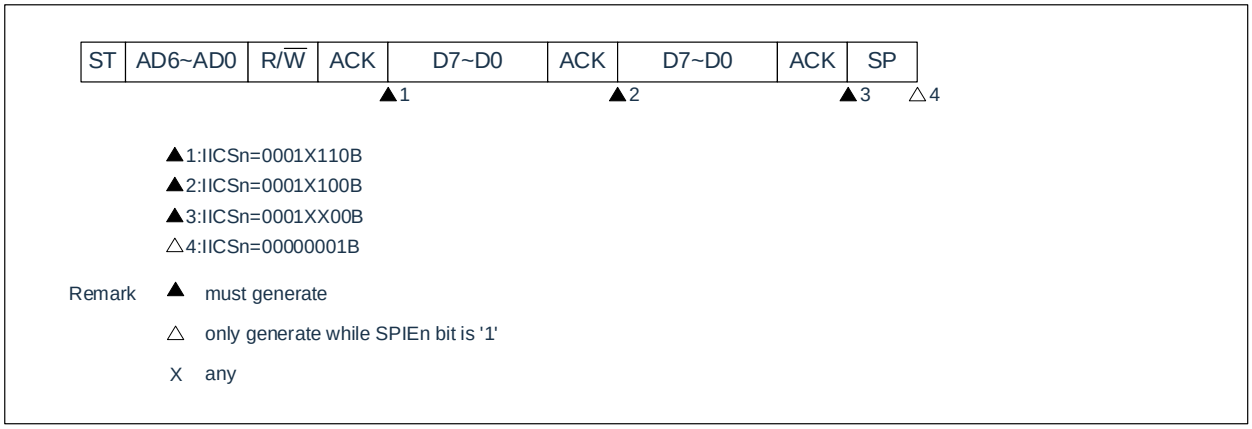
(2) Slave Run (When You Receive a Slave Address)

(a) Start~Address~Data~Data~Stop

(i) WTIMn=0's situation



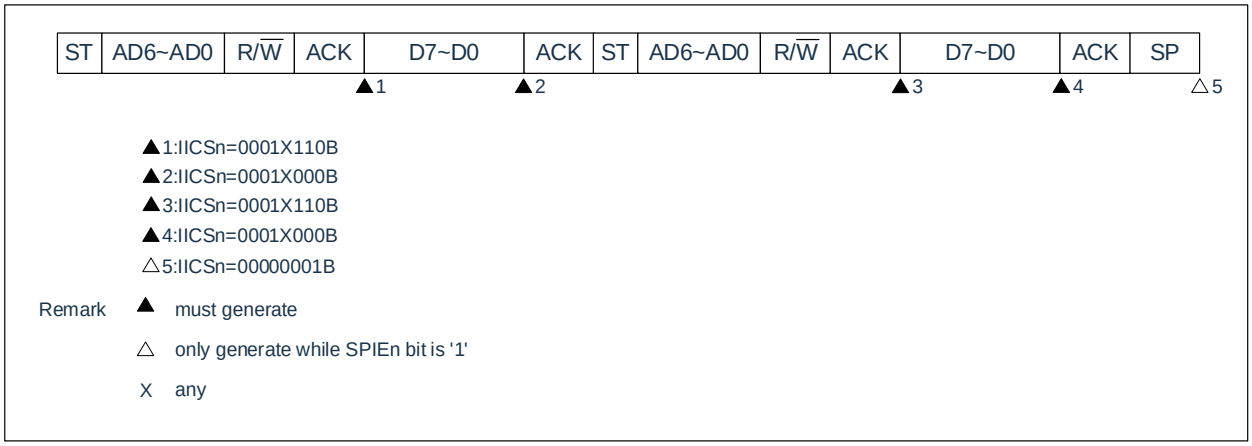
(ii) WTIMn=1's situation



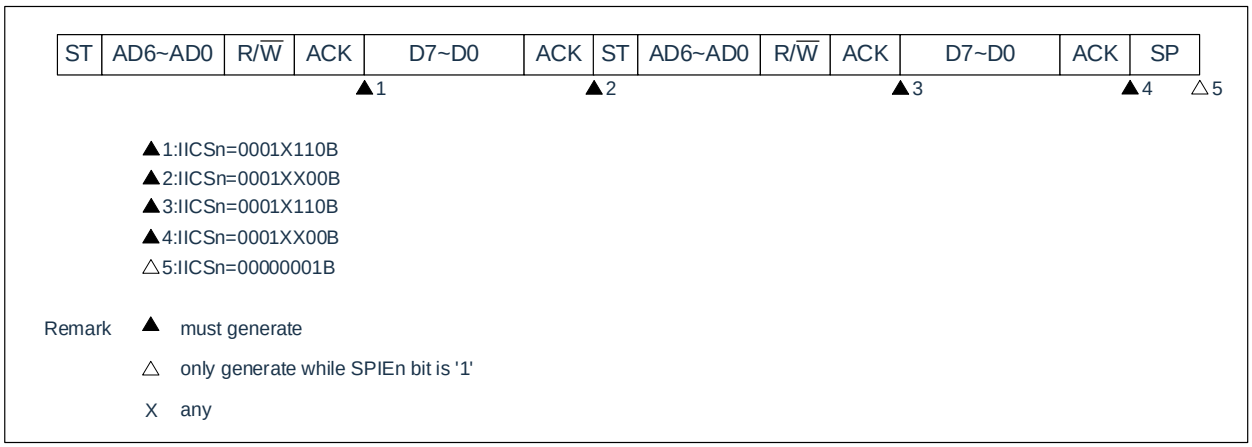
Remark n=0

(b) Start~Address~Data~Start~Address~Data~Stop

(i) The case of WTIMn=0 (SVAn is the same after the restart)



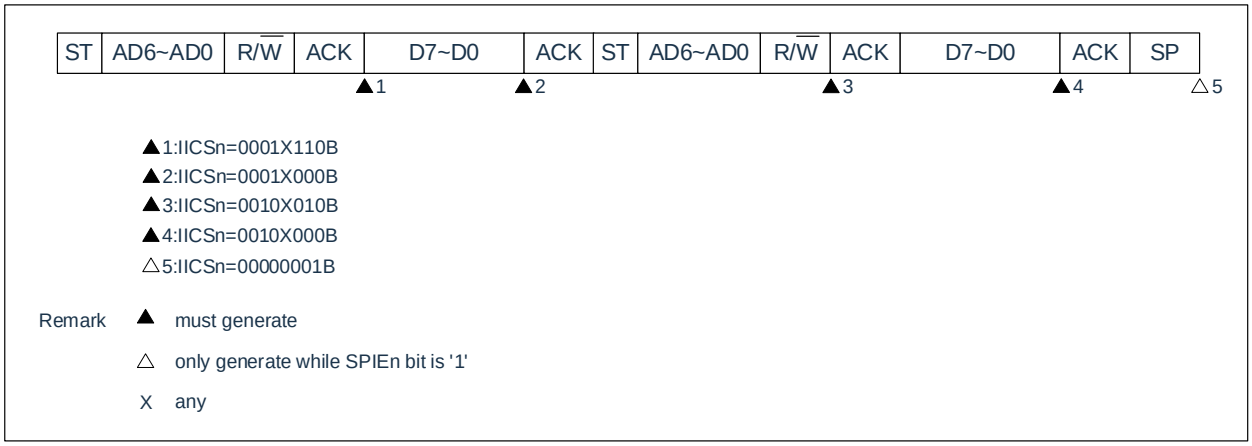
(ii) the situation of WTIMn=1 (same as SVAn upon reinitiation)



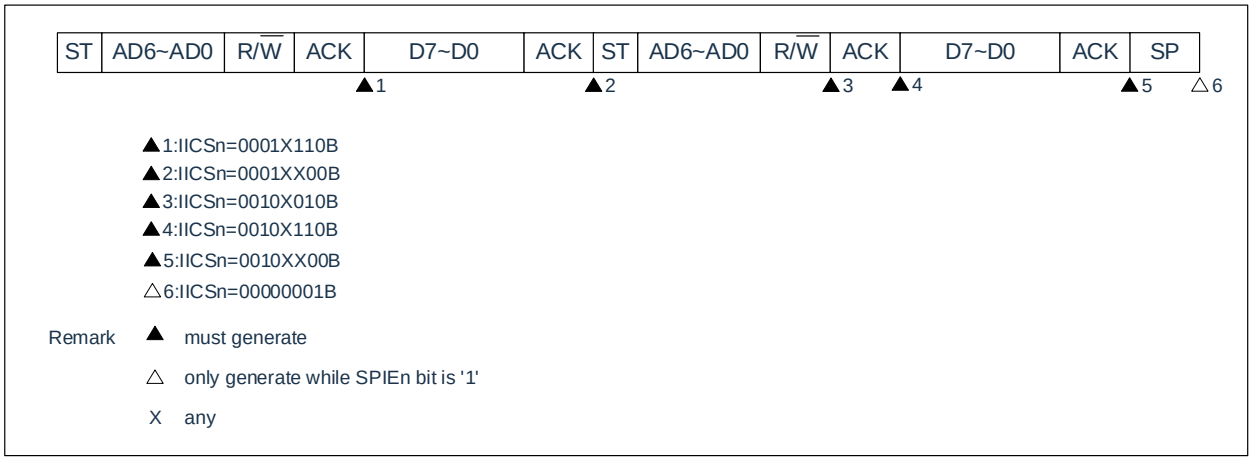
Remark n=0

(c) Start~Address~Data~Start~Code~Data~Stop

(i) The case of WTIMn=0 (different address after restart (extension code))



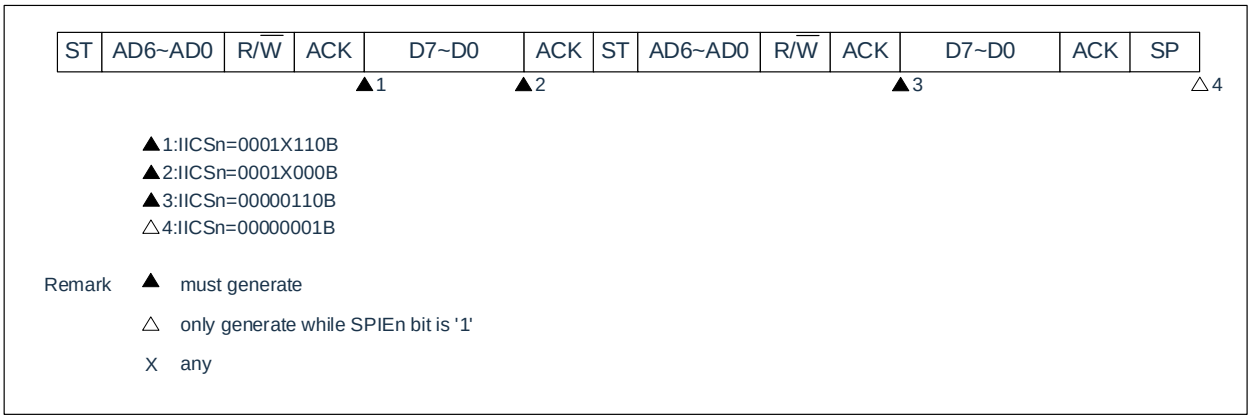
(ii) the case of WTIMn=1, with a different address (extension code) upon restart



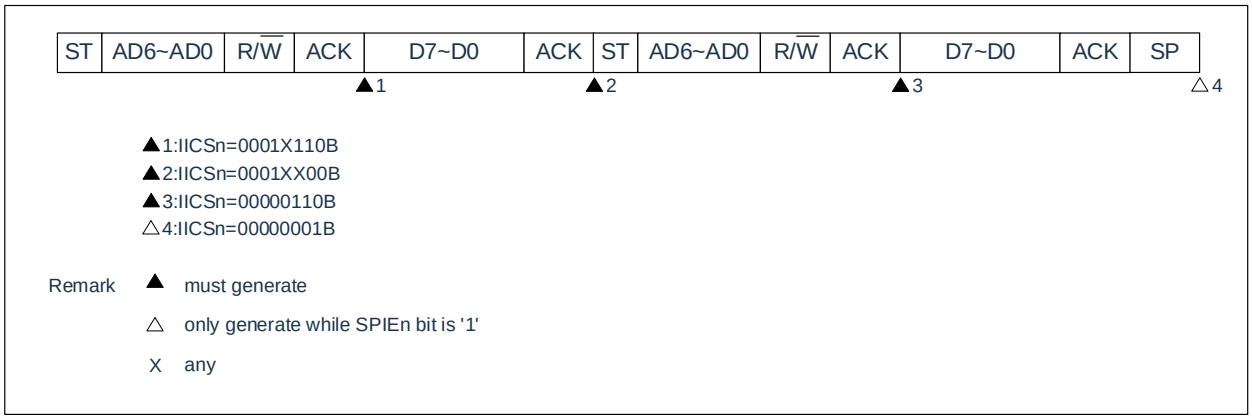
Remarks: n=0

(d) Start~Address~Data~Start~Address~Data~Stop

(i) The case of WTIMn=0 (addresses differ after restart (non-expander))



(ii) the case of WTIMn=1, with a different address (non-expander) after the restart



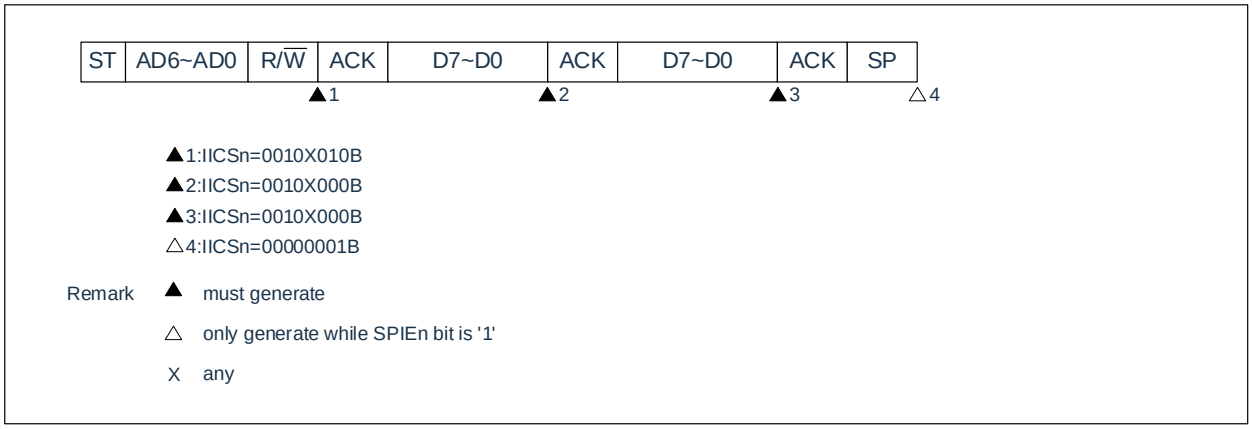
Remarks: n=0

(3) Slave Run (Case of Receiving Extension Code)

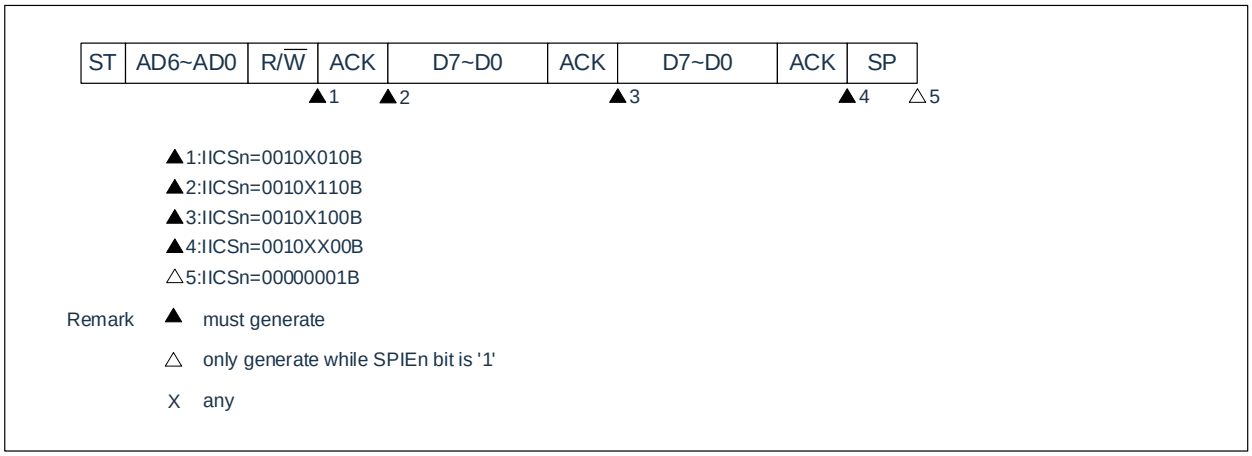
Always participate in communication when you receive an expansion code.

(a) Start~Code~Data~Data~Stop

(i) WTIMn=0's situation



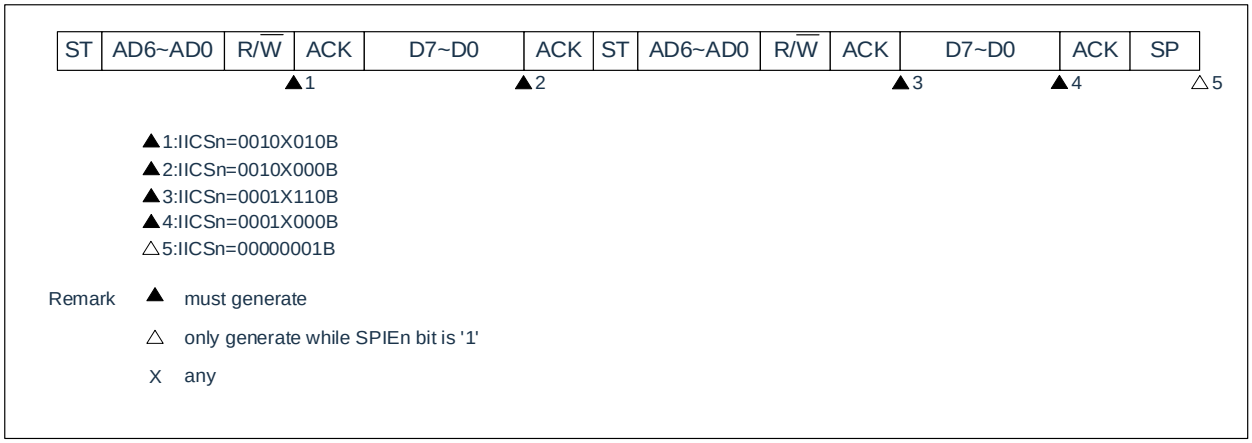
(ii) WTIMn=1's situation



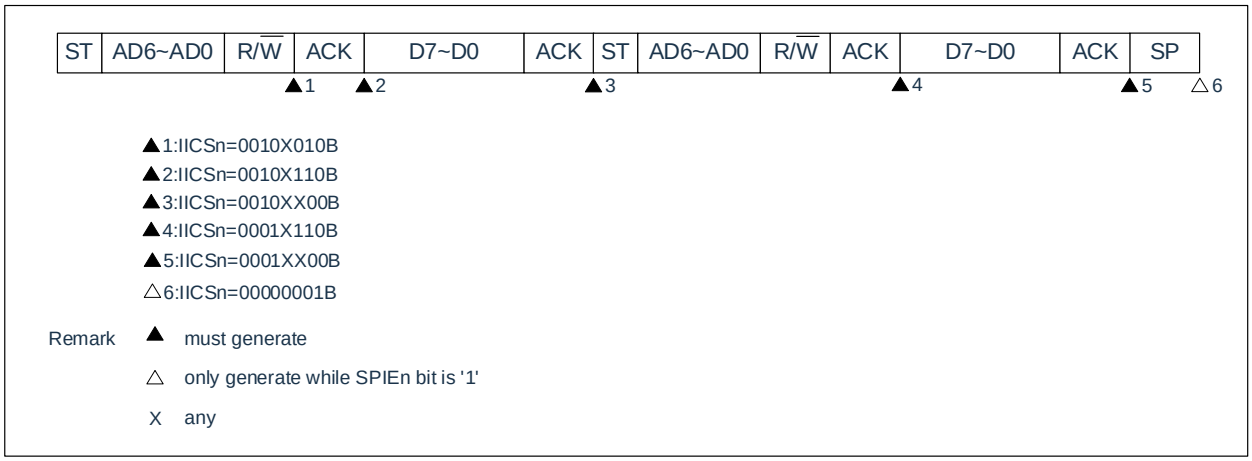
Remarks: n=0

(b) Start~Code~Data~Start~Address~Data~Stop

(i) The case of WTIMn=0 (SVAn is the same after the restart)



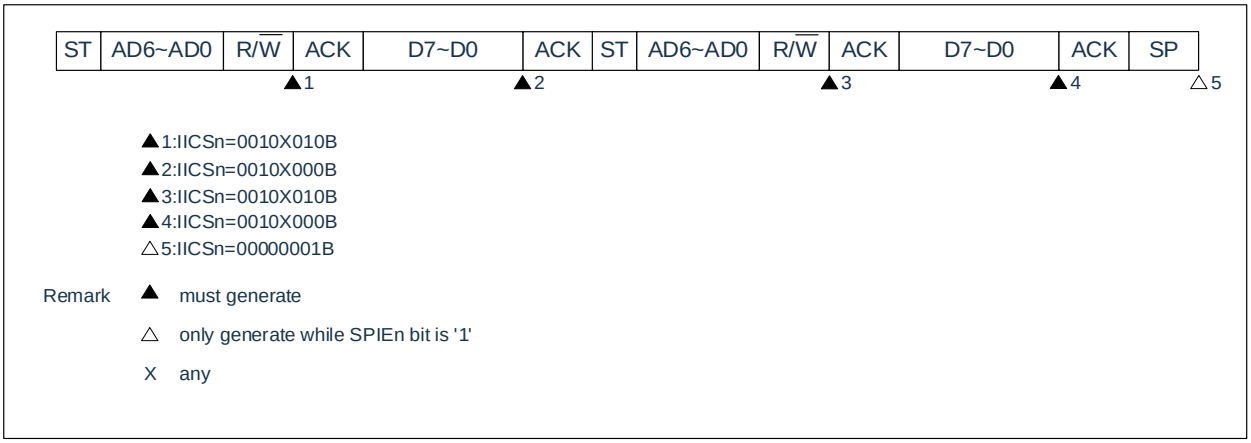
(ii) the situation of WTIMn=1 (same as SVAn upon reinitiation)



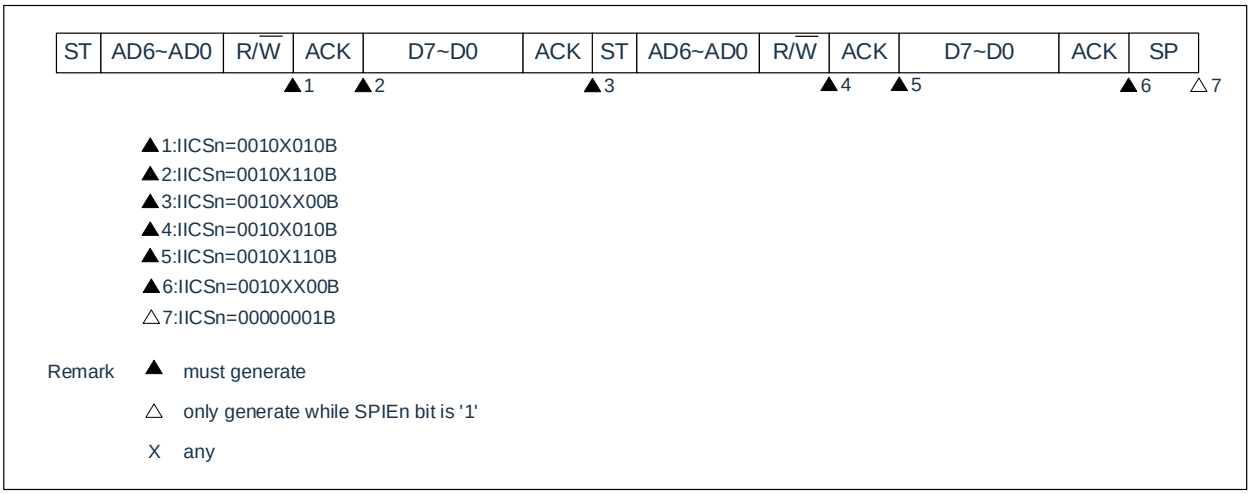
Remarks: n=0

(c) Start~Code~Data~Start~Code~Data~Stop

(i) The case of WTIMn=0 (receive an extension after restart)



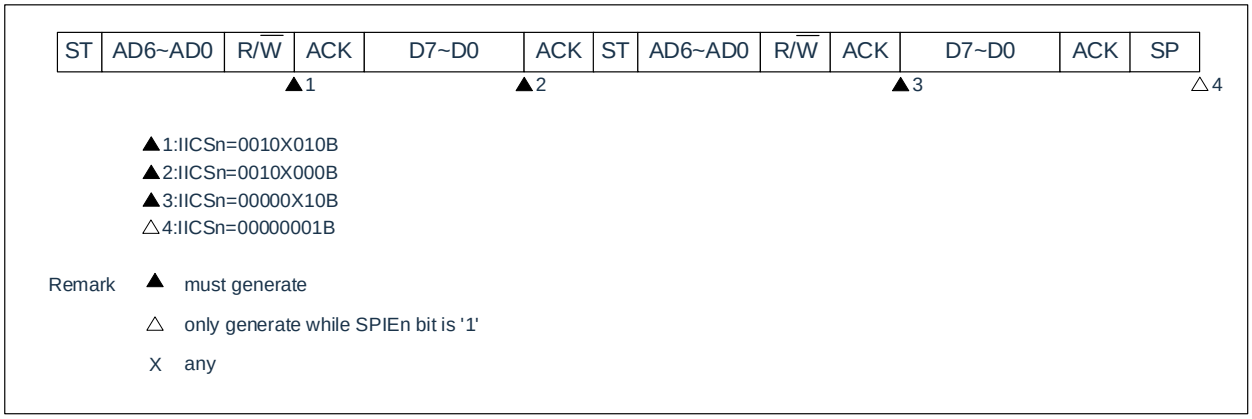
(ii) the case of WTIMn=1 (receiving an extension upon restart)



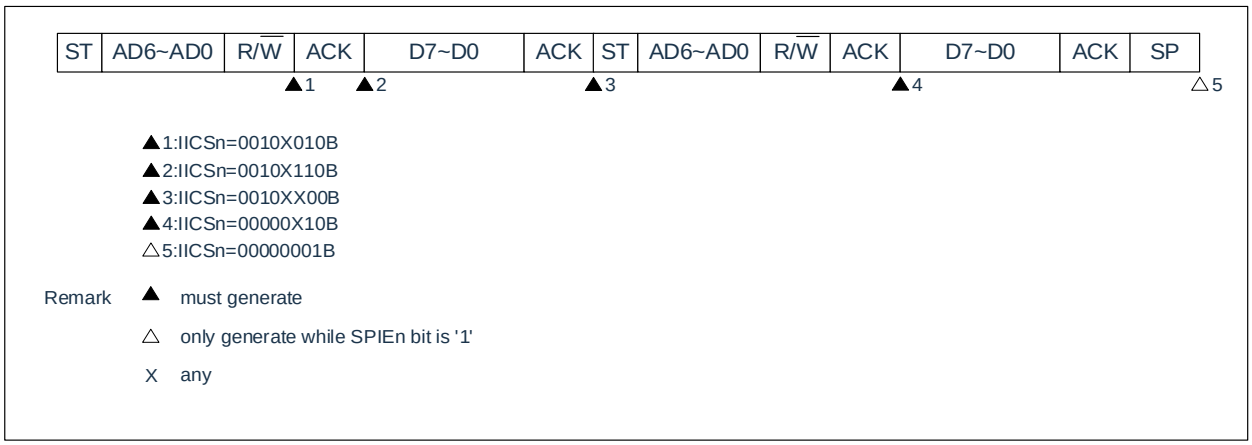
Remarks: n=0

(d) Start~Code~Data~Start~Address~Data~Stop

(i) The case of WTIMn=0 (addresses differ after restart (non-expander))



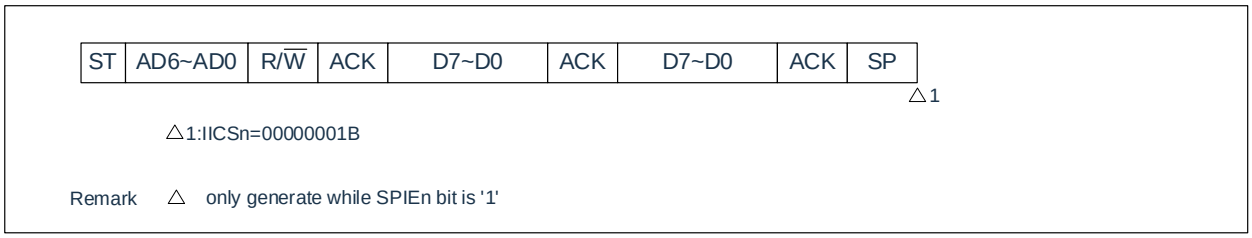
(ii) the case of WTIMn=1, with a different address (non-expander) after the restart



Remarks: n=0

(4) Not participating in the operation of the communication

(a) Start~Code~Data~Data~Stop

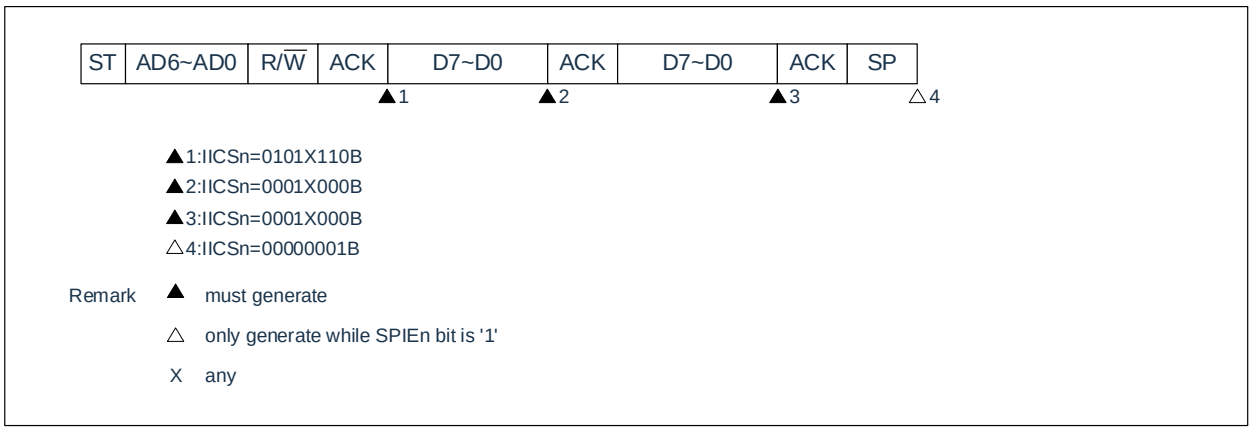


(5) The run of the arbitration failure (run as a slave after the arbitration failure)

When used as a master device in a multi-master system, the MSTSn bit must be read each time a INTIICAn interrupt request signal is generated to confirm arbitration.

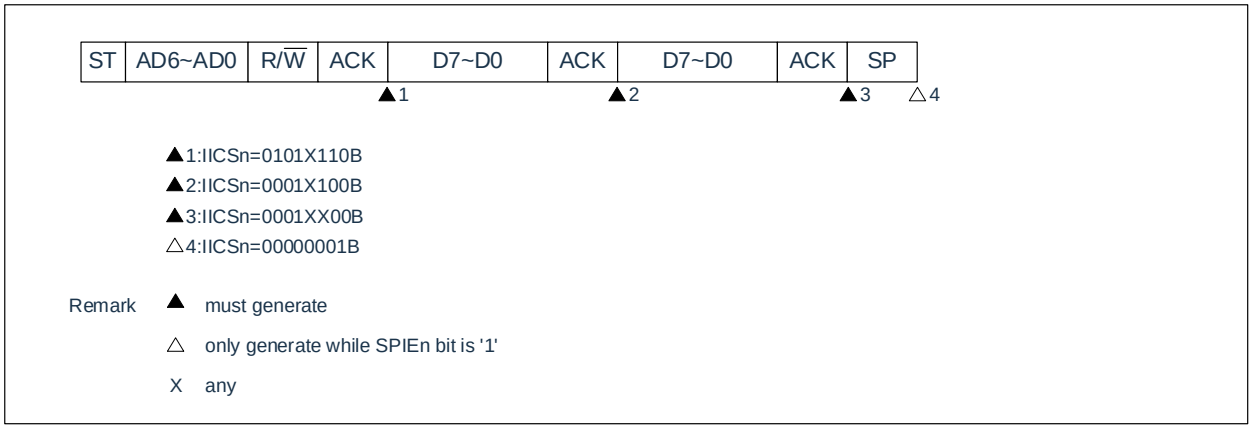
(a) Arbitration failure during the sending of secondary address data

(i) WTIMn=0's situation



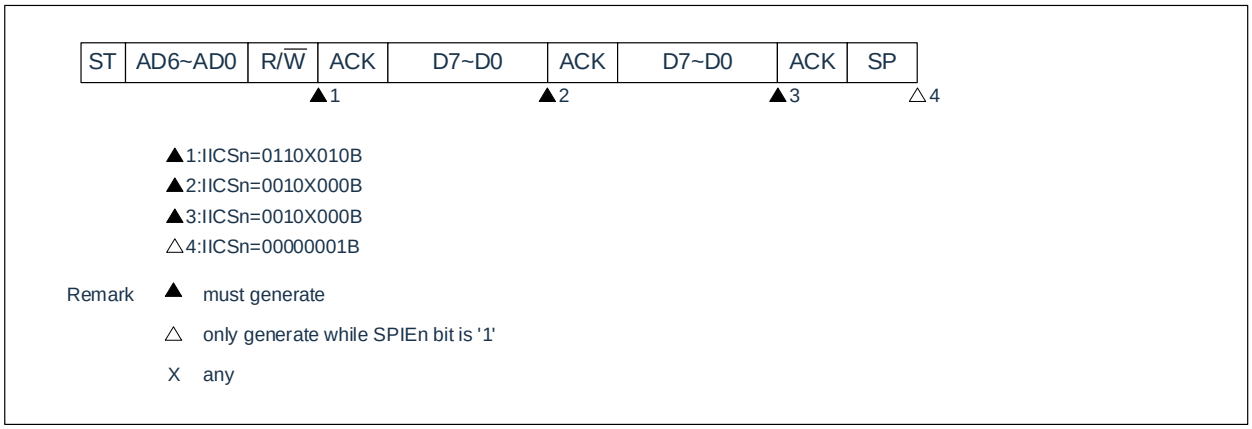
Remarks: n=0

(ii) WTIMn=1's situation



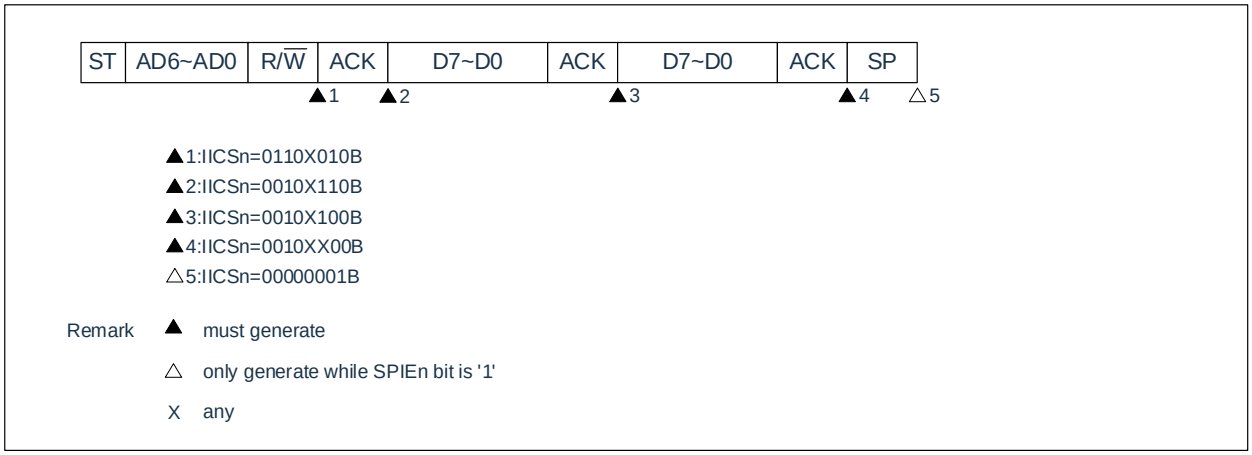
(b) Arbitration failure during the sending of an extension code

(i) WTIMn=0's situation



Remarks: n=0

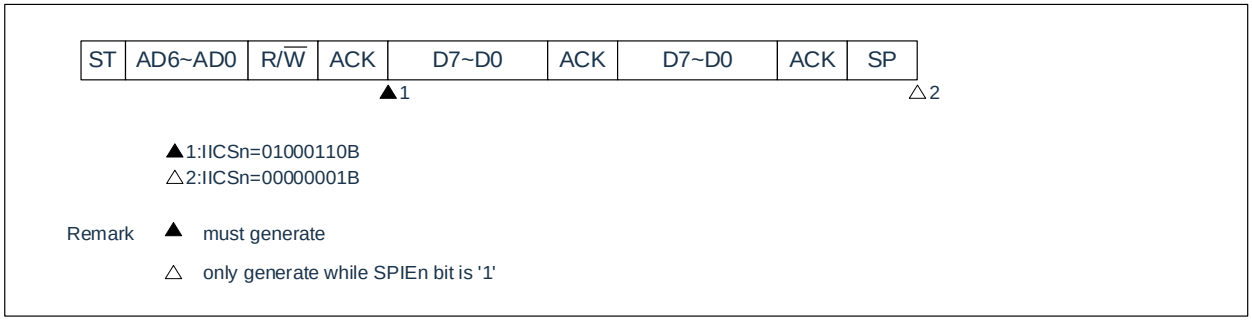
(ii) WTIMn=1's situation



(6) Run of arbitration failure (do not participate in communication after arbitration failure)

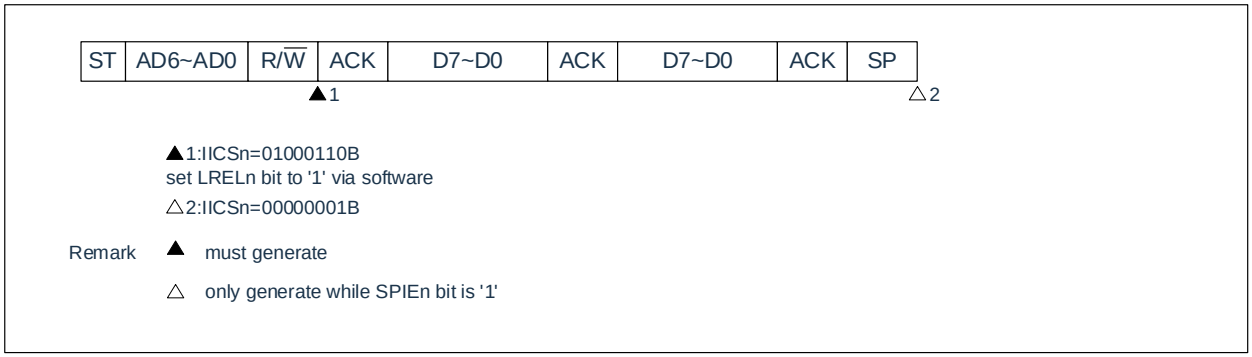
When used as a master device in a multi-master system, the MSTSn bit must be read each time a INTIICAn interrupt request signal is generated to confirm arbitration.

(a) The case of a quorum failure during the sending of secondary address data (WTIMn=1)



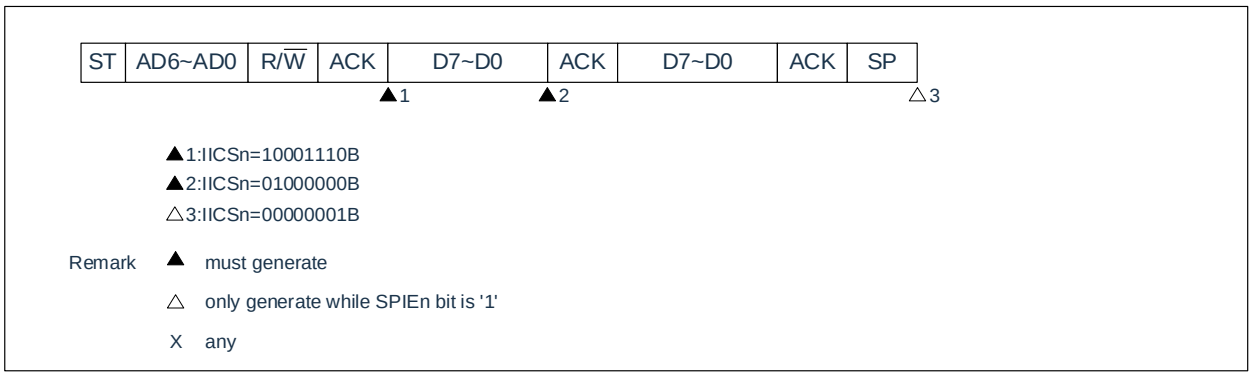
Remarks: n=0

(b) Arbitration failure during the sending of an extension code



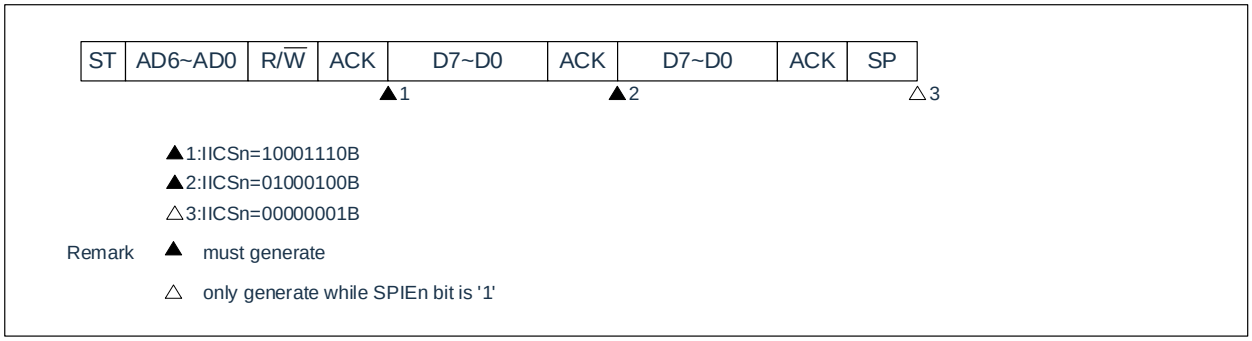
(c) Arbitration failure when transferring data

(i) WTIMn=0's situation



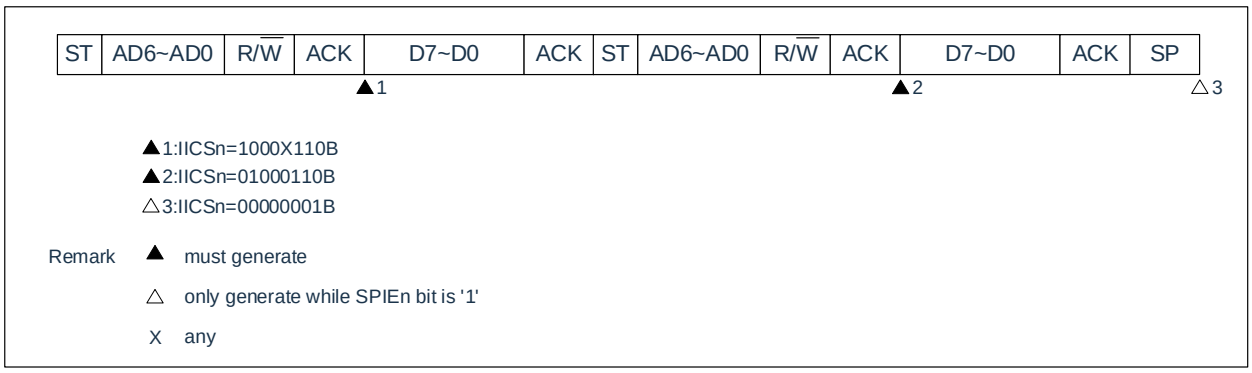
Remarks: n=0

(ii) WTIMn=1's situation



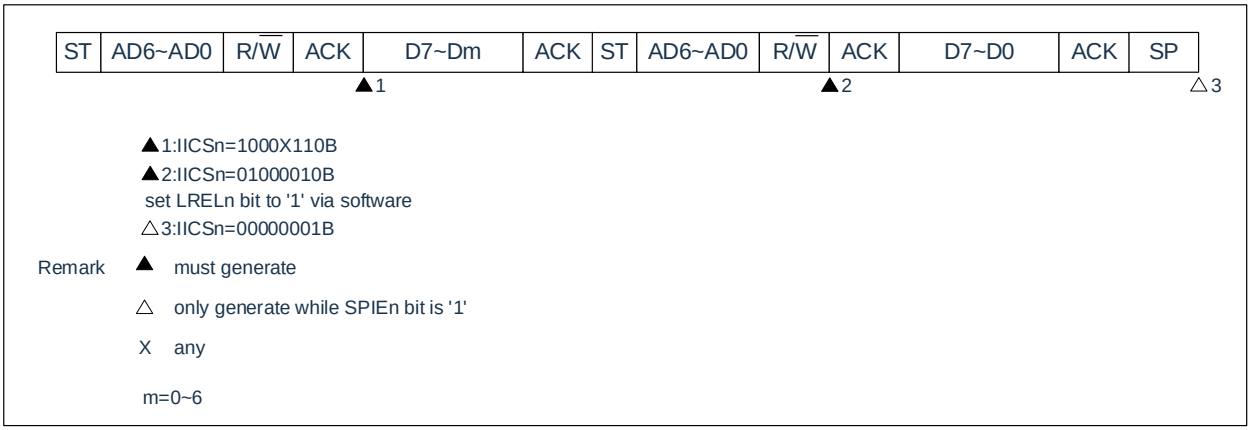
(d) A situation where arbitration fails due to restart conditions when data is transmitted

(i) Non-expander (e.g., SVAn is different)

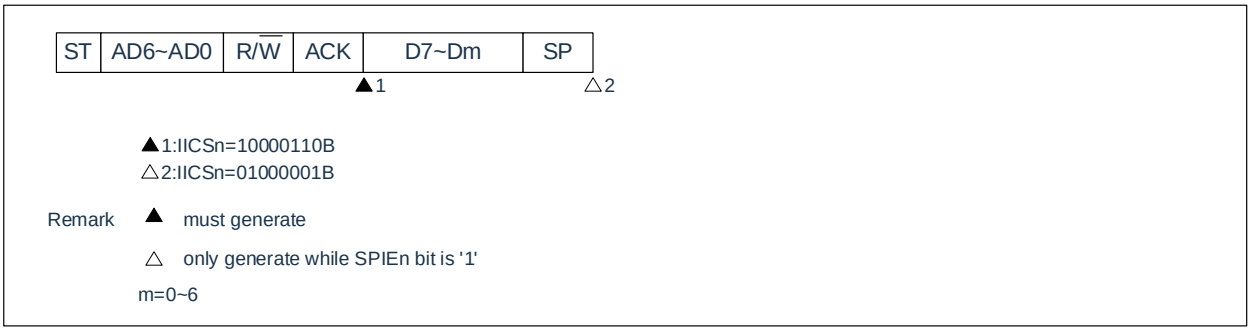


Remarks: n=0

(ii) Expander



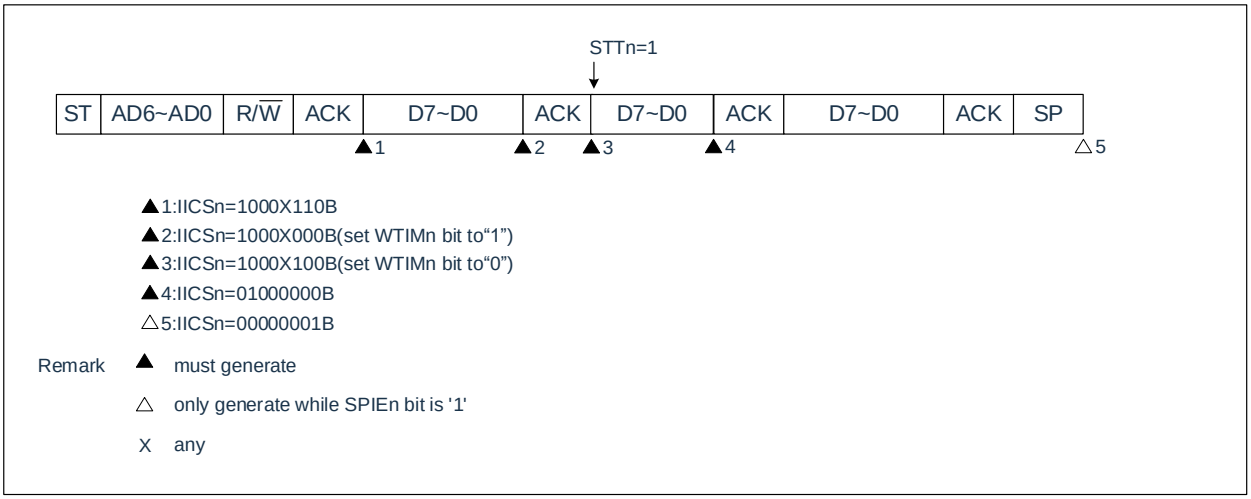
(e) A situation in which arbitration fails due to a stop condition when transferring data



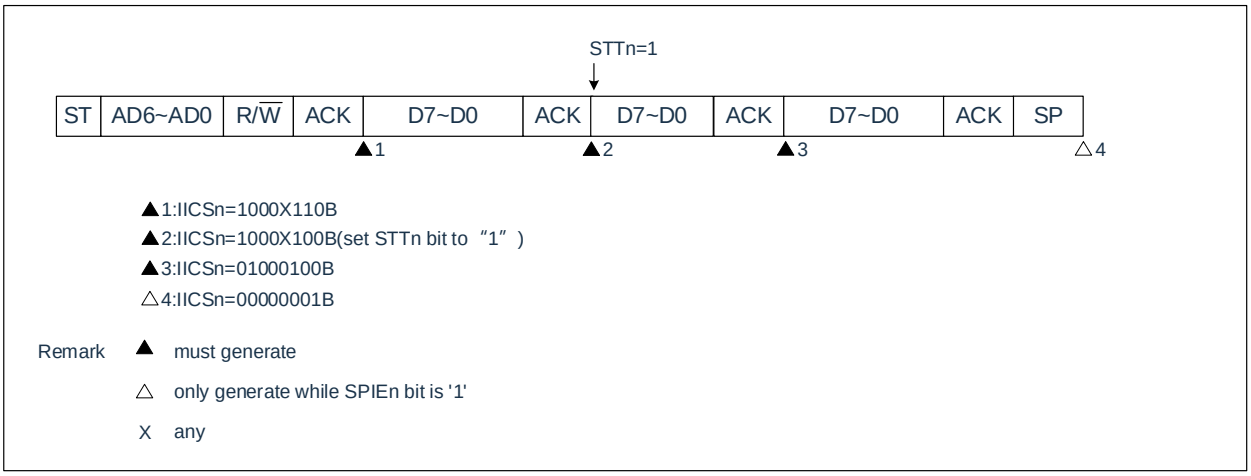
Remarks: n=0

(f) A case where arbitration fails because the data is low when a restart condition is to be generated

(i) WTIMn=0's situation



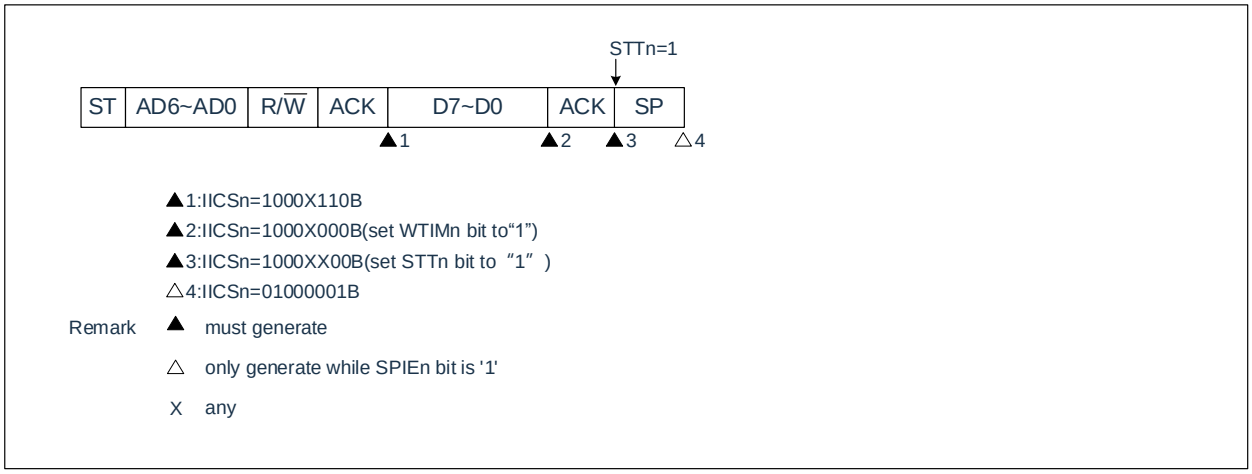
(ii) WTIMn=1's situation



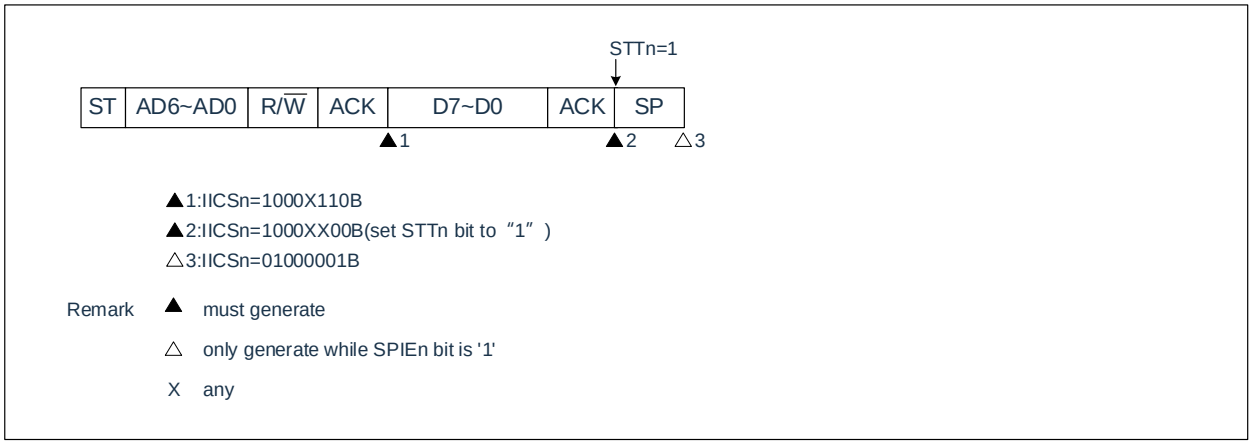
Remarks: n=0

(g) A situation where arbitration fails due to a stop condition when a restart condition is to be generated

(i) WTIMn=0's situation



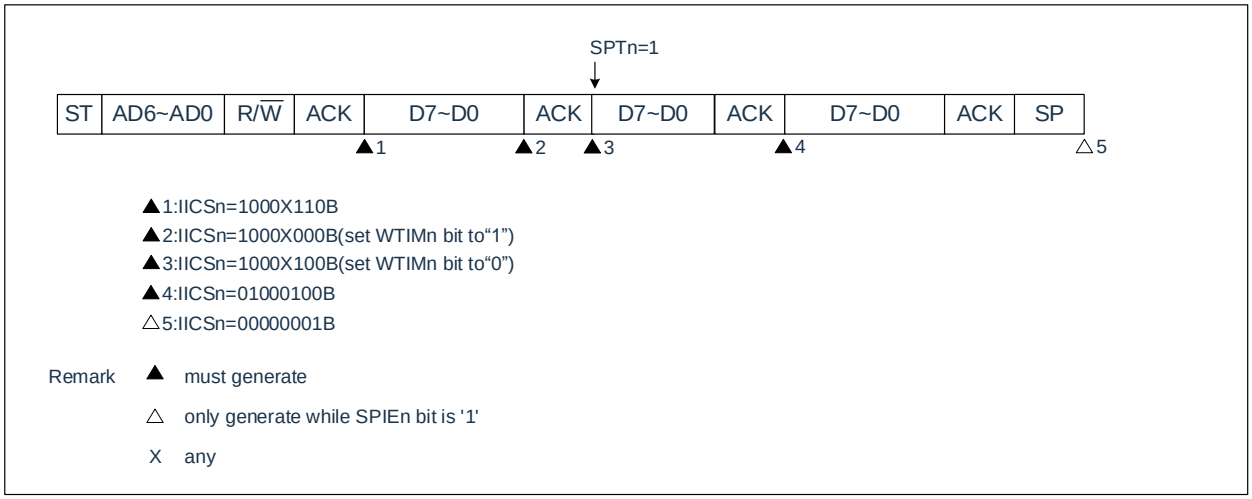
(ii) WTIMn=1's situation



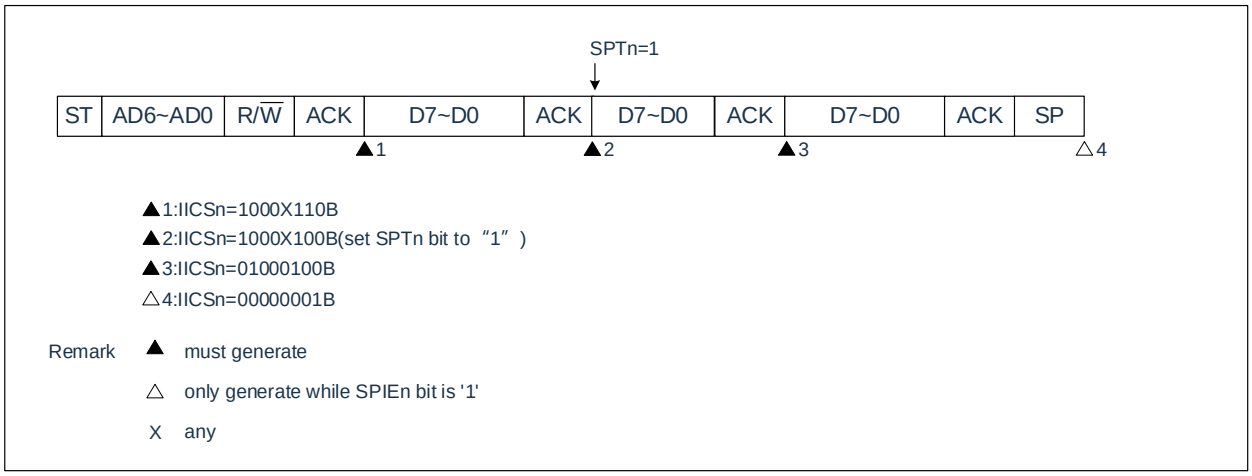
Remarks: n=0

(h) A case where arbitration fails because the data is low when a stop condition is to be generated

(i) WTIMn=0's situation



(ii) WTIMn=1's situation



Remarks: n=0

20.6 Timing diagram

In I2C bus mode, a master device selects a slave device as the communication object among a number of slave devices by outputting an address to a serial bus. The master device sends a TRCn bit (bit3 of the IICA status register n (IICSn) representing the data transmission direction after the slave device address and starts serial communication with the slave device. The timing diagram of the data communication is shown in Figure 20-32 and Figure 20-33.

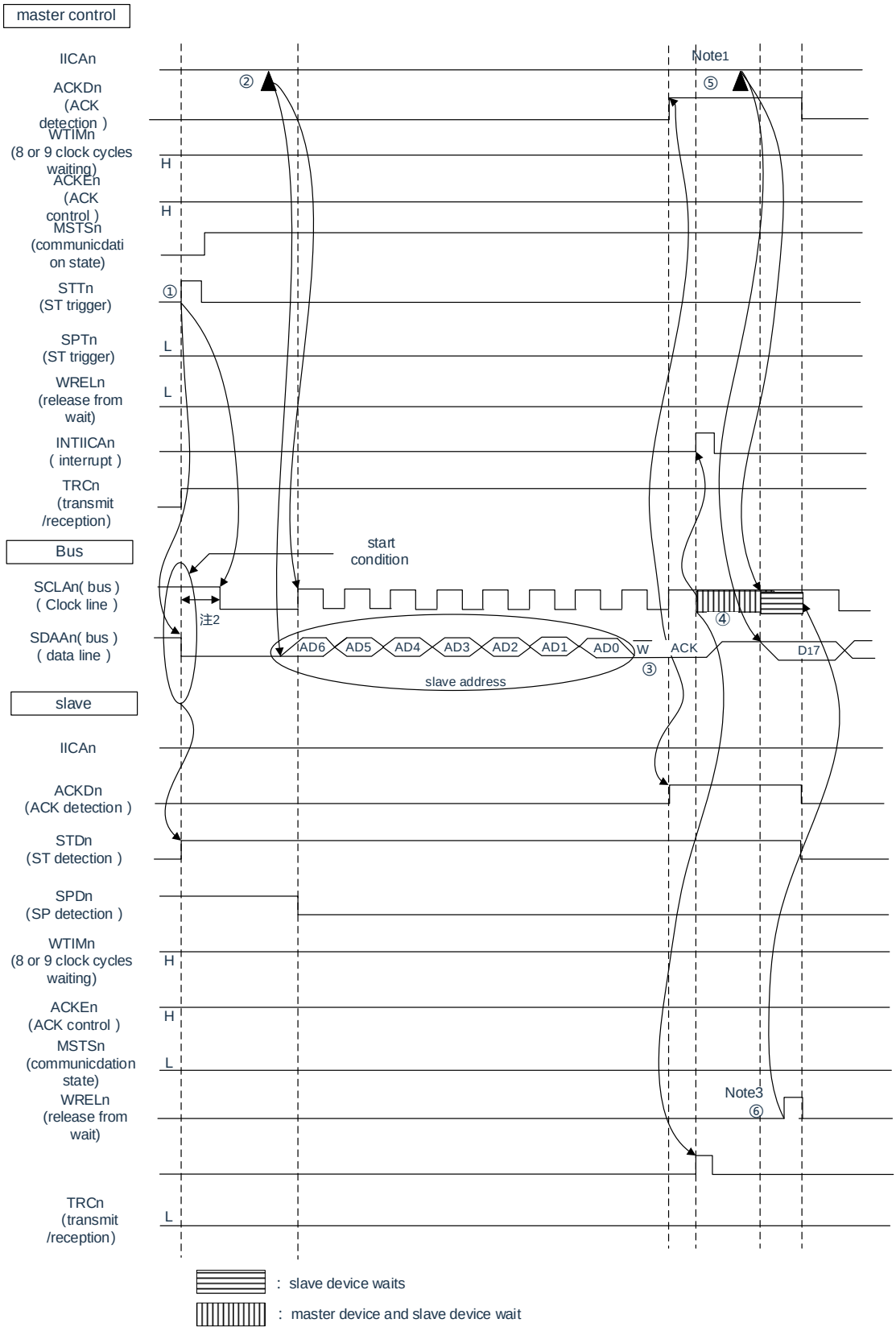
The IICA shift register n (IICAn) is shifted synchronously with the descent edge of the serial clock (SCLAn), and the transmission data is transmitted to the SO latch preferentially from the SDAAn pin.

Take the data entered by the SDAAn pin to IICAn at the rising edge of the SCLAn.

Remarks: n=0

Figure 20-32 Communication example of master device slave
(Main control equipment: Select 9 clock wait, slave: Select 9 clock wait) (1/4)

(1) Start Condition ~ Address ~ Data



Note 1. To unblock the wait during a master send, you must write data to IICAn instead of placing the WRELn location bit.

2. The time to decrease the SDAAn pin signal to the SCLAn pin signal is at least 4.0μs when set to standard mode and at least 0.6μs when set to fast mode.
3. To relieve the wait during dependent receipt, you must either place the IICAn "FFH" or place the WRELn position.

FIG. 20-32 Descriptions of (1) to (6) of (1) starting condition ~ address ~ data are as follows:

①If the master triggers the setting (STTn=1), the bus data line (SDAAn) drops and generates a start condition (SDAAn changes from 1 to 0). After that, if the start condition is detected, the master enters the master communication state (MSTSn=1), and after holding time.

②If the master sends the IICA shift register n(IICAn) a write address + W (send), the slave address is sent.

③On the secondary side, if the receiving address and the local station address (value of SVAn) are the same [note], an ACK is sent to the host via hardware. The master detects ACK (ACKDn=1) at the rising edge of the 9th clock.

④The master generates an interrupt on the descending edge of the 9th clock (INTIICAn: End of address delivery interrupt). A slave device of the same address enters a waiting state (SCLAn=0) and an interrupt is generated (INTIICAn: Address Matching Interrupt) [Note].

⑤The host writes sending data to the IICAn register, thus relieving the host from waiting.

⑥If the client unwaits (WRELn=1), the master starts transferring data to the client.

Note: If the address sent and the slave address are different, the dependent party does not return ACK(NACK:SDAAn=1) to the host party and does not generate INTIICAn interrupts (address matching interrupts) and does not enter a wait state.

However, the master generates an INTIICAn interrupt for both the ACK and the NACK (the end of address sending interrupt).

Note 1. (1) of Figure 20-32①～⑮It is a series of operation steps of data communication through I2C bus.

FIG. 20-32' (1) Start Condition~Address~Data' illustrates steps ①～⑥.

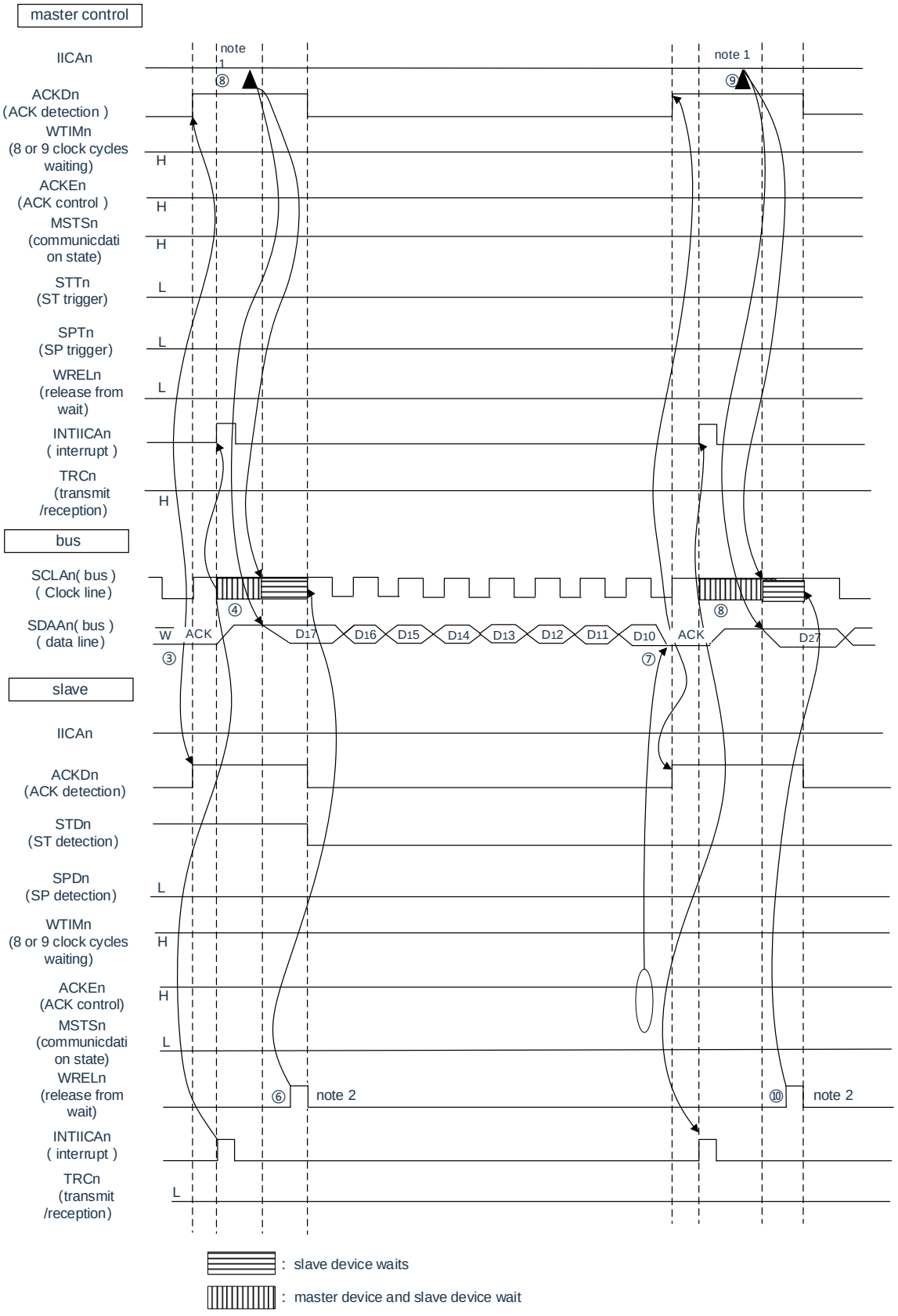
FIG. 20-32' (2) Address~Data~Data' Explanation Steps ③～⑩.

FIG. 20-32 "(3) Data ~ Data ~ Stopping Conditions" Explanation of Step ⑦～⑮.

2.n=0

Figure 20-32 Communication example of master device slave
(Main control equipment: Select 9 clock wait, slave: Select 9 clock wait) (2/4)

(2) Address~Data~Data



Note 1. To unblock the wait during a master send, you must write data to IICAn instead of placing the WRELn location bit.

2. To relieve the wait during dependent receipt, you must either place the IICAn "FFH" or place the WRELn position.

FIG. 20-32 Descriptions of (3) to (10) for (2) Address ~ Data ~ Data:

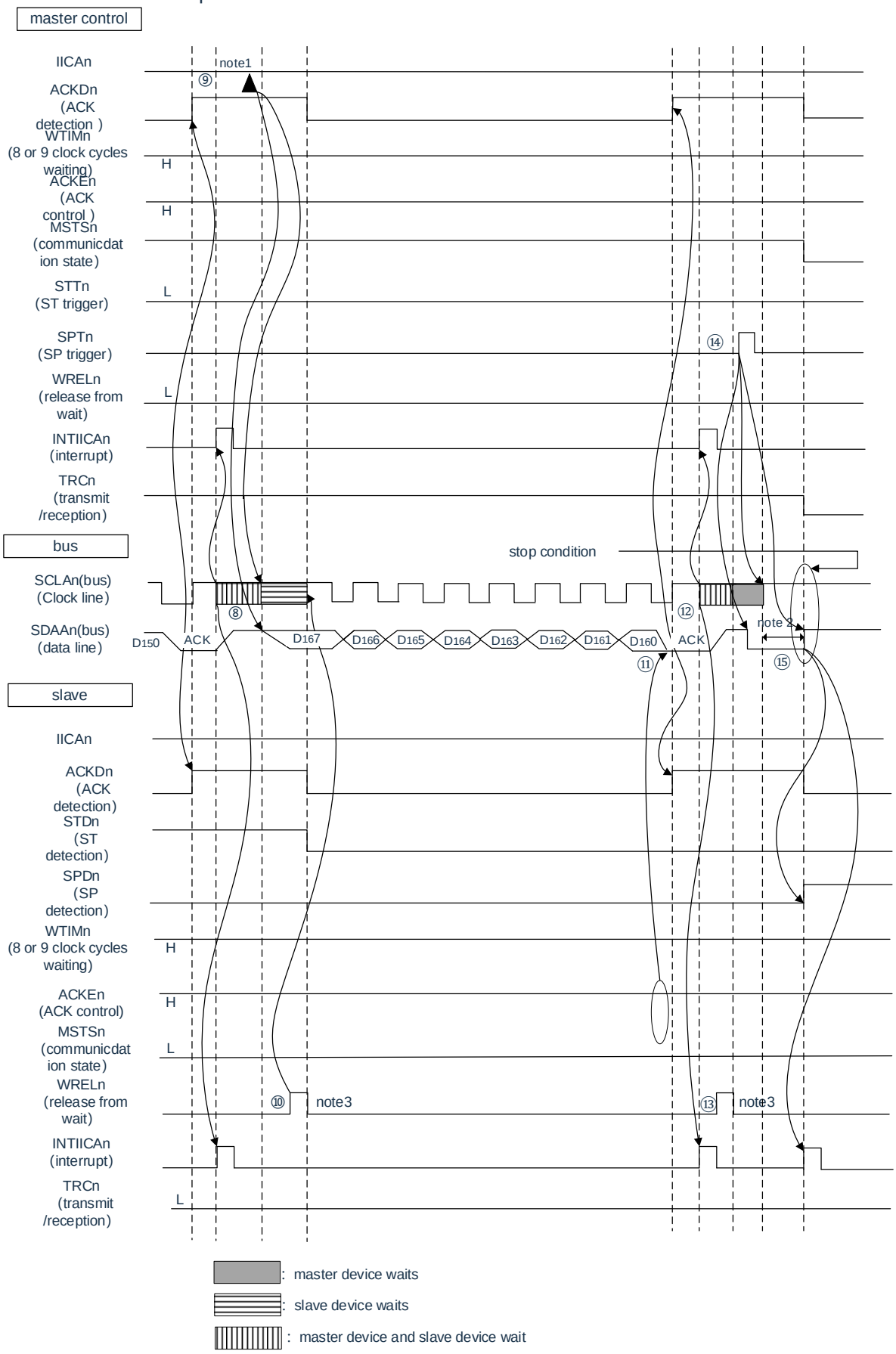
(3) In the secondary party, if the receiving address and the local station address (value of SVAn) are the same note, the ACK is sent to the master via hardware. The master detects ACK (ACKDn=1) at the rising edge of the 9th clock.
(4) The master generates an interrupt at the descending edge of the 9th clock (INTIICAn: End of address delivery interrupt). A slave device of the same address enters a waiting state (SCLAn=0) and an interrupt is generated (INTIICAn: Address Matching Interrupt) Note.
(5) The main controller writes and transmits data to IICA shift register n (IICAn), and eliminates the main controller waiting.
(6) If the dependent party unwaits (WRELn=1), the master starts transferring data to the dependent party.
(7) After the data transmission is finished, the slave side transmits ACK through hardware to the master side because the ACKEn bit is 1. The master detects ACK (ACKDn=1) at the rising edge of the 9th clock.
(8) The master and slave enter a waiting state (SCLAn=0) at the descending edge of the ninth clock, and both generate interrupts (INTIICAn: End of delivery interrupt).
(9) The main control side writes sending data to the IICAn register to release the main control side waiting.
(10) If the slave reads the received data and relieves the wait (WRELn=1), the master starts to transmit data to the slave.

Note If the address sent is different from the slave address, the slave does not return ACK (NACK:SDAAn=1) to the master and does not generate INTIICAn interrupts (address matching interrupts) or enter a wait state.
However, the master generates an INTIICAn interrupt for both the ACK and the NACK (the end of address sending interrupt).

Note 1. (1) of Figure 20-32 ①～⑮It is a series of operation steps of data communication through I2C bus.
FIG. 20-32' (1) Start Condition~Address~Data' illustrates steps ①～⑥.
FIG. 20-32' (2) Address~Data~Data' Explanation Steps ③～⑩.
FIG. 20-32' (3) Data-Data-Stopping Condition' illustrates steps ⑦～⑮.
2.n=0

Figure 20-32 Communication example of master device slave
(Main control equipment: Select 9 clock wait, slave: Select 9 clock wait) (3/4)

(14) Data~Data~Stop Condition



Note 1. To unblock the wait during a master send, you must write data to IICAn instead of placing the WRELn location bit.

2. The time from the SCLAn pin signal to generating the stop condition after issuing the stop condition is at least 4.0μs when set to standard mode and at least 0.6μs when set to fast mode.

3. To relieve the wait during dependent receipt, you must either place the IICAn "FFH" or place the WRELn position.

FIG. 20-32 (7)-15 of "3Data-Data-Stopping Condition" are described as follows:

⑦After the data transfer is finished, the slave's ACKEn bit is "1", so the master sends ACK through hardware. The master detects ACK (ACKDn=1) at the rising edge of the 9th clock.
⑧The master and the slave enter a waiting state (SCLAn=0) at the descending edge of the 9th clock, and both generate interrupts (INTIICAn: End of delivery interrupt).
⑨The master sends data to the IICA shift register n(IICAn) to release the master from waiting.
⑩If the slave reads the received data and the wait is WRELn=1, the master starts transferring data to the slave.
⑪After the data transfer is completed, the slave (ACKEn=1) sends an ACK to the master via the hardware. The master detects ACK (ACKDn=1) at the rising edge of the 9th clock.
⑫The master and the slave enter a waiting state (SCLAn=0) at the descending edge of the 9th clock, and both generate interrupts (INTIICAn: End of delivery interrupt).
⑬The slave reads the received data and unwaits (WRELn=1).
⑭If the master triggers the SPTn=1, the bus data line SDAAn=0 is cleared and the bus clock line is set SCLAn=1.
⑮If a stop condition is generated, the dependent party detects the stop condition and generates an interrupt (INTIICAn: Stop condition interrupt).

Note 1. (1) Figure 20-32 ①～⑮ is a series of operation steps of data communication through I2C bus.

FIG. 20-32' (1) Start Condition~Address~Data' illustrates steps (①～⑥.

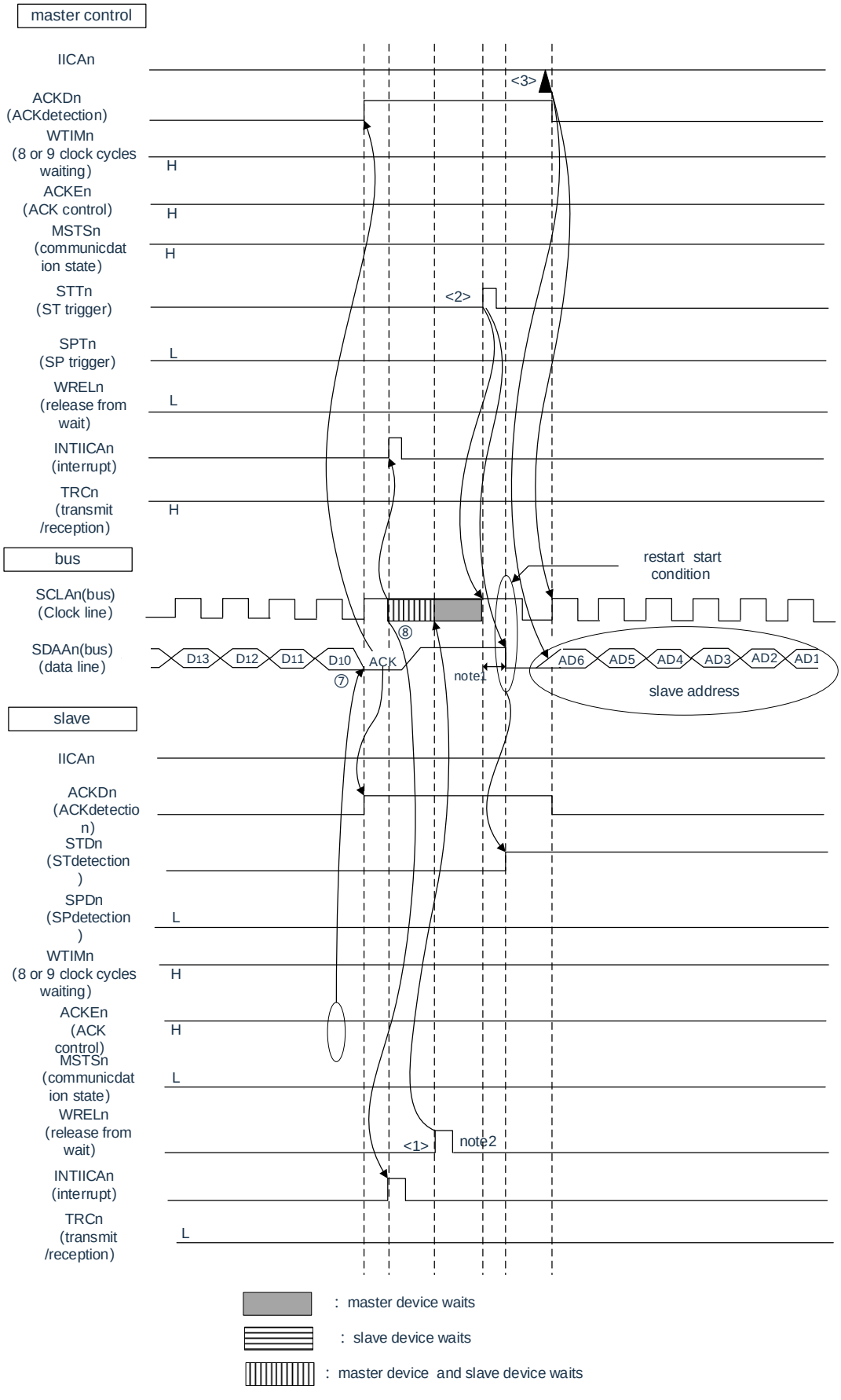
FIG. 20-32' (2) Address~Data~Data' Explanation Steps ③～⑩.

FIG. 20-32 '(3) Data-Data-Stopping Condition' Explanation Steps ⑦～⑮ .

2.n=0

Figure 20-32 Communication example of master device slave
(Main control equipment: Select 9 clock wait, slave: Select 9 clock wait) (4/4)

(4) Data ~ Restart Condition ~ Address



Note 1. The time from the SCLAn pin signal to the generation start condition after the release restart condition is at least 4.7 s in standard mode and at least 0.6 s in fast mode.

2. To relieve the wait during dependent receipt, you must either place the IICAn "FFH" or place the WRELn position.

FIG. 20-32 The operation instructions for "(4) Data~Restart Condition~Address" are as follows. After steps (7) and (8) are executed, <1>~<3> is executed, thereby returning to data sending step (3).

(7) After the data transmission is finished, the slave side transmits ACK through hardware to the master side because the ACKEn bit is 1. The master detects ACK (ACKDn=1) at the rising edge of the 9th clock.

(8) The master and slave enter a waiting state (SCLAn=0) at the descending edge of the ninth clock, and both generate interrupts (INTIICAn: End of delivery interrupt).

<1> The client reads the received data and unwaits (WRELn=1).

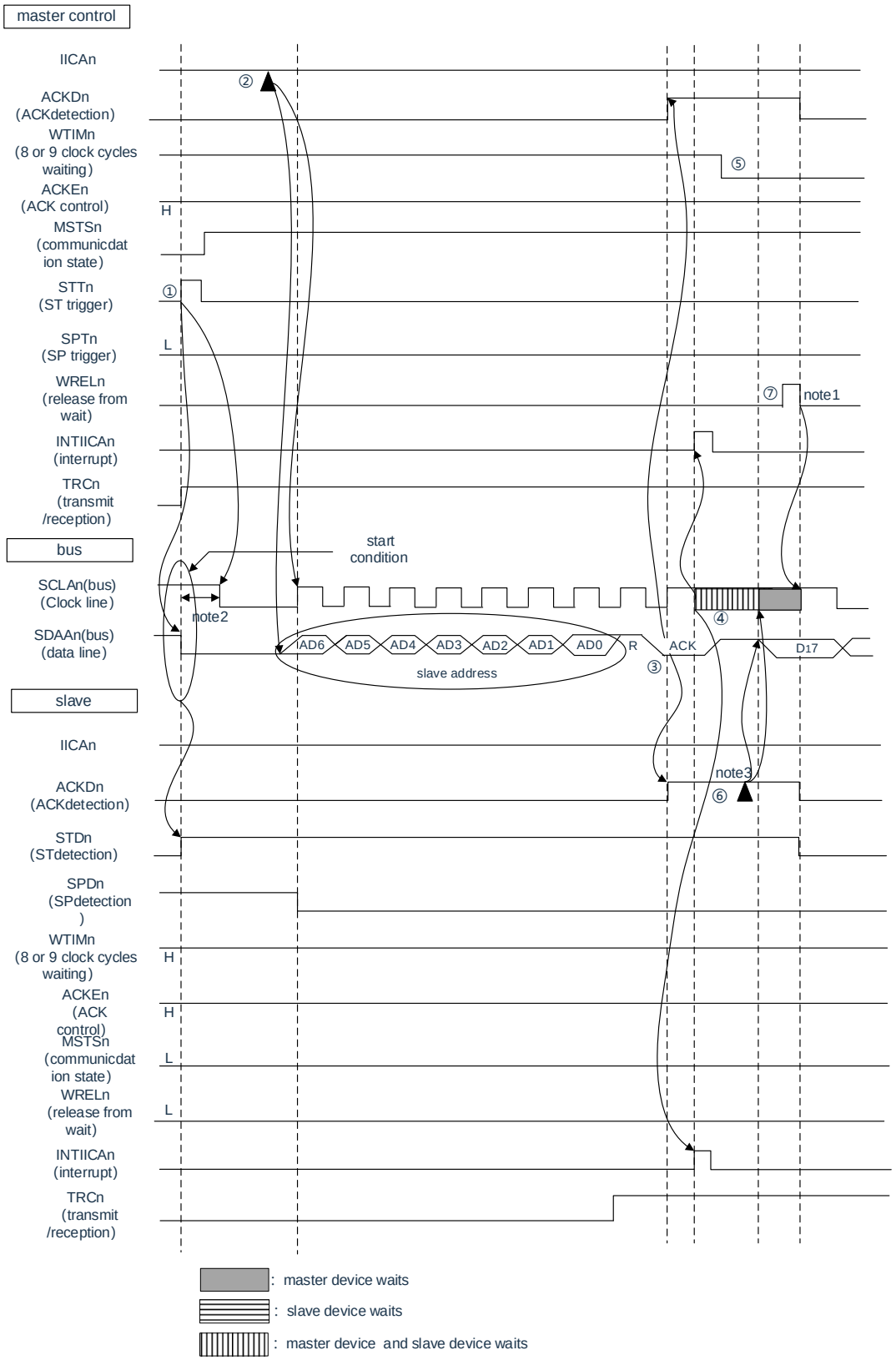
<2> If the master triggers the start condition (STTn=1) again, the bus clock line rises (SCLAn=1) and the bus data line falls (SDAAn=0) after the preparation time of the restart condition to generate the start condition (0). Then, if a start condition is detected, the bus clock line descends (SCLAn=0) after a hold time, and communication preparations are ended.

<3> If the master sends the IICA shift register n(IICAn) a write address +R/W, the slave address is sent.

Remark n=0

Figure 20-33 Example of communication for a slave master device
(Main control equipment: Select 8-clock wait, slave: Select 9 clock wait) (1/3)

(1) Start Condition ~ Address ~ Data



Note 1. To unwait during receiving by the master, you must either place the IICAn in "FFH" or WRELn position.

2. The time to decrease the SDAAn pin signal to the SCLAn pin signal is at least 4.0μs when set to standard mode and at least 0.6μs when set to fast mode.

3. To relieve the wait during a slave send, you must write data to IICAn instead of placing the WRELn position bit.

FIG. 20-33 Descriptions of (1) to (7) of "1th Condition ~ Address ~ Data" are as follows:

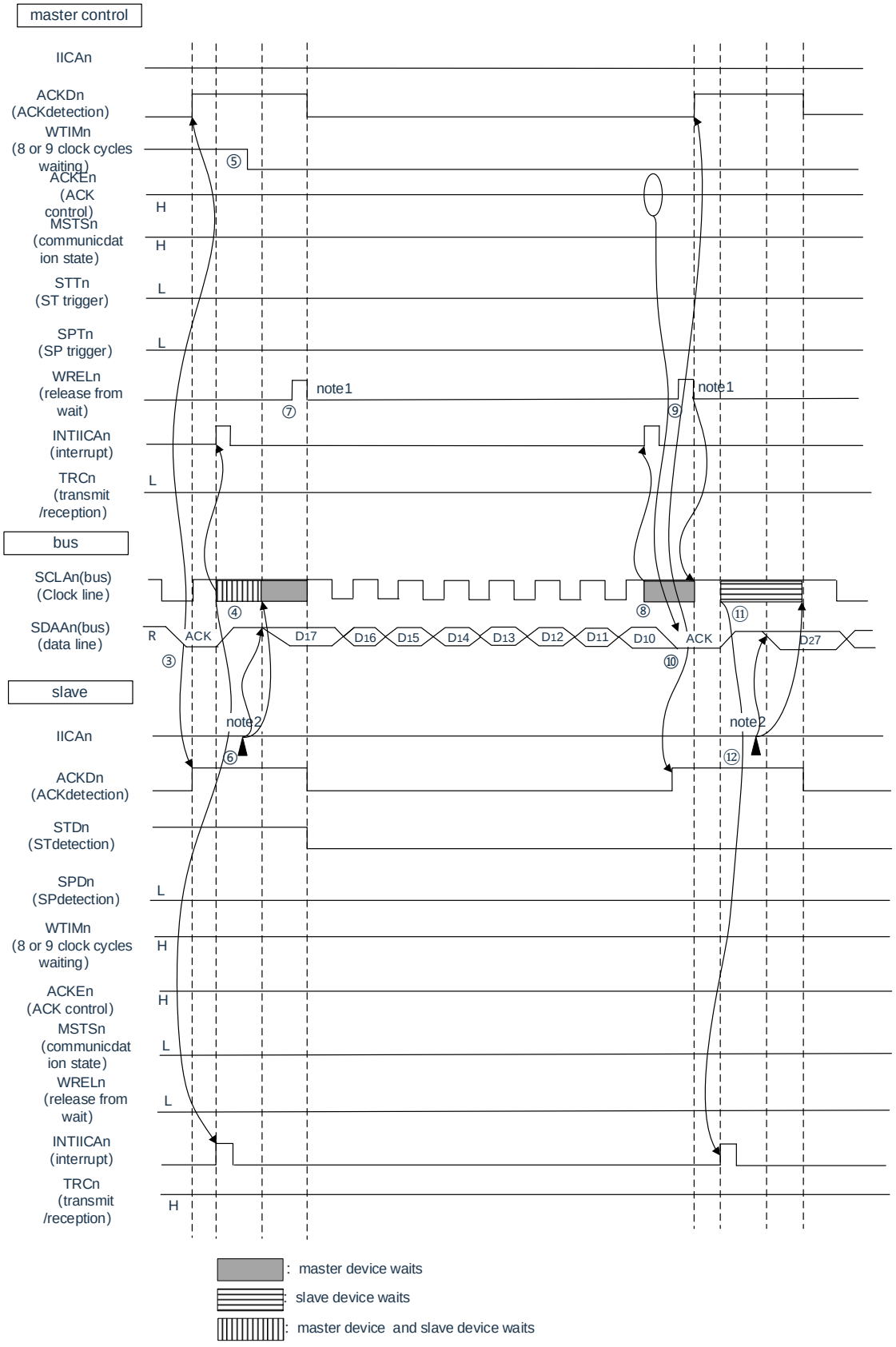
(1) If the master triggers the setting (STTn=1), SDAAn descends to generate the starting condition (SCLAn=1 changes SDAAn from "1"). After that, if the start condition is detected, the master enters the master communication state (MSTSn=1), and the bus clock line descends (SCLAn=0) after holding time.
(2) If the master sends the IICA shift register n(IICAn) a write address + R (receive), the slave address is sent.
(3) In the secondary party, if the receiving address and the local station address (value of SVAn) are the same note, the ACK is sent to the master via hardware. The master detects ACK (ACKDn=1) at the rising edge of the 9th clock.
(4) The master generates an interrupt at the descending edge of the 9th clock (INTIICAn: End of address delivery interrupt). A slave device of the same address enters a waiting state (SCLAn=0) and an interrupt is generated (INTIICAn: Address Matching Interrupt) Note.
(5) The main controller changed the waiting time to the eighth clock (WTIMn=0).
(6) The dependent party writes sending data to the IICAn register, and the dependent party is relieved from waiting.
(7) The main control unit removes the waiting (WRELn=1) and starts the data transmission from the slave equipment.

Note If the address sent is different from the slave address, the slave does not return ACK (NACK:SDAAn=1) to the master and does not generate INTIICAn interrupts (address matching interrupts) or enter a wait state.
However, the master generates an INTIICAn interrupt for both the ACK and the NACK (the end of address sending interrupt).

Note 1. The (1) to (19) of FIG. 20-33 are a series of operational steps for data communication via I2C bus.
FIG. 20-33' (1) Start Condition ~ Address ~ Data' Explanation of Steps (1) ~ (7).
FIG. 20-33 '(2) Address~Data~Data' Explanation Steps (3~12) .
FIG. 20-33' (3) Data ~ Data ~ Stopping Conditions' Explaining Steps (8) ~ (19) .
2.n=0

Figure 20-33 Example of communication for a slave master device
(Main control equipment: Select 8-clock wait, slave: Select 9 clock wait) (2/3)

(2) Address~Data~Data



Note 1. To unwait during receiving by the master, you must either place the IICAn in "FFH" or WRELn position.

2. To relieve the wait during a slave send, you must write data to IICAn instead of placing the WRELn position bit.

FIG. 20-33 (3)- (12) of "2) Address-Data" are described as follows:

(3) In the secondary party, if the receiving address and the local station address (value of SVAn) are the same note, the ACK is sent to the master via hardware. The master detects ACK (ACKDn=1) at the rising edge of the 9th clock.

(4) The master generates an interrupt at the descending edge of the 9th clock (INTIICAn: End of address delivery interrupt). A slave device of the same address enters a waiting state (SCLAn=0) and an interrupt is generated (INTIICAn: Address Matching Interrupt) Note.

(5) The main controller changed the waiting time to the eighth clock (WTIMn=0).

(6) The dependent party writes the transmission data to the IICA shift register n(IICAn), and relieves the dependent party.

(7) The main control unit removes the waiting (WRELn=1) and starts the data transmission from the slave equipment.

(8) The master enters a waiting state (SCLAn=0) at the descending edge of the eighth clock and generates an interrupt (INTIICAn: End of delivery interrupt). Because the ACKEn bit of the master is "1", the slave is sent an ACK through the hardware.

(9) The main control side reads the received data and relieves the waiting (WRELn=1).

(10) The slave detects an ACK (ACKDn=1) at the rising edge of the 9th clock.

(11) The slave enters a waiting state (SCLAn=0) at the descent edge of the 9th clock and generates an interrupt (INTIICAn: End of delivery interrupt).

(12) If the slave writes sending data to the IICAn register, the slave is relieved from waiting and the data transfer from the slave to the master is started.

Note: If the address sent and the slave address are different, the dependent party does not return ACK(NACK:SDAAn=1) to the host party and does not generate INTIICAn interrupts (address matching interrupts) and does not enter a wait state.

However, the master generates an INTIICAn interrupt for both the ACK and the NACK (the end of address sending interrupt).

Note 1. The (1) to (19) of FIG. 20-33 are a series of operational steps for data communication via I2C bus.

FIG. 20-33' (1) Start Condition ~ Address ~ Data' Explanation of Steps (1) ~ (7).

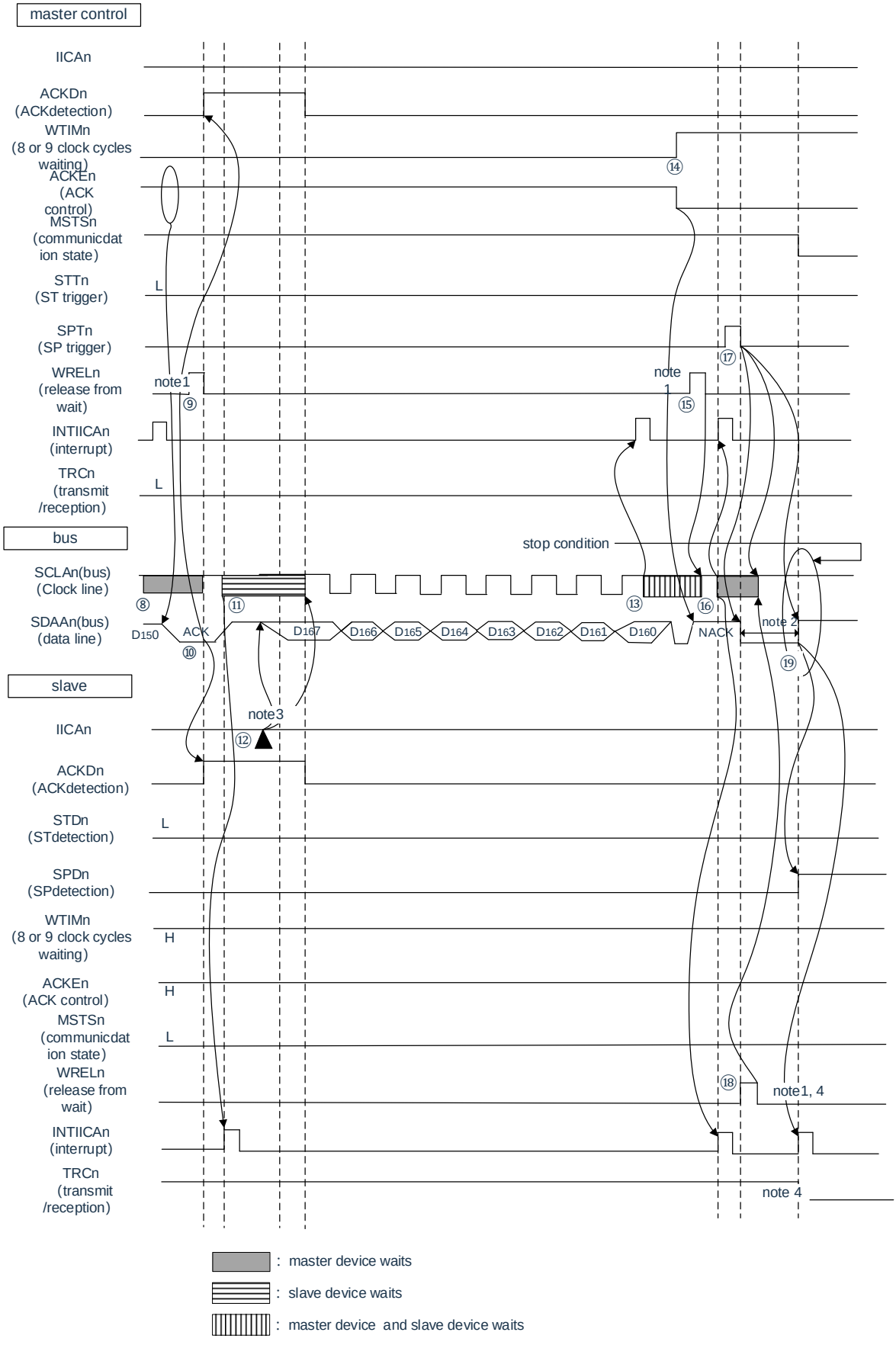
FIG. 20-33 '(2) Address~Data~Data' illustrates steps (3~12).

FIG. 20-33' (3) Data-Data-Stopping Condition' explains steps (8)-19.

2.n=0

Figure 20-33 Example of communication for a slave master device (Main control equipment: Select 8 9 clock waits, slave: Select 9 clock waits) (3/3)

(3) Data~Data~Stop Condition



Note 1. To unwait, you must either place the IICAn in "FFH" or place the WRELn position.

2. The time from the SCLAn pin signal to generating the stop condition after issuing the stop condition is at least 4.0μs when set to standard mode and at least 0.6μs when set to fast mode.
3. To relieve the wait during a slave send, you must write data to IICAn instead of placing the WRELn position bit.
4. Clear the WRELn bit during the sending from the parent if the wait is canceled by the TRCn bit placement.

FIG. 20-33 Descriptions of (8) to (19) for (3) Data~Data~Stopping Conditions are as follows:

⑧. The master enters a waiting state (SCLAn=0) at the descending edge of the eighth clock and generates an interrupt (INTIICAn: End of delivery interrupt). Because the ACKEn bit of the master is "0", the slave is sent an ACK through the hardware.

⑨. The master reads the received data and unwaits (WRELn=1).

⑩. The slave detects ACK (ACKDn=1) at the rising edge of the 9th clock.

⑪. The slave enters a waiting state (SCLAn=0) along the descending edge of the 9th clock, and generates an interrupt (INTIICAn: End of delivery interrupt).

⑫. If the slave writes sending data to the IICA shift register n (IICAn), the slave waits and starts data transmission from the slave to the master.

⑬. The master generates an interrupt on the descending edge of the 8th clock (INTIICAn: End of Transfer interrupt) and enter a wait state (SCLAn=0). Because of the ACK control (ACKEn=1), the bus data line at this stage becomes low level (SDAAn=0).

⑭. The master is set to NACK Acknowledgement (ACKEn=0) and changes the wait time to the ninth clock (WTIMn=1). If the master unwaits (WRELn=1), the slave detects NACK (ACKDn=0) on the rising edge of the 9th clock.

⑮. The master and the slave enter a waiting state (SCLAn=0) at the descending edge of the 9th clock, and both generate interrupts (INTIICAn: End of delivery interrupt).

⑯. If the master issues a stop condition (SPTn=1), the bus data line (SDAAn=0) is cleared and the master waits. After that, the master is in standby until the bus clock line is SCLAn=1.

⑰. The slave stops sending after confirming the NACK, and in order to end the communication, the wait is canceled (WRELn=1). If the dependent party is relieved of waiting, 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

⑲. (SDAAn=1), and then issue the stop condition (SDAAn changed from "0" to "1" via SCLAn=1). If a stop condition is generated, the dependent party detects the stop condition and generates an interrupt (INTIICAn: Stop condition interrupt).

Chapter 21 CAN Controller

21.1 Summary Description

The chip has on-chip CAN controller (controller LAN) functionality and complies with the standard CAN protocol of ISO 11898.

21.1.1 feature

- ISO 11898 compliant and tested against ISO/DIS 16845 (CAN compliance)
- Standard and extended frames for receiving and transmitting
- Up to 1 Mbps. (CAN Input Clock 8 MHz)
- 1 channel with 16 message caches
- Receive/Send History List Feature
- automatic block transfer function
- Multi-cache receive block functionality
- Masking settings for four modes per channel

21.1.2 Functional Overview

Table 21-1 lists the functions of the CAN controller.

Table 21-1. Functional Overview

Features	Details
agreement	ISO 11898 CAN Protocol (standard and extended frame send/receive)
baud rate	Max: 1 Mbps (CAN input clock 8MHz)
data storage	Information is stored in CAN RAM
information number	- 16 message caches per channel - Each message cache may be set as both a receive message cache and a transmit message cache
message receiving	- Unique ID can be set to each message cache. - Four Mask settings per channel - Each time a message is received and stored in the message cache, a receive interrupt is generated. - More than two receive message caches can be used as receive FIFO (multi-cache receive block function) - Receive history list feature
message sending	- The Unique ID may be set to each message cache - Each cache can be associated to a send complete interrupt - Message buffers 0 through 7 are designated as transmission message buffers and can be used for automatic block transmission. Message transfer intervals are programmable (Automatic Block Transfer Function (ABT) - Transfer history list feature
remote frame processing	Remote frame processing via message cache transfer
timestamp function	- When using a 16-bit timer, you can set a timestamp function for message reception. Timestamp capture triggers are optional (can detect the SOF or EOF of a message frame)
diagnostic function	- Readable error counter - - "Valid Protocol Operation Flags" for verifying bus connections - Only mode - single-shot mode - Decoding of CAN protocol error types - self-test mode
Release from bus shutdown state	- The software can set a mandatory release from the bus shutdown state (ignoring timing constraints) - Do not automatically release from bus shutdown state (software settings must be re-enabled).
power saving mode	- CAN sleep mode (can be woken by CAN bus) - CAN Stop Mode (cannot be woken by CAN bus)

21.1.3 configuration

The CAN controller consists of the following four modules

(1) interaction

This module provides an internal interactive bus to send and receive signals between the CAN module and the host.

(2) Memory Control Module (MCM)

This block controls access to the CAN RAM in the CAN protocol layer and the CAN module.

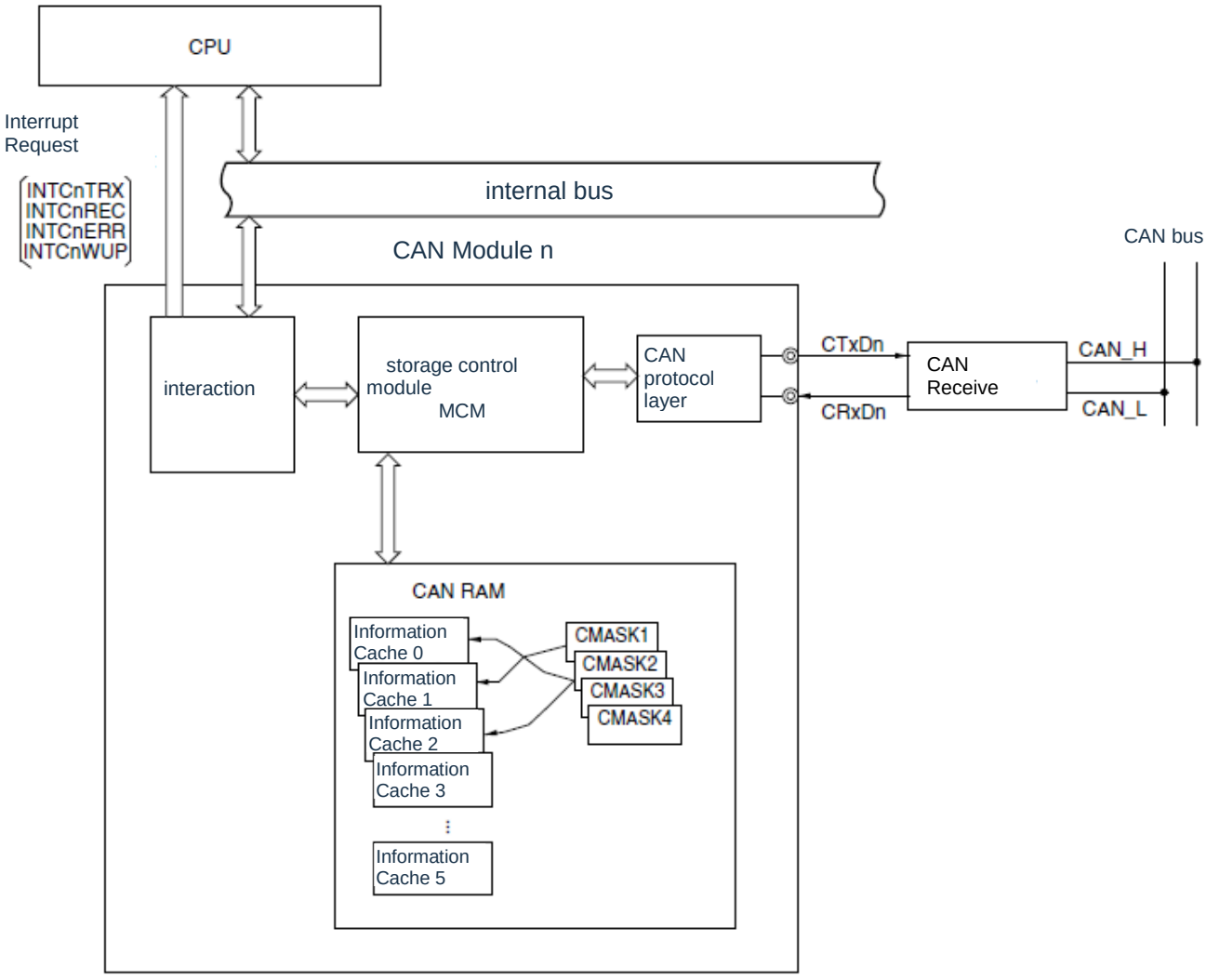
(3) CAN protocol layer

This block of functionality involves the operation of the CAN protocol and its associated settings. .

(4) CAN RAM

This is a CAN memory function block for storing message ID, message data, etc.

Figure 21-1. Block diagram of the CAN module n



(n = 0)

21.2 CAN protocol

CAN (Controller Area Network) is a high speed multi-channel communication protocol for real-time communication in automotive applications. CAN is specified by ISO 11898. For more information, see the ISO 11898 specification.

CAN specification is usually divided into two layers: The physical layer and the data link layer, in turn the data link layer contains logical links and media access control. The composition of these layers is shown as follows:

Figure 21-2. Composition of layers

High

Low

data link Tier[1]	· Logical Link Control (LLC)	· acceptance filter · overload notification · recovery management
	· Media Access Control (MAC)	· Data Packaging/Unpackaging · Frame encoding (fill/erase fill) · media access management · error detection and calibration · Error Reporting · Answer · Serialized/Unserialized
physical layer		Provisions for signal level and bit descriptions

1: CAN Controller Specification

21.2.1 frame format

(1) standard frame format

- The standard frame format uses 11-bit identifiers, which means it can process 2048 signals.

(2) extended format frame

- The extended format frame uses a 29-bit (11-bit + 18-bit) identifier, which handles 2048 x 218 signals.
- - Sets the extended format frame when the SRR and IDE bits are set to CMOS equal to 1 in the arbitrated field.

21.2.2 Frame Type

The four frame types in the following table are used in the CAN protocol.

Table 21-2. Frame Types

Frame Type	Description
data frame	Frames for transmitting data
remote frame	Signal to apply for a data frame
error frame	Signal for reporting errors
overload frame	Used to set a delay to wait for the next data frame or remote frame

(3) bus value

The bus value is divided into explicit and implicit

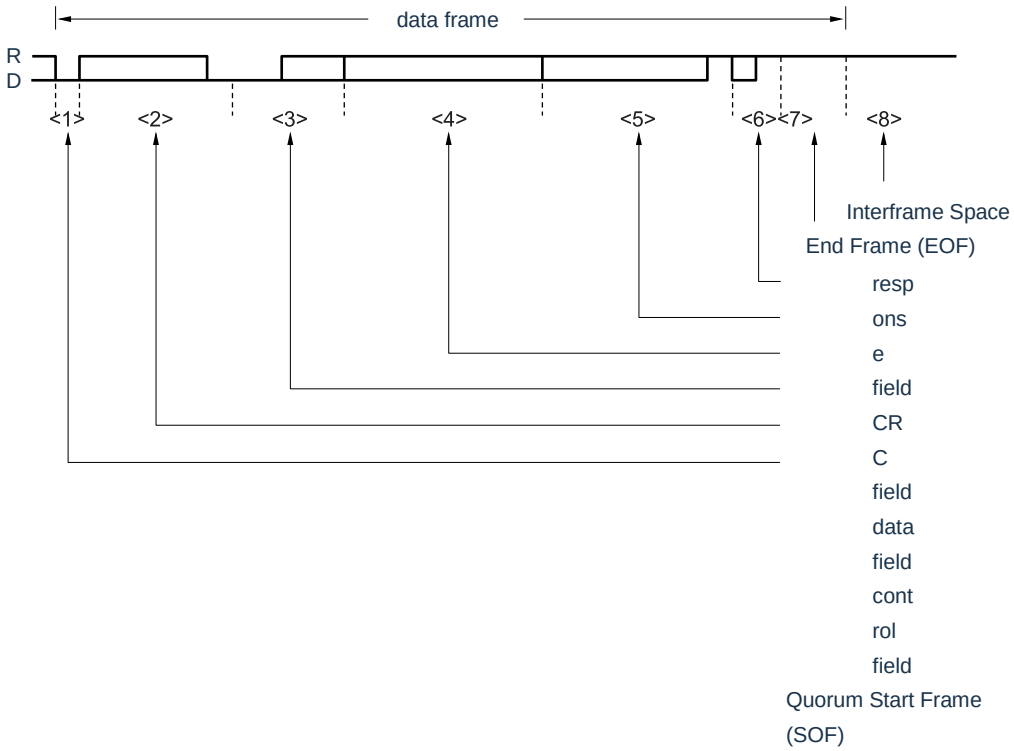
- The explicit value is logical 0
- Implicit value is logical 1
- When the bus is explicit and implicit at the same time, the status of the bus is explicit

21.2.3 data frames and remote frames

(4) data frame

The data frame consists of seven different bit fields

Figure 21-3. Data frame



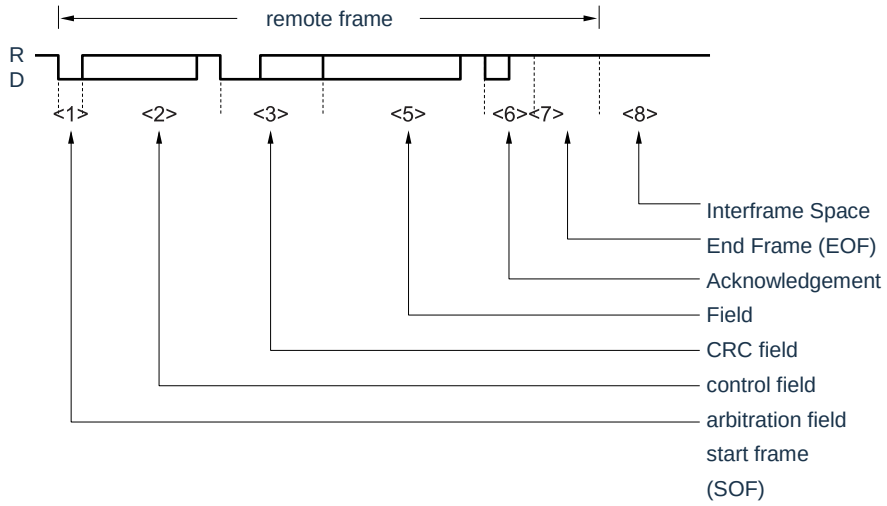
Note D: Explicit=0

R: Implicit=1

(5) remote frame

The remote frame is composed of 6 bit fields.

Figure 21-4. Remote frames



Note 1. Even if the data length of the control bit field is not "0000B", the data bit field will not be sent.

2. D: Explicit=0

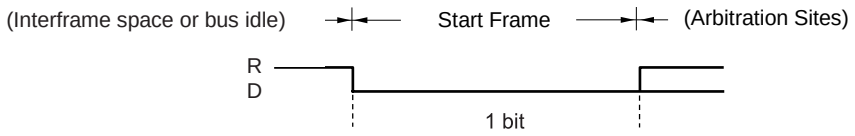
R: Implicit=1

(6) potential field description

<1> Start Frame (SOF)

The start frame bit field is at the beginning of a data frame or a remote frame.

Figure 21-5. Start Frame (SOF)



Note D: Explicit=0

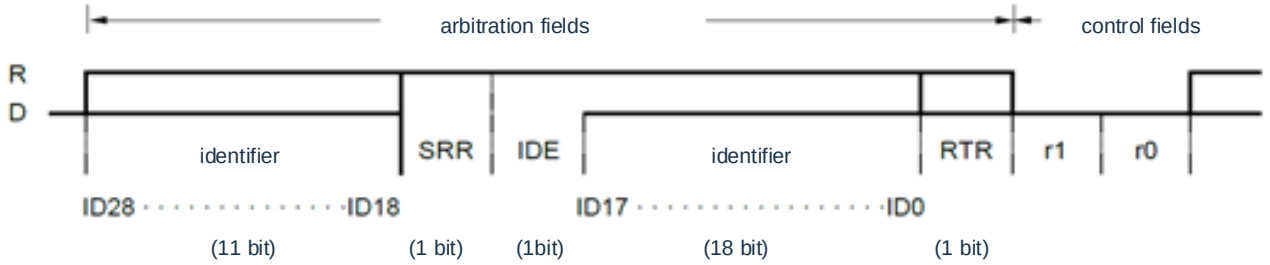
R: Implicit=1

- If dominant bits are detected in the bus idle state, hard synchronization is performed (the current TQ is specified as a SYNC segment)
- If an explicit bit is sampled at a sampling point after such a hard synchronization, the bit is assigned as SOF. If an implicit bit is detected, the protocol layer returns to the bus idle state and treats the preceding explicit pulse as interference only. In this case, no error frames are generated.

<2> Arbitration Court

Arbitrations are used to set priority, data/remote and frame formats

Figure 21-6. The Arbitral Tribunal (in Standard Format)



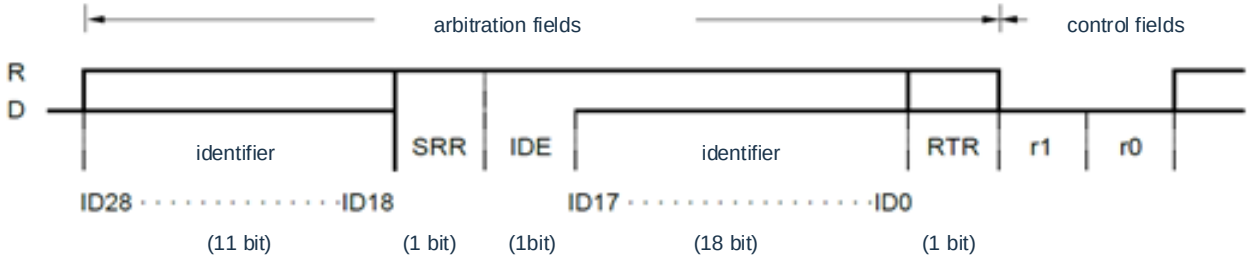
Note 1. ID18 to ID28 is an identifier.

2. The identifier is transmitted first by MSB.

Note D: Explicit=0

R: Implicit=1

Figure 21-7. Quorum (in extended frame mode)



Note 1. ID18 to ID28 is an identifier.

2. The identifier is MSB-first sent.

Note D: Explicit=0

R: Implicit=1

Table 21-3. RTR Frame Settings

Frame Type	RTR bit
data frame	0(D)
remote frame	1(R)

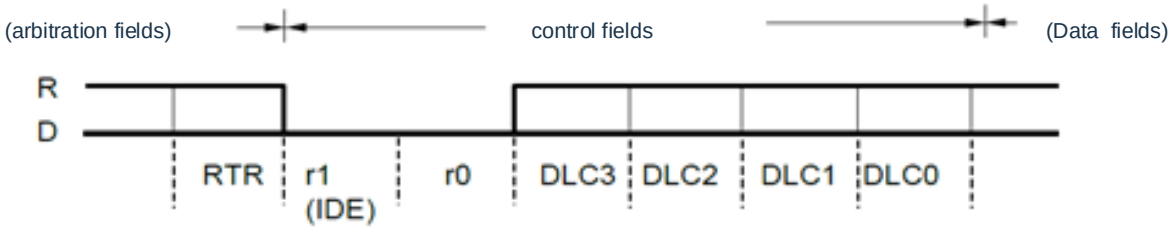
Table 21-4. Frame Formatting (IDE bits) and Identifier (ID) Bits Settings

frame format	SRR bit	IDE bit	number of digits
standard frame format	None	0(D)	11 bit
extended frame format	1(R)	1(R)	29 bit

<3> control field

The control field sets 'DLC' as the number of bytes of data in the data field (DLC=0 to 8).

Figure 21-8. Control field



Note D: Explicit=0

R: Implicit=1

In standard format frames, the IDE bit and r1 bit of the control field are the same

Table 21-5. Data length settings

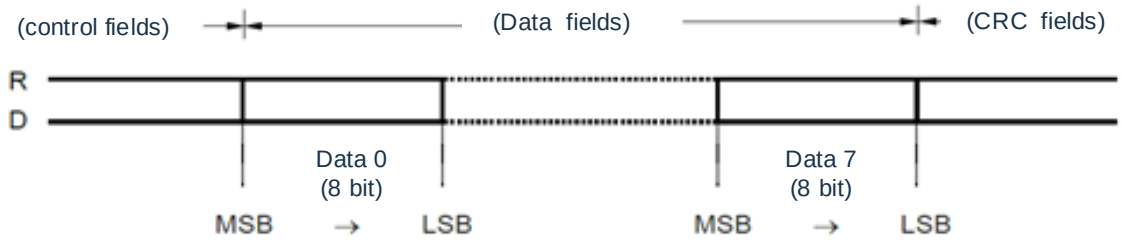
data length code				data byte calculation
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
Other Values				8 bytes (regardless of value from DLC0 to

Note: In a remote frame, there is no data field even if the data length code is not (0000B)

<4> Data Field

The data field contains the amount of data (in bytes) that controls the field settings. Up to 8 data units can be set.

Figure 21-9. Data fields



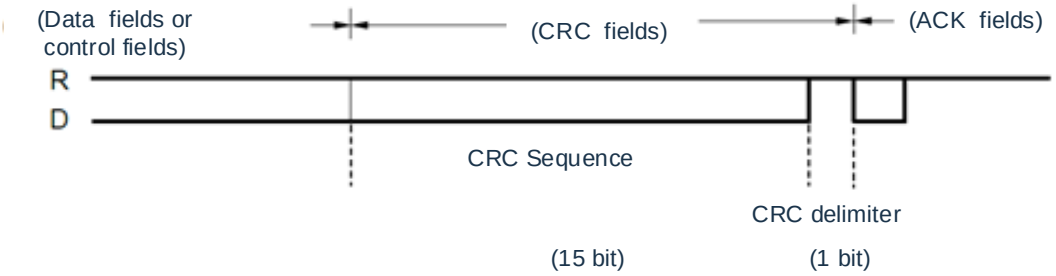
Note D: Explicit=0

R: Implicit=1

<5> CRC Field

A CRC field is a 16-bit field used to detect errors in data transfers.

Figure 21-10. CRC field



Note D: Explicit=0

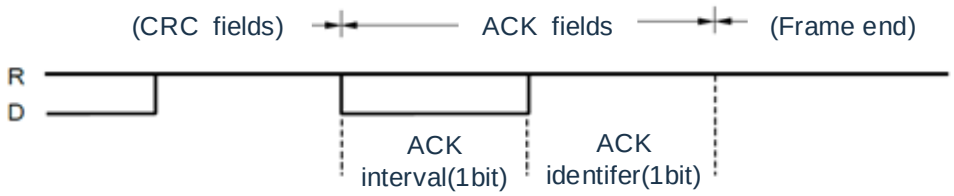
R: Implicit=1

- Polynomial P(X) for generating a 15-bit CRC sequence is shown as follows:
- . $P(X)=X15+X14+X10+X8+X7+X^4+X^3+1$
- Sending Node: CRC sequence calculated by sending data (before bit filling) of start frame, arbitration field, control field, and data field
- Receiving Node: A calculated CRC sequence is compared using a data bit that excludes a fill bit in the received data with a CRC sequence in the CRC field. If two CRC sequences do not match, the node issues an error frame.

<6> acknowledgement field

The response field is mainly to respond to normal reception

Figure 21-11. Response field



Note D: Explicit=0

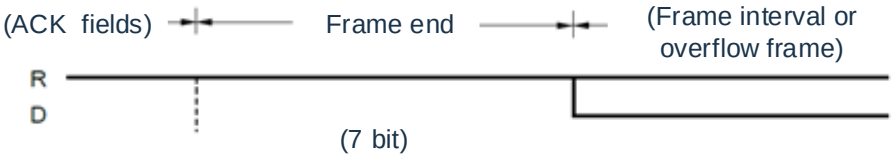
R: Implicit=1

- If no CRC error is detect, that receive node sets the response to an dominant bit
- The sending node outputs two hidden bits

<7> End of frame (EOF)

The frame end field indicates the end of a data frame or a remote frame.

Figure 21-12. End of Frame (EOF)



Note D: Explicit=0

R: Implicit=1

<8> interframe space

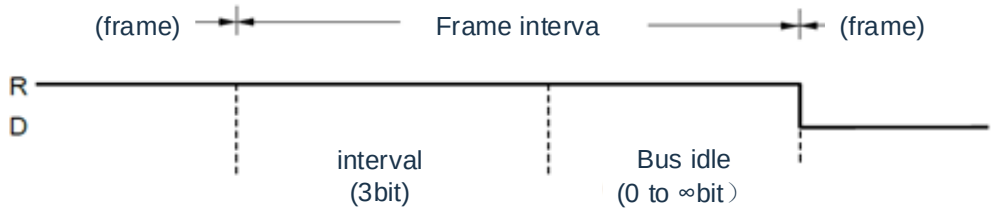
Interframe space is used to insert between data frames, remote frames, error frames, or overloaded frames to distinguish between two frames.

- Different bus states Different error-based states

(a)) Error Activity Node

The inter-frame space contains a 3-bit gap field and a bus idle field.

Figure 21-13. Interframe Space (Error Active Node)



Note 1. Bus Idle: Node status where the bus is not in use

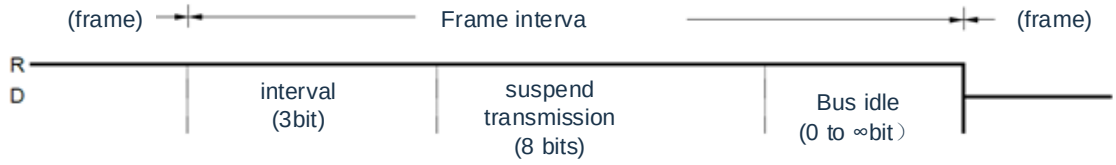
2. D: Explicit=0

R: Implicit=1

(b) error active node

The interframe space contains a gap field, a suspend field, and a bus idle field.

Figure 21-14. Interframe Space (Error Active Node)



Note 1. Bus idle: The bus is not used by any node.

Pending transfer: In the case of error passivity, the 8 hidden bits are sent at the node.

2. D:

Explicit=0

R:

Stealth=1

In general, the gap field is 3 bits, if the sending node detects an dominant bit in the third bit of the gap field, the sending operation will be performed regardless.

- Error Status Action

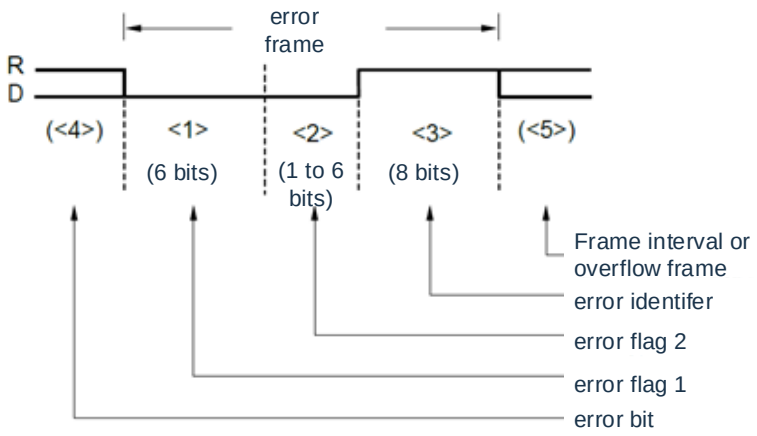
Table 21-6. Operation in error state

Error Status	Action
error initiative	Three bit gaps can be transmitted immediately.
error passive	You can send 8 bits after the gap.

21.2.4 error frame

If an error is detected, error frames are sent at the next node.

Figure 21-15. Error Frame



Note D: Explicit=0

R: Stealth=1

Table 21-7. Definition of Error Frames

No.	First Name	number of digits	Definitions
<1>	Error Flag1	6	Error active node: Six dominant bits are output continuously. Error passive node: Six hidden bits are output continuously. When one node is outputting a passive error flag, another node outputs an explicit bit, and the passive error flag is not cleared until six consecutive bits of the same polarity are detected.
<2>	Error Flag2	0~6	The node accepts the error flag 1 to detect a bit fill error and issues this error flag.
<3>	error delimiter	8	The output is 8 hidden bits. If an explicit bit is detected at the eighth bit, the next bit sends out an overload frame.
<4>	error bit	-	Error detected. An error flag is issued immediately after the error bit. . In that case of a CRC error, the error bit is then issue after the ACK delimiter.
<5>	Interframe space/overload frame	-	An inter-frame space or an overload frame is sent out at this time

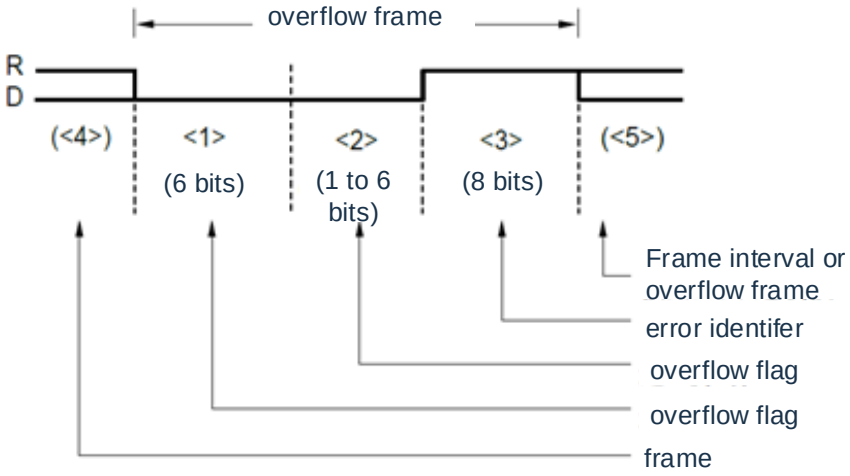
21.2.5 overload frame

Overload frames are issued in the following cases.

- When the receiving node does not complete the receiving operation [1]
- During a gap, if the first two bits are detected
- If the dominant bit is in the last bit of the end frame (seventh bit) or in the last bit of the error delimiter/overload delimiter (eighth bit)

The internal speed of the 1:CAN is fast enough to process all received frames without overloading them.

Figure 21-16. Overload frame



Note D: Explicit=0

R: Stealth=1

Table 21-8. Field definition for overloaded frames

No	First Name	Bit number	Definitions
<1>	overload flag	6	Continuous output of 6 dominant bits
<2>	Overload flag for other nodes	0 to 6	An overload flag is received in the inter-frame space and issued
<3>	overload delimiter	8	Continuous output of 8 stealth bits If an explicit bit is detected at the eighth bit, then the next bit issues an overload frame
<4>	frame	-	Output after the end frame, error delimiter, overload delimiter.
<5>	Interframe space/overload frame	-	The interframe space or overloaded frames are issued

21.3 Features

21.3.1 bus priority setting

(1) When a node starts transmitting:

- When the bus is idle, the node output data starts

(2) When more than one node starts transmission:

- The node continuously outputs the longest dominant bit from the first bit of the arbitration field to capture the priority of the bus (the dominant bit is the bus value if both the dominant bit and the invisible bit transmit simultaneously).
- The transmission node compares its output arbitration field with the data level on the bus.

Table 21-9. Bus Priority Settings

level matching	continuous transmission
level mismatch	When a mismatch is detected, the transmission is stopped and reception is started at the next bit

(3) Priority of data and remote frames

- When the data frame and the remote frame are on the bus at the same time, the data frame has a higher priority, because the RTR bit of the data frame, the last bit of the arbitration frame is the dominant bit

Note If the extended format data frame and the standard format remote frame conflict on the bus (if ID18 to ID28 are the same), the standard format remote frame takes precedence.

21.3.2 bit filling

Bit-filling is when five consecutive identical bits appear, establishing synchronization by inserting a bit-reversed number of complementary bits, thereby preventing explosive errors.

Table 21-10. Bit Population

Send	When a data frame or a remote frame is sent, if at the beginning of the frame and the ACK field, five identical bits appear continuously, and a data opposite to the data bit is inserted in the next bit.
Receive	When a data frame or a remote frame is received, when the same five consecutive bits appear between the start frame and the ACK field, the data is continuously received after the bit is deleted

21.3.3 multi-master

Identifier determines the priority of the bus (node capture to transmission function) Any node can be the master device of the bus

21.3.4 multicast

Even if there is only one transmission node, two or more nodes can receive the same data at the same time, because the same identifier can be set at two or more nodes

21.3.5 CAN Sleep Mode/CAN Stop Mode Features

The CAN sleep/CAN stop mode enables the CAN controller to enter a waiting mode, thereby obtaining lower power consumption. .

The CAN sleep mode may wake from bus operation, but that CAN stop mode may not be wake by bus operation (the CAN stop mode is controlled by the CPU).

21.3.6 error control function

(4) Error Type

Table 21-11. Error Types

Type	Error Description		Detect Status	
	detection method	detection condition	Send/Receive	Field/Frame
bit error	Comparing output level and bus level	mismatched levels	Send/Receive Node	bits of data between the start frame and the end frame on the bus, error frames and overload frames
Fill Error	Detecting received data at a fill bit	6 bits with the same output level continuously	receiving node	Start frame to CRC sequence
CRC error	Comparing CRC from received data and received CRC timing	CRC mismatch	receiving node	CRC field
Format Error	Detecting fields/frames in a mixed format	Detect mixed formatting errors	receiving node	CRC delimiter ACK field end frame error frame overload frame
ACK Error	Detecting an ACK gap in a sending node	Detecting stealthy bit in ACK gap	sending node	ACK gap

(5) Error frame output timing

Table 21-12. Error Frame Output Timing

Type	output timing
bit error, fill error, format error, ACK	Output error frames at the output timing after error detection
CRC error	The output timing after the ACK delimiter starts to output the error frame

(6) error handling

The sending node resends the data frame or the remote frame after the error frame (frames are not resold in single-shot mode).

(7) Error Status

(a) Type of error status

The next three error states are defined by the CAN specification

- error initiative
- error passive
- bus shutdown

The error type is determined by the CAN error count register (C0ERC) TEC0 to TEC7 and REC0 to REC6 values, shown in Table 21-13.
The current error is determined by the information register (C0INFO) of the CAN module.
When each counter becomes equal or greater than the error warning level (96), the TECS0 or RECS0 of the C0INFO register is set to 1. In this case, the status of the bus must be tested because there may be a serious failure. A value of 128 or greater for the error counter indicates an error passive state with the TECS1 or RECS1 of register C0INFO placed 1.

- If the transmission error counter has a value greater than or equal to 256 (in fact, the transmission error counter does not indicate a value greater than or equal to 256), the bus shutdown state arrives and the BOFF bit of the C0INFO register is set to 1.
- At start-up, if only one node on the bus is active (for example: In a particular case, such as when the bus is only connected to a local site), ACK is not returned, even if data is transmitted. Therefore, repeated transmission error frames and data are transmitted. In an error passive state, the transmission error counter does not increase and the bus shutdown state does not occur.

Table 21-13. Types of Error States

Type	Action	Value of error counter	C0 INFO register	Get operational details of error status
error initiative	Send	0-95	TECS1, TECS0 =00	- Output one active error flag (6 persistent explicit bits)
	Receive	0-95	RECS1, RECS0 =00	
	Send	96-127	TECS1, TECS0 =01	
	Receive	96-127	RECS1, RECS0 =01	
error passive	Send	128-255	TECS1, TECS0 =11	- Outputs a passive error flag (6 persistent stealth bits) - In transit, 8 stealth bits are sent following the gap (pending transmission).
	Receive	128 or more	RECS1, RECS0 =11	
bus shutdown	Send	256 or more (not specified) [1]	BOFF =1, TECS1, TECS0 =11	- Communication is impossible. Messages are not stored when frames are received and the next <1>, <2>, and <3> are completed. <1> TSOUT binding. <2> REC increase/decrease. <3> VALID bit is set. -If the CAN module enters the initialization mode, the conversion request to the operation mode is made, when 11 persistent invisible bits are detected 128 times, the error counter is reset to 0, the error active state is recorded

1: When the BOFF bit is 1, the transmission error counter (TEC) is invalid, if an error increases the error counter by 8, the counter no longer increases.

(b) error counter

When the error occurs, the error counter counts upward; When successful acceptance and sending, the error counter counts down. When an error is detected, the value of the error counter is updated immediately.

Table 21-14. Error counters

Status	Transmission error counter (TEC0~TEC7)	Receive error counter (REC0~REC6)
The receiving node detected an error (except the active error flag or	No change.	+1 (when REPS bit=0)
Receive node detects dominant bit after error flag of error frame	No change.	+8 (when REPS bit=0)
Sending node issues error flag [As expected, the error counter will not change in the following cases] <1> The ACK error is detected in the passive state and the dominant bit is not detected when the passive error flag is issued <2> An arbitration field fill error is detected, sending an invisible bit as a fill, but an explicit bit is detected	+8	No change.
Bit error detected on active error flag or overload flag output (error - active sending node)	+8	No change.
Bit error detected on active error flag or overload flag output (error - active receive node)	No change.	+8 (when REPS bit =0)
When the node starts to detect 14 successive dominant bits from the active error flag or the overload flag, then 8 successive dominant bits are detected. When a node detects 8 consecutive dominant bits after a	+8 at transfer)	+8 (When received, when REPS =0)
Transmission node completes the transfer with no errors (±0 if the error counter =0)	-1	No change.
Receive node completes receive without error	No change.	- -1 (1 REC6 to REC0 127, when REPS bit=0) - ±0 (REC6 to REC0 = 0, When REPS bit=0) Values 119 to 127

(c) Bit errors occur in intermittency

overload frame generation

Warning If an error occurs, the error flag output (active or passive) is controlled based on the content of the transmission error counter and the receipt error pair prior to the error occurrence. After the error flag is output, the value of the error counter is incremented.

(8) bus shutdown recovery

When the CAN module is in the bus closed state, the CAN module permanently sets its output signal (CTxD) as the hidden bit.

The CAN module recovers from the bus shutdown state in the following bus shutdown recovery sequence.

<1> Request to enter CAN initialization mode

<2> Apply to enter CAN mode

(a) Recover from normal recovery sequence

(b) Skip recovery sequence force recovery operation

(a) Recover from bus-off state following normal recovery sequence

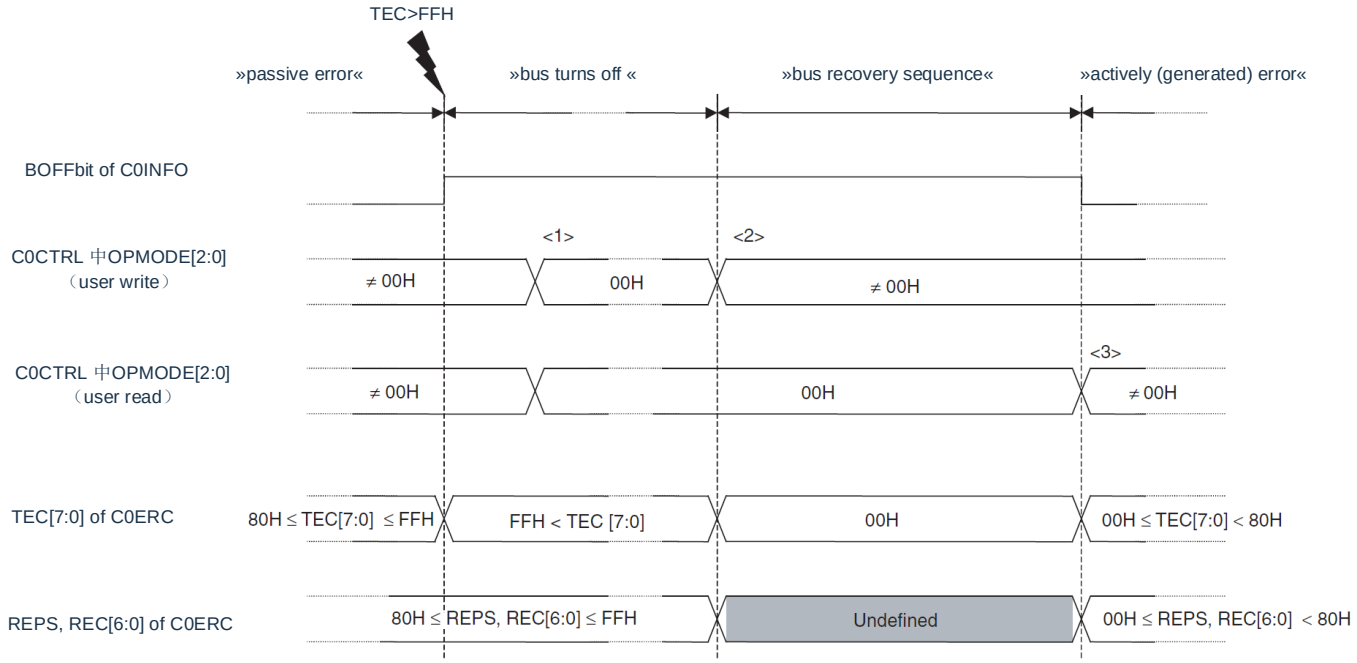
The CAN module is released for the first time into the initialization request (Reference Timing <1> in Figure 21-17).

Next, the user requests from the initial state to the mode of operation (reference timing <2> in Figure 21-17). This initiates an operation to allow the CAN module to recover from the bus shutdown state. The CAN protocol ISO 11898 defines the conditions under which a module can recover from a bus-off state, requiring 11 consecutive hidden bits to be detected 128 times. At this point, the request to change the mode to the operational mode remains pending until the recovery condition is met. When recovery conditions are met. (With reference to timing <3> in Figure 21-17), the CAN module can enter the requested mode of operation. The CAN module will remain in initialization mode until it enters this mode. By reading the OPMODE bit of the C0CTRL register, you can confirm the completion of the requested mode of operation..
The BOFF bit hold setting (set to 1) for the C0INFO register during bus shutdown and in the bus shutdown recovery sequence. In the bus-off recovery sequence, the receive error counter (REC[6:0]) calculates the number of consecutive hidden bits detected on the bus. Therefore, you can check the recovery status by reading the REC_6:0.

Warning

1. If that bus-off recovery sequence is interrupt by entering the initialization mode and re-entering any mode of operation, the bus-off recovery sequence will restart from scratch from this point, and the wait phase will again be 128 times 11 hidden bits.
   a) In a bus-off recovery sequence, REC [6:0] counts (+1) each time 11 consecutive hidden bits are detected. The CAN module may enter a CAN sleep mode or a CAN stop mode even during bus shutdown. To start the bus shutdown recovery sequence, you must transfer to initialization mode once.
2. However, when the CAN module is in the CAN sleep mode or the CAN stop mode, the request to transition to the initialization mode is not accepted, so the CAN sleep mode must be released first. In this case, once the CAN sleep mode is released, the bus-off recovery sequence is initiated and there is no need to transition to the initialization mode. If the CAN module detects an explicit edge on the CAN bus in sleep mode during bus shutdown, the sleep mode will leave and the bus shutdown recovery sequence will start (in the state of providing the CAN clock, the PSMODE must be cleared by software after the explicit edge).

Figure 21-17. Recovery operation from bus-off state by normal recovery sequence



(b) Skip mandatory recovery of bus-off recovery sequence

Regardless of the bus state, the CAN module can be forced to release from the bus shutdown state by skipping the bus shutdown recovery sequence. It's a process.

First, the CAN module requests to enter the initialization mode. For actions and points to note at this time, see (a) Recovering Operations from a Bus Off State with a Normal Recovery Sequence.

Next, the module requests to enter operation mode. Also, the CCERC bit of the C0CTRL register must be set to 1.

Therefore, the bus shutdown recovery sequence defined by CAN protocol ISO 11898 is skipped, and the module immediately enters operation mode. In this case, the module is connected to the CAN bus after monitoring 11 consecutive hidden bits. For more information, see Figure 21-82.

Warning This feature is not defined by the CAN protocol ISO 11898. When using this feature, thoroughly evaluate its impact on the network system.

(9) Initialize CAN Module Error Counter Register (C0ERC) in Initialization Mode

If it is necessary to initialize that CAN module error count register (C0ERC) and the CAN module information register (C0INFO) to debug or evaluate the program, then it may be necessary to perform a test of the program by performing a test on the When initialization is complete, the CCERC bit is automatically cleared to 0.

Warning 1. This feature is enabled only in initialization mode. Registers C0ERC and C0INFO are not initialized even if the CCERC bit is set to 1 in CAN mode of operation.

2. The CCERC bit can be set to be set at the same time as the request enters CAN mode of operation.

21.3.7 baud rate control

(10) pre-scaling

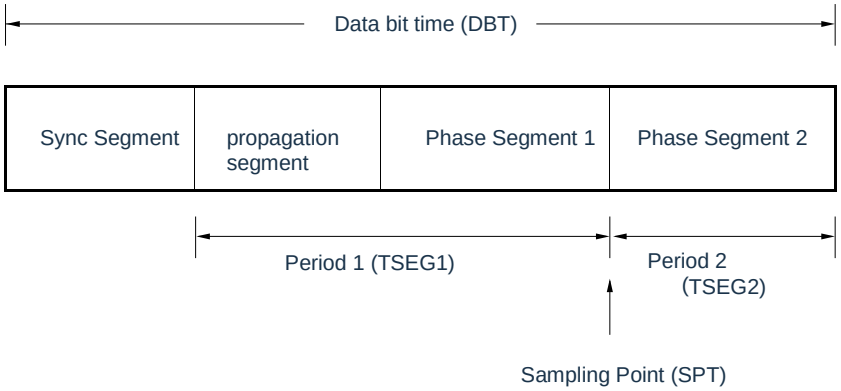
The CAN controller has a pre-scaling for dividing a clock (fCAN) provided to the CAN. This pre-scaling generates a CAN protocol layer basic clock (fTQ) derived from the CAN module system clock (fCANMOD), divided between 1 and 256 (see 21.6 (12) CAN bit rate pre-scaler register (C0BRP)

(11) Data bit time (8-25 time quantization)

A data bit time is defined as shown in FIGS. 21-18.

The CAN controller takes time period 1, time period 2, and resynchronization jump width (SJW) as parameters of data bit time, as shown in Figure 21-18. The time period 1 is equivalent to the sum of the propagation (prop) and phase period 1 defined by the CAN protocol specification. Period 2 is equivalent to phase 2..

Figure 21-18. Time Bucket Settings



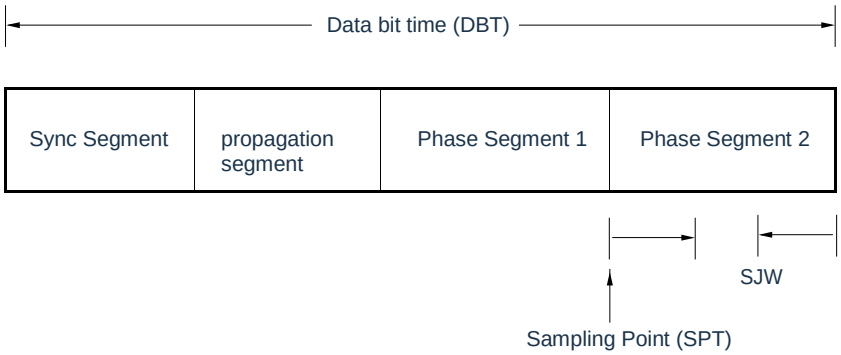
Segment name	Settable range	Instructions for setting to CAN specification
Time Period 1 (TSEG1)	2TQ-16TQ	—
Period 2 (TSEG2)	1TQ-8TQ	The IPT of the CAN controller is 0TQ. Therefore, in order to comply with the CAN protocol specification, the length must be set here to be less than or equal to phase 1.
Resynchronization Jump Width (SJW)	1TQ-4TQ	The length of time period 1 minus 1TQ or 4TQ, whichever is smaller

Note IPT: Information processing time

TQ: time share

Reference: The CAN standard, ISO 11898, define that segment that make up the data bit time as shown in Figure 21-19.

Figure 21-19. Reference: Data bit time configuration defined by the CAN specification



Segment name	segment length	Description
Sync Segment	1	This segment starts at the edge where the level changes from implicit to explicit when hard synchronization is established.
propagation segment	Programmable from 1 to 8 or more	This segment absorbs the delay of the output buffer, CAN bus, and input buffer. Set the length of this segment to return ACK before phase 1 begins. Propagation Period Time (Output Buffer Delay) + 2x (CAN Bus Delay) + (Input Buffer Delay)
Phase Segment 1	Programmable 1 to 8	This segment compensates for errors in data bit time. The longer this segment, the wider the allowable range, but the slower the communication.
Phase Segment 2	Phase segment 1 or IPT, take maximum	
SJW	Programmable from 1TQ to the length of segment 1 or then 4TQ, minimum	This width sets the upper limit of the extended or collapsed phase segment during resynchronization

Note IPT: information processing time

TQ: time share

(12) synchronization data bit

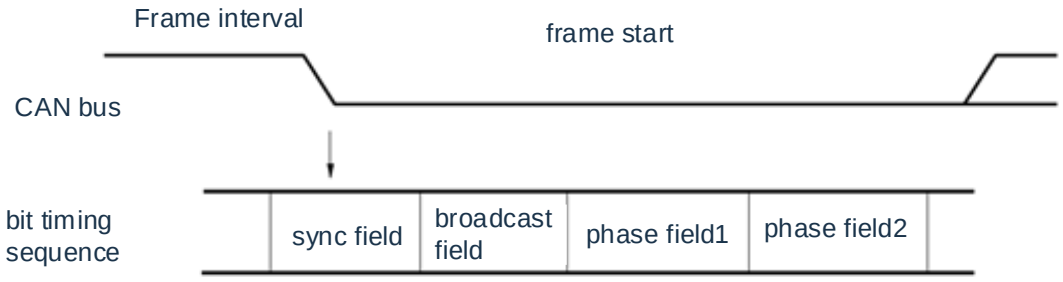
- The receiving node establishes synchronization by level changes on the bus because it has no asynchronous signal.
- The transmission node transmits data synchronously at the bit time of the transmission node

(a) hardware synchronization

This synchronization is established when the receiving node detects the start of a frame in the inter-frame space.

- When a falling edge is detected on the bus, TQ represents the synchronization segment and the next segment is the propagation segment. In this case, synchronization is established regardless of SJW.

Figure 21-20. Hard synchronization in identifying dominant bits during bus idleness



(b) resynchronization

If a level change is detect on that bus during reception, a synchronization is established again (only if the covert level has previously been sampled).

- The phase error of the edge is given by the relative position of the detected edge and the synchronization segment.

<phase error signal>

0: If the edge is within the synchronization segment

Positive: if that edge is before the sample point (phase error)

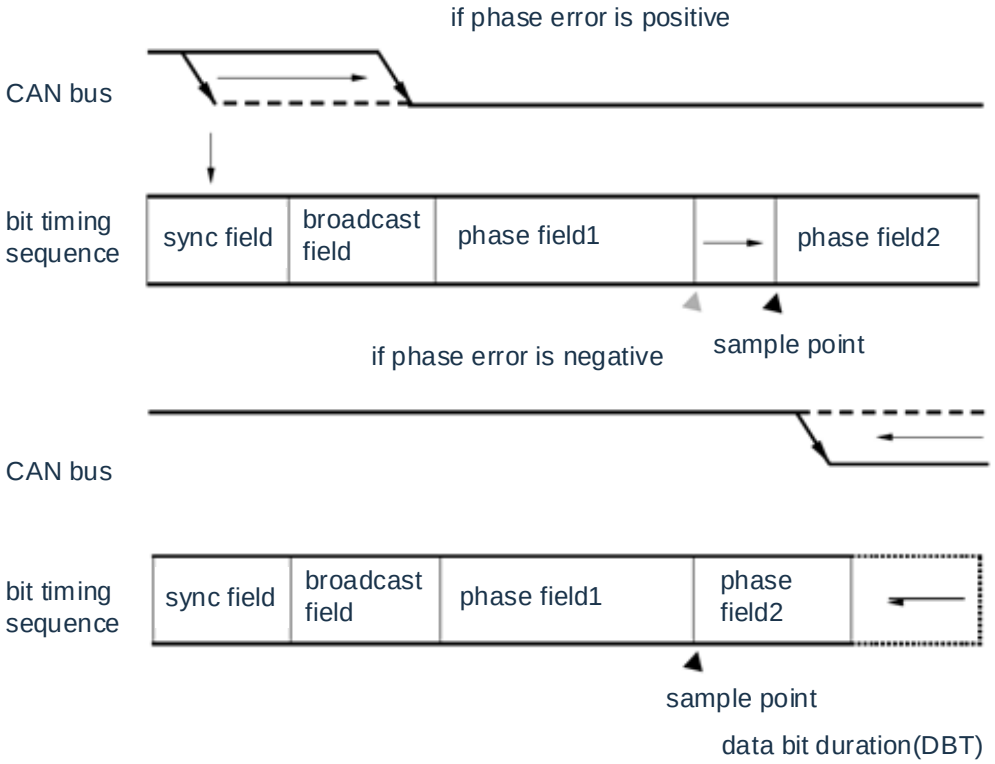
Negative: if that edge is aft the sampling point (phase error)

If the phase error is positive: Phase segment 1 is longer with the specified SJW.

If the phase error is negative: Phase segment 2 is shortened by the specified SJW.

- The sampling points of the data of the receiving node move relatively due to the "difference" of baud rate between the transmitting node and receiving node.

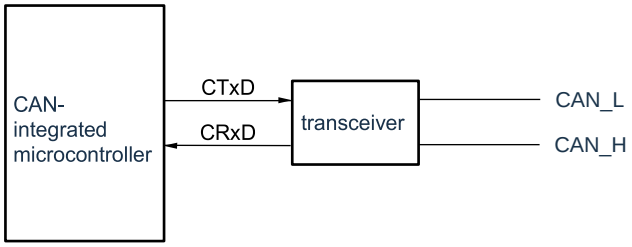
Figure 21-21. Resynchronization



21.4 Connection to target system

The CAN-integrated microcontroller must use an external transceiver to connect to the CAN bus.

Figure 21-22. Connecting to the CAN Bus



21.5 Internal register for CAN controller

21.5.1 CAN controller configuration

Table 21-15. List of CAN Controller Registers (1/2)

Items	register name
control register	Peripheral Enable Register 0 (PER0)
	Serial Communication Pin Selection Register (PIOR3)
	Port registers 0, 5 (P0, P5)
	Port Mode Registers 0, 5 (PM0, PM5)
CAN global register	CAN Global Module Control Register (C0GMCTRL)
	CAN Global Module Clock Select Register (C0GMCS)
	CAN Global Automatic Block Transfer Control Register (C0GMABT)
	CAN Global Automatic Block Transfer Delay Set-up Register (C0GMABTD)
CAN module register	CAN Module Mask 1 Register L (C0MASK1L)
	CAN Module Mask 1 Register H (C0MASK1H)
	CAN Module Mask 2 Register L (C0MASK2L)
	CAN Module Mask 2 Register H (C0MASK2H)
	CAN Module Masking 3 Register L (C0MASK3L)
	CAN Module Masking 3 Register H (C0MASK3H)
	CAN Module Masking 4 Register L (C0MASK4L)
	CAN Module Mask 4 Register H (C0MASK4H)
	CAN Module Control Register (C0CTRL)
	CAN module last error code register (C0LEC)
	CAN Module Information Register (C0INFO)
	CAN Module Error Count Register (C0ERC)
	CAN Module Interrupt Enable Register (C0IE)
	CAN Module Interrupt State Register (C0INS)
	CAN Module Bit Rate Scaling Register (C0BRP)
	CAN Module Bit Rate Register (C0BTR)
	CAN module last input pointer register (C0LIPT)
	CAN Module Receive History List Register (C0RGPT)
	CAN Module Last Output Pointer Register (C0LOPT)
	CAN Module Send History List Register (C0TGPT)
	CAN Module Timestamp Register (C0TS)

Note that the CAN global register is identified by the CGM<register function>.

CAN module registers are identified by C<register function>.

The message buffer registers are identified by the CM<register function>.

Table 21-15. List of CAN Controller Registers (2/2)

Items	register name
message cache register	CAN message data byte 01 register m (C0MDB01m)
	CAN message data byte 0 register m (C0MDB0m)
	CAN message data byte 1 register m (C0MDB1m)
	CAN message data byte 23 register m (C0MDB23m)
	CAN message data byte 2 register m (C0MDB2m)
	CAN Message Data Byte 3 Register m (C0MDB3m)
	CAN message data byte 45 register m (C0MDB45m)
	CAN message data byte 4 register m (C0MDB4m)
	CAN Message Data Byte 5 Register m (C0MDB5m)
	CAN message data byte 67 register m (C0MDB67m)
	CAN message data byte 6 register m (C0MDB6m)
	CAN message data byte 7 register m (C0MDB7m)
	CAN message data length register m (C0MDLCm)
	CAN message configuration register m (C0MCONFm)
	CAN message ID register L m (C0MIDLm)
	CAN message ID register H m (C0MIDHm)
	CAN message control register m (C0MCTRLm)

Note 1. CAN global registers are identified by CGM<register function>.

The module registers are identified by C<register function>.

The message buffer registers are identified by the CM<register function>.

2. m = 0 to 15

21.5.2 register access type

The peripheral I/O registers of the CAN controller are located in the range of 0x40045400 to 0x400455FF.

Table 21-16. Register Access Types (1/9)

Address	register name	Symbol	read/write	bit operation unit			Default
				1	8	16	
0x40045400H	CAN0 Global Module Control Register	C0GMCTRL	read/write				0000H
0x40045406H	CAN0 Global Automatic Block Transfer	C0GMBT					0000H
0x40045408H	CAN0 Global Automatic Block Transfer	C0GMABTD					00H
0x40045402H	CAN0 Global Module Clock Selection	C0GMCS					0FH
0x40045440H	CAN0 Module Mask 1 Register	C0MASK1L					No definition
0x40045442H		C0MASK1H					No definition
0x40045444H	CAN0 Module Mask 2 Register	C0MASK2L					No definition
0x40045446H		C0MASK2H					No definition
0x40045448H	CAN0 Module Mask 3 Register	C0MASK3L					No definition
0x4004544AH		C0MASK3H					No definition
0x4004544CH	CAN0 Module Mask 4 Register	C0MASK4L					No definition
0x4004544EH		C0MASK4H					No definition
0x40045450H	CAN0 Module Control Register	C0CTRL					0000H
0x40045452H	CAN0 Module Last Error Code Register	C0LEC					00H
0x40045453H	CAN0 module information register	C0INFO	read				00H
0x40045454H	CAN0 Module Error Count Register	C0ERC					0000H
0x40045456H	CAN0 module interrupt enable register	C0IE	read/write				0000H
0x40045458H	CAN0 module interrupt state register	C0INS					0000H
0x4004545AH	CAN0 module bit rate scaling register	C0BRP					FFH
0x4004545CH	CAN0 module bit rate register	C0BTR					370FH
0x4004545EH	CAN0 module last input pointer register	C0LIPT	read				No definition
0x40045460H	CAN0 module receive history list register	C0RGPT	read/write				xx02H
0x40045462H	CAN0 module last output pointer register	C0LOPT	read				No definition
0x40045464H	CAN0 module send history list register	C0TGPT	read/write				xx02H
0x40045466H	CAN0 module timestamp register	C0TS					0000H

Table 21-16. Register Access Types (2/9)

Address	register name	Symbol	read/ write	bit operation unit			Default
				1	8	16	
0x40045500H	CAN0 message data byte 01 register 00	C0MDB0100	read/ write	-	-		No definition
0x40045500H	CAN0 message data byte 0 register 00	C0MDB000		-		-	No definition
0x40045501H	CAN0 Message Data Byte 1 Register 00	C0MDB100		-		-	No definition
0x40045502H	CAN0 message data byte 23 register 00	C0MDB2300		-	-		No definition
0x40045502H	CAN0 Message Data Byte 2 Register 00	C0MDB200		-		-	No definition
0x40045503H	CAN0 Message Data Byte 3 Register 00	C0MDB300		-		-	No definition
0x40045504H	CAN0 message data byte 45 register 00	C0MDB4500		-	-		No definition
0x40045504H	CAN0 message data byte 4 register 00	C0MDB400		-		-	No definition
0x40045505H	CAN0 message data byte 5 register 00	C0MDB500		-		-	No definition
0x40045506H	CAN0 message data byte 67 register 00	C0MDB6700		-	-		No definition
0x40045506H	CAN0 message data byte 6 register 00	C0MDB600		-		-	No definition
0x40045507H	CAN0 message data byte 7 register 00	C0MDB700		-		-	No definition
0x40045508H	CAN0 Message Data Length Register 00	C0MDLC00		-		-	0000xxxxB
0x40045509H	CAN0 Message Configuration Register 00	C0MCONF00		-		-	No definition
0x4004550AH	CAN0 Message ID Register 00	C0MIDL00		-	-		No definition
0x4004550CH		C0MIDH00		-	-		No definition
0x4004550EH	CAN0 Message Control Register 00	C0MCTRL00		-	-		00x0000 000xx00B
0x40045510H	CAN0 message data byte 01 register 01	C0MDB0101		-	-		No definition
0x40045510H	CAN0 message data byte 0 register 01	C0MDB001		-		-	No definition
0x40045511H	CAN0 message data byte 1 register 01	C0MDB101		-		-	No definition
0x40045512H	CAN0 message data byte 23 register 01	C0MDB2301		-	-		No definition
0x40045512H	CAN0 Message Data Byte 2 Register 01	C0MDB201		-		-	No definition
0x40045513H	CAN0 message data byte 3 register 01	C0MDB301		-		-	No definition
0x40045514H	CAN0 message data byte 45 register 01	C0MDB4501		-	-		No definition
0x40045514H	CAN0 message data byte 4 register 01	C0MDB401		-		-	No definition
0x40045515H	CAN0 message data byte 5 register 01	C0MDB501		-		-	No definition
0x40045516H	CAN0 message data byte 67 register 01	C0MDB6701		-	-		No definition
0x40045516H	CAN0 message data byte 6 register 01	C0MDB601		-		-	No definition
0x40045517H	CAN0 message data byte 7 register 01	C0MDB701		-		-	No definition
0x40045518H	CAN0 message data length register 01	C0MDLC01		-		-	0000xxxxB
0x40045519H	CAN0 Message Configuration Register 01	C0MCONF01		-		-	No definition
0x4004551AH	CAN0 message ID register 01	C0MIDL01		-	-		No definition
0x4004551CH		C0MIDH01		-	-		No definition
0x4004551EH	CAN0 Message Control Register 01	C0MCTRL01		-	-		00x0000 000xx00B

Table 21-16. Register Access Types (3/9)

Address	register name	Symbol	read/ write	bit operation unit			Default
				1	8	16	
0x40045520H	CAN0 message data byte 01 register 02	C0MDB0102	read/ write	-	-		No definition
0x40045520H	CAN0 message data byte 0 register 02	C0MDB002		-		-	No definition
0x40045521H	CAN0 Message Data Byte 1 Register 02	C0MDB102		-		-	No definition
0x40045522H	CAN0 message data byte 23 register 02	C0MDB2302		-	-		No definition
0x40045522H	CAN0 Message Data Byte 2 Register 02	C0MDB202		-		-	No definition
0x40045523H	CAN0 message data byte 3 register 02	C0MDB302		-		-	No definition
0x40045524H	CAN0 message data byte 45 register 02	C0MDB4502		-	-		No definition
0x40045524H	CAN0 message data byte 4 register 02	C0MDB402		-		-	No definition
000F625H	CAN0 message data byte 5 register 02	C0MDB502		-		-	No definition
0x40045526H	CAN0 message data byte 67 register 02	C0MDB6702		-	-		No definition
0x40045526H	CAN0 message data byte 6 register 02	C0MDB602		-		-	No definition
0x40045527H	CAN0 message data byte 7 register 02	C0MDB702		-		-	No definition
0x40045528H	CAN0 Message Data Length Register 02	C0MDLC02		-		-	0000xxxxB
0x40045529H	CAN0 Message Configuration Register 02	C0MCONF02		-		-	No definition
0x4004552AH	CAN0 message ID register 02	C0MIDL02		-	-		No definition
0x4004552CH		C0MIDH02		-	-		No definition
0x4004552EH	CAN0 Message Control Register 02	C0MCTRL02		-	-		00x0000 000xx00B
0x40045530H	CAN0 message data byte 01 register 03	C0MDB0103		-	-		No definition
0x40045530H	CAN0 message data byte 0 register 03	C0MDB003		-		-	No definition
0x40045531H	CAN0 message data byte 1 register 03	C0MDB103		-		-	No definition
0x40045532H	CAN0 message data byte 23 register 03	C0MDB2303		-	-		No definition
0x40045532H	CAN0 Message Data Byte 2 Register 03	C0MDB203		-		-	No definition
0x40045533H	CAN0 message data byte 3 register 03	C0MDB303		-		-	No definition
0x40045534H	CAN0 message data byte 45 register 03	C0MDB4503		-	-		No definition
0x40045534H	CAN0 message data byte 4 register 03	C0MDB403		-		-	No definition
0x40045535H	CAN0 message data byte 5 register 03	C0MDB503		-		-	No definition
0x40045536H	CAN0 message data byte 67 register 03	C0MDB6703		-	-		No definition
0x40045536H	CAN0 message data byte 6 register 03	C0MDB603		-		-	No definition
0x40045537H	CAN0 message data byte 7 register 03	C0MDB703		-		-	No definition
0x40045538H	CAN0 Message Data Length Register 03	C0MDLC03		-		-	0000xxxxB
0x40045539H	CAN0 Message Configuration Register 03	C0MCONF03		-		-	No definition
0x4004553AH	CAN0 message ID register 03	C0MIDL03		-	-		No definition
0x4004553CH		C0MIDH03		-	-		No definition
0x4004553EH	CAN0 Message Control Register 03	C0MCTRL03		-	-		00x0000 000xx00B

Table 21-16. Register Access Types (4/9)

Address	register name	Symbol	read/ write	bit operation unit			Default
				1	8	16	
0x40045540H	CAN0 message data byte 01 register 04	C0MDB0104	read/ write	-	-		No definition
0x40045540H	CAN0 message data byte 0 register 04	C0MDB004		-		-	No definition
0x40045541H	CAN0 Message Data Byte 1 Register 04	C0MDB104		-		-	No definition
0x40045542H	CAN0 message data byte 23 register 04	C0MDB2304		-	-		No definition
0x40045542H	CAN0 Message Data Byte 2 Register 04	C0MDB204		-		-	No definition
0x40045543H	CAN0 message data byte 3 register 04	C0MDB304		-		-	No definition
0x40045544H	CAN0 message data byte 45 register 04	C0MDB4504		-	-		No definition
0x40045544H	CAN0 message data byte 4 register 04	C0MDB404		-		-	No definition
0x40045545H	CAN0 message data byte 5 register 04	C0MDB504		-		-	No definition
0x40045546H	CAN0 message data byte 67 register 04	C0MDB6704		-	-		No definition
0x40045546H	CAN0 message data byte 6 register 04	C0MDB604		-		-	No definition
0x40045547H	CAN0 message data byte 7 register 04	C0MDB704		-		-	No definition
0x40045548H	CAN0 Message Data Length Register 04	C0MDLC04		-		-	0000xxxxB
0x40045549H	CAN0 Message Configuration Register 04	C0MCONF04		-		-	No definition
0x4004554AH	CAN0 message ID register 04	C0MIDL04		-	-		No definition
0x4004554CH		C0MIDH04		-	-		No definition
0x4004554EH	CAN0 Message Control Register 04	C0MCTRL04		-	-		00x0000 000xx00B
0x40045550H	CAN0 message data byte 01 register 05	C0MDB0105		-	-		No definition
0x40045550H	CAN0 message data byte 0 register 05	C0MDB005		-		-	No definition
0x40045551H	CAN0 message data byte 1 register 05	C0MDB105		-		-	No definition
0x40045552H	CAN0 message data byte 23 register 05	C0MDB2305		-	-		No definition
0x40045552H	CAN0 Message Data Byte 2 Register 05	C0MDB205		-		-	No definition
0x40045553H	CAN0 message data byte 3 register 05	C0MDB305		-		-	No definition
0x40045554H	CAN0 message data byte 45 register 05	C0MDB4505		-	-		No definition
0x40045554H	CAN0 message data byte 4 register 05	C0MDB405		-		-	No definition
0x40045555H	CAN0 message data byte 5 register 05	C0MDB505		-		-	No definition
0x40045556H	CAN0 message data byte 67 register 05	C0MDB6705		-	-		No definition
0x40045556H	CAN0 message data byte 6 register 05	C0MDB605		-		-	No definition
0x40045557H	CAN0 message data byte 7 register 05	C0MDB705		-		-	No definition
0x40045558H	CAN0 Message Data Length Register 05	C0MDLC05		-		-	0000xxxxB
0x40045559H	CAN0 Message Configuration Register 05	C0MCONF05		-		-	No definition
0x4004555AH	CAN0 message ID register 05	C0MIDL05		-	-		No definition
0x4004555CH		C0MIDH05		-	-		No definition
0x4004555EH	CAN0 Message Control Register 05	C0MCTRL05		-	-		00x0000 000xx00B

Table 21-16. Register Access Types (5/9)

Address	register name	Symbol	read/ write	bit operation unit			Default
				1	8	16	
0x40045560H	CAN0 message data byte 01 register 06	C0MDB0106	read/ write	-	-		No definition
0x40045560H	CAN0 message data byte 0 register 06	C0MDB006		-		-	No definition
0x40045561H	CAN0 Message Data Byte 1 Register 06	C0MDB106		-		-	No definition
0x40045562H	CAN0 message data byte 23 register 06	C0MDB2306		-	-		No definition
0x40045562H	CAN0 Message Data Byte 2 Register 06	C0MDB206		-		-	No definition
0x40045563H	CAN0 message data byte 3 register 06	C0MDB306		-		-	No definition
0x40045564H	CAN0 message data byte 45 register 06	C0MDB4506		-	-		No definition
0x40045564H	CAN0 message data byte 4 register 06	C0MDB406		-		-	No definition
0x40045565H	CAN0 message data byte 5 register 06	C0MDB506		-		-	No definition
0x40045566H	CAN0 message data byte 67 register 06	C0MDB6706		-	-		No definition
0x40045566H	CAN0 message data byte 6 register 06	C0MDB606		-		-	No definition
0x40045567H	CAN0 message data byte 7 register 06	C0MDB706		-		-	No definition
0x40045568H	CAN0 Message Data Length Register 06	C0MDLC06		-		-	0000xxxxB
0x40045569H	CAN0 Message Configuration Register 06	C0MCONF06		-		-	No definition
0x4004556AH	CAN0 Message ID register 06	C0MIDL06		-	-		No definition
0x4004556CH		C0MIDH06		-	-		No definition
0x4004556EH	CAN0 Message Control Register 06	C0MCTRL06		-	-		00x0000 000xx00B
0x40045570H	CAN0 message data byte 01 register 07	C0MDB0107		-	-		No definition
0x40045570H	CAN0 message data byte 0 register 07	C0MDB007		-		-	No definition
0x40045571H	CAN0 Message Data Byte 1 Register 07	C0MDB107		-		-	No definition
0x40045572H	CAN0 message data byte 23 register 07	C0MDB2307		-	-		No definition
0x40045572H	CAN0 Message Data Byte 2 Register 07	C0MDB207		-		-	No definition
0x40045573H	CAN0 Message Data Byte 3 Register 07	C0MDB307		-		-	No definition
0x40045574H	CAN0 message data byte 45 register 07	C0MDB4507		-	-		No definition
0x40045574H	CAN0 message data byte 4 register 07	C0MDB407		-		-	No definition
0x40045575H	CAN0 message data byte 5 register 07	C0MDB507		-		-	No definition
0x40045576H	CAN0 message data byte 67 register 07	C0MDB6707		-	-		No definition
0x40045576H	CAN0 message data byte 6 register 07	C0MDB607		-		-	No definition
0x40045577H	CAN0 message data byte 7 register 07	C0MDB707		-		-	No definition
0x40045578H	CAN0 Message Data Length Register 07	C0MDLC07		-		-	0000xxxxB
0x40045579H	CAN0 Message Configuration Register 07	C0MCONF07		-		-	No definition
0x4004557AH	CAN0 Message ID register 07	C0MIDL07		-	-		No definition
0x4004557CH		C0MIDH07		-	-		No definition
0x4004557EH	CAN0 Message Control Register 07	C0MCTRL07		-	-		00x0000 000xx00B

Table 21-16. Register Access Types (6/9)

Address	register name	Symbol	read/ write	bit operation unit			Default
				1	8	16	
0x40045580H	CAN0 message data byte 01 register 08	C0MDB0108	read/ write	-	-		No definition
0x40045580H	CAN0 message data byte 0 register 08	C0MDB008		-		-	No definition
0x40045581H	CAN0 message data byte 1 register 08	C0MDB108		-		-	No definition
0x40045582H	CAN0 message data byte 23 register 08	C0MDB2308		-	-		No definition
0x40045582H	CAN0 Message Data Byte 2 Register 08	C0MDB208		-		-	No definition
0x40045583H	CAN0 message data byte 3 register 08	C0MDB308		-		-	No definition
0x40045584H	CAN0 message data byte 45 register 08	C0MDB4508		-	-		No definition
0x40045584H	CAN0 message data byte 4 register 08	C0MDB408		-		-	No definition
0x40045585H	CAN0 message data byte 5 register 08	C0MDB508		-		-	No definition
0x40045586H	CAN0 message data byte 67 register 08	C0MDB6708		-	-		No definition
0x40045586H	CAN0 message data byte 6 register 08	C0MDB608		-		-	No definition
0x40045587H	CAN0 message data byte 7 register 08	C0MDB708		-		-	No definition
0x40045588H	CAN0 Message Data Length Register 08	C0MDLC08		-		-	0000xxxxB
0x40045589H	CAN0 Message Configuration Register 08	C0MCONF08		-		-	No definition
0x4004558AH	CAN0 message ID register 08	C0MIDL08		-	-		No definition
0x4004558CH		C0MIDH08		-	-		No definition
0x4004558EH	CAN0 Message Control Register 08	C0MCTRL08		-	-		00x0000 000xx00B
0x40045590H	CAN0 message data byte 01 register 09	C0MDB0109		-	-		No definition
0x40045590H	CAN0 message data byte 0 register 09	C0MDB009		-		-	No definition
0x40045591H	CAN0 message data byte 1 register 09	C0MDB109		-		-	No definition
0x40045592H	CAN0 message data byte 23 register 09	C0MDB2309		-	-		No definition
0x40045592H	CAN0 message data byte 2 register 09	C0MDB209		-		-	No definition
0x40045593H	CAN0 message data byte 3 register 09	C0MDB309		-		-	No definition
0x40045594H	CAN0 message data byte 45 register 09	C0MDB4509		-	-		No definition
0x40045594H	CAN0 message data byte 4 register 09	C0MDB409		-		-	No definition
0x40045595H	CAN0 message data byte 5 register 09	C0MDB509		-		-	No definition
0x40045596H	CAN0 message data byte 67 register 09	C0MDB6709		-	-		No definition
0x40045596H	CAN0 message data byte 6 register 09	C0MDB609		-		-	No definition
0x40045597H	CAN0 message data byte 7 register 09	C0MDB709		-		-	No definition
0x40045598H	CAN0 Message Data Length Register 09	C0MDLC09		-		-	0000xxxxB
0x40045599H	CAN0 Message Configuration Register 09	C0MCONF09		-		-	No definition
0x4004559AH	CAN0 message ID register 09	C0MIDL09		-	-		No definition
0x4004559CH		C0MIDH09		-	-		No definition
0x4004559EH	CAN0 Message Control Register 09	C0MCTRL09		-	-		00x0000 000xx00B

Table 21-16. Register Access Types (7/9)

Address	register name	Symbol	read/ write	bit operation unit			Default
				1	8	16	
0x400455A0H	CAN0 message data byte 01 register 10	C0MDB0110	read/ write	-	-		No definition
0x400455A0	CAN0 message data byte 0 register 10	C0MDB010		-		-	No definition
0x400455A1	CAN0 Message Data Byte 1 Register 10	C0MDB110		-		-	No definition
0x400455A2H	CAN0 message data byte 23 register 10	C0MDB2310		-	-		No definition
0x400455A2	CAN0 Message Data Byte 2 Register 10	C0MDB210		-		-	No definition
0x400455A3	CAN0 message data byte 3 register 10	C0MDB310		-		-	No definition
0x400455A4H	CAN0 message data byte 45 register 10	C0MDB4510		-	-		No definition
0x400455A4	CAN0 message data byte 4 register 10	C0MDB410		-		-	No definition
0x400455A5	CAN0 message data byte 5 register 10	C0MDB510		-		-	No definition
0x400455A6H	CAN0 message data byte 67 register 10	C0MDB6710		-	-		No definition
0x400455A6	CAN0 message data byte 6 register 10	C0MDB610		-		-	No definition
0x400455A7	CAN0 message data byte 7 register 10	C0MDB710		-		-	No definition
0x400455A8H	CAN0 message data length register 10	C0MDLC10		-		-	0000xxxxB
0x400455A9H	CAN0 Message Configuration Register 10	C0MCONF10		-		-	No definition
0x400455AAH	CAN0 message ID register 10	C0MIDL10		-	-		No definition
0x400455ACH		C0MIDH10		-	-		No definition
0x400455AEH	CAN0 message control register 10	C0MCTRL10		-	-		00x0000 000xx00B
0x400455B0H	CAN0 message data byte 01 register 11	C0MDB0111		-	-		No definition
0x400455B0	CAN0 message data byte 0 register 11	C0MDB011		-		-	No definition
0x400455B1	CAN0 message data byte 1 register 11	C0MDB111		-		-	No definition
0x400455B2H	CAN0 message data byte 23 register 11	C0MDB2311		-	-		No definition
0x400455B2	CAN0 message data byte 2 register 11	C0MDB211		-		-	No definition
0x400455B3	CAN0 message data byte 3 register 11	C0MDB311		-		-	No definition
0x400455B4H	CAN0 message data byte 45 register 11	C0MDB4511		-	-		No definition
0x400455B4	CAN0 message data byte 4 register 11	C0MDB411		-		-	No definition
0x400455B5	CAN0 message data byte 5 register 11	C0MDB511		-		-	No definition
0x400455B6H	CAN0 message data byte 67 register 11	C0MDB6711		-	-		No definition
0x400455B6	CAN0 message data byte 6 register 11	C0MDB611		-		-	No definition
0x400455B7	CAN0 message data byte 7 register 11	C0MDB711		-		-	No definition
0x400455B8H	CAN0 message data length register 11	C0MDLC11		-		-	0000xxxxB
0x400455B9H	CAN0 message configuration register 11	C0MCONF11		-		-	No definition
0x400455BAH	CAN0 message ID register 11	C0MIDL11		-	-		No definition
0x400455BCH		C0MIDH11		-	-		No definition
0x400455BEH	CAN0 message control register 11	C0MCTRL11		-	-		00x0000 000xx00B

Table 21-16. Register Access Types (8/9)

Address	register name	Symbol	read/ write	bit operation unit			Default
				1	8	16	
0x400455C0H	CAN0 message data byte 01 register 12	C0MDB0112	read/ write	-	-		No definition
0x400455C0H	CAN0 message data byte 0 register 12	C0MDB012		-		-	No definition
0x400455C1H	CAN0 message data byte 1 register 12	C0MDB112		-		-	No definition
0x400455C2H	CAN0 message data byte 23 register 12	C0MDB2312		-	-		No definition
0x400455C2H	CAN0 message data byte 2 register 12	C0MDB212		-		-	No definition
0x400455C3H	CAN0 message data byte 3 register 12	C0MDB312		-		-	No definition
0x400455C4H	CAN0 message data byte 45 register 12	C0MDB4512		-	-		No definition
0x400455C4H	CAN0 message data byte 4 register 12	C0MDB412		-		-	No definition
0x400455C5H	CAN0 message data byte 5 register 12	C0MDB512		-		-	No definition
0x400455C6H	CAN0 message data byte 67 register 12	C0MDB6712		-	-		No definition
0x400455C6H	CAN0 message data byte 6 register 12	C0MDB612		-		-	No definition
0x400455C7H	CAN0 message data byte 7 register 12	C0MDB712		-		-	No definition
0x400455C8H	CAN0 message data length register 12	C0MDLC12		-		-	0000xxxxB
0x400455C9H	CAN0 message configuration register 12	C0MCONF12		-		-	No definition
0x400455CAH	CAN0 message ID register 12	C0MIDL12		-	-		No definition
0x400455CCH		C0MIDH12		-	-		No definition
0x400455CEH	CAN0 message control register 12	C0MCTRL12		-	-		00x0000 000xx00B
0x400455D0H	CAN0 message data byte 01 register 13	C0MDB0113		-	-		No definition
0x400455D0H	CAN0 message data byte 0 register 13	C0MDB013		-		-	No definition
0x400455D1H	CAN0 message data byte 1 register 13	C0MDB113		-		-	No definition
0x400455D2H	CAN0 message data byte 23 register 13	C0MDB2313		-	-		No definition
0x400455D2H	CAN0 message data byte 2 register 13	C0MDB213		-		-	No definition
0x400455D3H	CAN0 message data byte 3 register 13	C0MDB313		-		-	No definition
0x400455D4H	CAN0 message data byte 45 register 13	C0MDB4513		-	-		No definition
0x400455D4H	CAN0 message data byte 4 register 13	C0MDB413		-		-	No definition
0x400455D5H	CAN0 message data byte 5 register 13	C0MDB513		-		-	No definition
0x400455D6H	CAN0 message data byte 67 register 13	C0MDB6713		-	-		No definition
0x400455D6H	CAN0 message data byte 6 register 13	C0MDB613		-		-	No definition
0x400455D7H	CAN0 message data byte 7 register 13	C0MDB713		-		-	No definition
0x400455D8H	CAN0 Message Data Length Register 13	C0MDLC13		-		-	0000xxxxB
0x400455D9H	CAN0 Message Configuration Register 13	C0MCONF13		-		-	No definition
0x400455DAH	CAN0 message ID register 13	C0MIDL13		-	-		No definition
0x400455DCH		C0MIDH13		-	-		No definition
0x400455DEH	CAN0 message control register 13	C0MCTRL13		-	-		00x0000 000xx00B

Table 21-16. Register Access Types (9/9)

Address	register name	Symbol	read/ write	bit operation unit			Default
				1	8	16	
0x400455E0H	CAN0 message data byte 01 register 14	C0MDB0114	read/ write	-	-		No definition
0x400455E0H	CAN0 message data byte 0 register 14	C0MDB014		-		-	No definition
0x400455E1H	CAN0 message data byte 1 register 14	C0MDB114		-		-	No definition
0x400455E2H	CAN0 message data byte 23 register 14	C0MDB2314		-	-		No definition
0x400455E2H	CAN0 message data byte 2 register 14	C0MDB214		-		-	No definition
0x400455E3H	CAN0 message data byte 3 register 14	C0MDB314		-		-	No definition
0x400455E4H	CAN0 message data byte 45 register 14	C0MDB4514		-	-		No definition
0x400455E4H	CAN0 message data byte 4 register 14	C0MDB414		-		-	No definition
0x400455E5H	CAN0 message data byte 5 register 14	C0MDB514		-		-	No definition
0x400455E6H	CAN0 message data byte 67 register 14	C0MDB6714		-	-		No definition
0x400455E6H	CAN0 message data byte 6 register 14	C0MDB614		-		-	No definition
0x400455E7H	CAN0 message data byte 7 register 14	C0MDB714		-		-	No definition
0x400455E8H	CAN0 Message Data Length Register 14	C0MDLC14		-		-	0000xxxxB
0x400455E9H	CAN0 message configuration register 14	C0MCONF14		-		-	No definition
0x400455EAH	CAN0 message ID register 14	C0MIDL14		-	-		No definition
0x400455ECH		C0MIDH14		-	-		No definition
0x400455EEH	CAN0 message control register 14	C0MCTRL14		-	-		00x0000 000xx00B
0x400455F0H	CAN0 message data byte 01 register 15	C0MDB0115		-	-		No definition
0x400455F0H	CAN0 message data byte 0 register 15	C0MDB015		-		-	No definition
0x400455F1H	CAN0 message data byte 1 register 15	C0MDB115		-		-	No definition
0x400455F2H	CAN0 message data byte 23 register 15	C0MDB2315		-	-		No definition
0x400455F2H	CAN0 message data byte 2 register 15	C0MDB215		-		-	No definition
0x400455F3H	CAN0 message data byte 3 register 15	C0MDB315		-		-	No definition
0x400455F4H	CAN0 message data byte 45 register 15	C0MDB4515		-	-		No definition
0x400455F4H	CAN0 message data byte 4 register 15	C0MDB415		-		-	No definition
0x400455F5H	CAN0 message data byte 5 register 15	C0MDB515		-		-	No definition
0x400455F6H	CAN0 message data byte 67 register 15	C0MDB6715		-	-		No definition
0x400455F6H	CAN0 message data byte 6 register 15	C0MDB615		-		-	No definition
0x400455F7H	CAN0 message data byte 7 register 15	C0MDB715		-		-	No definition
0x400455F8H	CAN0 Message Data Length Register 15	C0MDLC15		-		-	0000xxxxB
0x400455F9H	CAN0 message configuration register 15	C0MCONF15		-		-	No definition
0x400455FAH	CAN0 message ID register 15	C0MIDL15		-	-		No definition
0x400455FCH		C0MIDH15		-	-		No definition
0x400455FEH	CAN0 message control register 15	C0MCTRL15		-	-		00x0000 000xx00B

21.5.3 register bit configuration

Table 21-17. CAN Global Register Bit Configuration

Address	Symbol	Bit 7/15	bit 6/14	bit 5/13	bit 4/12	bit 3/11	bit 2/10	bit 1/9	bit 0/8
0x40045400H	C0GMCTRL(W)	0	0	0	0	0	0	0	ClearGOM
0x40045401H		0	0	0	0	0	0	SetEFSD	SetGOM
0x40045400H	C0GMCTRL(R)	0	0	0	0	0	0	EFSD	GOM
0x40045401H		MBON	0	0	0	0	0	0	0
0x40045406H	C0GMBT	0	0	0	0	0	0	0	ClearA BTTRG
0x40045407H		0	0	0	0	0	0	SetA BTCLR	SetA BTTRG
0x40045406H	C0GMBT	0	0	0	0	0	0	ABTCLR	ATTRG
0x40045407H		0	0	0	0	0	0	0	0
0x40045408H	C0GMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0
0x40045402H	C0GMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0

Note that the actual register address calculation refers to the following formula:

Register Address = Global Register Area Offset (dependent CH) + Offset address listed in the table above

When Notes (R) Read

(W) When writing

Table 21-18. Bit configuration of CAN module register (1/2)

Address	Symbol	Bit 7/15	bit 6/14	bit 5/13	bit 4/12	bit 3/11	bit 2/10	bit 1/9	bit 0/8
0x40045440H	C0MASK1L	CM1ID[7:0]							
0x40045441H		CM1ID [15:8]							
0x40045442H	C0MASK1H	CM1ID [23:16]							
0x40045443H		0	0	0	CM1ID [28:24]				
0x40045444H	C0MASK2L	CM2ID [7:0]							
0x40045445H		CM2ID [15:8]							
0x40045446H	C0MASK2H	CM2ID [23:16]							
0x40045447H		0	0	0	CM2ID [28:24]				
0x40045448H	C0MASK3L	CM3ID[7:0]							
0x40045449H		CM3ID [15:8]							
0x4004544AH	C0MASK3H	CM3ID [23:16]							
0x4004544BH		0	0	0	CM3ID [28:24]				
0x4004544CH	C0MASK4L	CM4ID [7:0]							
0x4004544DH		CM4ID [15:8]							
0x4004544EH	C0MASK4H	CM4ID [23:16]							
0x4004544FH		0	0	0	CM4ID [28:24]				
0x40045450H	C0CTRL(W)	Clear CCERC	Clear AL	Clear VALID	ClearPS MODE1	ClearPS MODE0	ClearOP MODE2	ClearOP MODE1	ClearOP MODE0
0x40045451H		SetC CERC	SetAL	0	SetPSM ODE1	SetPSM ODE0	SetOPM ODE2	SetOPM ODE1	SetOPM ODE0
0x40045450H	C0CTRL(R)	CCERC	AL	VALID	PSMOD E1	PSMOD E0	OPMOD E2	OPMOD E1	OPMOD E0
0x40045451H		0	0	0	0	0	0	RSTAT	TSTAT
0x40045452H	C0LEC(W)	0	0	0	0	0	0	0	0
0x40045452H	C0LEC(R)	0	0	0	0	0	LEC2	LEC1	LEC0
0x40045453H	C0INFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0
0x40045454H	C0ERC	TEC [7:0]							
0x40045455H		REPS	REC[7:0]						
0x40045456H	C0IE	0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0
0x40045457H		0	0	SetCIE5	SetCIE4	SetCIE3	SetCIE2	SetCIE1	SetCIE0
0x40045456H	C0IE	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
0x40045457H		0	0	0	0	0	0	0	0
0x40045458H	C0INS	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0
0x40045459H		0	0	0	0	0	0	0	0

Note that the actual register address calculation refers to the following formula:

Register Address = Global Register Area Offset (dependent CH) + Offset address listed in the table above

Note (R) Read

(W) Write Time

Table 21-18. Bit configuration of CAN module register (2/2)

Address	Symbol	Bit 7/15	bit 6/14	bit 5/13	bit 4/12	bit 3/11	bit 2/10	bit 1/9	bit 0/8
0x40045458H	C0INS	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0
0x40045459H		0	0	0	0	0	0	0	0
0x4004545AH	C0BRP	TQPRS [7:0]							
0x4004545CH	C0BTR	0	0	0	0	TSEG1 [3:0]			
0x4004545DH		0	0	SJW[1:0]		0	TSEG2 [2:0]		
0x4004545EH	C0LIPT	LIPT [7:0]							
0x40045460H	C0RGPT(W)	0	0	0	0	0	0	0	Clear ROVF
0x40045461H		0	0	0	0	0	0	0	0
0x40045460H	C0RGPT(R)	0	0	0	0	0	0	RHPM	ROVF
0x40045461H		RGPT[7:0]							
0x40045462H	C0LOPT	LOPT [7:0]							
0x40045464H	C0TGPT(W)	0	0	0	0	0	0	0	Clear TOVF
0x40045465H		0	0	0	0	0	0	0	0
0x40045464H	C0TGPT(R)	0	0	0	0	0	0	THPM	TOVF
0x40045465H		TGPT [7:0]							
0x40045466H	C0TS (W)	0	0	0	0	0	ClearT SLOCK	Clear TSSEL	Clear TSEN
0x40045467H		0	0	0	0	0	SetT SLOCK	SetT SSEL	Set TSEN
0x40045466H	C0TS	0	0	0	0	0	SLOCK	TSSEL	TSEN
0x40045467H		0	0	0	0	0	0	0	0

Note that the actual register address calculation refers to the following formula:

Register Address = Global Register Area Offset (dependent CH) + Offset address listed in the table above

Note (R) Read

(W) Write Time

Table 21-19. Bit configuration for message buffer registers

Address	Symbol	Bit 7/15	bit 6/14	bit 5/13	bit 4/12	bit 3/11	bit 2/10	bit 1/9	bit 0/8
0x400455x0H	C0MDB01m	Message Data (Byte0)							
0x400455x1H		Message Data (Byte1)							
0x400455x0H	C0MDB0m	Message data (byte 0)							
0x400455x1H	C0MDB1m	Message Data (Byte1)							
0x400455x2H	C0MDB23m	Message Data (bytes 2)							
0x400455x3H		Message Data (Byte3)							
0x400455x2H	C0MDB2m	Message Data (bytes 2)							
0x400455x3H	C0MDB3m	Message Data (Byte3)							
0x400455x4H	C0MDB45m	Message Data (bytes 4)							
0x400455x5H		Message Data (byte 5)							
0x400455x4H	C0MDB4m	Message Data (bytes 4)							
0x400455x5H	C0MDB5m	Message Data (byte 5)							
0x400455x6H	C0MDB67m	Message Data (byte6)							
0x400455x7H		Message Data (byte7)							
0x400455x6H	C0MDB6m	Message Data (byte6)							
0x400455x7H	C0MDB7m	Message Data (byte7)							
0x400455x8H	C0MDLCm	0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0
0x400455x9H	C0MCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0
0x400455xAH	C0MIDLm	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
0x400455xBH		ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
0x400455xCH	C0MIDHm	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
0x400455xDH		DE	0	0	ID28	ID27	ID26	ID25	ID24
0x400455xEH	C0MCTRLm (W)	0	0	0	ClearMOW	ClearIE	ClearDN	ClearTRQ	ClearRDY
0x400455xFH		0	0	0	0	SetIE	0	SetTRQ	SetRDY
0x400455xEH	C0MCTRLm(R)	0	0	0	MOW	BE	DN	TRQ	RDY
0x400455xFH		0	0	MUC	0	0	0	0	0

Note that the actual register address calculation refers to the following formula:

Register Address = Global Register Area Offset (dependent CH) + Offset address listed in the table above

Note 1. (R) Read

(W) Write Time

2. m = 0 to 15

3.x = 0 to F

21.6 Bit Settings/Clean-up

The CAN control register includes a register whose bit can be set or cleared through a CPU and a CAN interface.

If written directly to the following register, an operation error occurs. Do not write any values directly through bit operations, read/modify/write, or direct write to the target value.

- CAN Global Control Register (C0GMCTRL)
- CAN Global Automatic Block Transfer Control Register (C0GMBT)
- CAN Module Control Register (C0CTRL)
- CAN Module Interrupt Enable Register (C0IE)
- CAN Module Interrupt Interrupt State Register (C0INS)
- CAN Module Receive History List Register (C0RGPT)
- CAN Module Send History List Register (C0TGPT)
- CAN Module Timestamp Register (C0TS)
- CAN Message Control Register (C0MCTRLm)

Note m = 0 to 15

All 16 bits in the above register can be read by common methods. Use the procedure described in FIGS. 21-23 below to set or clear the lower 8 bits in these registers.

The setting or clearing of the lower 8 bits in the above register is performed in combination with the higher 8 bits (see 16 bits of data after the write operation in FIGS. 21-24). FIGS. 21-23 shows the relationship between the value of the set bit or clear bit and the set/clear/no change operation in the corresponding register.

Figure 21-23. Example of bit set-up/purge operation

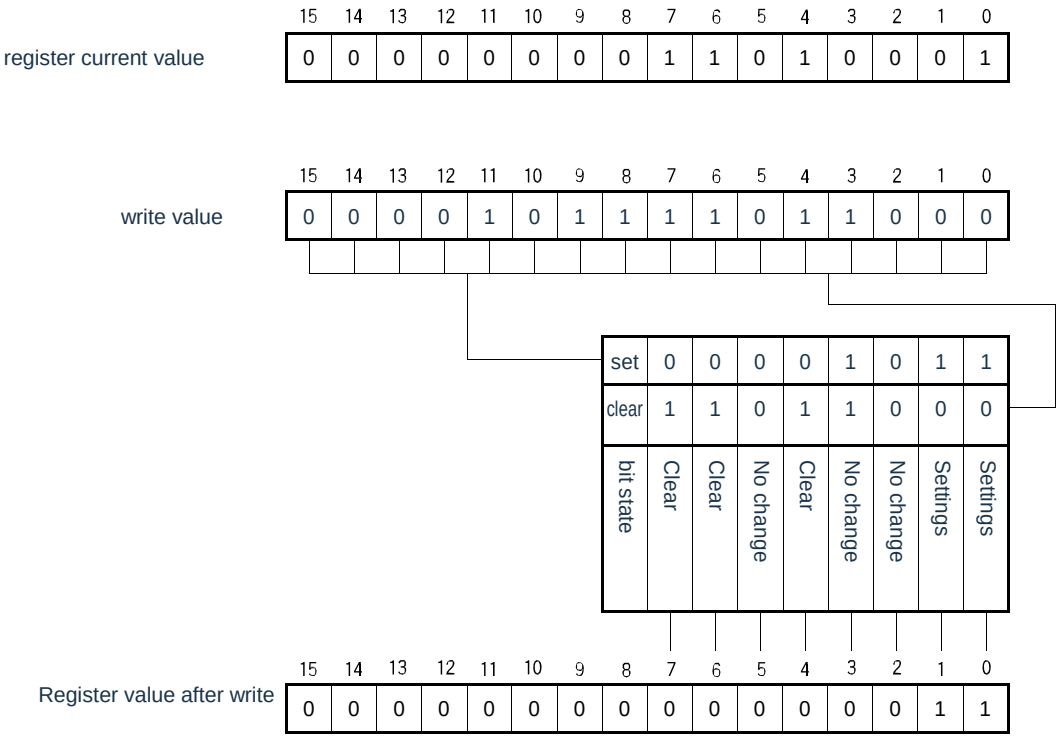


Figure 21-24. 16-bit data in a write operation

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Set7	Set6	Set5	Set4	Set3	Set2	Set1	Set0	Clear7	Clear6	Clear5	Clear4	Clear3	Clear2	Clear1	Clear0

Seton	Clearn	Set/Cleared n Status
0	0	No change
0	1	0
1	0	1
1	1	No change

Note n = 0 to 7

21.7 control register

Note m = 0 to 15

(1) Peripheral clock selection register (PER0)

PER0 is used to enable or disable each peripheral hardware macro. The PER0 can be set with 8-bit memory manipulation instructions.

The reset signal clears this register to 00H.

Note Set the PCKSEL register before you start operating each peripheral hardware device.

Figure 21-25. Format of the peripheral clock selection register (PER0)

	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	CAN0EN	TM40EN

CAN0	Control of input clock of CAN module
0	Stop provide an input clock. · Cannot write the SFR used by CAN0. · CAN0 is in a reset state.
1	Provides an input clock. · Can read and write SFRs used by CAN0.

(2) CAN Global Module Control Register (C0GMCTRL)

The C0GMCTRL register is used to control the operation of the CAN module.

Figure 21-26. CAN Global Module Control Register Format (C0GMCTRL) (1/2)

(a) read

C0GMCTRL	15	14	13	12	11	10	9	8
	MBON	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	EFSD	GOM

(a) write

C0GMCTRL	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	Set EFSD	Set GOM
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear GOM

(a) read

MBON	Message cache register, bit enable for transmitting/receiving history list register
0	Disable read/write to message cache registers and send/receive history list registers
1	Enable read/write to message cache registers and send/receive history list registers

Note 1. When MBON bits are cleared (to 0), software access message buffers (C0MDB0m, C0MDB1m, C0MDB01m, C0MDB2m, C0MDB3m, C0MDB23m, C0MDB4m, C0MDB5m, C0MDB45m, C0MDB6m, C0MDB7m, C0MDB67m, C0MDLCm, C0MCONFm and C0MIDLm) are disabled.

2. This bit is read-only, writing 1 to MBON when it is 0, the value of MBON does not change, and access to the message cache register or register-related send or receive history is prohibited.

Note When the CAN module enters CAN sleep mode/CAN stop mode or the GOM bit is cleared (to 0), the MBON bit is cleared (to 0).

The MBON bit is set to 1 when the CAN sleep mode/CAN stop mode is released or the GOM bit is set.

Figure 21-26. CAN Global Module Control Register Format (C0GMCTRL) (2/2)

EFSD	bit-enabled forcible shutdown
0	Disable Shutdown by Writing GOM=0
1	Enable to close by writing GOM=0.

Note To request a forced shutdown, you must clear the GOM bit of 0 in subsequent operations and write immediately after the EFSD bit is set to 1. If an access to another register, including a read C0GMCTRL register, is performed without clearing the GOM bit immediately after the EFSD bit is set to 1, the EFSD bit is forced to clear to 0 and the forced shutdown request is invalid.

When performing DMA, requests for forced shutdown may be ignored. Be sure to read the EFSD bit and confirm that forced shutdown is enabled before issuing a forced shutdown request. If forced shutdown cannot be enabled because DMA is in progress, it is recommended to temporarily stop DMA.

GOM	global operation mode bit
0	Disabled in CAN module operation
1	The CAN module is activated

Note that GOM bits are cleared 0 only after initialization mode or immediate EFSD is set to 1.

(b) write

SetEFSD	EFSD bit settings
0	The EFSD bit does not change
1	EFSD is set to 1

SetGOM	ClearGOM	GOM bit settings
0	1	GOM bit cleared to 0
1	0	GOM bit set to 1
Other Values		GOM bit not changed

Note that the GOM bit and the EFSD bit are set separately.

(3) CAN Global Module Clock Select Register (C0GMCS)

The system clock used by C0GMCS and C1GMCS to select the CAN module.

Figure 21-27. CAN Global Module Clock Select Register Format (C0GMCS)

C0GMCS	7	6	5	4	3	2	1	0
	0	0	0	0	CCP3	CCP2	CCP1	CCP0

CCP3	CCP2	CCP1	CCP1	CAN Module System Clock (fCANMOD)
0	0	0	0	f_{CAN}/1
0	0	0	1	f_{CAN}/2
0	0	1	0	f_{CAN}/3
0	0	1	1	f_{CAN}/4
0	1	0	0	f_{CAN}/5
0	1	0	1	f_{CAN}/6
0	1	1	0	f_{CAN}/7
0	1	1	1	f_{CAN}/8
1	0	0	0	f_{CAN}/9
1	0	0	1	f_{CAN}/10
1	0	1	0	f_{CAN}/11
1	0	1	1	f_{CAN}/12
1	1	0	0	f_{CAN}/13
1	1	0	1	f_{CAN}/14
1	1	1	0	f_{CAN}/15
1	1	1	1	f_{CAN}/16 (default)

Note fCAN: Clock provided to CAN (fMAIN)

(4) CAN Global Automatic Block Transfer Control Register (C0GMBT)

C0GMABT registers are used to control automatic block transfer (ABT) operations

Figure 21-28. Format of CAN Global Automatic Block Transfer Control Register (C0GMBT) (1/2)

(a) read

C0GMBT	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	ABTCLR	ATTRG

(a) write

C0GMBT	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	Set ABTCLR	Set ATTRG
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	ABTCLR	ClearA BTTRG

Note that before changing the ABT's normal operation mode to initialization mode, make sure that the C0GMABT register is set to default (0000H) and that the C0GMABT register is definitely initialized to default (0000H).

(a) read

ABTCLR	Automatic block transfer clear status bit
0	Clear Automatic Transfer Engine Complete
1	Automatic Transmission Engine is being cleared

Notes. 1. Set the ABTCLR bit to 1 when ABTTRG is cleared to 0.
There is no guarantee when ABTTRG is 1 and ABTCLR is 1.
2. When the automatic block transfer engine is cleared by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared to 0 once the requested purge process is complete.

Figure 21-28. Format of CAN Global Automatic Block Transfer Control Register (C0GMBT) (2/2)

ATTRG	Automatic Block Transmit Status Bit
0	Automatic block transfer stopped
1	Automatic block transfer in progress

Note Do not set the ABTTRG bit (ABTTRG =1) in initialization mode. If the ABTTRG bit is set in the initialization mode, the operation cannot be guaranteed after the CAN module enters the normal operation mode using ABT. Do not set the ABTTRG bit (1) when C0CTRL.TSTAT is 1. Prior to setting ABTTRG, confirm TSTAT = 0.

(b) write

SetABTCLR	Auto Block Transmission Engine Clear Request Bit
0	The Auto Block Transmission engine is idle or operating
1	Request to clear automatic block transfers. When the automatic block transfer engine is cleared, automatic block transfer starts from message cache 0 after setting ABTTRG to 1

SetABTTRG	ClearABTTRG	Automatic block transfer start bit
0	1	Request to stop automatic block transfer
1	0	Request to start automatic block transfer
Other Values		The ABTTRG bit does not change.

Note When receiving messages from other nodes or sending messages other than ABT messages (message buffers 8 through 15), transmission may not start immediately even if the ABTTRG bit is set to 1.

Even if the ABTTRG bit is cleared to 0, the transfer will not be aborted until the ABT message transfer currently in transit is complete (successful or not). After that, the transmission is aborted.

(5) CAN Global Automatic Block Transfer Delay Set-up Register (C0GMABTD)

The C0GMABTD register is used to set the time interval for the transmission of message buffer data allocated to the ABT in normal operation mode of the ABT.

Figure 21-29. Format of the CAN Global Automatic Block Transfer Delay Setting Register (C0GMABTD)

C0GMABTD	7	6	5	4	3	2	1	0
	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

ABTD3	ABTD2	ABTD1	ABTD0	Data frame interval for automatic block transmission (units: Data bit time (DBT))
0	0	0	0	0 DBT (default)
0	0	0	1	2^5DBT
0	0	1	0	2^6DBT
0	0	1	1	2^7DBT
0	1	0	0	2^8DBT
0	1	0	1	2^9DBT
0	1	1	0	2^{10}DBT
0	1	1	1	2^{11} DBT
1	0	0	0	2^{12} DBT
Other Values				prohibited

Note 1. Do not change the contents of C0ABTTRG when GMABTD is 1.

2. The time that ABT messages are actually transmitted to the CAN bus varies depending on the status transmitted from other workstations or the manner in which ABT messages (message buffers 8 to 15) are requested to be transmitted.

(6) CAN module mask register (C0MASKaL, C0MASKaH) (a = 1, 2, 3, or 4)

The C0MASKaL and C0MASKaH registers extend the number of entries into the same packet cache by comparing the masked portion with the packet ID and invalidating the ID of the masked portion.

Figure 21-30. Format of the CAN module mask register (C0MASKaL, C0MASKaH) (a = 1, 2, 3, or 4) (1/2)

- CAN Module Masking 1 register (C0MASK1L, C0MASK1H)

	15	14	13	12	11	10	9	8
C0MASK1L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

	15	14	13	12	11	10	9	8
C0MASK1H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN Module Masking 2 register (C0MASK2L, C0MASK2H)

	15	14	13	12	11	10	9	8
C0MASK2L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

	15	14	13	12	11	10	9	8
C0MASK2H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

Figure 21-30. Format of the CAN module mask register (C0MASKaL, C0MASKaH) (a = 1, 2, 3, or 4) (2/2)

- CAN Module Masking 3 register (C0MASK3L, C0MASK3H)

	15	14	13	12	11	10	9	8
C0MASK3L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

	15	14	13	12	11	10	9	8
C0MASK3H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN Module Masking 4 register (C0MASK4L, C0MASK4H)

	15	14	13	12	11	10	9	8
C0MASK4L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

	15	14	13	12	11	10	9	8
C0MASK4H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

CMID0 to CMID28	Set screening mode for ID bits
0	The ID bit of the message buffer set from CMID0 to CMID28 is compared to the ID bit of the received message frame.
1	The ID bit of the message buffer set from CMID0 to CMID28 is not compared to the ID bit of the received message frame (they are masked)

Note masking is always determined by the 29-bit ID length. If the mask is assigned to a message with a standard ID, then CMID0 to CMID17 bits are ignored; Therefore, only CMID18 to CMID28 for received IDs are masked. The same mask can be used for standard and extended IDs.

(7) CAN Module Control Register (C0CTRL)

The C0CTRL register is used to control the operation mode of the CAN module.

Figure 21-31. Format of CAN Module Control Register (C0CTRL) (1/4)

(a) read

C0CTRL	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	RSTAT	TSTAT
	7	6	5	4	3	2	1	0
	CCERC	AL	VALID	PSMODE1	PSMODE0	OPMODE2	OPMODE1	OPMODE0

(b) write

C0CTRL	15	14	13	12	11	10	9	8
	Set CCERC	Set AL	0	Set PSMODE1	Set PSMODE0	Set OPMODE2	Set OPMODE1	Set OPMODE0
	7	6	5	4	3	2	1	0
	Clear CCERC	Clear AL	Clear VALID	ClearP SMODE1	ClearP SMODE0	ClearO PMODE2	ClearO PMODE1	ClearO PMODE0

(a) read

RSTAT	Receive status bit
0	Reception stopped
1	Receive in progress

Note - The RSTAT bit is set to 1 under the following conditions:

- The SOF bit of the received frame is detected
- Arbitration loss occurs in transmission frames

- The RSTAT bit is cleared 0 under the following conditions (timing):
  - Hidden bit detected at second bit in interframe space
  - Transition to initial mode, first bit in interframe space

Figure 21-31. Format of CAN Module Control Register (C0CTRL) (2/4)

TSTAT	Transmit Status Bit
0	Transmission stopped
1	Transmission in progress

Note - The RSTAT bit is set to 1 under the following conditions (timing):

- - SOF bits of the transmitted frame are detected

- The RSTAT bit is cleared 0 under the following conditions (timing):
  - During transition to bus off state
  - Arbitration loss occurs in transmission frames
  - Hidden bit detected at second bit in interframe space
  - The transition to the initial mode, occurs at the first bit in the inter-frame space

CCERC	Error counter clearing bit
0	The C0 ERC and C0 INFO registers are not cleared in the initialization mode.
1	The C0 ERC and C0 INFO registers are cleared in the initialization mode

Note 1. The CCERC bit is used to clear the C0ERC and C0INFO registers for re-initialization or forced recovery from the bus-off state. This bit can only be set to 1 in initialization mode.

2. When the C0ERC and C0INFO registers are cleared, the CCERC bit is also automatically cleared to 0.
3. The CCERC bit is set to 1 when you request to change the initialization mode to operational mode.
4. The received data may be corrupted when the CCERC bit is set to 1 immediately after entering INIT mode from Self-Test mode.

AL	Sets the bit of the operation when the quorum is lost
0	When an arbitration loss occurs in single-shot mode, no retransmission is performed
1	Perform a retransmission when an arbitration loss occurs in single-shot mode.

Note The AL bit is only valid in single-shot mode.

VALID	Valid Receive Message Frame Detection Bits
0	No valid message frames have been received since the VALID bit was last cleared to 0
1	When the VALID bit is cleared to 0, a valid message frame is received

Note 1. Detecting valid received message frames does not depend on storage in the received message buffer (data frame) or the sent message buffer (remote frame).

2. Clear the VALID bit before changing initialization mode to operation mode
3. If only two CAN nodes are connected to the CAN bus, one transmitting the message frame in normal operation mode and the other in receive-only mode, the VALID bit is not set to 1 until the transmission node enters an error passive state because no response is generated in receive-only mode.
4. To clear the VALID bit, set the Clear VALID bit to 1, then confirm that the VALID bit is cleared, and if it is not, perform the clear process again.

Figure 21-31. Format of CAN Module Control Register (C0CTRL) (3/4)

PSMODE1	PSMODE0	power saving mode
0	0	No power saving mode is selected
0	1	CAN sleep mode
1	0	Set Off
1	1	CAN Stop Mode

Note 1. Transition to or wake from the CAN stop mode through the CAN sleep mode. The request to directly execute the CAN stop mode of entry and exit will be ignored.

2. After exiting power-saving mode, you must check the MBON flag for C0GMCTRL before accessing the message buffer.

3. The request for CAN sleep mode remains pending until the software cancels or enters the appropriate bus condition (bus idle). The software can check the actual status by reading the PSMODE

OPMODE2	OPMODE1	OPMODE0	operation mode
0	0	0	No operation mode selected (CAN mode in initialization mode)
0	0	1	Normal operation mode
0	1	0	Normal mode of operation with automatic block transfer (Normal
0	1	1	Receive mode only
1	0	0	single-shot mode
1	0	1	self-test mode
Other Values			Disable from setting

Note: It may take some time to transfer to initialization or power saving mode. Be sure to verify the successful mode change by reading before continuing.

Note that the OPMODE[2:0] bit is read-only in CAN sleep and CAN stop mode.

(b) Writing

SetCCERC	Set CCERC bit
1	Set the CCERC bit to 1
0	CCERC hold

SetAL	ClearAL	Setting AL Bits
0	1	AL cleared 0.
1	0	AL is set to 1
Other Values		AL remains unchanged

Figure 21-31. Format of CAN Module Control Register (C0CTRL) (4/4)

ClearVALID	Set VALID bit
0	VALID hold
1	VALID cleared 0.

SetPSMODE0	ClearPSMODE0	Set PSMODE0 bit
0	1	PSMODE0 cleared 0.
1	0	PSMODE0 is set to 1.
Other Values		PSMODE0 is unchanged.

SetPSMODE1	ClearPSMODE1	Set PSMODE1 bit
0	1	PSMODE1 Clear 0
1	0	PSMODE1 Set 1.
Other Values		PSMODE1 is unchanged.

SetOPMODE0	ClearOPMODE0	Set OPMODE0 bit
0	1	OPMODE0 clear 0.
1	0	OPMODE0 Set 1.
Other Values		OPMODE0 does not change.

SetOPMODE1	ClearOPMODE1	Set OPMODE1 bit
0	1	OPMODE1 clear 0.
1	0	OPMODE1 Set 1.
Other Values		OPMODE1 is unchanged.

SetOPMODE2	ClearOPMODE2	Set OPMODE2 bit
0	1	OPMODE2 clear 0.
1	0	OPMODE2 Set 1.
Other Values		OPMODE2 is unchanged.

(8) CAN module last error code register (C0LEC)

The C0LEC register provides error information for the CAN protocol.

Figure 21-32. Format of the CAN module last error code register (C0LEC)

	7	6	5	4	3	2	1	0
C0LEC	0	0	0	0	0	LEC2	LEC1	LEC0

Note 1. The contents of the C0 LEC register are not cleared when the CAN switches from the operation mode to the initialization mode.

2. If you attempt to write a value other than 00H to the C0LEC register by software, the access is ignored.

LEC2	LEC1	LEC0	Last CAN protocol error message
0	0	0	No errors
0	0	1	Fill Error
0	1	0	Format Error
0	1	1	ACK Error
1	0	0	Bit error (CAN module attempts to transmit hidden bits (except arbitration fields) as part of the transmission message, but the value on the CAN bus is an explicit bit)
1	0	1	Bit error (CAN module attempts to transmit explicit bits as part of a transmission message, ACK bits, error frame, or overloaded frame, but the value on the CAN bus is an implicit bit)
1	1	0	CRC error
1	1	1	No definition

(9) CAN Module Information Register (C0INFO)

The C0INFO register indicates the state of the CAN module

Figure 21-33. Format of the CAN Module Information Register (C0INFO)

	7	6	5	4	3	2	1	0
C0INFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0

BOFF	bus off status bit
0	Non-bus off state (transmission error counter less than 255) (transmission counter value less than 256)
1	Bus Off State (Transmission Error Counter greater than 255) (Transmission Counter value equal to or greater than 256)

TECS1	TECS0	Send error counter status bit
0	0	Send error counter value is less than warning level (<96).
0	1	The value of the send error counter is within the warning range (96 to 127).
1	0	No definition
1	1	The value of the send error counter is in the error passive or bus closed state

RECS1	RECS0	Receive error counter status bit
0	0	Receive error counter value less than warning level (<96)
0	1	The value of the receive error counter is in the warning range (96 to 127).
1	0	No definition
1	1	The value of the receive error counter is within the error passive range (128).

(10) CAN Module Error Counter Register (C0ERC)

The C0 ERC register records the count value of the transmit/receive error counter.

Figure 21-34. CAN Module Error Counter Register Format (C0ERC)

C0ERC

15	14	13	12	11	10	9	8
REPS	REC6	REC5	REC4	REC3	REC2	REC1	REC0
7	6	5	4	3	2	1	0
TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0

REPS	Receive error passive status bit
0	Receive error counter is not error passive (<128)
1	Receive error counter in error passive range (128)

REC6-REC0	Receive error counter bit
0-127	Number of receive errors. These bits reflect the status of the receive error counter. The number of errors is defined by the CAN protocol.

Note that REC [6:0] of the receive error counter is not valid in the receive error passive state (RECS [1:0] =11B)

TEC7-TEC0	Send error counter bit
0-255	Number of transmission errors. These bits reflect the status of the transmission error

Note that TEC [7:0] sending error counters is not valid in bus off state (BOFF =1)

(11) CAN Module Interrupt Enable Register (C0IE)

The C0 IE register is used to enable or disable interruption of the CAN module.

Figure 21-35. CAN Module Interrupt Enable Register Format (C0IE) (1/2)

(a) read

C0IE

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0

(b) write

C0IE

15	14	13	12	11	10	9	8
0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
7	6	5	4	3	2	1	0
0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0

(a) read

CIE5-CIE0	CAN module interrupt enable bit
0	Interrupt output corresponding to interrupt state register C0INTS[5:0] bit is disabled
1	Interrupt output corresponding to interrupt state register C0INTS[5:0] bit enabled

(b) write

SetCIE5	ClearCIE5	Set CIE5 bit
0	1	CIE5 clear 0.
1	0	CIE5 Set 1.
Other Values		CIE5 hasn't changed.

SetCIE4	ClearCIE4	Set CIE4 bit
0	1	CIE4 bit clear 0.
1	0	CIE4 position 1.
Other Values		CIE4 is unchanged.

Figure 21-35. CAN Module Interrupt Enable Register Format (C0IE) (2/2)

SetCIE3	ClearCIE3	Set CIE3 bit
0	1	CIE3 clear 0.
1	0	CIE3 Set 1.
Other Values		CIE3 is unchanged.

SetCIE2	ClearCIE2	Set CIE2 bit
0	1	CIE2 clear 0.
1	0	CIE2 Set 1.
Other Values		CIE2 is unchanged.

SetCIE1	ClearCIE1	Set CIE1 bit
0	1	CIE1 clear 0.
1	0	CIE1 Set 1.
Other Values		CIE1 is unchanged.

SetCIE0	ClearCIE0	Set CIE0 bit
0	1	CIE0 clear 0.
1	0	CIE0 Set 1.
Other Values		CIE0 is unchanged.

(12) CAN Module Interrupt State Register (C0INS)

C0INTS register indicates CAN module interrupt state

Figure 21-36. CAN Module Interrupt State Register Format (C0INS)

(a) read

	15	14	13	12	11	10	9	8
C0INTS	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0

(b) write

	15	14	13	12	11	10	9	8
C0INTS	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0

(a) read

UNITS5-UNITS0	CAN interrupt status bit
0	No related interrupt source events are pending.
1	A related interrupt source event is pending.

Interrupt StatusBit	Related interrupt source events
CINTS5	Wake from CAN sleep mode [1]
CINTS4	arbitration loss interrupt
CINTS3	CAN protocol error interrupt
CINTS2	CAN Error Status Interrupt
CINTS1	The completion interrupt of the message buffer m received by the valid message frame
CINTS0	Interrupt normal completion of message frame transfer from message buffer m

1 The CINTS5 bit is set only when the CAN module wakes from the CAN sleep mode via CAN bus operation. The CINTS5 bit is not set when the software releases the CAN sleep mode.

(b) write

ClearUNITS5-UNITS0	CINTS0 to CINTS5 bit
0	CINTS0 to CINTS5 has not changed
1	CINTS0 to CINTS5 cleared 0

Note: When each status needs to be confirmed during interrupt handling, use the software to clear the status bits of this register, as they will not be automatically cleared.

(13) CAN Module Bit Rate Scaling Register (C0BRP)

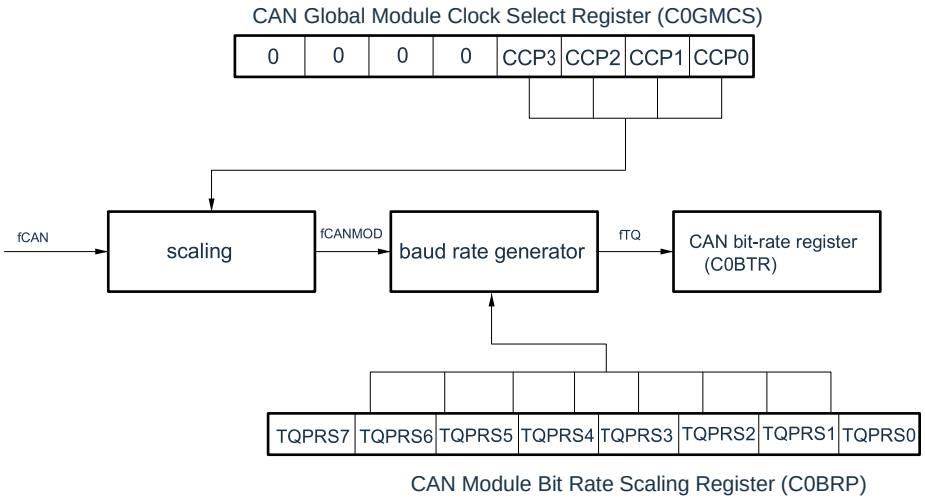
The C0BRP register is used to select the CAN protocol layer basic clock (fTQ). baud rate set to C0BTR register

Figure 21-37. CAN Module Bit Rate Scaling Register Format (C0BRP)

	7	6	5	4	3	2	1	0
C0BRP	TQPRS7	TQPRS6	TQPRS5	TQPRS4	TQPRS3	TQPRS2	TQPRS1	TQPRS0

TQPRS7-TQPRS0	CAN Protocol Layer Base System Clock (fTQ).
0	f_{CANMOD}/1
1	f_{CANMOD}/2
:	:
n	f_{CANMOD}/(n+1)
:	:
255	f_{CANMOD}/256 (default)

Figure 21-38. CAN Global Clock



Note: The C0 BRP register can only be written in initialization mode

Note fCAN: Provide the CAN with a clock (f_{MAIN})

f_{CANMOD}: CAN module system clock

f_{TQ}: CAN protocol layer basic system clock

(14) CAN Module Bit Rate Register (C0BTR)

The C0BTR register controls the bit time of the baud rate.

Figure 21-39. CAN Module Bit Rate Register Format (C0BTR) (1/2)

	15	14	13	12	11	10	9	8
C0BTR	0	0	SJW1	SJW0	0	TSEG22	TSEG21	TSEG20
	7	6	5	4	3	2	1	0
	0	0	0	0	TSEG13	TSEG12	TSEG11	TSEG10

SJW1	SJW0	Length of the synchronization jump
0	0	1TQ
0	1	2TQ
1	0	3TQ
1	1	4TQ (default)

TSEG22	TSEG21	TSEG20	Length of time period 2
0	0	0	1TQ
0	0	1	2TQ
0	1	0	3TQ
0	1	1	4TQ
1	0	0	5TQ
1	0	1	6TQ
1	1	0	7TQ
1	1	1	8TQ (default)

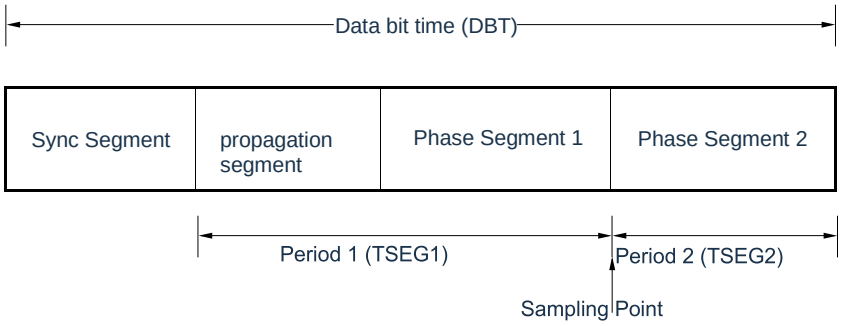
Figure 21-39. CAN Module Bit Rate Register Format (C0BTR) (2/2)

TSEG13	TSEG12	TSEG11	TSEG10	Length of time period 1
0	0	0	0	Disable from setting
0	0	0	1	2TQ[1]
0	0	1	0	3TQ[1]
0	0	1	1	4TQ
0	1	0	0	5TQ
0	1	0	1	6TQ
0	1	1	0	7TQ
0	1	1	1	8TQ
1	0	0	0	9TQ
1	0	0	1	10TQ
1	0	1	0	11TQ
1	0	1	1	12TQ
1	1	0	0	13TQ
1	1	0	1	14TQ
1	1	1	0	15TQ
1	1	1	1	16TQ (default)

1 These settings must be performed when the C0 BRP register is 00 H

Note TQ = 1/f_{TQ} (f_{TQ}: CAN protocol layer base system clock)

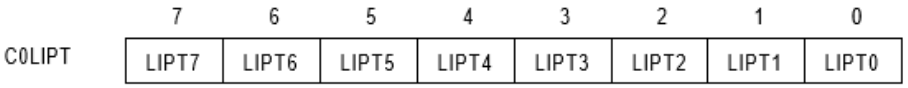
Figure 21-40. Data bit time



(15) CAN module last input pointer register (C0LIPT)

The C0LIPT register indicates the number of message buffers at which the data frame or remote frame was last stored.

Figure 21-41. CAN Module Last Input Pointer Register Format (C0LIPT)



LIPT7-LIPT0	Last Input Pointer Register (C0LIPT)
0 to 15	When the C0LIPT register is read, the contents of the element indexed by the last input pointer (LIPT) of the received history list are read. This content indicates the number of message buffers at which the data frame or remote frame was last stored.

Note: If the data or remote frame is never stored in the message buffer, the read value of the C0LIPT register is not defined. If the RHPM bit of the C0RGPT register is set to 1 after the CAN module changes from the initialization mode to the operation mode, the read value of the C0LIPT register is not defined.

(16) CAN module receive history list register (C0RGPT)

The C0RGPT register is used to read the received history list.

Figure 21-42. CAN Module Receive History List Register Format (C0RGPT) (1/2)

(a) read

C0RGPT	15	14	13	12	11	10	9	8
	RGPT7	RGPT6	RGPT5	RGPT4	RGPT3	RGPT2	RGPT1	RGPT0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	RHPM	ROVF

(b) write

C0RGPT	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear ROVF

(a) read

RGPT7-RGPT0	Receive history list get pointer
0 to 15	When reading the C0RGPT register, the contents of the index element of the Receive History List Acquisition Pointer (RGPT) are read. This content indicates the number of message buffers in which data frames or remote frames are stored.

RHPM [1]	Receive history list pointer match
0	The receive history list has at least one message buffer number that has not been read
1	Receive history list has no message buffer number not yet read

1 Invalid RGPT0 to RGPT7 read value when RHPM is 1

ROVF[1]	Receive history list overflow bit
0	All message buffer numbers that have not yet been read will be preserved. The received and stored message cache numbers in all new data frames or remote frames have been recorded in the received history list (the received history list has an empty element)
1	At least 23 entries have been stored since the host processor last serviced the RHL (i.e., read the C0RGPT). The first 22 entries are stored sequentially, and each time a newly received message is stored, the last entry can be overwritten, because when the ROVF bit is set, all buffer numbers are stored in the LIPT-1 location. As a result, the order received cannot now be fully restored.

1 If ROVF is set, RHPM is no longer purged while messages are stored, but if the software reads all the entries for C0RGPT, RHPM is still set

Figure 21-42. CAN module receiving history list register format (C0RGPT) (2/2)

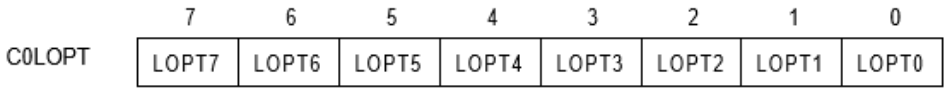
(b) write

ClearROVF	Set ROVF bit
0	ROVF doesn't change.
1	ROVF clear 0.

(17) CAN module last output pointer register (C0LOPT)

The C0LOPT register indicates the number of message buffers that were last transferred to a data frame or remote frame.

Figure 21-43. CAN Module Last Output Pointer Register Format (C0LOPT)



LOPT7-LOPT0	Send History List Last Output Pointer (LOPT)
0 to 15	When the C0LOPT register is read, the contents of the element indexed by the last outbound pointer (LOPT) of the received history list are read. This content indicates the number of message buffers that were last transferred to a data frame or remote frame.

Note: If a data frame or remote frame is never transmitted from a message buffer, the value read from the C0LOPT register is not defined. If the THPM bit is set to 1 after the CAN module has changed from the initialization mode to the operation mode, the read value for the C0LOPT register is not defined.

(18) CAN Module Send History List Register (C0TGPT)

The C0TGPT register is used to read the list of transmission history.

Figure 21-44. CAN Module Send History List Register Format (C0TGPT) (1/2)

(a) read

C0TGPT

15	14	13	12	11	10	9	8
TGPT7	TGPT6	TGPT5	TGPT4	TGPT3	TGPT2	TGPT1	TGPT0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	THPM	TOVF

(b) write

C0TGPT

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

(a) read

TGPT7-TGPT0	Send history list read pointer
0 to 15	When reading the C0TGPT register, the contents of the Read Pointer (TGPT) index element of the sending history list are read. This content indicates the number of message buffers that were last transferred to a data frame or remote frame.

THPM[1]	Transmit History PointMatch
0	The send history list has at least one message buffer number that has not been read.
1	Send a history list to a message buffer number that has not been read.

1 Invalid TGPT0 to TGPT7 reading value when THPM is 1.

TOVF	Send history list overflow bit
0	All message buffer numbers that have not yet been read will be preserved. All numbers of the message buffer to which new data frames or remote frames are transferred are recorded in the Transmission History List (which has an empty element).
1	At least 7 entries (read C0TGPT) have been stored since the host processor last serviced the THL. The first six entries are stored in order, and each time a message is transmitted, the last entry can be overwritten, because when the TOVF bit is set, all buffer numbers are stored in the LOPT-1 location. As a result, the transmission

1. If TOVF is set, THPM is no longer cleared during message transfer, but if the software reads all the entries for CTGPT, THPM is still set.

Note: In normal operation mode with ABT, the transfer history of message buffer 0 to 7 is not put into the list

Figure 21-44. CAN Module Send History List Register Format (C0TGPT) (2/2)

(b) write

ClearTOVF	Set TOVF bit
0	TOVF hasn't changed
1	TOVF clear 0.

(19) CAN Module Timestamp Register (C0TS)

C0 TS register for controlling timestamp function

Figure 21-45. CAN Module Timestamp Register Format (C0TS) (1/2)

(a) read

C0TS

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

(b) write

C0TS

15	14	13	12	11	10	9	8
0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN
7	6	5	4	3	2	1	0
0	0	0	0	0	ClearT SLOCK	Clear TSSEL	Clear TSEN

Note: When the CAN module is in the normal operation mode with ABT, the locking function with the timestamp function must not be used.

(a) read

SLOCK	Timestamp lock function enable bit
0	Timestamp locking stopped. Switches the TSOUT signal each time a selected timestamp capture event occurs.
1	The timestamp lock function enables. Switches the TSOUT signal each time the selected timestamp capture event occurs. However, when the data frame correctly receives the message buffer 0, the TSOUT output signal is locked [1].

1 TSEN bit auto clear 0.

Figure 21-45. CAN Module Timestamp Register Format (C0TS) (2/2)

TSSEL	Timestamp capture event selection bit
0	Timestamp capture event in SOF.
1	Timestamp capture event last bit in EOF

TSEN	TSOUT signal operation setting bit
0	TSOUT signal rollover operation is disabled.
1	TSOUT signal flip operation enabled.

Note: The signal TSOUT is output from the CAN macro to the timer resource, depending on the implementation. See Chapter 6, TIMER Array Units.

(b) write

SetTSLOCK	ClearTSLOCK	TSLOCK settings bit
0	1	TSLOCK clear 0.
1	0	TSLOCK Set 1.
Other Values		TSLOCK does not change.

SetTSSEL	ClearTSSEL	TSSEL settings bit
0	1	TSSEL clear 0.
1	0	TSSEL 1.
Other Values		TSSEL does not change

SetTSEN	ClearTSEN	TSEN settings bit
0	1	TSEN clear 0.
1	0	TSEN Set 1.
Other Values		TSEN does not change.

(20) CAN Message Data Byte Register (C0MDBxm) (x = 0 to 7), (C0MDBzm) (z = 01, 23, 45, 67)

The C0MDBxm, C0MDBzm registers are used to store the data of the sending/receiving message. The C0MDBxm register can be accessed in 8-bit units. The C0MDBzm register can access the 16-bit unit C0MDBxm register.

Figure 21-46. CAN message data byte register format (C0MDBxm) (x = 0 to 7), (C0MDBzm) (z = 01, 23, 45, 67) (1/2)

Address: Reference Table 21-16
Reset value: No definition
R/W

- C0MDBxm register

C0MDB0m

7	6	5	4	3	2	1	0
MDATA07	MDATA06	MDATA05	MDATA04	MDATA03	MDATA02	MDATA01	MDATA00

C0MDB1m

7	6	5	4	3	2	1	0
MDATA17	MDATA16	MDATA15	MDATA14	MDATA13	MDATA12	MDATA11	MDATA10

C0MDB2m

MDATA27	MDATA26	MDATA25	MDATA24	MDATA23	MDATA22	MDATA21	MDATA20

C0MDB3m

MDATA37	MDATA36	MDATA35	MDATA34	MDATA33	MDATA32	MDATA31	MDATA30

C0MDB4m

MDATA47	MDATA46	MDATA45	MDATA44	MDATA43	MDATA42	MDATA41	MDATA40

C0MDB5m

MDATA57	MDATA56	MDATA55	MDATA54	MDATA53	MDATA52	MDATA51	MDATA50

C0MDB6m

7	6	5	4	3	2	1	0
MDATA67	MDATA66	MDATA65	MDATA64	MDATA63	MDATA62	MDATA61	MDATA60

C0MDB7m

7	6	5	4	3	2	1	0
MDATA77	MDATA76	MDATA75	MDATA74	MDATA73	MDATA72	MDATA71	MDATA70

Note m = 0 to 15

Figure 21-46. CAN message data byte register format (C0MDBxm) (x = 0 to 7), (C0MDBzm) (z = 01, 23, 45, 67) (2/2)

- C0MDBzm register

C0MDB01m

15	14	13	12	11	10	9	8
MDATA 0115	MDATA 0114	MDATA 0113	MDATA 0112	MDATA 0111	MDATA 0110	MDATA 019	MDATA 018
7	6	5	4	3	2	1	0
MDATA017	MDATA016	MDATA015	MDATA014	MDATA013	MDATA012	MDATA011	MDATA010

C0MDB23m

15	14	13	12	11	10	9	8
MDATA 2315	MDATA 2314	MDATA 2313	MDATA 2312	MDATA 2311	MDATA 2310	MDATA 239	MDATA 238
7	6	5	4	3	2	1	0
MDATA237	MDATA236	MDATA235	MDATA234	MDATA233	MDATA232	MDATA231	MDATA230

C0MDB45m

15	14	13	12	11	10	9	8
MDATA 4515	MDATA 4514	MDATA 4513	MDATA 4512	MDATA 4511	MDATA 4510	MDATA 459	MDATA 458
7	6	5	4	3	2	1	0
MDATA457	MDATA456	MDATA455	MDATA454	MDATA453	MDATA452	MDATA451	MDATA450

C0MDB67m

15	14	13	12	11	10	9	8
MDATA 6715	MDATA 6714	MDATA 6713	MDATA 6712	MDATA 6711	MDATA 6710	MDATA 679	MDATA 678
7	6	5	4	3	2	1	0
MDATA677	MDATA676	MDATA675	MDATA674	MDATA673	MDATA672	MDATA671	MDATA670

Note m = 0 to 15

(21) CAN message data length register m (C0MDLCm)

The C0MDLCm register is used to set the number of bytes of the data segment of the message buffer.

Figure 21-47. AN Message Data Length Register m Format (C0MDLCm)

Address: Reference Table 21-16

Reset value: 0000xxxxB

R/W

	7	6	5	4	3	2	1	0
C0MDLCm	0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0

MDLC3	MDLC2	MDLC1	MDLC0	Data length for sending/receiving messages
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
1	0	0	1	Set Off (If these bits are set during transmission, 8 bytes of data are transmitted regardless of the DLC value set when the data frame is transmitted. However, the DLC actually transferred to the CAN bus is set to the DLC value of this register)[1]
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

1 The actual data and DLC values transferred to the CAN bus are as follows.

Type of frame sent	Length of data sent	DLC Transmission
data frame	The number of bytes that were specified by the DLC (8 bytes if the	MDLC [3:0]
remote frame	0 bytes	

Note: 1. Be sure to set bit 7 to 40000B.

2. The received data is stored in the C0MDBxm based on the number of bytes corresponding to the DLC of the received frame (however, the upper limit is 8). C0MDBxm is not defined without data storage.

Note m = 0 to 15

(22) CAN Message Configuration Register (C0MCONFm)

The C0MCONFm register specifies the type of message buffer and sets the mask.

Figure 21-48. CAN Message Configuration Register Format (C0MCONFm) (1/2)

Address: Reference Table 21-16

Reset Value: No Definition

R/W

	7	6	5	4	3	2	1	0
C0MCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0

OWS	overlay control bit
0	The message buffer of received data frame [1] is not overwritten by the newly received data frame. The newly received data frame is discarded.
1	The message buffer that has received Data Frame [1] is overwritten by the newly received data frame.

1 "Message buffer received from data frame" is the receive message buffer with its DN bit set to 1

Note Receives and stores remote frames regardless of the OWS and DN bit settings. Remote frames that meet other criteria (ID match, RTR=0, TRQ=0) are always received and stored in the appropriate message buffer (generate an interrupt, set DN flag, MDLC [3:0] bit update, and logged into the Receive history list)

RTR	Remote Frame Request Bit [1]
0	Send a data frame.
1	Send a remote frame

The 1 RTR bit specifies the type of message frame transmitted from the message buffer defined as the transmission message buffer. The RTR of the received frame's transmission message buffer is 0 even if a valid remote frame has been received. Even if a remote frame matching an ID is received from the CAN bus, the RTR bit of the transmission message buffer has been set to 1 so that the remote frame can be transmitted, and the remote frame may not be received or stored (interrupt generated, DN flag set, MDLC (3:0) bit updated).

MT2	MT1	MT0	Message cache type setting bit
0	0	0	Send message cache
0	0	1	Receive message cache (no masking settings)
0	1	0	Receive message cache (1 masking setting)
0	1	1	Receive message cache (2 masking settings)
1	0	0	Receive message cache (3 masking settings)
1	0	1	Receive message cache (4 masking settings)
Other Values			Disable from setting

Note m = 0 to 15

Figure 21-48. CAN Message Configuration Register Format (C0MCONFm) (2/2)

MA0	message cache allocation bit
0	Message cache not used
1	Using message caching

Note: Make sure you write 0 for bits 1 and 2.

Note m = 0 to 15

(23) CAN message ID register m (C0MIDLm and C0MIDHm)

C0MIDLm and C0MIDHm registers are used to set identifiers (IDs)

Figure 21-49. CAN message ID register m format (C0MIDLm and C0MIDHm)

Address: Reference Table 21-16
Reset Value: No Definition
R/W

C0MIDLm

15	14	13	12	11	10	9	8
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
7	6	5	4	3	2	1	0
ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

C0MIDHm

15	14	13	12	11	10	9	8
IDE	0	0	ID28	ID27	ID26	ID25	ID24
7	6	5	4	3	2	1	0
ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16

DE	Format Mode Specified Bit
0	Standard format mode (ID18 to ID28: 11-bit) [1]
1	Extended format mode (ID0 to ID28: 29-bit)

1 ID0 to ID17 bit not used.

ID0 to ID28	Message ID
ID18 to ID28	11-bit standard ID value (when IDE =0)
ID0 to ID28	29-bit extended ID value (when IDE =1)

Note 1. Be sure to write the bits 13 and 14 of the C0MIDHm register 0.

2. Make sure that the ID value is aligned to this register according to the given location. Note that for standard IDs, the ID value must be moved into the ID11 to the ID28 location.

Note m = 0 to 15

(24) CAN message control register m (C0MCTRLm)

The C0MCTRLm register is used to control the operation of the message buffer.

Figure 21-50. CAN Message Control Register m format (C0MCTRLm) (1/3)

Address: Reference table 21-16.

Reset value: 000xx000B

R/W

(a) read

C0MCTRLm	15	14	13	12	11	10	9	8
	0	0	MUC	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	MOW	IE	DN	TRQ	RDY

(b) write

C0MCTRLm	15	14	13	12	11	10	9	8
	0	0	0	0	Set IE	0	Set TRQ	Set RDY
	7	6	5	4	3	2	1	0
	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY

(a) read

MUC[Note]	message cache data update bit
0	CAN module did not update message buffer (receive and store)
1	The CAN module is updating the message buffer (receiving and storing).

Note The MUC bit is not defined before the first receive and store

MOW	Message Cache Overlay Status Bit
0	Newly received data frames do not overwrite message buffers.
1	Newly received data frame overlay message buffer

Note Even if the remote frame is received and stored in the transmission message cache, the DN is equal to 1 and the MOW bit is not set to 1. .

BE	Message cache interrupt request enable bit
0	Receive message buffer: Interrupt for valid message receipt completion is disabled. Transmission message buffer: Interrupt for normal message transfer completion
1	Receive message buffer: Valid message receipt completes interrupt activation. Transmission message buffer: Normal message transfer complete interrupt activation.

Note m = 0 to 15

Figure 21-50. CAN Message Control Register m format (C0MCTRLm) (2/3)

DN	message cache data update bit
0	Data or remote frames are not stored in the message buffer.
1	Data frames or remote frames are stored in the message buffer.

TRQ	Message cache send request bits
0	There are no pending or transmitting message frame transfer requests in the message buffer.
1	The message buffer maintains the transmission of the message frame pending or in the process of transmitting the message frame. .

Note: Do not set both the TRQ bit and the RDY bit (1). Set the RDY bit (1) before setting the TRQ bit.

RDY	Message Cache Ready Bits
0	Message buffers can be written by software. CAN module failed to write to message buffer
1	Ignore message buffers written through software (except write access to RDY, TRQ, DN, and MOW bits). The CAN module can write the message buffer. .

Note:1. Do not clear the RDY bits (0) during the message transfer process.

Clear the RDY bit (0) following the Transfer Abort procedure to redefine the message buffer. .

2. When the RDY bit is not cleared (although it has been cleared), clear it again.
3. Before writing to the message buffer register, ensure that the RDY bits are cleared. This acknowledgement is performed by reading the RDY bits. However, you do not need to confirm setting the TRQ bit, clearing the DN bit, setting the RDY bit, or clearing the MOW bit of the C0MCTRLm register. .

(b) write

ClearMOW	Setting MOW Bits
0	The MOW bit did not change.
1	The MOW bit is cleared 0.

SetIE	ClearIE	Set IE bits
0	1	IE cleared 0.
1	0	IE is 1.
Other Values		IE position unchanged.

Note: Always set IE and RDY bits individually.

ClearDN	Set DN Bit
0	DN does not change.
1	DN cleared 0.

Note: Do not set the DN bit to 1 with software. Make sure you write 0 for 10.

Note m = 0 to 15

Figure 21-50. CAN Message Control Register m format (C0MCTRLm) (3/3)

SetTRQ	ClearTRQ	TRQ set bit
0	1	TRQ clear 0.
1	0	TRQ Set 1.
Other Values		No change in TRQ

Note: When a message is received from another node or a transmission cancelation message, the transmission may not start immediately even if the TRQ bit is set to 1.
Even if the TRQ bit is cleared to 0, the transfer does not abort. If messages are being transmitted, the transmission continues until the transmission is complete (successful or unsuccessful).

SetRDY	ClearRDY	RDY Set Bit
0	1	RDY clear 0.
1	0	RDY 1.
Other Values		RDY doesn't change.

Note that the IE bit and the RDY bit are set separately.

Note m = 0 to 15

(25) Serial Communication Pin Selection Register 1 (PIOR3)

The PIOR3 register is used to switch the input source to the timer array unit and the CAN communication pin. This register can be read or written in 8-bit format.

The PIOR3 register can be used to select CTxD, CRxD pins on two different ports.

Figure 21-51. Serial Communication Pin Select Register 1 Format (PIOR3)

Address: 0x4004087C

Reset value: 00H

R/W

	7	6	5	4	3	2	1	0
PIOR3	0	0	0	0	PIOR33	PIOR32	PIOR31	PIOR30

PIOR33	Communication Pin Selection FCAN1	
	CTxD0	CRxD0
0	P02	P03
1	P51	P50

(26) Port Mode Registers 0, 5 (PM0, PM5)

The PM0 and PM5 registers are used to set ports 0 and 5 as input or output.

When using the P02/CTxD0 or P51/CTxD0 pin for serial data output, clear the PM02 or PM51 bit "0" and set the P02 or P51 output latch to "1".

Set the PM03 or PM50 bits to "1" when serial data entry is made using P03/CRxD0 or P50/CRxD0 pins. At this point, the output latch for P03 or P50 may be "0" or "1".

You can set PM0, PM5 registers using the 8-bit memory action instruction. Setting these registers to the FFH generates a reset signal. .

Note: Serial data output and input are selected at either port of each channel using the appropriate registers.

Note: Pin positioning is product specific, refer to 1.3 pin configuration and 2.1 pin feature list.

Address: 0x40040320

Reset value: FFH

R/W

	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00

Address: 0x40040325

Reset value: FFH

R/W

	7	6	5	4	3	2	1	0
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50

PMmn	PMmn pin I/O pin mode selection (m = 0, 5; n = 0 to 7)
0	Output Mode (Output Cache On)
1	Input Mode (Output Cache Off)

21.8 CAN controller initialization

21.8.1 CAN module initialization

Before enabling the CAN module for operation, the system clock of the CAN module needs to be determined by setting the CCP[3:0] bit of the C0GMCS register by software. Do not change the setting of the CAN module system clock after the CAN module is working.

Enable the CAN module by setting the GOM bit of the C0GMCTRL register.
For procedures on initializing the CAN module, see Operation of the 21.16 CAN Controller.

21.8.2 Initialization of message cache

When the CAN module is enabled, the message buffer contains undefined values. Before switching the CAN module from initialization mode to one of the operation modes, minimum initialization is required for all message buffers, even if the application is not used.

- Clear RDY, TRQ, and DN bits of C0MCTRLm (0)
- Clear C0MCONFm MA0 bits (0)

Note m = 0 to 15

21.8.3 Redefining message caching

Redefining the message buffer means changing the ID and control information of the message buffer when receiving or transmitting messages without affecting other send/receive operations.

(1) Redefining message caching in initialization mode

Place the CAN module in initialization mode once and change the ID and control information of the message buffer in initialization mode. After you change the ID and control information, set the CAN module to the operation mode.

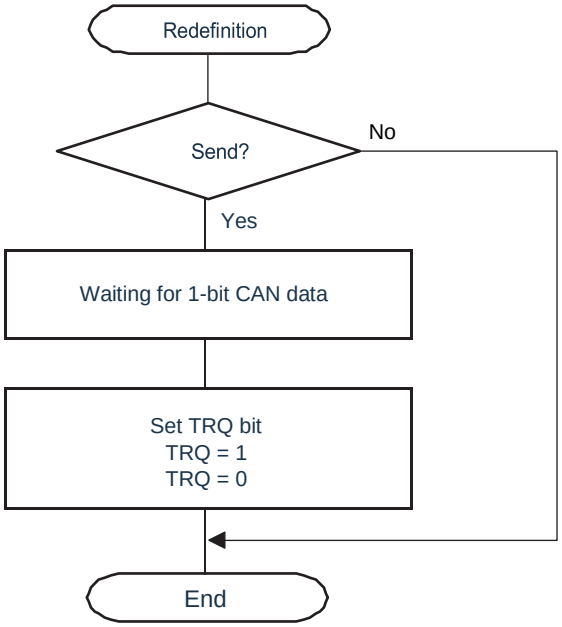
(2) Redefining message caching in receiving

The redefining operation is shown in Figure 21-66.

(3) Redefining message caching in sending

To override the contents of the transmission message buffer that has been set for a transmission request, perform a transmission abort process (see 21.10.4(1) Transmission abort process outside of Automatic Block Transfer (ABT) in normal mode of operation and 21.10.4(2) ABT transfer in normal mode of operation with Transmission abort outside of Automatic Block Transfer). Confirm that the transfer has been aborted or completed, and redefine the message buffer. After redefining the transmission message buffer, set up the transmission request using the procedure described below. However, a 1-bit latency is not required when the transmission request is set to a redefined message buffer without aborting an ongoing transmission.

Figure 21-53. Setting Send Request (TRQ) to Send Message Cache after Redefinition



Note 1. When receiving a message, receive filtering is performed according to the ID and mask set as the buffer for each received message. If that proces in figure 21-66 is not followed, the content of the message buffer after the redefinition may conflict with the receive result (the result of the receive filtering). If this occurs, check that the IDs and IDEs that were first received and stored in the message buffer after the redefinition are the IDs and IDEs that were stored after the message buffer was redefined. If IDs and IDEs are not stored after the redefinition, redefine the message buffer.

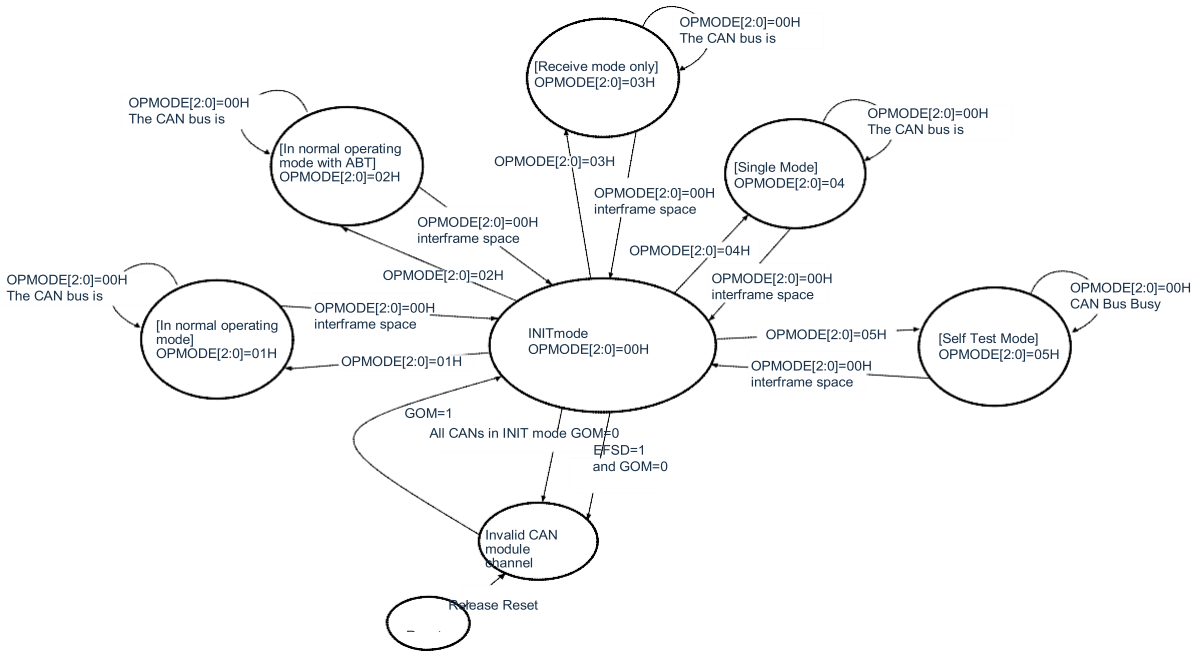
2. When sending messages, the transmission priority is checked against the ID, IDE, and RTR bits set to each of the sending message buffers. Select the transmission message buffer with the highest priority for transmission. If the procedure in FIGS. 21-53 is not observed, messages without the highest priority of ID may be transmitted after redefinition.

21.8.4 Transition from initialization to operation

The CAN module can switch to the following operation mode.

- normal operation mode
- Normal operation mode with ABT
- Receive mode only
- single-shot mode
- self-test mode

Figure 21-54. Transition to Operation mode



The transition from the initialization mode to the operation mode is controlled by the bit string OPMODE[2:0] in the C0CTRL register.

Changing from one operation mode to another requires a transition from both to the initialization mode. Do not directly change one mode of operation to another mode of operation, otherwise there is no guarantee of operation.

When the CAN bus is not in inter-frame space (i.e., frame receiving or sending is in progress) and the CAN module changes the value of the inter-frame space (OPMODE[2:0] to 00H), the transition from the operational mode to the initialization mode is pending. After issuing a request to change the mode to the initialization mode, the OPMODE [2:0] bit is read until its value changes to 000B to confirm that the module has entered the initialization mode (see Figure 21-64).

21.8.5 Reset CAN module error counter C0ERC

If you need to reset the CAN module error counter C0ERC and the CAN module information register C0INFO, set the CCERC bit of the C0CTRL register to 1 in initialization mode when reinitializing from the bus-off state or forcing recovery. When this bit is set to 1, the CAN module error counter C0ERC and the CAN module information register C0INFO are cleared to their default values.

21.9 message receiving

21.9.1 message receiving

Under all operation modes, complete message buffer areas are analyzed to find the appropriate buffer to store the newly received messages. All message buffers that meet the following criteria are included in this assessment (RX search process):

- Used message cache
  (C0MCONFm MA0 bit is 1)
- Cache set to receive messages
  (C0MCONFm's MT[2:0] bit is set to 001B,010B,011B,100B,or101B.)
- Ready to receive
  (C0MCTRLm RDY bit set to 1)

When there are two or more message caches to receive a message, the message is stored in a higher priority cache. For example, when an unmaskable receive message buffer and a received message buffer linked to mask 1 have the same ID, the received message is not stored in the message buffer linked to mask 1, even if the message buffer has not received the message and the message is already in the unmaskable receive message buffer. In other words, when set to store messages in two or more message buffers with different priorities, the message buffer with the highest priority always stores the messages; Messages are not stored in lower priority message buffers. This also applies when the message buffer with the highest priority fails to receive and store messages (i.e. DN=1 indicates that the message was received, but rewriting is disabled because OWS = 0). In this case, messages are not actually received and stored in the candidate message buffer with the highest priority, but are not stored in the lower priority message buffer.

priority	Storage with the same ID set	
1 (High)	Message cache without masking	DN =0
		DN = 1 and OWS = 1
2	Message cache linked to Mask 1	DN =0
		DN = 1 and OWS = 1
3	Message cache linked to Mask 2	DN =0
		DN = 1 and OWS = 1
4	Message cache linked to Mask 3	DN =0
		DN = 1 and OWS = 1
5 (Low)	Message cache linked to Mask 4	DN =0
		DN = 1 and OWS = 1

Note m = 0 to 15

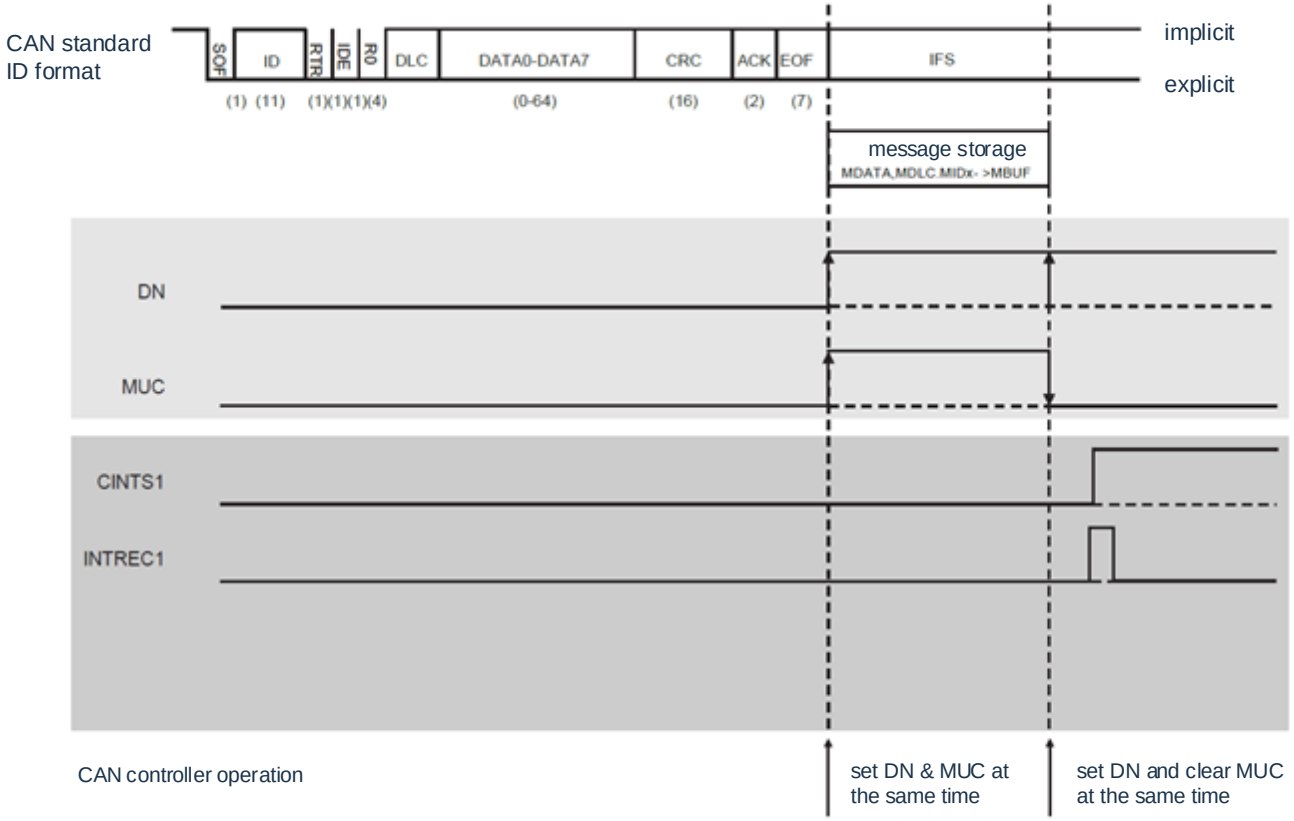
21.9.2 read received data

To maintain data consistency when reading the CAN message buffer, perform data reading from FIGS. 21-76 to 21-78.

During message reception, the CAN module sets the DN of the C0MCTRLm register twice: At the beginning of storing data into the message buffer, again at the end of this stored procedure. The MUC bit of the message buffer C0MCTRLm register is set during this save. (see Figure 21-55).

The receive history list is also updated before the stored procedure. In addition, during the stored procedure (MUC_1), the RDY bit of the C0MCTRLm register of the message buffer is locked to avoid a coincidental data WR by the CPU. Note that when the CPU accesses the message buffer, the stored process may be disturbed (delayed).

Figure 21-55. DN and MUC Bit Set-up Cycles (for Standard ID Format)



Note m = 0 to 15

21.9.3 Receive history list feature

The Receive History List (RHL) function in the Receive History List records the number of receive message buffers that receive and store each data frame or remote frame. The RHL is consist of storage elements corresponding to at most 23 messages, that last message entry point (LIPT) has a corresponding C0LIPT register and the receive history list acquisition point (RGPT) has a corresponding C0RGPT register.

The RHL is in an undefined state immediately after the CAN module transitions from an initialization mode to one of the operation modes.

The C0LIPT register contains the contents of the RHL element indicated by the value of LIPT pointer minus 1. Therefore, by reading the C0LIPT register, it is possible to check the number of message buffers that first receive and store data frames or remote frames. The LIPT pointer acts as a write pointer, indicating which portion of the number of RHLa message buffers is logged. Each time a data frame or remote frame is received and stored, the corresponding message buffer number is recorded to the RHL element indicated by the LIPT pointer. The LIPT pointer is automatically incremented each time you record to RHL. Thus, the number of message buffers that receive and store frames is recorded in chronological order.

The RGPT pointer is used as a read pointer to read the message buffer number of the record from the RHL. This pointer indicates the first RHL element that the CPU has not read. By reading the C0RGPT register by software, you can read the number of message buffers that have been received and stored for data or remote frames. The RGPT pointer is automatically incremented each time the message buffer number is read from the C0RGPT register.

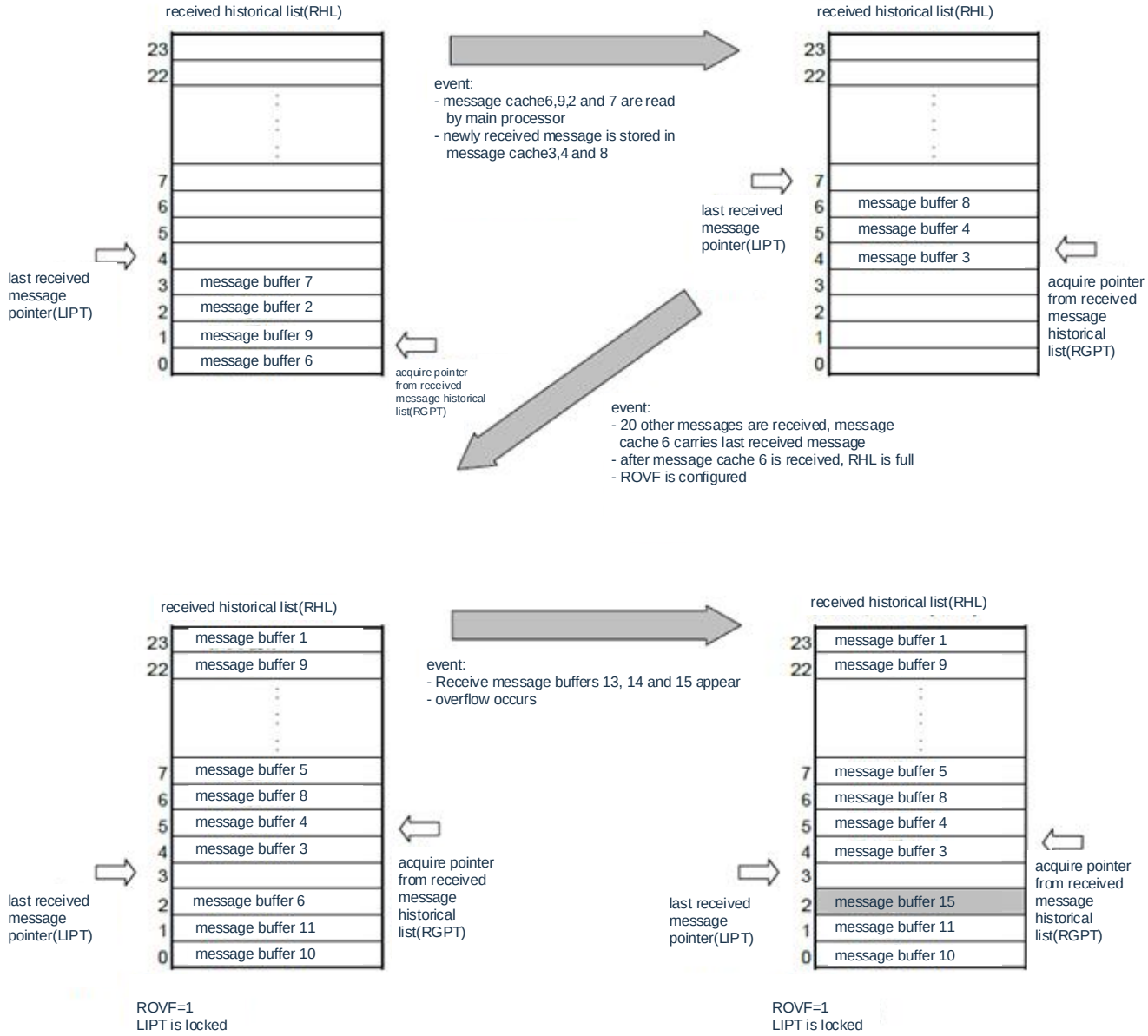
If the value of the RGPT pointer matches the value of the LIPT pointer, the RHPM bit of the C0RGPT register (Receive history list pointer match) is set to 1. This indicates that the message buffer number not read in RHL is not reserved. If a new message buffer number is logged, the LIPT pointer is incremented and the RHPM bit is cleared because its value no longer matches the value of the RGPT pointer. In other words, the number of unread message buffers is present in the RHL.

If the LIPT pointer increments and matches the value of the RGPT pointer minus 1, the ROVF bit of the C0RGPT register (Receive History List Overflow) is set to 1. This indicates that the RHL is full of unread message buffers. When further message reception and storage occur, the message buffer number of the last record is overwritten with the number of message buffers that the new message was received and stored. In this case, the message buffer number recorded in the RHL is not fully chronological after the ROVF bit (1) is set. However, the message itself is not lost and can be located in the message buffer memory with the help of DN bits by CPU search.

Note: If the history list is under an overflow condition (ROVF has been set), it is still possible to read the history list contents until the history list is empty (indicated by the RHPM flag). However, the history list is still overflowing until the ROVF is cleared through the software. If the ROVF is not cleared, the RHPM flag is not updated (cleared) on the message of the new received frame. This can lead to a situation where RHPM represents an empty history list, although receipts have occurred and the history list is overrun (ROVF and RHPM have been set).

The order of occurrence is maintained as long as the RHL contains 23 or fewer entries. If more receive occurs when that host processor doe not read the RHL, the complete receive sequence cannot be recovered.

Figure 21-56. Receiving history list



ROVF=1 means that LIPT equals RGPT-1, and the message buffer number is stored in the element indicated by LIPT-1.

21.9.4 mask function

For any message buffer used for receiving, you can select the receive mask (or no mask) assigned to the quarter.

By masking, message ID comparisons can be reduced by masking bits, allowing multiple different IDs to be received in a buffer.

When the masking function takes effect, the identifier bit with the mask definition "1" in the received message is not compared to the corresponding identifier bit in the message buffer.

However, this comparison is performed in bits that are defined by the mask as 0.

For example, all messages with a standard format ID (where ID25 to ID27 bits are "0" and ID22 and ID2 4 bits are "1") will be stored in the message buffer 14. The procedure for this example is as follows.

<1> identifier stored in message cache

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

x = Never mind

<2> identifier configured in message cache 14 (example)

(Use CAN 0 message ID registers L14 and H14 (C0MIDL14 and C0MIDH14))

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x
ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
x	x	x	x	x	x	x	x	x	x	x
ID6	ID5	ID4	ID3	ID2	ID1	ID0				
x	x	x	x	x	x	x				

ID25 to ID27 ID cleared to "0", ID24 and ID22 set to "1", registered (initialized) to message buffer 14.

Remark The message buffer 14 is set to a standard format identifier linked to mask 1 (MT[2:0 of the C0MCONF14 register 010B).

<3> CAN Module 1 Mask Settings (mask1) (example)

(Use the CAN0 module mask 1 registers L and H (C0MASK1L and C0MASK1H))

CMID28	CMID27	CMID26	CMID25	CMID24	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18
1	0	0	0	0	1	0	1	1	1	1

CMID17	CMID16	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8	CMID7
1	1	1	1	1	1	1	1	1	1	1

CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
1	1	1	1	1	1	1

1: No comparison

(masked) 0: comparison

The CMID24 to CMID27 and CMID22 bits are cleared to "0" and the CMID0 to CMID21, CMID23, and CMID28 bits are set to "1".

21.9.5 multi-buffer receive block function

The multiple buffer receive block (MBRB) function is used to store data blocks sequentially in two or more message buffers without CPU interaction by setting the same ID to two or more message buffers having the same message buffer type.

For example, suppose that the same message buffer type is set to 5 message buffers (message buffers 10 to 14) and give them the same ID. If the first message ID received is the same as the buffer ID, the message is stored in the message buffer 10. At this point, the DN bit of the message buffer 10 is set, preventing the message buffer from being overwritten when subsequent messages are received.

If a next message with a match ID is received, the message is received and stored in the message buffer 11. Each time a message with a matching ID is received, it is stored in message buffers 12, 13, and 14 in order (ascending order). Even when a block of multiple messages is received, the messages are stored and received after scanning, without overwriting previously received matching ID data.

It is possible to check whether a data block is received and stored by setting the IE bit of the C0MCTRLm register for each message buffer, e.g. if the data block is composed of k messages, the k message buffer will initialize to receive the data block. The IE bits in the message buffer 0 through (k-2) are cleared to 0 (interrupt disabled) and the IE bits in the message buffer k-1 are set to 1 (interrupt enabled). In this case, a receipt completion interrupt occurs when the message is received and stored in the message buffer k-1 indicating that the MBRB is full. Alternatively, you can issue a warning that MBRB is about to overflow by clearing the IE bits of the message buffer 0 through (k-3) and setting the IE bits of the message buffer k-2.

The basic condition for storing the received data in each message buffer of the MBRB is the same as for storing the data in a single message buffer.

Note 1. You can configure MBRB for each message buffer of the same type. Therefore, even if there is a vacancy in the message buffer of another MBRB with a matching ID but different message buffer type, the received message is not stored in the message buffer, but is discarded.

2. MBRB does not have a ring cache structure. Therefore, when a message is stored in the message buffer with the highest number in the MBRB configuration, the newly received message will not be stored in the message buffer with the lowest message cache number in the message buffer.
3. MBRB runs according to receiving and storage conditions; There are no settings specific to MBRB, such as function-enabled bits. MBRB is automatically configured by setting the same message buffer type and ID to two or more message buffers. .
4. Using MBRB, Match ID indicates Match ID after mask. Even if the IDs set to each message buffer are different, if the ID masked by the mask register matches, it is considered a matching ID, and a buffer with this ID is considered a storage target for the message. .
5. Please refer to 21.9.1 message receiving for the priority of MBRBs.

Note m = 0 to 15

21.9.6 remote frame receiving

In all operation modes, when a remote frame is received, a message buffer that can store the remote frame is searched from all message buffers that meet the following conditions.

- For message caching
  (MA0 bit of C0MCONFm register 1)
- Set to transmit message cache
  (MT [2:0] bit of the C0MCONFm register is set to 000B)
- Ready to receive
  (The RDY bit of the C0 MCTRLm register is set to 1.)
- Set as Transmission Message
  (RTR bit clear 0 for C0MCONFm register)
- Transmission request is not set.
  (TRQ bit of C0MCTRLm register cleared 0)

When a remote frame is accepted, if the ID of the received remote frame matches the ID of the message buffer that meets the above criteria, do the following:

- The MDLC [3:0] bit of the C0MDLCm register stores the received DLC value
- The C0MDATA0m to C0MDATA7m of the data interval is not updated (saving the data before receiving).
- The DN bit of the C0MCTRLm register is set to 1.
- The CINTS1 bit of the C0INTS register is set to 1 if the IE bit in the received and stored message buffer register C0MCTRLm is set to 1.
- Output Receive Complete Interrupt (C0MCTRLm) if the IE bit is set to 1 in the message buffer register CIE1 for the received and stored frame and the C0IE register is set to 1.
- The message buffer number is recorded in the receipt history list.

Note that the coverage control of the OWS bit of the message buffer C0MCONFm register and the DN bit of the C0MCTRLm register is not affected when the message buffer is searched for to receive and store remote frames. The OWS settings are ignored and the DN is set in any case. If multiple transmission message buffers have the same ID and the received remote frame ID matches the ID, the remote frame is stored in the buffer with the lowest number of message buffers.

Notes m = 0 to 15

21.10 message sending

21.10.1 message sending

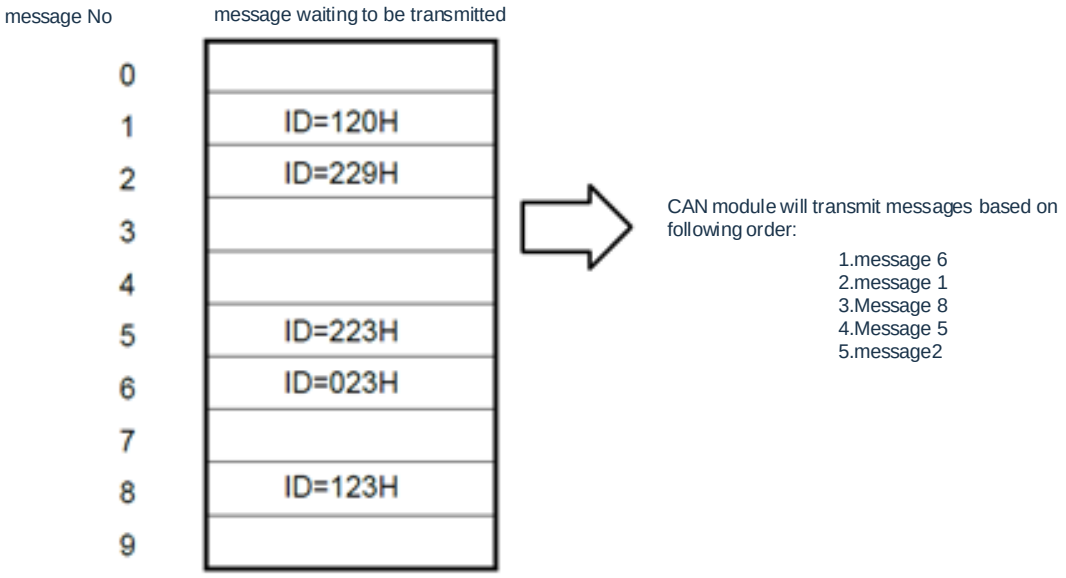
In all operation modes, if the TRQ bit is set to 1 in the message buffer that satisfies the following conditions, the message buffer for the message to be transmitted is searched.

- Used for message caching
  (MA0 Position 1 for C0MCONFm Register)
- Set to send message cache
  (The MT [2:0] bit of the C0MCONFm register is set to 000B.)
- Ready to Send
  (RDY position 1 for C0MCTRLm register.)

CAN system is a multi-master communication system. In such a system, the priority of the message transmission is determined according to the message identifier (ID). In order to achieve software transmission processing when there are multiple messages waiting for transmission, the CAN module uses hardware to detect the message ID with the highest priority and automatically identifies the message, which eliminates the need for software priority control.

The sending priority is controlled by an identifier (ID).

Figure 21-57. Examples of message processing



After a transmission message search, a transmission message having the highest priority of the transmission message buffer of a pending transmission request (a message buffer with the TRQ bit set to 1) is transmitted.

If a new send request is set, the send message buffer with the new send request is compared to the send message buffer with the pending send request. If a new send request has a higher priority, the request is transmitted unless the transmission of lower priority messages has started. However, if the transmission of lower priority messages has started, a new send request will be sent later. To address this priority reversal effect, the software can perform a transmission abort request on a low priority message. The highest priority is determined according to the following rules.

priority	Condition	Description
1 (H)	Value of the first 11 bits of ID [ID18 to ID28]:	First send the message frame of the lowest value indicated by the first 11 bits of ID. If the value of the 11-bit standard ID is equal to or less than the first 11 bits of the 29-bit extension ID, the 11-bit standard ID takes precedence over the message frame with the 29-bit extension ID.
2	Frame Type	Data frames with 11-bit standard IDs (RTR bit cleared to 0) have higher priority than remote frames with standard IDs and message frames with extended IDs.
3	ID Type	Message frames with standard ID (ID bit cleared to 0) have higher priority than message frames with extended ID
4	ID 18-bit low value [ID0 to ID17]:	If a plurality of transmit pending extended ID message frames have the same value and the same frame type (equal to the RTR bit value) in the first 11 bits of the ID, the message frame with the lowest median value of 18 bits below the extended ID is transmitted first.
5 (Low)	message cache number	If two or more message buffers request to send a message frame with the same ID, the message from the lowest message buffer number in the message buffer is transmitted first.

Note 1. If the automatic block transfer request bit ABTTRG is set to 1 in normal mode with ABT, the TRQ bit of only one message cache in the ABT message cache group is set to 1. If the ABT mode is triggered by a ABTTRG bit, one of the TRQ bits in the ABT region is set to 1 (buffers 0 through 7). In addition to this TRQ bit, applications can request other TX message buffer transfers that are not part of the ABT zone (set TRQ to 1). In this case, the Interval Arbitrating Process (TX-search) evaluates all TX message buffers with the TRQ bit set to 1 and selects the message buffer with the highest priority identifier for the next transmission. If two or more identifiers have the highest priority (i.e., the same identifier), the message at the lowest message buffer number is transmitted first.

After the message frame is successfully transmitted, the following actions are performed:

- The TRQ flag for the appropriate send message buffer is automatically cleared to 0.
- The C0INTS register has the send completion status bit CINTS0 set to 1 (if the interrupt enable bit (IE) of the corresponding send message buffer is set to 1).
- Interrupt request signal INTC0TRX output (if the CIE0 bit of the C0IE register is set to 1 and the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1.

2. When you change the contents of the send buffer, you must clear the RDY flag for the buffer before updating the contents. As with internal transmission operations, the RDY flag may be temporarily locked and the status of the RDY must be checked by software after the change.
3. m = 0 to 15

21.10.2 Send History List Feature

The Transmission History List (THL) feature records the number of the message buffer in the Transmission History List where the data or remote frame is sent. The THL contains storage elements corresponding to up to seven messages, a last outgoing message pointer (LOPT) with corresponding C0LOPT registers, and a transmit history list acquisition pointer (TGPT) of the C0TGPT register.

When the CAN module transitions from one of the initialization modes to one of the operation modes, the THL is immediately undefined.

The C0LOPT register contains the contents of the THL element indicated by the value of LOPT pointer minus 1. Therefore, by reading the C0LOPT register, you can check the number of message buffers that first transmit a data frame or a remote frame. The LOPT pointer is used as a write pointer to indicate which part of the THL that records the message buffer number is recorded. The corresponding message buffer number is recorded to the THL element indicated by the LOPT pointer whenever a data frame or remote frame is transmitted. The LOPT pointer is automatically incremented each time the THL is logged. In this way, the message buffer numbers of the frames received and stored are recorded in chronological order.

The TGPT pointer is used as a read pointer to read the message buffer number of the record from the THL. This pointer indicates the first THL element that the CPU has not read. The software reads the C0TGPT register to read the number of message buffers that have completed the transfer. The TGPT pointer is automatically incremented each time the message buffer number is read from the C0TGPT register. .

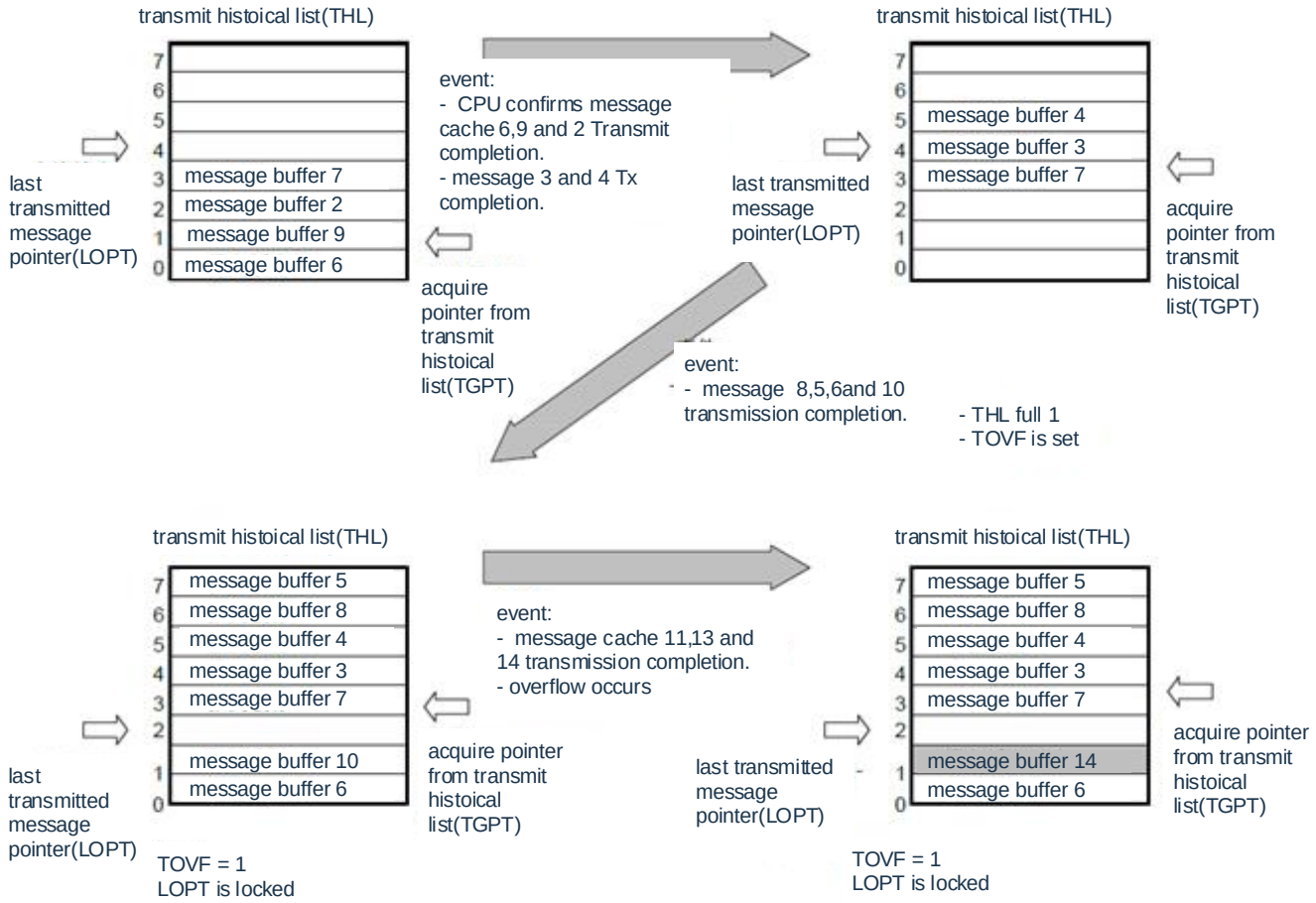
If the value of the TGPT pointer matches the value of the LOPT pointer, the THPM bit (Transmission History List Pointer Match) of the C0TGPT register is set to 1. This indicates that the unread message buffer number is not preserved in THL. If a new message buffer number is logged, the LOPT pointer is incremented and the THPM bit is cleared because its value no longer matches the value of the TGPT pointer. In other words, the number of unread message buffers exists in THL.

If the LOPT pointer increments and matches the value of the TGPT pointer minus 1, the TOVF bit of the C0TGPT register (Transmission History List Overflow) is set to 1. This means that the THL is full of unread message buffer numbers. If a new message is received and stored, the last recorded message buffer number is overwritten with the number of message buffers of the subsequent message transmission. When the TOVF bit is set to 1, the message buffer number recorded in THL does not fully reflect the chronological order. However, other transmission messages can be found by a CPU search applied to all transmission message buffers unless the CPU has not overridden the transmission object in one of the previous buffers. A total of up to six transmission-complete scans occur without THL overflows.

Note: If the history list is under an overflow condition (TOVF is set), the history list content can still be read until the history list is empty (indicated by the THPM flag). However, the history list remains overflowed until the TOVF is cleared by the software. If TOVF is not cleared, the THPM flag is not updated (cleared) after the new message is successfully transferred. This may cause this situation, where THPM indicates an empty history list, although it has been successfully transmitted, and the history list is overflowed (TOVF and THPM are set).

Remark m = 0 to 15

Figure 21-58. Sending a history list



TOVF=1 means that LOPT equals TGPT-1, and the message buffer number is stored in the element indicated by LOPT-1.

21.10.3 Automatic Block Transfer (ABT)

The Automatic Block Transfer (ABT) feature is used to successfully continuously transfer two or more data frames without CPU interaction. The maximum number of transmitted message buffers allocated to the ABT function is 8 (message buffer numbers 0 to 7).

By setting the OPMODE[2:0] bit of the C0CTRL register to 010B, the Normal Operation Mode with Automatic Block Transfer (referred to here as the ABT mode) can be selected.

To issue an ABT transfer request, first define the message buffer through the software. Set the MA0 bit (1) in all message buffers for ABT and define all buffers as transmission message buffers by setting the MT[2:0] bit to 000B. Ensure that ABT sets the ID for each message buffer, even if the same ID is used for all message buffers. To use two or more IDs, use the C0MIDLm and C0MIDHm registers to set the ID of each message buffer. Set C0MDLCm and C0MDATA0m to C0MDATA7m before sending a transfer request for the ABT feature.

After completing the initialization of the message buffer of ABT, the RDY bit needs to be set to 1. In ABT mode, the TRQ bits are not operated by the software.

After you have prepared the data for the ABT message buffer, set the ABTTRG bit to 1. Automatic block transfer is then initiated. When you start ABT, the TRQ bit in the first message buffer (message buffer 0) is automatically set to 1. When the data transmission of the message buffer 0 is completed, the TRQ bit of the next message buffer is automatically set, and the message buffer 1. In this way, the transmission will occur in turn.

The program can insert a delay time during a time interval during which a Transfer Request (TRQ) is automatically set when a continuous transmission is performed. The delay time to insert is defined by the C0GMABTD register. The unit of delay time is DBT (data bit time). DBT depends on the settings of the C0BRP and C0BTR registers. .

In transmission objects within an ABT area, the priority of the transmission ID is not evaluated. The data in the message buffers 0 through 7 is transmitted sequentially. When the transmission of the data frame from the message buffer 7 is completed, the ABTTRG bit is automatically cleared to 0 and the ABT operation is completed. .

If the RDY bit of the ABT message buffer is cleared during ABT, no data frames are transmitted from the buffer, the ABT stops, and the ABTTRG bit is cleared. The RDY and ABTTRG bits can then be set to 1 by software to restore transmission from the message buffer that the ABT stopped. In order not to resume transmission from the message buffer of the stopped ABT, the internal ABT engine can be reset by setting the ABTCLR bit to 1 when the ABT mode is stopped and the ABTRG bit is cleared to 0. In this case, if the ABTCLR bit is cleared to 0 and then the ABTTRG bit is set to 1, the transmission is initiated from the message buffer 0. .

Interrupts can be used to check whether a data frame is transmitted from all the message buffers of the ABT. To do this, it is necessary to clear the IE bit of the C0MCTRLm register for each message buffer other than the last message buffer to be 0.

If a transmission message buffer other than the ABT function (message buffer 8 to 15) is assigned to the transmission message buffer, the next message to be transmitted is determined by the priority of the transmission ID of the transmission ABT message buffer currently in a suspended state and is the transmission ID of the message buffer other than the buffer used by the ABT function.

Data frames transmitted from an ABT message buffer are not recorded in the Transmission History List (THL).

Note 1. Set the ABTCLR bit to 1 and clear the ABTTRG bit to 0 to resume ABT operations at 0 buffer count. If the ABTCLR bit is set to 1 and the ABTTRG bit is set to 1, subsequent actions are not guaranteed. .

2. If you clear the Automatic Block Transfer Engine by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared when the purge request is completed.
3. Do not set the ABTTRG bit in initialization mode. If you set the ABTTRG bit in initialization mode, the correct operation is not guaranteed after the mode changes from initialization mode to ABT mode.
4. Do not set the TRQ bit of the ABT message buffer to 1 by the software in the normal operation mode of ABT. Otherwise, operation cannot be guaranteed.
5. The C0GMABTD register is used to set a delay time from the completion of the previous ABT message to the setting of the TRQ bit of the next ABT message, and when the transmission request is set in the order of the ABT message numbers, the transmission is continuously carried out in the ABT mode. The time that messages are actually transmitted to the CAN bus varies depending on the status of transmission from other sites and the set-up status of the message transmission request other than the ABT message (message buffers 8 to 15).
6. If a transmission request is made to a message other than an ABT message, and no delay time is inserted in the automatically set ABT transmission request interval (C0GMABTD = 00H), the message other than an ABT message may not be dependent on the priority of the ABT message. .
7. Do not clear the RDY bit to 0 when ABTTRG =1.
8. If a message is received from another node while a normal mode of operation with an ABT is active, the TX message in the ABT region may be transmitted with a frame delay, although the C0GMABTD register is set to 00H.

Notes m = 0 to 15

21.10.4 Transmission abort handling

(1) Transfer abort process outside of automatic block transfer (ABT) in normal operation mode

The user can abort the transfer request by clearing the TRQ bit of the C0MCTRLm register to 0, and if the abort is successful, the TRQ bit is cleared immediately. The TSTAT bit and C0TGPT register of the C0CTRL register indicating the CAN bus transfer status can be used to check whether the transfer was successfully aborted (see the processing in FIGS. 21-72 for details).

(2) The ABT transmission is subject to transmission abort processing outside of automatic block transmission in normal operation mode

The user can clear the ABTTRG bit of the C0GMABT register to 0 to abort the transfer request. After checking the ABTTRG bit = 0 of the C0GMABT register, clear the TRQ bit of the C0MCTRLm register to 0. If the abort is successful, the TRQ bit is cleared immediately. You can use the TSTAT bit and C0TGPT register of the C0CTRL register that indicates the state of the transfer on the CAN bus to check whether the transfer was successfully aborted (for more information, see the processing in FIGS. 21-74).

(3) ABT Transmission Abort Handling in Normal Mode with Automatic Block Transmission

To abort a started ABT, clear the C0GMABT register with a ABTTRG bit of 0. In this case, if the ABT message is currently being transmitted and the transmission is complete (successful or not), the ABTTRG bit remains 1 and is cleared to 0 immediately after the transmission is complete. This will abort the ABT.

If the last transmission (before the ABT) is successful, the ABT's normal operation mode remains, with the internal ABT pointer pointing to the next message buffer to be transmitted. .

If the transmission is wrong, the location of the internal ABT pointer depends on the status of the TRQ bit in the message buffer of the last transmission. If the TRQ bit is set to 1 when the request to clear the ABTTRG bit, the internal ABT pointer points to the message buffer of the last transmission (see the process in Figure 21-73 for details). If the TRQ bit is cleared to 0 when the ABTTRG bit is cleared, the internal ABT pointer increments (+1) and points to the next message buffer in the ABT region (see the process in Figure 21-74 for more information).

Note: Make sure that ABT is aborted by clearing ABTRG to 0, which may not be possible if the request to abort the transfer is by clearing the RDY bit.

When the normal mode of operation of the ABT is restored after the ABT has been aborted and the ABTTRG bit is set to 1, you can determine the next ABT message buffer to transfer from the following table.

Status TRQ of ABT message cache	Abort after successful transfer	Abort after error transfer
Settings (1)	Next message cache [1] in the ABT interval	Message caching with the same ABT interval
Clear (0)	Next message cache [1] in the ABT interval	Next message cache [1] in the ABT interval

1: The above recovery operation can only be performed if there is a message buffer in the ABT region that is ready to use ABT. For example, a abort request issued when the ABT of the message buffer 7 is being processed is considered as completed, not aborted, and if the transmission of the message buffer 7 has been successfully completed, the ABT TRG has been cleared to 0. If the RDY bit in the next message buffer in the ABT region is cleared to 0, the internal ABT pointer is retained, but no recovery operation is performed even if ABT TRG is set to 1 and the ABT ends immediately. .

Notes m = 0 to 15

21.10.5 remote frame transmission

Remote frames can only be transmitted from the transmission message buffer. The RTR bit setting through the C0MCONFm register is data frame or remote frame transmission. Set the RTR bit to (1) Set up remote frame transfer.

Remark m = 0 to 15

21.11 power saving mode

21.11.1 CAN sleep mode

The CAN sleep mode may be use to set that CAN controller to standby mode to reduce power consumption. The CAN module may enter CAN sleep mode from all operation modes. The release of the CAN sleep mode returns the CAN module to an operation mode before entering the CAN sleep mode.

In that CAN sleep mode, the CAN module does not transmit the message even if the transmission request is issue or hang.

(1) Enter CAN sleep mode
The CPU issues a CAN sleep mode conversion request by writing 01B to the PSMODE[1:0] bit of the C0CTRL register.
This conversion request is only acknowledged under the following conditions. .

- The CAN module is already in the following operation mode
  - normal operation mode
  - Normal operation mode with ABT
  - Accept only mode
  - single-shot mode
  - self-test mode
  - Stop mode in all above operation modes
- CAN bus is idle (the 4th bit in inter-frame space is implicit) Note
- No transmission requests are pending

Note If the CAN bus is fixed to explicit, requests that transition to CAN sleep mode will be suspended.

In addition, that transition from the CAN stop mode to the CAN sleep mode is also independent of the CAN bus state.

Note: If a sleep mode request is pending and a message is received in the mailbox, the sleep mode request is not canceled but executed immediately after the mail store completes. If the CPU will perform an RX interrupt, this may cause FCAN in sleep mode. Therefore, in the sleep-free mode, the interrupt must use the MBON flag to check access to the message buffer and the receive history list register.

If none of the above conditions are met, the CAN module will run as follows:

- If that request enter the CAN sleep mode from the initialization mode, the CAN sleep mode conversion request is ignored and the CAN module remains in the initialization mode.
- If that CAN bus status is not bus idle (i.e. the CAN bus status is transmit or receive), the CAN sleep mode cannot be immediately converted to CAN sleep mode when requested in one of the mode of operation. In this case, the CAN sleep mode transition request remains suspended until the CAN bus state becomes bus idle (the fourth bit in the inter-frame space is an implicit bit). The PSMODE[1:0] bit remain 00B for that period from a CAN sleep mode request to a successful transition. The PSMODE [1:0] bit is set to 01B when the module enters CAN sleep mode. .
- If a request for conversion to an initialization mode and a request for conversion to a CAN sleep mode are issued simultaneously while the CAN module is in one of the operation modes, the request for conversion to the initialization mode is enabled. The CAN module enters an initialization mode at a predetermined time. At this time, the CAN sleep mode request does not remain suspended, but will be ignored.
- The initialization request has priority over the sleep mode request even if both the initialization mode and the sleep mode are not requested (i.e., the first request is not granted when the second request is issued). Sleep mode requests are canceled when initialization mode is requested. When a pending request for the initial mode appears, the subsequent request for the sleep mode is deleted at publication.

(2) Status of CAN sleep mode

The CAN module is in the following state after entering the CAN sleep mode.

- The internal operating clock has been stopped and power consumption is minimal.
- The CAN Acceptance Pin (CRxD) that detects the Down Edge function remains valid to wake the CAN module from the CAN bus.
- To wake the CAN module from the CPU, you can write data to the PSMODE[1:0] of the CAN Module Control Register (C0CTRL), but not to other CAN module registers or bits.
- In addition to C0LIPT, C0RGPT, C0LOPT and C0TGPT, the CAN module's Other registers can be read.
- CAN message cache registers cannot be written or read
- The MBON bit of the CAN0 Global Control Register (C0GMCTRL) is cleared.
- Applications for conversion to initialization mode are not acknowledgemented and are ignored.

(3) CAN sleep mode release

The CAN sleep mode is released by the following events:

- When the CPU writes PSMODE [1:0] bits of the 00B to C0 CTRL register
- The CAN receive pin (CRxD) detected a falling edge (i.e., that CAN bus level from implicit to explicit)

Note 1. Even if the drop edge occurs at the SOF receiving messages, it is not received and stored. If that CPU turn off the CAN clock while the CAN is sleep mode. Even so, the CAN sleep mode will not be released, and PSMODE[1:0] will continue to be 01B unless the CAN clock is provided again. In addition, that receive messages will not be receive later.

2. If a falling edge on the CAN receive pin (CRxD) is detected in the state where the CAN clock is provided, the PSMODE0 bits need to be cleared by software (see the processing in Figure 21-81 for details).

When the sleep mode is released, the CAN module returns to the operation mode when the CAN sleep mode is requested, and the PSMODE [1:0] bit of the C0CTRL register is reset to 00B. If the CAN sleep mode is released by a change in the CAN bus state, the CINTS5 bit of the C0INTS register is set to 1 regardless of the CIE bit of the C0IE register. After the CAN module is released from the CAN sleep mode, the CAN module enters the CAN bus again by automatically detecting 11 consecutive hidden levels on the CAN bus. User applications must wait until MBON=1 before accessing the message buffer. When the CAN module is in CAN sleep mode, a conversion request to the initialization mode is issued, and the request will be ignored. The CPU releases sleep mode through software before entering initialization mode.

Note: Note that if the CAN sleep mode is released by a CAN bus event; Therefore, if a CAN bus event occurs, a wake-up interrupt may occur even after requesting sleep mode.

Notes m = 0 to 15

21.11.2 CAN Stop Mode

The CAN stop mode may be used to set the CAN controller to standby mode to reduce power consumption. The CAN module can only enter CAN stop mode from CAN sleep mode. Release of CAN stop mode requires CAN module to be in CAN sleep mode.

The CAN stop mode can only be released by writing 01B to the PSMODE[1:0] bit of the C0CTRL register, not by a change in the CAN bus state (into CAN sleep mode). No messages are transmitted even if a transmission request is made or suspended.

(1) Enter CAN stop mode

A CAN stop mode conversion request is issued by writing 11B to the PSMODE[1:0] bit of the C0CTRL register. A CAN stop mode request is acknowledgemented only when the CAN module is in CAN sleep mode. Applications are ignored in other modes.

Note: To set the CAN module to CAN stop mode, the module must be in CAN sleep mode. To verify that the module is in sleep mode, check that PSMODE[1:0] is 01B, and then request CAN stop mode. If a bus change occurs at the CAN receive pin (CRxD) when this procedure is performed, the CAN sleep mode is automatically released. In this case, the CAN stop mode transition request cannot be acknowledgemented (however, in the state of providing the CAN clock, the PSMODE0 bits need to be cleared by software after a bus change occurs to the CAN receive pin (CRxD)).

(2) CAN Stop Mode Status

After entering CAN stop mode, the CAN module is in one of the following states.

- The internal operation clock stops and the power consumption is minimum.
- To wake up CAN module, the data can be written to PSMODE[1:0] bit of CAN module control register (C0CTRL).
- All CAN0 registers except the C0LIPT, C0RGPT, C0LOPT, and C0TGPT registers are readable.
- CAN0 message cache registers cannot be written or read.
- The MBON bit of the CAN0 Global Control Register (C0GMCTRL) is cleared.
- Applications that go to initialization mode are not acknowledgemented and are ignored.

(3) Release CAN Stop Mode

CAN stop mode can only be released by writing 01B to the PSMODE [1:0] bit of the C0CTRL register. When the CAN stop mode is released, the CAN module enters the CAN sleep mode. . .

When the CAN module is in the CAN stop mode, the request is switched to the initialization mode, and the request will be ignored. Before entering initialization mode. Before entering initialization mode, the CPU must release the stop mode and then the CAN sleep mode. It is not possible to go directly from CAN stop mode to another mode of operation, without CAN sleep mode, that request will be ignored. .

Remark m = 0 to 15

21.11.3 Power-saving mode

In some application systems, it may be necessary to put the CPU in power-saving mode to reduce power consumption. The CAN bus can wake the CPU from the power saving state by using the power saving mode specific to the CAN module and the power saving mode specific to the CPU. .

The following is an example of using power saving mode.

First, the CAN module is placed in CAN sleep mode (PSMODE = 01B). Next, place the CPU in power saving mode. If the CAN receive pin (CRxD) in this state detects an edge transition from implicit to explicit, the CINTS5 bit in the CAN module is set to 1. If the CIE5 bit of the C0CTRL register is set to 1, a wake-up interrupt (INTC0UP) is generated. The CAN module is automatically released from the CAN sleep mode (PSMODE = 00B) and returned to normal operation mode (however, in the state of providing the CAN clock, the PSMODE0 bits need to be cleared by software after the CAN receive pin (CRxD) detects a bus change. The CPU responds to the INTC0WUP and can release its own power saving mode and return to normal operation mode. .

To further reduce CPU power consumption, internal clocks, including those of the CAN module, may stop. In this case, the operation clock provided to the CAN module will stop after the CAN module is placed in the CAN sleep mode. The CPU then enters a power save mode in which the clock supplied to the CPU stops. If the CAN receive pin (CRxD) in this state detects an edge transition from implicit to explicit, the CAN module can set the CINTS5 bit to 1 and generate a wake-up interrupt (INTC0UP) even if no clock is provided. However, other functions no longer work because the clock supply to the CAN module has stopped and the module remains in CAN sleep mode. The CPU, in response to the INTC0WUP, releases its power saving mode, restores power to the internal clock (including supplying the clock to the CAN module) after the oscillation stabilization time has elapsed, and starts to execute instructions. When the clock power is restored, the CAN module is immediately released from CAN sleep mode and returned to normal operation mode (PSMODE=00B).

21.12 interrupt function

The CAN module provides six different interrupt sources.

The occurrence of these interrupt sources is stored in an interrupt state register. Four separate interrupt request signals are generated from six interrupt sources. When generating an interrupt request signal corresponding to two or more interrupt sources, the interrupt state register may be used to identify the interrupt source. After an interrupt source occurs, the software must clear the corresponding interrupt status bit to 0.

Table 21-20. List of CAN module interrupt sources

No.	interrupt status bit		interrupt enable bit		interrupt request signal	interrupt source description
	First Name	register	First Name	register		
1	CINTS0 Note	C0INS	CIE0 Note	C0IE	INTC0TRX	Successful transfer of message frames from message cache
2	CINTS1 Note	C0INS	CIE1 Note	C0IE	INTC0REC	The message cache receives valid message frames
3	CINTS2	C0INS	CIE2	C0IE	INTC0ERR	CAN Module Status Error Interrupt (Supplement 1)
4	CINTS3	C0INS	CIE3	C0IE		CAN Module Protocol Error Interrupt (Supplement 2)
5	CINTS4	C0INS	CIE4	C0IE		CAN Module Quorum Loss Interrupt
6	CINTS5	C0INS	CIE5	C0IE	INTC0WOUP	The CAN module wakes up the interrupt from sleep mode (Supplement 3)

Note must set the IE bit (Message Buffer Interrupt Enable Bit) in the C0MCTRLm register of the appropriate message buffer to 1 so that the message buffer participates in interrupt generation. .

Supplement 1. This interrupt is generated when the Send/Receive Error counter is at the warning level or in the error passive or bus off state.

2. This interrupt is generated when a content error, a form error, an ACK error, a bit error, or a CRC error occurs.
3. This interrupt is generated when the CAN module is woken from CAN sleep mode because a falling edge (CAN bus transition from implicit to explicit) is detected at the CAN receive pin.

Notes m = 0 to 15

21.13 Diagnostic features and special operation modes

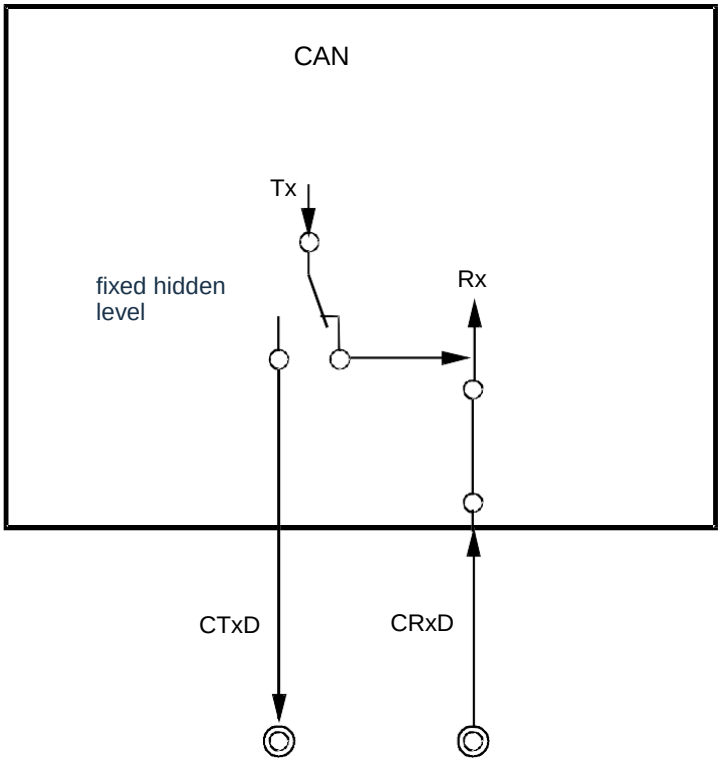
The CAN module provides only receive mode, single-shot mode and self-test mode to support operation of CAN bus diagnostic function or specific CAN communication method. .

21.13.1 Receive only mode

Only the receiving mode is used to monitor the received messages without causing any interference to the CAN bus and can be used for the CAN bus analysis node.

For example, this mode can be used for automatic baud rate detection. The baud rate in the CAN module will change until a "valid receive" is detected, so the baud rate in the module match ("valid receive" means that message frames are received in the CAN protocol layer without errors and with appropriate ACK connections to CAN bus nodes). Effective reception does not require message frames to be stored in the received message buffer (data frame) or the sent message buffer (remote frame). A valid reception event is indicated by setting the VALID bit of the C0CTRL register (1).

Figure 21-59. Terminal connection in CAN mode only receiving mode



In receive-only mode, message frames cannot be transferred from the CAN module to the CAN bus. Message caching is defined as the transition request that sends the message caching is pending.

The CAN transmit pin (CTxD) in the CAN module is fixed to the hidden level in the receive-only mode. Therefore, when receiving a message frame, no active error flag can be sent from the CAN module to the CAN bus even if the CAN bus error is detected. The Transmission Error Counter TEC is never updated because the transmission from the CAN module cannot be issued. Therefore, the CAN module in receive mode only does not enter the bus-off state.

In addition, ACK does not return to the CAN bus in this mode after the message frame is effectively received. Internally, the local node recognizes that it has transmitted an acknowledgment (ACK). Failed to transfer overload frame to CAN bus.

Note: If only two CAN nodes are connected to the CAN bus and one of the nodes is running in receive-only mode, there is no ACK on the CAN bus. Due to the lack of ACKs, the transmission node will transmit the active error flag and repeat the message frames. The transmission node becomes error passive after sending the message frame 16 times (assuming the error counter starts at 0 and no other errors occur). After the 17th frame of the message is transmitted, the Transmission node generates a passive error flag. The receiving node in receive mode only detects the first valid frame at this time, and the VALID bit is set to 1 for the first time. .

21.13.2 single-shot mode

In a single-shot mode, automatic retransmission defined in the CAN protocol is turned off (according to the CAN protocol, message frame transmissions aborted due to arbitration loss or error occurrence must be repeated (without software control)). All other behaviors of a single-shot mode are the same as for a normal mode of operation. Single-shot mode cannot be used in combination with normal mode with ABT.

Single-pass bad mode disables retransmission of aborted message frame transmissions according to the AL bit settings of the C0CTRL register. When the AL bit is cleared to 0, retransmission upon quorum loss and upon error occurrence is disabled. If the AL bit is set to 1, retransmission is disabled when an error occurs, but enabled when arbitration is lost. Therefore, the following event clears the TRQ bit in the message buffer defined as the sending message buffer to 0. .

- Successfully sent message frames
- Arbitration Loss Occurs for Sending Message Frames
- Error occurs when sending message frame

By examining the CINTS4 and CINTS3 bits of the C0INTS register, the event arbitration loss and error occurrence can be distinguished, and the error type can be identified by reading the LEC[2:0] bits of the C0LEC register.

When the message frame is successfully transmitted, the transmission completion interrupt bit CINTS0 of the C0INTS register is set to 1. If the CIE0 bit of the C0IE register is set to 1, an interrupt request signal is output.

Single-shot mode can be used when simulating time-triggered communication methods, such as TTCAN level 1.).

Note: AL bits are only valid in single-shot mode. Does not affect retransmission of arbitration losses in other modes of operation.

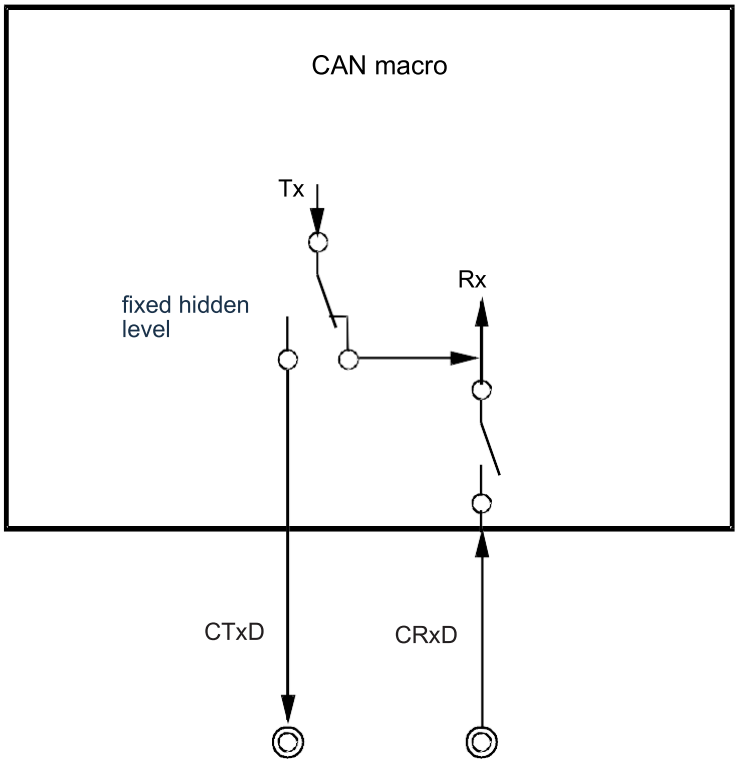
21.13.3 self-test mode

In the self-test mode, the message frame transmission and reception can be tested without connecting the CAN node to the CAN bus without affecting the CAN bus. .

In the self-test mode, the CAN module is completely disconnected from the CAN bus, but the transmission and reception are internally circulated. The CAN transmit pin (CTxD) is fix to that hidden level. .

If a CAN module detects a CAN receive pin (CRxD) upstream falling edge after entering a CAN sleep mode from a self-test mode, the module is released from the CAN sleep mode in the same manner as other modes of operation. (However, to release the CAN sleep mode, the PSMODE0 bits need to be cleared by software after a falling edge on the CAN receive pin (CRxD) is detected in the CAN clock state. For the module to remain in CAN sleep mode, the CAN receive pin (CRxD) needs to be used as the port pin.

Figure 21-60. CAN module terminal connection in self-test mode



21.13.4 Receive/Send Operation in Operation Mode

Table 21-21 shows a summary of receive/send operations in each mode of operation.

Table 21-21. Summary of Send/Receive in Operational Mode

operation mode	Send Data/Remote Frames	Send a reply	Send error/overload frames	Transmission Retry	Automatic Block Transfer (ABT)	Set VALID bit	Store data to message cache
initialization mode	No	No	No	No	No	No	No
normal operation mode	Yes	Yes	Yes	Yes	No	Yes	Yes
Normal operation mode with ABT	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Receive mode only	No	No	No	No	No	Yes	Yes
single-shot mode	Yes	Yes	Yes	No Note1	No	Yes	Yes
self-test mode e	Yes Note2	Yes Note2	Yes Note2	Yes Note2	No	Yes Note2	Yes Note2

Notes 1. When arbitration is lost, the AL bits of the C0CTRL register can control retransmission.

2. Each signal is generated not to the outside but to the CAN module.

21.14 timestamp function

CAN is an asynchronous serial protocol. All nodes connected to the CAN bus have local autonomous clocks. Therefore, the clock of the node is irrelevant (that is, the clock is asynchronous and may have different frequencies).

However, in some applications, you need to establish a common time base through the network (=global time base). In order to establish a global time base, the timestamp function is needed. The basic mechanism of the timestamp function is to capture the timer value triggered by the signal on the CAN bus. .

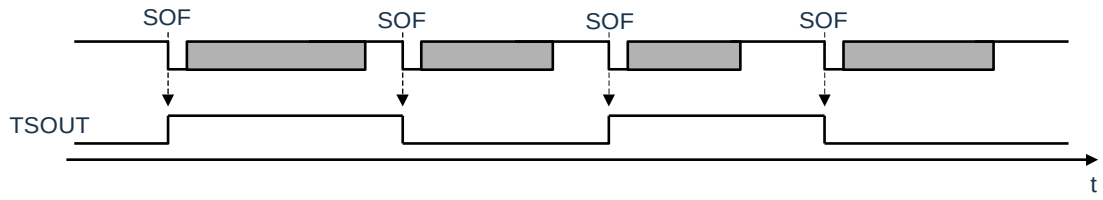
21.14.1 timestamp function

The CAN controller support that capture of timer value triggered by a particular frame. In addition to that CAN controller, a 16-bit on-chip capture timer unit in the microcontroller system is also use. A 16-bit capture timer unit captures a timer value according to a trigger signal (TSOUT) for capturing a timer value output when a data frame is received from the CAN controller. The CPU can retrieve the time at which the capture event occurred, that is, the timestamp of the message received from the CAN bus, by reading the captured value. The TSOUT signal can be selected from the following two event sources and specified by the TSSEL bit of the C0TS register.

- SOF Events (Frame Start) (TSSEL =0)
- EOF event (last bit of the end frame) (TSSEL =1)

The TSOUT signal enable requires the C0TS register to have a TSEN bit of 1.

Figure 21-61. Time sequence diagram of acquisition signal TSOUT



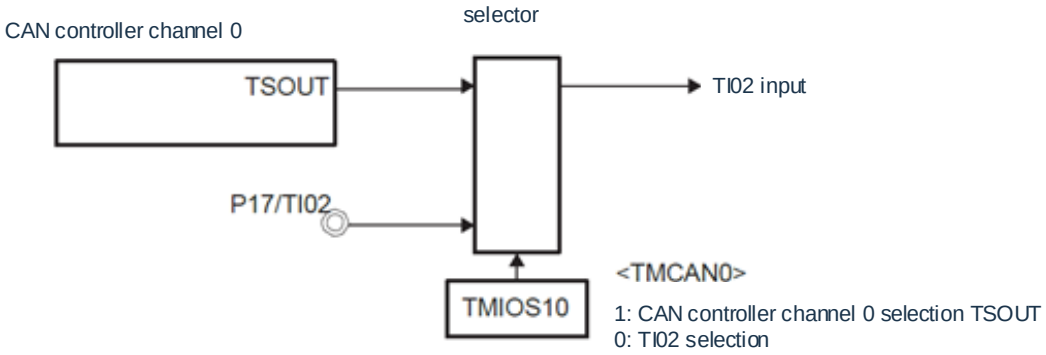
The TSOUT signal switch its level when a selected event occur during that reception of the data frame (in the above time sequence diagram, the SOF serve as the source of the trigger event). To use the TSOUT signal to capture timer values, you must detect capture signals for rising and falling edges at runtime. .

This timestamp function is controlled by the TSLOCK bit of the C0TS register. When TSLOCK is cleared to 0, the TSOUT bit switches when the selected event occurs. If the TSLOCK bit is set to 1, TSOUT switches when the selected event occurs, but the switch stops because the TSEN bit automatically clears to 0 when the message is stored in the message buffer 0. This inhibits the subsequent switching of the TSOUT so that the timestamp value (=last captured) of the last switch is saved as the timestamp value of the received data frame in the message buffer 0. .

Note: The timestamp function using TSLOCK bits stops switching TSOUT bits by receiving the data frames in the message buffer 0. Therefore, message buffer 0 must be set as the receive message buffer. Because the receive message buffer cannot receive remote frames, it is not possible to stop the TSOUT bit switching by receiving remote frames. When a data frame is received in a message buffer other than the message buffer (message buffer 0), switching of the TSOUT bits does not stop. For these reasons, when the CAN module is in the normal mode of operation with ABT, it cannot receive data frames in the message buffer 0, because the message buffer 0 must be set as the send message buffer. Therefore, in this mode of operation, you cannot use the TSLOCK bit to stop the function of switching TSOUT bits.

By switching the input source (using TMIOS1), the capture trigger signal (TSOUT of CAN controller channel 0) can be input to channel 2 of timer array unit 0 without having to externally connect the TSOUT of CAN controller channel 0 and TI02.

Figure 21-62. Switching of input sources



Note 1. TIOS10: Time Input Select bit 0 of register TIOS1 (Reference section 6 Time Array unit).

2. The pins available vary by product. For more information, see the 2.1 Pin Feature List.

21.15 Baud rate setting

21.15.1 Baud rate setting

Make sure that the settings are within the limits to ensure proper operation of the CAN controller, as shown below. .

(a) 5TQ SPT (sampleing points)
17TQ SPT = TSEG1 +1TQ
(b) 8TQ DBT (data bit time) 25TQ
DBT = TSEG1 + TSEG2 + 1TQ = TSEG2 +SPT
(c) 1TQ SJW (Sync Jump Width) 4TQ
SJW DBT –SPT
(d) 4TQ TSEG1 16TQ [3 (TSEG1 [3:0] 15’s set value]
(e) 1TQ TSEG2 8TQ [0 (TSEG2 [2:0] 7’s set value]

Note TQ = 1/fTQ (fTQ: CAN protocol layer base system clock)
TSEG1 [3:0]: Bit 0 to 3TSEG2[2:0] of the CAN0 bit rate register (C0BTR): Bits 8 through 10 of the CAN0 Bit Rate Register (C0BTR)

Table 21-22 show a combination of bit rates that meet that above condition.

Table 21-22. Bit rate combinations may be set (1/3)

Valid bit rate settings					C0BTR register setting value		Sampling points (in %)
DBT length	Sync Segment	propagation	Phase Segment 1	Phase Segment 2	TSEG1 [3:0]	TSEG2 [2:0]	
25	1	8	8	8	1111	111	68.0
24	1	7	8	8	1110	111	66.7
24	1	9	7	7	1111	110	70.8
23	1	6	8	8	1101	111	65.2
23	1	8	7	7	1110	110	69.6
23	1	10	6	6	1111	101	73.9
22	1	5	8	8	1100	111	63.6
22	1	7	7	7	1101	110	68.2
22	1	9	6	6	1110	101	72.7
22	1	11	5	5	1111	100	77.3
21	1	4	8	8	1011	111	61.9
21	1	6	7	7	1100	110	66.7
21	1	8	6	6	1101	101	71.4
21	1	10	5	5	1110	100	76.2
21	1	12	4	4	1111	011	81.0
20	1	3	8	8	1010	111	60.0
20	1	5	7	7	1011	110	65.0
20	1	7	6	6	1100	101	70.0
20	1	9	5	5	1101	100	75.0
20	1	11	4	4	1110	011	80.0
20	1	13	3	3	1111	010	85.0
19	1	2	8	8	1001	111	57.9
19	1	4	7	7	1010	110	63.2
19	1	6	6	6	1011	101	68.4
19	1	8	5	5	1100	100	73.7
19	1	10	4	4	1101	011	78.9
19	1	12	3	3	1110	010	84.2
19	1	14	2	2	1111	001	89.5
18	1	1	8	8	1000	111	55.6
18	1	3	7	7	1001	110	61.1
18	1	5	6	6	1010	101	66.7
18	1	7	5	5	1011	100	72.2
18	1	9	4	4	1100	011	77.8
18	1	11	3	3	1101	010	83.3
18	1	13	2	2	1110	001	88.9
18	1	15	1	1	1111	000	94.4

Table 21-22. Bit rate combinations may be set (2/3)

Valid bit rate settings					C0BTR register setting value		Sampling points (in %)
DBT length	Sync Segment	propagation	Phase Segment 1	Phase Segment 2	TSEG1 [3:0]	TSEG2 [2:0]	
17	1	2	7	7	1000	110	58.8
17	1	4	6	6	1001	101	64.7
17	1	6	5	5	1010	100	70.6
17	1	8	4	4	1011	011	76.5
17	1	10	3	3	1100	010	82.4
17	1	12	2	2	1101	001	88.2
17	1	14	1	1	1110	000	94.1
16	1	1	7	7	0111	110	56.3
16	1	3	6	6	1000	101	62.5
16	1	5	5	5	1001	100	68.8
16	1	7	4	4	1010	011	75.0
16	1	9	3	3	1011	010	81.3
16	1	11	2	2	1100	001	87.5
16	1	13	1	1	1101	000	93.8
15	1	2	6	6	0111	101	60.0
15	1	4	5	5	1000	100	66.7
15	1	6	4	4	1001	011	73.3
15	1	8	3	3	1010	010	80.0
15	1	10	2	2	1011	001	86.7
15	1	12	1	1	1100	000	93.3
14	1	1	6	6	0110	101	57.1
14	1	3	5	5	0111	100	64.3
14	1	5	4	4	1000	011	71.4
14	1	7	3	3	1001	010	78.6
14	1	9	2	2	1010	001	85.7
14	1	11	1	1	1011	000	92.9
13	1	2	5	5	0110	100	61.5
13	1	4	4	4	0111	011	69.2
13	1	6	3	3	1000	010	76.9
13	1	8	2	2	1001	001	84.6
13	1	10	1	1	1010	000	92.3
12	1	1	5	5	0101	100	58.3
12	1	3	4	4	0110	011	66.7
12	1	5	3	3	0111	010	75.0
12	1	7	2	2	1000	001	83.3
12	1	9	1	1	1001	000	91.7

Table 21-22. Bit rate combinations may be set (3/3)

Valid bit rate settings					C0BTR register setting value		Sampling points (in %)
DBT length	Sync Segment	propagation	Phase Segment 1	Phase Segment 2	TSEG1 [3:0]	TSEG2 [2:0]	
11	1	2	4	4	0101	011	63.6
11	1	4	3	3	0110	010	72.7
11	1	6	2	2	0111	001	81.8
11	1	8	1	1	1000	000	90.9
10	1	1	4	4	0100	011	60.0
10	1	3	3	3	0101	010	70.0
10	1	5	2	2	0110	001	80.0
10	1	7	1	1	0111	000	90.0
9	1	2	3	3	0100	010	66.7
9	1	4	2	2	0101	001	77.8
9	1	6	1	1	0110	000	88.9
8	1	1	3	3	0011	010	62.5
8	1	3	2	2	0100	001	75.0
8	1	5	1	1	0101	000	87.5
7 Note	1	2	2	2	0011	001	71.4
7 Note	1	4	1	1	0100	000	85.7
6 Note	1	1	2	2	0010	001	66.7
6 Note	1	3	1	1	0011	000	83.3
5 Note	1	2	1	1	0010	000	80.0
4 Note	1	1	1	1	0001	000	75.0

Note Setting DBT with a value of 7 or less is valid only if the value of the C0BRP register is not 00H.

Note: The values in Table 21-22 do not guarantee the operation of the network system. The efficiency of the network system is detected by the oscillation error and the delay of the CAN bus and the CAN transceiver.

21.15.2 Representative example of baud rate setting

Tables 21-23 and 21-24 show representative examples of baud rate settings.

Table 21-23. Representative example of baud rate setting (f_{CANMOD} = 8 MHz) (1/2)

Set the baud rate value (in kbps)	C0 BRP Partition Ratio	C0BRP register setting value	Valid Bit Rate Settings (in kbps)					C0 BTR register set value		Sampling Point (in %)
			DBT length	Sync Segment	propagation segment	Phase Segment 1	Phase Segment 2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	8	1	1	3	3	0011	010	62.5
1000	1	00000000	8	1	3	2	2	0100	001	75.0
1000	1	00000000	8	1	5	1	1	0101	000	87.5
500	1	00000000	16	1	1	7	7	0111	110	56.3
500	1	00000000	16	1	3	6	6	1000	101	62.5
500	1	00000000	16	1	5	5	5	1001	100	68.8
500	1	00000000	16	1	7	4	4	1010	011	75.0
500	1	00000000	16	1	9	3	3	1011	010	81.3
500	1	00000000	16	1	11	2	2	1100	001	87.5
500	1	00000000	16	1	13	1	1	1101	000	93.8
500	2	00000001	8	1	1	3	3	0011	010	62.5
500	2	00000001	8	1	3	2	2	0100	001	75.0
500	2	00000001	8	1	5	1	1	0101	000	87.5
250	2	00000001	16	1	1	7	7	0111	110	56.3
250	2	00000001	16	1	3	6	6	1000	101	62.5
250	2	00000001	16	1	5	5	5	1001	100	68.8
250	2	00000001	16	1	7	4	4	1010	011	75.0
250	2	00000001	16	1	9	3	3	1011	010	81.3
250	2	00000001	16	1	11	2	2	1100	001	87.5
250	2	00000001	16	1	13	1	1	1101	000	93.8
250	4	00000011	8	1	3	2	2	0100	001	75.0
250	4	00000011	8	1	5	1	1	0101	000	87.5
125	4	00000011	16	1	1	7	7	0111	110	56.3
125	4	00000011	16	1	3	6	6	1000	101	62.5
125	4	00000011	16	1	5	5	5	1001	100	68.8
125	4	00000011	16	1	7	4	4	1010	011	75.0
125	4	00000011	16	1	9	3	3	1011	010	81.3
125	4	00000011	16	1	11	2	2	1100	001	87.5
125	4	00000011	16	1	13	1	1	1101	000	93.8
125	8	00000111	8	1	3	2	2	0100	001	75.0
125	8	00000111	8	1	5	1	1	0101	000	87.5

Note: The values in Table 21-23 do not guarantee the operation of the network system. The efficiency of the network system is detected by crystal oscillation error and delay of CAN bus and CAN transceiver.

Table 21-23. Representative example of baud rate setting (f_{CANMOD} = 8 MHz) (2/2)

Set the baud rate value (in kbps)	C0 BRP Partition Ratio	C0BRP register setting value	Valid Bit Rate Settings (in kbps)					C0 BTR register set value		Sampling Point (in %)
			DBT length	Sync Segment	propagation segment	Phase Segment 1	Phase Segment 2	TSEG1 [3:0]	TSEG2 [2:0]	
100	4	00000011	20	1	7	6	6	1100	101	70.0
100	4	00000011	20	1	9	5	5	1101	100	75.0
100	5	00000100	16	1	7	4	4	1010	011	75.0
100	5	00000100	16	1	9	3	3	1011	010	81.3
100	8	00000111	10	1	3	3	3	0101	010	70.0
100	8	00000111	10	1	5	2	2	0110	001	80.0
100	10	00001001	8	1	3	2	2	0100	001	75.0
100	10	00001001	8	1	5	1	1	0101	000	87.5
83.3	4	00000011	24	1	7	8	8	1110	111	66.7
83.3	4	00000011	24	1	9	7	7	1111	110	70.8
83.3	6	00000101	16	1	5	5	5	1001	100	68.8
83.3	6	00000101	16	1	7	4	4	1010	011	75.0
83.3	6	00000101	16	1	9	3	3	1011	010	81.3
83.3	6	00000101	16	1	11	2	2	1100	001	87.5
83.3	8	00000111	12	1	5	3	3	0111	010	75.0
83.3	8	00000111	12	1	7	2	2	1000	001	83.3
83.3	12	00001011	8	1	3	2	2	0100	001	75.0
83.3	12	00001011	8	1	5	1	1	0101	000	87.5
33.3	10	00001001	24	1	7	8	8	1110	111	66.7
33.3	10	00001001	24	1	9	7	7	1111	110	70.8
33.3	12	00001011	20	1	7	6	6	1100	101	70.0
33.3	12	00001011	20	1	9	5	5	1101	100	75.0
33.3	15	00001110	16	1	7	4	4	1010	011	75.0
33.3	15	00001110	16	1	9	3	3	1011	010	81.3
33.3	16	00001111	15	1	6	4	4	1001	011	73.3
33.3	16	00001111	15	1	8	3	3	1010	010	80.0
33.3	20	00010011	12	1	5	3	3	0111	010	75.0
33.3	20	00010011	12	1	7	2	2	1000	001	83.3
33.3	24	00010111	10	1	3	3	3	0101	010	70.0
33.3	24	00010111	10	1	5	2	2	0110	001	80.0
33.3	30	00011101	8	1	3	2	2	0100	001	75.0
33.3	30	00011101	8	1	5	1	1	0101	000	87.5

Note: The values in Table 21-23 do not guarantee the operation of the network system. The efficiency of the network system is detected by the oscillation error and the delay of the CAN bus and the CAN transceiver.

Table 21-24. Representative example of baud rate setting (f_{CANMOD} = 16 MHz) (1/2)

Set the baud rate value (in kbps)	C0 BRP Partition Ratio	C0BRP register setting value	Valid Bit Rate Settings (in kbps)					C0 BTR register set value		Sampling Point (in %)
			DBT length	Sync Segment	propagation segment	Phase Segment 1	Phase Segment 2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	16	1	1	7	7	0111	110	56.3
1000	1	00000000	16	1	3	6	6	1000	101	62.5
1000	1	00000000	16	1	5	5	5	1001	100	68.8
1000	1	00000000	16	1	7	4	4	1010	011	75.0
1000	1	00000000	16	1	9	3	3	1011	010	81.3
1000	1	00000000	16	1	11	2	2	1100	001	87.5
1000	1	00000000	16	1	13	1	1	1101	000	93.8
1000	2	00000001	8	1	3	2	2	0100	001	75.0
1000	2	00000001	8	1	5	1	1	0101	000	87.5
500	2	00000001	16	1	1	7	7	0111	110	56.3
500	2	00000001	16	1	3	6	6	1000	101	62.5
500	2	00000001	16	1	5	5	5	1001	100	68.8
500	2	00000001	16	1	7	4	4	1010	011	75.0
500	2	00000001	16	1	9	3	3	1011	010	81.3
500	2	00000001	16	1	11	2	2	1100	001	87.5
500	2	00000001	16	1	13	1	1	1101	000	93.8
500	4	00000011	8	1	3	2	2	0100	001	75.0
500	4	00000011	8	1	5	1	1	0101	000	87.5
250	4	00000011	16	1	3	6	6	1000	101	62.5
250	4	00000011	16	1	5	5	5	1001	100	68.8
250	4	00000011	16	1	7	4	4	1010	011	75.0
250	4	00000011	16	1	9	3	3	1011	010	81.3
250	4	00000011	16	1	11	2	2	1100	001	87.5
250	8	00000111	8	1	3	2	2	0100	001	75.0
250	8	00000111	8	1	5	1	1	0101	000	87.5
125	8	00000111	16	1	3	6	6	1000	101	62.5
125	8	00000111	16	1	7	4	4	1010	011	75.0
125	8	00000111	16	1	9	3	3	1011	010	81.3
125	8	00000111	16	1	11	2	2	1100	001	87.5
125	16	00001111	8	1	3	2	2	0100	001	75.0
125	16	00001111	8	1	5	1	1	0101	000	87.5

Note: The values in Table 21-24 do not guarantee the operation of the network system. The efficiency of the network system is detected by the oscillation error and the delay of the CAN bus and the CAN transceiver.

Table 21-24. Representative example of baud rate setting (f_{CANMOD} = 16 MHz) (2/2)

Set the baud rate value (in kbps)	C0 BRP Partition Ratio	C0BRP register setting value	Valid Bit Rate Settings (in kbps)					C0 BTR register set value		Sampling Point (in %)
			DBT length	Sync Segment	propagation segment	Phase Segment 1	Phase Segment 2	TSEG1 [3:0]	TSEG2 [2:0]	
100	8	00000111	20	1	9	5	5	1101	100	75.0
100	8	00000111	20	1	11	4	4	1110	011	80.0
100	10	00001001	16	1	7	4	4	1010	011	75.0
100	10	00001001	16	1	9	3	3	1011	010	81.3
100	16	00001111	10	1	3	3	3	0101	010	70.0
100	16	00001111	10	1	5	2	2	0110	001	80.0
100	20	00010011	8	1	3	2	2	0100	001	75.0
83.3	8	00000111	24	1	7	8	8	1110	111	66.7
83.3	8	00000111	24	1	9	7	7	1111	110	70.8
83.3	12	00001011	16	1	7	4	4	1010	011	75.0
83.3	12	00001011	16	1	9	3	3	1011	010	81.3
83.3	12	00001011	16	1	11	2	2	1100	001	87.5
83.3	16	00001111	12	1	5	3	3	0111	010	75.0
83.3	16	00001111	12	1	7	2	2	1000	001	83.3
83.3	24	00010111	8	1	3	2	2	0100	001	75.0
83.3	24	00010111	8	1	5	1	1	0101	000	87.5
33.3	30	00011101	24	1	7	8	8	1110	111	66.7
33.3	30	00011101	24	1	9	7	7	1111	110	70.8
33.3	24	00010111	20	1	9	5	5	1101	100	75.0
33.3	24	00010111	20	1	11	4	4	1110	011	80.0
33.3	30	00011101	16	1	7	4	4	1010	011	75.0
33.3	30	00011101	16	1	9	3	3	1011	010	81.3
33.3	32	00011111	15	1	8	3	3	1010	010	80.0
33.3	32	00011111	15	1	10	2	2	1011	001	86.7
33.3	37	00100100	13	1	6	3	3	1000	010	76.9
33.3	37	00100100	13	1	8	2	2	1001	001	84.6
33.3	40	00100111	12	1	5	3	3	0111	010	75.0
33.3	40	00100111	12	1	7	2	2	1000	001	83.3
33.3	48	00101111	10	1	3	3	3	0101	010	70.0
33.3	48	00101111	10	1	5	2	2	0110	001	80.0
33.3	60	00111011	8	1	3	2	2	0100	001	75.0
33.3	60	00111011	8	1	5	1	1	0101	000	87.5

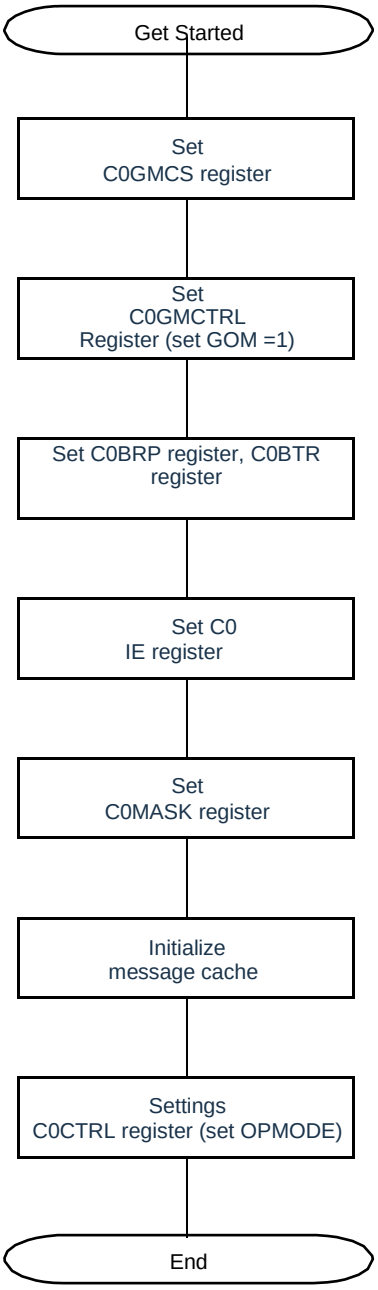
Note: The values in Table 21-24 do not guarantee the operation of the network system. The efficiency of the network system is detected by the oscillation error and the delay of the CAN bus and the CAN transceiver.

21.16 Operation of CAN controller

This chapter's operation flow is the CAN controller's operation flow. Development Refer to the process flow in this chapter.

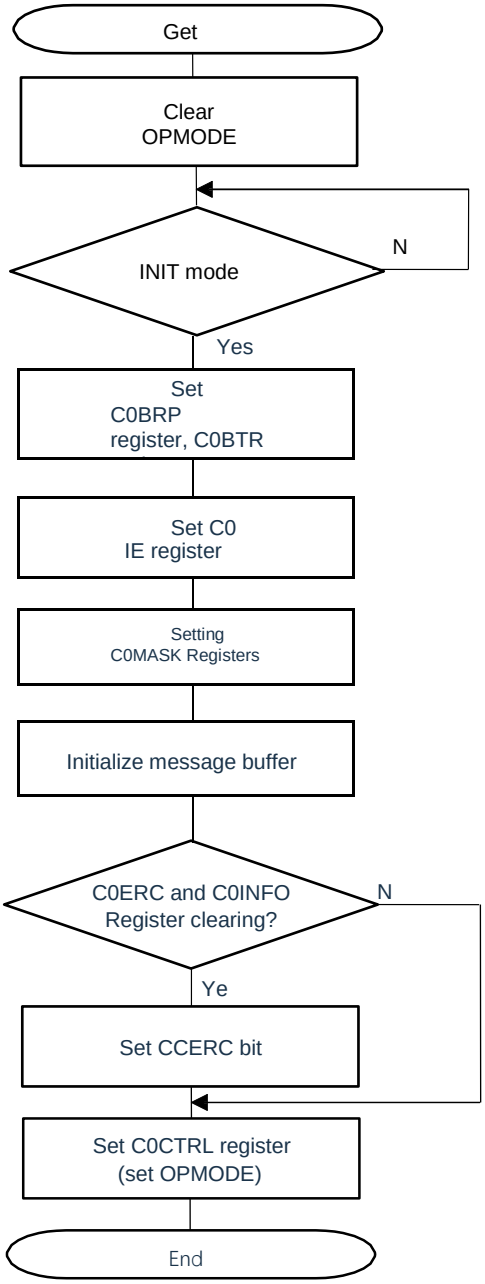
Notes m = 0 to 15

Figure 21-63. Initialization



Note OPMODE: Normal operation mode, Normal operation mode with ABT, Receive only mode, Single-shot mode, Self Test mode

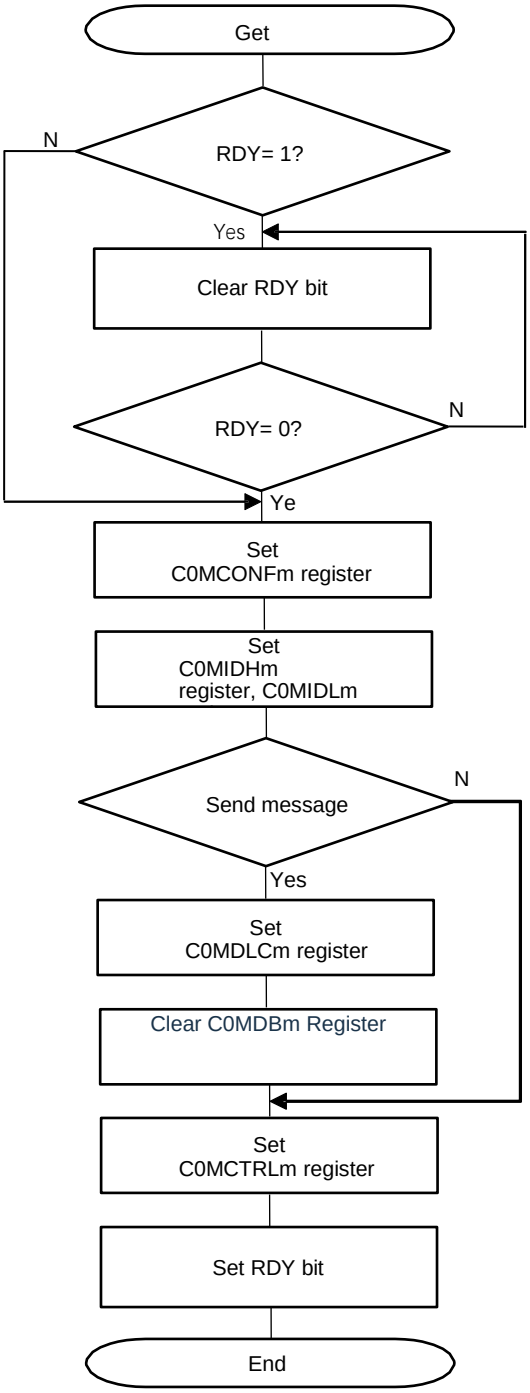
Figure 21-64. Reinitialization



Note: After you set the CAN module to initialization mode, do not immediately set to another mode of operation. If you need to immediately set the module to a different mode of operation, go to a register other than the C0CTRL and C0GMCTRL registers (for example, set up a message buffer).

Note OPMODE: Normal Operation Mode, Normal Operation Mode with ABT, Receive Only Mode, Single-shot mode, Self Test Mode.

Figure 21-65. Message cache initialization



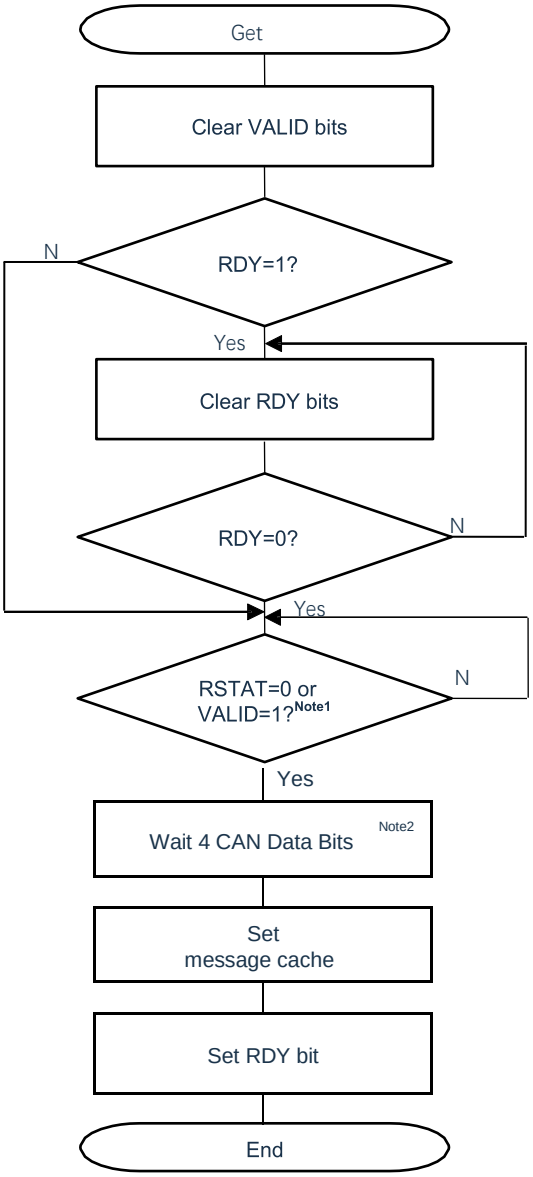
Note: 1. The RDY bits must be cleared before the message cache is initialized.

2. Reference the following settings for unused message caching.

- Clear the RDY of the C0MCTRLm register, the TRQ and DN bits are 0.
- Clear the C0MCONFm register with MA0 bit 0.

Figure 21-66 shows the received message cache processing (C0MCONFm register MT[2:0] bits =001B to 101B).

Figure 21-66. Message Cache Redefinition



Notes 1. The acknowledgement of receipt of the message is because the RDY bit is set after the message has been fully received. .

2. Avoid redefining message buffers during stored message receiving operations by waiting for additional 4 CAN data bits. .

Figure 21-67 shows the processing of the transmitted message buffer during transmission (C0MCONFm register MT[2:0] bit=000B).

Figure 21-67. Redefinition of message cache during transmission

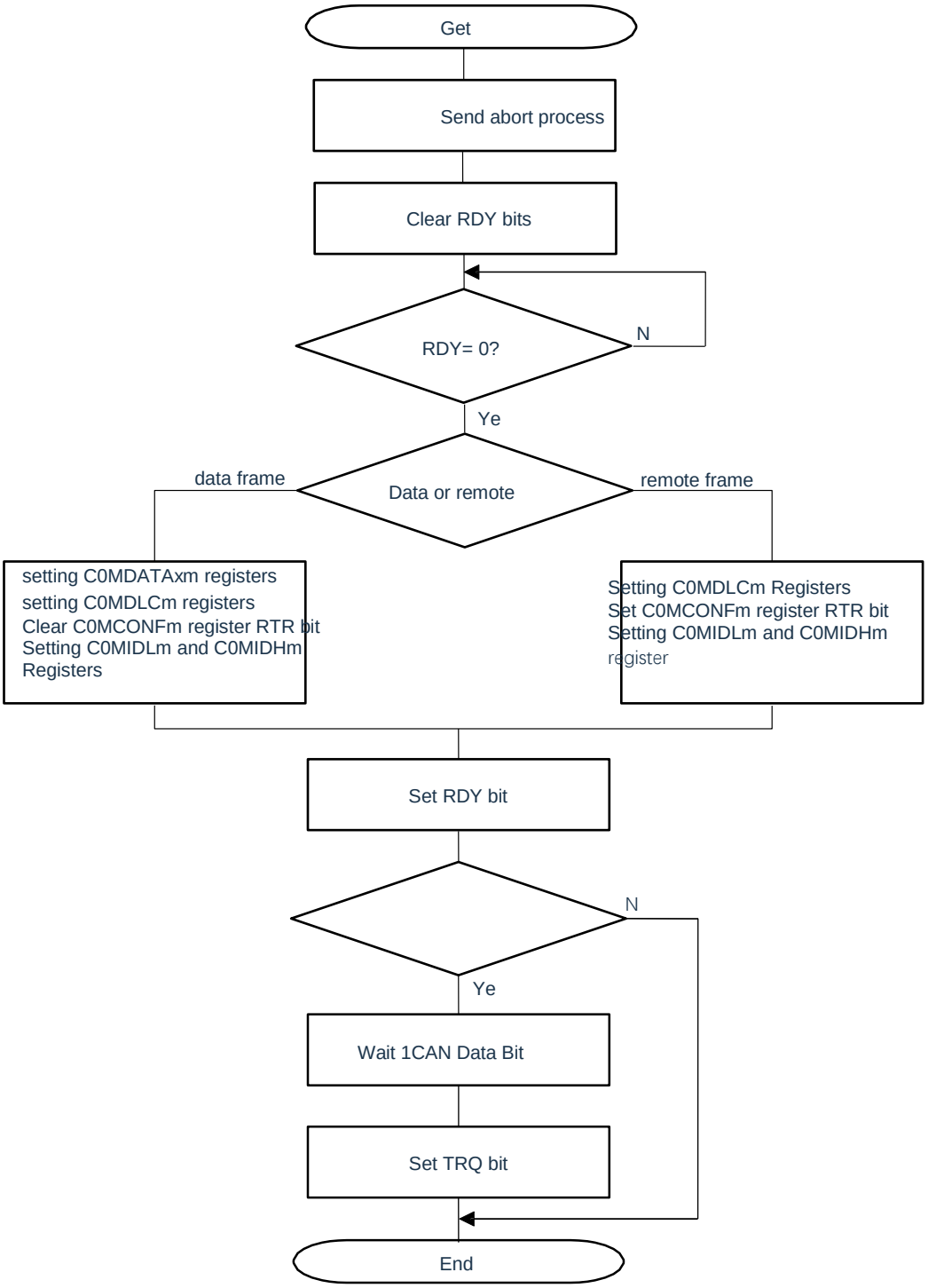
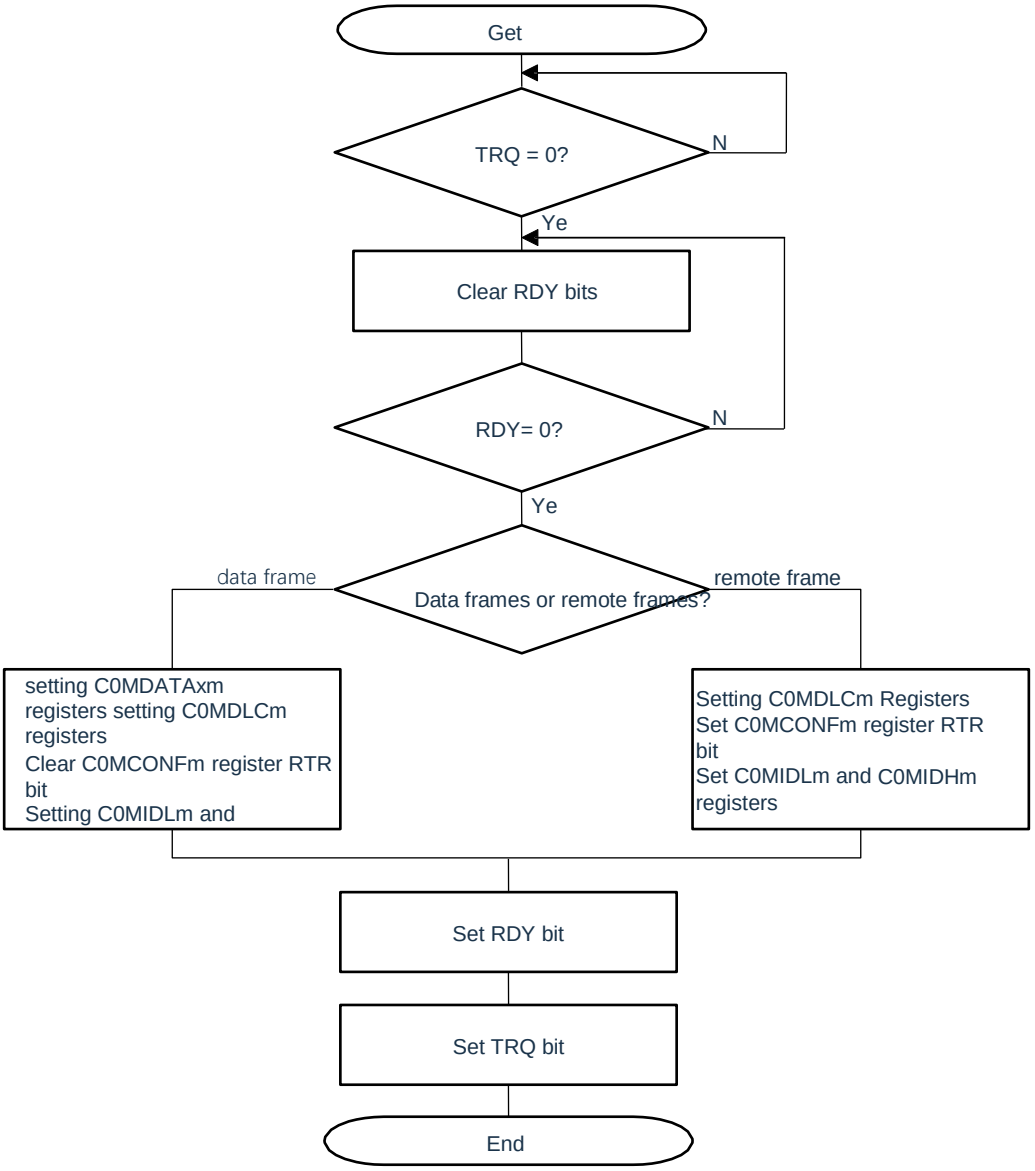


Figure 21-68 shows the transmit message cache processing (C0MCONFm register MT[2:0] bit=000B).

Figure 21-68. Message sending processing

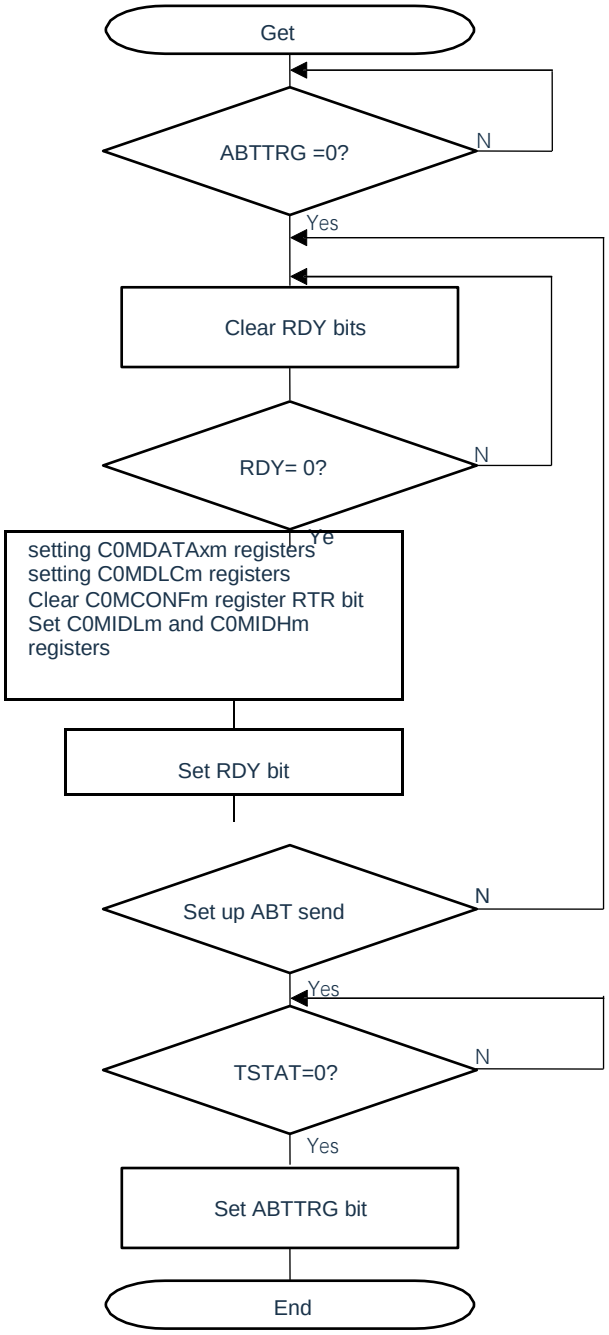


Note: 1. Set the TRQ bit after RDY is set.

2. RDY bit and TRQ bit cannot be set simultaneously.

Figure 21-69 shows the transmit message cache processing (C0MCONFm register MT[2:0] bit=000B).

Figure 21-69. ABT Message Delivery Processing

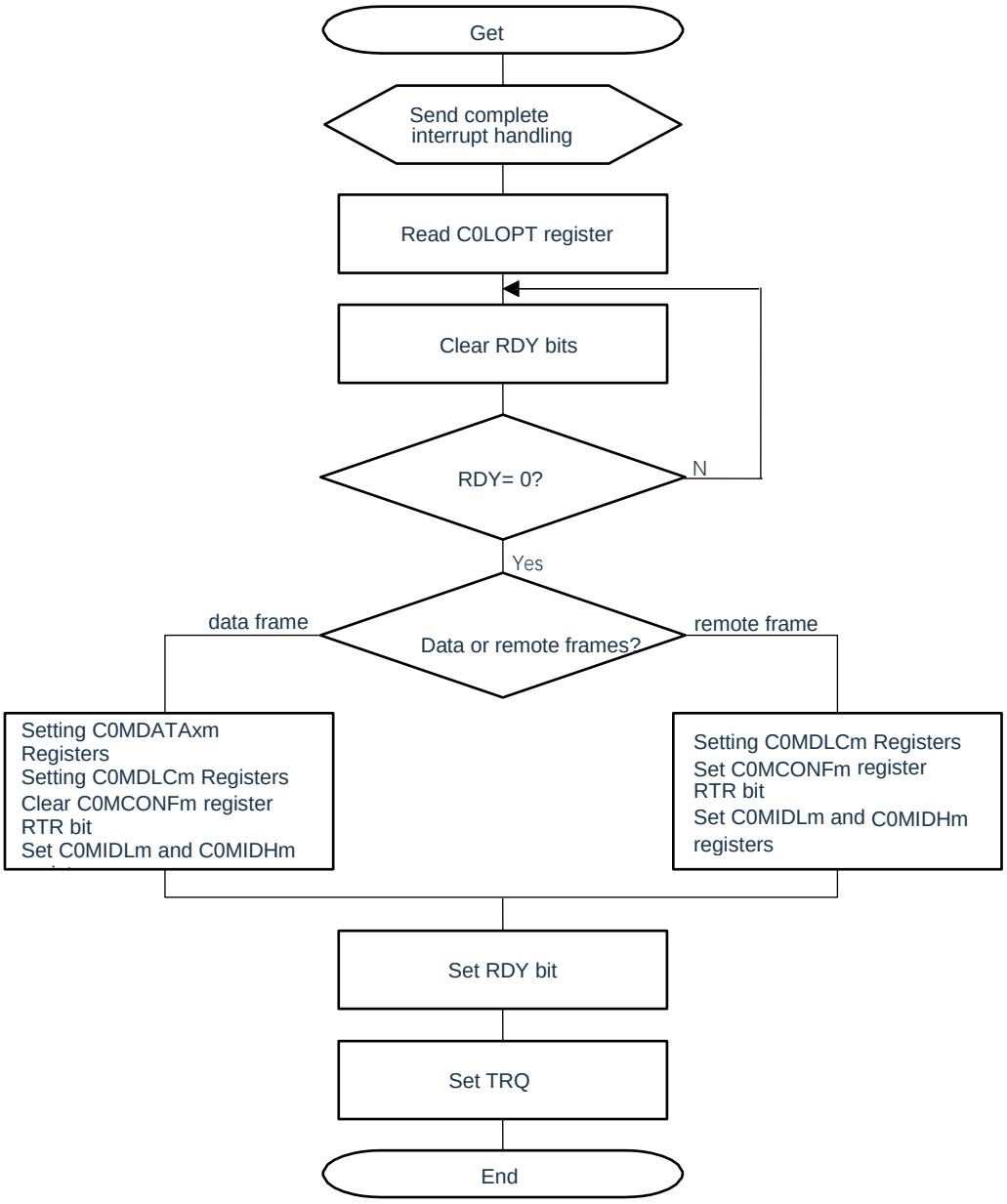


Note: After the TSTAT bit is cleared to 0, the ABTTRG bit should be set to 1. You must continuously check the TSTAT bit and set the ABTTRG bit to 1. .

Note: This process (using ABS normal mode of operation) can only be applied to message buffers 0 through 7.

For message buffers other than ABT message buffers, see Figure 21-68.

Figure 21-70. Transmit Through Interrupt (using C0LOPT register)

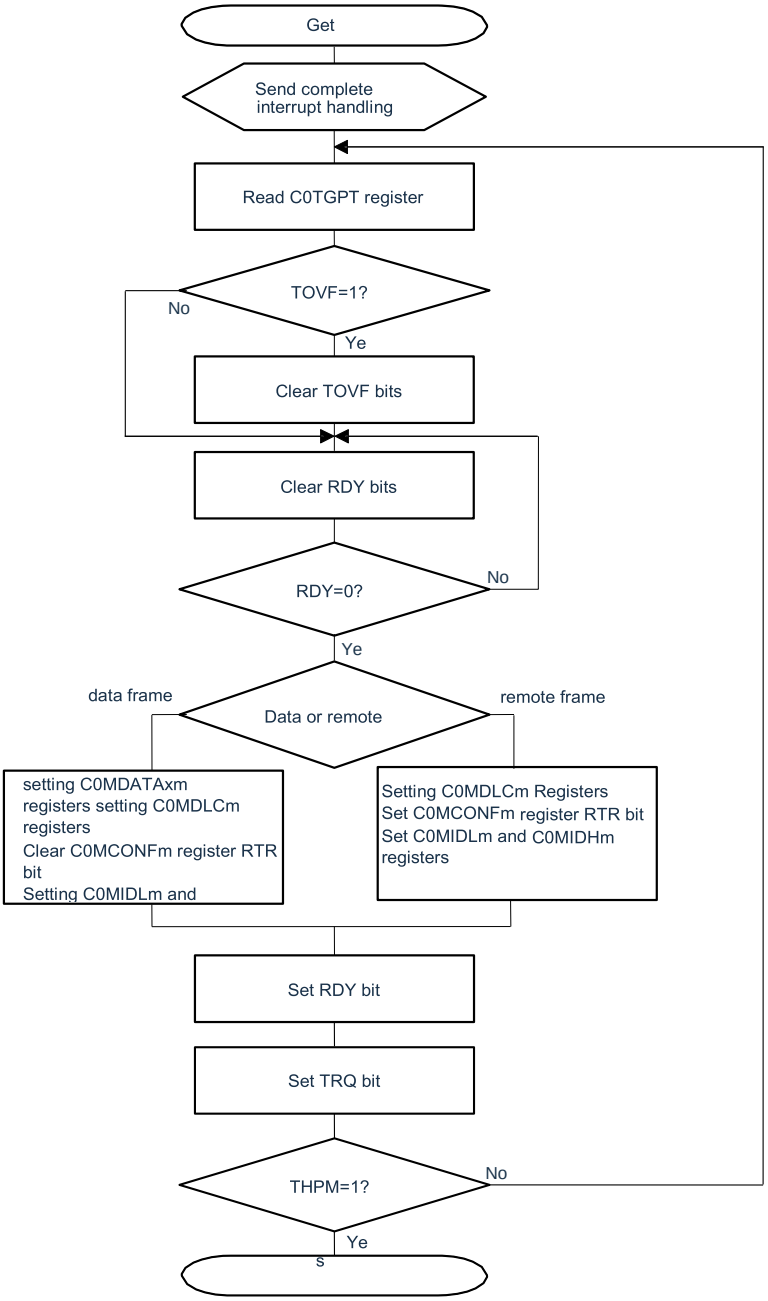


Note 1. The TRQ bit is set after the RDY setting.

2. RDY bit and TRQ bit cannot be set simultaneously.

Note: Check the MBON flag at the beginning and end of the interrupt to check access to the message buffer and the TX history list register to prevent pending sleep patterns. If MBON is detected to be cleared, the results and reprocessing operations must be discarded after MBON is set again. . It is recommend to remove some sleep mode requests before handle tx interrupts.

Figure 21-71. Send by interrupt (using C0TGPT register)

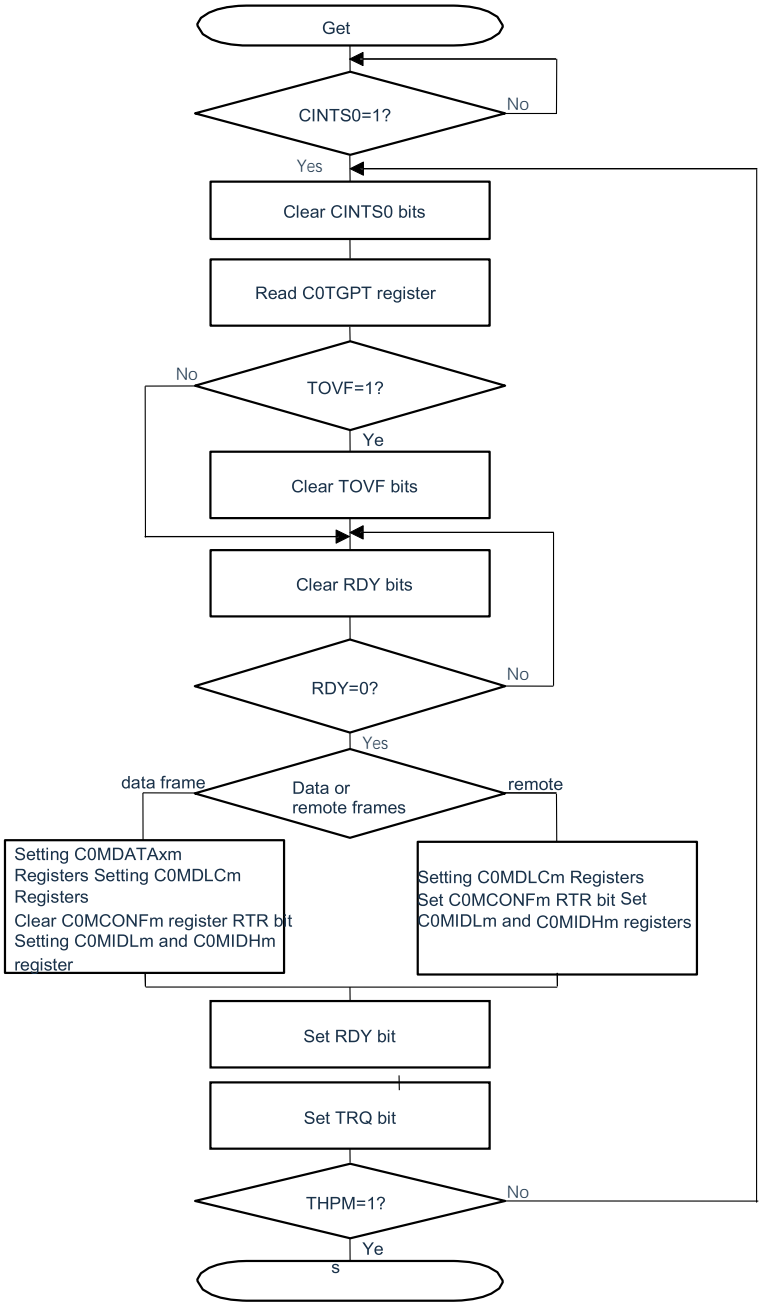


Note 1. The TRQ bit is set after the RDY bit is set.
2. RDY bit and TRQ bit cannot be set simultaneously.

Note 1. Check the MBON flag at the beginning and end of the interrupt to check access to the message buffer and the TX history list register to prevent pending sleep patterns. If MBON is detected to be cleared, the results and reprocessing operations must be discarded after MBON is set again. It is recommend to remove some sleep mode requests before handle tx interrupts.

2. If TOVF is set, the Transmission History list is inconsistent. Consider scanning all configured transmission buffers to complete the transfer.

Figure 21-72. Polling transmission over software



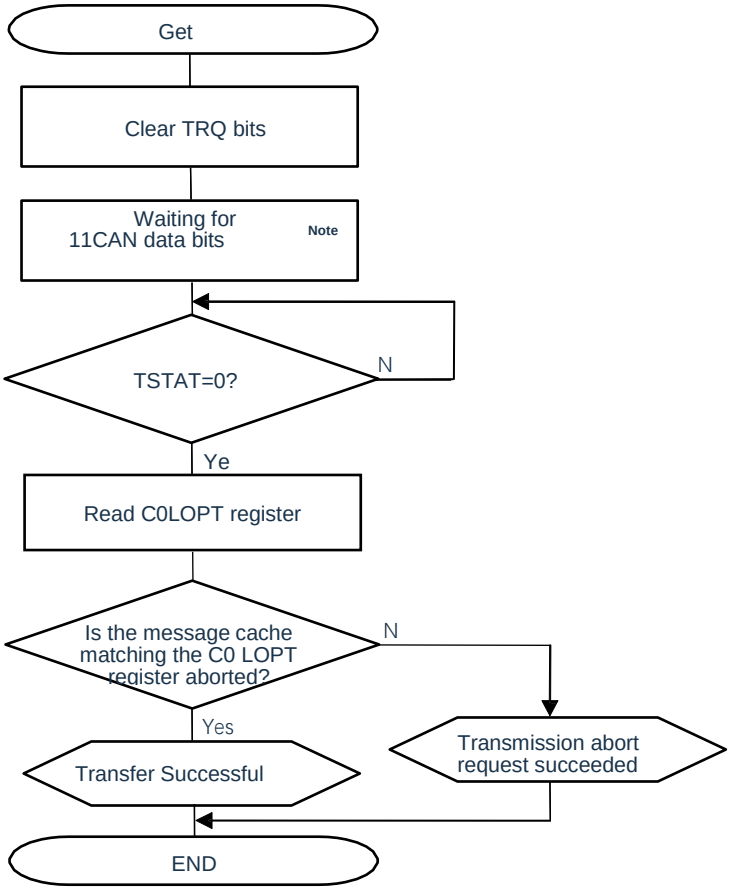
Note 1. The TRQ bit is set after the RDY bit is set.

2. RDY bit and TRQ bit cannot be set simultaneously.

Note 1. Check the MBON flag at the beginning and end of the interrupt to check access to the message buffer and the TX history list register to prevent pending sleep patterns. If MBON is detected to be cleared, the results and reprocessing operations must be discarded after MBON is set again.

2. If TOVF is set, the Transmission History list is inconsistent. Consider scanning all configured transmission buffers to complete the transfer.

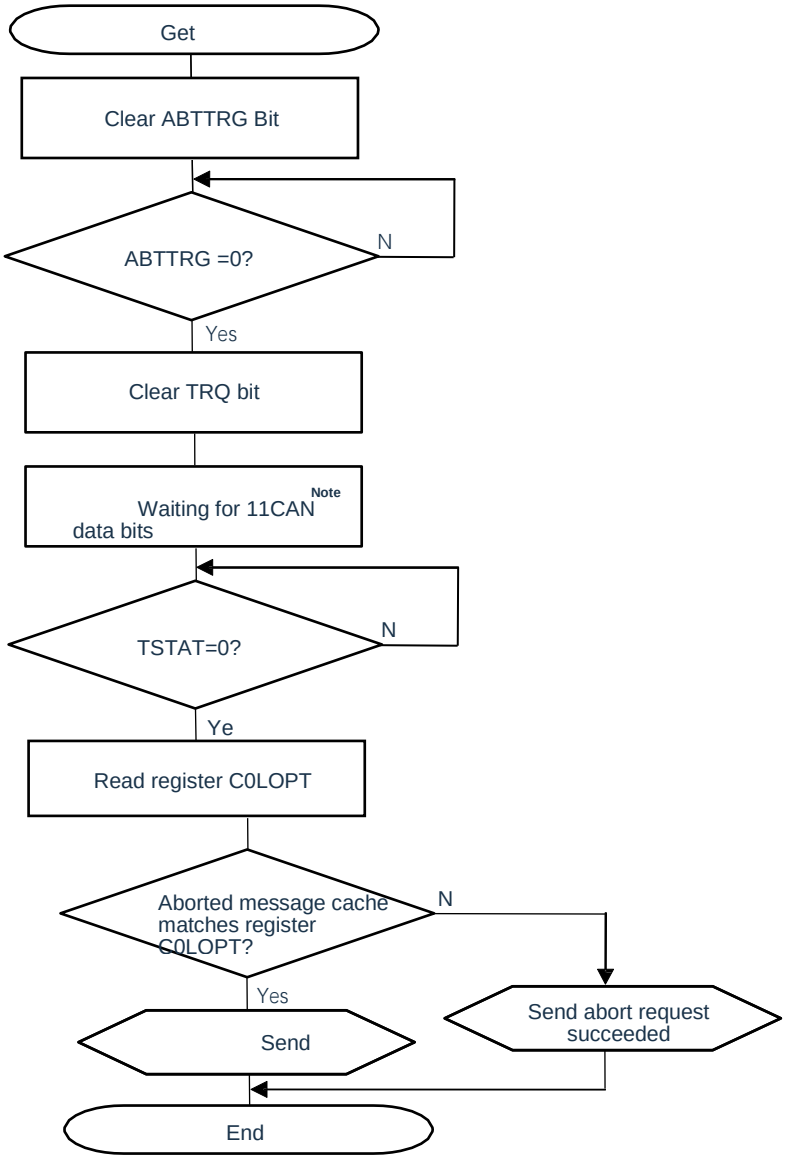
Figure 21-73. Transmission abort process (except for normal operation mode with ABT)



Note Even if the TRQ bit is cleared, it is possible to initiate the transmission without stopping, because the transmission request to the protocol layer may have been accepted between 11 bits, interframe space (3 bits), and suspended transmission (8 bits).

Note: 1. The transmission request abort process is performed by clearing the TRQ bit, not the RDY bit.
2. Before issuing a sleep mode conversion request, verify that there are no remaining transmission requests using this process.
3. The user application can periodically check the TSTAT bit, or it can check after the transfer is interrupted.
4. During the transmission abort process, do not execute new transmission requests, including in other message buffers.
5. When the transmission continues in the same message cache or only one message cache is used, it is possible to judge whether the transmission abort request is successful or not, which may cause a conflict. In this case, the historical information indicated by the C0TGPT register, etc., is used to determine.

Figure 21-74. Send abort handling (except for ABT transmission) (normal operation companion ABT)

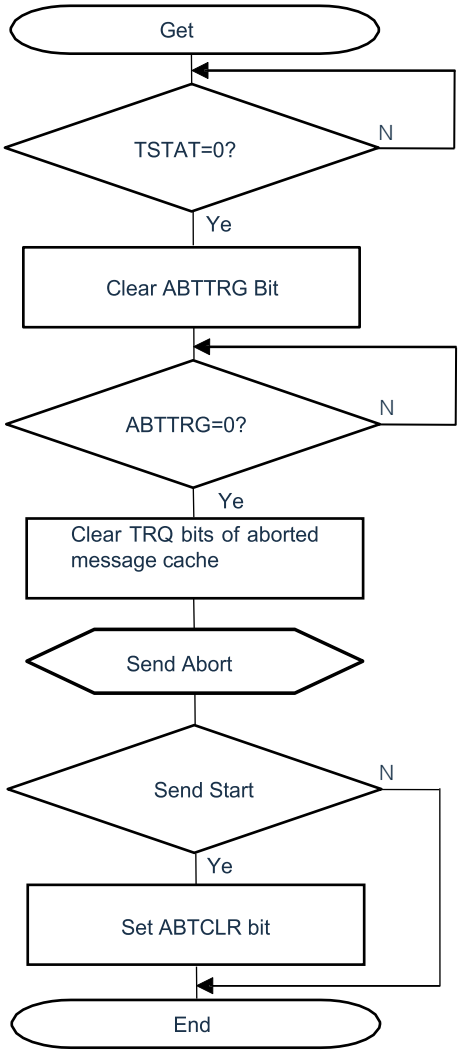


Note Even if the TRQ bits are cleared, it is possible to initiate the send without aborting, because the transfer request to the protocol layer may have been accepted between 11 bits, interframe space (3 bits), and suspended transmission (8 bits).

Note: 1. The transmission request abort process is performed by clearing the TRQ bit, not the RDY bit. .
2. Before issuing a sleep mode conversion request, verify that no transfer requests use this process.
3. The user application can periodically check the TSTAT bit, or it can check after the transfer is interrupted. .
4. During the transmission abort process, do not execute new transmission requests, including in other message buffers. .
5 When the transmission of the same message buffer is continuous or only one message buffer is used, it may lead to conflict to judge whether the transmission abort request is successful. In this case, the history information indicated by the C0TGPT register is used for judgment. .

Figure 21-75 show that when transmission ABT message cache is aborted, send message stops without skipping the processing of recovery

Figure 21-75. ABT send abort process (normal operation mode with ABT)

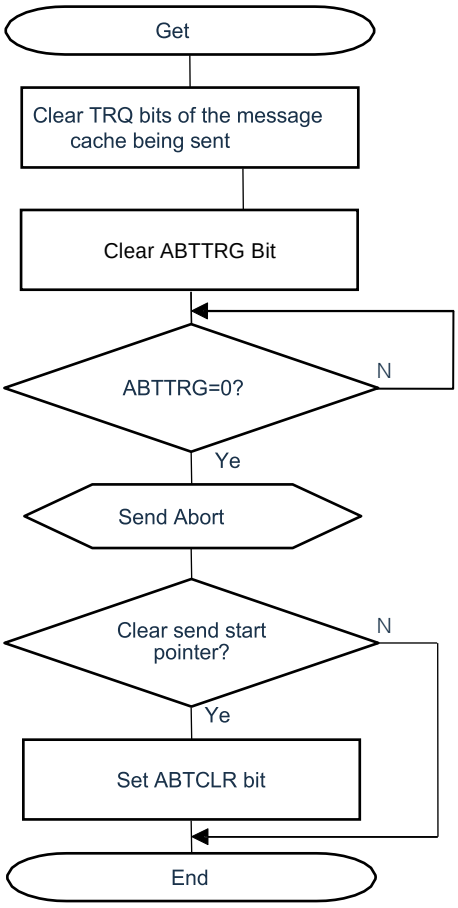


Note: 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

2. A CAN sleep mode/CAN stop mode conversion request is issued after the ABTTRG bit is cleared (after the ABT mode is discontinued) following the procedure shown in FIGS. 21-75 or 14-76. When purging a transmission request in an area other than the ABT area, follow the procedure shown in Figure 21-74. .

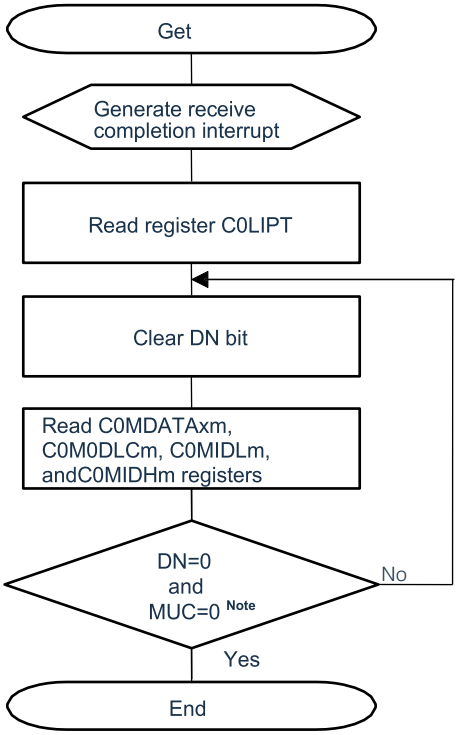
Figure 21-76 show that when transmission ABT message cache abort, send message stops skipping recovery processing. .

Figure 21-76. ABT send request abort process (normal mode of operation with ABT)



Note: 1. Do not set any transmission requests while the ABT transmission abort process is in progress.
2. Issue a CAN sleep mode/CAN stop mode request after the ABTTRG is cleared (stopped in ABT mode), following the procedure shown in FIGS. 21-75 or 21-76. When purging a transmission request in an area other than the ABT area, follow the procedure shown in Figure 21-74. .

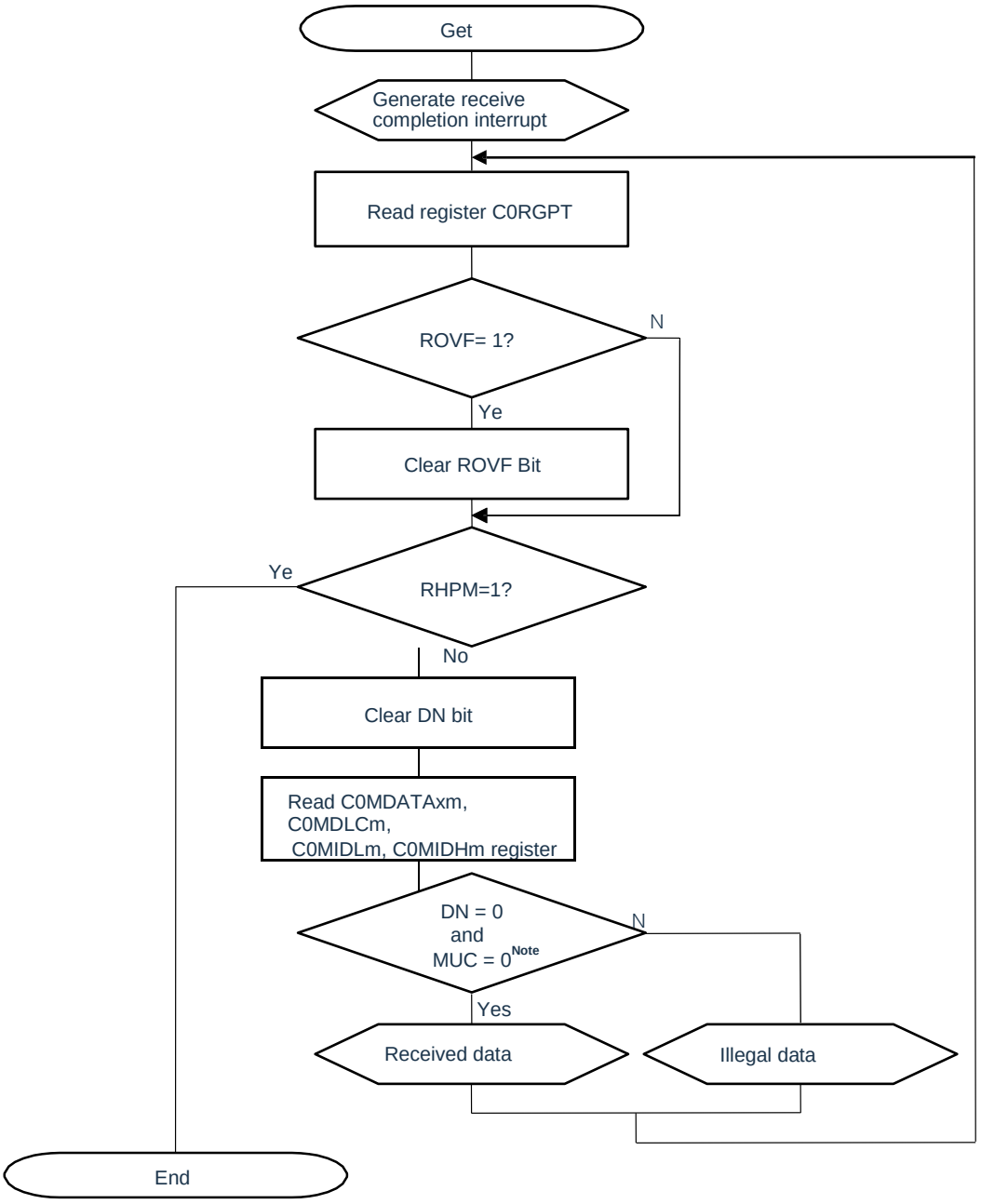
Figure 21-77. Receiving by interrupt (using register C0LIPT)



Note checks MUC and DN bits using read

Note: Check the MBON flag at the beginning and end of the interrupt to check access to the message buffer and the receive history list register to prevent execution of the suspended sleep mode. If it is detected that MBON is cleared, the operation and results of the process must be discarded before the MBON is set again. Before handling RX interrupts, it is recommended to cancel any sleep mode requests.

Figure 21-78. Receiving by interrupt (using register C0RGPT)

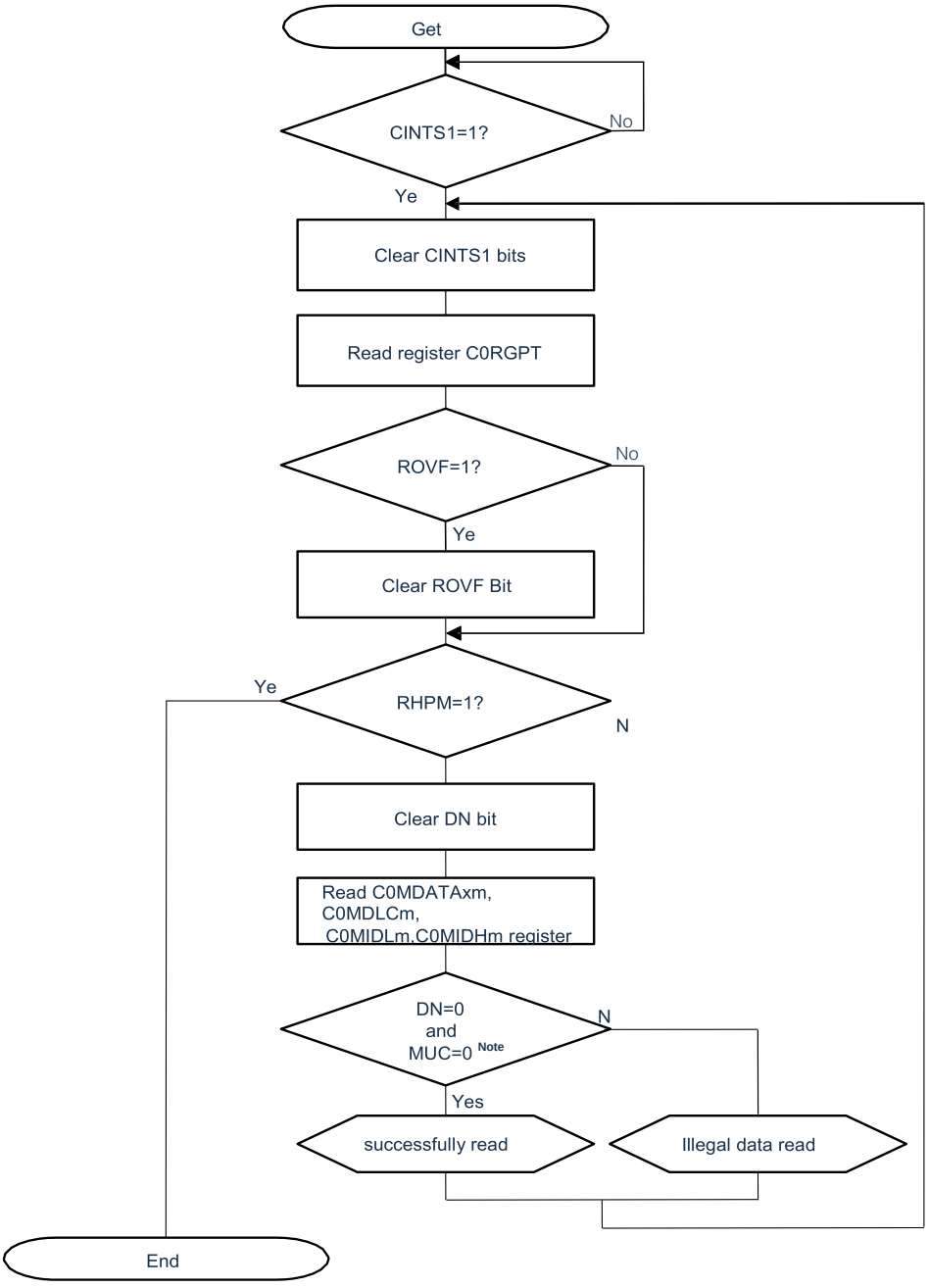


Note checks the MUC and DN bits using reading.

Remarks 1. Check the MBON flag at the beginning and end of the interrupt to check access to the message buffer and the receive history list register to prevent execution of the suspended sleep mode. If it is detected that MBON is cleared, the operation and results of the process must be discarded before the MBON is set again. Before handling RX interrupts, it is recommended to cancel any sleep mode requests.

3. If ROVF is set, the receive history list is inconsistent. Consider scanning all configured receive buffers for receiving.

Figure 21-79. Receive by software polling

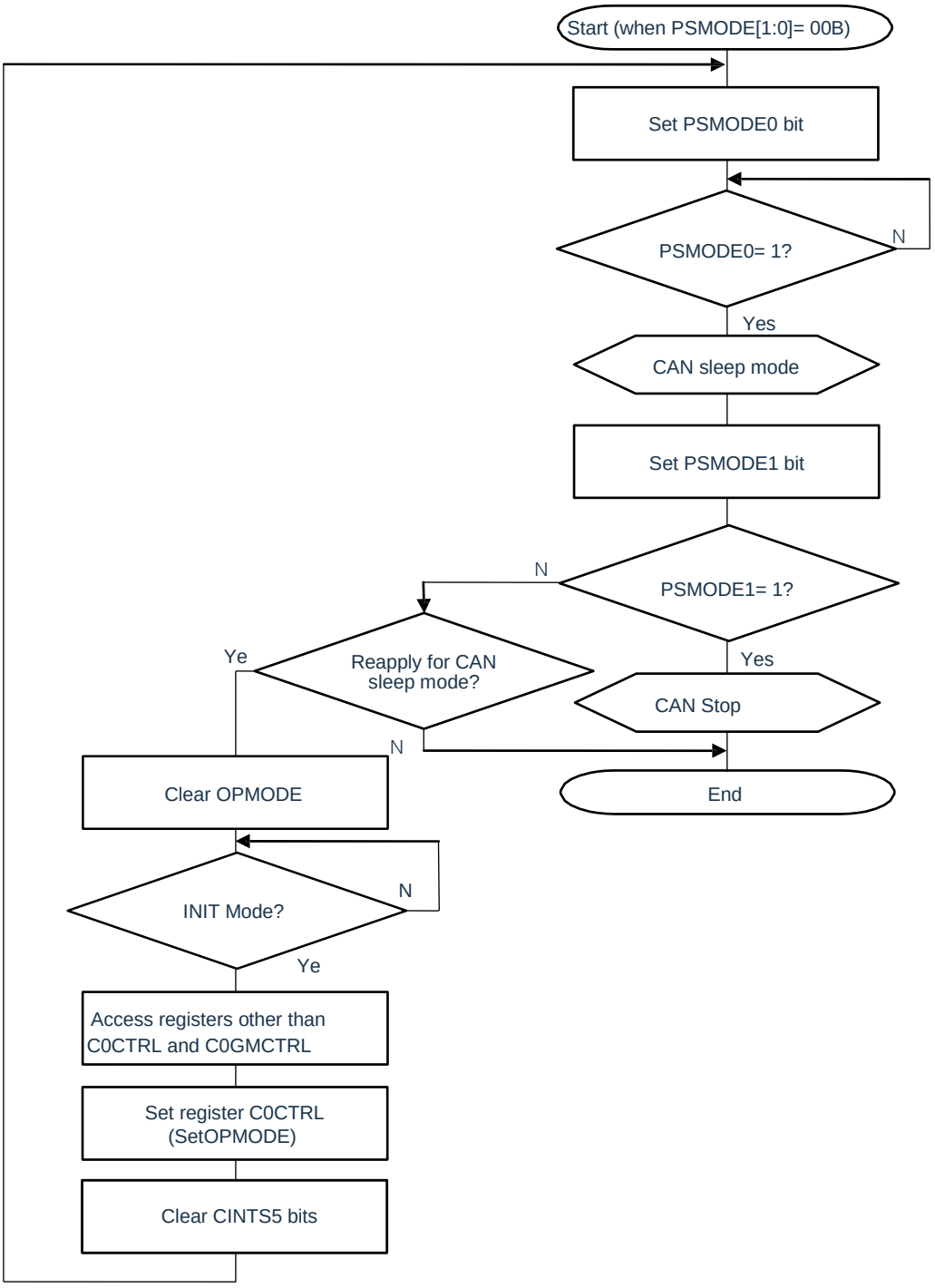


Note checks the MUC and DN bits using reading.

Remarks1. Check the MBON flags for polling start and end to check access to message buffers and receive history list registers to prevent pending sleep patterns from being executed. If MBON is detected to be cleared, the operation and results of the process must be discarded and processed after MBON is set again.

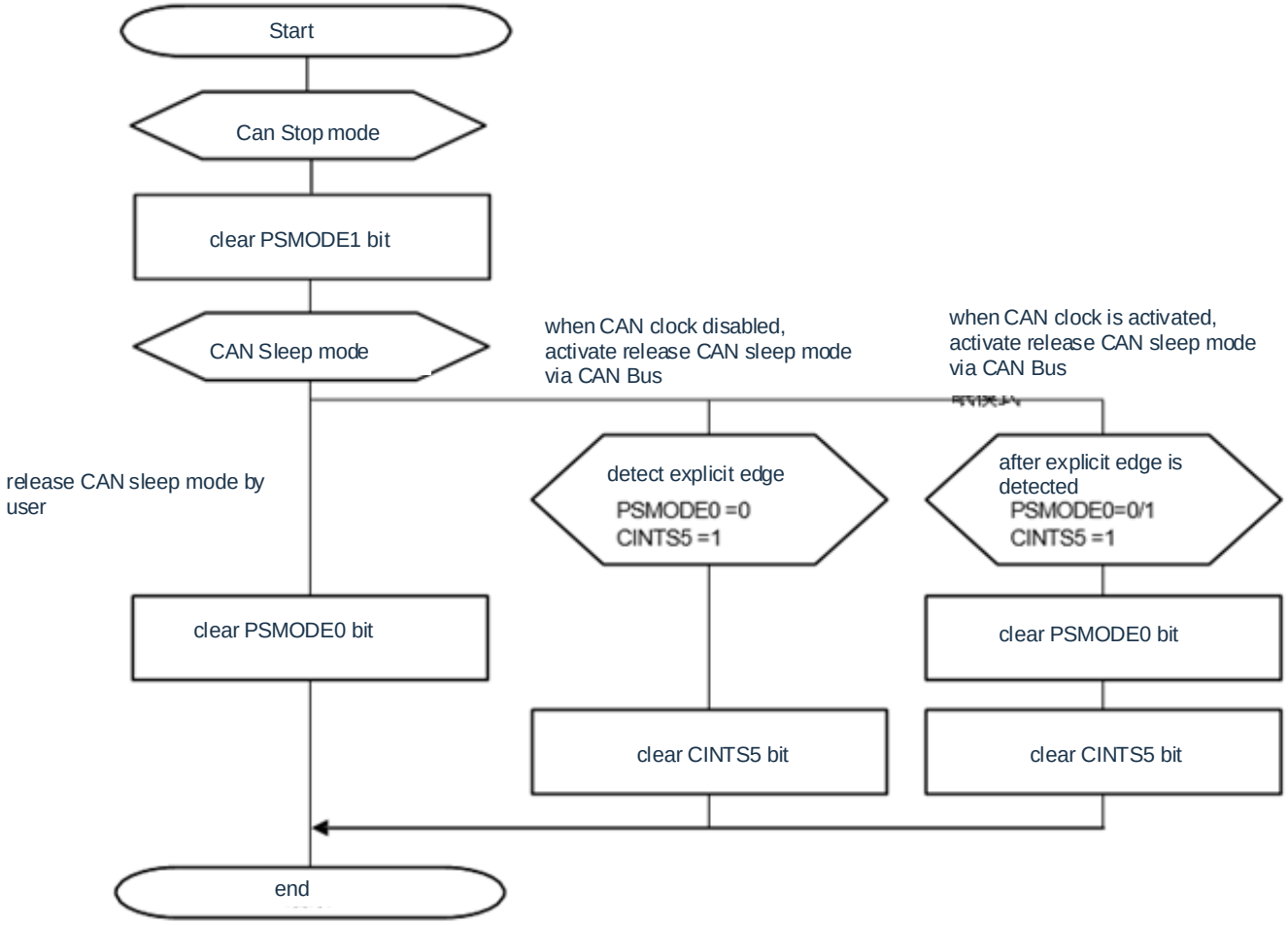
2. If ROVF is set, the receive history list is inconsistent. Consider scanning all configured receive buffers for receiving.

Figure 21-80. Setting CAN Sleep/Stop Mode



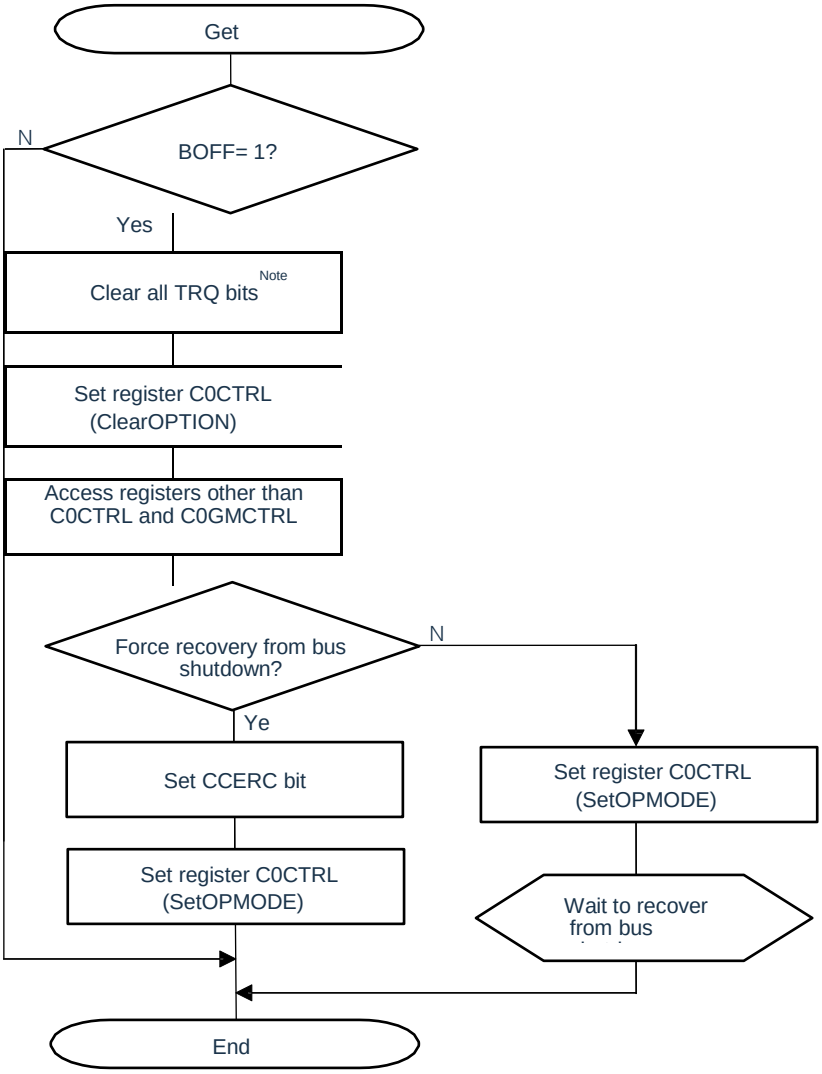
Note: To abort transmission before requesting CAN sleep mode, perform according to FIGS. 21-73 or 21-74.

Figure 21-81. Clear CAN sleep/stop mode



Note "CAN Clock Off": The CAN module clock is turned off and the CAN module is in sleep mode by CPU standby mode.

Figure 21-82. Bus Shutdown Recovery (Normal Operation Mode with ABT)

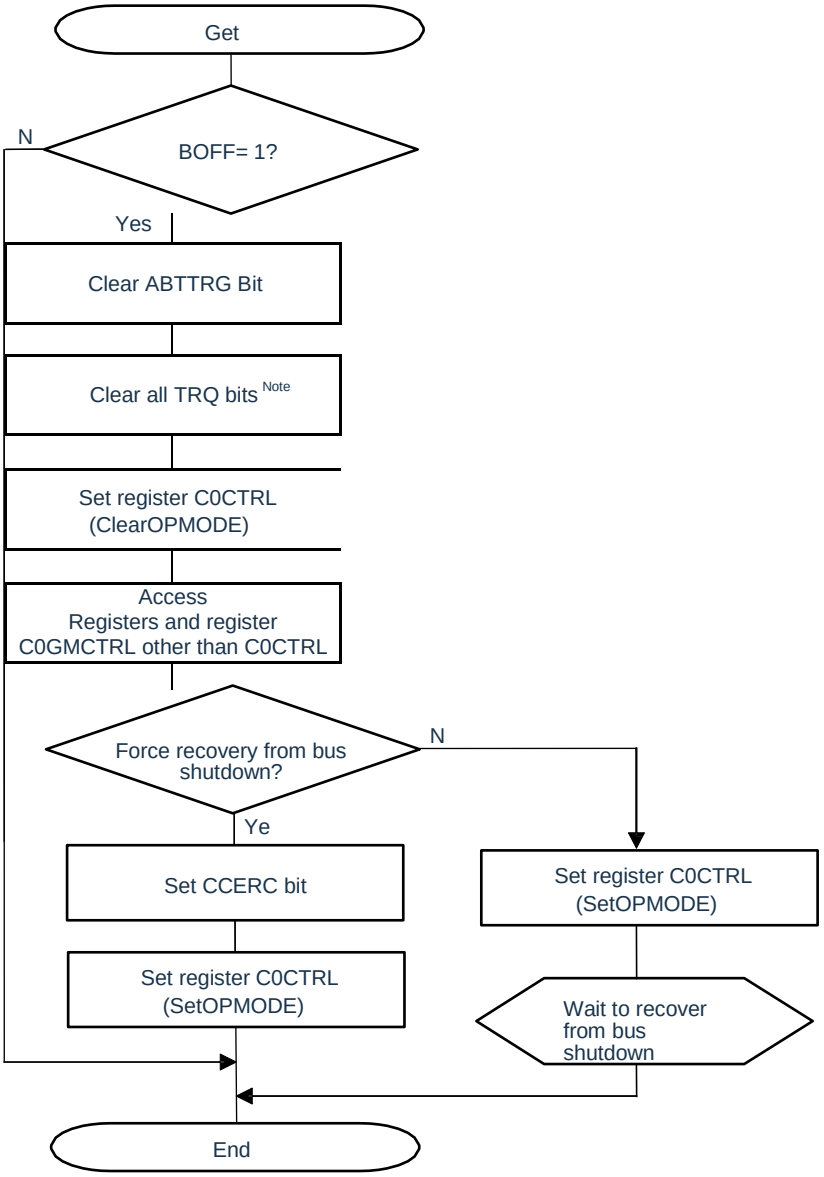


Note Even if the TRQ bit is cleared, it is possible to start a transfer without being aborted because a transfer request to the protocol layer may have been accepted between 11 bits, interframe space (3 bits), and pending transfers (8 bits).. .

Note: The receive error counter is cleared when a request is made to switch from the initialization mode to any other mode of operation to perform the bus-off recovery sequence again in the bus-off recovery sequence. Therefore, it is necessary to detect 11 consecutive hidden bits on the bus again for 128 times. .

Note OPMODE: normal operation mode, normal operation mode with ABT, receive mode only, single-shot mode, self-test mode

Figure 21-83. Bus Shutdown Recovery (Normal Operation Mode with ABT)



Note: When re-initialization of the message buffer is performed by clearing the RDY bits prior to starting the bus shutdown recovery sequence, all TRQ bits are cleared.

Note: The receive error counter is cleared when a request is made to switch from an initialization mode to another operation mode to perform a bus-off recovery sequence again in the bus-off recovery sequence. Therefore, it is necessary to detect 11 consecutive hidden bits on the bus 128 times again. .

Note OPMODE: normal operation mode, normal operation mode with ABT, receive mode only, single-shot mode, self-test mode

Figure 21-84. Normal shutdown handling

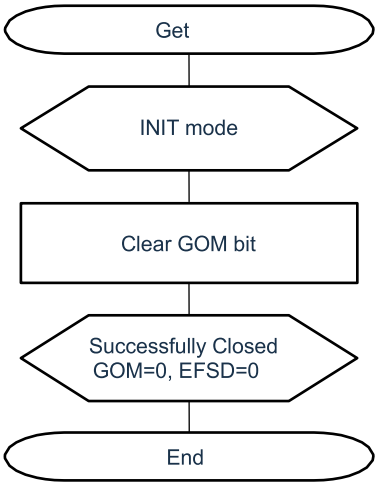
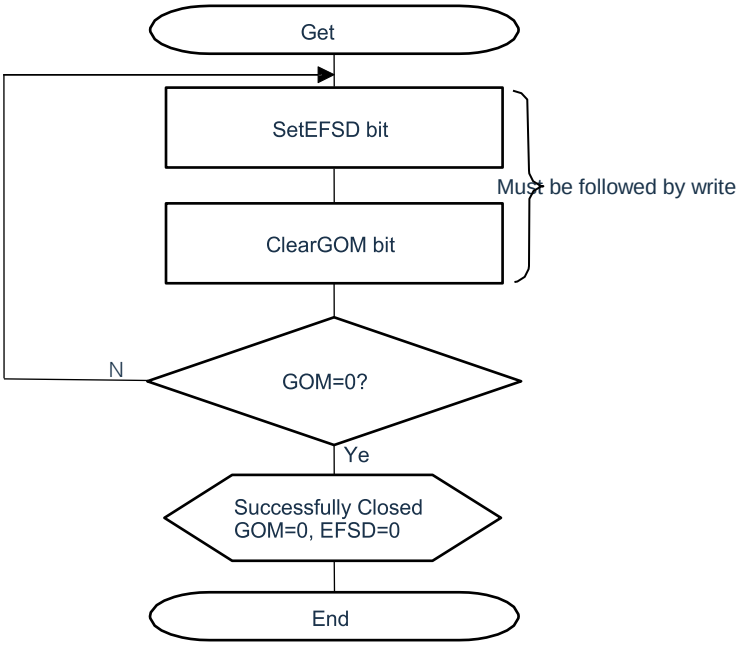


Figure 21-85. Forced shutdown processing



Note: Do not read or write any registers through software between setting the EFSD bit and clearing the GOM bit.

Note that if an interrupt or DMA occurs, it is not considered sequential access and the forced shutdown request is not valid.

Figure 21-86. Error Handling

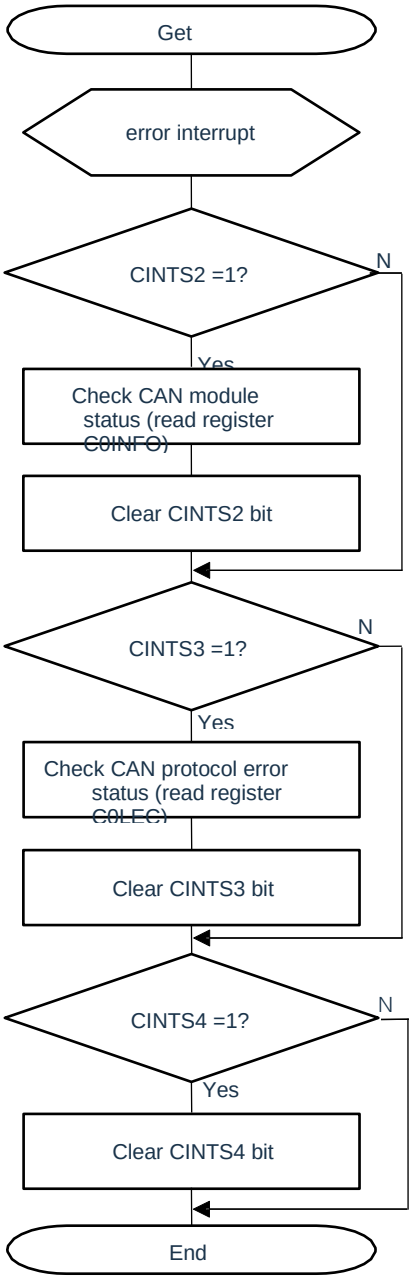
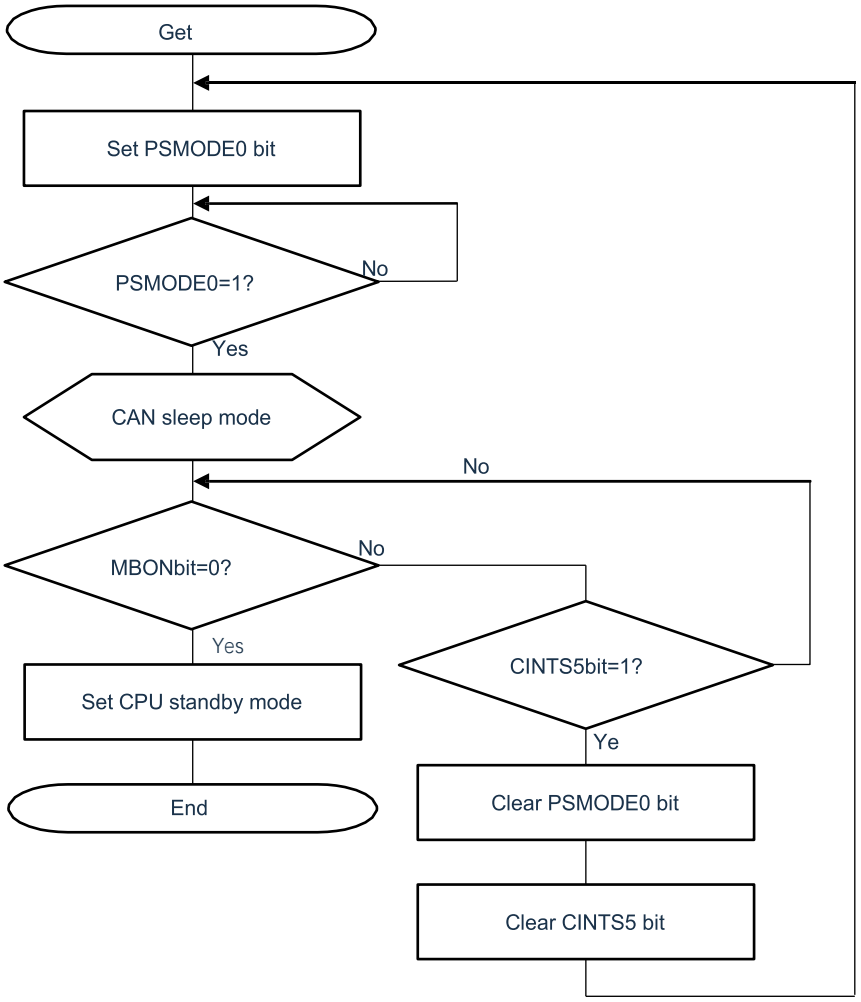
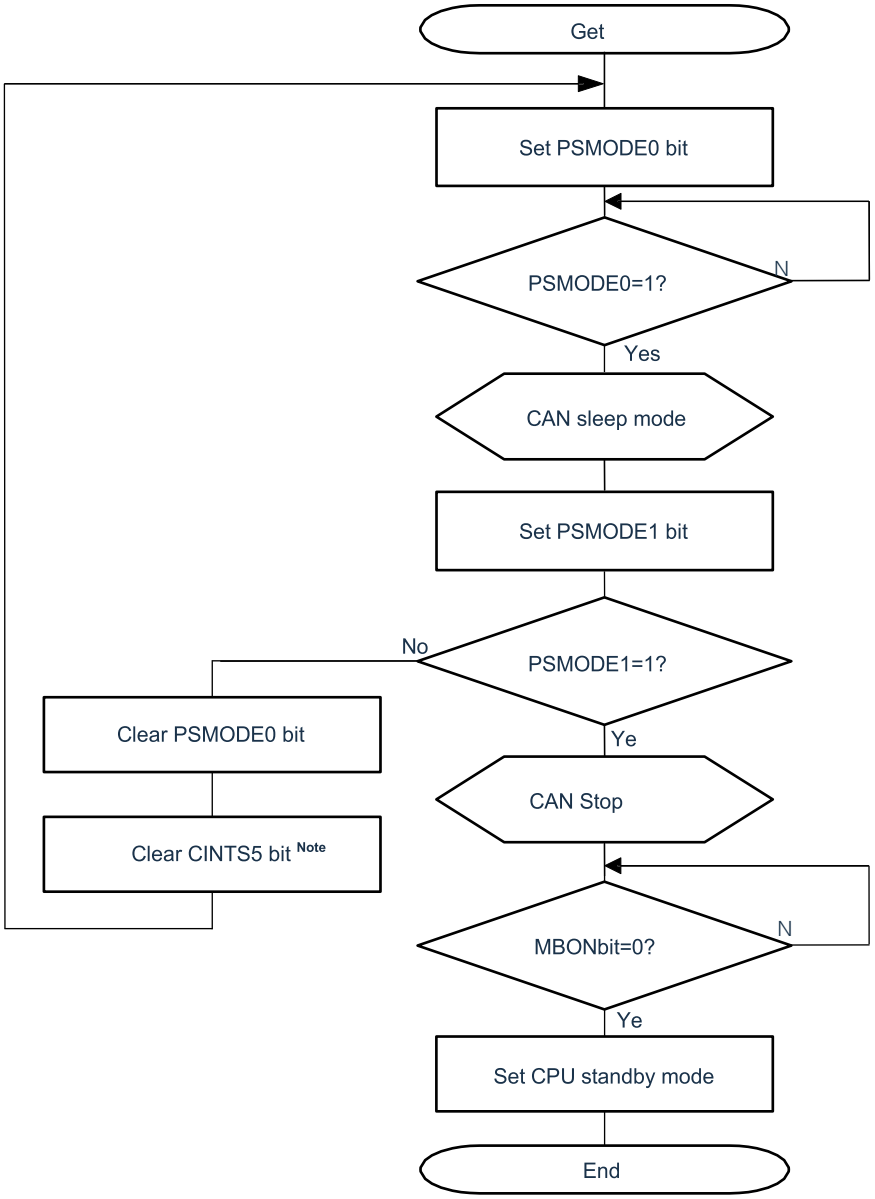


Figure 21-87. Setting CPU Standby (from CAN sleep mode)



Note: Before the CPU is set to enter CPU standby mode, check whether the CAN is in sleep mode. When the CAN sleep mode is detected, until the CPU is set to a standby mode, the CAN sleep mode may be canceled by the wake-up of the CAN bus.

Figure 21-88. Setting CPU Standby (from CAN Stop Mode)



Note in interrupt wake-up

Note: CAN stop mode can only be released by writing 01B to the PSMODE[1:0] bit of the C0CTRL register, not by changing the CAN bus state. .

Chapter 22 IrDA

The IrDA implements the transmission and reception of IrDA communication waveforms in accordance with IrDA (InfraredDataAssociation) 1.0 by cooperating with SCI.

22.1 Features of IrDA

If the IrDA function is set to be valid by the IRE bit of the IRCR register, TxD2 and RxD2 signals of the SCI can encode or decode the IrDA1.0 protocol waveform (IrTxD/IrRxD pin).

In systems that support the IrDA1.0 protocol, the transfer rate can be changed as needed after communication is initiated at a transfer rate of 9600 bps. IrDA does not have the built-in feature to automatically change the transfer rate, so you must change the settings by software to change the transfer rate.

The following baud rates can be set when selecting high-speed internal oscillators (fIH=24,12,6,3MHz).

· 115.2kbps, 57.6kbps, 38.4 kbps, 19.2 kbps, 9600bps,2400bps

A schematic diagram of the collaboration between IrDA and SCI is shown in Figure 22-1.

Figure 22-1 Schematic diagram of IrDA's partnership with SCI

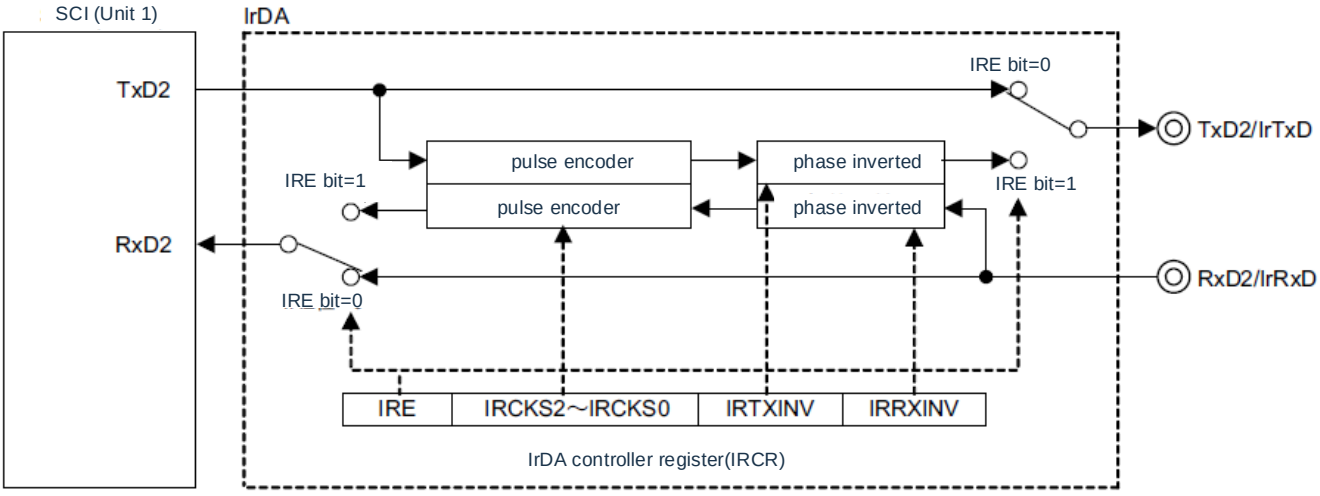


Table 22-1 Pin structure of IrDA

Pin Name	Input/Output	Features
IrTxD	output	Output pin for sending data
IrRxD	Input	Input pin for receiving data

22.2 Register controlling IrDA

Control the IrDA functionality through the following registers.

- Peripheral Enable Register 0 (PER0s).
- IrDA control register (IRCR)

22.2.1 Peripheral Enable Register 0 (PER0s).

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.

Bit6 (IRDAEN) must be set to'1' when using IrDA.

The PER0 register is set by an 8-bit memory operation instruction.

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

Figure 22-2 Format of Peripheral Enable Register 0 (PER0)

Location: 40020420H. After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDEN	ADCEN	ICAHN	SCI1EN	SCI0EN	CAN0	TM40EN

IRDEN	Control of input clock of IrDA
0	Stop provide an input clock. • Cannot write the SFR used by IrDA. · IrDA is in a reset state.
1	Provides an input clock. • Can read and write SFRs used by IrDA.

Note 1. When you set up IrDA, you must first set the IRDAEN location "1". When the IRDAEN bit is '0', the write operation of the control register of the IrDA is ignored, and the read values are all initial.

22.2.2 IrDA control register (IRCR)

This is the register that controls the IrDA function. Polarity switching between the received and transmitted data, clock selection for IrDA, and serial input/output

Output pin functionality (usual serial and IrDA functionality) switching selection. The IRCR register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 22-3 Format of IrDA Control Register (IRCR)

Address: 40044000H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
IRCR	HEAT	IRCKS2	IRCKS1	IRCKS0	IRTXINA	IRR	0	0

HEAT	Allow for IrDA
0	Serial input/output pins are used as the usual serial function
1	Serial input/output pins are used as IrDA functionality

IRCKS2	IRCKS1	IRCKS0	Clock Selection for IrDA
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 from setting

IRTXINA	Polarity Switching of IrTxD Data
0	IrTxD output of sent data
1	IrTxD output after reversing transmit data

IRR	Polarity Switching of IrRxD Data
0	Use input data for IrRxD pins as received data
1	The data after the input data of the inverted IrRxD pin is used as the receiving data

Note 1. bit1 and bit0 must be set to 0.

2. The IRCKS[2:0], IRTXINV, and IRRXINV bits can only be set if the IRE bit is '0'.

22.3 Running of IrDA

22.3.1 Procedures for IrDA Communication

(1) Initial set-up process for IrDA communications

Follow these steps to initially set up IrDA.

1. Position "1" for IRDAEN of PER0 register.
2. Sets the IRCR register.
3. Sets the relevant register for the SCI (referring to the UART mode set-up step).

(2) Stop flow for IrDA traffic

1. Set the IrTxD pin status after IrDA communication is stopped by setting the port register and the port mode register.

Remark When IrDA reset is performed by step 3, the IrTxD pin may change the output state by switching to the data output of the usual serial interface UART.

- Output low level case from IrTxD pin
  Place the port register "0". Immediately after this setting, the IrTxD pin is fixed at a low level.
- Output a high level condition from the IrTxD pin
  Place the port register "1". With this setting, the IrTxD pin is immediately fixed to a high level after the IrDA reset in step 3.
- Setting the port mode register "1" with the IrTxD pin in the Hi-Z state.
  Immediately after this setting, the IrTxD pin changes to the Hi-Z state.

2. Stop the STm0 bit and STm1 position "1" of the STm register (SCI-dependent register) for the channel.
3. The PER0 register's IRDAEN position is "0" for IrDA reset.

The STm0 bit and STm1 position of the STm register cannot be '1' or the IRE position of the IrDA '0'.

(3) Steps when sending IrDA frame errors

When a frame error occurs during IrDA communication, the following settings must be made in order to be set to a state where subsequent data can be received.

1. The STm1 position of the STm register of the SCI is "1" (stopping the operation of channel 1).
2. The SSm1 position of the SCI's SSm register is "1" (start running channel 1 of SCI).

Remark m: Cell Number (m=0)

For frame error handling of SCI, refer to Chapter 19 “Universal Serial Communication Unit."

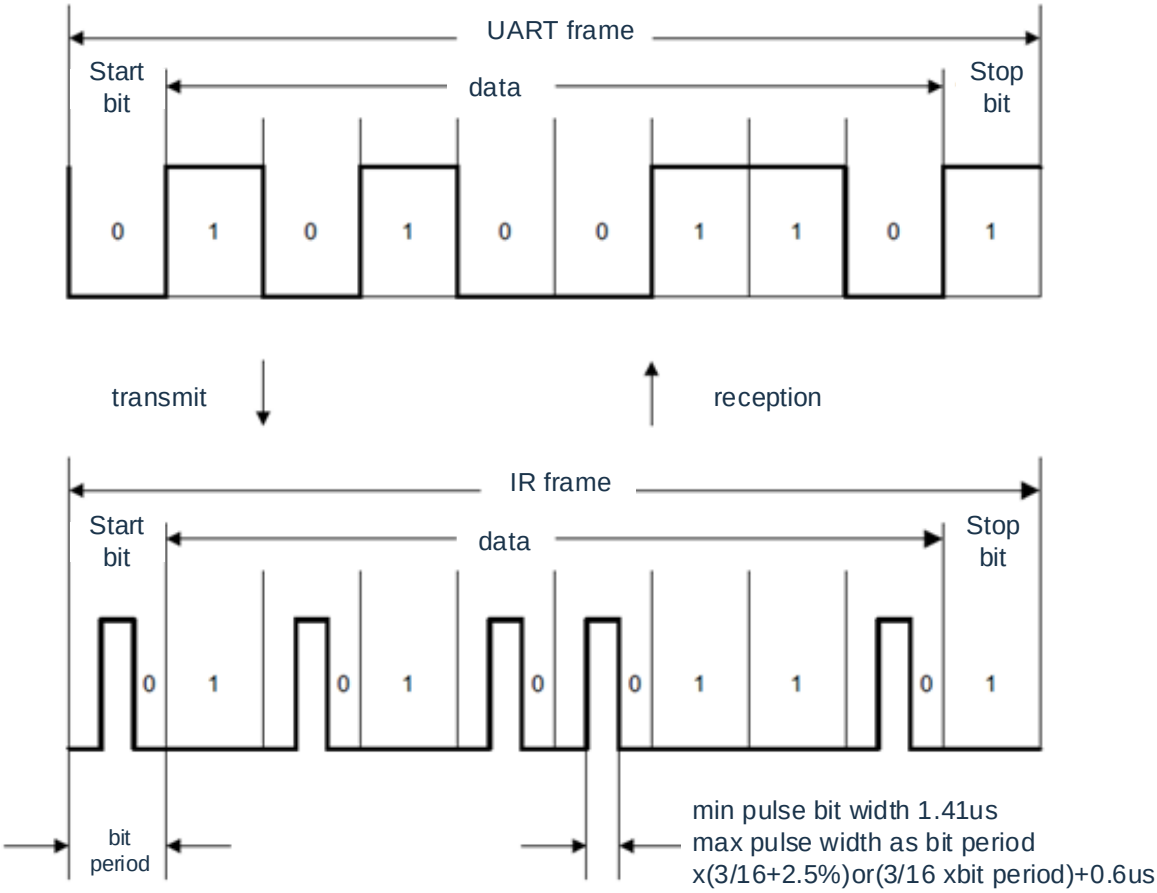
22.3.2 Send

At transmission, the output signal (UART frames) from the SCI is converted to IR frames via IrDA (reference FIGS. 22-4).

When the IRTXINV bit is "0" and the serial data is "0", high-level pulses (initial values) of 3/16 bit period are output. In addition, that high level pulse width may be changed accord to the set value of the IRCKS2 to IRCKS0 bit. According to the standard, the high level pulse width is specified as a minimum of 1.41uS and a maximum of (3/16+2.5%)x bit period or (3/16x bit period)+0.6us.

When the CPU or peripheral hardware clock (fCLK) is 24 MHz, the minimum high level pulse width is 1.5ss. In addition, no pulses are output when the serial data is "1".

Figure 22-4 Send/Receive Run Graph for IrDA



22.3.3 Receive

When receiving, the data of IR frame is converted to UART frame through IrDA, then input to SCI. Low level data is output when the IRRXINV bit is '0' and a high level pulse is detected. If no pulse is present in a 1-bit period, high level data is output. It must be note that pulses less than that minimum pulse width of 1. 41 s cannot be identified.

22.3.4 Selection of high level pulse width

If the pulse width at transmission is less than 3/16, the suitable setting (minimum pulse width) of IRCKS2~IRCKS0 bit and the setting high level pulse width are shown in Table 22.

Table 22-2 Set value for IRCKS2~IRCKS0 bit

fCLK [MHz]	Project	<Upper Segment> Bit Rate [kbps] <Lower Segment>Bit Rate 3/16 [μs]					
		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 [μs]	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 [μs]	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 [μs]	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 [μs]	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 [μs]	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 [μs]	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 [μs]	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 [μs]	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 [μs]	2.67	2.67	2.67	2.67	2.67	1.50

Note 1. "-" indicates that communication standards are not met.

2. The pulse width cannot be less than the bit rate 3/16.

22.4 Precautions when using IrDA

1. A runtime clock that allows or disables the provision of IrDA can be set through a peripheral admission register. The register cannot be accessed because the initial state is to disable clock provisioning. Prior to setting the register, it is necessary to set the state of the peripheral admission register to allow the provision of the IrDA running clock.
2. In sleep mode, the IrDA feature runs continuously.
3. Do not use SCI's initialization during IrDA communication (SS bit=1).
4. The IRRXINV bit, IRTXINV bit, and IRCKS[2:0] bit of the IRCR register can only be set if the IRE bit is '0'.

Chapter 23 Enhanced DMA

23.1 The Function of DMA

The DMA is a function of transferring data between memories without using a CPU. Starting DMA for data transfer is interrupted by peripheral functions. When DMA and CPU access the same unit in FLASH, SRAM0, SRAM1 or peripheral module simultaneously, their bus use rights are higher. When DMA and CPU access different units in FLASH, SRAM0, SRAM1 or peripheral module respectively, they do not interfere with each other.

The specifications of the DMA are shown in Table 23-1.

Table 23-1 Specifications for DMA (1/2)

Project		Specifications
Boot Source		Up to 37 boot sources
Distributable control data		Group 40
Transmissionable address space	address space	full address range space
	Source	Full address range space optional
	Objectives	Full address range space optional
Maximum Transfers	normal mode	65535 times
	repetition mode	65535 times
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
	repetition mode	65535 bytes
delivery unit		8-/16-32-bit
transfer mode	normal mode	Ends after the transfer of the DMACTj register from "1" to "0".
	repetition mode	After the transfer of the DMACTj register from "1" to "0", the address of the repeat region is initialized and the DMRLDj The value of the register reloads into the DMACTj register and continues to be transferred.
address control	normal mode	Fixed or incremental
	repetition mode	Fix or increment the address of the non-repeating area.
Priority of the start-up source		Refer to Table 23-5 DMA Boot Source and Vector Addresses."

Table 23-1 Specifications for DMA (2/2)

Project		Specifications
interrupt request	normal mode	When the DMACTj register is transferred from "1" to "0", the CPU is requested to start the source interrupt and interrupts.
	repetition mode	When the DMACRj register has RPTINT bit '1' allowing interrupt generation) and the DMACTj register changes.
Transfer Start		If the DMAENi register's DMAENi0~DMAENi7 position "1" (Enable start-up), data transfer is started each time DMA start-up source occurs.
Delivery Stop	normal mode	· Position DMAENi0~DMAENi7 "0" (Disable boot). · When the DMACTj register changes from "1" to "0" data transfer ends
	repetition mode	· Position DMAENi0~DMAENi7 "0" (Disable boot). · End of data transfer when the RPTINT bit is "1" (interrupts allowed) and the DMACTj

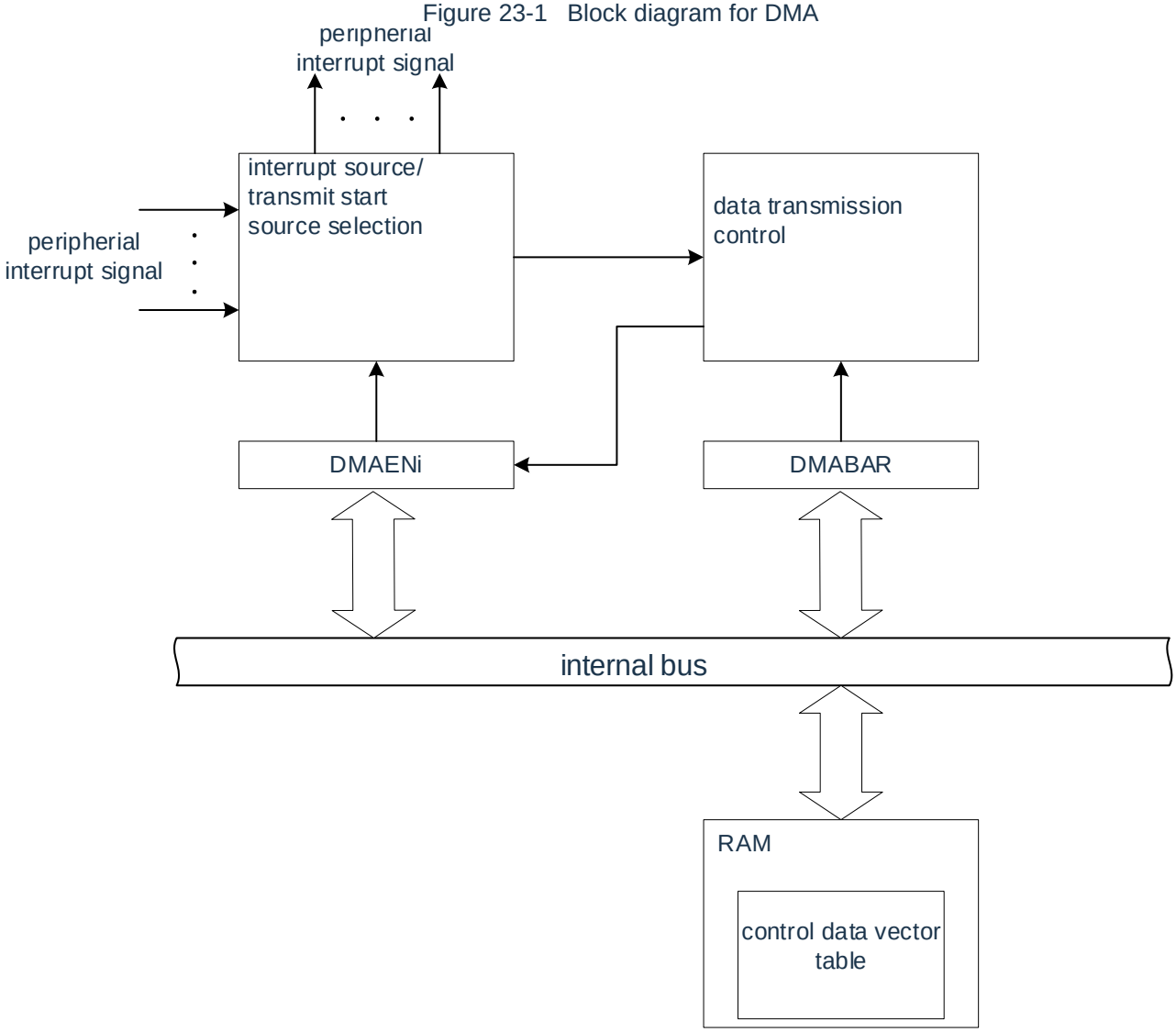
Note In deep sleep mode, flash memory cannot be used as a source for DMA transfer because it stops working.

Remark i=0~4, j=0~39

23.2 Structure of DMA

The block diagram for DMA is shown in Figure 23-1

Figure 23-1 Block diagram for DMA



23.3 Register for controlling DMA

The registers that control the DMA are shown in Table 23-2.

Table 23-2 Register for controlling DMA

register name	symbol
Peripheral Enable Register 1	PER1
DMA Boot Enable Register 0	DMAEN0
DMA Boot Enable Register 1	DMAEN1
DMA Boot Enable Register 2	DMAEN2
DMA Boot Enable Register 3	DMAEN3
DMA Boot Enable Register 4	DMAEN4
DMA base address register	DMABAR

The control data of the DMA is shown in Table 23-3.

The control data of the DMA is allocated in the DMA control data area of the RAM. The DMA control data area and the 704-byte area containing the DMA vector table area (the starting address of the stored control data) are set through the DMABAR register.

Table 23-3 Control data for DMA

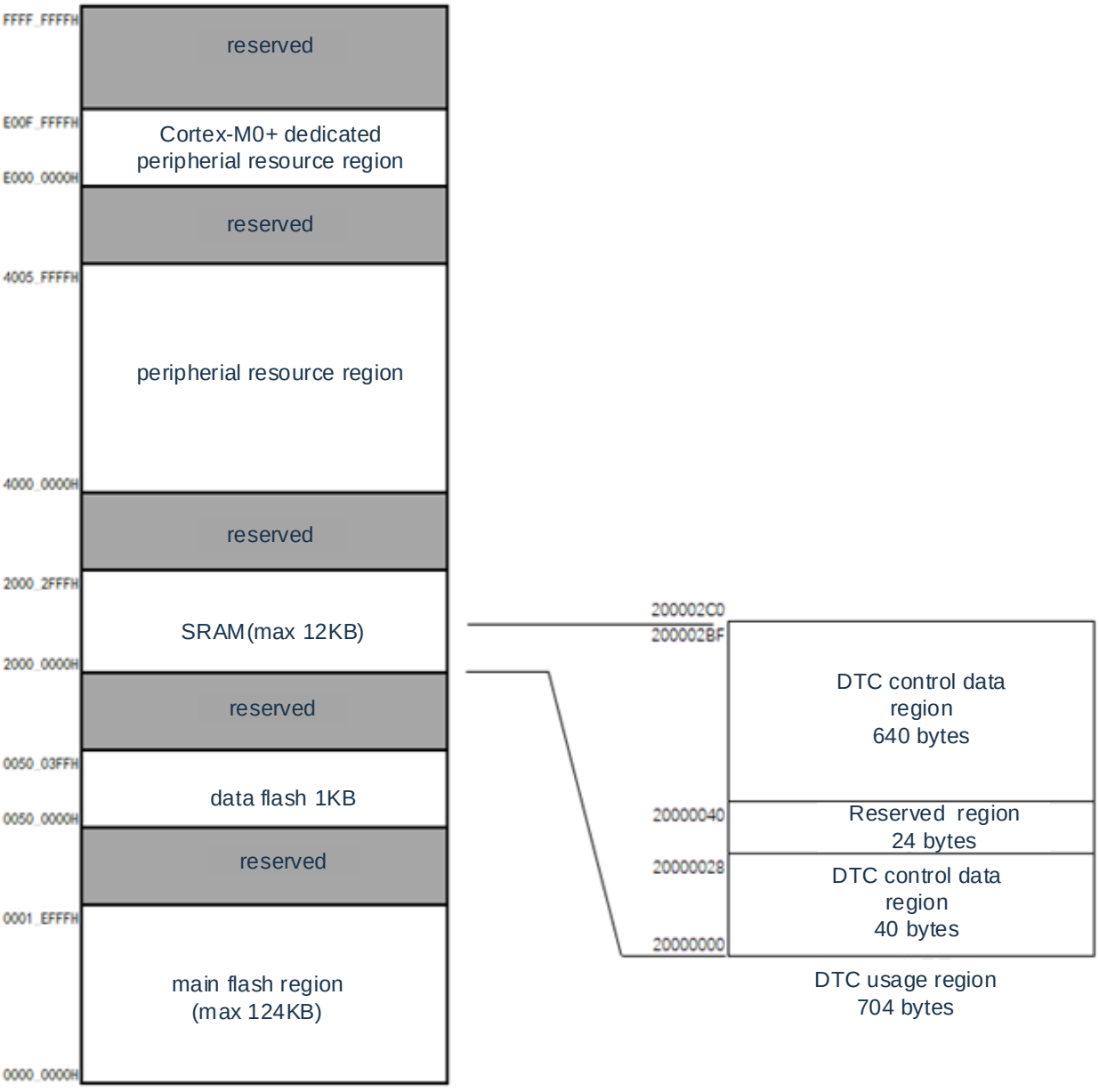
register name	symbol
DMA control register j	DMACRj
DMA block size register j	DMBLSj
DMA transfer number register j	DMACTj
DMA transfer times reload register j	DMRLDj
DMA source address register j	DMSARj
DMA destination address register j	DMDARj

Remark j=0~39

23.3.1 DMA controls the allocation of data areas and DMA vector table areas

A 704-byte region of the control data and the vector table assigned to the DMA is set to the RAM region through the DMABAR register.
An example of a memory image with the DMABAR register set to "20000000H" is shown in Figure 23-2.
Space not used by DMA in 640 bytes of the DMA control data area can be used as RAM.

Figure 23-2 An example of a memory image when the DMABAR register is set to "20000000H"



23.3.2 Controlling Data Allocation

Starting from the starting address, control data is allocated in the order of DMACRj, DMBLSj, DMACTj, DMRLDj, DMSARj, DMDARj registers.

The starting address is set by the DMABAR register, and the lower 10 bits are set by the vector tables assigned by each starting source.

The distribution of control data is shown in Figure 23-3.

Note 1. You must change the data for the DMACRj,DMBLSj,DMACTj, DMRLDj, DMSARj, DMDARj registers when the DMAENi0~DMAENi7 bit of the corresponding DMAENi (i=0~4) is "0".

2. Access to DMACRj, DMBLSj, DMACTj, DMRLDj, DMSARj, and DMDARj cannot be performed via DMA transfer.

Figure 23-3 Control allocation of data (DMABAR set to 2000000H)

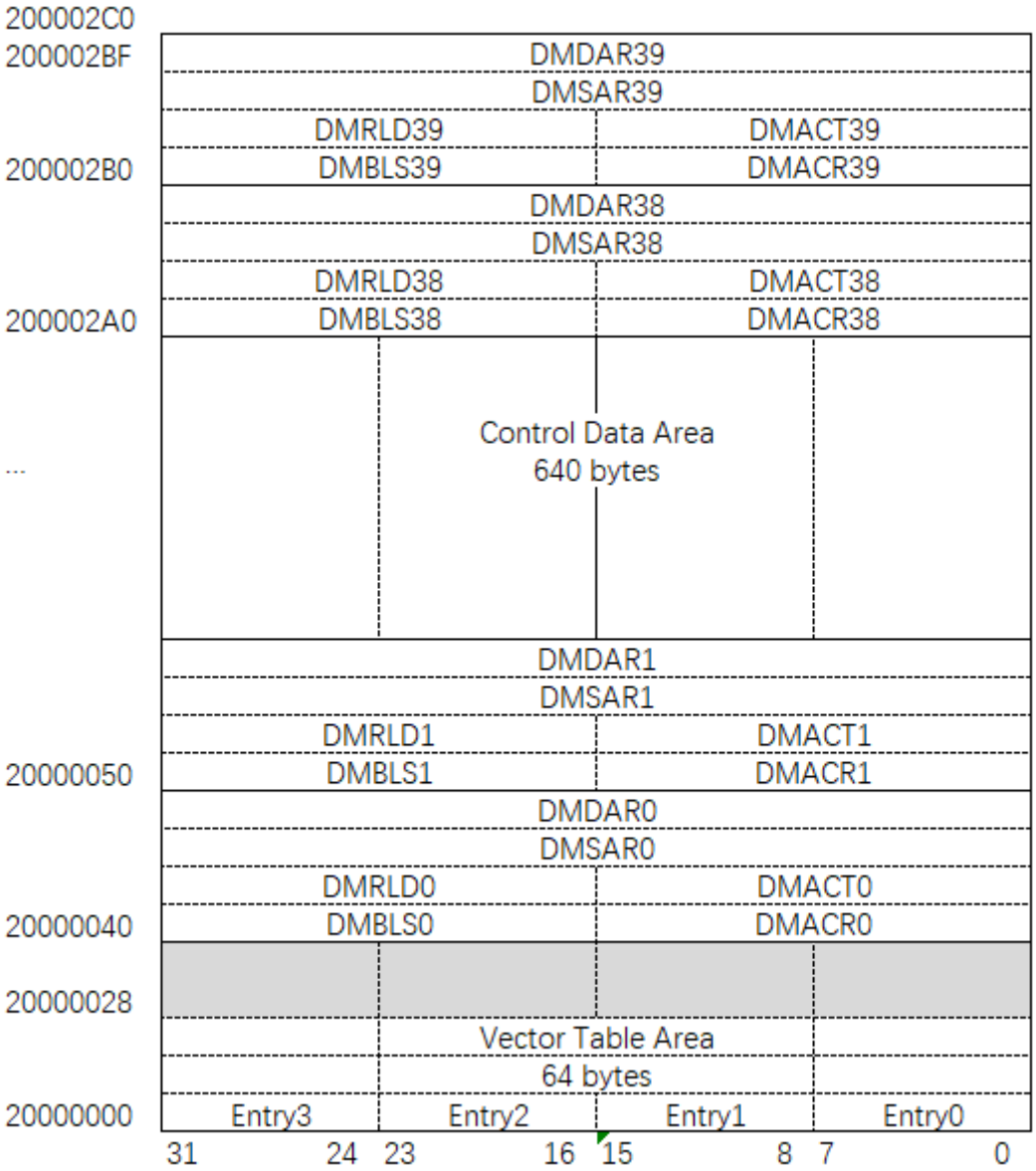


Table 23-4 The starting address of the control data

j	Address
19	baseaddr+170H
18	baseaddr+160H
17	baseaddr+150H
16	baseaddr+140H
15	baseaddr+130H
14	baseaddr+120H
13	baseaddr+110H
12	baseaddr+100H
11	baseaddr+F0H
10	baseaddr+E0H
9	baseaddr+D0H
8	baseaddr+C0H
7	baseaddr+B0H
6	baseaddr+A0H
5	baseaddr+90H
4	baseaddr+80H
3	baseaddr+70H
2	baseaddr+60H
1	baseaddr+50H
0	baseaddr+40H

j	Address
39	baseaddr+2B0H
38	baseaddr+2A0H
37	baseaddr+290H
36	baseaddr+280H
35	baseaddr+270H
34	baseaddr+260H
33	baseaddr+250H
32	baseaddr+240H
31	baseaddr+230H
30	baseaddr+220H
29	baseaddr+210H
28	baseaddr+200H
27	baseaddr+1F0H
26	baseaddr+1E0H
25	baseaddr+1D0H
24	baseaddr+1C0H
23	baseaddr+1B0H
22	baseaddr+1A0H
21	baseaddr+190H
20	baseaddr+180H

Remark baseaddr: Setting value for the DMABAR register

23.3.3 vector table

Once the DMA is started, control data assigned to the DMA control data area is read by data read from a vector table assigned to each start source.

The DMA boot source and vector addresses are shown in Table 23-5. The vector table of each starting source has 1 byte, which stores the data of "00H"~27H, and selects 10 sets of data from the 40 sets. The high 22 bits of the vector address are set by the DMABAR register, and the low 10 bits are assigned '00H' to '27H'.

Note The starting address of the DMA control data area set in the vector table must be changed if the DMAENi0~DMAENi7 bit of the corresponding DMAENi (i=0~4) register is '0'.

Figure 23-4 Starting address and vector table for control data
A case where the DMABAR register is set to "2000000H"

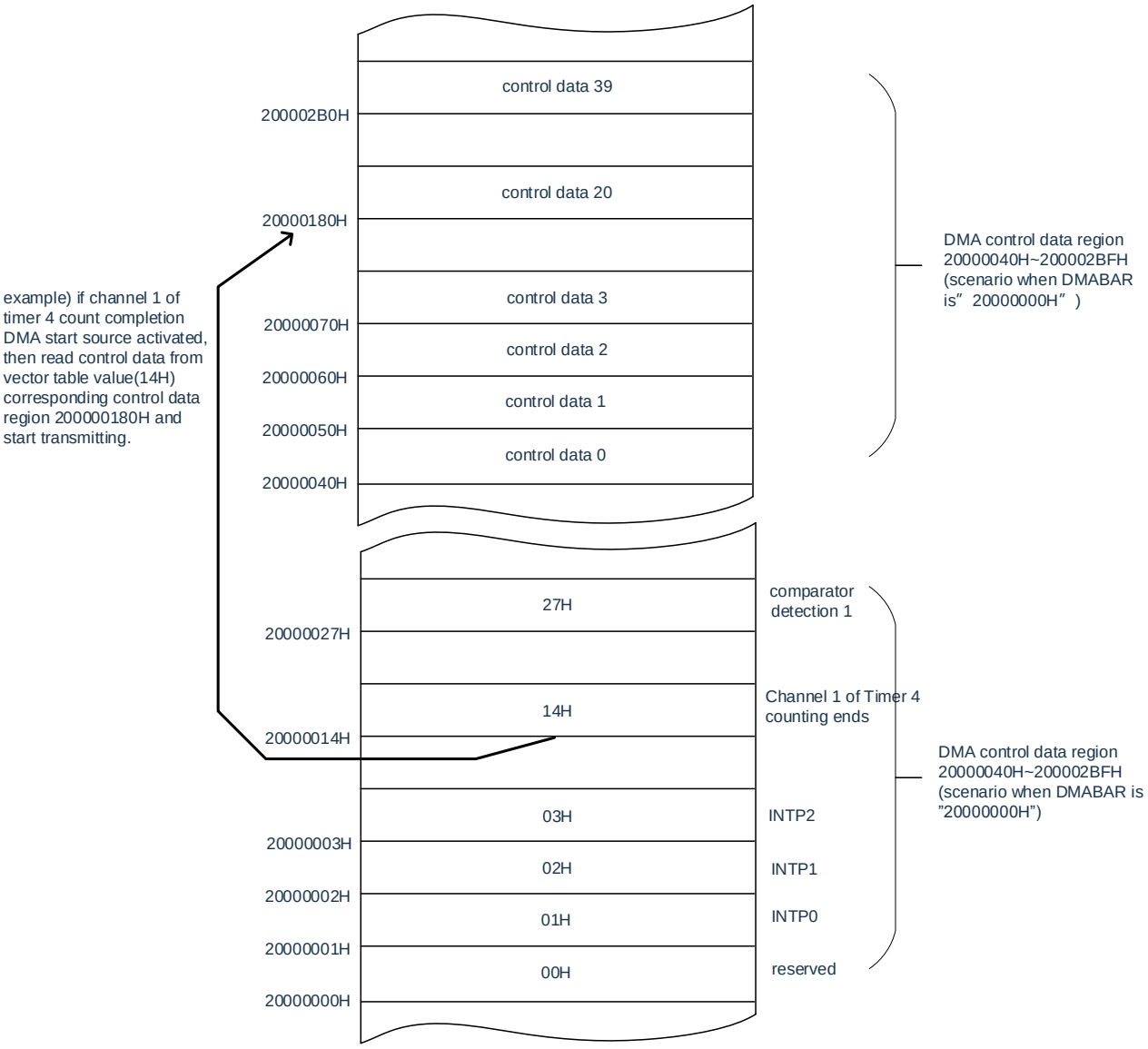


Table 23-5 DMA boot source and vector addresses

DMA boot source (interrupt request	Source	vector address	priority
Retention	0	DMABAR REGISTER SETTING ADDRESS+00H	High
INTP0	1	DMABAR REGISTER SETTING ADDRESS+01H	
INTP1	2	DMABAR REGISTER SETTING ADDRESS+02H	
INTP2	3	DMABAR REGISTER SETTING ADDRESS+03H	
INTP3	4	DMABAR REGISTER SETTING ADDRESS+04H	
INTP4	5	DMABAR REGISTER SETTING ADDRESS+05H	
INTP5	6	DMABAR REGISTER SETTING ADDRESS+06H	
INTP6	7	DMABAR REGISTER SETTING ADDRESS+07H	
INTP7	8	DMABAR REGISTER SETTING ADDRESS+08H	
key input	9	DMABAR REGISTER SETTING ADDRESS+09H	
A/D conversion ended	10	DMABAR REGISTER SETTING ADDRESS+0AH	
End of transfer received by UART0/End of transfer by CSI01 or	11	DMABAR REGISTER SETTING ADDRESS+0BH	
End of transfer sent by UART0/End of transfer by CSI00 or	12	DMABAR REGISTER SETTING ADDRESS+0CH	
End of transfer received by UART1/End of transfer by CSI11 or	13	DMABAR REGISTER SETTING ADDRESS+0DH	
End of transfer sent by UART1/End of transfer by CSI10 or	14	DMABAR REGISTER SETTING ADDRESS+0EH	
End of transfer received by UART2/End of transfer by CSI21 or	15	DMABAR REGISTER SETTING ADDRESS+0FH	
End of transfer sent by UART2/End of transfer by CSI20 or	16	DMABAR REGISTER SETTING ADDRESS+10H	
IICA0 communication ended.	17	DMABAR REGISTER SETTING ADDRESS+11H	
Retention	18	DMABAR REGISTER SETTING ADDRESS+12H	
Count or capture end for channel 0 for Timer4	19	DMABAR REGISTER SETTING ADDRESS+13H	
Count or capture end for channel 1 of Timer4	20	DMABAR REGISTER SETTING ADDRESS+14H	
Count or capture ends for channel 2 of Timer4	21	DMABAR REGISTER SETTING ADDRESS+15H	
Count or capture ends for channel 3 of Timer4	22	DMABAR REGISTER SETTING ADDRESS+16H	
A 15-bit interval timer generates a count interrupt	23	DMABAR REGISTER SETTING ADDRESS+17H	
Flash Read-Write Wipe End	24	DMABAR REGISTER SETTING ADDRESS+18H	
Overflow of TimerC	26	DMABAR REGISTER SETTING ADDRESS+1AH	
Comparison match A0 for TimerM	27	DMABAR REGISTER SETTING ADDRESS+1BH	
Comparison matching B0 for TimerM	28	DMABAR REGISTER SETTING ADDRESS+1CH	
Comparison match C0 for TimerM	29	DMABAR REGISTER SETTING ADDRESS+1DH	
Comparison match D0 for TimerM	30	DMABAR REGISTER SETTING ADDRESS+1EH	
Comparison Match A1 for TimerM	31	DMABAR REGISTER SETTING ADDRESS+1FH	
Comparison matching B1 for TimerM	32	DMABAR REGISTER SETTING ADDRESS+20H	
Comparison match C1 for TimerM	33	DMABAR REGISTER SETTING ADDRESS+21H	
Comparison Match D1 for TimerM	34	DMABAR REGISTER SETTING ADDRESS+22H	
TimerB's Comparison Match A	35	DMABAR REGISTER SETTING ADDRESS+23H	
Comparison Match B for TimerB	36	DMABAR REGISTER SETTING ADDRESS+24H	
Underflow of TimerA	37	DMABAR REGISTER SETTING ADDRESS+25H	
Comparator Detection 0	38	DMABAR REGISTER SETTING ADDRESS+26H	
Comparator Detection 1	39	DMABAR REGISTER SETTING ADDRESS+27H	Lo

23.3.4 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.

You must set bit3 (DMAEN) to "1" when using DMA.

The PER1 register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 23-5 Format of Peripheral Admission Register 1(PER1)

Address: 0x4002081A After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN	TMMEN	DMAEN	PWMPEN	TMCEN	TMAEN

DMAEN	Providing control of an input clock of a DMA
0	Stop provide an input clock. · DMA cannot run.
1	Provides an input clock. · DMA can run.

23.3.5 DMA control register j(DMACRj) (j=0~39).

The DMACRj register controls the mode of operation of the DMA.

Figure 23-6 Format of DMA control register j(DMACRj)

Address Refer to 23.3.2 Control Data Allocation.. After reset: indefinite value 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	CHEN	DAMOD	SUMMER	RPTSEL	MODE

SZ	Selection of the length of the transmitted data
00	8-bit
01	16 bit
10	32-bit
11	Disable from setting

RPTINT	Allow/Disable for Repeat Mode Interrupt
0	Interrupts are prohibited.
1	Interrupt allowed.
The setting for the RPTINT bit is not valid when the MODE bit is '0' (normal mode).	

CHEN	Chain Transmission Allowed/Disabled
0	Prohibit chain transmission.
1	Allow chain transmission.
The CHNE location of the DMACR 39 register must be "0" (Chain transfer is disabled).	

DAMOD	Control of Delivery Destination Address
0	fixed
1	incremental
The DAMOD bit is not set when the MODE bit is "1" and the RPTSEL bit is "0".	

SUMMER	Control of Delivery Source Address
0	fixed
1	incremental
The SAMOD bit is not set when the MODE bit is "1" (repeat mode) and RPTSEL bit is "1".	

RPTSEL	Selection of repeat regions
0	The delivery target is a repeat region.
1	The transfer source is a repeat region.
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	repetition mode

Note Access to the DMACRj register is not possible through DMA transfers.

23.3.6 DMA block size register j(DMBLSj) (j=0~39).

This register sets the block size of the transfer data started 1 time.

Figure 23-7 Format of DMA block size register j(DMBLSj)

Address Refer to 23.3.2 Control Data Allocation.. After reset: indefinite value 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	Transmission Block Size		
	8-bit transmission	16-bit transmission	32-bit transmission
00H	Disable from setting	Disable from setting	Disable from setting
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
. . .	. . .	. . .	. . .
FFFFFH	65535 bytes	131070 bytes	262140 bytes

Note 1. Access to the DMBLSj register cannot be performed via DMA transfer.

23.3.7 DMA transfer times register j(DMACTj) (j=0~39).

This register sets the number of data transfers for DMA. Each time you initiate a DMA transfer, you reduce 1.

Figure 23-8 Format of DMA transfer number register j(DMACTj)

Symbol: DMACTj

15	14	13	12	11	10	9	8
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 23.3.2 Control Data Allocation.. After reset: indefinite value R/W

DMACTj	Number of transfers
00H	Disable from setting
01H	1 time
02H	2 times
03H	3 times
. . .	. . .
FDH	253 times
FEH	254 times
FFH	255 times
. . .	. . .
FFFFFH	65535 times

Note 1. Access to the DMACTj register cannot be performed via DMA transfer.

23.3.8 DMA transfer times reload register j (DMRLDj) (j=0~39)

This register sets the initial value of the transfer number register in repeat mode. In repeat mode, the value of this register must be the same as the initial value of the DMACT register because it is reloaded into the DMACT register.

Figure 23-9 Format of DMA transfer number reload register j(DMRLDj)

Address Refer to 23.3.2 Control Data Allocation.. After reset: indefinite value 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

Note 1. Access to the DMRLDj register cannot be performed via DMA transfer.

23.3.9 DMA source address register j(DMSARj) (j=0~39).

This register specifies the transmission 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 23-10 Format of DMA source address register j(DMSARj)

Address Refer to 23.3.2 Control Data Allocation.. After reset: indefinite value R/W

symbol	31	30	29	28	27	26	25	24
DMSARj	DMSARj31	DMSARj30	DMSARj29	DMSARj28	DMSARj27	DMSARj26	DMSARj25	DMSARj24
	23	22	21	20	19	18	17	16
	DMSARj23	DMSARj22	DMSARj21	DMSARj20	DMSARj19	DMSARj18	DMSARj17	DMSARj16
	15	14	13	12	11	10	9	8
	DMSARj15	DMSARj14	DMSARj13	DMSARj12	DMSARj11	DMSARj10	DMSARj9	DMSARj8
	7	6	5	4	3	2	1	0
	DMSARj7	DMSARj6	DMSARj5	DMSARj4	DMSARj3	DMSARj2	DMSARj1	DMSARj0

Note 1. Access to the DMSARj register cannot be performed via DMA transfer.

23.3.10 DMA destination address register j(DMDARj) (j=0~39).

This register specifies the destination 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 23-11 Format of DMA destination address register j(DMDARj)

Address Refer to 23.3.2 Control Data Allocation.. After reset: indefinite value R/W

symbol	31	30	29	28	27	26	25	24
DMDARj	DMDARj31	DMDARj30	DMDARj29	DMDARj28	DMDARj27	DMDARj26	DMDARj25	DMDARj24
	23	22	21	20	19	18	17	16
	DMDARj23	DMDARj22	DMDARj21	DMDARj20	DMDARj19	DMDARj18	DMDARj17	DMDARj16
	15	14	13	12	11	10	9	8
	DMDARj15	DMDARj14	DMDARj13	DMDARj12	DMDARj11	DMDARj10	DMDARj9	DMDARj8
	7	6	5	4	3	2	1	0
	DMDARj7	DMDARj6	DMDARj5	DMDARj4	DMDARj3	DMDARj2	DMDARj1	DMDARj0

Note: Access to the DMDARj register cannot be made via DMA transfer.

23.3.11 DMA Boot Enable Register i (DMAENi) (i=0~4)

This is an 8-bit register that controls whether or not starting DMA through each interrupt source. The correspondence of that interrupt source and the DMAENi0~DMAENi7 bit is shown in Table 23-6.

The DMAENi register can be set by an 8-bit memory operation instruction.

Note 1. You must change the DMAENi0~DMAENi7 bit at a location that does not produce a start source for the bit.

2. Access to the DMAENi register cannot be performed via DMA transfer.

3. The assigned function is different for the product, and the location "0" must be not assigned.

Figure 23-12 Format of DMA start-up permit register i(DMAENi) (i=0~4)

Addresses: 40005000H (DMAEN0), 40005001H (DMAEN1),
40005002H (DMAEN2), 40005003H (DMAEN3),
40005004H (DMAEN4) After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
DMAENi	DMAENj7	DMAENj6	DMAENj5	DMAENj4	DMAENj3	DMAENj2	DMAENj1	DMAENj0

DMAENi7	Allow i7 for DMA boot
0	Disable start-up.
1	Allow start-up.
The DMAENi7 bit changes to "0" depending on the condition of the end of the delivery interrupt.	

DMAENi6	Allow i6 for DMA boot
0	Disable start-up.
1	Allow start-up.
The DMAENi6 bit changes to "0" depending on the condition of the end of the delivery interrupt.	

DMAENi5	Allow i5 for DMA boot
0	Disable start-up.
1	Allow start-up.
The DMAENi5 bit changes to "0" depending on the condition of the end of the delivery interrupt.	

DMAENi4	Allow i4 for DMA boot
0	Disable start-up.
1	Allow start-up.
The DMAENi4 bit changes to "0" depending on the condition of the end of the delivery interrupt.	

DMAENi3	Allow i3 for DMA start-up
0	Disable start-up.
1	Allow start-up.
The DMAENi3 bit changes to "0" depending on the condition of the end of the delivery interrupt.	

DMAENi2	Allow i2 for DMA start-up
0	Disable start-up.
1	Allow start-up.
The DMAENi2 bit changes to "0" depending on the condition of the end of the delivery interrupt.	

DMAENi1	Allow i1 for DMA start-up
0	Disable start-up.
1	Allow start-up.
The DMAENi1 bit changes to "0" depending on the condition of the end of the delivery interrupt.	

DMAENi0	Allow i0 for DMA start-up
0	Disable start-up.
1	Allow start-up.
The DMAENi0 bit changes to "0" depending on the condition of the end of the delivery interrupt.	

Table 23-6 The Correspondence of Interrupt Source and DMAENi0~DMAENi7 Bits

register	DMAENi7 bit	DMAENi6 bit	DMAENi5 bit	DMAENi4 bit	DMAENi3 bit	DMAENi2 bit	DMAENi1 bit	DMAENi0 bit
DMAEN0	INTP6	INTP5	INTP4	INTP3	INTP2	INTP1	INTP0	Retention
DMAEN1	UART2 Receive End of transfer for /CSI21 End of transfer or buffer empty /IIC21 end of transfer	UART1 Send End of transfer for /CSI10 End of transfer or buffer empty /IIC10 end of transfer	UART1 Receive End of transfer for /CSI11 End of transfer or buffer empty /IIC11 end of transfer	UART0 Send End of transfer for /CSI00 End of transfer or buffer empty /IIC00 end of transfer	UART0 Receive End of Transfer for / Transmission of CSI01 End or Buffer Null/IIC01 End of transfer for	A/D Conversion End	KEY Input	INTP7
DMAEN2	15-bit interval timer interrupt	timer array single Channel 3 of Element 0 Count End of or Person capture ended	timer array single Channel 2 of Element 0 Count End of or Person capture ended	timer array single Channel 1 of Element 0 Count End of or Person capture ended	timer array single Channel 0 of Element 0 Count End of or Person capture ended	Retention	IICA0 communication end	UART2 Send End of Transfer for / Transmission of CSI20 End or Buffer Null/IIC20 End of transfer for
DMAEN3	Comparison Match A1 for TimerM	TimerM's Comparison Match D0	TimerM's Comparison Match C0	TimerM's Comparison Match B0	TimerM's Comparison Match A0	TimerC's overspill	Retention	Flash Erase/Write End
DMAEN4	Comparator Detection 1	Comparator Detection 0	TimerA's overspill	Comparison Match B for TimerB	TimerB's Comparison Match A	Comparison Match D1 for TimerM	Comparison match C1 for TimerM	Comparison matching B1 for TimerM

Note The location "0" without the allocation function must be.

Remark i=0~4

23.3.12 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.

Note 1. The DMABAR register must be changed in a state where all DMA boot sources are set to disable booting.

2. DMABAR registers can only be overwritten 1 time.

3. Access to the DMABAR register cannot be performed via DMA transfer.

4. Refer to the note for "23.3.1 DMA control data area and DMA vector table area assignment" for allocations.

5. Set the register to maintain 1024Byte alignment, that is, set the low 10 bit to zero. DMA hardware ignores low 10 bits.

6. The register can only be accessed by WORD, BYTE and HALFWORD access are ignored.

Figure 23-13 Format of DMA base address register (DMABAR)

Address: 40005008H After reset: 00000000H R/W

symbol	31	30	29	28	27	26	25	24
DMABARj	DMABARj 31	DMABARj 30	DMABARj 29	DMABARj 28	DMABARj 27	DMABARj 26	DMABARj 25	DMABARj 24
	23	22	21	20	19	18	17	16
	DMABARj 23	DMABARj 22	DMABARj 21	DMABARj 20	DMABARj 19	DMABARj 18	DMABARj 17	DMABARj 16
	15	14	13	12	11	10	9	8
	DMABARj 15	DMABARj 14	DMABARj 13	DMABARj 12	DMABARj 11	DMABARj 10	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

23.4 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 40 sets of control data to a DMA control data area and to perform transfer of 40 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=0~39) register is '1' (allowing chain transfer), the continuous data transfer (chain transfer) is read by 1 start-up sources.

The transmission source address and the transmission destination 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.

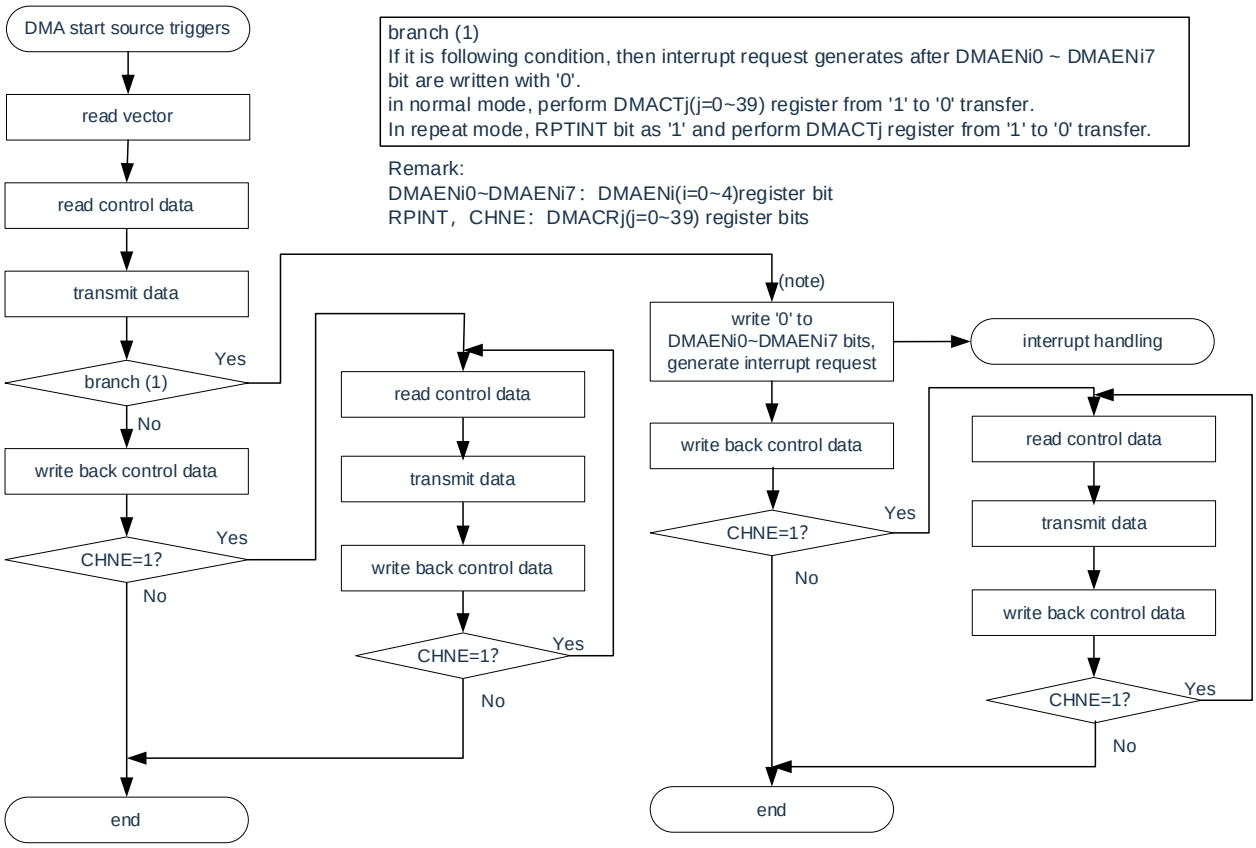
23.4.1 Boot Source

The DMA is started by a peripheral function interrupt signal and the interrupt signal for starting the DMA is selected by a DMAENi (i=0~4) register. When the data transfer (the initial transfer is continued during chain transfer) is set to the following two conditions, the corresponding DMAENi register DMAENi0-DMAENi7 position.

- In normal mode, the DMACTj (j=0~39) register is transferred to "0".
- In repeat mode, the RPTINT bit of the DMACRj register is '1' (interrupt allowed) and the DMACTj register is '0' transferred.

The internal flow chart of the DMA is shown in Figure 23-14.

Figure 23-14 Flowchart of internal operation of DMA



Note: In a data transfer initiated by a setting allowing chain transfer (CHNE=1), the DMAENi0~DMAENi7 bit is not written "0" and no interrupt request is generated.

23.4.2 normal mode

At 8-bit transmission, the transmission data of 1 start-up is 1~65535 bytes. In 16-bit transmission, the transmission data of 1 start-up is 2~131070 bytes; At 32-bit delivery, the 1-initiated delivery data is 4-262140 bytes. The number of transmissions was 1-65535 times. If the DMACTj (j=0~39) register is transferred to '0', the interrupt request corresponding to the start source is generated to the interrupt controller in the DMA operation and the DMAENi0~DMAENi7 position of the corresponding DMAENi (i=0~4) register is '0'.

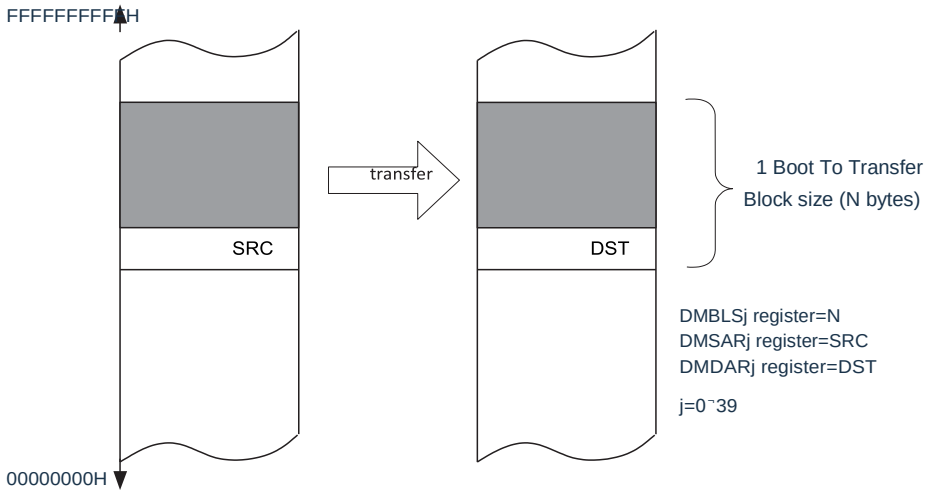
Register functions and data transfers in normal mode are shown in Tables 23-7 and Figures 23-15 respectively.

Table 23-7 Register Function in Normal Mode

register name	symbol	Features
DMA block size register j	DMBLSj	1 Boot The size of the data block to be transferred
DMA transfer number register j	DMACTj	Number of data transfers
DMA transfer times reload register j	DMRLDj	Do not use note.
DMA source address register j	DMSARj	The source address where the data is sent
DMA destination address register j	DMDARj	Destination address for data transfer

Note When parity error reset (RPERDIS=0) is allowed to occur through the RAM parity error detection function, initialization must be performed (00H). Remark j=0~39

Figure 23-15 Normal mode data transfer



Settings for the DMACR				Control of source address	Control of Target Address	Source address after delivery	Destination address after
DAMOD	SUMME	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"

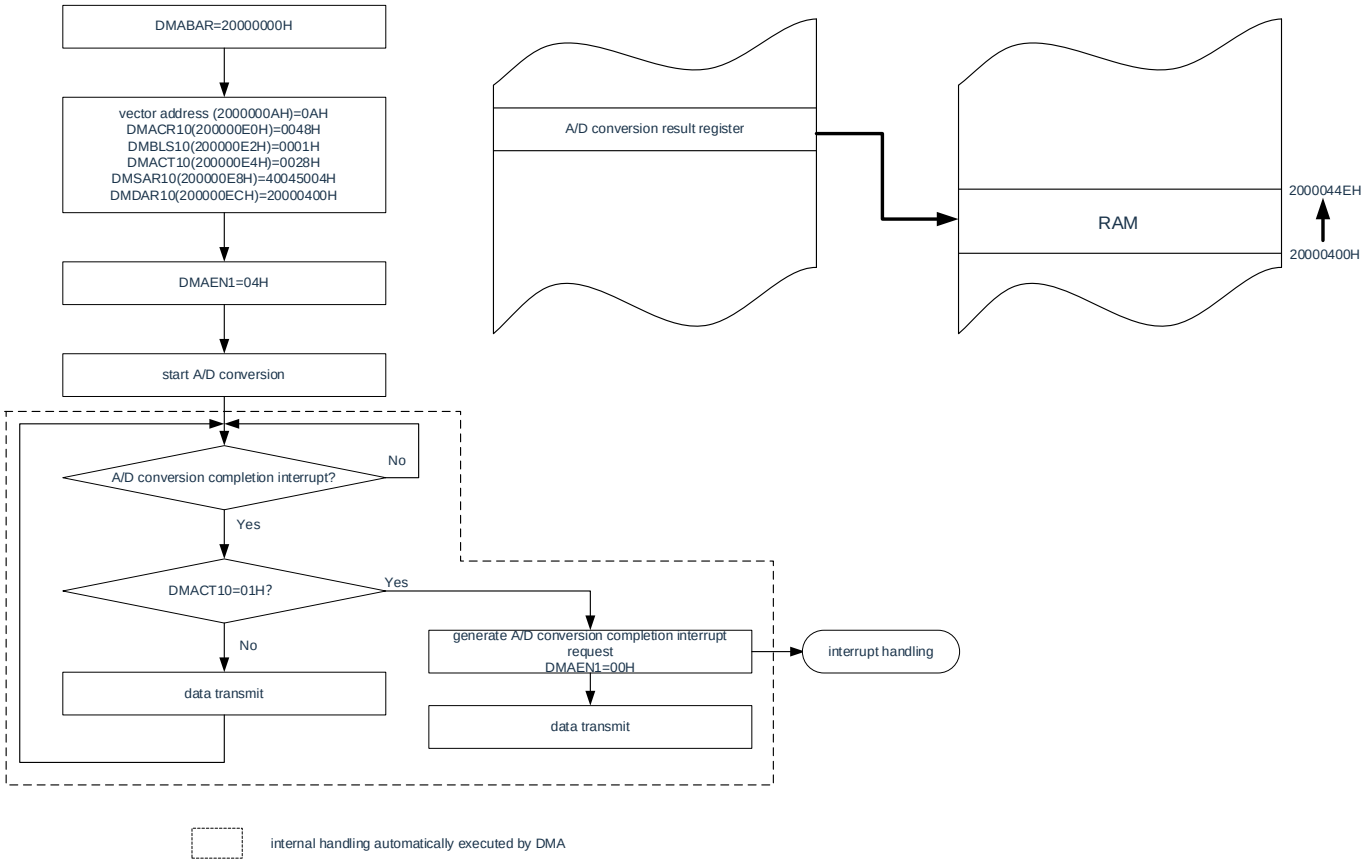
(1) Example of normal mode use 1: Continuous reading A/D conversion results

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 allocated in 20000 AH, and the control data is allocated in 200000E0H~20000EFH.

· The 2-byte data of 40045004H (40045005H) is transferred 40 times to 20000400H of RAM~200044FHof 80 bytes.

Figure 23-16 Example of normal mode use 1: Continuously taking 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.

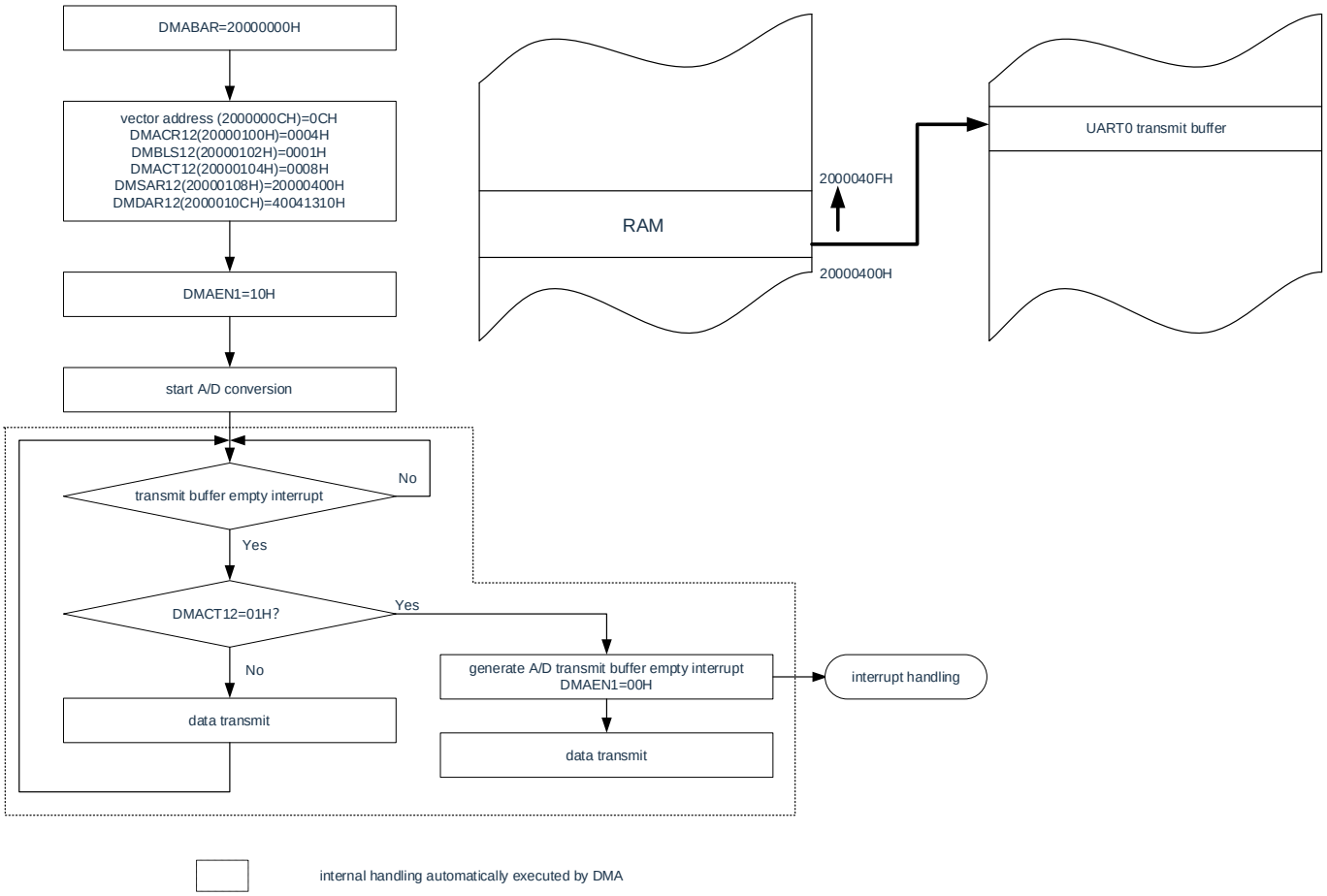
(2) Example of Normal Mode Usage 2: UART0 Continuous Transmit

The DMA is initiated over the transmission buffer of the UART0 and the value of the RAM is transferred to the transmission buffer of the UART0.

· The vector address is assigned in 2000000CH, and the control data is assigned in 20000100H~2000010FH.

· Transfer 8 bytes of RAM 20000400H~2000407H to the UART0 transmission buffer (40041310H).

Figure 23-17 Example of Normal Mode Usage 2: UART0 Continuous Send Out



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.

You must start the first UART0 send with the software. The DMA is initiated by an air interrupt of a transmission buffer, and then automatically performs a second subsequent transmission.

23.4.3 repetition mode

The 1-time-initiated transfer data is 1-65535 bytes. The transmission source or the transmission target is designated as a repeat region, and the number of transmissions is 1-65535 times. Once the specified number of transfers is over, the DMACTj (j=0~39) register and the address specified as the repeat region are initialized and then repeatedly transferred. When the DMACRj register has a RPTINT bit of '1' (interrupt allowed) and the DMACTj register becomes '0', the DMAENi register is disabled. When the RPTINT bit of the DMACRj register is '0', no interrupt request is generated even if the DMACTj register becomes '0' and the DMAENi00.

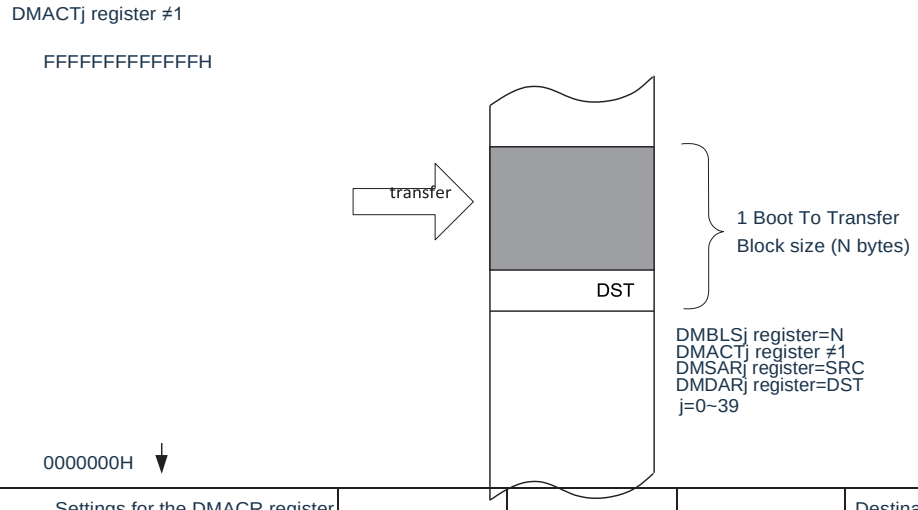
Register functions and data transfer for the repeat mode are shown in Tables 23-8 and Figures 23-18 respectively.

Table 23-8 Register Function for Repeat Mode

register name	symbol	Features
DMA block size register j	DMBLSj	1 Boot The size of the data block to be transferred
DMA transfer number register j	DMACTj	Number of data transfers
DMA transfer times reload register j	DMRLDj	Reload the value of this register into the DMACT register. (initialize the number of transfers of data)
DMA source address register j	DMSARj	The source address where the data is sent
DMA destination address register j	DMDARj	Destination address for data transfer

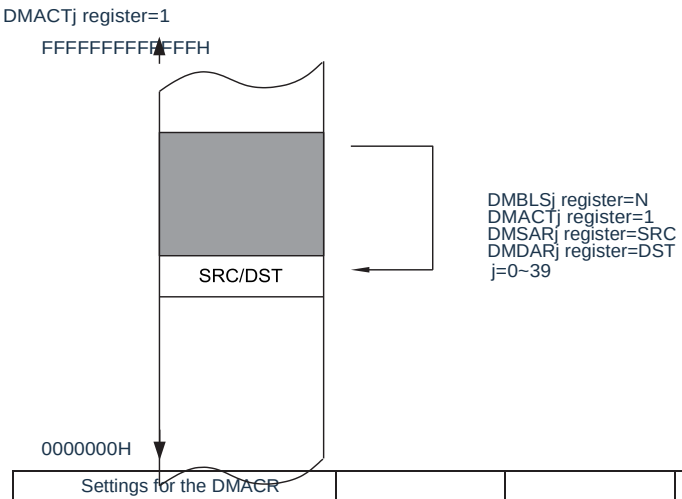
Remark j=0~39

Figure 23-18 Repeat mode data transfer



Settings for the DMACR register				Control of source address	Control of Target Address	Source address after delivery	Destination address after delivery
DAMOD	SUMMER	RPTSEL	MODE				
0	X	1	1	repeat region	fixed	SRC+N	DST
1	X	1	1	repeat region	incremental	SRC+N	DST+N
X	0	0	1	fixed	repeat region	SRC	DST+N
X	1	0	1	incremental	repeat region	SRC+N	DST+N

X:"0" or "1"



Settings for the DMACR register				Control of source address	Control of Target Address	Source address after delivery	Destination address after delivery
DAMOD	SUMMER	RPTSEL	MODE				
0	X	1	1	repeat region	fixed	SRC	DST
1	X	1	1	repeat region	incremental	SRC	DST+N
X	0	0	1	fixed	repeat region	SRC	DST
X	1	0	1	incremental	repeat region	SRC+N	DST

SRC0: Initial value of source address DST0: Initial value of target address X:'0' or'1'

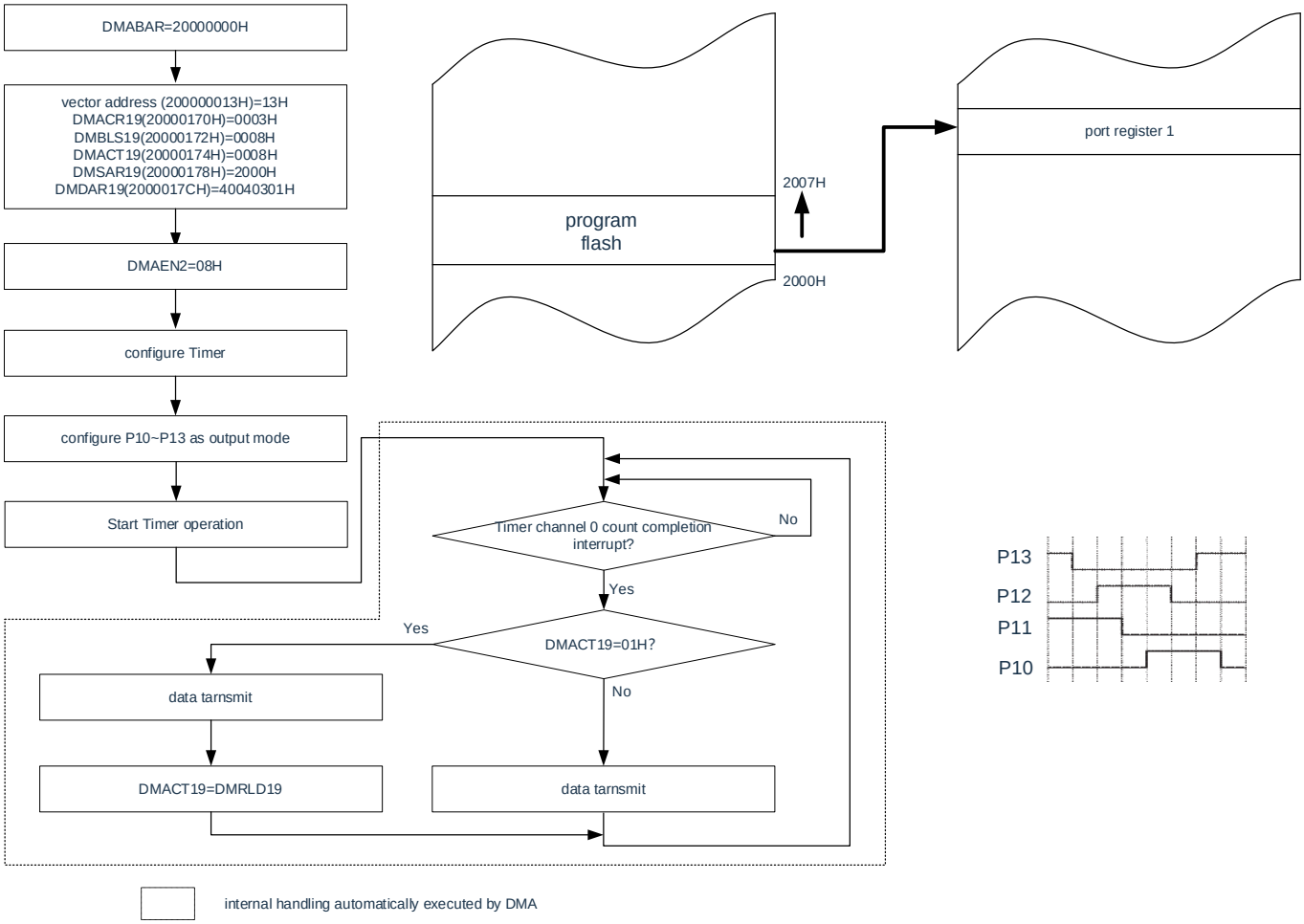
Note 1. When using repeat mode, you must set the data length of the repeat region within 65535 bytes.

(1) Example of Using repetition Mode 1: Controlling Pulse Output Using a Port Stepping Motor

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 allocated in 2000013H, and the control data is allocated in 2000170H~2000017FH.
- The 8 bytes of 02000H~02007H of code flash memory are transferred to port register 1 (40040301H).
- Suppress repeating pattern interrupts.

Figure 23-19 Example of Using Duplicate Mode 1: Controlling Pulse Output Using a Port Stepping Motor



To stop the output, you must clear the bit3 of the DMAEN2 after stopping the timer running.

(2) Example of Using Repeating Patterns 2: Sine Wave Output Using 8-bit D/A Converter

Using the channel 0 interval timer function using Timer4 and by interrupting starting DMA, the sine wave table saved in the data flash memory is transferred to 8-bit D/A conversion value setting register 0 (40044734H).

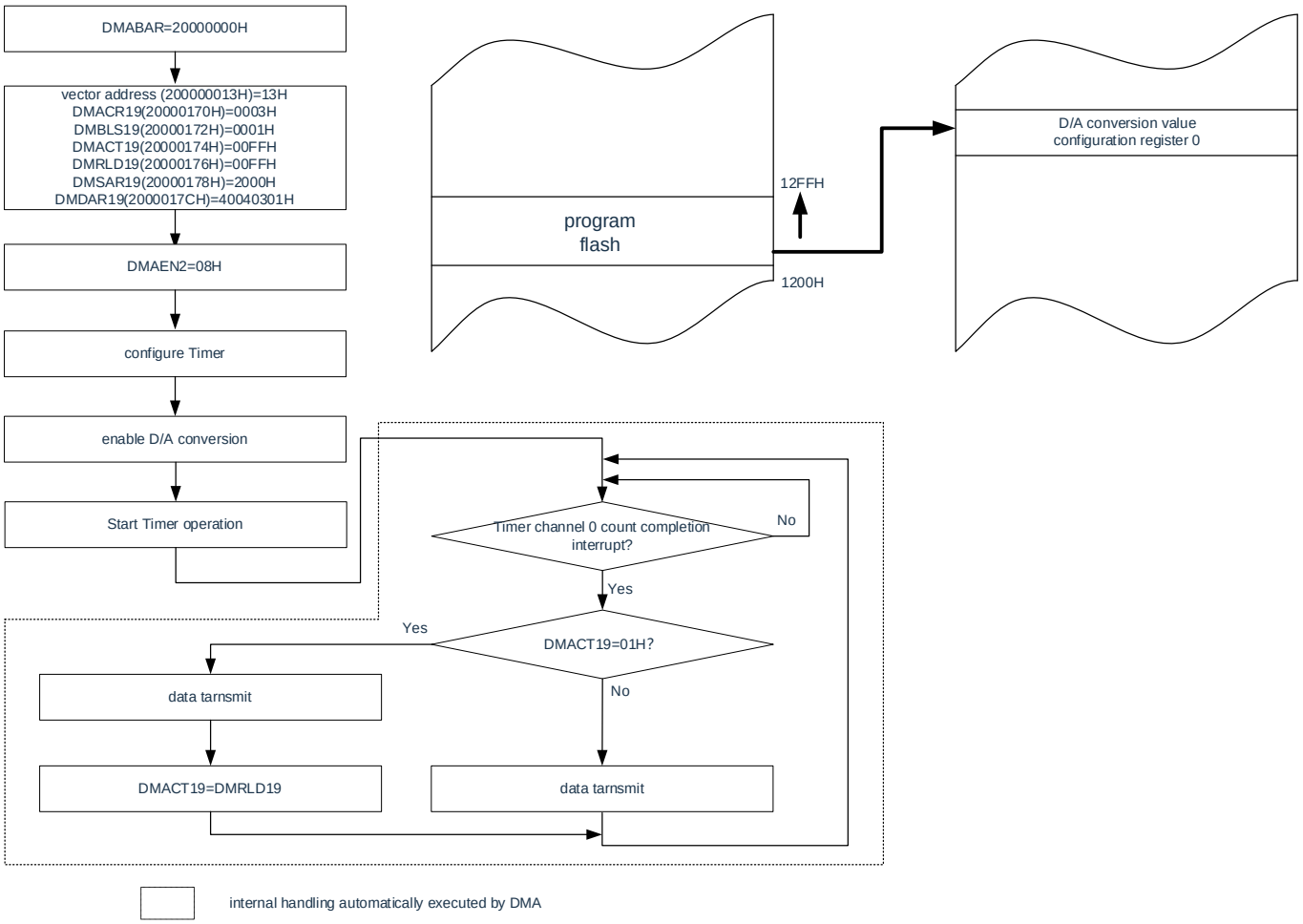
Set the interval time of the timer to D/A output preparation time.

· The vector address is allocated in 2000013H, and the control data is allocated in 2000170H~2000017FH.

· The 255 bytes of 1200H~12 FEH data of the data flash memory are transferred to D/A conversion value setting register 0 (4044734H).

· Suppress repeating pattern interrupts.

Figure 23-20 Example of Using Repeating Patterns 2: Sine Wave Output Using 8-bit D/A Converter



To stop the output, you must clear the bit3 of the DMAEN2 after stopping the timer running.

23.4.4 chain transmission

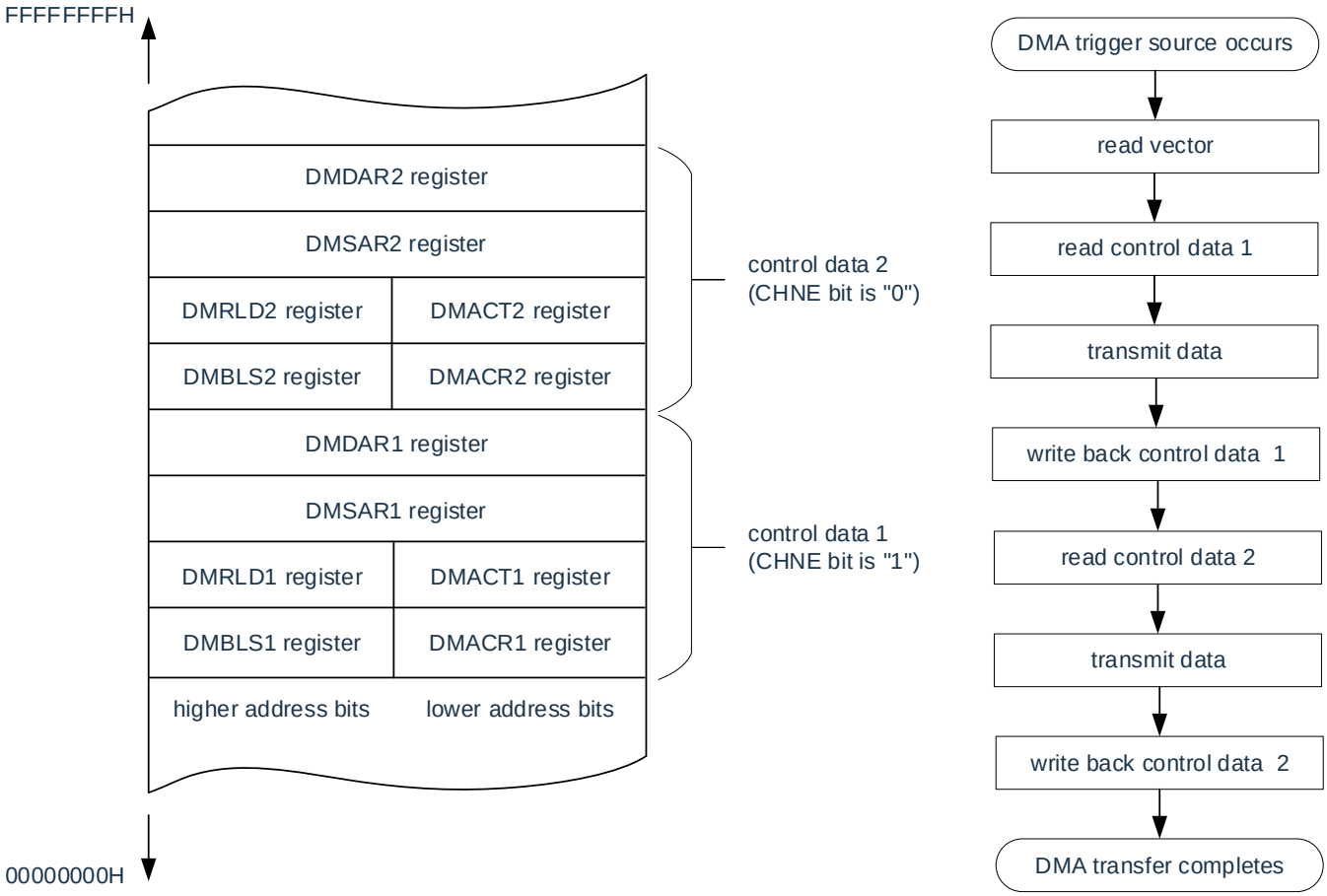
When the CHNE bit of DMACRj (j=0~38) register is '1 (Allow chain transmission), the transmission of multiple data can be carried out continuously through 1 start sources.

Once the DMA is started, control data assigned to the DMA control data area is read by selecting control data from data read from a vector address corresponding to the start source. If the CHNE bit of the read control data is '1' (allowing chain transmission), the transmission is continued by reading the next allocated control data after transmission. Repeat until the control data transfer with the CHNE bit "0" (Chain Transfer Disabled) is complete.

When the plurality of control data is used for chain transmission, the transmission times of the first control data setting are valid, and the transmission times of the second subsequent processing are invalid.

The flow chart for chain transfer is shown in Figure 23-21.

Figure 23-21 Flow chart for chain transfer



Note 1. The CHNE location of the DMACR 39 register must be "0" (Chain Transfer Disabled).

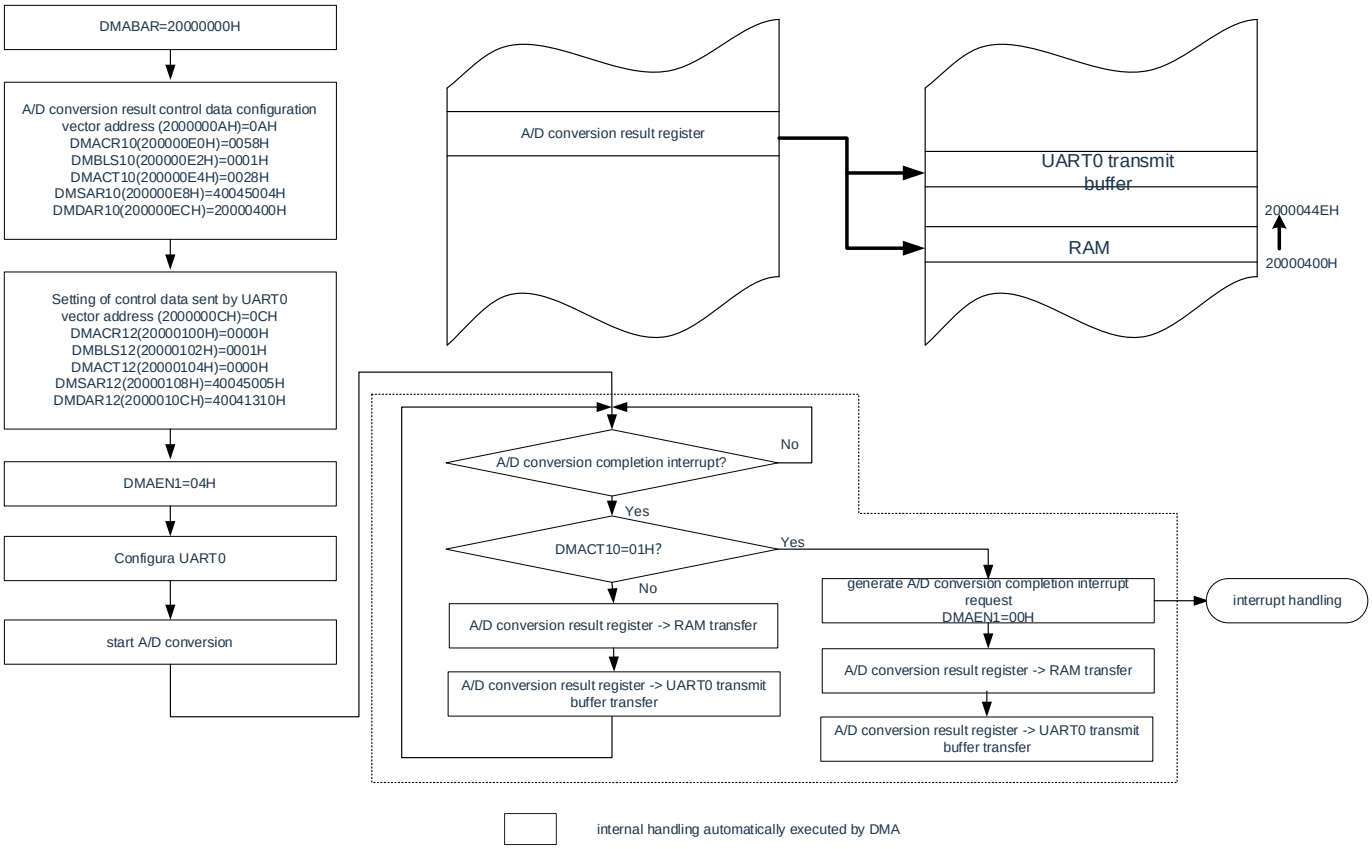
2. When data is transferred after the second time of chain transfer, DMAENi0~DMAENi7 bit of DMAENi (i=0~4) is unchanged as '0' (DMA start disabled) and no interrupt request is generated.

(1) Examples of the use of chain transmission: Continuously taking A/D conversion result for UART0 transmission

The DMA is started by an A/D conversion end interrupt and the A/D conversion result is transmitted to RAM for UART0 transmission.

- The vector addresses are 200000AH and 200000CH respectively.
- The control data of A/D conversion results were distributed in the range of 200000E0H~200000 EFH.
- The control data sent by UART0 is distributed in the range of 20000100H~2000010 FH.
- The two-byte data of 40045004H (40045005H) is transferred to 20000400H-2000044FH of RAM, and the high 1-byte (40045005H) of A/D is transferred to UART0 transmission buffer (40041310H).

Figure 23-22 Examples of the use of chain transmission: Continuously taking A/D conversion result for UART0 transmission



23.5 Precautionswhen using DMA

23.5.1 DMA control data and vector table settings

· The DMA base address register (DMABAR) must be changed in a state where all DMA boot sources are set to disable boot.

· The DMA base address register (DMABAR) can only be overridden 1 time.

· You must change the DMACRj, DMBLSj, DMACTj, DMRLDj, DMSARj, DMDARj register data when the DMAENi0~DMAENi7 bit of the corresponding DMAENi (i=0~4) is '0' (DMA boot disabled) register.

· The starting address of the DMA control data area set in the vector table must be changed when the DMAENi0~DMAENi7 bit of the corresponding DMAENi (i=0~4) register is '0' (DMA boot disabled).

23.5.2 DMA controls the allocation of data areas and DMA vector table areas

Areas where DMA control data and vector tables can be allocated vary depending on the product and usage conditions.

· The stack, DMA control data, and DMA vector table sections cannot overlap.

· The DMRLD register must be initialized (0000H) even when normal mode is used when parity error reset (RPERDIS=0) is allowed by RAM.

23.5.3 The number of execution clocks for DMA

The execution of the DMA at start-up and the number of clocks required are shown in Table 23-9.

Table 23-9 Performance and Required Number of Clocks at DMA Start-up

reading vector	control data		reading 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, refer to "Table 23-10 The number of clocks required to write back control data".

2. Refer to "Table 23-11 for the number of clocks required to read and write data The number of clocks required to read and write data".

Table 23-10. The number of clocks required to write back control data

Settings for the DMACR register				Address Settings		Control register write-back				Clock Count
DAMOD	SUMMER	RPTSEL	MODE	Source	Objectives	DMACTj register	DMRLDj register	DMSARj register	DMDARj register	
0	0	X	0	fixed	fixed	write-back	write-back	Do not write back	Do not write back	1
0	1	X	0	incremental	fixed	write-back	write-back	write-back	Do not write back	2
1	0	X	0	fixed	incremental	write-back	write-back	Do not write back	write-back	2
1	1	X	0	incremental	incremental	write-back	write-back	write-back	write-back	3
0	X	1	1	repeat region	fixed	write-back	write-back	write-back	Do not write back	2
1	X	1	1		incremental	write-back	write-back	write-back	write-back	3
X	0	0	1	fixed	repeat region	write-back	write-back	Do not write back	write-back	2
X	1	0	1	incremental		write-back	write-back	write-back	write-back	3

Remark j=0~23,X:"0" or "1"

Table 23-11 The number of clocks required to read and write data

Execution status	RAM	code flash	data flash	special function register (SFR)	Extended Special Function Register (2ndSFR)	
					No Wait	when
reading data	1	2	4	1	1	1+Wait Callout
write data	1	—	—	1	1	1+Wait Callout

23.5.4 Response time for DMA

The DMA response time is shown in Table 23-12. The DMA response time is the time from the detection of the DMA boot source to the start of the DMA transfer, excluding the number of DMA execution clocks.

Table 23-12 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

· Maximum Response Time for Accessing Waiting
TimerA Registers: Maximum response time for
each condition +1 clock

Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU/peripheral hardware clock)

23.5.5 Start Source for DMA

· You cannot enter the same boot source during the period from the input DMA boot source to the end of the DMA transfer.

· At the location where the DMA boot source was generated, the corresponding DMA boot allowed bit for the boot source cannot be operated.

· If that DMA start source transmit contention, the priority is determine when the CPU accept the DMA transfer to start the start source. Refer to the "23.3.3 Vector Table" for starting source priority.

· If DMA start is permitted in one of the following states, DMA transfer is started and an interrupt is generated after transfer. Therefore, the monitor flag (CnMON) of the comparator must be confirmed as necessary to allow the DMA to start.
- Interrupt request (CnEDG=0) is generated by single edge detection of comparator note and interrupt request (CnEPO=0) is generated IVCMP>IVREF (or internal reference 1.45V).
- An interrupt request (CnEDG=0) is generated by one edge detection of the comparator and an interrupt request (CnEPO=1) is generated by the comparator drop.

(n=0, 1)

23.5.6 Operation in standby mode

Status	DMA Operation
sleep mode	Capable of operation (not allowed in low power RTC mode).
deep sleep mode	Able to accept DMA boot source and perform DMA transfer note 1

Note 1. In deep sleep mode, DMA transfer can be performed after a DMA start source is detected, and return to deep sleep mode after transfer. However, flash memory cannot be set as a transfer source because that code flash and data flash memory stop running in deep sleep mode.

Chapter 24 Coordination Controller(EVENTC)

24.1 Features of EVENTC

EVENTC links events output by each peripheral function to each other by the peripheral function. The event link allows for collaborative operation between peripheral functions without passing through the CPU.

EVENTC has the following features:

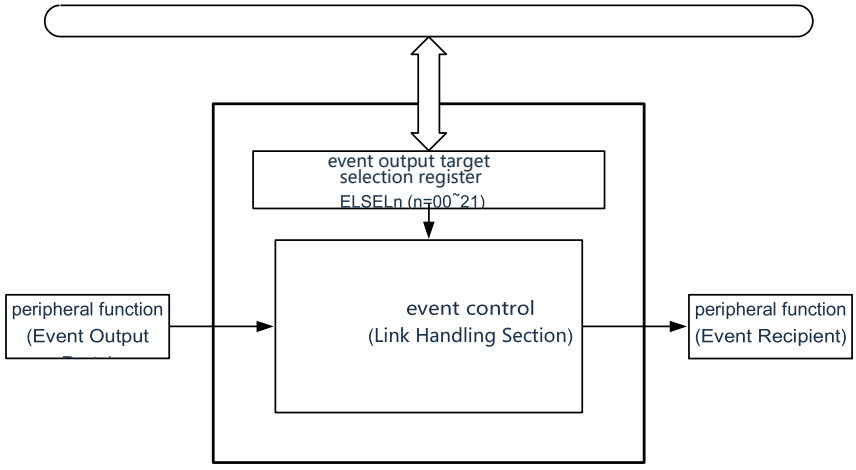
· According to the product, event signals of 21-22 peripheral functions can be directly linked to the designated peripheral functions.

· According to the product, the event signal can be used as a starting source for operation of 1 of 10 peripheral functions.

24.2 Structure of EVENTC

The EVENTC box is shown in Figure 24-1.

Figure 24-1 Block diagram for EVENTC



24.3 control register

The controller register is shown in Table 24-1.

Table 24-1 Register controlling EVENTC

register name	symbol
Event Output Target Selection Register 00	ELSEL00
Event Output Target Selection Register 01	ELSEL01
Event Output Target Selection Register 02	ELSEL02
Event output target selection register 03	ELSEL03
Event output target selection register 04	ELSEL04
Event output target selection register 05	ELSEL05
Event output target selection register 06	ELSEL06
Event output target selection register 07	ELSEL07
Event output target selection register 08	ELSEL08
Event output target selection register 09	ELSEL09
Event Output Target Selection Register 10	ELSEL10
Event output target selection register 11	ELSEL11
Event output target selection register 12	ELSEL12
Event Output Target Selection Register 13	ELSEL13
Event output target selection register 14	ELSEL14
Event output target selection register 15	ELSEL15
Event Output Target Selection Register 16	ELSEL16
Event Output Target Selection Register 17	ELSEL17
Event Output Target Selection Register 18	ELSEL18
Event Output Target Selection Register 19	ELSEL19
Event Output Target Selection Register 20	ELSEL20
Event output target selection register 21	ELSEL21

24.3.1 Output target selection register n(ELSELRn) (n=00~21).

The ELSELRn register links each event signal to an event recipient peripheral function (link target peripheral function) when the event is accepted. Multiple event inputs cannot be linked to the same event output target (event recipient). Otherwise, the operation of the event recipient peripheral function may be inconsistent and the event signal may not be normally accepted. In addition, event link generation source and event output target cannot be set to the same function.

The ELSELRn register must be set during the period when all event output peripheral functions do not generate event signals.

The ELSELRn register (n=00~21) and peripheral function are shown in Table 24-2, and the ELSELRn register (n=00~21) are shown in Table 24-3.

Figure 2 4-2 Format of event output target selection register n(ELSELRn)

Location: 40043400H (ELSELR00)~40043415H (ELSELR21) After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
ELSELN	0	0	0	0	ELSELn3	ELSELn2	ELSELn1	ELSELn0

ELSELn3	ELSELn2	ELSELn1	ELSELn0	Event Link Selection
0	0	0	0	Disable event link.
0	0	0	1	Select Run Note 1 for the linked peripheral function 1.
0	0	1	0	Select Run Note 1 for the linked peripheral function 2.
0	0	1	1	Select Run Note 1 for the Linked Peripheral function 3.
0	1	0	0	Select Run Note 1 for the linked peripheral function 4.
0	1	0	1	Select Run Note 1 for the linked peripheral function 5.
0	1	1	0	Select Run Note 1 for the linked peripheral function 6.
0	1	1	1	Select Run Note 1 for the Linked Peripheral function 7.
1	0	0	0	Select Run Note 1 for the linked peripheral function 8.
1	0	0	1	Select Run Note 1 for the Linked Peripheral function 9.
Other than above				Disable from setting.

Note 1. Refer to Table 24-3 ELSELRn registers (n=00~21) for set values and run responses for link target peripheral features accepting events..

Table 24-2 ELSELRn register (n=00~21) and correspondence of peripheral function

register	Event Occurrence Source (output source for event	Event Content
ELSEL00	External Interrupt Edge Detection 0	INTP0
ELSEL01	External Interrupt Edge Detection 1	INTP1
ELSEL02	External Interrupt Edge Detection 2	INTP2
ELSEL03	External Break Edge Detection 3	INTP3
ELSEL04	External Interrupt Edge Detection 4	INTP4
ELSEL05	External Break Edge Detection 5	INTP5
ELSEL06	key return signal detection	INTKR
ELSEL07	RTC fixed cycle/alarm clock consistent detection	INTRTC
ELSEL08	Timer M Input Capture A0/Comparison Match A0	INTTMM0
ELSEL09	Timer M Input Capture B0/Comparison Match B0	INTTMM0
ELSEL10	Timer M Input Capture A1/Comparison Match A1	INTTMM1
ELSEL11	Timer M Input Capture B1/Comparison Match B1	INTTMM1
ELSEL12	Timer M underflow	TMM underflow signal
ELSEL13	Timer A End of Spillover/Pulse Width Measurement Period/End of Pulse Period Measurement Period	INTTMA
ELSEL14	Timer B Input Capture A/Comparison Match A	INTMB
ELSEL15	Timer B Input Capture B/Comparison Match B	INTMB
ELSEL16	End of count/end of capture for Timer4 channel 00	INTTM00
ELSEL17	Count End/Capture End for Timer4 Channel 01	INTTM01
ELSEL18	Count End/Capture End for Timer4 Channel 02	INTTM02
ELSEL19	Count End/Capture End for Timer4 Channel 03	INTTM03
ELSEL20	Comparator Detection 0	INTCMP0
ELSEL21	Comparator Detection 1	INTCMP1

Table 24-3 The setting value of the ELSELRn register (n=00~21) and the corresponding of the run when the link target peripheral function accepts the event

ELSELRn register's ELSELn3~ELSELn0 bits	Link Target No.	Link Target Peripherals	Run when event is accepted
0001B	1	A/D converter	Start the A/D conversion.
0010B	2	Timer Input Note 1 for Timer4 Channel 0	delay counter, measurement of input pulse interval, external event counter
0011B	3	Timer4 Timer Input Note 2 for Channel 1	delay counter, measurement of input pulse interval, external event counter
0100B	4	Timer A	count source
0101B	5	Timer B	Input capture for TBIO1
0110B	6	Timer M	TMIOD0's Input Capture, Mandatory Cut-off of Pulse Output
0111B	7	Timer M	TMIOD1's Input Capture, Mandatory Cut-off of Pulse Output
1000B	8	DA0 Note 3	real-time output
1001B	9	DA1 Note 3	real-time output

Note 1. To select timer input for Timer4 channel 0 as link target peripheral function, the running clock of channel 0 must be set to fCLK through timer clock select register 0 (NFEN1) and TI00 pin (TNFEN00=0) and sets the timer input used by channel 0 as the event input signal of the Coordination Controllerby the timer input selection register 0 (TIS0).

2. To select the timer input of Timer4 channel 1 as the linking target peripheral function, the running clock of channel 1 must be set to fCLK through timer clock selection register 0 (TPS0), the noise filter of TI01 pin is set to OFF (TNFEN01=0) through noise filter permit register 1 (NFEN1), and the timer input of channel 1 is set to EVENTC event input signal through timer input selection register 0 (TIS0).

3. If you want to enter deep sleep when the real-time output mode of D/A conversion is active, you must disable event linking for EVENTC before entering deep sleep mode.

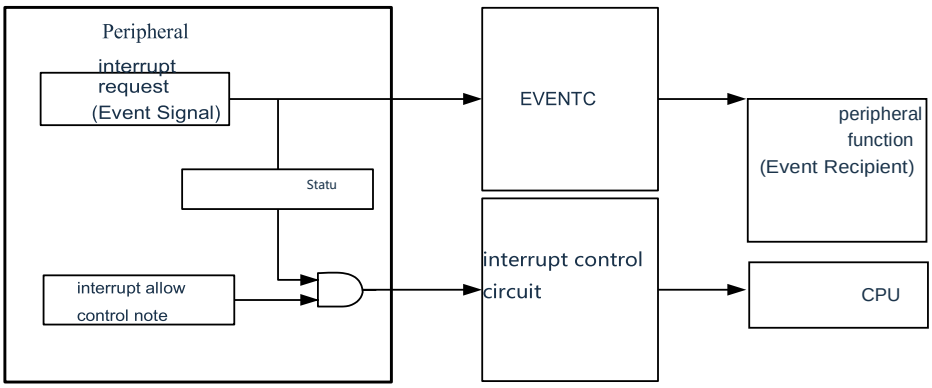
24.4 Operation of EVENTC

The path used for the interrupt request of the interrupt control circuit and the path used for the EVENTC event are independent of each other. Therefore, each event signal is independent of interrupt control and can be used as an event signal for peripheral function operation of the event receiver.

The relationship between interrupt handling and EVENTC is shown in Figure 24-3. This diagram is a relation that takes as an example peripheral functions with interrupt request status flags and interrupt allowed bits (control allowed or disabled).

The operation of a peripheral function accepting an event through EVENTC is based on the operation of the recipient peripheral function after the event was received (reference to the setting value of the ELSELRn register (n=00~21) and the corresponding to the operation of the linked target peripheral function accepting the event").

Figure 24-3 Relationship between interrupt handling and EVENTC



Note Some peripheral features do not have this feature.

The response of the peripheral function accepting the event is shown in Table 24-4.

Table 24-4 Accept the response of the event's peripheral function

Event Acceptance Target No.	Function of Event Link Target	Run after event acceptance	response
1	A/D converter	A/D conversion	The EVENTC event changes directly to the hardware trigger of the A/D conversion.
2	Timer4 Timer input for channel 0	External event counter for measuring input pulse width of delay counter	Edge detection is perform after 3 or 4 fCLK cycles from that EVENTC event occur.
3	Timer4 Timer input for channel 1	External event counter for measuring input pulse width of delay counter	Edge detection is perform after 3 or 4 fCLK cycles from that EVENTC event occur.
4	Timer A	count source	The EVENTC event becomes the count source for timer A directly.
5	Timer B	Input capture for TBIOB	The count is triggered upon generation after 2 or 3 fCLK cycles from that EVENTC event.
6	Timer M	Input capture for TMIOD0	A count start trigger is generated after a running clock cycle of two or three timers M has occurred EVENTC event.
		Forced Cutoff of Pulse Output	The EVENTC event is changed to a forced cutoff state after a running clock cycle of two or three timers M.
7	Timer M	Input capture for TMIOD1	A count start trigger is generated after a running clock cycle of two or three timers M has occurred EVENTC event.
		Forced Cutoff of Pulse Output	The EVENTC event is changed to a forced cutoff state after a running clock cycle of two or three timers M.
8	Channel 0 of the D/A converter	Live Out (Channel 0)	The D/A transition to channel 0 begins after 2 or 3 fCLK cycles from the occurrence of a EVENTC event.
9	Channel 1 for D/A Converter	Live Out (Channel 1)	The D/A conversion of channel 1 begins after 2 or 3 fCLK cycles from the occurrence of the EVENTC event.

Chapter 25 Interrupt function

The Cortex-M0+ processor has a built-in nested vector interrupt controller (NVIC) that supports up to 32 interrupt request (IRQ) inputs and 1 unmaskable interrupt (NMI) input, plus multiple internal exceptions.

The system extends 32 interrupt request (IRQ) inputs and 1 unmaskable interrupt (NMI) input to support up to 64 interrupt sources and one unmaskable interrupt source. This User Handbook describes only the extended features in this system. For the Cortex-M0+ processor with built-in NVIC features, refer to the User Handbook for Cortex-M0+ processors.

The actual number of interrupt sources varies by product.

		24pin	32 Pin	36 Pin	44pin	40,48,52 pins	64 pins
maskable interrupt	External	9	12	11	10	12	13
	Internal	33	33	33	33	33	33

25.1 Types of interrupt functionality

There are two types of interrupts.

(1) maskable interrupt

This is a blocked interrupt. If that interrupt mask register is not open, the interrupt request will not be respond even if it is generated.

It can generate standby release signal, release deep sleep mode and sleep mode.

The masking interrupt is divided into external interrupt request and internal interrupt request.

(2) unmaskable interrupt

This is an interrupt that does not accept mask control and the CPU must respond once the interrupt request is generated.

25.2 interrupt source and structure

Interrupt source list reference table 25-1.

Table 25-1 List of interrupt sources (1/4)

interrupt handling	interrupt source	interrupt source		Internal/External	basic structure Type Note 1	64 Pin	52 Pin	48 pins	44 pins	40 Pin	36 Pin	32 Pin	24 pins
		Name	trigger										
masking	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	INTP4	Detection of Pin Input Edges			○	○	○	○	○	○	○	○
	6	INTP5	Detection of Pin Input Edges			○	○	○	○	○	○	○	○
	7	INTST2/INTSSPI20/INTEGER20	End of transmission of UART2 transmission or end of transmission of buffer interrupt /SSPI20 or end of transmission of buffer	Internal	(A)	○	○	○	○	○	○	○	○
	8	INTESR2/INTESSPI21/INTEGER21	Transfer end of UART2 receipt /Transfer end of SSPI21 or buffer interrupt /IIC21 transfer end			○	○	○	○	○ Note 3	○ Note 3	○ Note 3	○ note3
	9	INTSRE2	Communication error occurred for UART2 receipt			○	○	○	○	○	○	○	○
	10	INTEST0/INTESSPI00/INTEGER00	End of transmission of UART0 transmission or end of transmission of buffer interrupt /SSPI00 or end of transmission of buffer			○	○	○	○	○	○	○	○
	11	INTESR0/INTESSPI01/INTEGER01	Transfer end of UART0 receipt /Transfer end of SSPI01 or buffer interrupt /IIC01 transfer end			○	○	○	○	○ Note 4	○ Note 4	○ Note 4	○ note4
	12	INTSRE0	Communication error occurred for UART0 receipt			○	○	○	○	○	○	○	○

Note 1. Basic constituent types (A) to (D) correspond to (A) to (D) of FIG.

2. This is the case where the bit7 (LVIMD) of the voltage detection level register (LVIS) is set.

3. INTSR2 only.

4. INTSR0 only.

Table 25-1 List of interrupt sources (2/4)

interrupt handling	interrupt source	interrupt source		Internal/External	basic structure Type Note 1	64 Pin	52 Pin	48 pins	44 pins	40 Pin	36 Pin	32 Pin	24 pins
		Name	trigger										
masking	13	INTST1/INTSSPI10/INTEGER10	End of transmission of UART1 transmission or end of transmission of buffer interrupt /SSPI10 or end of transmission of buffer interrupt /IIC10	Internal	(A)	○	○ Note 2	○ Note 2	○ Note 2	○ Note 2	○ Note 2	○ Note 2	○ Note 2
	14	INTSR1/INTSSPI11/INTEGER11	Transfer end of UART1 receipt /Transfer end of SSPI11 or buffer interrupt /IIC11 transfer end			○	○	○	○	○	○	○	○
	15	INTSRE1	Communication error occurred for UART1 receipt			○	○	○	○	○	○	○	○
	16	INTICA0	IICA0 communication ended			○	○	○	○	○	○	○	○
	17	INTTM00	Timer channel 00's count or capture ends			○	○	○	○	○	○	○	○
	18	INTTM01	Timer channel 01 count end or capture end			○	○	○	○	○	○	○	○
	19	INTTM02	Timer channel 02's count or capture ends			○	○	○	○	○	○	○	○
	20	INTTM03	Timer channel 03's count or capture ends			○	○	○	○	○	○	○	○
	21	INTAD	A/D conversion ended			○	○	○	○	○	○	○	○
	22	INTRTC	Fixed period / for real-time clock alarm clock consistent detection			○	○	○	○	○	○	○	○
	23	INTKR	Detection of key return signal	External	(C)	○	○	○	○	○	-	-	-
	24	INTCMP0	Comparator Detection 0	Internal	(A)	○	○	○	○	○	○	○	○
	25	INTCMP1	Comparator Detection 1			○	○	○	○	○	○	○	○
	26	INTTMA	Underflow of timer A			○	○	○	○	○	○	○	○
	27	INTTMM0	Input capture, comparison matching, overflow, and overflow interrupt of timer M0			○	○	○	○	○	○	○	○
	28	INTTMM1	Input capture, comparison matching, overspill, underspill interruption of timer M1			○	○	○	○	○	○	○	○
	29	INTMB	input capture, comparison matching, overflow, overflow interrupt of timer B			○	○	○	○	○	○	○	○
	30	INTMC	Overflow of timer C			○	○	○	○	○	○	○	○
	31	INTFL	Flash programming is over			○	○	○	○	○	○	○	○

Note1. Basic composition types (A) ~ (D) correspond to (A) ~ (D) in Figure 25-1 respectively.

2. INTST1 only.

Table 2 5-1: List of interrupt sources (3/4)

interrupt handling	interrupt source Number	interrupt source		Internal/External	basic structure Type Note 1	64 Pin	52 Pin	48 pins	44 pins	40 Pin	36 Pin	32 Pin	24 pins
		Name	trigger										
masking	32	Retention	-	-	-	-	-	-	-	-	-	-	-
	33	INTP6	Detection of Pin Inputs	External	(B)	○	○	○	-	○	○	○	-
	34	INTP7				○	-	-	-	-	-	○	-
	35	INTP8				○	○	○	○	○	○	○	○
	36	INTP9				○	○	○	-	○	-	-	-
	37	INTP10				○	○	○	○	○	○	○	○
	38	INTP11				○	○	○	○	○	○	○	○
	39	Retention	-	-	-	-	-	-	-	-	-	-	-
	40	Retention	-	-	-	-	-	-	-	-	-	-	-
	41	INTC0ERR	CAN error interrupt	Internal	(A)	○	○	○	○	○	○	○	○
	42	Retention	-	-	-	-	-	-	-	-	-	-	-
	43	Retention	-	-	-	-	-	-	-	-	-	-	-
	44	INTTM01H	Timer channel 01 count end or capture end (High 8-bit timer operation)	Internal	(A)	○	○	○	○	○	○	○	○
	45	Retention	-	-	-	-	-	-	-	-	-	-	-
	46	Retention	-	-	-	-	-	-	-	-	-	-	-
	47	INTTM03H	Timer channel 03's count or capture ends (High 8-bit timer operation)	Internal	(A)	○	○	○	○	○	○	○	○
	48	INTDIV	Divider calculation complete	Internal	(A)	○	○	○	○	○	○	○	○
	49	Retention	-	-	-	-	-	-	-	-	-	-	-
	50	Retention	-	-	-	-	-	-	-	-	-	-	-
	51	Retention	-	-	-	-	-	-	-	-	-	-	-
	52	Retention	-	-	-	-	-	-	-	-	-	-	-
	53	Retention	-	-	-	-	-	-	-	-	-	-	-
	54	INTIT	Detection of interval signals	Internal	(A)	○	○	○	○	○	○	○	○
	55	INTC0REC	CAN Receive End	Internal	(A)	○	○	○	○	○	○	○	○
	56	INTC0WOUP	CAN Wake	Internal	(A)	○	○	○	○	○	○	○	○
	57	INTC0TRX	End of CAN send	Internal	(A)	○	○	○	○	○	○	○	○
	58	Retention	-	-	-	-	-	-	-	-	-	-	-
	59	Retention	-	-	-	-	-	-	-	-	-	-	-
	60	Retention	-	-	-	-	-	-	-	-	-	-	-
	61	Retention	-	-	-	-	-	-	-	-	-	-	-
	62	Retention	-	-	-	-	-	-	-	-	-	-	-
	63	Retention	-	-	-	-	-	-	-	-	-	-	-

Note1. Basic composition types (A) ~ (D) correspond to (A) ~ (D) in Figure 25-1 respectively.

Table 25-1 List of interrupt sources (4/4)

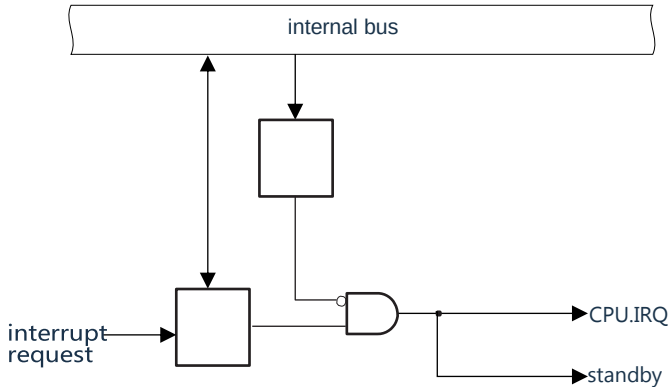
interrupt handling	interrupt source number	interrupt source		Internal/External	Basic Structure Type Note 1
		Name	trigger		
unmaskable	—	INTWDT	watchdog timer interval interrupt note 2	Internal	(D)

Note: 1. Basic constituent types (A) to (D) correspond to (A) to (D) of Fig. 25-1.

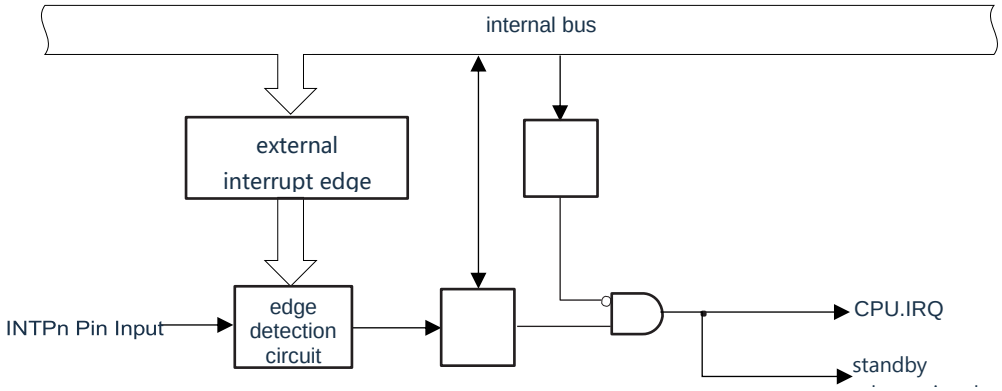
2. This is the case where the bit7(WDTINT) of option bytes (000C0H) is set.

Figure 25-1 Basic structure of interrupt functionality

(A) Internally masked interrupt



(B) Externally Masked Interrupt (INTPn)



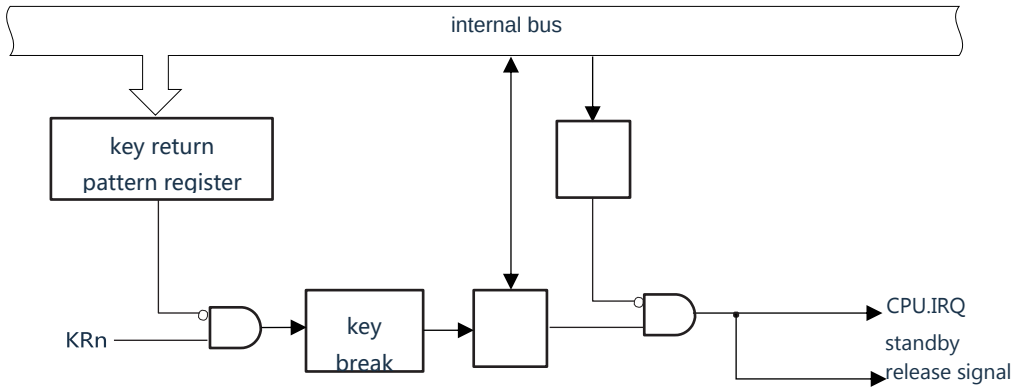
Note:32 Pin: n=0~8,10~11

36 Pin: n=0~6,8,10~11

40, 48, 52 Pin: n=0~6,8~11

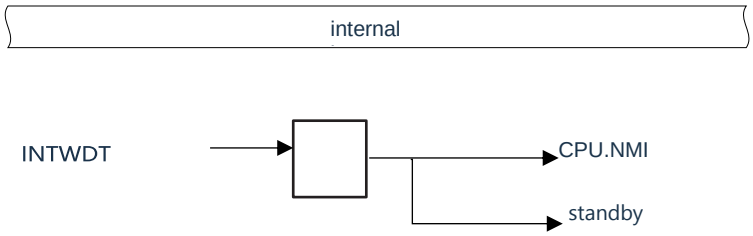
64 Pin: n=0~11

(C) Externally Masked Interrupt (INTKR)



NOTE: 40 Pin: n=0, 2~5
48 Pin: n=0~5
52,64 pins: n=0~7

(D) Unmaskable interrupt



Note: The interrupt request flag IF for unmaskable interrupts does not have a physical register and cannot generate an interrupt request through a bus read-write register.

25.3 Register for controlling interrupt function

Interrupt functionality is controlled through the following 4 registers.

- Interrupt request flag register (IF00~IF31)
- interrupt mask register (MK00~MK31)
- External interrupt rising edge allow registers (EGP0, EGP1)
- External interrupt descend edge permit registers (EGN0, EGN1)

25.3.1 interrupt request flag register (IF00~IF31)

The interrupt request flag is set to '1' by generating a corresponding interrupt request or execution instruction. The interrupt request flag is clear" 0" by generate a reset signal or executing instructions.

The IF00L~IF31L, IF00H~IF31H register, or the IF00~IF31 register are set by the 8-bit memory operation instruction.

After the reset signal is generated, the value of these registers changes to '000_0000H'.

Figure 2 5-2 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
IFmH	Reserved							IFH
	7	6	5	4	3	2	1	0
IFmL	Reserved							IFL

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.

IFmH	Interrupt Request Flags for Interrupt Sources Numbered 32-63
0	No interrupt request signal is generated.
1	An interrupt request is generated in the interrupt request state.

Note: 1. The source of the interrupt and the interrupt request flag register are shown in Table 25-2.
2. The interrupt request flag register corresponds to CPU. IRQ, see Figure 25-4.
3. Interrupt request flag registers do not clear from zero, must write 0 to register after interrupt response.

25.3.2 interrupt mask register (MK00~MK31)

The interrupt mask setting allows or disables the corresponding masked interrupt handling.

The MK00L~MK31L, MK00H~MK31H register, or the MK00~MK31 register are set by the 8-bit memory operation instruction.

After the reset signal is generated, the values of these registers become "FFFF_FFFF".

FIG. 25-3 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
MKmH	Reserved							MKH
	7	6	5	4	3	2	1	0
MKmL	Reserved							MKL

MKmL	Interrupt Handling Control Note 1 for Interrupt Sources Numbered 0 to 31
0	Allow interrupt handling.
1	Disable interrupt handling.

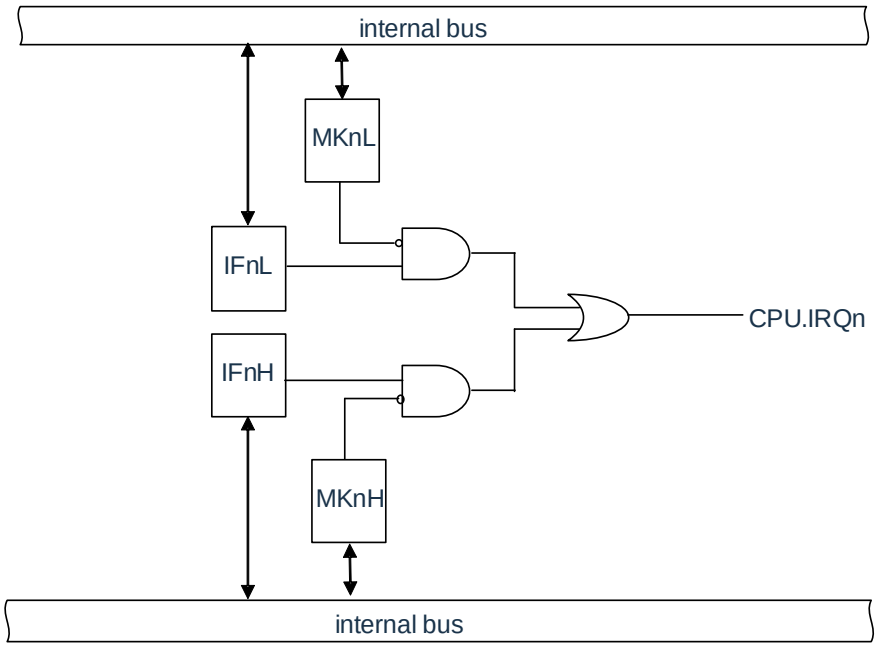
MKmH	Interrupt Handling Control Note 2 for Interrupt Sources Numbered 32 to 63
0	Allow interrupt handling.
1	Disable interrupt handling.

Note: 1. The corresponding relationship between interrupt source and interrupt request mask register is shown in Table 25-2.
2. The corresponding relationship between interrupt request mask register and CPU. IRQ is shown in Figure 25-4.

Table 25-2 The Corresponding Relation of Interrupt Source and Flag Register

Number	interrupt source	interrupt request flag register	interrupt mask register	Number	interrupt source	interrupt request flag register	interrupt mask register
0	INTLVI	IF00.IFL	MK00.MKL	32	Retention	IF00.IFH	MK00.MKH
1	INTP0	IF01.IFL	MK01.MKL	33	INTP6	IF01.IFH	MK01.MKH
2	INTP1	IF02.IFL	MK02.MKL	34	INTP7	IF02.IFH	MK02.MKH
3	INTP2	IF03.IFL	MK03.MKL	35	INTP8	IF03.IFH	MK03.MKH
4	INTP3	IF04.IFL	MK04.MKL	36	INTP9	IF04.IFH	MK04.MKH
5	INTP4	IF05.IFL	MK05.MKL	37	INTP10	IF05.IFH	MK05.MKH
6	INTP5	IF06.IFL	MK06.MKL	38	INTP11	IF06.IFH	MK06.MKH
7	INTST2/INTSSPI20/INTEGER20	IF07.IFL	MK07.MKL	39	Retention	IF07.IFH	MK07.MKH
8	INTESR2/INTESSPI21/INTEGER	IF08.IFL	MK08.MKL	40	Retention	IF08.IFH	MK08.MKH
9	INTSRE2	IF09.IFL	MK09.MKL	41	INTC0ERR	IF09.IFH	MK09.MKH
10	INTEST0/INTESSPI00/INTEGER00	IF10.IFL	MK10.MKL	42	Retention	IF10.IFH	MK10.MKH
11	INTESR0/INTESSPI01/INTEGE	IF11.IFL	MK11.MKL	43	Retention	IF11.IFH	MK11.MKH
12	INTSRE0	IF12.IFL	MK12.MKL	44	INTTM01H	IF12.IFH	MK12.MKH
13	INTST1/INTSSPI10/INTEGER1	IF13.IFL	MK13.MKL	45	Retention	IF13.IFH	MK13.MKH
14	INTSR1/INTSSPI11/INTEGER1	IF14.IFL	MK14.MKL	46	Retention	IF14.IFH	MK14.MKH
15	INTSRE1	IF15.IFL	MK15.MKL	47	INTTM03H	IF15.IFH	MK15.MKH
16	INTICA0	IF16.IFL	MK16.MKL	48	INTDIV	IF16.IFH	MK16.MKH
17	INTTM00	IF17.IFL	MK17.MKL	49	Retention	IF17.IFH	MK17.MKH
18	INTTM01	IF18.IFL	MK18.MKL	50	Retention	IF18.IFH	MK18.MKH
19	INTTM02	IF19.IFL	MK19.MKL	51	Retention	IF19.IFH	MK19.MKH
20	INTTM03	IF20.IFL	MK20.MKL	52	Retention	IF20.IFH	MK20.MKH
21	INTAD	IF21.IFL	MK21.MKL	53	Retention	IF21.IFH	MK21.MKH
22	INTRTC	IF22.IFL	MK22.MKL	54	INTIT	IF22.IFH	MK22.MKH
23	INTKR	IF23.IFL	MK23.MKL	55	INTC0REC	IF23.IFH	MK23.MKH
24	INTCMP0	IF24.IFL	MK24.MKL	56	INTC0WOU	IF24.IFH	MK24.MKH
25	INTCMP1	IF25.IFL	MK25.MKL	57	INTC0TRX	IF25.IFH	MK25.MKH
26	INTTMA	IF26.IFL	MK26.MKL	58	Retention	IF26.IFH	MK26.MKH
27	INTTMM0	IF27.IFL	MK27.MKL	59	Retention	IF27.IFH	MK27.MKH
28	INTTMM1	IF28.IFL	MK28.MKL	60	Retention	IF28.IFH	MK28.MKH
29	INTMB	IF29.IFL	MK29.MKL	61	Retention	IF29.IFH	MK29.MKH
30	INTMC	IF30.IFL	MK30.MKL	62	Retention	IF30.IFH	MK30.MKH
31	INTFL	IF31.IFL	MK31.MKL	63	Retention	IF31.IFH	MK31.MKH

Figure 25-4 Relationship of Flag Registers to CPU.IRQ



25.1.1 external interrupt rising edge permit register (EGP0, EGP1), external interrupt falling edge permit register (EGN0, EGN1)

These registers set the effective edge of INTP0~INTP11.

The EGP0, EGP1, EGN0, EGN1 registers are set by 8-bit memory operation instructions. After the reset signal is generated, the values of these registers change to '00H'.

Figure 25-5 Format of external interrupt rising edge permit registers (EGP0, EGP1) and external interrupt falling edge permit registers (EGN0, EGN1)

Address: 40045B38H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: 40045B39H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: 40045B 3AH After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	EGP11	EGP10	EGP9	EGP8

Address: 40045B 3BH After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	EGN11	EGN10	EGN9	EGN8

EGPn	EGN	Effective edge selection for INTPn pins (n=0~11)
0	0	Edges are not detected.
0	1	descending edge
1	0	rising edge
1	1	Rising and falling along the bilateral

The ports corresponding to the EGPn and EGNn bits are shown in Table 25-3.

Table 25-3 Interrupt request signal corresponding to EGPn and EGNn bits

Detect allowed bits		interrupt request signal	64 pins	52,48,40 pins	36 Pin	32 Pin
EGP0	EGN0	INTP0	○	○	○	○
EGP1	EGN1	INTP1	○	○	○	○
EGP2	EGN2	INTP2	○	○	○	○
EGP3	EGN3	INTP3	○	○	○	○
EGP4	EGN4	INTP4	○	○	○	○
EGP5	EGN5	INTP5	○	○	○	○
EGP6	EGN6	INTP6	○	○	○	○
EGP7	EGN7	INTP7	○	—	—	○
EGP8	EGN8	INTP8	○	○	○	○
EGP9	EGN9	INTP9	○	○	—	—
EGP10	EGN10	INTP10	○	○	○	○
EGP11	EGN11	INTP11	○	○	○	○

Note If you switch the input port used by the external interrupt function to output mode, a INTPn interrupt may occur due to detection of valid edges. When switching to output mode, the port mode register (PMxx) must '0' after prohibiting edge detection (EGPn, EGNn=0, 0).

Note 1. Refer to "2.1 Port Features" for edge detection.

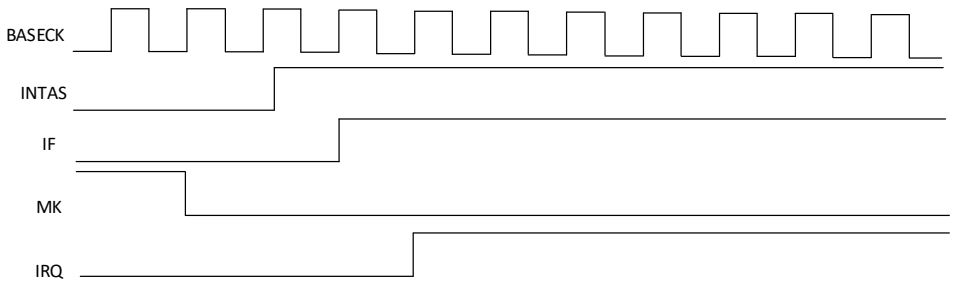
2.n=0~11

25.4 Operation of interrupt handling

25.4.1 Acceptance of maskable interrupt requests

If the interrupt request flag is set to '1' and the masking (MK) flag of the interrupt request is cleared to '0', the interrupt request can be passed to NVIC.

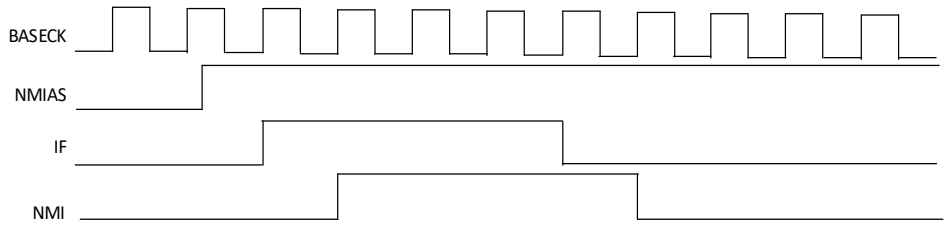
From the interrupt request flag set 1 to the IRQ of the CPU set 1, only 1 clock is required.



25.4.2 Acceptance of Unmaskable Interrupt Request

If a non-maskable interrupt request is generate, that interrupt request flag will be placed " 1 " and passed directly to the NVIC.

From the interrupt request flag set 1 to the NMI of the CPU set 1, only 1 clock is required.



Chapter 26 key interrupt function

The number of input channels for the key interrupts is different based on the product.

26.1 Function of key interrupt

A key interrupt (INTKR) can be generated by inputting a descending edge to key interrupt input pins (KR0 to KR7).

Table 2 6-1 Key interrupt detection pin assignment

key interrupt pin	Key Return Mode Register (KRM)
KR0	KRM0
KR1	KRM1
KR2	KRM2
KR3	KRM3
KR4	KRM4
KR5	KRM5
KR6	KRM6
KR7	KRM7

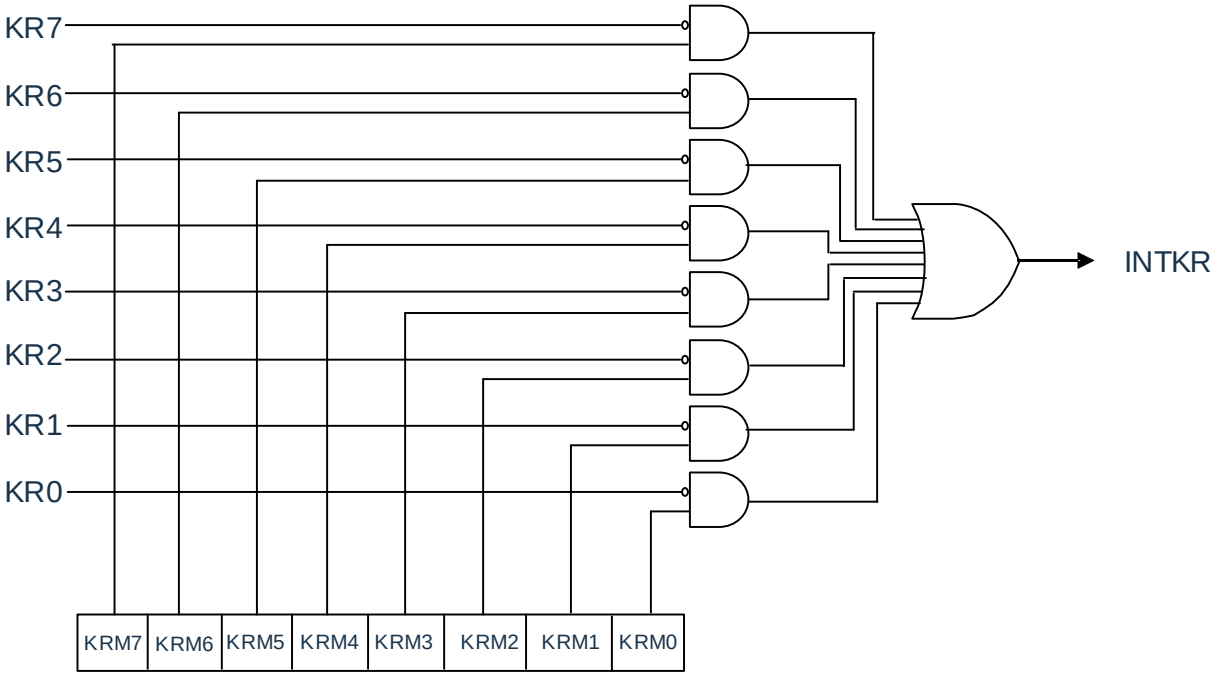
26.2 Structure of key break

Key interrupts consist of the following hardware.

Table 2 6-2 Structure of key break

Project	control register
control register	Key Return Mode Register (KRM) Port Mode Register (PMx)

Figure 2 6-1 Box for key break



Key return mode register

26.3 Register for controlling key interrupt

The key interrupt function is controlled through the following registers.

- Key Return Mode Register (KRM)
- Port Mode Register (PMx)

26.3.1 Key Return Mode Register (KRM)

The KRM0~KRM7 bit controls the KR0~KR7 signal.

The KRM register is set by an 8-bit memory operation instruction.

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

Figure 2 6-2 Format of Key Return Mode Register (KRM)

Address: 40044B37H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	Control of key interrupt mode
0	The key interrupt signal is not detected.
1	A key interrupt signal is detected.

Note 1. Internal pull resistors can be used by interrupting the key to the object position "1" of the pull-up resistor register.

2. An interrupt is generated if the object position bit of the KRM register is made in a state where a low level is input to the key interrupt input pin.

To ignore this interrupt, you must set the KRM register after the interrupt mask prevents the interrupt from handling. The interrupt request flag must then be cleared after waiting for the $_{key}$ interrupt input low level width (tKR) (refer to the data manual) to allow interrupt handling.

3. Pins not used in key break mode can be used as the usual port.

Notes 1.n=0~7

26.3.2 Port Mode Register (PMx)

When used as key interrupt input pin (KR0~KR7), the PMxn bits must be set 1. In this case, the output latch for Pxn may be "0" or "1".

The PMx register is set by an 8-bit memory operation instruction.

After the reset signal is generated, the value of this register changes to FFH.

The internal pull-up resistance can be used in bit by a pull-up resistance selection register (PUx).

For the format of the port mode register, refer to "2.3.1 Port Mode Register (PMxx)".

Chapter 27 standby function

27.1 standby function

The standby function is the function of further reducing the working current of the system, there are two modes below.

(1) sleep mode

The sleep mode is a mode that stops the CPU running the clock. If a high speed system clock oscillation circuit, a high speed internal oscillator or a sub-system clock oscillation circuit are oscillating before setting a sleep mode, each clock continues to oscillate. While this mode does not allow the operation current to drop to the level of a deep sleep mode, it is a valid mode when you want to restart processing immediately through interrupt requests or when you want to perform frequent intermittent runs.

(2) deep sleep mode

A deep sleep mode is a mode in which the oscillation of a high speed system clock oscillation circuit and a high speed internal oscillator are stopped and the whole system is stopped. The invention can greatly reduce the working current of the CPU.

Since that deep sleep mode can be released by interrupt request, intermittent operation can also be perform. However, in that case of the X1 clock, since a wait time for ensure oscillation stability is required to cancel the deep sleep mode, a sleep mode must be selected if immediate processing by interrupt request is required.

In either mode, the register, flag and data storage are all maintained as content before the standby mode and also the output latches and output buffers of input/output ports.

Note 1. Deep sleep mode is only available when the CPU is running at the main system clock. The CPU cannot be set to deep sleep mode when running at the secondary system clock. Sleep mode is used regardless of whether the CPU is running at the primary system clock or the secondary system clock.

2. When transitioning to deep sleep mode, the WFI instruction must be executed after the peripheral hardware is stopped running at the main system clock.

3. In order to reduce the operation current of A/D converter, the bit7 (ADCS) and bit0 (ADCE) of A/D converter mode register 0 (ADM0) must be cleared to "0", and the WFI instruction must be executed after stopping the A/D converter running.

4. It is possible to choose whether to continue or stop the oscillation of the low-speed internal oscillator in the sleep mode or the deep sleep mode by the option byte. Refer to "Chapter 33 Option Bytes" for details.

27.2sleep mode

27.2.1 Sleep mode configuration

When the SLEEPDEEP bit of the SCR register is 0, the WFI instruction is executed and the sleep mode is started. In sleep mode, that CPU stop acting, but the value of the internal register is still maintain, and the peripheral module remain in the state before entering sleep mode. The states of peripheral modules, vibrators, etc. in sleep mode are shown in Table 27-1.

Sleep mode can be set whether the CPU clock before setting is a high speed system clock or a high speed internal oscillator clock or a sub-system clock.

Note When the interrupt mask flag is "0" (which allows interrupt handling) and the interrupt request flag is "1" (which generates an interrupt request signal), the interrupt request signal is used to deactivate sleep mode. Therefore, even if the WFI command is executed in this case, it does not transfer to sleep mode.

Table 27-1 Health in Sleep Mode (1/2)

Sleep mode settings / Project			A case where a WFI instruction is executed while the CPU is running at the main system clock		
			CPU uses high speed internal oscillator clock ($_{fIH}$) Run	The CPU is running in an $X_{1\ clock}$ (fX)	CPU External Main System Clock ($_{fEX}$) Run
system clock			Stop providing clock to CPU.		
	main system clock	fIH	Continue running (cannot stop).	No Operation.	
		fX	No Operation.	Continue running (cannot stop).	Cannot run.
		fEX		Cannot run.	Continue running (cannot stop).
	sub-system clock	fXT	Keep the state set before sleep mode.		
		fEXS			
	low-speed internal oscillation instrument clock	fIL	bit0 (WDSTBYON) and bit4 (WDTON) via option bytes (000C0H) and sub-system clockticks Set for the WUTMMCK0 bit of the mode control register (OSMC). WUTMCK0=1: oscillation 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 while DMA is running).		
Port (latch)			Keep the state set before sleep mode.		
DIV			It works.		
Timer4					
Real-time clock (RTC)					
15-bit interval timer					
watchdog timer			Refer to the Chapter 14 watchdog timer for reference.		
Timer A			It works.		
Timer M					
Timer B					
timer C					
Clock Out/Buzzer Out					
A/D converter					
D/A converter					
comparator					
universal serial communication unit (SCI)					
Serial Interface (IICA)					
FCAN					
Data transfer controller (DMA)					
linkage controller			Links between functional blocks that can run.		

power-on reset function		It works.
voltage detection function		
external interrupt		
key interrupt function		
CRC operation Features	high speed CRC	
	universal CRC	When DMA is performed in the operation of the RAM area, it can run.
RAM Parity		Can be run while performing DMA.
SFR Protection Features		

Remark Stop Running: Automatically stops running when transferred to sleep mode.

Disable Run: Stop running before transferring to sleep mode.

f_{IH}: high speed internal oscillator clock f_{IL}: low-speed internal oscillator clock

f_X: X1 clock f_{EX}: External Master System

f_{XT}:XT1 clock f_{EXS}: external sub-system

Table 27-1 Health in Sleep Mode (2/2)

Project \ Sleep mode settings			A case where a WFI instruction is executed while the CPU is running at a sub-system clock	
			The CPU runs in XT1 clock (fXT).	CPU runs with external secondary system clock (fEXS)
system clock			Stop providing clock to CPU.	
	main system clock	fIH	No Operation.	
		fX		
		fEX		
	sub-system clock	fXT	Continue running (cannot stop).	Cannot run.
		fEXS	Cannot run.	Continue running (cannot stop).
	low-speed internal oscillation instrument clock	fIL	bit0 (WDSTBYON) and bit4 (WDTON) via option bytes (000C0H) and sub-system clockticks Set for the WUTMMCK0 bit of the mode control register (OSMC). • WUTMCK0=1: oscillation • WUTMMCK0=0 and WDTON=0: Stop • WUTMMCK0=0, WDTON=1 and WDSTBYON=1: oscillation • WUTMMCK0=0, WDTON=1 and WDSTBYON=0: Stop	
CPU			Stop running.	
code flash				
RAM			Stop running (can run while DMA is running).	
Port (latch)			Keep the state set before sleep mode.	
DIV			When RTCLPC=0, can be run (otherwise it is prohibited).	
Time4			When RTCLPC=0, can be run (otherwise it is prohibited).	
Real-time clock (RTC)			It works.	
15-bit interval timer				
watchdog timer			Refer to the "Chapter 14 watchdog timer.".	
Timer A			When RTCLPC=0, can be run (otherwise it is prohibited).	
Timer M				
Timer B				
timer C				
Clock Out/Buzzer Out				
A/D converter			No Operation.	
D/A converter			When RTCLPC=0, can be run (otherwise it is prohibited).	
comparator			It works.	
universal serial communication unit (SCI)			When RTCLPC=0, can be run (otherwise it is prohibited).	
Serial Interface (IICA)			No Operation.	
FCAN			When RTCLPC=0, can be run (otherwise it is prohibited).	
Data transfer controller (DMA)			When RTCLPC=0, can be run (otherwise it is prohibited).	
linkage controller			Links between functional blocks that can run.	
power-on reset function			It works.	
voltage detection function				
external interrupt				
key interrupt function				
CRC operation Features	high speed CRC		No Operation.	
	universal CRC		When DMA is performed in the operation of the RAM area, it can run.	
RAM Parity Error Detection			Can be run while performing DMA.	
SFR Protection Features				

Remark Stop Running: Automatically stops running when transferred to sleep mode.

Disable Run: Stop running before transferring to sleep mode.

fIH: high speed internal oscillator clock fIL: low-speed internal oscillator clock

fX: X1 clock fEX: External Master System

fXT:XT1 clock fEXS: external sub-system

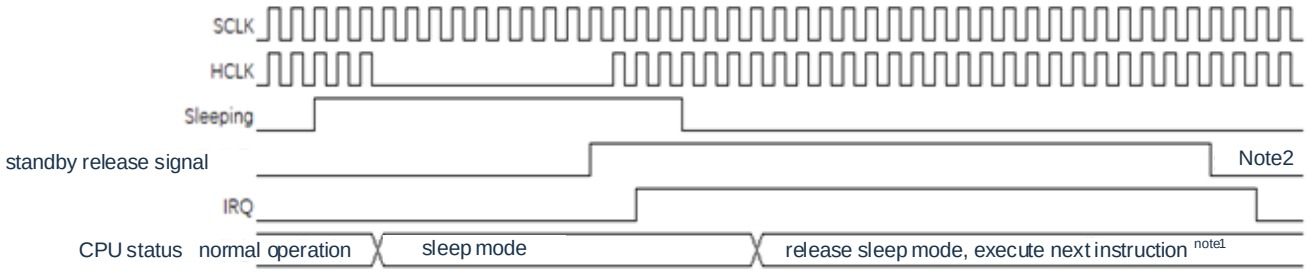
27.2.2 Exit from Sleep Mode

The sleep mode can be arbitrarily interrupted and the external reset terminal, the POR reset, the low voltage detection reset, the RAM parity error reset, the WDT reset and the software reset is released.

1) Discontinue by interrupt

When an unmaskable interrupt is generated, and in a state that interrupts are allowed to be accepted, the sleep mode is released, and the CPU starts processing the interrupt service program.

FIG. 27-1: Sleep mode release by interrupt request



Note:

① It takes 16 clocks from the standby shutdown signal to the shutdown of sleep mode and the start of the interrupt service program.

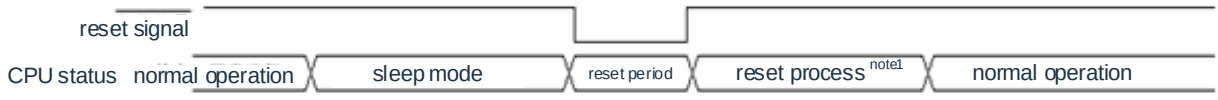
② The standby release signal cannot be cleared by itself, the register must be cleared. It is common to write register erase in interrupt service programs.

Note: Before entering sleep mode, only the mask bit corresponding to the interrupt that is expected to be used to relieve sleep mode should be zeroed out.

2) Release by reset

When a reset signal is generated, the CPU is in the reset state and the sleep mode is released. As with the usual reset, the program is executed after being transferred to the reset vector address.

FIG. 27-2: Sleep mode release by reset



Note 1: Refer to Chapter 28 Reset Features for reset processing times." For reset processing times of power-on reset (POR) circuits and voltage detection (LVD) circuits, refer to "Chapter 29 Power-on Reset Circuits".

27.3deep sleep mode

27.3.1 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, as well as the vibration generator are out of operation. However, the state of the CPU internal register value, RAM data, peripheral module, I/O is maintained. Peripheral modules, the vibration generator operation in deep sleep mode see Table 27-2.

Deep sleep mode can only be set if the pre-set CPU clock is the master system clock.

Note When the interrupt mask flag is "0" (which allows interrupt handling) and the interrupt request flag is "1" (which generates interrupt request signal). Therefore, if the WFI instruction is executed in this case, it is immediately released upon entering deep sleep mode. After executing the WFI instruction and after a deep sleep mode release time, return to the operation mode.

Table 2 7-2 Running State in Deep Sleep Mode

Settings for deep sleep mode / Project			A case where a WFI instruction is executed while the CPU is running at the main system clock		
			CPU uses high speed internal oscillator clock ($_{fIH}$) Run	The CPU is running in an X_1 clock (fX)	CPU External Main System Clock ($_{fEX}$) Run
system clock			Stop providing clock to CPU.		
	main system clock	fIH	Stop		
		fX			
		fEX			
	sub-system clock	fXT	Keep the state set before deep sleep mode.		
		fEXS			
	fIL		bit0 (WDSTBYON) and bit4 (WDTON) via option bytes (000C0H) and sub-system clockticks Set for the WUTMMCK0 bit of the mode control register (OSMC). WUTMCK0=1: oscillation 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)			Keep the state set before deep sleep mode.		
DIV			No Operation.		
timer array unit			No Operation.		
Real-time clock (RTC)			It works.		
15-bit interval timer					
watchdog timer			Refer to the "Chapter 14 watchdog timer.".		
Timer A			· Can run in event count mode when no TAIO input filter is selected. · When the sub-system clock is selected as the count source and the RTCLPC bit of the OSMC register is "0", it works. · When low-speed internal oscillator is selected as the counting source, it can run. · Beyond the above: No Operation.		
Timer M			No Operation.		
Timer B					
timer C					
Clock Out/Buzzer Out			When the secondary system clock is selected as the count clock and the RTCLPC bit is "0", it runs (otherwise it is disabled).		
A/D converter			Can wake up.		
D/A converter			Can run (keep set to state before deep sleep mode).		
comparator			Can run (only if no digital filter is used).		
universal serial communication unit (SCI)			Only SSPIp and UARTq wake up. Disable operation except SSPIp and UARTq.		
Serial Array Unit (IICA)			Can wake by address matching.		
FCAN			No Operation.		
Data transfer controller (DMA)			Can accept DMA boot source.		
linkage controller			Links between functional blocks that can run.		
power-on reset function			It works.		
voltage detection function					
external interrupt					
key interrupt function					

CRC operation Features	high speed CRC	Stop running.
	universal CRC	
RAM Parity		
SFR Protection Features		

Note 1. Stop running: Automatically stops running when transferred to deep sleep mode.

Disable Run: Stop running before transferring to deep sleep mode.

f_{IH} : high speed internal oscillator clock

f_{IL} : low-speed internal oscillator clock

f_X : X1 clock

f_{EX} : external main system clock

f_{XT} :XT1 clock

f_{EXS} : external sub-system clock

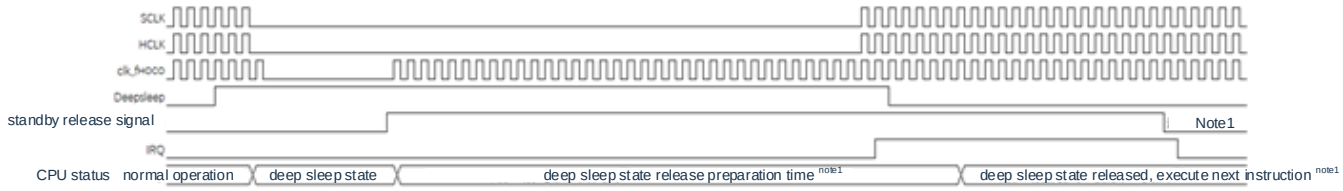
27.3.2 Exist form Deep sleep mode

You can exit from Deep sleep mode by two methods.

(a) Exit via unmaskable interrupt request

If an unmaskable interrupt request occurs, deep sleep mode is exited. After the oscillation stabilization time, if the state is allowed to accept the interrupt, the vector interrupt is processed. If you are in a state that prevents you from accepting interrupts, execute the instruction for the next address.

Figure 27-3 Deep sleep mode release by interrupt request



Note 1. Standby Release Signal: For more information on the standby release signal, refer to "Figure 25-1 Basic Structure of Interrupt Functionality."

2. Preparation time for deep sleep state release:

When the CPU clock is a high speed internal oscillation clock or external clock input before entering deep sleep mode:

At least 20μs

When the CPU clock is a high speed system clock (X1 oscillation) before entering deep sleep mode:

At least 20μs with a longer oscillation stabilization time (set by OSTS)

3. Wait: It takes 14 clock from CPU.IRQ validity to start executing interrupt service program.

Note: 1. Before entering sleep mode, only the mask bit corresponding to the interrupt that is expected to be used to relieve sleep mode should be zeroed out.

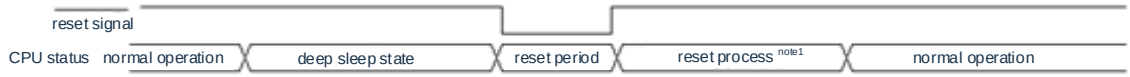
2. The CPU clock must be temporarily switched to the high speed internal oscillator clock before the WFI instruction is executed when the CPU is running at a high speed system clock (X1 oscillation) and the oscillation stability time after the deep sleep mode is removed is to be shortened.

Remark The oscillation accuracy of the high speed internal oscillator clock is stably waiting to change due to temperature conditions and deep sleep mode.

(b) Exit by generating a reset signal

Deep sleep mode is exited by generating a reset signal. Then, as with the usual reset, the program is executed after being transferred to the reset vector address.

Figure 27-4 Relieving deep sleep mode by resetting



Note Refer to "Chapter 28 Reset Features" for the reset processing time." For reset processing times of power-on reset (POR) circuits and voltage detection (LVD) circuits, refer to Chapter 29 Power-on Reset Circuits.

Chapter 28 reset function

The following 7 methods generate a reset signal.

(1) External reset via the RESETB pin.
(2) The inner reset is generated by the program runaway detection of the watchdog timer.
(3) Internal reset is generated by comparing the power supply voltage and the detection voltage of the POR circuit.
(4) Internal reset is generated by comparing the power supply voltage of the voltage detection circuit (LVD) with the detection voltage.
(5) Internal reset occurs because the system reset request register bit (AIRCR.SYSRESETREQ) is set to 1.
(6) Internal reset occurs due to RAM parity error.
(7) Internal reset occurs due to access to the illegal memory.

The internal reset and the external reset are the same, and after the reset signal is generated, the program is executed from the user-defined program start address.

When a low level is input to the RESETB pin, or the watchdog timer detects program runout, or the POR circuit and LVD circuit voltage.

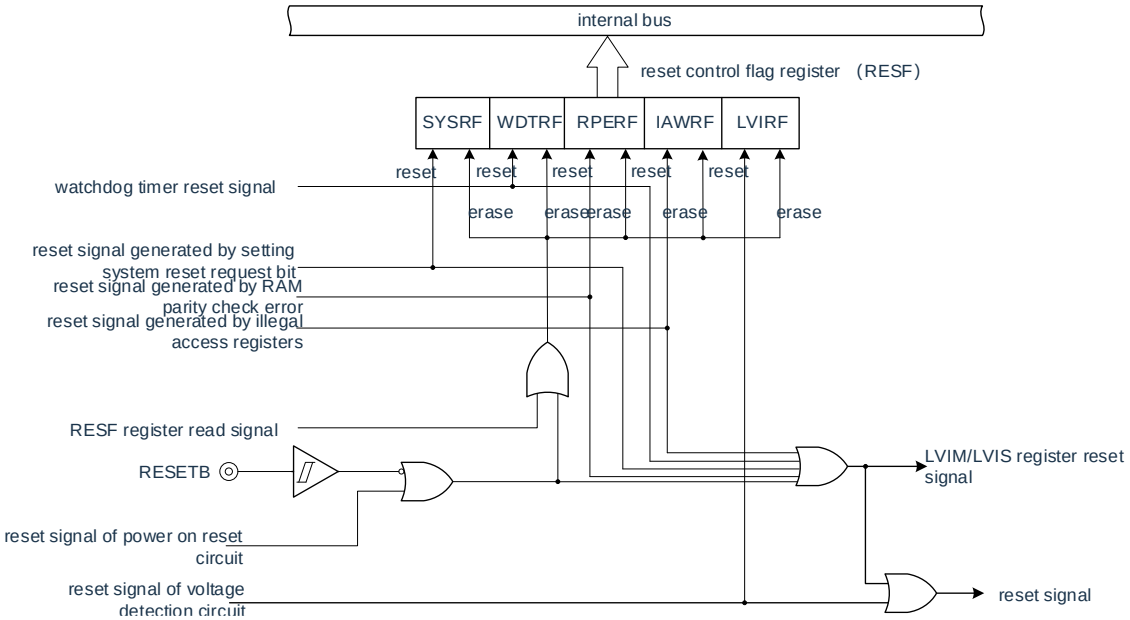
Note 1. When performing an external reset, you must input a low level of at least 10us to the RESETB pin. If an external reset is performed while the supply voltage is rising, power must be turned on after low input to the RESETB pin and at least 10us low within the operating voltage range indicated by the AC characteristics of the datasheet before high input.

2. The oscillation of the X1 clock, XT1 clock, high speed internal oscillator clock and low speed internal oscillator clock is stopped during reset signal generation. Invalid input for external master and external sub system clocks.

3. If a reset occurs, the SFRs are initialized so that the port pin becomes:

- P40: High impedance during external reset or POR reset. High level (internal pull-up resistance of connection) during other reset and after accepting reset.
- P130: A low level is output during and after the reset is accepted.
- P40, Ports other than P130: High impedance dure and after accepting reset.

Figure 28-1 Block diagram for reset function



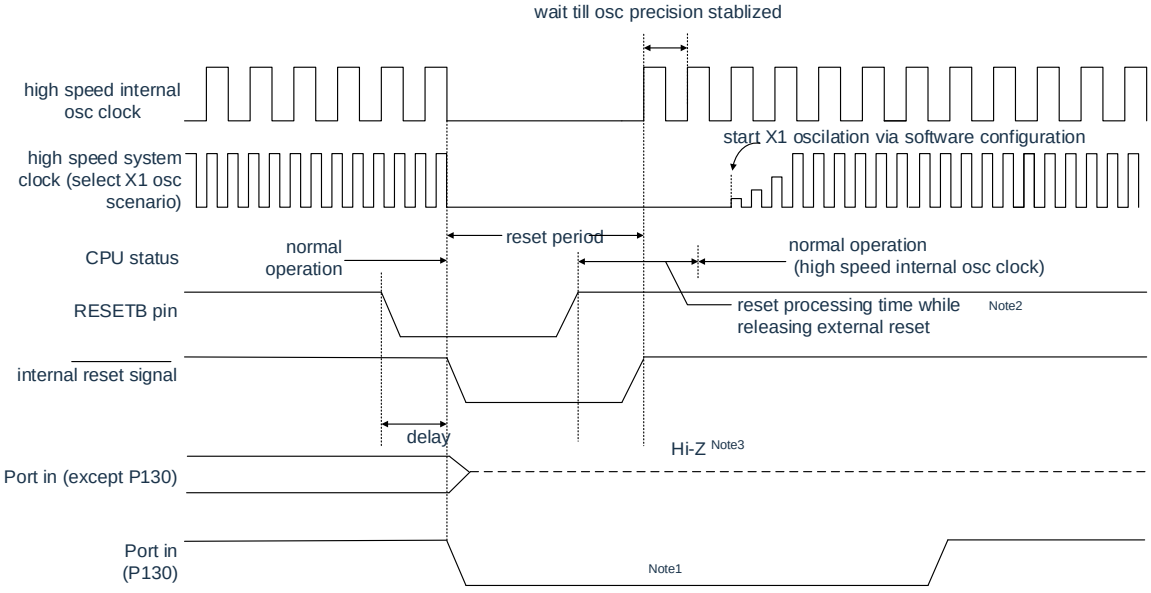
Note The internal reset of the LVD circuit does not reset the LVD circuit.

Note
1.LVIM: voltage detection register
2.LVIS: Voltage detection level register

reset sequence

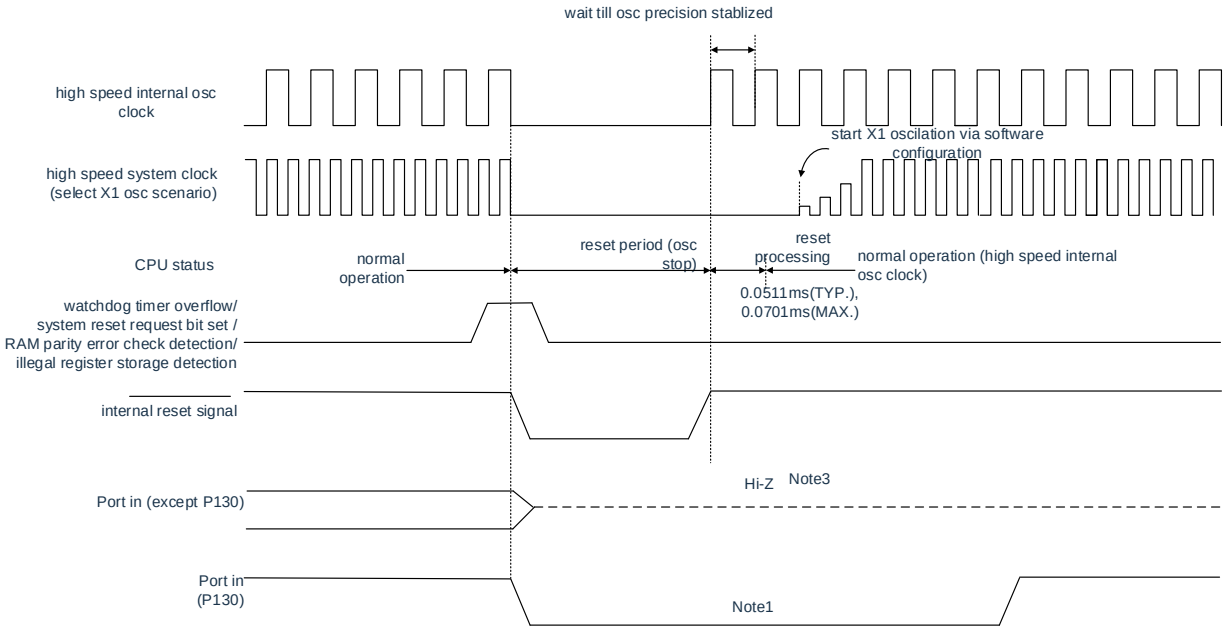
Reset occurs when a low level is input to the RESETB pin. Then, if a high level is inputted to the RESETB, the reset state is released and the program is started to execute with a high speed internal oscillator clock after the reset process is completed.

Figure 28-2 Reset Sequence for 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 internal oscillator clock after the reset process is finished.

Figure 28-3 Reset timing generated by overflowing of the watchdog timer, setting of the system reset request bit, detection of RAM parity errors or detection of illegal memory accesses



Note 1. If a reset occurs, P130 outputs a low level. Therefore, if the P130 is set to high level output before the reset occurs, the output of the P130 can be virtually output as the reset signal. To remove the reset signal from the external device, the P130 must be set to a high level output by software.

2. Reset processing time when removing external reset:

The first time since the POR was removed:0.672ms (TYP.), 0.832ms (MAX.)
0.399ms (TYP.), 0.519ms (MAX.) (Without LVD) after
POR 2 times :0.531ms (TYP.), 0.675ms (MAX.)
0.259ms (TYP.), 0.362ms (MAX.) (without LVD)

When the power supply voltage rises, a voltage stable waiting time 0.99ms (TYP.), 2.30ms (MAX.) is required before the reset processing time when the external reset is removed.

3. Port pin P40 becomes:

· High impedance during external reset or POR reset.
· High level (internal pull-up resistance of connection) during other reset and after accepting reset.

Note The watchdog timer is no exception and reset when an internal reset occurs.

If $VDD \geqslant V_{POR}$ or $VDD \geq VLVD$ is satisfied after reset for reset generated by voltage detection of POR circuit and LVD circuit, the reset state is released and the program is started by high speed internal oscillator clock after reset processing. Refer to "Chapter 29 Power-on Reset Circuit" and "Chapter 30 Voltage Detection Circuit".

Remark V_{POR}: Voltage rise detection voltage for POR power supply
V_{LVD}:LVD detection voltage

Table 28-1 Operational status during reset

Project			reset period
system clock			Stop providing clock to CPU.
	main system clock	fIH	Stop running.
		fX	Stop running (X1 and X2 pins in input port mode).
		fEX	Invalid clock input (pin in input port mode).
	sub-system clock	fXT	It works.
		fEXS	Invalid clock input (pin in input port mode).
	fIL		Stop running.
CPU			
code flash			Stop running.
RAM			Stop running.
Port (latch)			High impedance injection [1]
DIV			Stop running.
timer array unit			
Timer A			
Timer M			
Timer B			
timer C			
Real-time clock (RTC)			When the POR is reset, the operation stops. It can be run at other reset.
15-bit interval timer			Stop running.
watchdog timer			
Clock Out/Buzzer Out			
A/D converter			
D/A Converter Note 1			
Comparator Note 1			
universal serial communication			
Serial Interface (IICA)			
FCAN			
Data transfer controller (DMA)			
power-on reset function			It can be run for detection.
voltage detection function			When the LVD is reset, it can run. Stop the operation when other reset occurs.
external interrupt			Stop running.
key interrupt function			
CRC operation	high speed CRC		
	universal CRC		
RAM Parity			
SFR Protection Features			

Note 1. Port pin P40, P130 becomes the following state:

·P40: High impedance during external reset or POR reset. High level during other reset (internal pull-up resistance of that connection).

·P130: A low level is output during reset.

Remark fIH : high speed internal oscillator clock

fX X1 oscillating clock

fEX : external main system clock

fXT :XT1 Oscillation Clock

fEXS : external sub-system clock

fIL : low-speed internal oscillator clock

28.1 Register which identifies reset source

28.1.1 Reset control flag register (RESF)

The BAT32A237 microcontroller has a variety of 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 operation instruction.

The SYSRF, WDTRF, RPERF, IAWRF and LVIRF flags are cleared through the input of RESETB, reset of POR circuit and read of RESF register. To determine the reset source, you must save the value of the RESF register to any RAM and then determine it by its RAM value.

Figure 28-4 Format of Reset Control Flag Register (RESF)

Address: 40020440H After reset: Indefinite 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 generated by system reset request bit being set
0	No internal reset request was generated or the RESF register was cleared.
1	An internal reset request is generated.

WDTRF	Internal reset request generated by watchdog timer (WDT)
0	No internal reset request was generated or the RESF register was cleared.
1	An internal reset request is generated.

RPERF	Internal reset request due to RAM parity error
0	No internal reset request was generated or the RESF register was cleared.
1	An internal reset request is generated.

IAWRF	Accessing internal reset requests generated by illegal memory
0	No internal reset request was generated or the RESF register was cleared.
1	An internal reset request is generated.

LVIRF	Internal reset request generated by a voltage detection circuit (LVD)
0	No internal reset request was generated or the RESF register was cleared.
1	An internal reset request is generated.

Note1. Different due to reset source. Refer to Table 28-2.

Note that when RAM parity error reset (RPERDIS=0) is allowed, the "RAM AREA USED" must be initialized when accessing data; When executing instructions from the RAM area, the area of "RAM area used +10 bytes" must be initialized. By generating a reset, a state allowing generation of a RAM parity error reset (RPERDIS=0) is entered. Refer to "31.3.3 RAM Parity Error Detection Feature" for details.

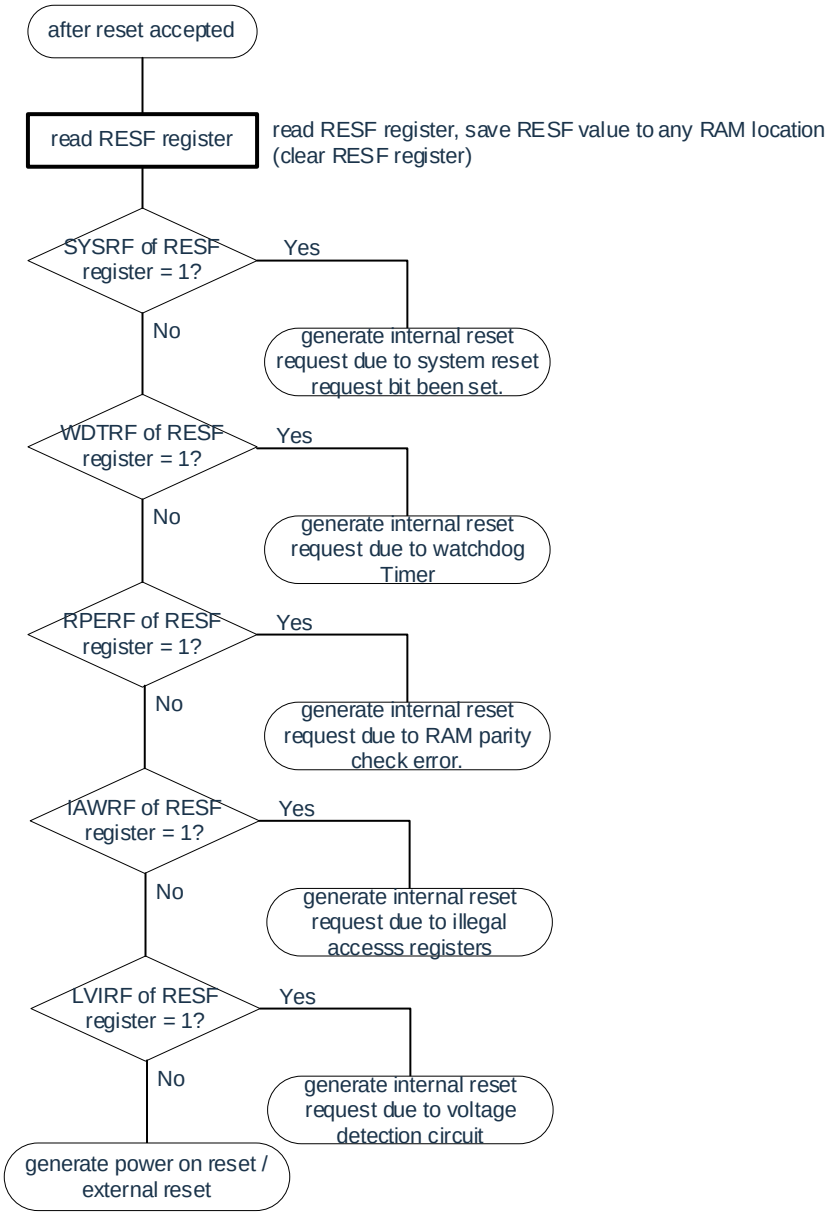
The RESF register status at the time of the reset request is shown in Table 28-2.

Table 2 8-2 RESF register state when reset request occurs

Source / Reset / Flag	RESETB Input	POR generated reset	System reset request position bit generated reset	WDT generated reset	RAM parity error generated reset	Access to reset generated by illegal memory	Reset of LVD generation
SYSRF	Clear "0"	Clear "0"	Place "1"	hold	hold	hold	hold
WDTRF			hold	Place "1"			
RPERF				hold	Place "1"		
IAWRF					hold	Place "1"	
LVIRF						hold	Place "1"

The confirmation steps for the reset source are shown in Figure 28-5.

Figure 28-5 Confirmation step for reset source



Note The above procedure is an example of a confirmation step.

Chapter 29 Power-on reset circuit

29.1 Function of power-on reset circuit

The power-on reset circuit (POR) has the following functions.

- An internal reset signal is generated when the power is turned on.

  If the supply voltage ($_{VDD}$) exceeds the detection voltage ($_{VPOR}$), the reset is released. However, the reset state must be maintained by a voltage detection circuit or an external reset before the power supply voltage reaches the operation voltage range indicated by the AC characteristic of the data manual.

- A power supply voltage ($_{VDD}$) and a detection voltage ($_{(}$VPDR) are compared. $_{When\ VDD}$<VPDR, an internal reset signal is generated. However, when the supply voltage drops, it must be reset by deep sleep mode transfer, voltage detection circuit or external reset before the supply voltage falls below the operation voltage range indicated by the AC characteristic of the data manual. When you restart the operation, you must verify that the supply voltage is back in the operation voltage range.

Note When the power-on reset circuit generates an internal reset signal, the reset control flag register (RESF) is cleared '00H'.

Note 1. BAT32A237 has multiple built-in hardware that generates internal reset signals. When the internal reset signal is generated by the access of the watchdog timer (WDT), the voltage detection (LVD) circuit, the system reset request position bit, RAM parity error or illegal memory; When an internal reset signal is generated by WDT, LVD, system reset request bit setting, RAM parity error or illegal memory access, the RESF register '00H' is not set. For more information on RESF registers, refer to Chapter 28 Reset Features..

2. VPOR: POR power supply voltage rise detection voltage

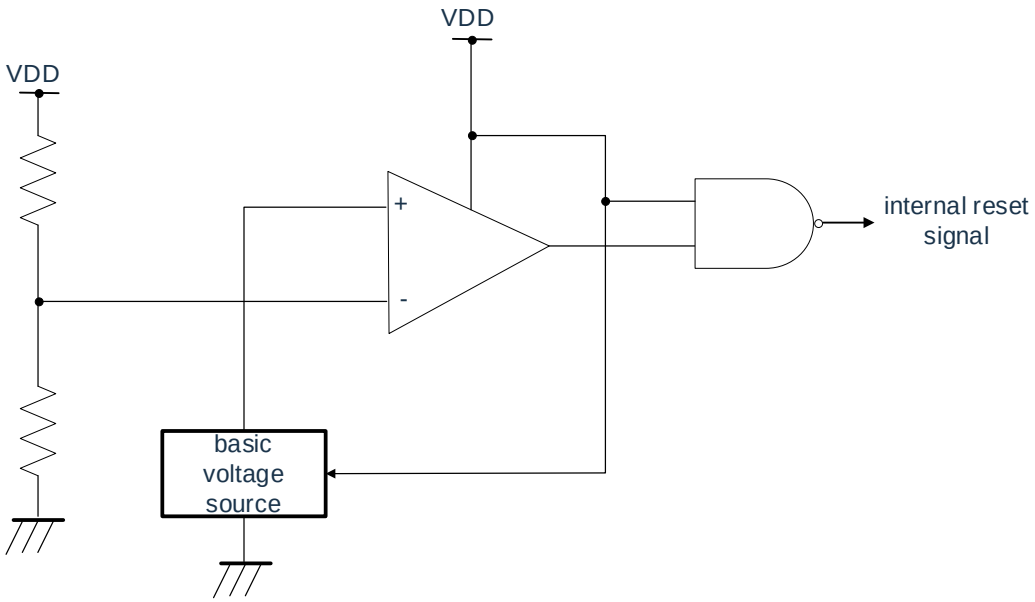
$_{VPDR}$:Voltage Drop Detection Voltage of POR Power Supply

For more information, refer to the POR circuit characteristics in the data guide.

29.2 Structure of power-on reset circuit

A block diagram of the power-on reset circuit is shown in Figure 29-1.

Figure 29-1 Block diagram of power-on reset circuit

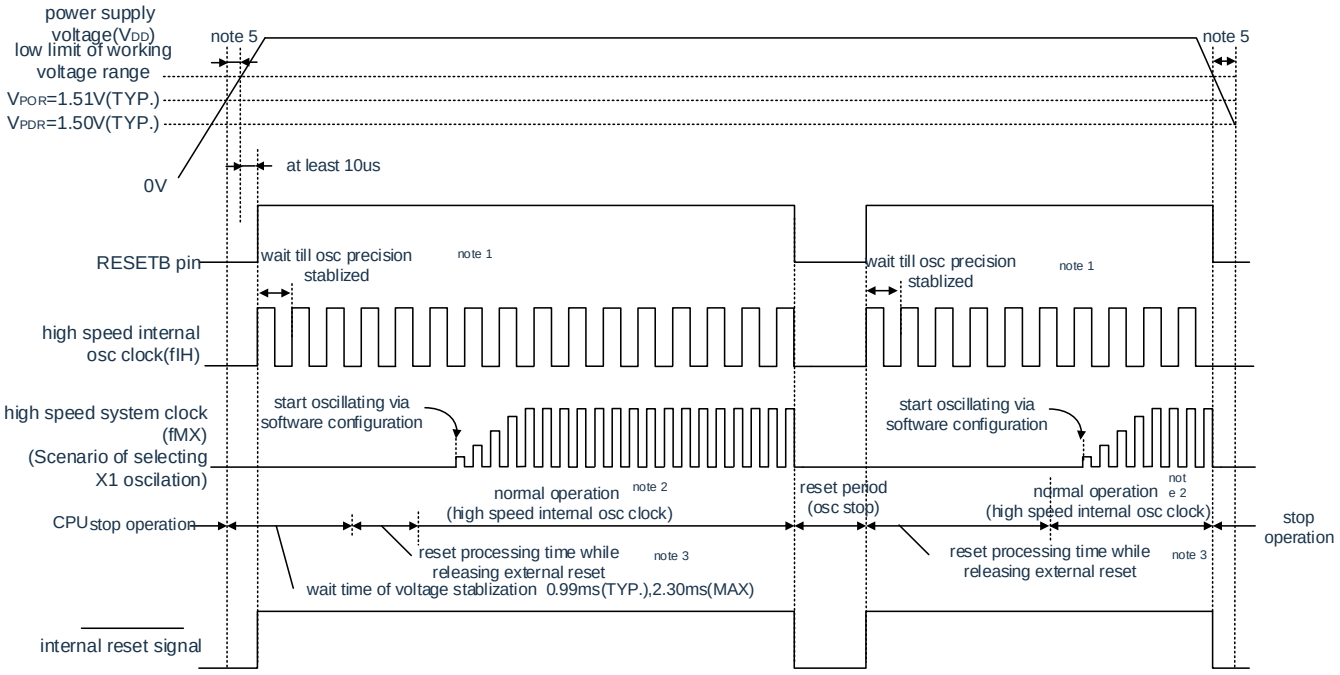


29.3 Operation of power-on reset circuit

The generation timing of the internal reset signal of the power-on reset circuit and the voltage detection circuit is as follows.

Figure 29-2 Generation sequence of internal reset signal of power-on reset circuit and voltage detection circuit (1/3)

(1) Cases of External Reset Input Using the RESETB Pin



Note1. The internal reset processing time includes the oscillation precision stable waiting time of the high speed internal oscillator clock.

2. The CPU clock can be switched from a high speed internal 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 that case of use the XT1 clock, it is necessary to switch after confirming the oscillation steady time by use of the timer function or the like.

3. The time to start the normal operation is not only VPOR (1.51 V (TYP.). The reset process time when the external reset is removed is as follows:

   The first time since the POR was removed:0.672ms (TYP.), 0.832ms (MAX.)

   0.399ms (TYP.), 0.519ms (MAX.) (without LVD)

4. The reset processing time for the external reset release after the POR release is as follows:

   After the 2nd time after POR removal :0.531ms (TYP.), 0.675ms (MAX.)

   0.259ms (TYP.), 0.362ms (MAX.) (without LVD)

5. 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.

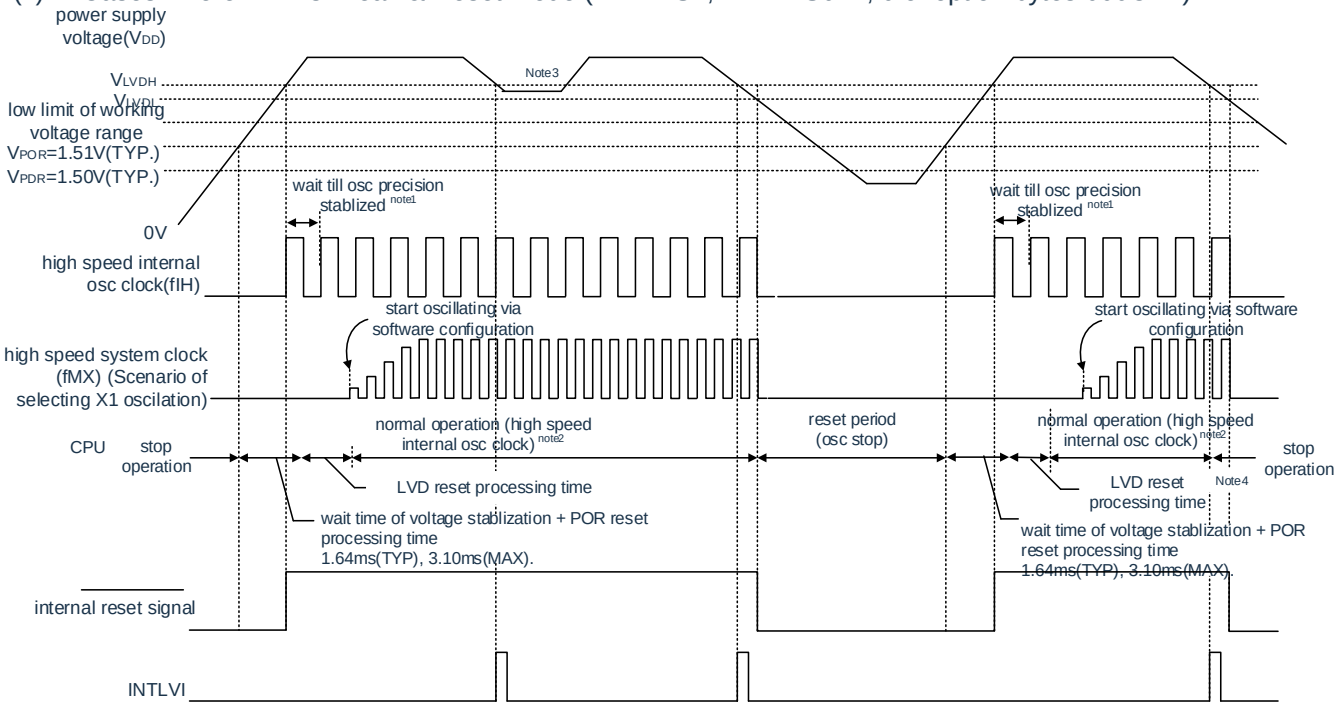
Remark V_{POR}: Voltage rise detection voltage for POR power supply

V_{PDR}:Voltage Drop Detection Voltage of POR Power Supply

Note When LVD is OFF, you must use an external reset of the RESETB pin. For details, refer to "Chapter 30 Voltage Detection Circuits."

Figure 29-2 Generation sequence of internal reset signal of power-on reset circuit and voltage detection circuit (2/3)

(2) Cases where LVD is Break & Reset mode (LVIMDS1, LVIMDS0=1, 0 of option bytes 000C1H)



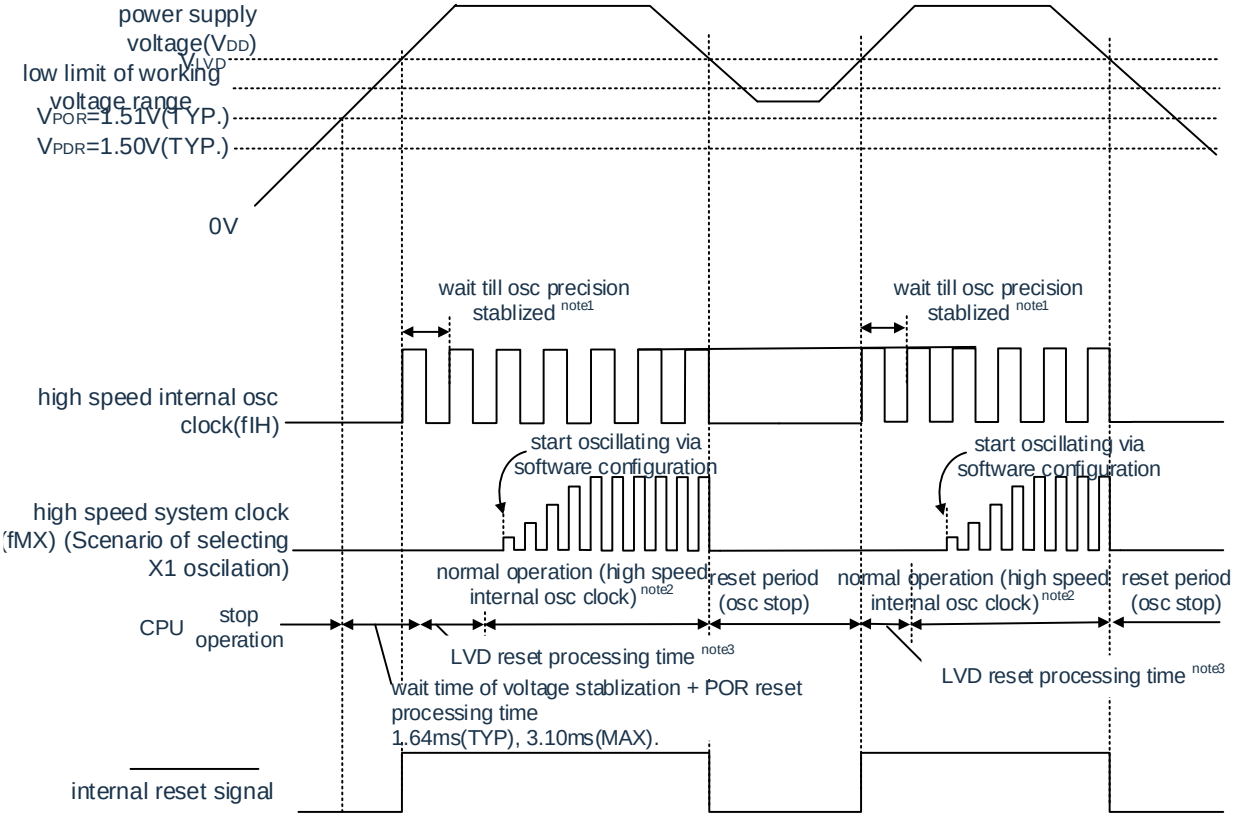
Note1. The internal reset processing time includes the oscillation precision stable waiting time of the high speed internal oscillator clock.

2. The CPU clock can be switched from a high speed internal 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 that case of use the XT1 clock, it is necessary to switch after confirming the oscillation steady time by use of the timer function or the like.
3. After generating an interrupt request signal (INTLVI), the LVILV bit and the LVIMD position of the voltage detection level register (LVIS1) are Therefore, it is necessary to consider the possibility that the power supply voltage is recovered to the high voltage detection voltage VLVDH or higher than the low voltage detection voltage.
4. The LVD reset processing time is required after reaching the LVD detection level ($_{VLVDH}$).
   LVD reset processing time: 0ms~0.0701ms (MAX.)

Remark VLVDH, VLVDL :LVD detection voltage
VPOR Voltage rise detection voltage of POR power supply
VPDR. Voltage drop detection voltage of POR power supply

Figure 29-2 Generation sequence of internal reset signal of power-on reset circuit and voltage detection circuit (3/3)

(1) The case of LVD reset mode (LVIMDS1, LVIMDS0=1, 1 for option bytes 000C1H). The LVD



Note1. The internal reset processing time includes the oscillation precision stable waiting time of the high speed internal oscillator clock.

2. The CPU clock can be switched from a high speed internal 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 that case of use the XT1 clock, it is necessary to switch after confirming the oscillation steady time by use of the timer function or the like.

3. The LVD reset processing time is required after reaching the LVD detection level ($_{VLVD}$).
   LVD reset processing time: 0ms~0.0701ms (MAX.)

4. When the power supply voltage is lowered, if the power supply voltage is restored only after the internal reset of the voltage $_{detection}$ circuit (LVD) occurs."
   LVD reset processing time: 0.0511ms (TYP.), 0.0701ms (MAX.)

Note 1.$_{VLVDH, VLVDL}$:LVD detection voltage
$_{VPOR}$: POR power supply up detection voltage
$_{VPDR.}$: POR power supply down detection voltage

2. When the LVD interrupt mode (LVIMD1, LVIMD0=0, 1 for option byte 000C1H) is selected, the time from power on to normal operation is the same as NOTE3.

Chapter 30 voltage detection circuit

30.1 Function of Voltage Detection Circuit

The voltage detection circuit sets an operation mode and a detection voltage ($_{VLVDH}$, VLVDL, VLVD) through an option byte (000C1H). The voltage detection (LVD) circuit has the following functions.

- A power supply voltage ($_{(}$VDD) and a detection voltage ($_{(}$VLVDH$_{,}$ $_{VLVDL}$, $_{VLVD}$) are compared to generate an internal reset or internal interrupt signal.
- The detection voltage of the supply voltage (VLVDH, VLVDL) can select 12 detection levels by option bytes (see Chapter 33 Option Bytes).
- Can also run in deep sleep mode.
- When the supply voltage rises, the reset state must be maintained through the voltage detection circuit or external reset before the 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. The operation 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 (VLVDH, VLVDL) are selected by the option byte 000C1H, a high voltage detection level (VLVDH) for reset release or interruption generation, and a low voltage detection level (VLVDL) for reset generation.

(b) Reset mode (LVIMDS1, LVIMDS0=1, 1 for option bytes)

The 1 detection voltage (VLVD) selected by the option byte 000C1H is used to generate or de-reset.

(c) Interrupt mode (LVIMDS1, LVIMDS0=0, 1 for option bytes)

The 1 detection voltage (VLVD) selected by option byte 000C1H is used to generate an interrupt or de-reset. In each mode, the following interrupt signal and internal reset signal are generated.

Interrupt & Reset Mode (LVIMDS1, LVIMDS0=1,0)	reset mode (LVIMDS1, LVIMDS0=1,1)	interrupt mode (LVIMDS1, LVIMDS0=0,1)
When the operation voltage drops, VDD<VLVDH is detected When VLVDH, the interrupt request signal is generated. when detecting when VDD<VLVDL, internal reset occurs; When VDD≥VLVDH is detected, the internal reset is removed.	When $_{VDD}$≥VLVD is detected, the internal reset is removed. When $_{VDD<}$VLVD is detected, an internal reset occurs.	After the reset occurs, the internal reset state of the LVD relays Continue until VDD≥VLVD. when detected When VDD≥VLVD, the internal reset of LVD is released. After the internal reset of the LVD is removed, if the When VDD<VLVD or VDD≥VLVD, the An interrupt request signal (INTLVI) is generated.

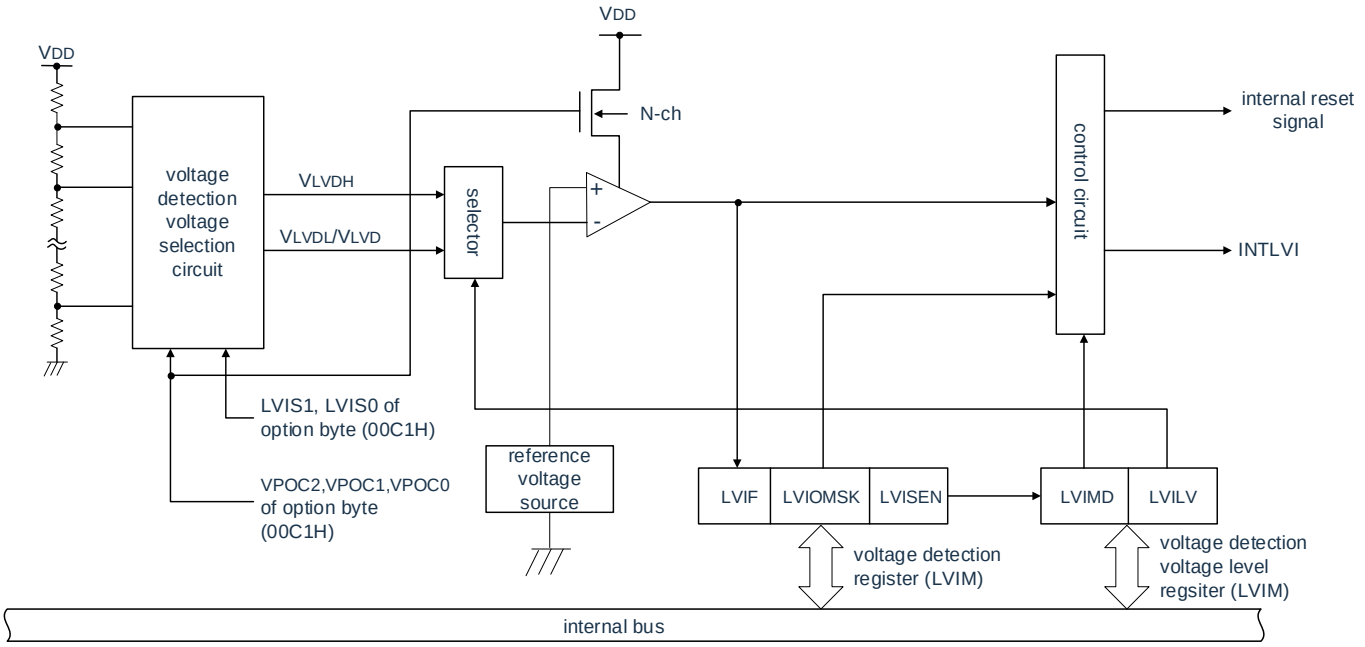
When the voltage detection circuit is running, it can read the voltage detection mark (LVIF: A voltage detection register (LVIM) bit0) confirms whether the supply voltage is greater than or equal to the detection voltage or less than the detection voltage.

If a reset occurs, set the bit0(LVIRF) of the reset control flag register (RESF) to "1. For more information on RESF registers, refer to Chapter 28 Reset Features..

30.2 Structure of voltage detection circuit

The block diagram of the voltage detection circuit is shown in Figure 30-1.

Figure 30-1 Block diagram of voltage detection circuit



30.3 Register for controlling voltage detection circuit

The voltage detection circuit is controlled by the following registers.

· Voltage detection register (LVIM)
· Voltage detection level register (LVIS)

30.1.1 Voltage detection register (LVIM)

This register setting allows or disables the rewrite of the voltage detection level register (LVIS) and confirms the masking status of the LVD output. The LVIM register is set by an 8-bit memory operation instruction.
After the reset signal is generated, the value of this register changes to "00H".

Figure 30-2 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	Settings for the Voltage Detection Level Register (LVIS) to allow/disable rewriting
0	Disable overwriting LVIS registers (LVIOMSK=0 (LVD output mask is invalid)).
1	Allow override of LVIS register (LVIOMSK=1 (LVD output mask is valid)).

LVIOMSK	Masking status flag for LVD output
0	Invalid LVD output mask.
1	The LVD output mask active 4.

LVIF	voltage detection mark
0	Power Supply Voltage (VDD) ≥ The detection voltage (VLVD) or LVD is OFF.
1	Power Supply Voltage (VDD) <Detection Voltage (VLVD)

Note1. The reset value changes due to the reset source.
When the LVD is reset, the value of the LVIM register is not reset and the original value is kept. Clear LVISEN "0" when other reset occurs.

2.bit0 and bit1 are read-only bits.

3. The initial value cannot be changed in other modes only if you select interrupt&reset mode (option bytes LVIMDS1 and LVIMDS0 bits '1' and '0' respectively).

4. The LVIOMSK bit automatically changes to "1" during the following period only if you select interrupt & reset mode (option byte LVIMDS1 and LVIMDS0 bits "1").
- Period of LVISEN=1
· Waiting time from the onset of LVD interruption to the stabilization of the LVD detection voltage
· Wait time from changing the value of the LVILV bit until the LVD detection voltage stabilizes

30.3.1 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 operation instruction. After the reset signal is generated, the value of this register changes to 00H/01H/81H note 1.

Figure 30-3 Format of Voltage Detection Level Register (LVIS)

Address: FFFAAH 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})

Note1. The reset value changes due to the settings of the reset source and option bytes. When LVD reset occurs, do not clear this register '00H'.
When a reset other than LVD occurs, the value of this register is as follows:
· LVIMDS1, LVIMDS0=1, 000:00H for option bytes
· LVIMDS1, LVIMDS0=1, 1:81H for option bytes
· LVIMDS1, LVIMDS0=0, 1:01H for option bytes
2. Write "0" only if you select interrupt & reset mode (LVIMDS1 and LVIMDS0 bits of option bytes are "1". Not in other cases. In interrupt&reset mode, value replacement is automatically performed by either creating a reset or by interrupting.

Note 1. To override the LVIS register, you must follow the steps in Figures 30-7 and 30-8.
2. The operation mode and the detection voltage of each mode of the LVD are selected by the option byte 000C1H (V_{LVDH}, V_{LVDL},V_{LVD}). User Option Bytes (000C1H/
010C1H) is shown in Table 30-1. For details on option bytes, refer to "Chapter 33 Option Bytes."

Table 30-1 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

· Settings for LVD (Interrupt & Reset Mode)

detection			Setting value for option bytes						
VLVDH		VLVDL	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode Settings	
rise	declin	declin						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 the above is prohibited.						

· Settings for LVD (reset mode)

detection		Setting value for option bytes						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode Settings	
rise	decline						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 the above is prohibited.						

Note 1. For details on LVD circuits, refer to Chapter 30 Voltage Detection Circuits."

2. The detection voltage is a TYP. value. For more information, refer to the LVD circuit characteristics in the data guide.

Table 30-1 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

· Settings for LVD (interrupt mode)

detection voltage		Setting value for option bytes						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode Settings	
rise	decline						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 the above is prohibited.						

· LVD is OFF (external reset using RESETB pin)

detection voltage		Setting value for option bytes						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode Settings	
rise	decline						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		Setting values other than the above is prohibited.						

Note 1. You must write "1" to bit4.

2. 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 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.

The operation voltage range depends on the setting of the user option byte (000C2H).

Note 1.×: Ignore

2. For details on LVD circuits, refer to Chapter 30 Voltage Detection Circuits."

3. The detection voltage is TYP. value. For more information, refer to the LVD circuit characteristics in the data guide.

30.4 Operation of voltage detection circuit

30.4.1 Settings when used as reset mode

Set the operation mode (reset mode (LVIMDS1, LVIMDS0=1, 1) and the detection voltage (VLVD) by option byte 000C1H. If you set the reset mode, the operation starts in the following initially set state.

- Set the bit7(LVISEN) of the LVIM to "0" (Disable override of the voltage detection level register (LVIS).
- The initial value of the voltage detection level register
  (LVIS) is set to '81H'. bit7 (LVIMD) is "1" (reset
  mode).
  bit0 (LVILV) is "1" (voltage detection level: VLVD).

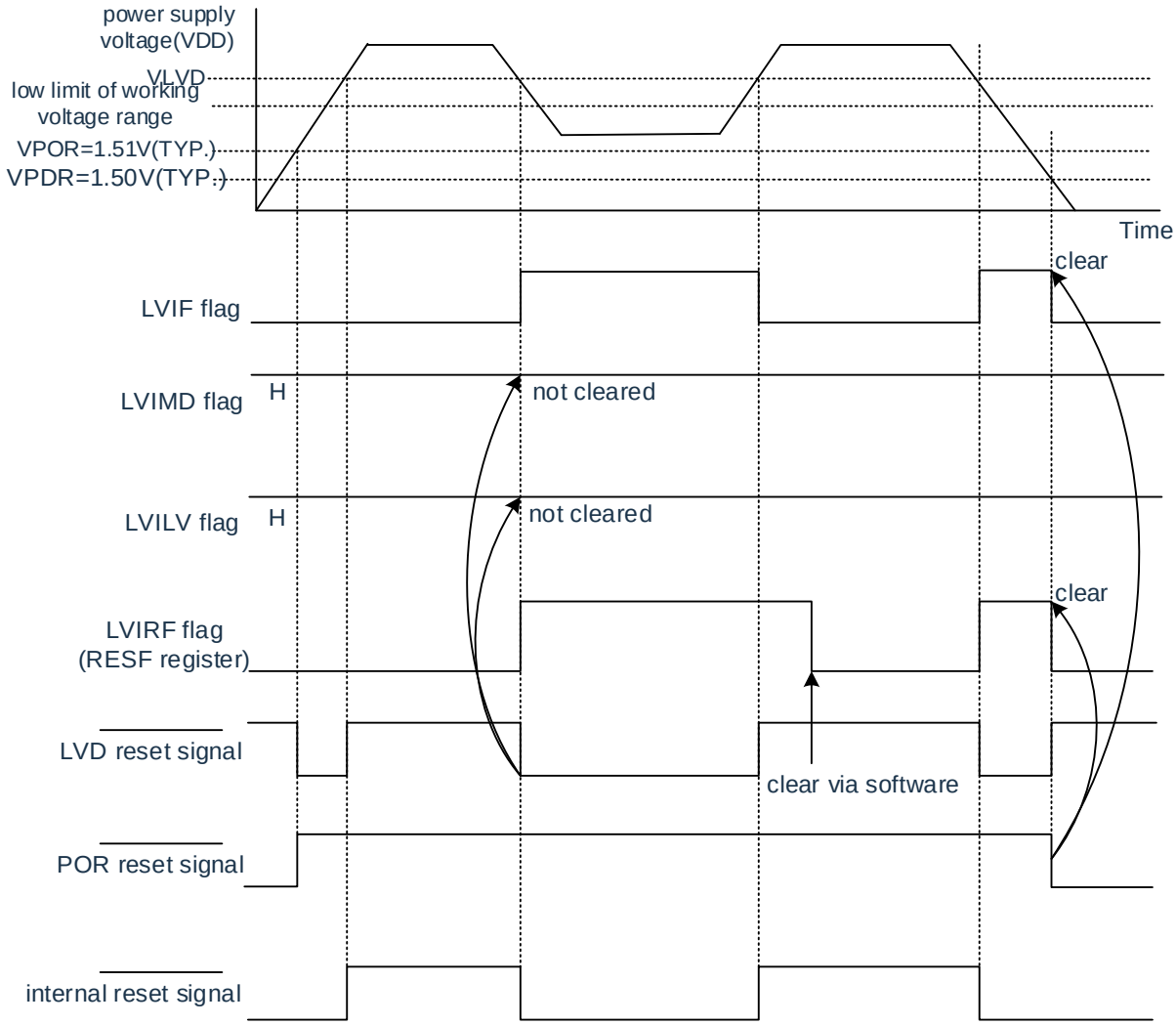
● Operation of LVD reset mode

The reset mode (LVIMDS1, LVIMDS0=1, 1 of option bytes) maintains the internal reset state of the LVD before the supply voltage (VDD) exceeds the voltage detection level (VLVD) when the power supply is switched on. If the supply voltage (VDD) exceeds the voltage detection level (VLVD), the internal reset is released.

If the supply voltage (VDD) is lower than the voltage detection level (VLVD) when the operation voltage falls.

The generation timing of the internal reset signal of the LVD reset mode is shown in FIG. 30-4.

Figure 30-4 Generation timing of internal reset signal (LVIMDS1, LVIMDS0=1, 1 of option bytes)



Remark V_{POR}: Voltage rise detection voltage for POR power supply

V_{PDR}:Voltage Drop Detection Voltage of POR Power Supply

30.4.2 Settings when used as interrupt mode

Set the operation mode (interrupt mode (LVIMDS1, LVIMDS0=0, 1) and the detection voltage (VLVD) via option byte 000C1H. If you set interrupt mode, the operation starts in the following initially set state.

- Set the bit7(LVISEN) of the LVIM to "0" (Disable override of the voltage detection level register (LVIS).
- The initial value of the voltage detection level register
  (LVIS) is set at '01H'. bit7 (LVIMD) is "0" (interrupt mode).
  bit0 (LVILV) is "1" (voltage detection level: VLVD).

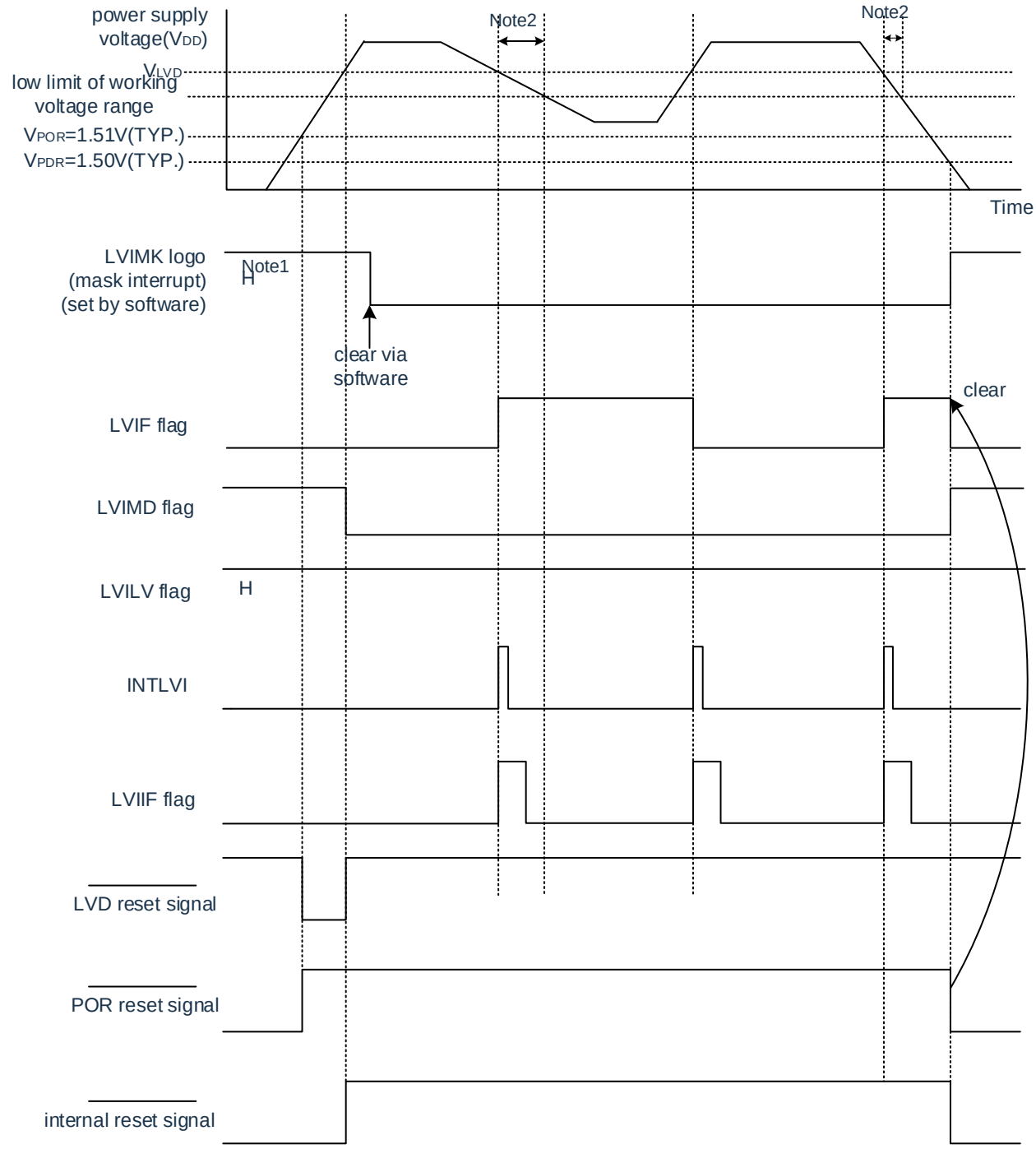
● Operation of LVD interrupt mode

After reset is generated, the interrupt mode (LVIMDS1, LVIMDS0=0,1 of option bytes) maintains the internal reset state of the LVD before the supply voltage (VDD) exceeds the voltage detection level (VLVD). If the supply voltage (VDD) exceeds the voltage detection level (VLVD), the internal reset of the LVD is released.

When the supply voltage (VDD) exceeds the voltage detection level (VLVD) after the internal reset of the LVD is removed. When the operation voltage drops, the operation voltage must be reset by deep sleep mode transfer or external reset before the operation voltage falls below the operation voltage range indicated by the AC characteristic of the data manual. When you restart the operation, you must verify that the supply voltage is back in the operation voltage range.

The generation timing of the interrupt request signal of the LVD interrupt mode is shown in FIG. 30-5.

Figure 30-5 The generation timing of the interrupt signal (LVIMDS1, LVIMDS0=0, 1 of option bytes)



Note 1. After the reset signal is generated, the LVIMK flag becomes "1".

2. When the operation voltage is decreased, the operation voltage must be reset by the transfer of deep sleep mode or external reset before the operation voltage is lower than the AC characteristic of the manual. When you restart the operation, you must verify that the supply voltage is back in the operation voltage range.

Remark V_{POR}: Voltage rise detection voltage for POR power supply

V_{PDR}:Voltage Drop Detection Voltage of POR Power Supply

30.4.3 Settings when used as interrupt & reset mode

Set the operation mode (LVIMDS1, LVIMDS0=1, 0) and the detection voltage (VLVDH, VLVDL).

If you set interrupt & reset mode, the operation starts in the following initially set state.

- Set the bit7(LVISEN) of the LVIM to "0" (Disable override of the voltage detection level register (LVIS).
- The initial value of the voltage detection level register (LVIS) is set at '00H'. bit7 (LVIMD) is "0" (interrupt mode).
  bit0 (LVILV) is "0" (High voltage detection level: VLVDH).

●LVD Break&Reset Mode Operation

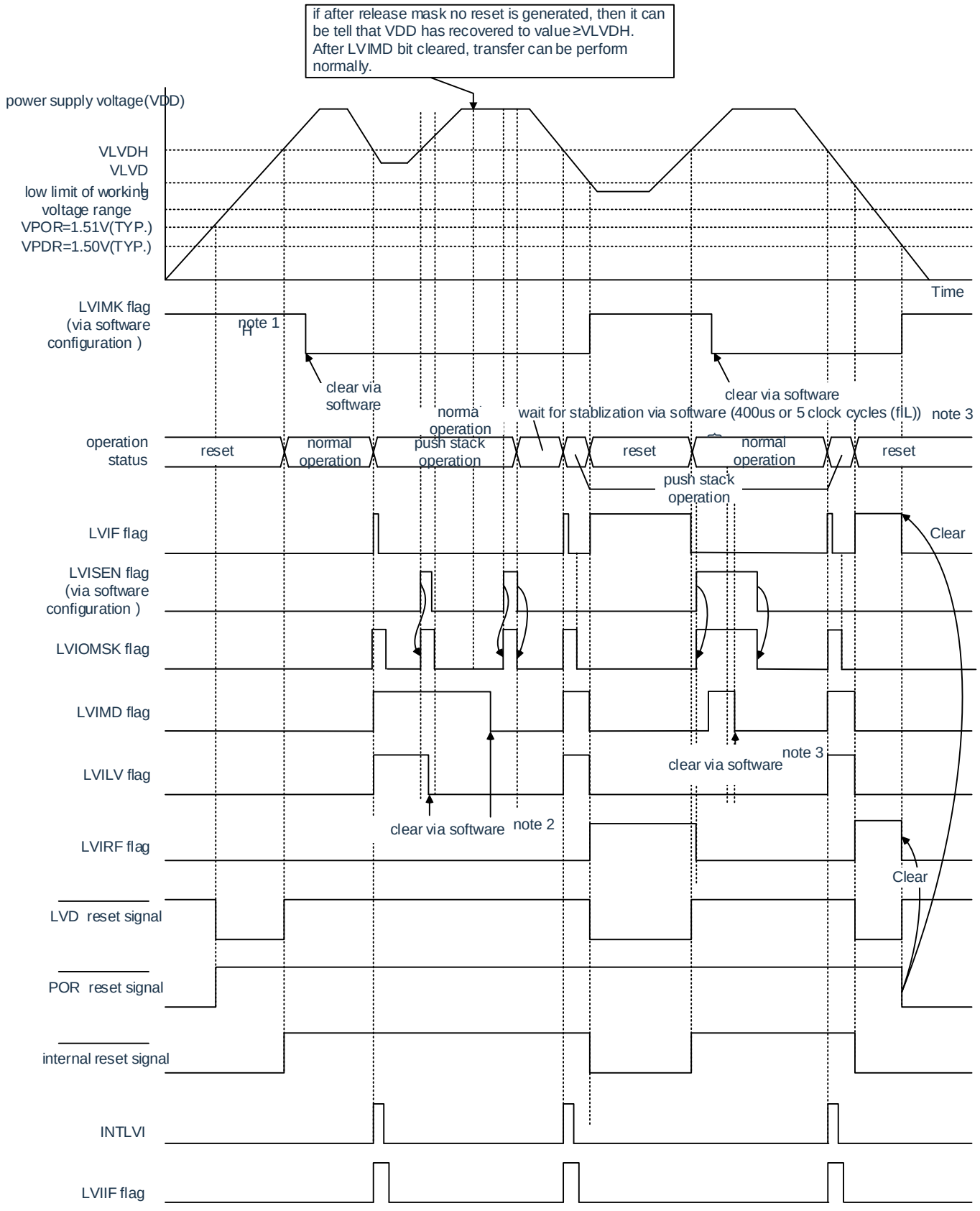
Interrupt & reset mode (LVIMDS1, LVIMDS0=1, 0 of option bytes) holds the internal reset state of LVD until the supply voltage (VDD) exceeds the high voltage detection level (VLVDH). if that supply voltage (VDD) exceed the high voltage detection level (VLVDH), the internal reset is removed.

When the operation voltage is decreased, if the supply voltage VDD is lower than the high voltage detection level VLVDH, an interrupt request signal LVD INTLVI is generated. Thereafter, if the supply voltage (VDD) is lower than the low voltage detection level (VLVDL), an internal reset of the LVD is generated. However, after INTLVI occurs, no interrupt request signal is generated even if the supply voltage (VDD) is recovered to a high voltage detection voltage (VLVDH) or higher under a low voltage detection voltage (VLVDL).

When using the LVD interrupt & reset mode, the settings must follow the steps of the flowchart shown in Confirmation/Reset Steps of Figure 30-7 and Initial Set-up Steps of Interrupt & Reset Mode.

The internal reset signal and the generation time sequence of the interrupt signal of the LVD interrupt & reset mode are shown in Figure 30-6.

Figure 30-6 Reset&Interrupt Signal Generation Timing (LVIMDS1, LVIMDS0=1,0 for option bytes)(1/2)



Note 1. After the reset signal is generated, the LVIMK flag becomes "1".

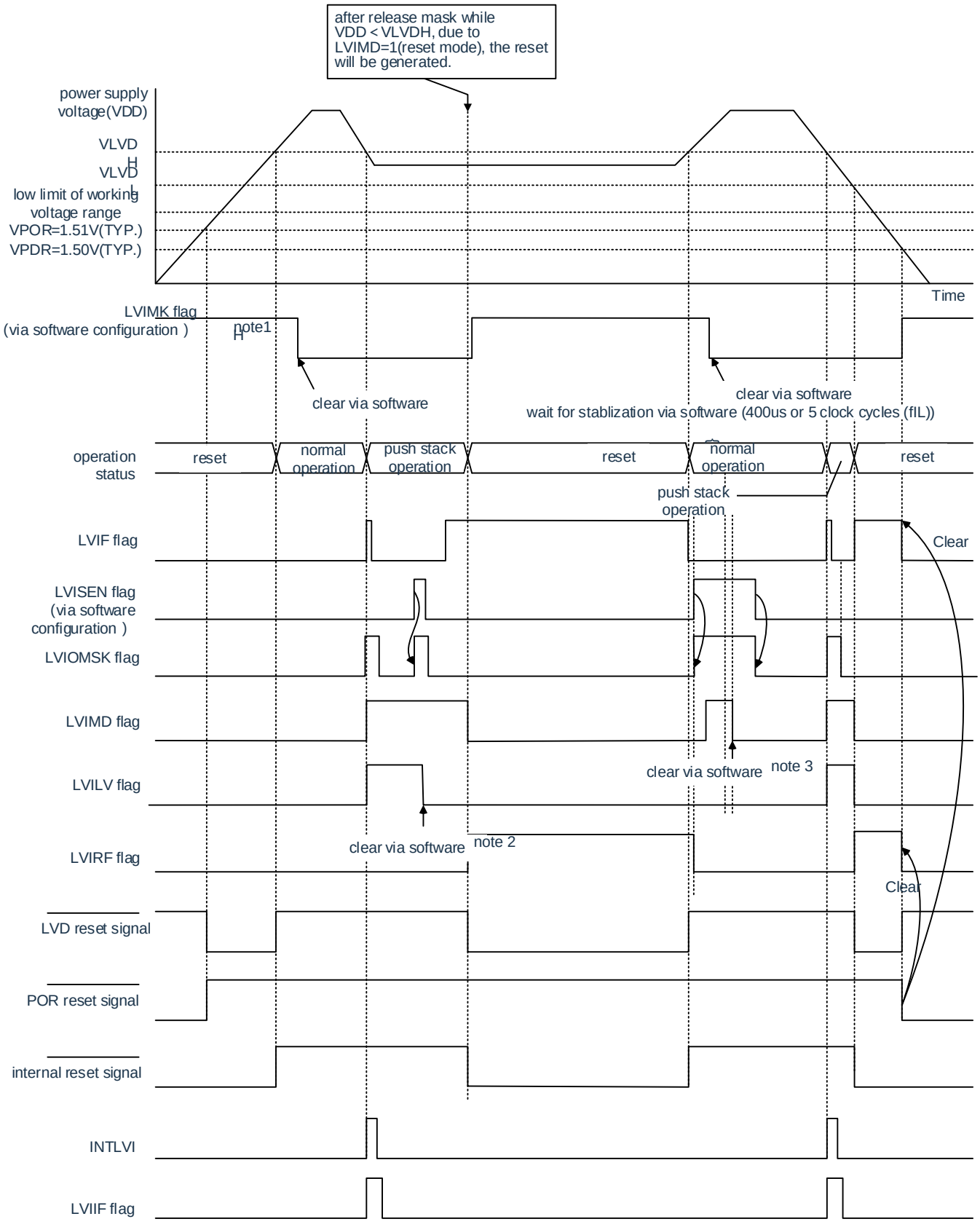
2. When using interrupt & reset mode, you must follow the Confirm/reset set-up steps for Figure 30-7 operation voltage after an interrupt.

3. When in use & reset mode, you must follow the Figure 30-8 Initial Set-up Steps for Interrupt & Reset Mode after resetting.

Remark V_{POR}: Voltage rise detection voltage for POR power supply

V_{PDR}:Voltage Drop Detection Voltage of POR Power Supply

Figure 30-6 Interrupt&reset signal generation timing (LVIMDS1, LVIMDS0=1,0 for option bytes) (2/2)



Note1. After the reset signal is generated, the LVIMK flag becomes "1".

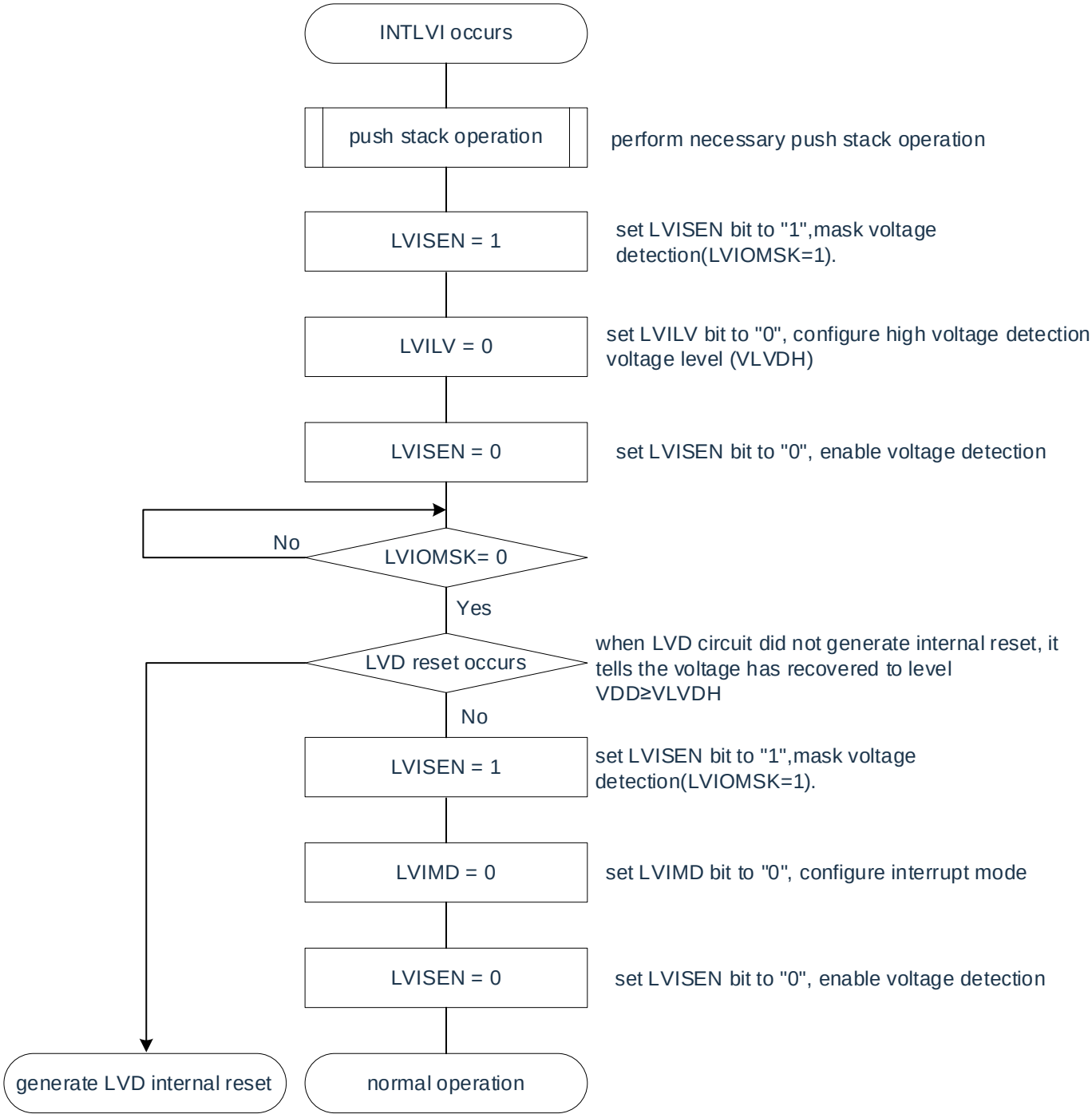
2. When using interrupt & reset mode, you must follow the Confirm/reset set-up steps for Figure 30-7 operation voltage after interrupt.

3. When using interrupt & reset mode, you must follow the "Figure 30-8 Initial Set-up Steps for Interrupt & Reset Mode" after resetting.

Remark $_{VPOR}$: Voltage rise detection voltage for POR power supply

$_{VPDR}$:Voltage Drop Detection Voltage of POR Power Supply

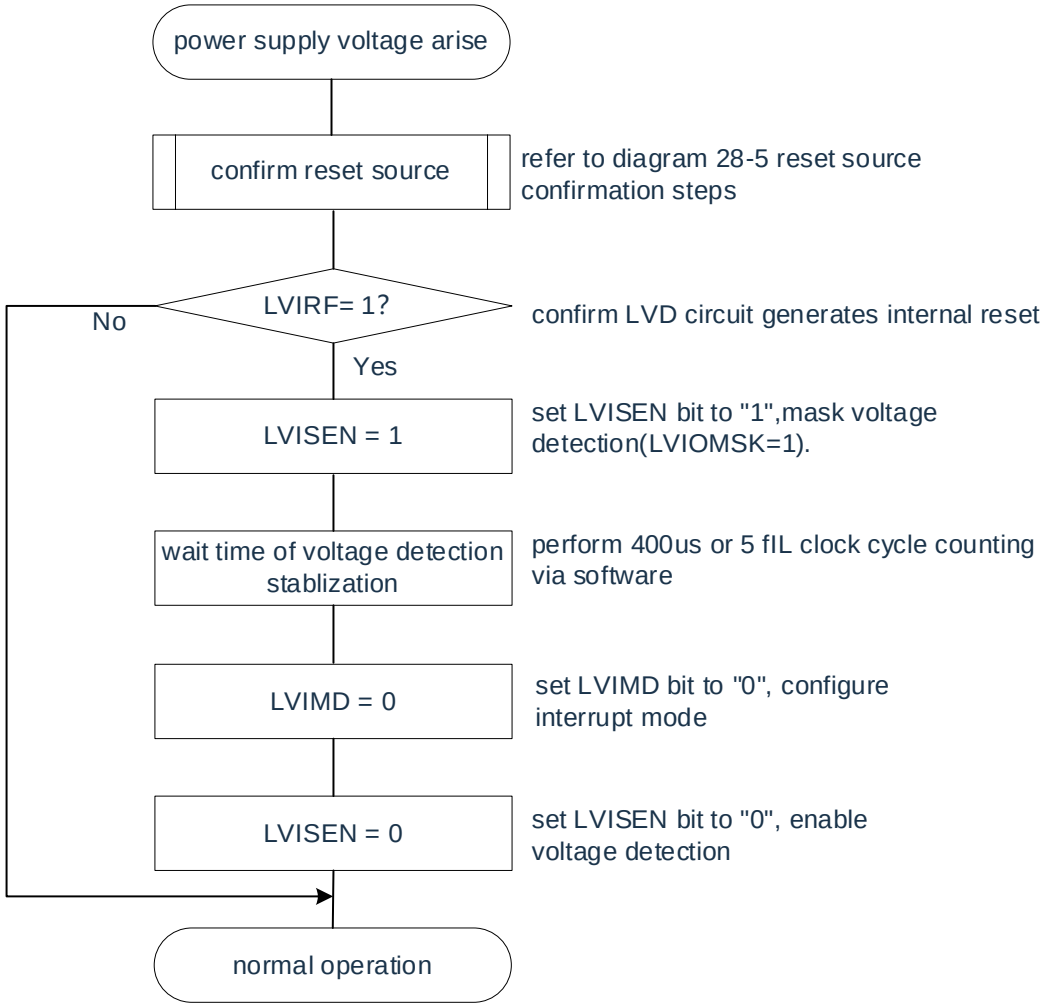
Figure 30-7 Set-up steps for confirming/resetting the working voltage



If interrupt & reset mode (LVIMDS1, LVIMDS0=1,0) is set, a voltage detection stability wait time of 400us or 5 fIL clocks is required after the LVD reset (LVIRF=1) is removed. The LVIMD bit must be initialized with '0' after waiting for voltage detection to stabilize. The LVISEN position '1' must be masked from the generation of reset or interruption during the counting of the voltage detection stable latency and when LVIMD bits are overridden.

The initial set-up steps for the interrupt & reset mode are shown in Figure 30-8.

Figure 30-8 Initial set-up steps for interrupt & reset mode



Remark: $_{fIL}$: low-speed internal oscillator clock frequency

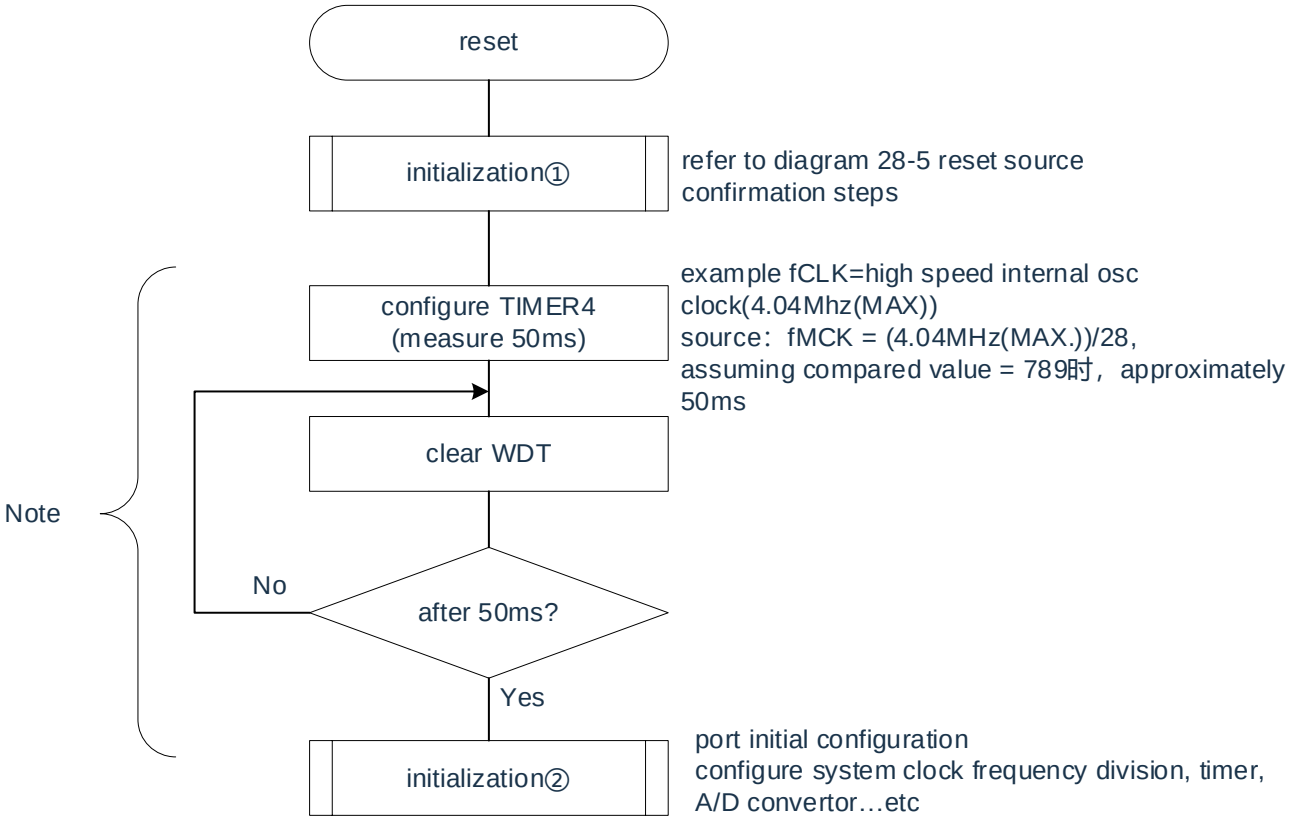
30.2 Precautions for Voltage Detection Circuits

(1) Voltage fluctuations on power

For a system in which a supply voltage (VDD) fluctuates for a certain time near an LVD detection voltage, it is possible to repeatedly enter a reset state and a reset-released state. It can set the time from the reset to the start of the single-chip microcomputer by the following processing.

<Process> After the reset is removed, the initial setting of the port or the like must be performed after waiting for different power supply voltage fluctuation times for each system by using the timer software counter.

Figure 30-9 Example of software processing when the fluctuation of the supply voltage near the LVD detection voltage is not more than 50 ms



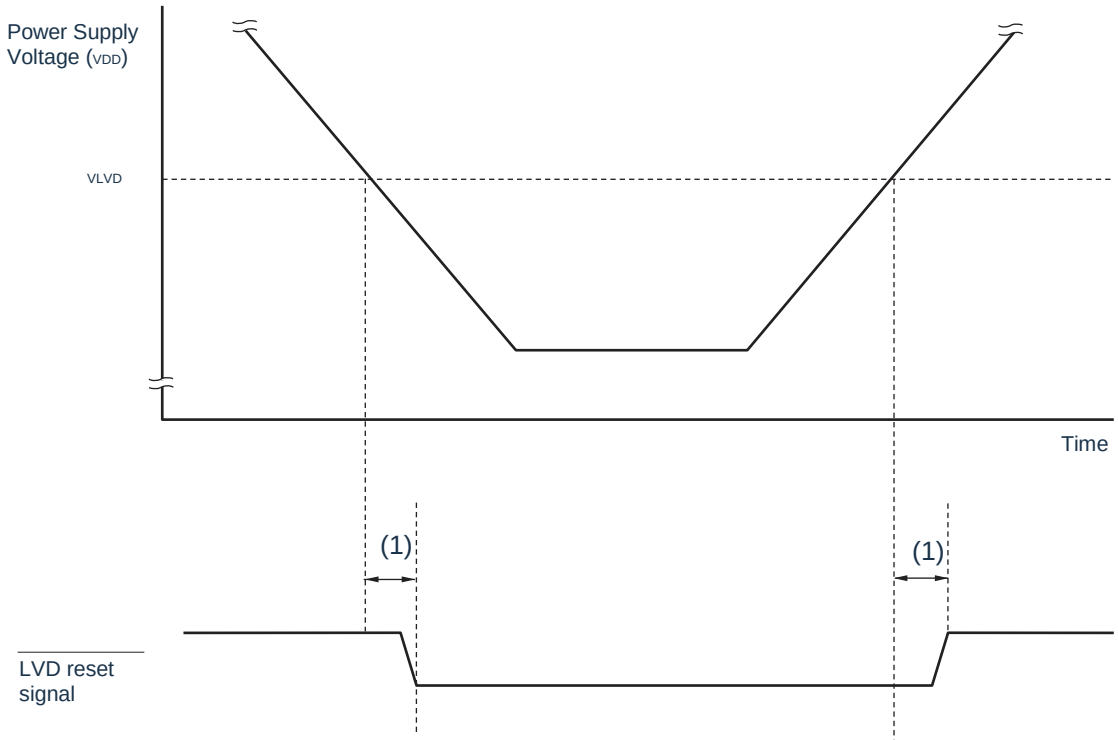
Note If a reset occurs again during this period, the

initialization process (2) is not transferred.

(2) Delay from generation of LVD reset source to generation or release of LVD reset

Delay may occur from that time the supply voltage (VDD)<LVD detection voltage (VLVD) is met to the time of LVD reset. Similarly, a delay may occur from the LVD detection voltage (VLVD)≤supply voltage (VDD) until the LVD reset is removed (see FIG. 30-10).

Figure 30-10 Delay from generation of LVD reset source to generation or release of LVD reset



(1): Detection delay (300us (MAX.))

(3) About Power On When Setting LVD to OFF

When you set the LVD to OFF, you must use an external reset of the RESETB pin.

At least 10s low level must be entered for the RESETB pin during external reset. If external reset is performed while the supply voltage is rising, the RESETB pin must be powered on after low level input and maintained at least 10us low level input.

(4) Information about a decrease in operation voltage when the LVD is set to OFF and is set to LVD interrupt mode

If the operation voltage drops, the LVD must be reset by transfer of deep sleep mode or external reset before the operation voltage falls below the operation voltage range indicated by the AC characteristic of the data manual. When you restart the operation, you must verify that the supply voltage is back in the operation voltage range.

Chapter 31 Safety function

31.1 Summary of Safety Features

In response to the IEC60730 and EC61508 Safety standards, the following Safety features are built into the BAT32A237.

The purpose of this Safety function is to stop work safely when a fault is detected through the self-diagnosis of the single chip computer.

(1) Flash CRC Operation Function (High Speed CRC, Universal CRC)

A data error of the flash memory is detected by the CRC operation. The following two CRCs can be used respectively according to different uses and use conditions.

· "High Speed CRC"... In the initialization program, the CPU can be stopped and the entire code flash memory area can be checked at high speed.

· "Universal CRC"... In CPU operation, it is not limit to code flash area but can be used for multi-purpose inspection.

(2) RAM parity error detection function

Detects parity errors when reading RAM data.

(3) SFR Protection Features

Prevents SFR from being overridden due to CPU runaway.

(4) frequency detection function

A universal timer unit can be used for self-detection of CPU/peripheral hardware clock frequency.

(5) A/D Test Features

A/D converter self-check can be carried out through A/D conversion of positive (+) reference voltage, negative (negative-reference voltage, analog input channel (ANI), temperature sensor output and internal reference voltage output.

(6) Digital output signal level detection function of input/output port

When the input/output port is in the output mode, the output level of the pin can be read.

(7) Product Unique Identifier Register (128-bit)

31.2 Register used by Safety function

The following registers are used for each function of the Safety function.

register name	Features of Safety Features
· Flash CRC control register (CRC0CTL) · Flash CRC Operation Result Register (PGCRCL)	Flash CRC Operation (High Speed CRC)
• CRC input register (CRCIN) • CRC Data Register (CRCD)	CRC operation function (Universal CRC)
· RAM parity error control register (RPECTL)	RAM Parity Error Detection
· Special SFR Protection Control Register (SFRGD)	SFR Protection Features
· Timer input selection register 0 (TIS0)	frequency detection function
·A/D test register (ADTES)	A/D Test Features
· Port mode selection register (PMS)	Digital output signal level detection function of input/output pin

For the contents of each register, describe in "Operation of 31.3 Safety Features".

31.3 Operation of Safety functions

31.3.1 Flash CRC Operation Function (High Speed CRC)

The IEC60730 standard requires confirmation of the data in flash memory, and recommends CRC as a means of confirmation. This high speed CRC can check the entire code flash memory area in the initial set-up (initialization) program. High speed CRC can only be performed in sleep mode of the main system clock through a program in the RAM.

The high speed CRC stops the CPU running and reads 32 bits of data from the flash memory through 1 clock to operate. It is therefore characterized by a shorter time to complete the check (e.g. 64KB flash: 512us@32MHz).

CRC-generated polynomial corresponds to "X16+X12+X5+1" of CRC-16_CCITT.

The operation is performed with MSB precedence of bit31bit0.

Remark: Since that universal CRC is LSB priority, the result of the operation are different.

31.3.1.1 Flash 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 operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 31-1 Format of a flash CRC control register (CRC0CTL)

Address: 40021810H After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	FEA6	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	Operation Control of High Speed CRC Arithmetic Unit
0	Stop running.
1	Begin the operation by executing the HALT instruction.

FEA6	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	high speed CRC calculation range
0	0	0	0	0	0	0	00000H ~ 1FFBH(8K-4byte)
0	0	0	0	0	0	1	00000H ~ 3FFBH(16K-4byte)
0	0	0	0	0	1	0	00000H ~ 5FFBH(24K-4byte)
0	0	0	0	0	1	1	00000H ~ 7FFBH(32K-4byte)
0	0	0	0	1	0	0	00000H ~ 9FFBH(40K-4byte)
0	0	0	0	1	0	1	00000H ~ BFFBH(48K-4byte)
0	0	0	0	1	1	0	00000H ~ DFFBH(56K-4byte)
0	0	0	0	1	1	1	00000H ~ FFFBH(64K-4byte)
0	0	0	1	0	0	0	00000H ~ 11FFBH(72K-4byte)
0	0	0	1	0	0	1	00000H ~ 13FFBH(80K-4byte)
0	0	0	1	0	1	0	00000H ~ 15FFBH(88K-4byte)
0	0	0	1	0	1	1	00000H ~ 17FFBH(96K-4byte)
0	0	0	1	1	0	0	00000H ~ 19FFBH(104K-4byte)
0	0	0	1	1	0	1	00000H ~ 1BFFBH(112K-4byte)
0	0	0	1	1	1	0	00000H ~ 1DFFBH(120K-4byte)
0	0	0	1	1	1	1	00000H ~ 1FFFBH(128K-4byte)
1	x	x	x	x	x	x	00000H ~ 1EFFBH(124K-4byte)

Remark The expected value of that CRC operation result for comparison must be stored in the last 4 bytes of the flash memory in advance, so the operation range is minus 4 bytes.

31.3.1.2 flash CRC operation result register (PGCRCL)

This is a register that holds the results of high speed CRC operations.
The PGCRCL register is set by the 16-bit memory
operation instruction.
After the reset signal is generated, the value of
this register changes to "0000H".

Figure 31-2 Format of a flash 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

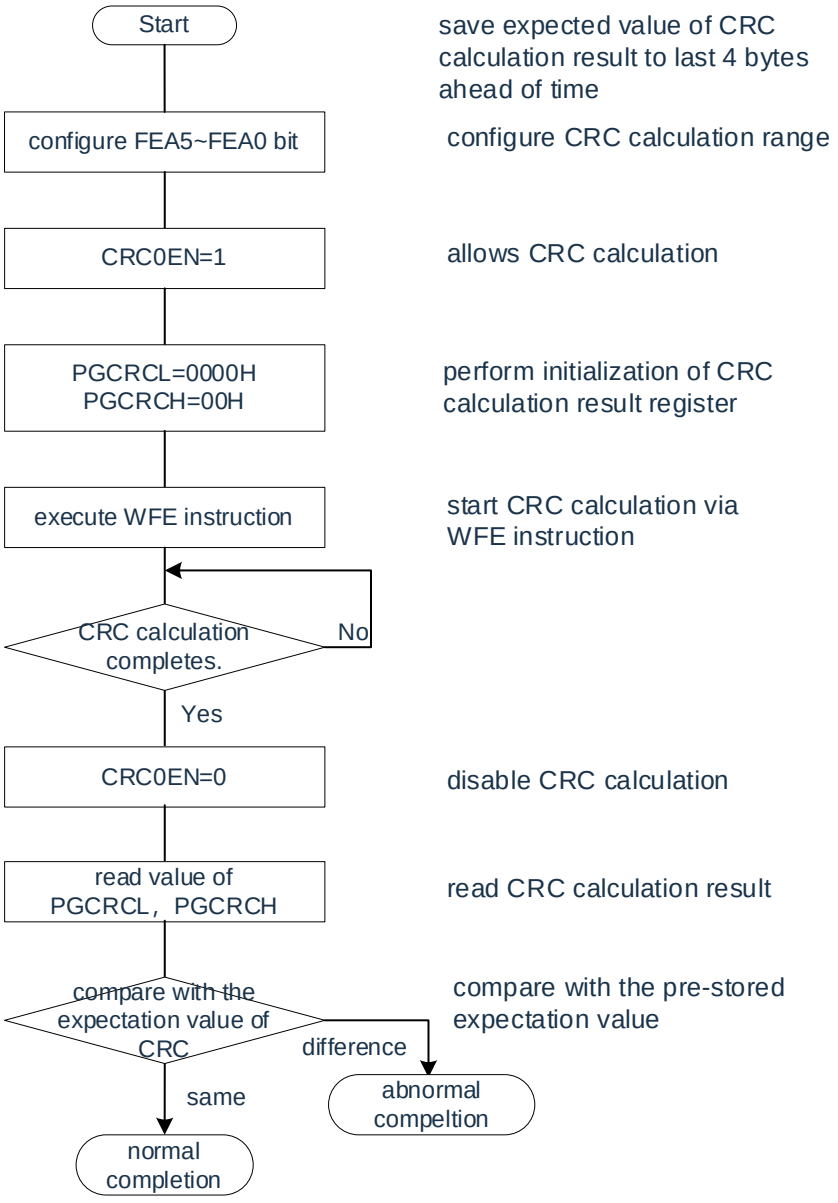
PGCRC15～0	Result of High Speed CRC
0000H～FFFFH	Save the result of high speed CRC.

Note The PGCRCL register can only be written if the CRC0EN (bit7 of the CRC0 CTL register) bit is "1".

The flow chart of the flash CRC operation function (high speed CRC) is shown in Figure 31-3.

<Operational Processes>

Figure 31-3 Flowchart for flash CRC operation function (high speed CRC)



Note 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.

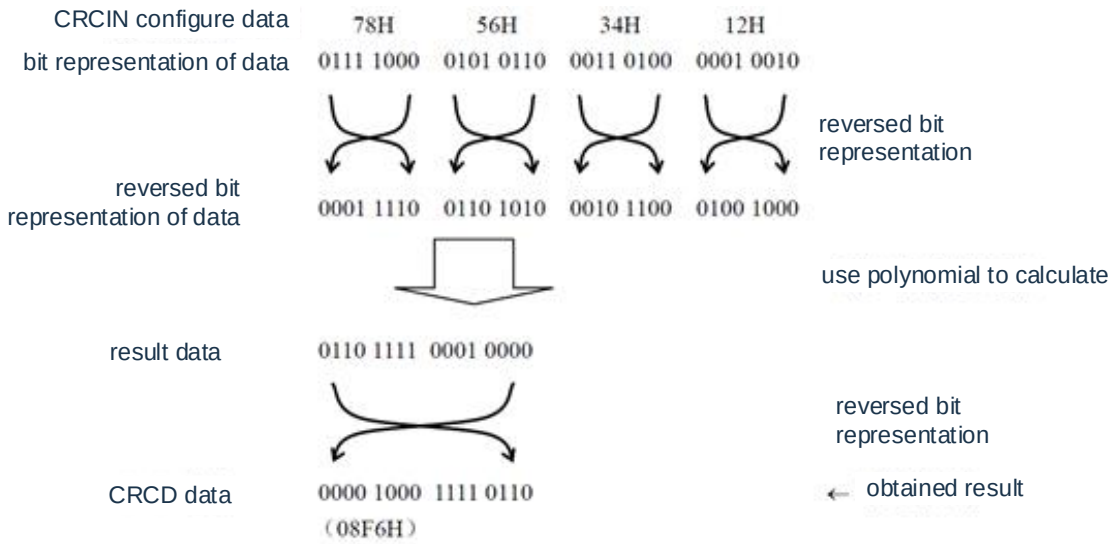
31.3.2 CRC Operation Function (Universal CRC)

In order to ensure Safety during operation, the IEC61508 standard requires data to be confirmed even in CPU operation.

The universal CRC can perform CRC operation as a peripheral function in CPU operation. The universal CRC is not limited to code flash areas and can be used for multi-purpose inspection. Specify the data to be confirmed by software (user program). The CRC operation in sleep mode can only be used during DMA transfer.

The CRC operation function can be used in the main system clock operation mode or the sub-system clock operation mode.

The CRC generation polynomial uses the "X16+X12+X5+1" of CRC-16-CCITT. Since communication is preferentially performed with LSB, calculation is performed after the bit order of the input data is reversed. For example, when sending data "12345678H" from LSB, the CRCIN register is written to the CRCD register in the order of "78H," "56H,"34H, and "12H". This is the result of a CRC operation for the following bit sequence after reversing the bit sequence of the data "12345678H".



Note When executing the program, because the program rewrites the setting line of the software breakpoint to the breakpoint command, the result of CRC is different.

31.3.2.1 CRC input register (CRCIN)

This is an 8-bit register that sets the CRC calculation data for a generic CRC.
The range can be set to "00H~FFH".
The CRCIN register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 31-4 Format of CRC Input Register (CRCIN)

Location: 400433 ACH After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
CRCIN								

bit7~0	Features
00H~FFH	data input

31.3.2.2 CRC Data Register (CRCD)

This is a register that holds the results of a generic CRC operation. The range that can be set is '0000H~FFFFH'.
After writing the CRCIN register, the CRC result is saved to the CRCD register after 1 CPU/peripheral hardware clock (fCLK). The CRCD register is set by the 16-bit memory operation instruction.
After the reset signal is generated, the value of this register changes to "0000H".

Figure 31-5 Format of CRC Data Register (CRCD)

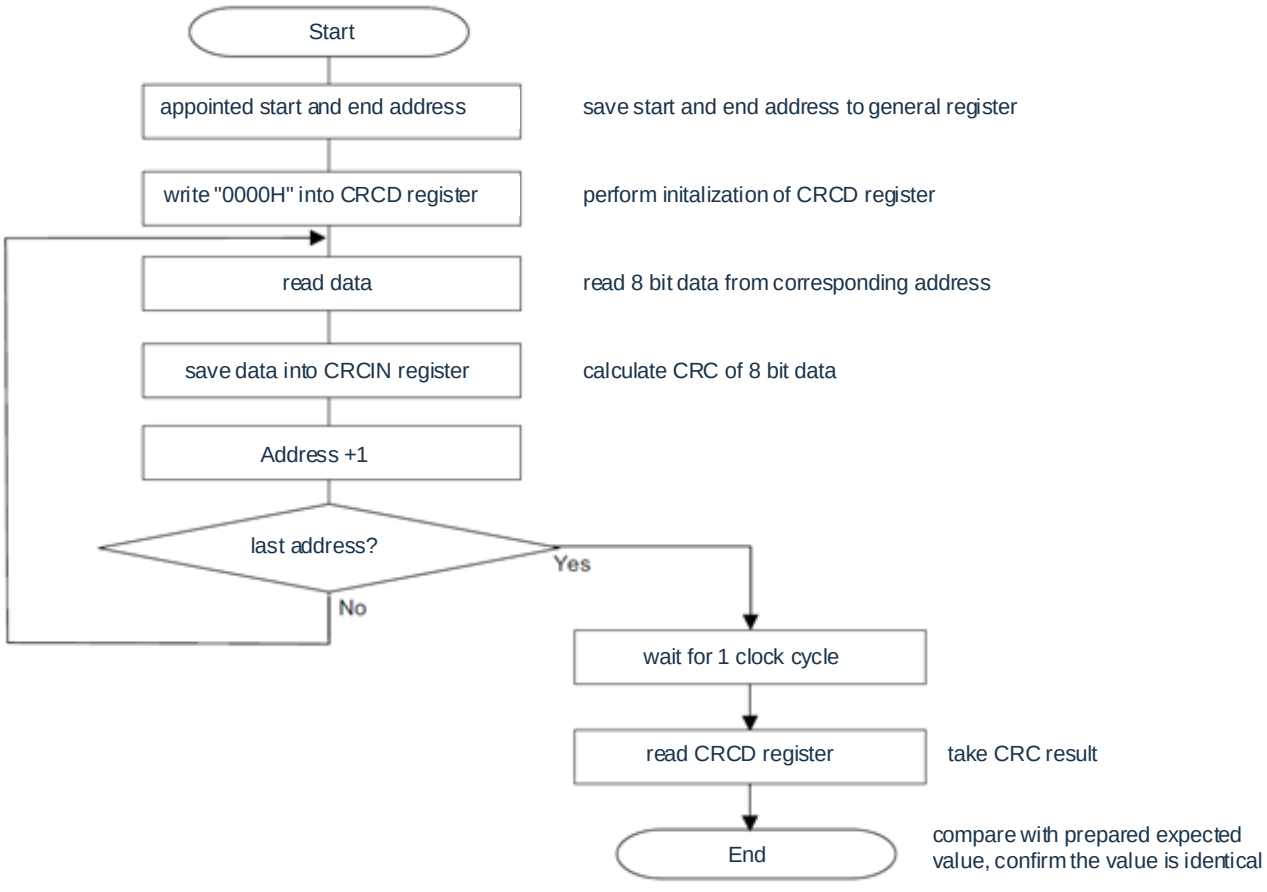
Location: 400432FAH After reset: 0000H R/W

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

Note 1. To read the write value of the CRCD register, the CRCD register must be read before writing the CRCIN register.
2. If the write operation of the CRCD register competes with the saving of the operation result, the write operation is ignored.

<Operational Processes>

Figure 31-6 Flowchart of CRC Operation Function (Universal CRC)



31.3.3 RAM Parity Error Detection

The IEC60730 standard requires that RAM data be acknowledged. Therefore, BAT32A237's RAM is appended with 1-bit parity bits per 8 bits. The RAM parity error detection function appends parity bits when writing data and checks parity bits when reading data, and can generate a reset when a parity error occurs.

31.3.3.1 RAM parity error control register (RPECTL)

This register controls the error-confirmed bit of the parity and the reset due to the parity error. The RPECTL register is set by an 8-bit memory operation instruction. After the reset signal is generated, the value of this register changes to "00H".

Figure 31-7 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	Mask for Parity Error Reset
0	Allows for parity error reset to occur.
1	Prevents parity error reset from occurring.

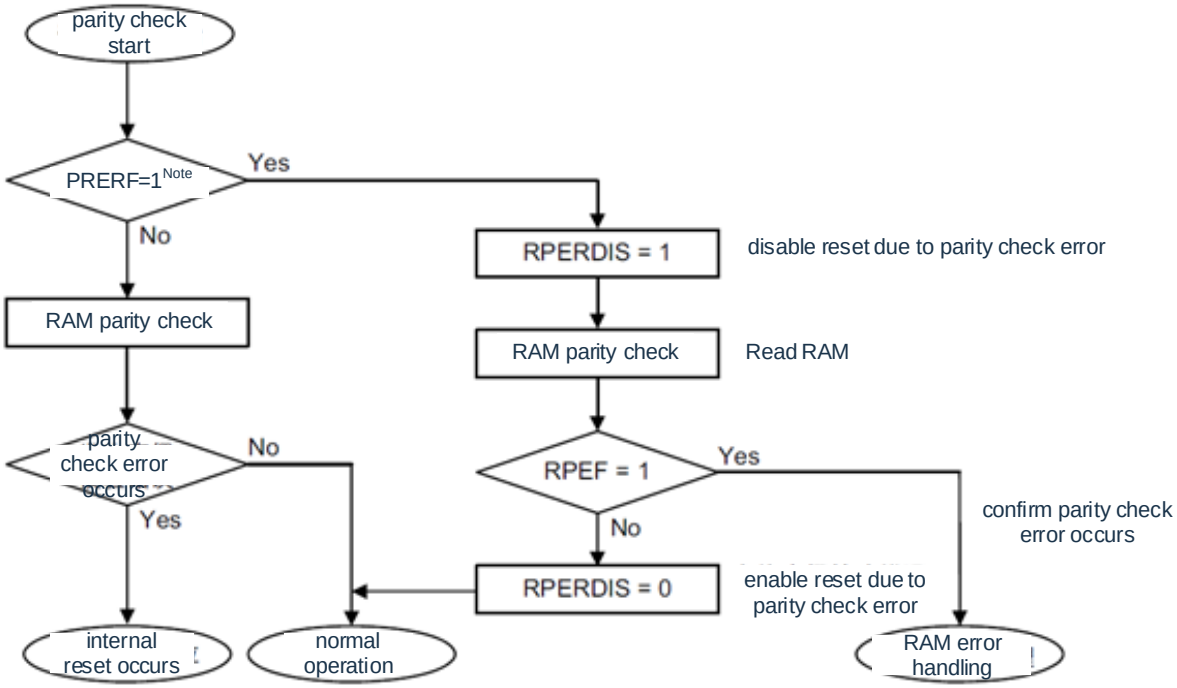
RPEF	Parity error status flag
0	No parity errors occurred.
1	A parity error occurred.

Note The parity bit is appended when the data is written and checked when the data is read.
Therefore, to allow RAM parity error reset (RPERDIS=0) to occur, you must "use RAM area" when accessing data
Initialize.
Since it is a pipelined operation, that CPU perform pre-reading, and a RAM parity error may occur due to uninitialized RAM area before reading the RAM area used. Therefore, to allow RAM parity error reset (RPERDIS=0), the area of "RAM Area + 10 bytes used" must be initialized.

Note 1. The initial state is allow for parity error reset (RPERDIS=0).

2. Even if set to disable parity error reset (RPERDIS=1), the RPEF flag is set to "1" when a parity error occurs. If the RPEF bit is set to allow parity error reset (RPERDIS=0), parity error reset occurs when RPERDIS is cleared "0".
3. Set the RPEF flag of the RPECTL register to "1" due to RAM parity errors, and clear the RPEF flag to "0". When the RPEF flag is "1", the RPEF flag remains "1" even when reading RAM without parity errors.
4. The scope of RAM parity detection does not include universal registers.

Figure 31-8 RAM Parity Flow



Note For confirmation of the internal reset of RAM parity errors, refer to Chapter 28 Reset Features, section.

31.3.4 SFR Protection Features

In order to ensure Safety during operation, the IEC61508 standard requires that even if the CPU is out of control, critical SFRs must be protected from rewriting. 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.

When set to SFR protection, the protected SFR writes are invalid but can be read normally.

31.3.4.1 SFR Protection Control Register (SFRGD)

This register controls whether the SFR protection feature is valid.
The SFR protection feature uses GCOMP bits, GPORT bits, GINT bits, and GCSC bits.
The SFRGD register is set by an 8-bit memory operation instruction.
After the reset signal is generated, the value of this register changes to "00H".

Figure 31-9 Format of SFR Protection Control Register (SFRGD)

Location: 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	Protection of control registers for comparator functions
0	Invalid. Control register capable of reading and writing comparator function.
1	Valid. The port-functional control register has invalid write operation and can be read. [Protected SFR]COMPPMDR, COMPFIR, COMPOCR, CVRCTL, CxRVM, PGAxCTL, PGASHMD, CMPSELx

GPORT	Port-functional control register protection
0	Invalid. Control register capable of reading and writing port function.
1	Valid. The port-functional control register has invalid write operation and can be read. [Protected SFR]PMxx, PUxx, PIMxx, POMxx, PMCxx, ADPC,PIORx Note

GINT	Protection of register for interrupt function
0	Invalid. Control register capable of reading and writing interrupt function.
1	Valid. The write operation of the control register of the interrupt function is invalid and can be read. [Protected SFR]IFxx, MKxx, PRxx, EGPx,EGNx

GCSC	Protection of control registers for clock control function, voltage detection circuit and RAM parity error detection function
0	Invalid. A control register capable of reading and writing clock control function, voltage detection circuit and RAM parity error detection function.
1	Valid. The write operation of the control register of the clock control function, the voltage detection circuit and the RAM parity error detection function is invalid and can be read. [Protected SFR]CMC, CSC, OSTS, CKC, PERx, OSMC, LVIM, LVIS,RPECTL

Note Pxx (port register) is not protected.

31.3.5 frequency detection function

The IEC60730 standard requires that the oscillation frequency be confirmed as normal.

The frequency detection function can use the CPU/peripheral hardware clock frequency (fCLK) and judge whether the ratio relation of two clocks is correct.

However, if one clock or two clocks stop oscillating, the ratio relation of two clocks cannot be judged.

<Clock to compare>

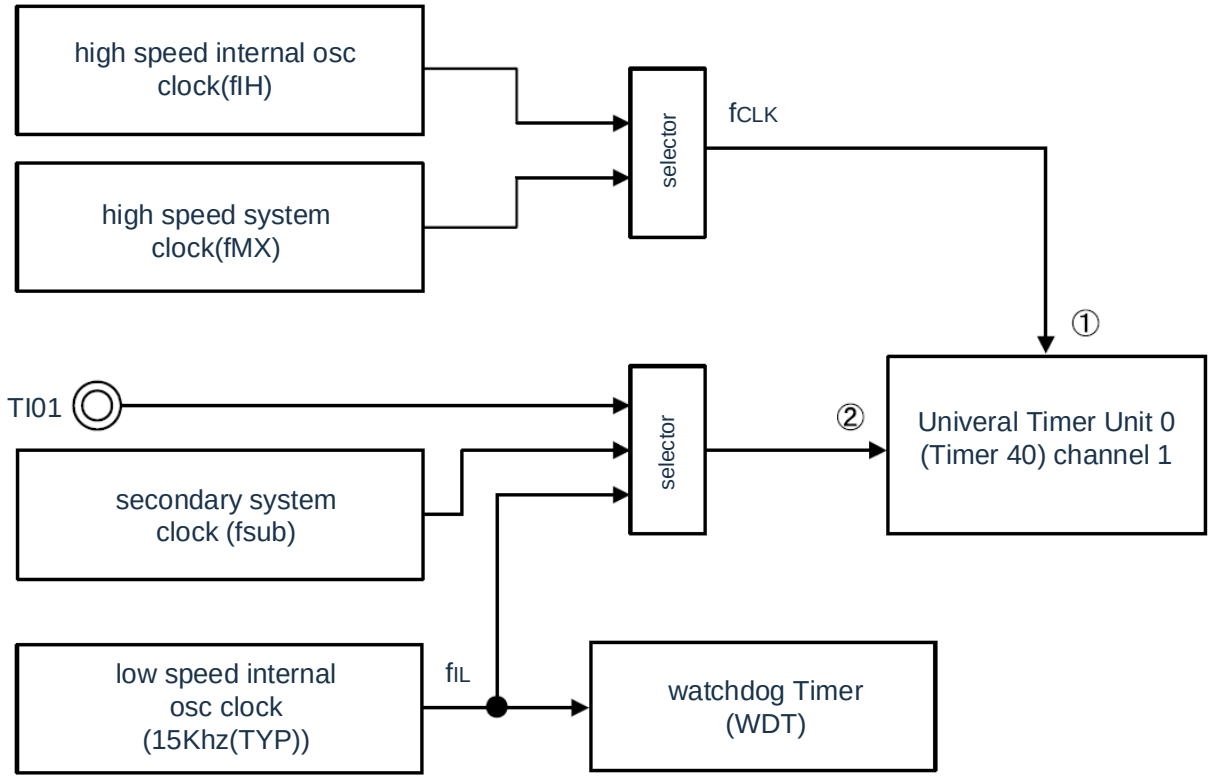
(1) Clock frequency of CPU/peripheral hardware ($_{fCLK}$):

- high speed internal oscillator clock ($_{fIH}$)
- High speed system clock ($_{fMX}$)

(2) Channel 1 input for Timer4:

- Timer input for channel 1 (TI01)
- Low speed internal oscillator clock (fIL:15kHz (TYP.))
- Secondary System Clock ($_{fSUB}$) Note

Figure 31-1 0 Structure of frequency detection function



When the measurement result of the input pulse interval is an abnormal value, it can be judged as "clock frequency abnormal. Refer to "Running as Input Pulse Interval Measurements, 6.8.4" for methods of measuring input pulse intervals.

Note Only products with a secondary system clock built in can be selected.

31.3.5.1 Timer Input Select Register 0 (TIS0).

For register descriptions, refer to Section 6.3.8.

31.3.6 A/D Test Function

The IEC60730 standard requires the A/D converter to be tested. The A/D test function confirms the normal operation of the A/D converter by A/D converting the positive/negative reference voltage, analog input channel (ANI).

The Analog Multiplexer can be confirmed by the following steps:

① Select the ANIx pin as the A/D conversion object (ADTES1, ADTES0=0, 0) through the ADTES register.

② A/D conversion of the ANIx pin (conversion result 1-1).

③ Negative (-) reference voltage of A/D converter is selected as A/D conversion object (ADTES1, ADTES0=1,0) through ADTES register.

④ An A/D conversion is performed on the negative (2-1) reference voltage of the A/D converter.

⑤ Select the ANIx pin as the A/D conversion object (ADTES1, ADTES0=0, 0) through the ADTES register.

⑥ A/D conversion of the ANIx pin (conversion result 1-2).

⑦ Positive (+) reference voltage of A/D converter is selected as A/D conversion object (ADTES1, ADTES0=1,1) through ADTES register.

⑧ A/D conversion of positive (+) reference voltage of A/D converter (conversion result 2-2).

⑨ Select the ANIx pin as the A/D conversion object (ADTES1, ADTES0=0, 0) through the ADTES register.

⑩ A/D conversion of the ANIx pin (conversion result 1-3).

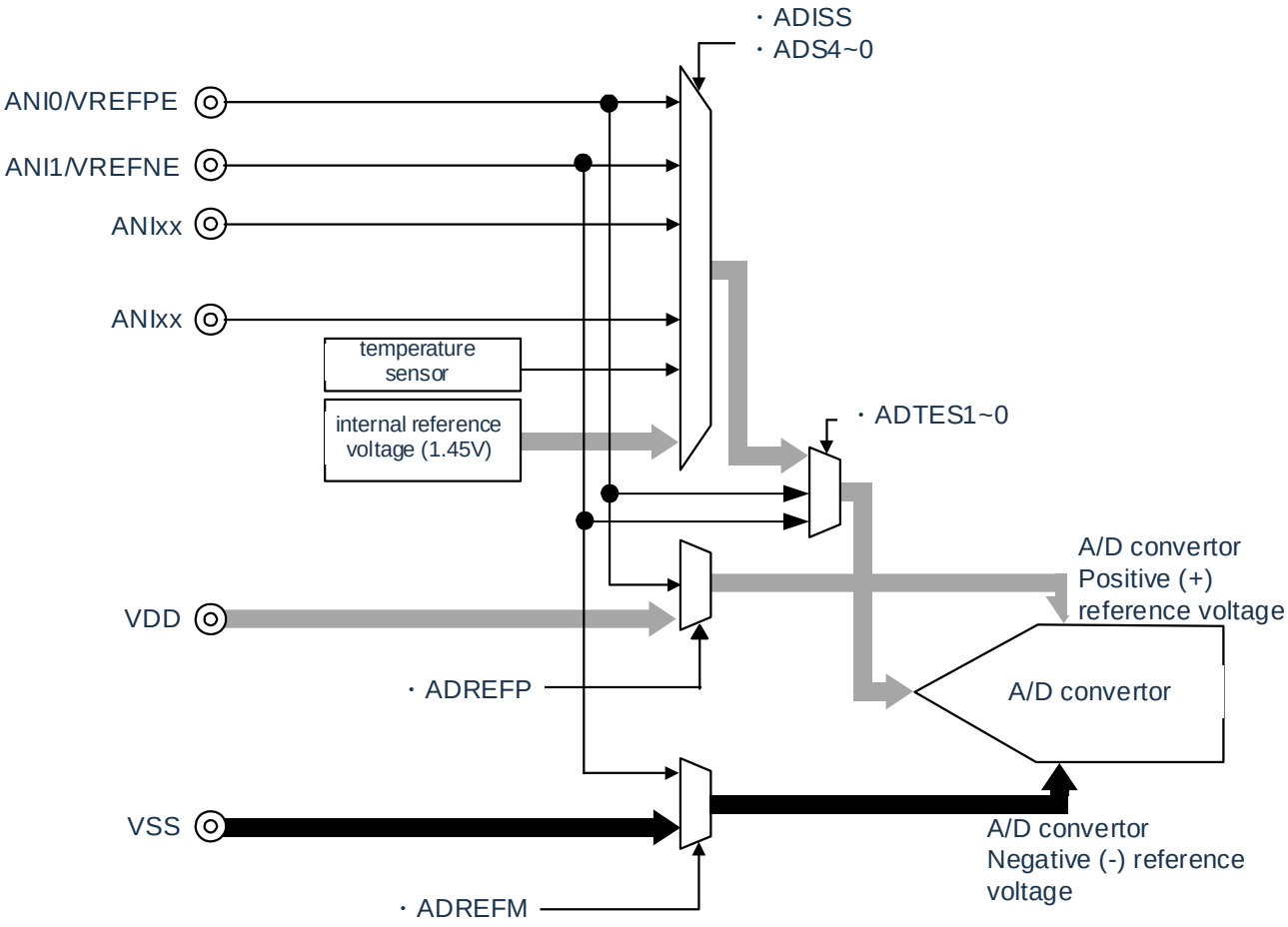
⑪ Confirm that Conversion Results 1-1, Conversion Results 1-2, and Conversion Results 1-3 are identical.

⑫ Confirm that A/D conversion results for Conversion Results 2-1 are all "0" and A/D conversion results for Conversion Results 2. By the above steps, the analog multiplexer can be selected and the wiring is confirmed not broken.

Note 1. During the conversion from (1) to (10), if the analog input voltage is variable, other methods must be used to confirm the analog multiplexer.

2. The conversion results contain errors, so it is necessary to consider the error when comparing the conversion results.

Figure 31-11 Structure of A/D Test Function



31.3.6.1 A/D test register (ADTES)

The register selects positive (+) reference voltage, negative (-) reference voltage, analog input channel (ANIxx), temperature sensor output voltage and internal reference voltage (1.45V) as AD conversion objects.

When used as an A/D test function, set the following:

· When measuring the zero scale, a negative (-) reference voltage is selected as the A/D conversion object.

· When the full scale is measured, a positive (+) reference voltage is selected as the A/D conversion object.

For register descriptions, refer to 15.2.10.

31.3.6.2 Analog input channel assignment register (ADS)

This register specifies the input channel of the analog voltage of the A/D conversion.

When measuring ANIxx, temperature sensor output, or internal reference voltage (1.45V) through A/D test function, the A/D test register (ADTES) must be set.

Refer to 15.2.7 for register descriptions.

31.3.7 Digital output signal level detection function of input/output pin

The IEC60730 standard requires that I/O functionality be confirmed.

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.

31.3.7.1 Port mode selection register (PMS)

This register selects whether the pin is output mode (PMm) with PMmn bit "0" for the read port or the pin output level.

The PMS register is set by an 8-bit memory operation instruction.

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

Figure 31-1 2 Format of Port Mode Selection Register (PMS)

Location: 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	The digital output level of the read pin.

Note 1. For pins that use the pulse output force cutoff function of the timer M to make the pins high impedance, the read value is "0" if the digital output level of the read pin.

Remark m=0~7, 12~14
n=0~7

31.3.8 Product Unique Identity Register

The product's unique ID is perfect for:

- Used as serial number (for example, USB character serial number or other terminal applications)
- When writing flash memory, the unique identification is used in combination with software encryption and decryption algorithm to improve the Safety of the code in the flash memory.
- Used to activate bootstrap process with safety mechanism

The reference number provided by the 128-bit Product Unique Identifier is unique to any BAT32 microcontroller in any case. In no case can the user modify the identity.

Base address: 0x0050_084C

Address Offset: 0x00

Read-only, with values written at factory

U_ID [31:0]

Address Offset: 0x04

Read-only, with values written at factory

U_ID [63:32]

Address Offset: 0x08

Read-only, with values written at factory

U_ID [95:64]

Address Offset: 0x0C

Read-only, with values written at factory

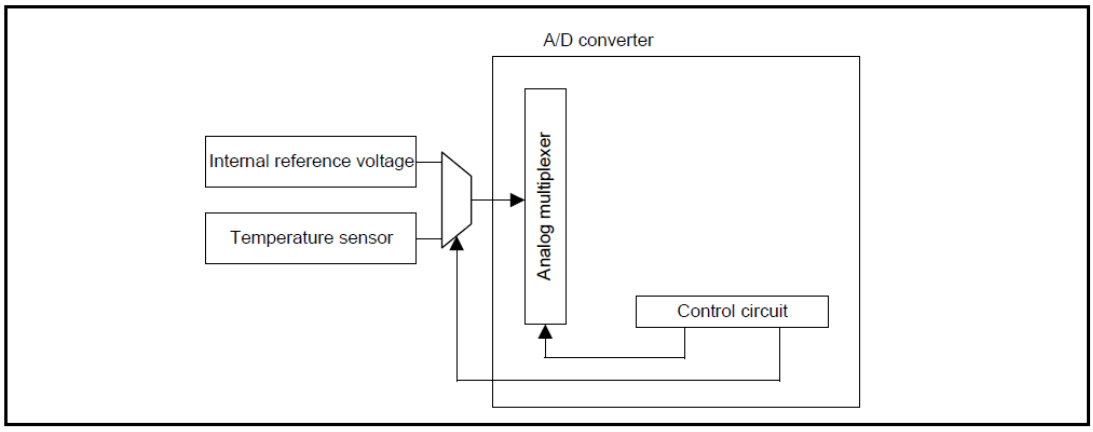
U_ID [127:96]

Chapter 32 temperature sensor

32.1 Function of temperature sensor

The temperature sensor on the chip can measure and monitor the core temperature of the product, thus ensuring the reliable operation of the product. The output voltage of the temperature sensor is proportional to the core temperature, and a linear relationship between the voltage and the temperature. Its output voltage is supply to ADC for conversion. Figure 32-1 shows a block diagram of the temperature sensor.

Figure 32-1temperature sensor block diagram



32.2 Register for temperature sensor

32.2.1 temperature sensor calibration data register TSN25

Address: 0x0050_066C

symbol	7	6	5	4	3	2	1	0	After Reset	R/W
TSN25	TSN25 [7:0]								-	R

Read - only register, used to record the temperature sensor calibration data of 25 degrees, automatically loaded when power on or reset startup, each chip has its own calibration data.

32.2.2 temperature sensor calibration data register TSN85

Address: 0x0050_0668

symbol	7	6	5	4	3	2	1	0	After Reset	R/W
TSN85	TSN85 [7:0]								-	R

A read-only register used to record the 85 degree calibration data of the temperature sensor. It is automatically loaded when powered on or reset to start. Each chip has its own calibration data.

32.3 Instructions for Using a Temperature Sensor

32.3.1 Principle of Using Temperature Sensor

The temperature (T) is proportional to the sensor voltage output (Vs), so the temperature is calculated as follows:

T = (Vs - V1) / slope + T1

T: Temperature Measured (° 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), slope = (V2 - V1)/(T2 - T1).

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 (slope = (V2 - V1)/(T2 - T1))
4. Subsequently, the temperature (T = (Vs -V1)/ slope + T1) is obtained by substituting the slope into the formula of the temperature characteristic.

32.3.2 Method for using temperature sensor

Method 1: In this product, the TSN25 register stores the voltage conversion value (CAL25) of the temperature sensor measured at Ta=Tj=25°C and VDD=3.0v. The TSN85 register stores the voltage conversion value (CAL85) of the temperature sensor measured at Ta=Tj=85°C and VDD=3.0v. Using these two sets of values, the temperature slope can be calculated:

slope = (V2 - V1)/(85 - 25).

V1 = 3.0 × CAL25 / 256 [V]
V2 = 3.0 × CAL85 / 256 [V]

Using the above results, the temperature can be calculated according to the following formula:

T = (Vs - V1) / slope + 25 [°C]

T: Measured temperature (°C)
Vs: 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 Properties, you can calculate the measured temperature directly using the following formula:

T = (Vs - V1) / slope + 25 [°C]

Note: The temperature generated by this method is lower than the accuracy measured by the method 1.

Chapter 33 Option Bytes

33.1 Feature of option bytes

BAT32A237 flash memory 000C0H~000C3H, 50004H~500005H is the option byte area.
The option byte is composed of user option byte (000C0H~000C2H) and flash data protection option byte (000C3H, 50004H~5000H). Automatically refer to option bytes to set the specified function when powered on or reset to start. When you use this product, you must set the following features in option bytes. You cannot change the initial value for a bit that does not have configuration capabilities. To use the boot switching function in the self-programming process, since 000C0H~000C3H is replaced by 010C0H~010C3H, 010C0H~000C3H must be set.

Note Option bytes must be set regardless of whether the features are used.

33.1.1 User Option Bytes (000C0H~000C2H/010C0H~010C2H)

(1) 000C0H/010C0H

- ○ Operation of watchdog timer
  - · Allow or disable the operation of the counter.
  - · Allows or stops the operation of the counter in sleep/deep sleep mode.
- ○ The setting of the overflow time of the watchdog timer
- ○ Settings during window opening of the watchdog timer
- ○ Setting for interval interrupts of watchdog timer
  - · Interrupt with or without interval.

Note 000C0H is replaced by 010C0H when booting a switch, so the 010C0H must be set to the same value as 000C0H.

(2) 000C1H/010C1H

- ○ Settings for the LVD Operation Mode
  - · Interrupt & Reset Mode
  - · reset mode
  - · interrupt mode
  - · LVD is OFF (External reset input using RESETB pins).
- ○ Settings for LVD detection levels (VLVDH, VLVDL, VLVD)

Note 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 operation voltage range indicated by the AC characteristic of the data manual. When the power supply voltage drops, it must be reset by the transfer of deep sleep mode, voltage detection circuit or external reset before the power supply voltage falls below the operation voltage range.
The operation voltage range depends on the setting of the user option byte (000C2H/010C2H).
2. At boot exchange, 000C1H is replaced by 010C1H, so you must set the same value for 010C1H.

(3) 000C2H/010C2H

- ○ Frequency setting of high speed internal oscillator
  - · Select from 1MHz~32MHz, 48MHz, 64MHz.

Note 000C2H is replaced by 010C2H when booting a switch, so the 010C2H must be set to the same value as 000C2H.

33.1.2 Flash Data Protection Option Bytes (000C3H/010C3H,50004H~500005H)

○ Control of Flash Data Protection in Chip Debugging

Level0: Allows read/write/erase operations on flash data via debugger

Level1: The chip full erase operation is allowed to flash data through debugger, and the read and write operation is not allowed.

Level2: Operations on flash data through debugger are not allowed.

○ Control of the boot switching function

· Disables or allows the use of the boot switching feature.

Note 000C3H is replaced by 010C3H when booting a switch, so the 010C3H must be set to the same value as 000C3H.

33.2 Format of user option bytes

Figure 33-1 Format of User Option Bytes (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/No Use of Watchdog Timer Interval Interrupt
0	Interrupt without interval.
1	When 75%+1/2 fILs of the overflow time are reached, interval interruptions occur.

WINDOW1	WINDOW0	Watchdog timer window open period note 2
0	-	Disable from setting.
1	0	75%
1	1	100%

WDTON	Counter Operation Control of Watchdog Timer
0	Disables the operation of the counter (stops counting after resetting is removed).
1	Allow the counter to run (count begins after reset is removed).

WDCS2	WDCS1	WDCS0	Overflow time of watchdog timer (f_{IL}=20kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.2ms)
0	0	1	$2^7/f_{IL}$ (6.4ms)
0	1	0	$2^8/f_{IL}$ (12.8ms)
0	1	1	$2^9/f_{IL}$ (25.6 ms)
1	0	0	$2^{11}/f_{IL}$ (102.4ms)
1	0	1	$2^{13}/f_{IL}$ (409.6ms)
1	1	0	$2^{14}/f_{IL}$ (819.2ms)
1	1	1	$2^{16}/f_{IL}$ (3276.8ms)

WDSTBYON	Counter operation control of watchdog timer (sleep mode)
0	In sleep mode, stop that counter from running note 2.
1	In sleep mode, the counter is allowed to run.

Note 1. When booting a switch, 000C0H is replaced by 010C0H, so you must set the same value for 010C0H.

2. When the WDSTBYON bit is "0", it is independent of the values of the WINDOW1 bit and

WINDOW0 bit, and the window is 100%.

Remark f_{IL}: Clock frequency of low speed internal oscillator

Figure 33-2 Format of user option bytes (000C1H/010C1H) (1/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

· Settings for LVD (Interrupt & Reset Mode)

detection voltage			Setting value for option bytes						
VLVDH		VLVDL	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode Settings	
rise	decline	decline						LVIMDS1	LVIMDS0
2.09V	2.04V	1.84V	0	0	1	0	1	1	0
2.71V	2.65V	2.45V		1	0	0	1		
3.75V	3.67V					0	0		
3.02V	2.96V	2.75V		1	1	0	1		
4.06V	3.98V					0	0		
—			Setting values other than the above is prohibited.						

Note: 000C1H is replaced by 010C1H when booting a switch, so the 010C1H must be set to the same value as 000C1H.

Note: You must write "1" to bit4.

Remark:

1. For details on LVD circuits, refer to Chapter 30 Voltage Detection Circuits."
2. The detection voltage is a TYP. value. For more information, refer to the LVD circuit characteristics in the data guide.

Figure 33-2 Format of user option bytes (000C1H/010C1H) (2/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

· Settings for LVD (reset mode)

detection voltage		Setting value for option bytes						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode Settings	
rise	decline						LVIMDS1	LVIMDS0
2.09V	2.04V	0	0	1	0	1	1	1
2.71V	2.65V		1	0	0	1		
3.02V	2.96V		1	1	0	1		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—		Setting values other than the above is prohibited.						

Note: 000C1H is replaced by 010C1H when booting a switch, so the 010C1H must be set to the same value as 000C1H.

Note: You must write "1" to bit4.

Remark:

1. For details on LVD circuits, refer to Chapter 30 Voltage Detection Circuits."
2. The detection voltage is a TYP. value. For more information, refer to the LVD circuit characteristics in the data guide.

Figure 33-2 Format of user option bytes (000C1H/010C1H) (3/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

· Settings for LVD (interrupt mode)

detection voltage		Setting value for option bytes						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode Settings	
rise	decline						LVIMDS1	LVIMDS0
2.09V	2.04V	0	0	1	0	1	0	1
2.71V	2.65V		1	0	0	1		
3.02V	2.96V		1	1	0	1		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—		Setting values other than the above is prohibited.						

Note 000C1H is replaced by 010C1H when booting a switch, so the 010C1H must be set to the same value as 000C1H.

Note You must write "1" to bit4.

Note 1. For details on LVD circuits, refer to Chapter 30 Voltage Detection Circuits."

2. The detection voltage is a TYP. value. For more information, refer to the LVD circuit characteristics in the data guide.

Figure 33-2 Format of user option bytes (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 pins)

detection voltage		Setting value for option bytes						
VLVDH		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode Settings	
rise	decline						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		Setting values other than the above is prohibited.						

Note 000C1H is replaced by 010C1H when booting a switch, so the 010C1H must be set to the

same value as 000C1H.

Note 1. You must write "1" to bit4.

2. 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 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 is lower than the working voltage range by the sleep mode transfer, the voltage detection circuit or the external reset.

The operation voltage range depends on the setting of the user option byte (000C2H/010C2H).

Note 1.×: Ignore

2. For details on LVD circuits, refer to Chapter 30 Voltage Detection Circuits."

3. The detection voltage is TYP. value. For more information, refer to the LVD circuit characteristics in the data guide.

Figure 33-3 Format of User Option Bytes (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	Clock frequency of high speed internal oscillator	
					fHOCO	fIH
1	1	0	0	0	64MHz	32MHz
1	0	0	0	0	48MHz	48MHz
0	1	0	0	0	32MHz	32MHz
0	0	0	0	0	24MHz	24MHz
0	1	0	0	1	32MHz	16MHz
0	0	0	0	1	24MHz	12MHz
0	1	0	1	0	32MHz	8MHz
0	0	0	1	0	24MHz	6MHz
0	1	0	1	1	32MHz	4MHz
0	0	0	1	1	24MHz	3MHz
0	1	1	0	0	32MHz	2MHz
0	1	1	0	1	32MHz	1MHz
Other than above					Disable from setting.	

Note 000C2H is replaced by 010C2H when booting a switch, so the 010C2H must be set to the same value as 000C2H.

Note

1. You must write "1" to bit 7-5.
2. The operation frequency range and the operation voltage range vary depending on the operation mode of the flash memory. For more information, refer to the AC characteristics of the data guide.
3. Changing the frequency of HOCO requires oscillation stability time, MCU is not allowed to modify FRQSEL4, 3, and power-on reset takes effect.

33.3 Format of Flash Data Protection Option bytes

The format of the Flash Data Protection option bytes is as follows.

Figure 33-4 Flash Data Protection Option Byte (000C3H/010C3H) Format

Address: 000C3H/010C3H Note

symbol	7	6	5	4	3	2	1	0
	OCDEN [7:0]							

Address: 50004H.

symbol	7	6	5	4	3	2	1	0
	OCDM[7:0]							

OCDM	OCDEN	Control of Flash Data Protection
3C	C3	Operations on flash data through debugger are not allowed.
Values other than 3C	C3	The chip full erase operation is allowed to flash data through debugger, and the read and write operation is not allowed.
Other than above		Allows read/write/erase operations on flash data via debugger

Note 000C3H is replaced by 010C3H when booting a switch, so the 010C3H must be set to the same value as 000C3H.

Note that the 50_0004H,50_005H address belongs to a data flash and if you use this address for data storage, you first determine that the value does not cause the protection option to be mistakenly set.

Address: 50005H.

symbol	7	6	5	4	3	2	1	0
	1	1	1	1	1	1	1	BTEN

BTEN	Control of Boot Switching Function
0	Using the Boot Switching feature, 0000H~0FFFH and 1000H~1FFFH content exchange
1	Disable switching feature enable

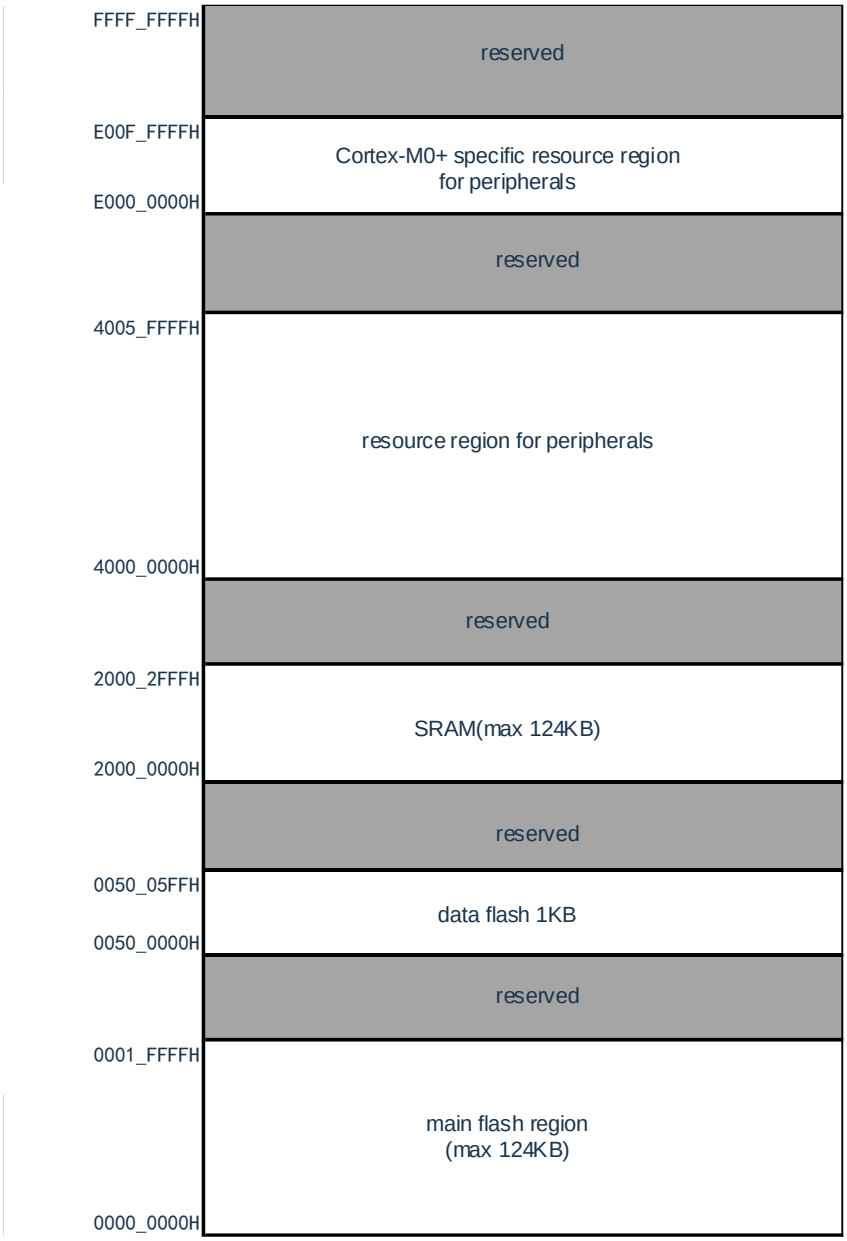
Note that the 50_0004H,50_005H address belongs to a data flash and if you use this address for data storage, you first determine that the value does not cause the protection option to be mistakenly set.

Chapter 34 FLASH control

34.1 FLASH Control Feature Description

The product contains a 128KByte capacity FLASH memory divided into 256 Sector, each Sector capacity of 512Byte. It can be use as program memory and data memory. The module supports erase, programming and read operations of the memory.

34.2 FLASH memory structure



34.3 Register controlling FLASH

The registers that control the FLASH are as follows:

- Flash write protection register (FLPROT)
- Flash Operation Control Register (FLOPMD1, FLOPMD2)
- Flash Erase Mode Control Register (FLERMD)
- Flash Status Register (FLSTS)
- Flash Full Slice Erase Time Control Register (FLCERCNT)
- Flash Page Erase Time Control Register (FLSERCNT)
- Flash write time control register (FLPROCNT)

34.3.1 Flash write protection register (FLPROT)

The Flash protection register is used to protect the FLASH operation control register.

Address: 0x40020020 After reset: 00000000H R/W

symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PRICE [7:1]							WRP

FLPROT

WRP	Operational register (FLOPMD1/FLOPMD2) write protection
0	Override of FLOPMD1/FLOPMD2 not allowed
1	Allow override of FLOPMD1/FLOPMD2

PRICE [7:1]	WRP write protection
78h	Allow override of WRP
Values other than 78 h	Override of WRP not allowed

34.3.2 FLASH Operation Control Register (FLOPMD1, FLOPMD2)

Flash Operation Control Register, which sets the erase and write operations for FLASH.

Address: 0x40020004 After reset: 00000000H R/W

symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	FLOPMD1 [7:0]							

FLOPMD1

Address: 0x40020008 After reset: 00H R/W

symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	FLOPMD2 [7:0]							

FLOPMD2

FLOPMD1	FLOPMD2	ACTION
55	AA	erase
AA	55	Write
00	00	read out
Other than above		Disable from setting

34.3.3 Flash erase control register (FLERMD)

Flash erase control register that sets 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	ACTION
0	0	sector erase without hardware verification
1	0	sector erase, hardware verification after erase
0	1	chip Erase Note
1	1	Disable from setting

NOTE: chip Erase erases only the code flash area and does not erase the data flash area. And chip erase does not support hardware verification.

34.3.4 Flash Status Register (FLSTS)

The state of the FLASH controller can be queried through the state register.

Address: 0x40020000 After reset: 00H R/W

symbol	7	6	5	4	3	2	1	0
FLSTS	0	0	0	0	0	EVF Notes	0	OVF Notes

OVF	FLASH Erase operation completed flag
0	The FLASH erase operation did not complete
1	FLASH erase operation complete

Note: OVF requires software to write "1" for clean-up. If not cleared, the next erase operation cannot occur.

EVF	FLASH Erase Hardware Validation Error Flag
0	No error in hardware verification after FLASH erase
1	Hardware verification error after FLASH erase

Note: EVF requires software to write "1" for clean-up.

34.3.5 Flash Full Slice Erase Time Control Register (FLCERCNT)

The FLCERCNT register allows you to set the time for the FLASH full-slice erase.

Address: 0x40020010 After reset: undecided 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
-	-	-	-	-	-	FLCERCNT [9:0]									

FLCERCNT

Load	Selection Note for Erase Time Settings
0	Erase time using hardware settings
1	Erase time using software settings (FLCERCNT[9:0])

Note: When the main clock is an internal high speed OCO or the external input clock is <=20 M, you can set the time using hardware without setting the FLCERCNT.

FLCERCNT [9:0]	Software Wipe Time Settings
Chip Erase Time = (CERCNT*2048*Tfclk) for >20ms hardware requirements	

34.3.6 Flash Page Erase Time Control Register (FLSERCNT)

The FLSERCNT register allows you to set the time for the FLASH full-slice erase.

Address: 0x40020014 After reset: undecided 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
-	-	-	-	-	-	FLSERCNT [9:0]									

FLSERCNT

Load	Selection Note for Erase Time Settings
0	Erase time using hardware settings
1	Erase time using software settings (FLSERCNT[9:0])

Note: When the main clock is an internal high speed OCO or the external input clock is <=20 M, you can set the time using hardware without setting the FLSERCNT.

FLSERCNT [9:0]	Software Wipe Time Settings
sector Erase Time = (SERCNT*256*Tfclk) for >4 ms hardware requirements	

34.3.7 Flash write time control register (FLPROCNT)

The FLPROCNT register allows you to set the time that the FLASH WORD is written to.

Address: 0x4002001C After reset: undecided R/W

symbol

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]								

FLPROCNT

Load0	Tprog Set Note
0	Write time using hardware settings
1	Erase time using software settings (FLPROCNT[9:0])

Note: When the main clock is an internal high speed OCO or the external input clock is <=20 M, you can set the time using hardware without setting the FLPROCNT.

FLPROCNT [8:0]	Software Wipe Time Settings
Write time = (PROCNT*4*Tfclk), required for >24us hardware requirements	

Load1	Write Action Set-up Time (Tpgs) Note
0	Write action set-up time using hardware settings
1	Erase time using software settings (FLPGSCNT8:0])

Note: When the main clock is an internal high speed OCO or the external input clock is <=20 M, you can set the time using hardware without setting the FLPGSCNT.

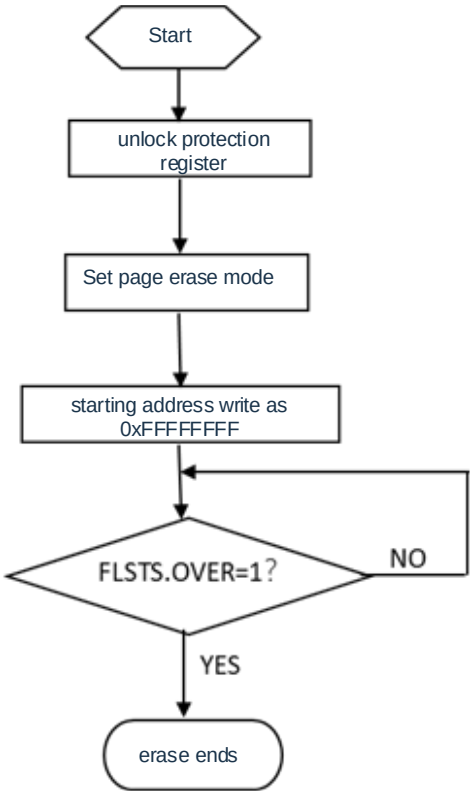
FLPGSCNT [8:0]	Software Wipe Time Settings
Write Action Set-up Time = (PGSCNT*Tfclk) to meet >5us hardware requirements	

34.4 FLASH operation method

34.4.1 Page Erase (sector erase)

sector erase, erase time which is hardware-based and can be configured via FLSERCNT. The operational process is as follows:

1) set FLERMD.ERMD0 to 1'b0, select sector erase mode, select the setting of ERMD 1 value according to whether hardware verification is needed;
2) Set FLPROT to 0xF1 to unprotect FLOPMD. Then set FLOPMD1 to 0x55 and FLOPMD2 to 0xAA.
3) Write any data to the first address of the erase target sector. Example: *((unsigned long *)0x00000200)= 0xffffffff.
4) When the software queries the status register FLSTS.OVF, OVF=1, the erase operation is complete.
5) If you set the erase to perform hardware verification (ERMD1=1), you can determine FLSTS.EVF by software, and query whether correct.
6) Before the next operation, the software places "1" to clear the FLSTS.



34.4.2 Full Chip Erase (chip erase)

chip erase, erase time which is hardware-based and can also be configured via FLCERCNT. The operational process is as follows:

1) set FLERMD.ERMD0 to 1'b1, select chip erase mode;
2) Set FLPROT to 0xF1 to unprotect FLOPMD. Then set FLOPMD1 to 0x55 and FLOPMD2 to 0xAA.
3) Write any data to any address in the code flash area.
4) When the software queries the status register FLSTS.OVF, OVF=1, the erase operation is complete.
5) Before the next operation, the software places "1" to clear the FLSTS.

34.4.3 Programming (word program)

word programming, programming time is implemented by hardware, can also be configured through PROCNT. The operational process is as follows:

1) Set FLPROT to 0xF1 to unprotect FLOPMD. then set FLOPMD1 to 0xAA and FLOPMD2 to 0x55,
2) Write the appropriate data to the destination address.
3) When the software queries the status register FLSTS.OVF, OVF=1, the write operation is complete.
4) Before the next operation, the software places "1" to clear the FLSTS.

34.1 flash read

The fastest fetch finger frequency supported by the FLASH built into this device is 32 MHz. When the HCLK frequency exceeds 32 MHz, the hardware inserts a 1 latency period when the CPU accesses the FLASH.

34.2 Precautions for FLASH Operations

- The FLASH memory has strict time requirements for the control signals of erasing and programming operations, and the inconformity of the timing of the control signals can cause the erasing operation and programming operation to fail. The setting of the erase parameter can be realized by hardware, or can be modified by modifying the parameter register. When using internal high-speed OCO, MAINOSC/external input clock=20 M, it is recommended to use hardware-set erase parameters without setting parameter registers.
- If that erase operation is perform from within the FLASH, the CPU stop fetch and the hardware automatically waits for the operation to complete before proceeding to the next instruction. If that operation is perform from the RAM, the CPU will not stop fetch and can continue with the next instruction.
- When the FLASH is in programming, if the CPU executes instructions to enter deep sleep, the system waits for the programming action to end before entering deep sleep.

Appendix Revision Records

Version	Date	Amendment content
V1.00	Mar 2022	make a preliminary version