TOSHIBA

32-bit TX System RISC
TX03 Series

TMPM322F10FG

Rev.1.03 17th/Dec/2010

TOSHIBA CORPORATION

Semiconductor Company

Modification history

Rev	Date	Page	Item
1.03	10.12.17	-	Release

ARM

TX03 Series 32-bit RISC Microprocessor
TMPM322F10FG

1. Overview and Features

The TX03 Series is comprised of 32-bit RISC microprocessors containing an ARM Cortex™-M3 processor core.

The TMPM322F10FG has the following features.

1.1 Features

(1) ARMCortex-M3 processor core

1) Realizes improved code efficiency through the use of the Thumb-2 instruction set.

- New 16-bit Thumb instructions for improved program flow

- New 32-bit Thumb instructions for improved performance and code density

- 32-bit and 16-bit instruction modes are automatically switched by a compiler.

2) Provides both high performance and low power consumption

- High performance

- A 32-bit multiplication (32 x 32 = 32 bits) can be executed in one clock cycle.

- Division takes 2 to 12 clock cycles.

- Low power consumption

- Optimized design using low power libraries

- Standby function that stops the operation of the processor core

3) High-speed interrupt response suitable for real-time control

- Time-consuming instructions can be aborted by interrupts.

- Stack push is automatically handled by hardware.

(2) On-chip program/data memory

Part Number	On-chip Flash ROM	On-chip RAM
TMPM322F10FG	1024 Kbytes	64 Kbytes

(3) Memory Controller
- Expandable up to 16 Mbytes (Shared program/data area)
- 16-bit width external data bus
- Chip select, wait controller: 4 channel

(4) DMA controller: 2 channels

(5) 16-bit timer: 16 channels
- 16-bit interval timer mode
- 16-bit event counter mode
- Input capture function
- 16-bit PPG output (4-phase synchronous output supported)

(6) Real-time clock (RTC): 1 channel
- Clock (hours, minutes and seconds)
- Calendar (month, week, date and leap year)
- Alarm output
- Alarm interrupt

(7) Watchdog timer: 1 channel
- Watchdog time-out function

(8) General-purpose serial interface: 12 channels
- UART or synchronous mode selectable (built-in 4-byte FIFO)

(9) Serial bus interface: 5 channels
- I^2C bus or clock synchronous mode selectable

(10)Synchronous serial bus interface (SSP): 1 channel
- SPI, SSI and Microwire formats supported

(11)USB host controller: 1 channel
- Compliant with Universal Serial Bus Specification Rev2.0
- Compliant with OpenHCI for USB Release 1.0a
- 12 Mbps (Full-speed)

(12)Remote control signal preprocessor: 2 channels
- Can receive up to 72 bits at a time.

(13) 10-bit AD converter: 16 channels
- Start by an internal timer trigger or external trigger
- Fixed-channel mode and scan mode
- Single mode and repeat mode
- AD monitoring function: 2 channels

(14) Backup module
- Reduces power consumption by shutting off most part of the device except:
  - 8-Kbyte backup RAM
  - ports (port states before entering BACKUP mode can be retained.)
  - remote control signal reception
  - real-time clock
  - key-on wake-up

(15) Interrupts
- 79 internal sources: Priority can be set in seven levels (excluding the watchdog timer interrupt).
- 15 external sources: Priority can be set in seven levels (including NMI).

(16) Input/output ports
- 118 pins: 102 input/output pins + 16 input-only pins

(17) Standby function
- Standby modes: IDLE2, IDLE1, SLEEP, STOP
- Sub clock operation (32.768 kHz) modes: SLOW
- Backup mode (partial shut-off): BACKUP SLEEP, BACKUP STOP

(18) Clock generator
- On-chip PLL (4x)
- Clock gearing: The high-frequency clock can be divided by 1, 2, 4 or 8.

(19) Endian
- Little endian

(20) Maximum operating frequency
- 48 MHz (when using a 12-MHz external resonator)

(21) Operating voltage range
- 3.0 V to 3.6 V (when operating USBHC and using the internal regulator)
- 2.7 V to 3.6 V (when stopping USBHC and using the internal regulator)

(22) Temperature range
- -40°C to 85°C (operations other than Flash write and erase)
- 0°C to 70°C (Flash write and erase)

(23) Package
- LQFP144-P-2020-0.50E (20mm × 20mm, 0.50-mm pitch)

1.2 Block Diagram
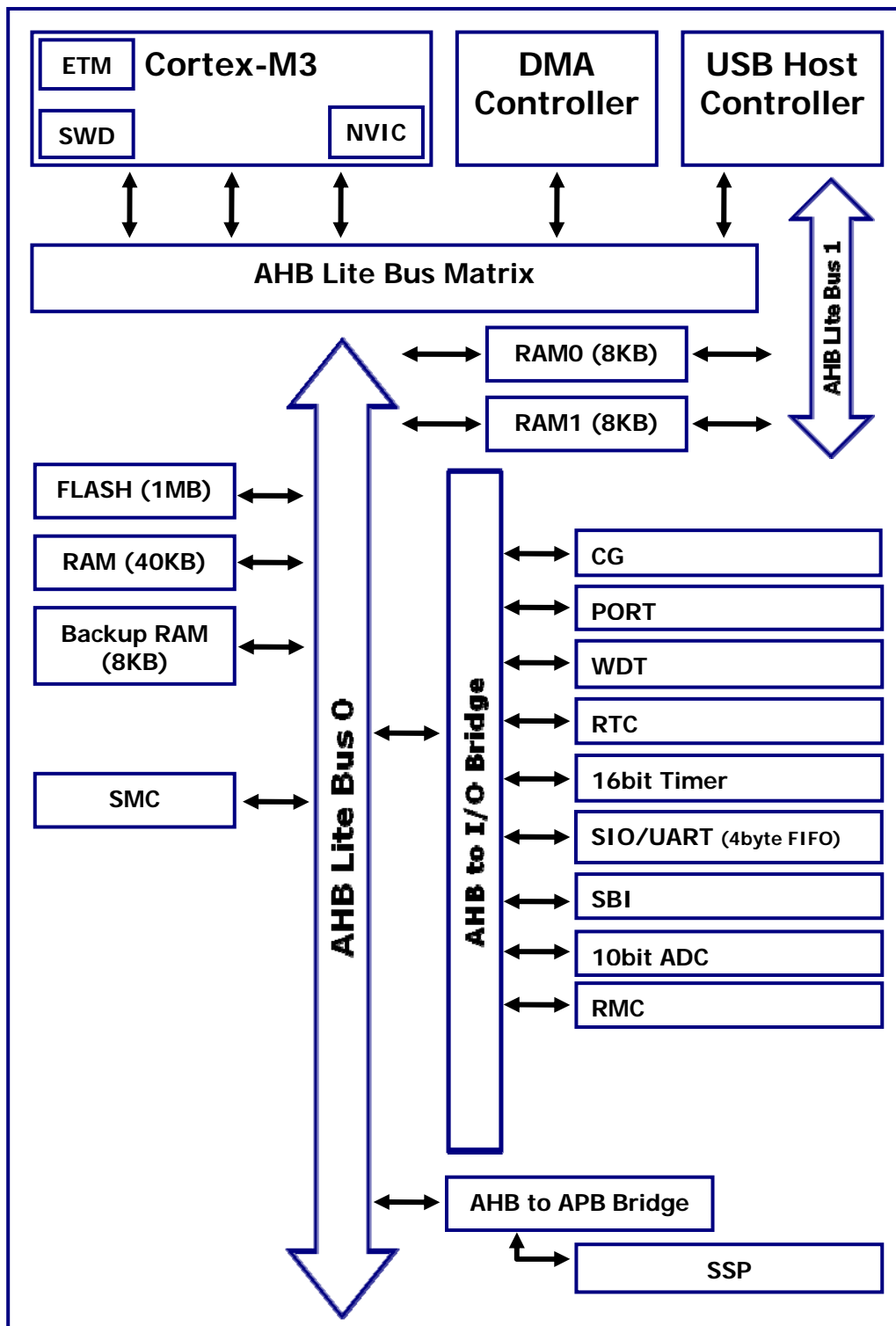
Figure 1.1 TMPM322F10FG Block Diagram

2 Pin Layout and Pin Functions

This chapter describes the pin layout, pin names and pin functions of TMPM322F10FG.

2.1 Pin Layout (Top view)
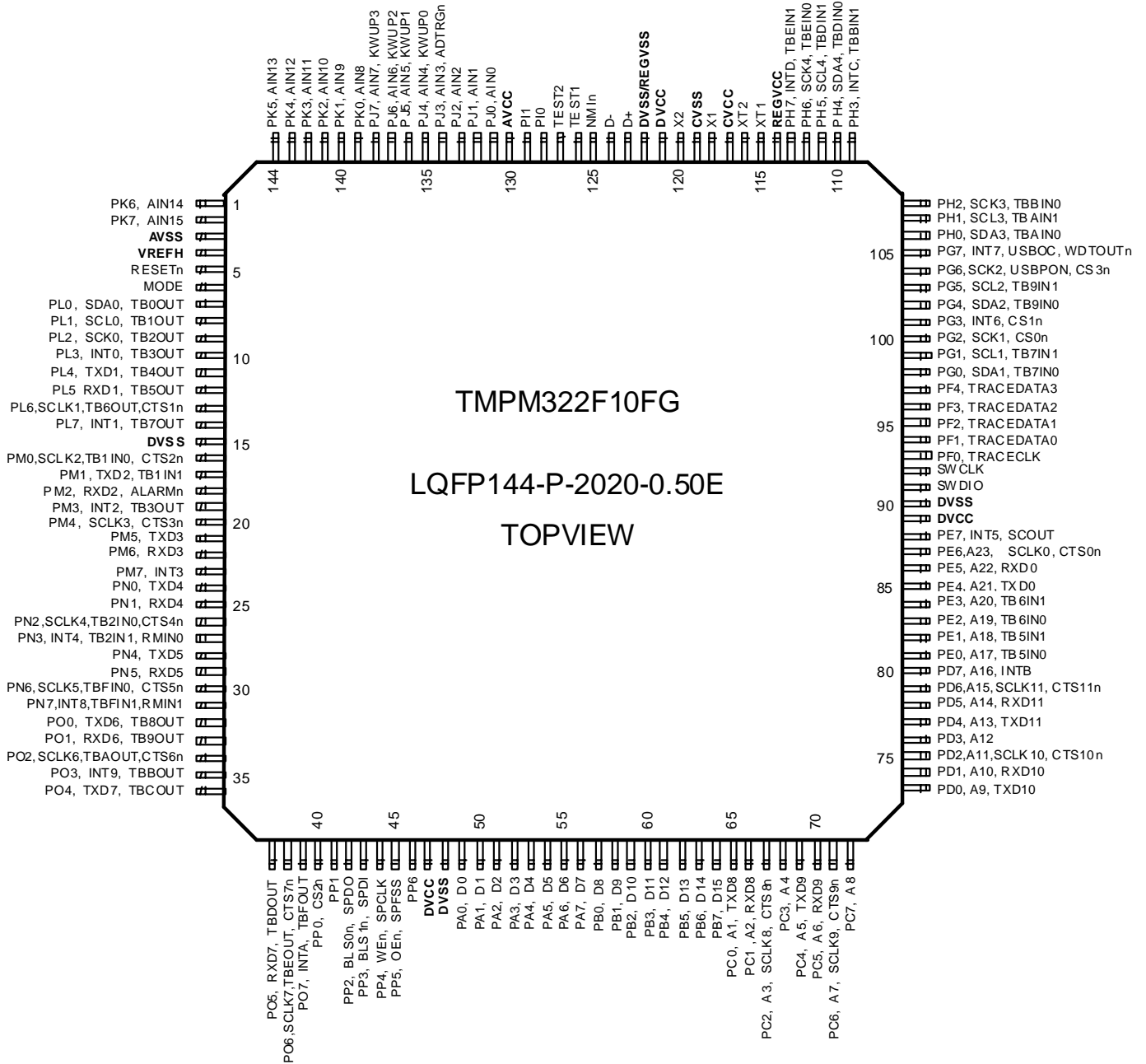
Fig.2.1.1 shows the pin layout of TMPM322F10FG.

Fig.2.1.1 Pin Layout (LQFP144)

2.2   Pin names and Functions

Table 2.2.1 sort the input and output pins of the TMPM322F10FG by pin or port. The table includes alternate pin names and functions for multi-function pins.

2.2.1 Sorted by Pin

Table2.2.1 Pin Names and Functions Sorted by Pin (1/10)

Type	# of Pins	Pin Name	Input/ Output	Function
Function	1	PK6 AN14	I I	Input port Analog input
	2	PK7 AN15	I I	Input port Analog input
PS	3	AVSS	—	A/D converter: GND pin (0V) Tie AVSS to power supply even if the A/D converter is not used.
	4	VREFH	—	Supplying the A/D converter with a reference power supply. Tie VREFH to power supply even if the A/D converter is not used.
RESET	5	RESETn	I	Reset input pin
Control	6	MODE	I	Mode pin: Fix to Low level.
Function	7	PLO SDA0 TB0OUT	I/O I/O O	I/O port If the serial bus interface operates -in the I2C mode : data pin -in the SIO mode: data pin Timer B output
	8	PL1 SCL0 TB1OUT	I/O I/O O	I/O port If the serial bus interface operates -in the I2C mode : clock pin -in the SIO mode: data pin Timer B output
	9	PL2 SCK0 TB2OUT	I/O I/O O	I/O port Inputting and outputting a clock if the serial bus interface operates in the SIO mode. Timer B output
	10	PL3 INT0 TB3OUT	I/O I O	I/O port Interrupt request pin Timer B output
	11	PL4 TXD1 TB4OUT	I/O O O	I/O port Sending serial data Timer B output
	12	PL5 RXD1 TB5OUT	I/O I O	I/O port Receiving serial data Timer B output
	13	PL6 SCLK1 TB6OUT CTS1n	I/O I/O O I	I/O port Serial clock input/ output Timer B output Handshake input pin
	14	PL7 INT1 TB7OUT	I/O I O	I/O port Interrupt request pin Timer B output

Table2.2.1 Pin Names and Functions Sorted by Pin (2/10)

Type	# of Pins	Pin Name	Input/Output	Function
PS	15	DVSS	—	GND pin
Function	16	PM0 SCLK2 TB1IN0 CTS2n	I/O I/O I I	I/O port Serial clock input/ output Inputting the timer B capture trigger Handshake input pin
	17	PM1 TXD2 TB1IN1	I/O O I	I/O port Sending serial data Inputting the timer B capture trigger
	18	PM2 RDX2 ALARMn	I/O I O	I/O port Receiving serial data ALARM output
	19	PM3 INT2 TB3OUT	I/O I O	I/O port Interrupt request pin Timer B output
	20	PM4 SCLK3 CTS3n	I/O I/O I	I/O port Serial clock input/ output Handshake input pin
	21	PM5 TXD3	I/O O	I/O port Sending serial data
	22	PM6 RXD3	I/O I	I/O port Receiving serial data
	23	PM7 INT3	I/O I	I/O port Interrupt request pin
	24	PN0 TXD4	I/O O	I/O port Sending serial data
	25	PN1 RXD4	I/O I	I/O port Receiving serial data
	26	PN2 SCLK4 TB2IN0 CTS4n	I/O I/O I I	I/O port Serial clock input/ output Inputting the timer B capture trigger Handshake input pin
	27	PN3 INT4 TB2IN1 RMIN0	I/O I I I	I/O port Interrupt request pin Inputting the timer B capture trigger Inputting signal to remote controller
	28	PN4 TXD5	I/O O	I/O port Sending serial data
	29	PN5 RXD5	I/O I	I/O port Receiving serial data
	30	PN6 SCLK5 TBFIN0 CTS5n	I/O I/O I I	I/O port Serial clock input/ output Inputting the timer B capture trigger Handshake input pin
	31	PN7 INT8 TBFIN1 RMIN1	I/O I I I	I/O port Interrupt request pin Inputting the timer B capture trigger Inputting signal to remote controller

Table2.2.1 Pin Names and Functions Sorted by Pin (3/10)

Type	# of Pins	Pin Name	Input/ Output	Function
Function	32	PO0 TXD6 TB8OUT	I/O O O	I/O port Sending serial data Timer B output
	33	PO1 RXD6 TB9OUT	I/O I O	I/O port Receiving serial data Timer B output
	34	PO2 SCLK6 TBAOUT CTS6n	I/O I/O O I	/O port Serial clock input/ output Inputting the timer B capture trigger Handshake input pin
	35	PO3 INT9 TBBOUT	I/O I O	I/O port Interrupt request pin Timer B output
	36	PO4 TXD7 TBCOUT	I/O O O	I/O port Sending serial data Timer B output
	37	PO5 RXD7 TBDOUT	I/O I O	I/O port Receiving serial data Timer B output
	38	PO6 SCLK7 TBEOUT CTS7n	I/O I/O O I	I/O port Serial clock input/ output Inputting the timer B capture trigger Handshake input pin
	39	PO7 INTA TBFOUT	I/O I O	I/O port Interrupt request pin Timer B output
	40	PP0 CS2n	I/O O	I/O port Chip select pin
	41	PP1	I/O	I/O port
	42	PP2 BLS0n SPDO	I/O O O	I/O port Bytelane pin Data output pin for SSP
	43	PP3 BLS1n SPDI	I/O O I	I/O port Bytelane pin Data input pin for SSP
	44	PP4 WEn SPCLK	I/O O I/O	I/O port Write strobe pin Clock pin for SSP
	45	PP5 OEn SPFSS	I/O O O	I/O port Output enable pin FSS pin for SSP
	46	PP6	I/O	I/O port
PS	47	DVCC	—	Power supply pin
	48	DVSS	—	GND pin

Table2.2.1 Pin Names and Functions Sorted by Pin (4/10)

Type	# of Pins	Pin Name	Input/ Output	Function
Function	49	PA0 D0	I/O I/O	I/O Port Data bus
	50	PA1 D1	I/O I/O	I/O Port Data bus
	51	PA2 D2	I/O I/O	I/O Port Data bus
	52	PA3 D3	I/O I/O	I/O Port Data bus
	53	PA4 D4	I/O I/O	I/O Port Data bus
	54	PA5 D5	I/O I/O	I/O Port Data bus
	55	PA6 D6	I/O I/O	I/O Port Data bus
	56	PA7 D7	I/O I/O	I/O Port Data bus
	57	PB0 D8	I/O I/O	I/O Port Data bus
	58	PB1 D9	I/O I/O	I/O Port Data bus
	59	PB2 D10	I/O I/O	I/O Port Data bus
	60	PB3 D11	I/O I/O	I/O Port Data bus
	61	PB4 D12	I/O I/O	I/O Port Data bus
	62	PB5 D13	I/O I/O	I/O Port Data bus
	63	PB6 D14	I/O I/O	I/O Port Data bus
	64	PB7 D15	I/O I/O	I/O Port Data bus

Table2.2.1 Pin Names and Functions Sorted by Pin (5/10)

Type	# of Pins	Pin Name	Input/ Output	Function
Function	65	PC0	I/O	I/O Port
		A1	O	Address Bus
		TXD8	O	Sending serial data
	66	PC1	I/O	I/O Port
		A2	O	Address Bus
		RXD8	I	Receiving serial data
	67	PC2	I/O	I/O Port
		A3	O	Address Bus
		SCLK8	I/O	Serial clock input/ output
		CTS8n	I	Handshake input pin
	68	PC3	I/O	I/O Port
		A4	O	Address Bus
	69	PC4	I/O	I/O Port
		A5	O	Address Bus
		TXD9	O	Sending serial data
	70	PC5	I/O	I/O Port
		A6	O	Address Bus
		RXD9	I	Receiving serial data
	71	PC6	I/O	I/O Port
		A7	O	Address Bus
		SCLK9	I/O	Serial clock input/ output
		CTS9n	I	Handshake input pin
	72	PC7	I/O	I/O Port
		A8	O	Address Bus
	73	PD0	I/O	I/O Port
		A9	O	Address Bus
		TXD10	O	Sending serial data
	74	PD1	I/O	I/O Port
		A10	O	Address Bus
		RXD10	I	Receiving serial data
	75	PD2	I/O	I/O Port
		A11	O	Address Bus
		SCLK10	I/O	Serial clock input/ output
		CTS10n	I	Handshake input pin
	76	PD3	I/O	I/O Port
		A12	O	Address Bus
	77	PD4	I/O	I/O Port
		A13	O	Address Bus
		TXD11	O	Sending serial data
	78	PD5	I/O	I/O Port
		A14	O	Address Bus
		RXD11	I	Receiving serial data
	79	PD6	I/O	I/O Port
		A15	O	Address Bus
		SCLK11	I/O	Serial clock input/ output
		CTS11n	I	Handshake input pin
	80	PD7	I/O	I/O Port
		A16	O	Address Bus
		INTB	I	Interrupt request pin

Table2.2.1 Pin Names and Functions Sorted by Pin (6/10)

Type	# of Pins	Pin Name	Input/Output	Function
Function	81	PE0 A17 TB5IN0	I/O O I	I/O port Adress bus Inputting the timer B capture trigger
	82	PE1 A18 TB5IN1	I/O O I	I/O port Adress bus Inputting the timer B capture trigger
	83	PE2 A19 TB6IN0	I/O O I	I/O port Adress bus Inputting the timer B capture trigger
	84	PE3 A20 TB6IN1	I/O O I	I/O port Adress bus Inputting the timer B capture trigger
	85	PE4 A21 TXD0	I/O O O	I/O port Adress bus Sending serial data
	86	PE5 A22 RXD0	I/O O I	I/O port Adress bus Receiving serial data
	87	PE6 A23 SCLK0 CTS0n	I/O O I/O I	I/O port Adress bus Serial clock input/ output Handshake input pin
	88	PE7 INT5 SCOUT	I/O I O	I/O port Interrupt request pin System clock output
PS	89	DVCC	—	Power supply pin
	90	DVSS	—	GND pin
Function/ debug	91	SWDIO	I/O	Debug pin
	92	SWDCK	I/O	Debug pin
	93	PF0 TRACECLK	I/O O	I/O port Debug pin
	94	PF1 TRACEDATA0	I/O O	I/O port Debug pin
	95	PF2 TRACEDATA1	I/O O	I/O port Debug pin
	96	PF3 TRACEDATA2	I/O O	I/O port Debug pin
	97	PF4 TRACEDATA3	I/O O	I/O port Debug pin

Table2.2.1 Pin Names and Functions Sorted by Pin (7/10)

Type	# of Pins	Pin Name	Input/Output	Function
Function	98	PG0	I/O	I/O port
		SDA1	I/O	If the serial bus interface operates
				-in the I2C mode : data pin
				-in the SIO mode: data pin
		TB7IN0	I	Inputting the timer B capture trigger
	99	PG1	I/O	I/O port
		SCL1	I/O	If the serial bus interface operates
				-in the I2C mode : clock pin
				-in the SIO mode: data pin
		TB7IN1	I	Inputting the timer B capture trigger
	100	PG2	I/O	I/O port
		SCK1	I/O	Inputting and outputting a clock if the serial bus interface operates in the SIO mode.
		CS0n	O	Chip select pin
	101	PG3	I/O	I/O port
		INT6	I	Interrupt request pin
		CS1n	O	Chip select pin
	102	PG4	I/O	I/O port
		SDA2	I/O	If the serial bus interface operates
				-in the I2C mode : data pin
				-in the SIO mode: data pin
		TB9IN0	I	Inputting the timer B capture trigger
	103	PG5	I/O	I/O port
		SCL2	I/O	If the serial bus interface operates
				-in the I2C mode : clock pin
				-in the SIO mode: data pin
		TB9IN1	I	Inputting the timer B capture trigger
	104	PG6	I/O	I/O port
		SCK2	I/O	Inputting and outputting a clock if the serial bus interface operates in the SIO mode.
		USBPON	O	USB PON output
		CS3n	O	Chip select pin
	105	PG7	I/O	I/O port
		INT7	I	Interrupt request pin
		USBOCn	I	USB OverCurrent input
		WDTOUTn	O	Watch Dog Timer output

Table2.2.1 Pin Names and Functions Sorted by Pin (8/10)

Type	# of Pins	Pin Name	Input/ Output	Function
	106	PH0 SDA3 TBAIN0	I/O I/O I	I/O port If the serial bus interface operates -in the I2C mode : data pin -in the SIO mode: data pin Inputting the timer B capture trigger
	107	PH1 SCL3 TBAIN1	I/O I/O I	/O port If the serial bus interface operates -in the I2C mode : clock pin -in the SIO mode: data pin Inputting the timer B capture trigger
	108	PH2 SCK3 TBBIN0	I/O I/O I	I/O port Inputting and outputting a clock if the serial bus interface operates in the SIO mode. Inputting the timer B capture trigger
	109	PH3 INTC TBBIN1	I/O I I	I/O port Interrupt request pin Inputting the timer B capture trigger
	110	PH4 SDA4 TBDIN0	I/O I/O I	I/O port If the serial bus interface operates -in the I2C mode : data pin -in the SIO mode: data pin Inputting the timer B capture trigger
	111	PH5 SCL4 TBDIN1	I/O I/O I	/O port If the serial bus interface operates -in the I2C mode : clock pin -in the SIO mode: data pin Inputting the timer B capture trigger
	112	PH6 SCK4 TBEIN0	I/O I/O I	I/O port Inputting and outputting a clock if the serial bus interface operates in the SIO mode. Inputting the timer B capture trigger
	113	PH7 INTD TBDIN1	I/O I I	I/O port Interrupt request pin Inputting the timer B capture trigger

Table2.2.1 Pin Names and Functions Sorted by Pin (9/10)

Type	# of Pins	Pin Name	Input/ Output	Function
PS	114	REGVCC	—	Power supply pin
Clock	115	XT1	I	Connected to a low-speed oscillator.
	116	XT2	O	Connected to a low-speed oscillator.
PS	117	CVCC	—	Power supply pin
Clock	118	X1	I	Connected to a high-speed oscillator.
PS	119	CVSS	—	GND pin
Clock	120	X2	O	Connected to a high-speed oscillator.
PS	121	DVCC	—	Power supply pin
	122	DVSS REGVSS	—	GND pin
Function	123	D+	I/O	USB D+ Pin
	124	D-	I/O	USB D- Pin
	125	NMIn	I	Non-maskable interrupt In case no use, fix to High level.
TEST	126	TEST1	—	OPEN or fix to Low level.
	127	TEST2	—	OPEN or fix to Low level.
Function	128	PI0	I/O	I/O port While "0" inputs RESET pin, PI0 enables input and pull-up. During a reset, "Low level" should not be input to PI0.
	129	PI1	I/O	I/O port (Note 3)
PS	130	AVCC	—	Power supply pin

Table2.2.1 Pin Names and Functions Sorted by Pin (10/10)

Type	# of Pins	Pin Name	Input/ Output	Function
Function	131	PJ0	I	Input port
		AN0	I	Analog input
	132	PJ1	I	Input port
		AN1	I	Analog input
	133	PJ2	I	Input port
		AN2	I	Analog input
	134	PJ3	I	Input port
		AN3	I	Analog input
		ADTRGn	I	Exteral trigger regquest for AD converter
	135	PJ4	I	Input port
		AN4	I	Analog input
		KWUP0	I	Key on Wakeup input
	136	PJ5	I	Input port
		AN5	I	Analog input
		KWUP1	I	Key on Wakeup input
	137	PJ6	I	Input port
		AN6	I	Analog input
		KWUP2	I	Key On Wakeup input
	138	PJ7	I	Input port
		AN7	I	Analog input
		KWUP3	I	Key On Wakeup input
	139	PK0	I	Input port
		AN8	I	Analog input
	140	PK1	I	Input port
		AN9	I	Analog input
	141	PK2	I	Input port
		AN10	I	Analog input
	142	PK3	I	Input port
		AN11	I	Analog input
	143	PK4	I	Input port
		AN12	I	Analog input
	144	PK5	I	Input port
		AN13	I	Analog input

(Note 1)	**MODE pin must be fix to Low-level.**
(Note 2)	**VREFH/AVCC pin must be connected to power supply and AVSS pin must be connected to GND. even if the A/D converter is not used.**
(Note 3)	**Nch open drain port.**

2.3 Pin Names and Power Supply Pins

Table2.3.1 Pin Names and Power Supplies

Pin name	Power supply
PA	DVCC
PB	DVCC
PC	DVCC
PD	DVCC
PE	DVCC
PF	DVCC
PG	DVCC
PH	DVCC
PI	DVCC
PJ	AVCC, AVSS
PK	AVCC, AVSS
PL	DVCC
PM	DVCC
PN	DVCC
PO	DVCC
PP	DVCC
X1, X2	CVCC, CVSS
XT1, XT2	DVCC
RESETn	DVCC
NMIn	DVCC
MODE	DVCC

2.4 Pin Numbers and Power Supply Pins

Table2.4.1 Pin Numbers and Power Supplies

Power supply	Pin number	Voltage range
DVCC	47, 89, 121	3.0V~3.6V (when operating USBHC) 2.7V~3.6V (when stopping USBHC)
REGVCC	114	
CVCC	117	
AVCC	130	
VREFH	4	3.0V~3.6V (when operating USBHC) 2.7V~3.6V (when stopping USBHC) A/D converter with a reference power supply

3. Operation

This section describes the basic components, functions and operation of the TMPM322F10FG.

3.1 System

3.1.1 Processor Core

The TMPM322F10FG has a high-performance 32-bit processor core (the ARM Cortex-M3 processor core). For information on the operations of this processor core, please refer to the "Cortex-M3 Technical Reference Manual" issued by ARM Limited. This chapter describes the functions unique to the TMPM322F10FG that are not explained in that document.

3.1.2 Core Information

The following table shows the revision of the processor core in the TMPM322F10FG. For further information on core revision, see the documents issued by ARM Limited.

Product Name	Core Revision
TMPM322F10FG	R2p0-00rel0

The Cortex-M3 core has the optional blocks. The optional blocks of the revision r2p0 are ETM^{TM} , MPU and WIC. Not MPU, WIC but ETM is contained in the TMPM322F10FG.

3.1.3　Core Block Diagram

The following shows the diagram of the Cortex-M3 core:
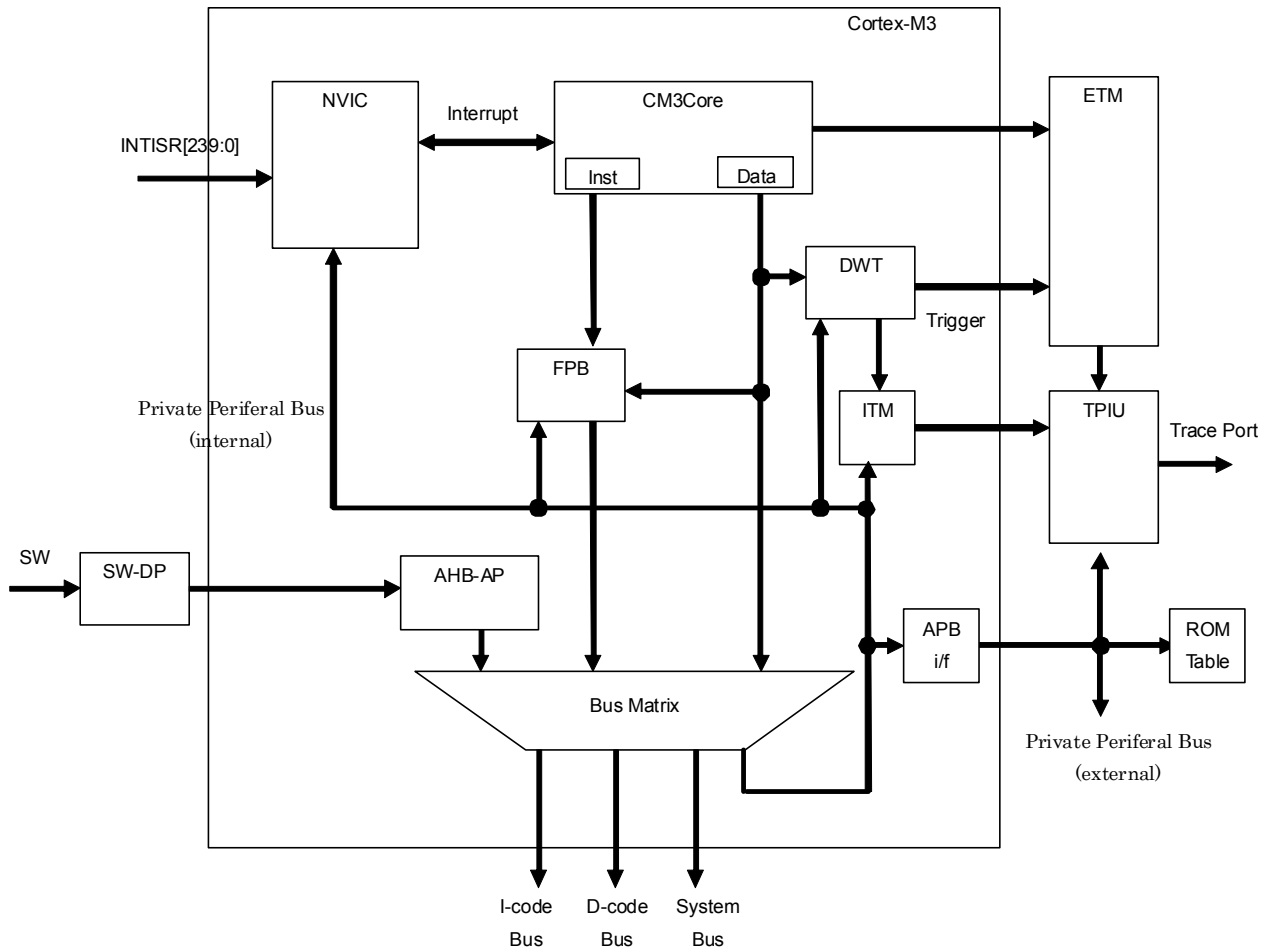
Figure3.1.1　Cortex-M3 Core Block Diagram

3.1.4 Core Configuration

The following shows the Cortex-M3 core configuration selected in the TMPM322F10FG:

Table3.1.1   Core Cofiguration

Configuration	Description
Number of interrupts	100 channels (except Watch Dog Timer Interrupt)
Interrupt priority level (Note 1)	4bit (16 levels)
MPU (Memory Protection Unit)	None
WIC	None
SW / SWJ-DP	SW-DP only (SWD connection only)
ETM (Embedded Trace Macrocell)	Installed
Endian	Little endian

(Note1) Refet to Section5 Exceptions of Cortex-M3 Technical Reference Manual.

3.1.5 Exceptions

The following list shows the Cortex-M3 exception types:

Thethe subsequent areas are the vector areas unique to the TMPM322F10FG. For more information, refer to section 3.5, "Interrupts."

Table 3.1.2   List of Exceptions

Exception	Address	Remarks
Top of Stack	0x0000_0000	Beginning of a stack
Reset	0x0000_0004	Reset
―	0x0000_0008	Reserved
Hard Fault	0x0000_000C	Hard fault
MPU Fault	0x0000_0010	Memory management
Bus Fault	0x0000_0014	Bus fault
Usage Fault	0x0000_0018	Usage fault
―	0x0000_001C	Reserved
―	0x0000_0020	Reserved
―	0x0000_0024	Reserved
―	0x0000_0028	Reserved
SVCall	0x0000_002C	Supervisor call
Debug Monitor	0x0000_0030	Debug monitor
PendSV	0x0000_0038	Software pending request
SysTick	0x0000_003C	SysTick interrupt

3.1.6 Reset

The TMPM322F10 has three reset sources: an external reset pin, WDT and SYSRESETREQ. For reset from the WDT, refer to the chapter on the WDT. For reset from SYSRESETREQ, refer to "Cortex-M3 Technical Reference Manual".

Note : Do not reset with <SYSRESETREQ> in SLOW mode.

3.1.6.1 Cold Reset

The power-on sequence must include the time for the internal regulator and oscillator to be stable. In the TMPM322F10, the internal regulator requires at least 200 μs to be stable. The time required to achieve stable oscillation varies with system. At cold reset, the external reset pin must be kept low for a duration of time sufficiently long enough for the internal regulator and oscillator to be stable. After the external reset (RESETn) signal is released, the internal reset signal remains asserted for a further 400 μs.

Fig.3.1.2 shows the power-on sequence.
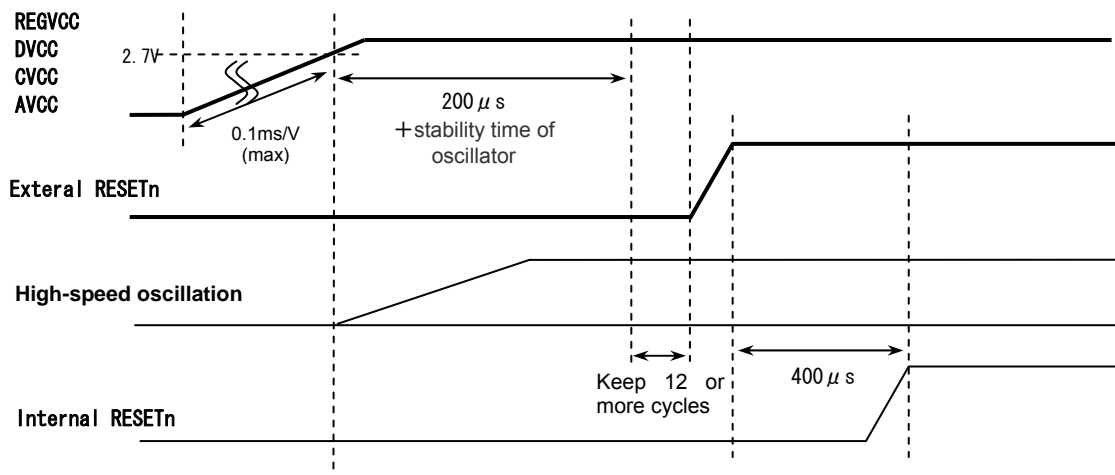
Fig.3.1.2   Cold Reset Sequence

(Note 1)	Turn on the power while the RESETn pin is fixed to "L". Release the RESETn pin while all the power supplies are stabilized within operating voltage.

3.1.6.2   Warm Reset

3.1.6.2.1   Reset Period

As a precondition, ensure that the power supply voltage is within the operating range and the internal high-frequency oscillator is providing stable oscillation. To reset the TMPM322F10, assert the RESETn signal (active low) for a minimum duration of 12 system clocks. After the external reset (RESETn) signal is released, the internal reset signal remains asserted for a further 400 μs.

3.1.6.2.2   After Reset

A warm reset initializes the majority of the Cortex-M3 processor core's system control registers and internal I/O registers. Registers that are only initialized by a cold reset are registers related to the processor core's system debug components (FPB, DWT, ITM), the clock generator's reset flag and the Flash security bit.

After reset, the PLL multiplication circuit is inactive and must be enabled in the CGPLLSEL register if needed.

When the reset exception handling is completed, the program branches to the reset interrupt service routine.

(Note 1)   The reset operation may alter the internal RAM state.

3.1.7   RAMWAIT

A part of RAM(0x2000_4000 to 0x2000_FFFF) is set to "1WAIT" after reset is released.

For performance improvement, set to "0WAIT".

3.1.7.1   RAMWAIT Register

RAMWAIT Resister

RAMWAIT
(0x41FF_F058)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0							
Function	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0							
Function	"0" is read.							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0							
Function	"0" is read.							
	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	-	-	RAM1WAIT
Read/Write	R							R/W
After reset	0							1
Function	"0" is read.							RAMWAIT setting 1:1WAIT 0:0WAIT

<Bit 0><RAM1WAIT>   : Specifies the RAMWAIT.

3.2   Debug Interface

3.2.1   Specification Overview

The TMPM322F10FG contains the SWD (Serial Wire Debug) unit as the debugging interface for connection to the ICE (In-Circuit Emulator), and the ETM (Embedded Trace Macrocell) unit for tracing and outputting internal programs. ETM outputs signals to dedicated pins (TRACEDATA[0] to [3]) through the TPIU (Trace Port Interface Unit) in the microprocessor.
For more information on SWD, ETM, and TPIU, please refer to the documents published by ARM.

Note that the TMPM322F10FG does not support connections based on the JTAG (Joint Test Action Group) Standards. Use SWD-compatible tools when connecting the TMPM322F10FG to the ICE.

3.2.2   Features of SWD

SWD supports the two-pin Serial Wire Debug Port (SWCLK, SWDIO)

3.2.3   Features of ETM

ETM supports four data signal pins (TRACEDATA[0]-[3]), one clock signal pin (TRACECLK).

3.2.4   Pin Functions

The Trace output pins(PF0 to PF4) of debug interface pins can also be used as general-purpose ports.

After reset, the Trace output pins are configured as general-purpose ports. These pins need to be programmed as required.

The table below summarizes the debug interface pin functions and related port settings after reset.

Pin Number	Port (Bit Name)	Debug Function	Port Settings after Reset
91	-	SWDIO	Debug pin(always Pull-Up)
92	-	SWCLK	Debug pin(always Pull-Down)
93	PF0	TRACECLK	general-purpose port
94	PF1	TRACEDATA0	general-purpose port
95	PF2	TRACEDATA1	general-purpose port
96	PF3	TRACEDATA2	general-purpose port
97	PF4	TRACEDATA3	general-purpose port

3.2.5   Conection with a Debug Tool

For how to connect a debug tool, refer to the method recommended by each manufacturer.

3.3　Memory Map

3.3.1　Memory Map

The memory maps for the TMPM322F10FG is based on the ARM Cortex-M3 processor core memory map. The internal ROM, internal RAM and internal I/O of the TMPM322F10FG is mapped to the code, SRAM and peripheral regions of the Cortex-M3 respectively. The SRAM and internal I/O regions are all included in the bit-band region.

The CPU register region is the processor core's internal register region.

For more information on each region, see the "Cortex-M3 Technical Reference Manual".

Note that access to regions indicated as "Fault" causes a memory fault if memory faults are enabled or a hard fault if memory faults are disabled. Do not access the vendor-specific region.

3.3.2 Memory Map of TMPM322F10

Fig.3.3.1 shows the memory map of the TMPM322F10FG.

TMPM322F10FG

Cortex-M3
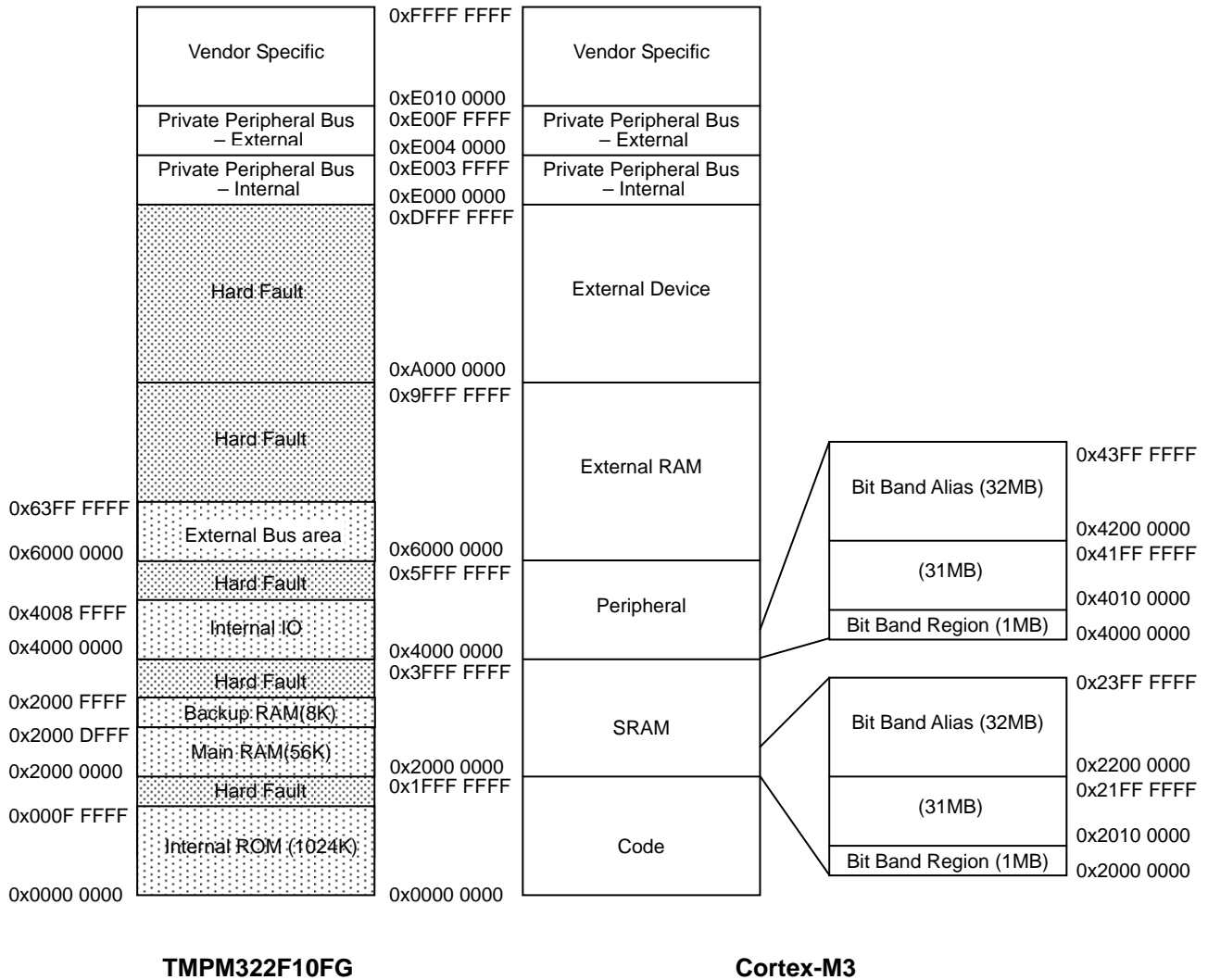
Fig.3.3.1   Memory Map (TMPM322F10FG)

A part of RAM(0x2000_4000 to 0x2000_FFFF) is set to "1WAIT" after reset is released.

For performance improvement, set to "0WAIT". For detail , refer to 3.1.7 RAMWAIT.

3.4 Clock/Mode Control

3.4.1 Features

The clock/mode control block enables to select clock gear, prescaler clock and warm-up of the PLL (including clock multiplication circuit) and oscillator.

The low power consumption mode can reduce power consumption.

This chapter describes how to control clocks, clock operating modes and mode transitions.

The clock/mode control block has the following functions:

- Controls the system clock.
- Controls the prescaler clock.
- Controls the PLL multiplication circuit.
- Controls the warm-up timer.

In addition to NORMAL mode, the TMPM322F10 can operate in various types of low power mode to reduce power consumption according to its usage conditions.

3.4.2 Registers

3.4.2.1 Register List

Table 3.4.1 shows registers and addresses of the clock generator.

Table 3.4.1 Registers of Clock Generator

Register name		Address
System control register	CGSYSCR	400F_4000H
Oscillation control register	CGOSCCR	400F_4004H
Standby control register	CGSTBYCR	400F_4008H
PLL selection register	CGPLLSEL	400F_400CH
System clock selection register	CGCKSEL	400F_4010H

3.4.2.2　System Control Register

CGSYSCR
(0x400F_4000)

	7	6	5	4	3	2	1	0
Bitsymbol							GEAR	
Read/Write	R					R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.					High-speed clock (fc) gear 000: fc　　　　100: fc/2 001: reserved　101: fc/4 010: reserved　110: fc/8 011: reserved　111: Reserved		

	15	14	13	12	11	10	9	8
Bitsymbol			FPSEL1	FPSEL0			PRCK	
Read/Write	R		R/W	R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.		fperiph select1 (Note) 0:depend on FPSEL0 setting 1: fs	fperiph select 0 0: fgear 1: fc	"0" is read.	Prescaler clock 000: fperiph/1　100: fperiph/16 001: fperiph/2　101: fperiph/32 010: fperiph/4　110: Reserved 011: fperiph/8　111: Reserved		

	23	22	21	20	19	18	17	16
Bitsymbol	USBHRES			FCSTOP			SCOSEL	
Read/Write	R/W	R		R/W	R		R/W	R/W
After reset	0	0	0	0	0	0	0	1
Function	USB HOST Controller RESET 0:RESET release 1:RESET	"0" is read.		A/D converter clock 0: supply 1: stop	"0" is read.		SCOUT output 00:fs 01:fsys/2 10:fsys 11:φT0	

	31	30	29	28	27	26	25	24
Bitsymbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

<Bit 2:0><GEAR>　　　　　: Specifies the high-speed clock (fc) gear.

<Bit 10:8><PRCK>　　　　: Specifies the prescaler clock to peripheral I/O.

<Bit 12><FPSEL0>　　　　: Specifies the source clock to fperiph.
　　　　　　　　　　　　　　When fc is selected, fperiph can be fixed irrespective of the clock gear setting.

<Bit 13><FPSEL1>　　　　: Selects the low-frequency clock (fs) as the source clock to fperiph.

<Bit 17:16><SCOSEL>　　: Enables to output the specified clock from SCOUT pin.

<Bit 20><FCSTOP>　　　　: Can be used to stop clock supply to the AD converter.
　　　　　　　　　　　　　　After reset, clock supply to the AD converter is enabled.
　　　　　　　　　　　　　　Before setting this bit to "1" to stop clock supply to the AD converter, make sure that the AD converter is not performing any conversion. The state of the AD converter can be checked by reading the ADMOD0<bit6> and ADMOD2<bit7>.

<Bit 23><USBHRES>　　　: Can be used to reset the USB HOST controller.

(Note)　CGSYSCR<FPSEL1> must be set to same value with CGCKSEL<SYSCK>.

3.4.2.3 Oscillation Control Register

CGOSCCR
(0x400F_4004)

	7	6	5	4	3	2	1	0
Bitsymbol					WUPSEL	PLLON	WUEF	WUEON
Read/Write	R/W				R/W	R/W	R	W
After reset	0	0	1	1	0	0	0	0
Function	Always Write "0011".(Note1)				Warm-up counter 0: X1 1: XT1	PLL operation 0: Stop 1: Oscillation	Status of Warm-up timer (WUP) 0:warm-up Completed 1:Warm-up in operation	Operation of warm-up timer (WUP) 0:don't care1:starting warm-up

	15	14	13	12	11	10	9	8
Bitsymbol	WUPTL		Reserved				XTEN	XEN
Read/Write	R/W		R/W		R		R/W	R/W
After reset	0	0	0	0	0	0	1	1
Function	Lower 2 bits of warm-up counter for low-speed oscillator		Write "0".		"0" is read.		Low-speed oscillator 0: Stop 1: Oscillation	High-speed oscillator 0: Stop 1: Oscillation

	23	22	21	20	19	18	17	16
Bitsymbol	WUPT							
Read/Write	R/W				R			
After reset	0	0	0	0	0	0	0	0
Function	Middle 4 bits of warm-up for high-speed/low-speed oscillator				"0" is read.			

	31	30	29	28	27	26	25	24
Bitsymbol	WUPT							
Read/Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	Upper 8 bits of warm-up counter for high-speed/low-speed oscillator							

<Bit 0><WUEON> : Enables to start the warm-up timer.

<Bit 1><WUEF> : Enables to monitor the status of the warm-up timer.

<Bit 2><PLLON> : Specifies operation of the PLL.
It stops after reset. Setting the bit is required.

<Bit 3><WUPSEL> : Specifies the oscillator to warm-up. A clock generated by the specified oscillator is used for the warm-up timer count.

<Bit 8><XEN> : Specifies operation of the high-speed oscillator.

<Bit 9><XTEN> : Specifies operation of the low-speed oscillator.
The low-speed oscillator must be stopped when it is not used.

<Bit 31:20,15:14><WUPT><WUPTL> : Used to specify the warm-up time for the high-speed/low-speed oscillator.
(Note2)　　　　　　　　　　The warm-up time is set, as shown below, with16 bits for the high-speed oscillator and with 18 bits for the low-speed oscillator with the lower 4 bits masked in both cases.
Warm-up time for high-speed oscillator: <Bit31:20><WUPT>
Warm-up time for low-speed oscillator: <Bit31:20,15:14><WUPT><WUPT>

> **(Note 1)** For warm-up setting, see "3.4.6.8 Warm-up".
>
> **(Note 2)** This field must always be set to "0011". Any other settings are prohibited.

3.4.2.4 Stadby Control Register

CGSTBYCR
(0x400F_4008)

	7	6	5	4	3	2	1	0
Bitsymbol							STBY	
Read/Write			R			R/W	R/W	R/W
After reset	0	0	0	0	0	0	1	1
Function	"0" is read.					Low power consumption mode 000 : Reserved 001 : STOP 010 : SLEEP 011 : IDLE2 100 : Reserved 101 : BACKUP STOP 110 : BACKUP SLEEP 111 : IDLE1		

	15	14	13	12	11	10	9	8
Bitsymbol							RXTEN	RXEN
Read/Write			R				R/W	R/W
After reset	0	0	0	0	0	0	0	1
Function	"0" is read.						Low-speed oscillator after releasing STOP mode 0: Stop 1: Oscillation	High-speed oscillator after releasing STOP mode 0: Stop 1: Oscillation

	23	22	21	20	19	18	17	16
Bitsymbol							PTKEEP	DRVE
Read/Write			R				R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.						IO Port control in the Backup mode. 0: Controllable 1:Hold On(Note 2)	Pin status in STOP mode 0: Active 1: Inactive

	31	30	29	28	27	26	25	24
Bitsymbol								
Read/Write				R				
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							

<Bit 2:0><STBY> : Specifies the low power consumption mode. (Note1)

<Bit 8><RXEN> : Specifies the high-speed oscillator operation after releasing the STOP mode.

<Bit 9><RXTEN> : Specifies the low-speed oscillator operation after releasing the STOP mode.

<Bit 16><DRVE> : Specifies the pin status in the STOP mode.

<Bit 17><PTKEEP> :Holds the I/O Port setting in the Backup Mode by <DRVE>="1".

(Note 1)	**The values indicated as "reserved" must not be specified.**
(Note 2)	**I/O Port holding effects all port except for port I, J, L, M and N by CGSTBYCR<PTKEEP>="1".**

3.4.2.5　PLL Selection Register

	7	6	5	4	3	2	1	0
Bitsymbol	ND					Reserved		PLLSEL
Read/Write	R/W					R/W		R/W
After reset	0	0	0	1	1	1	1	0
Function	Clcok multiplied by the PLL 00011: 4times Other: setting prohibition					After Reset "11" is set. Always wWrite "11".		Use of PLL 0: Disuse X1 selected 1: Use

	15	14	13	12	11	10	9	8
Bitsymbol	RS					IS		C2S
Read/Write	R/W				R	R/W		R/W
After reset	0	1	1	1	0	0	1	0
Function	Clcok multiplied by PLL 0111: 4times Other: setting prohibition				"0"is read	Clcok multiplied by PLL 01: 4times Other: setting prohibition		Clcok multiplied by PLL (C2S) 0: 4times 1: setting prohibition

	23	22	21	20	19	18	17	16
Bitsymbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0"is read							

	31	30	29	28	27	26	25	24
Bitsymbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0"is read							

CGPLLSEL (0x400F_400C)

<Bit 0><PLLSEL>　:　Specifies use or disuse of the clock multiplied by the PLL.

"X1" is automatically set after reset. Resetting is required when using the PLL.

<Bit 15:12,10:3><ND,C2S,IS,RS> :
Specifies of the clock multiplied by the PLL .
After reset 4times is set, When it set CGOSCCR<PLLON>="1",
CGPLLSEL<PLLSEL>="1"(PLL usage)
the 4 times clock mulipled of fosc can be used.

3.4.2.6 System Clock Selection Register

		7	6	5	4	3	2	1	0
CGCKSEL (0x400F_4010)	Bitsymbol	–	–	–	–	–	–	SYSCK	SYSCKFLG
	Read/Write	R						R/W	R
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.						System clock 0: High-speed (fc) 1: Low-speed (fs)	System clock status 0: High-speed (fc) 1: Low-speed (fs) Stable oscillation identical with <SYSCK> value.
		15	14	13	12	11	10	9	8
	Bitsymbol	–	–	–	–	–	–	–	–
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							
		23	22	21	20	19	18	17	16
	Bitsymbol	–	–	–	–	–	–	–	–
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							
		31	30	29	28	27	26	25	24
	Bitsymbol	–	–	–	–	–	–	–	–
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							

<Bit 0><SYSCKFLG> : Shows the status of the system clock.
Switching the oscillator with <SYSCK> generates time lag to complete. If the output of the oscillator specified in <SYSCK> is read out by <SYSCLKFLG>, the switching has been completed.

<Bit 1><SYSCK> : Enables to specify the system clock. Setting CGOSCCR<XEN> and <XTEN> to "1" in advance is required.

3.4.3 Clock Control

3.4.3.1 Clock System Block Diagram

Ｆｉｇ. 3. 4. 1 shows the clock system diagram. Each clock is defined as follows.

fosc   : Clock input from the X1 and X2 pins
fs   : Clock input from the XT1 and XT2 (low-speed clock)
fpll   : Clock quadrupled by PLL
fc   : Clock specified by CGPLLSEL<PLLSEL> (high-speed clock)
fgear   : Clock specified by CGSYSCR<GEAR>
fsys   : Clock specified by CGCKSEL<SYSCK> (system clock)
fperiph : Clock specified by CGSYSCR<FPSEL>
ΦT0   : Clock specified by CGSYSCR<PRCK> (prescaler clock)

The high-speed clock fc and the prescaler clock ΦT0 are dividable.

- High-speed clock: fc, fc/2, fc/4, fc/8

- Prescaler clock: fperiph, fperiph/2, fperiph/4, fperiph/8, fperiph/16,fperiph/32

3.4.3.2 Initial Values after Reset

Reset initializes the clock configuration as follows.

High-speed oscillator	: ON (oscillating)
Low-speed oscillator	: ON (oscillating)
PLL (phase locked loop circuit)	: OFF (stop)
High-speed clock gear	: fc (no frequency dividing)

Reset causes all the clock configurations excluding the low-speed clock (fs) to be the same as fosc.

fc = fosc
fsys = fosc
ΦT0 = foscPLL

For example, reset configures fsys as 12MHz when a 12MHz oscillator is connected to the X1 or X2 pin.
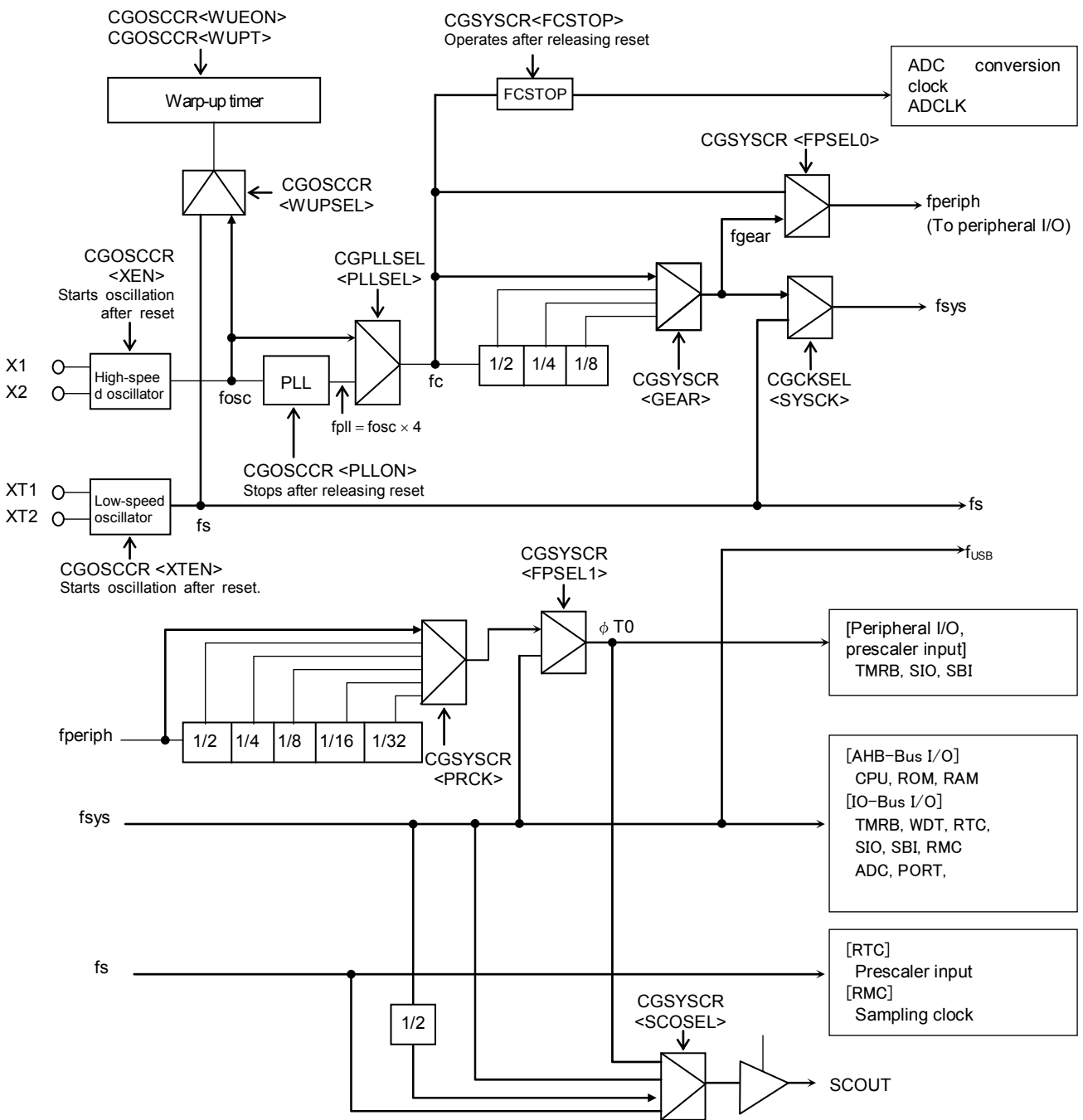
Fig. 3.4.1 Clock Block Diagram

(Note)   The input clocks to selector shown with an arrow are set as default after reset.

3.4.3.3 Clock Multiplication Circuit (PLL)

This circuit outputs the fpll clock that is quadruple of the high-speed oscillator output clock, fosc. This lowers the oscillator input frequency while increasing the internal clock speed.

The PLL is disabled after reset is released. To enable the PLL, set "1" to the CGOSCCR<PLLON> bit. The PLL requires a certain amount of time to be stabilized, which should be secured using the warm-up function.

(Note) **When the multiplication setting is changed, the PLL requires approximately 100 µs to stabilize.**
The PLL requires approximately 200 µs to stabilize after it is activated.

The following shows PLL setting saequnece after Reset release.

Operation Flow   Notes

Initial values after reset release CGOSCCR<PLLON>="0"(PLL Stopped) CGPLLSEL<PLLSEL>="0"(1x) CGPLLSEL<FS,ND,C2S,IS,RS>="4x" CGSYSCR<GEAR>= 1/1 (clock gear)

The high-speed oscillator and supply voltage must be stabilized before the PLL can be activated.

↓

Enable PLL operation CGOSCCR<PLLON>="1"(PLL active)

When the PLL is activated, CGOSCCR0<PLLON> must be held at "1" for approximately 200µs or longer for the PLL to stabilize.

↓

PLL operation selected CGPLLSEL<PLLSEL>="1"(PLL input)

↓

The system clock multiplied by PLL can be used.

Figure 3.4.2   PLL Setting Sequence after Reset Release

3.4.3.4 Warm-up Function

The warm-up function secures the stability time for the oscillator and the PLL with the warm-up timer. For further information, see "Warm-up".

The warm-up function is configured as described below.

The warm-up time is programmable through CGOSCCR<WUPT>. The warm-up timer is started by setting the CGOSCCR<WUEON>. Whether or not the warm-up is completed can be checked by reading CGOSCCR<WUEF>. After the completeion of warm-up is confirmed, switch the system clock by setting the CGCKSEL<SYSCK>.

When clock switching occurs, the current system clock can be checked by monitoring the CGCKSEL<SYSCKFLG>.

The warm-up time is calculated by:

$$\text{Number of warm-up cycles} = \frac{\text{Warm-up time to be set}}{\text{Input frequency cycle (s)}}$$

<Example 1>

Setting the warm-up time to 5 ms when using a high-speed 8-MHz oscillator

$$\frac{\text{Warm-up time to be set}}{\text{Input frequency cycle (s)}} = \frac{5\text{ms}}{1/8\text{MHz}} = 40{,}000 \text{ cycles}$$

$$= 9\text{C40 H (hex)}$$

Drop the lower 4 bits and set 9C4H to CGOSCCR<WUPT>.

```
Bit        31 30 29 28 27 26 25 24 23 22 21 20
9C4 H       1  0  0  1  1  1  0  0  0  1  0  0 b
```

<Setup example>

Switching from SLOW mode to NORMAL mode

```
CGOSCCR<WUPT> ="1001_1100_0100 ": Select warm-up time.
CGOSCCR＜WUPT＞ read              : Confirm the new warm-up time.
CGOSCCR<XEN>="1"        : Enable high-speed oscillation (fosc)
CGOSCCR<WUON>="1"       : Start the warm-up timer (WUP).
CGOSCCR<WUEF> read      : Wait unitl <WUEF>="0" (WUP completed).
CGCKSEL<SYSCK>="0"      : Switch the system clock to high-speed (fgear).
CGCKSEL<SYSCKFLG>read   : Confirm <SYSCKFLG>="0" (current system
                             clock= fgear)
CGOSCCR＜XTEN＞="0"      : Disable low-speed oscillation (fs).
                          (In a dual-clock system, this step is not required.)
```

<Example 2>
Setting the warm-up time to 1s when using a low-speed 32-kHz oscillator

$$\frac{\text{Warm-up time to be set}}{\text{Input frequency cycle (s)}} = \frac{1s}{1/32kHz} = 32{,}000 \text{ cycles}$$

$$= 7D00H \text{ (hex)}$$

Drop the lower 4 bits and set 7D0H to CGOSCCR<WUPT><WUPTL>.

Bit	31 30 29 28 27 26 25 24 23 22 21 20 15 14
7D0H	0 0 0 1 1 1 1 1 0 1 0 0 0 0 b

<Setup example>
Switching from NORMAL mode to SLOW mode

CGOSCCR<WUPTL>="00 "	: Select warm-up time.
CGOSCCR<WUPT>="0001_1111_0100 "	: Select warm-up time.
CGOSCCR<WUPTL>read	: Confirm the new warm-up time.
CGOSCCR<WUPT> read	: Confirm the new warm-up time.
CGOSCCR<XTEN>="1"	: Enable low-speed oscillation (fs).
CGOSCCR<WUEON>="1"	: Start the warm-up timer (WUP).
CGOSCCR<WUEF>read	: Wait until <WUEF>="0" (WUP completed).
CGCKSEL<SYSCK>="1"	: Switch the system clock to low-speed (fs).
CGCKSEL<SYSCKFLG>read	: Confirm <SYSCKFLG>="1" (current sytem clock = fs).
OSCCR1<XEN>="0"	: Disable high-speed oscillation (fosc).

(Note 1)   **When stable oscillation is achieved with an external oscillator, no warm-up is required.**

(Note 2)   **The warm-up timer operates according to the oscillation clock, and it may contain errors if there is any fluctuation in the oscillation frequency. Therefore, the warm-up time should be taken as approximate time.**

(Note 3)   **When switching the system clock, ensure that the switching has been completed by reading the CGSYSCR<SYSCKFLG>.**

(Note 4)   **After setting the warm-up time to CGOSCCR<WUPT>, wait until the new value takes effect before executing the WFI instruction to transition to the standby mode.**

3.4.3.5 System Clock

The TMPM322F10 offers two selectable system clocks: low-speed or high-speed. The high-speed clock is dividable.

- Input frequency from X1and X2 ： 12MHz

- Allows for oscillator connection or external clock input.

- Clock gear : 1/1, 1/2, 1/4, 1/8   (after Reset 1/1)

Table3.4.2 Range of High –frequency in quardple PLL（Unit :MHz）

Input Freq from X1 and X2	Min Operating Freq	Max Operating Freq	after reset (PLL=OFF, CG=1/1)	Clock gear(CG) PLL=ON				Clock gear(CG) PLL=OFF			
				1/1	1/2	1/4	1/8	1/1	1/2	1/4	1/8
12	1.5MHz	48MHz	12	48	24	12	6	12	6	3	1.5

Use to high precision resonator. (ex. Crystal unit)

- Input frequency from XT1 and XT2

Table 3.4.3 Range of Low Frequency

Input Frequency Range	Maximum Operating Frequency	Minimum Operating Frequency
30 ~ 34(kHz)	34 kHz	30 kHz

(Note 1)	**Switching of clock gear is executed when a value is written to the CGSYSCR<GEAR> register. The actual switching takes place after a slight delay.**

3.4.3.6 Prescaler Clock Control

Each peripheral function (TMRB0-F, SIO0-11 and SBI0-4) has a prescaler for dividing a clock. As the clock φT0 to be input to each prescaler, the "fperiph" clock specified in the CGSYSCR<FPSEL> can be divided according to the setting in the CGSYSCR<PRCK>. After the controller is reset, fperiph/1 is selected as φT0.

(Note) **To use the clock gear, ensure that you make the time setting such that prescaler output φTn from each peripheral function is slower than fsys (φTn < fsys). Do not switch the clock gear while the timer counter or other peripheral function is operating.**

3.4.3.7 System Clock Pin Output Function

The TMPM322F10 enables to output the system clock from a pin. The PE7/SCOUT pin can output the low speed clock fs, the system clock fsys and fsys/2, and the prescaler input clock for peripheral I/O φT0. By setting the port K registers, the PECR<PE7C> and PEFR3<PE7F3> to "1", the PE7/SCOUT pin (pin number 88) becomes the SCOUT output pin. The output clock is selected by setting the CGSYSCR<SCOSEL>.

Table 3.4.4 shows the pin states in each mode when the SCOUT pin is set to the SCOUT output.

Table 3.4.4 Scout Output State in Each Mode

SCOUT selection CGSYSCR \ Mode	NORMAL	SLOW	IDLE2,1	SLEEP	STOP/BACKUP
<SCOSEL> = "00"	Output the fs clock.				
<SCOSEL> = "01"	Output the fsys/2 clock.				
<SCOSEL> = "10"	Output the fsys clock.				
<SCOSEL> = "11"	Output the ΦT0 clock.			Fixed to "0" or "1".	

3.4.4 Modes and Mode Transitions

3.4.4.1 Mode Transitions

The NORMAL mode and the SLOW mode use the high-speed and low-speed clocks for system clock respectively.

The IDLE2/1, SLEEP and STOP modes can be used as the low power consumption mode that enables to reduce power consumption by halting processor core operation.

When the low-speed clock is not used, the SLOW and SLEEP modes cannot be used.

This product also has a backup feature to drastically reduce power consumption by executing other functions and cutting the main power supply

.

Fig. 3.4.3 shows a mode transition diagram

For a description of sleep-on-exit, refer to "Cortex-M3 Technical Reference Manual".
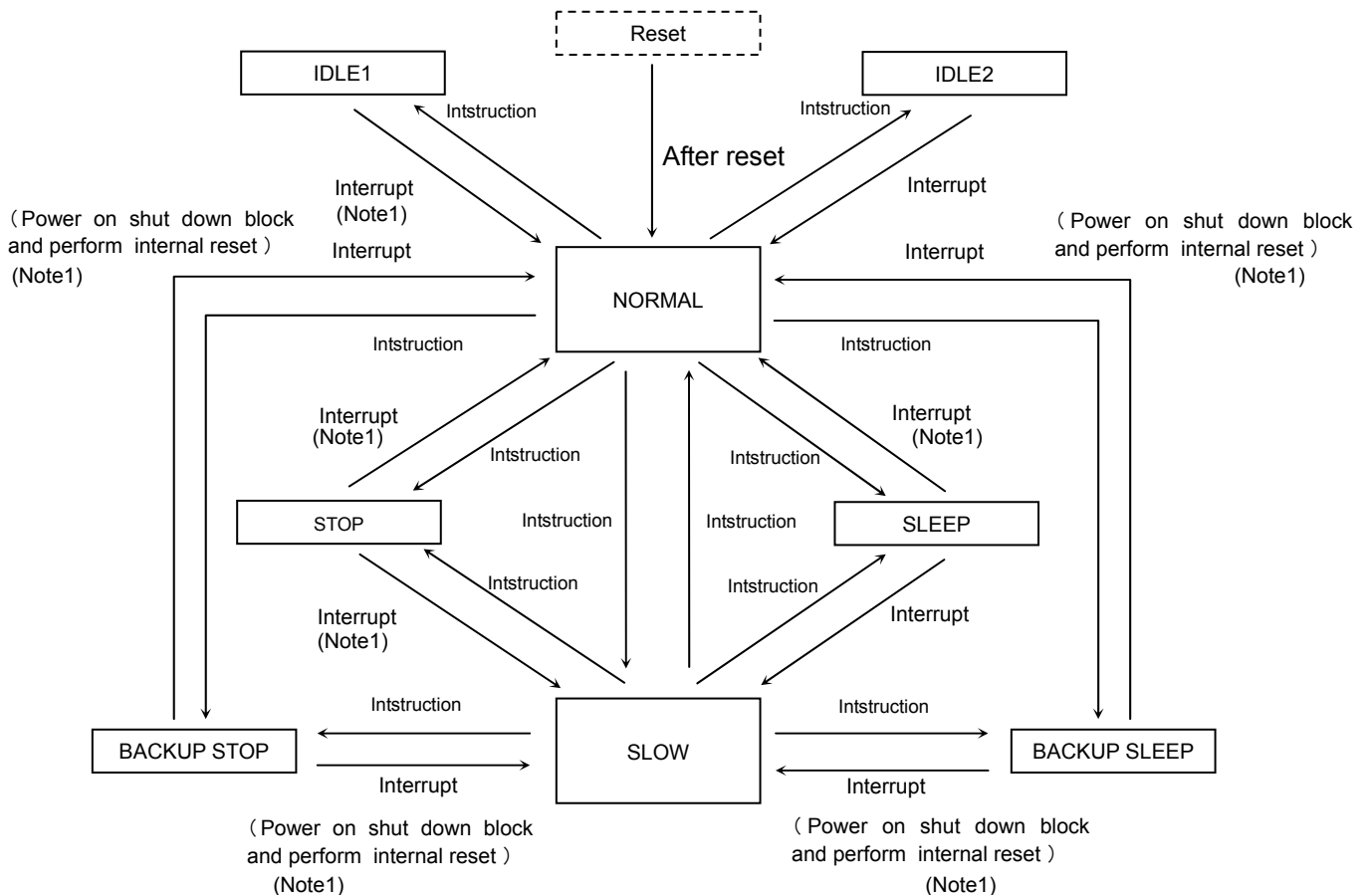
Fig. 3.4.3 Mode Transition Diagram

(Note 1)	**When returning to NORMAL mode, warm-up time is necessary.** **The warming up time must be configured from the modes executed before BACKUP mode; NORMAL or SLOW. Refer to the section 3.4.6.8 Warming up for more information about the warming up time.**
(Note 2)	**When the low-speed clock is not used, SLOW and SLEEP modes cannot be used.**
(Note 3)	**Transition from SLOW mode to IDLE1 or IDLE2 mode is not possible.**

3.4.5 Operation Mode

Two operation modes, NORMAL and SLOW, are available. The features of each mode are described below.

3.4.5.1 NORMAL Mode

This mode is to operate the CPU core and the peripheral hardware by using the high-speed clock. It is shifted to the NORMAL mode after reset. The low-speed clock can also be used.

3.4.5.2 SLOW Mode

This mode is to operate the CPU core and the peripheral hardware by using the low-speed clock with high-speed clock stopped.
The SLOW mode reduces power consumption compared to the NORMAL mode.
This mode allows only the following peripheral functions to operate: I/O ports,Timer(TMRB), real-time clock (RTC), remote control signal preprocessor (RMC) and Key on Wakeup function (KWUP).

(Note) **Be sure to stop peripheral functions except for the CPU, RTC, I/O ports, TMRB, RMC and KWUP before switching to the SLOW mode.**

3.4.6 Low Power Consumption Modes

The TMPM322F10 has four low power consumption modes: IDLE2, IDLE1, SLEEP and STOP. To shift to the low power consumption mode, specify the mode in the system control register CGSTBYCR<STBY> and execute the WFI (Wait For Interrupt) instruction. In this case, execute reset or generate the interrupt to release the mode. Releasing by the interrupt requires settings in advance. See " Interrpt " for details.

(Note 1)	**Transition to the low power consumption mode by executing the WFE (Wait For Event) instruction is prohibited as the TMPM322F10 does not offer any event for releasing the low power consumption mode.**
(Note 2)	**The TMPM322F10 does not support the low power consumption mode configured with the SLEEPDEEP bit in the Cortex-M3 core. Setting the SLEEPDEEP bit of the system control register is prohibited.**

The features of each mode are described as follows.

3.4.6.1 IDLE Mode (IDLE2, IDLE1)

The CPU is stopped in this mode.

Each peripheral function has one bit in its control register for enabling or disabling operation in the IDLE mode. When the IDLE mode is entered, peripheral functions for which operation in the IDLE mode is disabled stop operation and hold the state at that time.

The following peripheral functions can be enabled or disabled in the IDLE mode. For setting details, see the chapter on each peripheral function.

- 16-bit timer/event counter (TMRB)
- Serial channel (SIO)
- Serial bus interface (SBI)
- Remote control signal preprocessor (RMC)
- AD converter (ADC)
- Watchdog timer (WDT)

3.4.6.1.1 IDLE2 Mode

In the IDLE2 mode, the CPU and SSP are stopped and other peripheral functions remain active. The operating frequency range is equivalent to that in the NORMAL mode, enabling power reduction compared to the NORMAL mode without affecting the performance of the peripheral functions.

3.4.6.1.2   IDLE1 Mode

In the IDLE1 mode, power is supplied to fewer components compared to the IDLE2 mode to achieve greater power reduction. See Table 3.4.8 for the peripheral functions that can be used in the IDLE1 mode. The operating frequency (fsys) should be set to 1.5 MHz (fosc = 12 MHz, PLL disabled, and clock gear =1/8).

3.4.6.2 SLEEP Mode

The internal low-speed oscillator, real time clock and RMC operate. By releasing the SLEEP mode, the device returns to the preceding mode of the SLEEP mode and starts operation.

3.4.6.3 STOP Mode

All the internal circuits including the internal oscillator are brought to a stop.

By releasing the STOP mode, the device returns to the preceding mode of the STOP mode and starts operation.

The STOP mode enables to select the pin status by setting the CGSTBYCR<DRVE>. Table 3.4.5 shows the pin status in the STOP mode.

Table 3.4.5 Pin States in STOP Mode(1/2)

Pin Name	Input/Output	<DRVE>=0	<DRVE>=1
PA0 to PA7	Input mode	−	○
	Output mode	−	○
	D0 to D7	−(Note1)	−(Note1)
PB0 to PB7	Input mode	−	○
	Output mode	−	○
	D8 to D15	−(Note1)	−(Note1)
PC0 to PC7	Input mode, RXD8 to 9, SCLK8 to 9, CTS8n to 9n	−	○
	Output mode, TXD8 to 9, SCLK8 to 9	−	○
	A1 to A8	○	○
PD0 to PD7	Input mode, RXD10 to 11, SCLK10 to 11, CTS10n to 11n	−	○
	INTB	○	○
	Output mode, TXD10 to 11, SCLK10 to 11	−	○
	A9 to A16	○	○
PE0 to PE7	Input mode, TB5IN0 to 1, TB6IN0 to 1, RXD0, SCLK0, CTS0n	−	○
	INT5	○	○
	Output mode, TXD0, SCLK0, SCOUT	−	○
	A17 to A23	○	○
PF0 to PF4	Input mode	−	○
	Output mode	−	○
	TRACECLK, TRACEDATA0 to 3	○	○
PG0 to PG7	Input mode, SDA1 to 2, SCL1 to 2, SCK1 to 2, TB7IN0 to 1, TB9IN0 to 1, USB0Cn	−	○
	INT6 to 7	○	○
	Output mode, SDA1 to 2, SCL1 to 2, SCK1 to 2, USBPON, WDTOUTn	−	○
	CS0n, CS1n, CS3n	○	○
PH0 to PH7	Input mode, SDA3 to 4, SCL3 to 4, SCK3 to 4, TBAIN0 to 1, TBBIN0 to 1, TBDIN0 to 1, TBEIN0 to 1	−	○
	INTC to D	○	○
	Output mode, SDA3 to 4, SCL3 to 4, SCK3 to 4	−	○
PI0 to PI1	Input mode	−	○
	Output mode	−	○
PJ0 to PJ7	Input mode, ADTRGn	−	○
	AN0 to 7	−	−
	KWUP0 to 3	○	○
PK0 to PK7	Input mode	−	○
	AN8 to 15	−	−
PL0 to PL7	Input mode, SDA0, SCL0, SCK0, RXD1, SCLK1, CTS1n	−	○
	INT0 to 1	○	○
	Output mode, SDA0, SCL0, SCK0, TXD1, SCLK1, TB0 to 7OUT	−	○

Table 3.4.5 Pin States in STOP Mode(2/2)

Pin Name	Input/Output	⟨DRVE⟩=0	⟨DRVE⟩=1
PM0 to PM7	Input mode, RXD2 to 3, SCLK2 to 3, CTS2n to 3n, TB1IN0 to 1	−	−
	INT2 to 3	○	○
	Output mode, TXD2 to 3, SCLK2 to 3, ALARMn, TB3OUT	−	○
PN0 to PN7	Input mode, RXD4 to 5, SCLK4 to 5, CTS4n to 5n, TB2IN0 to 1, TBFIN0 to 1, RMIN0 to 1	−	○
	INT4, INT8	○	○
	Output mode, TXD4 to 5, SCLK4 to 5	−	○
PO0 to PO7	Input mode, RXD6 to 7, SCLK6 to 7	−	○
	INT9 to A	○	○
	Output mode, TXD6 to 7, SCLK6 to 7, TB8 to FOUT	−	○
PP0 to PP6	Input mode, SPDI, SPCLK	−	○
	Output mode, SPDO, SPCLK, SPFSS	−	○
	CS2n, BLS0n to 1n, WEn, OEn	○	○
SWDIO		Input(PU) orOutput	Input(PU) orOutput
SWCLK		Input(PD)	Input(PD)
D+, D−		Suspend (Note2)	Suspend (Note2)
NMIn		Input(PU)	Input(PU)
TEST1～2		−	−
RESETn		Input	Input
MODE		Input	Input
XT1		−	−
XT2		H level	H level
X1		−	−
X2		H level	H level

○   : Input mode /function input follows the setting of the port input control enable register (PxIE<n>).
               And, Output mode /function output follows the setting of the port control register (PxCR<n>).
−   : Input or output disabled.
Input   :The input gate is active.To prevent the input pin from floating, fix the input voltage to
               the level.
Output   :The pin is in the output state.
Suspend   : Suspend status (Input ON / Output OFF)
H level   :The pin is "H" level output.

(Note1)	: Complete the access to the external memory before change to the STOP mode.
(Note2)	: Suspend the USB port before change to the STOP mode.

3.4.6.4 BACKUP Mode

   BACKUP mode realizes the lowest power consumption by cutting off the internal power regulator. About more details, refer to the 3.20 BACKUP module.

3.4.6.5 Low Power Consumption Mode Setting

   The low power consumption mode is specified by the setting of the standby control register CGSTBYCR<STBY>.

   Table 3.4.6 shows the mode setting in the CGSTBYCR<STBY>.

Table 3.4.6 Low power consumption mode setting

Mode	CGSTBYCR <STBY>
Reserved	000
STOP	001
SLEEP	010
IDLE2	011
Reserved	100
BACKUP STOP	101
BACKUP SLEEP	110
IDLE1	111

(Note)   Do not use the settings indicated as "reserved".

3.4.6.6 Operational State in Each Mode

Table 3.4.7 show the operational state in each mode.

Table 3.4.7 Operational State in Each Mode

Block	NORMAL	SLOW	IDLE2	IDLE1 (Note1)	SLEEP	STOP	BACKUP SLEEP (Note2)	BACKUP STOP (Note2)
Processor core	○	○	−	−	−	−	×	×
DMAC	○	−	○	−	−	−	×	×
INTC	○	○	○	○	○	○	×	×
SMC	○	−	○	−	−	−	×	×
IO Port	○	○	○ (Note6)	○ (Note6)	○ (Note6)	○ (Note2)	△ (Note7)	△ (Note2, 7)
ADC	○	▲ (Note4)	△ (Note4)	▲ (Note4)	− (Note4)	− (Note4)	×	×
SIO	○	▲	△	Max 2ch	−	−	×	×
I2C	○	▲	△	Max 1ch	−	−	×	×
TMRB	○	○	△	Max 2ch	−	−	×	×
WDT	○	▲	△	△	−	−	×	×
SSP	○	▲	−	−	−	−	×	×
USB-HOST	○	▲ (Note5)	− (Note5)	− (Note5)	− (Note5)	− (Note5)	× (Note5)	× (Note5)
KWUP	○	○	○ (Note3)	○ (Note3)	○	○ (Note3)	△	△ (Note3)
RMC	○	○	△	△	○	−	△	−
RTC	○	○	△	△	○	−	△	−
CG	○	○	○	○	○	○	○	○
PLL	○	□	△	▲	−	−	×	×
High-speed oscillator (fc)	○	□	○	○	−	−	−	−
Low-speed oscillator (fs)	○	○	△	△	○	−	△	−
Main RAM	○	○	○	○	○	○	×	×
Backup RAM (Note2)	○	○	○	○	○	○	○	○

○ : Operating in corresponding mode

− : When shifting to corresponding mode, clock supply is automatically stopped.(Note8)

△ : ON/OFF selectable by sofwware in corresponding mode.

▲ : Before shifting to corresponding mode, must be stopped by software.

× : When shifting to corresponding mode, power supply is automatically stopped.

□ : After shifting to corresponding mode, must be stopped by software.

(Note1) When using the IDLE1 mode, disable the PLL, DMAC, ADC and SSP, limit the number of channels used in the TMRB, SIO and I2C, and make sure that the operating frequency does not exceed fsys=1.5MHz (fosc=12MHz, PLL disabled, clock gear =1/8).

(Note2) The state depends on the CGSYSCR<DRVE> bit.

(Note3) When the low-speed oscillator is stopped, or in a mode in which the low-speed oscillator is automatically stopped, only static pull-ups are enabled.

(Note4) Before shifting to corresponding mode, set ADMOD1(VREFON) to "0"

(Note5) Before shifting to corresponding mode, set SUSPEND state

(Note6) Port keep the level at the time of shifting standby mode.

(Note7) The state depends on the CGSYSCR<PTKEEP> bit.

(Note8) When shifting to corresponding mode, clock supply is automatically stopped. Shift the mode after confirming the operation of each block is completed if needed.

3.4.6.7 Releasing the Low Power Consumption Mode

The low power consumption mode can be released by an interrupt request, NMI or reset. The release source that can be used is determined by the low power consumption mode selected.

Details are shown in Table 3.4.8.

Table 3.4.8 Release Source in Each Mode

Low power consumption mode			IDLE2	IDLE1 (Note1)	SLEEP	STOP	BACKUP SLEEP (Note2)	BACKUP STOP (Note2)
Release source	Interrupt	INT0～4, 8 (Note3)	○	○	○	○	○	○
		INT5～7, 9～D (Note3)	○	○	○	○	×	×
		INTRTC	○	○	○	×	○	×
		INTTB0～F	○	○	×	×	×	×
		INTCAP10～30,50～70, 90～B0, D0～F0	○	○	×	×	×	×
		INTCAP11～31,51～71, 91～B1, D1～F1	○	○	×	×	×	×
		INTRX0～11, INTTX0～11	○	○	×	×	×	×
		INTSBI0～4	○	○	×	×	×	×
		INTRMCRX0, 1	○	○	○	×	○	×
		INTAD/INTADHP/INTADM	○	×	×	×	×	×
		INTKWUP	○	○	○	○	○	○
	NMIn (INTWDTn)		○	○	×	×	×	×
	NMIn （NMI pin）		○	○	○	○	○	○
	RESETn (RESET pin)		○	○	○	○	○	○

○：   After releasing, start to interrupt function. (RESET initialize LSI)
×：   Unavailable

(Note 1)	**For the warm-up time required to return from each mode, see "3.4.6.8 Warm-up".**
(Note 2)	**After releasing BACKUP mode, the reset is released. But, backup module does not initialize.**
(Note 3)	**To release the low power consumption mode by using the level mode interrupt, keep the level until the interrupt handling is started. Changing the level before then will prevent the interrupt handling from starting properly.**
(Note 4)	**For shifting to the low power consumption mode, set the CPU to prohibit all the interrupts other than the release source. If not, releasing may be executed by an unspecified interrupt.**

(Note) **When entering the IDLE1 mode from the NORMAL mode, the warm-up time must be set to 100 μsec or longer. If the warm-up time is set to less than 100 μsec, sufficient time cannot be secured for the internal system to resume operation properly when returning to the NORMAL mode from the IDLE1 mode.**

● Release by interrupt request

To release the low power consumption mode by an interrupt, the CPU must be set in advance to detect the interrupt. In addition to the setting in the CPU, the clock generator must be set to detect the interrupt to be used to release the SLEEP and STOP modes.

● Release by NMIn

There are two kinds of NMIn sources: WDT interrupt (INTWDTn) and NMIn pin. INTWDTn can only be used in the IDLE mode. The NMIn pin can be used to release all the lower power consumption modes.

● Release by reset

Any low power consumption modes can be released by reset from the RESET pin. After that, the mode switches to NORMAL and all the registers are initialized as is the case with normal reset.

Note that returning to NORMAL mode by reset does not induce the automatic warm-up. Keep the reset signal valid until the oscillator operation becomes stable.

Refer to "Interrupts" for details.

3.4.6.8 Warm-up

Mode transition may require the warm-up so that the internal oscillator provides stable oscillation.

In the mode transition from STOP to NORMAL/ SLOW or from IDEL1/SLEEP to NORMAL, the warm-up counter is activated automatically. And then the system clock output is started after the elapse of configured warm-up time. It is necessary to select the oscillator to be used for warm-up in the CGOCCR<WUPSEL> and to set the warm-up time in the CGOSCCR<WUPT><WUPTL> before executing the instruction to enter the STOP/ IDLE1/ SLEEP mode.

(Note)	In STOP/ SLEEP modes, the PLL is disabled. When returning from these modes, configure the warm-up time in consideration of the stability time of the PLL and the internal oscillator. It takes approx. 200μs for the PLL to be stabilized.

In the transition from NORMAL to SLOW/ SLEEP, the warm-up is required so that the internal oscillator to stabilize if the low-speed clock is disabled. Enable the low-speed clock and then activate the warm-up by software.

In the transition from SLOW to NORMAL when the high-speed clock is disabled, enable the high-speed clock and then activate the warm-up.

Table 3.4.9 shows whether the warm-up setting of each mode transition is required or not.

Table 3. 4. 9 Warm-up setting in mode transition

Mode transition	Warm-up setting
NORMAL→IDLE2,1	Not required
NORMAL→SLEEP	(Note 1)
NORMAL→SLOW	(Note 1)
NORMAL→STOP	Not required
NORMAL →BACKUP SLEEP	(Note1)
NORMAL →BACKUP STOP	Not requied
SLOW →NORMAL	(Note 2)
SLOW →SLEEP	Not required
SLOW →STOP	Not required
IDLE2 →NORMAL	Not required
IDLE1 →NORMAL	Over 100μs (Note 3)
SLEEP →NORMAL	Auto-warm-up
SLEEP →SLOW	Not required
STOP →NORMAL	Auto-warm-up (Note 4)
STOP →SLOW	Auto-warm-up (Note 4)
BACKUP SLEEP→NORMAL	Auto-warm-up (Note 3) High-Speed OSC : Over 500μs Low-Speed OSC : Over 2.5ms
BACKUP STOP →NORMAL	Auto-warm-up (Note 3) High-Speed OSC : Over 500μs

(Note 1)	If the low-speed clock is disabled, enable the low-speed clock and then activate the warm-up by software.
(Note 2)	If the high-speed clock is disabled, enable the high-speed clock and then activate the warm-up by software.
(Note 3)	The warm-up time must always be set as shown in the table above. Do not use any other settings.
(Note 4)	Returning to NORMAL mode by reset does not induce the automatic warm-up. Keep the reset signal valid until the oscillator operation becomes stable.

3.4.6.9 Clock Operations in Mode Transition

The clock operations in mode transition are described in the following sections 3.4.6.9.1 to 3.4.6.9.4.

3.4.6.9.1   Transition of operation modes: NORMAL→STOP→NORMAL

When returning to NORMAL mode from STOP mode, the warm-up is activated automatically. It is necessary to set the warm-up time before entering the STOP mode.

Returning to NORMAL mode by reset does not induce the automatic warm-up. Keep the reset signal asserted until the oscillator operation becomes stable.
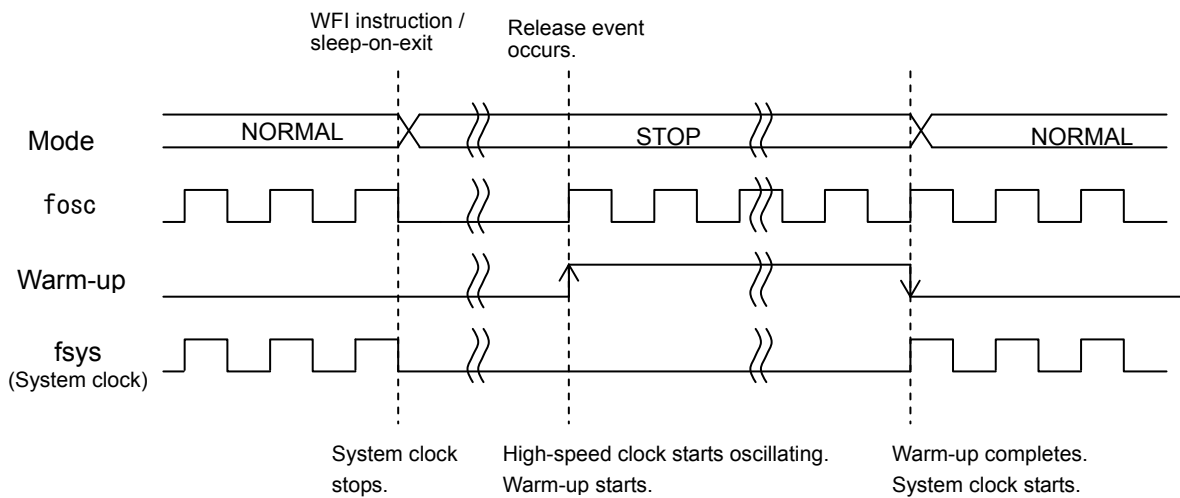
3.4.6.9.2   Transition of operation modes: NORMAL→SLEEP→NORMAL

When returning to NORMAL mode from SLEEP mode, the warm-up is activated automatically. It is necessary to set the warm-up time before entering the SLEEP mode.

Returning to NORMAL mode by reset does not induce the automatic warm-up. Keep the reset signal asserted until the oscillator operation becomes stable.
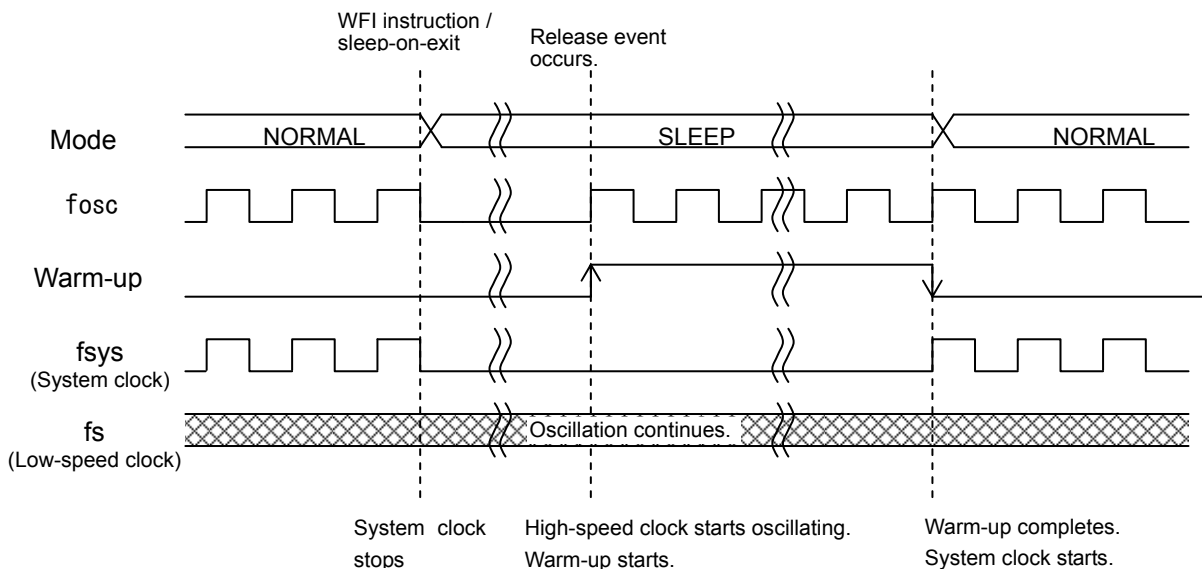
3.4.6.9.3   Transition of operation modes: SLOW→STOP→SLOW

The warm-up is activated automatically. It is necessary to set the warm-up time before entering the STOP mode.
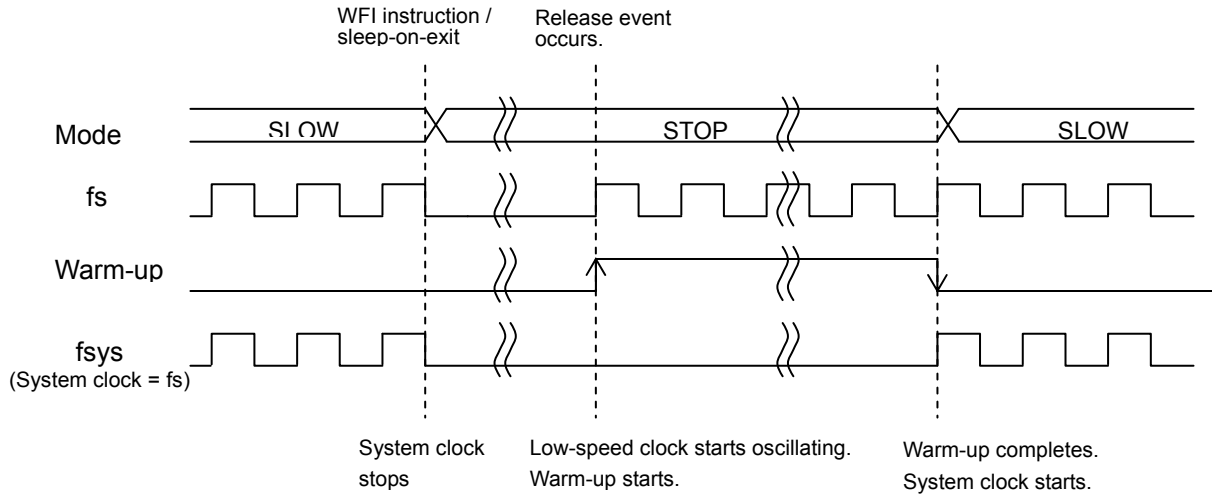
3.4.6.9.4   Transition of operation modes: SLOW→SLEEP→SLOW

The low-speed clock continues oscillation in the SLEEP mode. There is no need to make a warm-up setting.
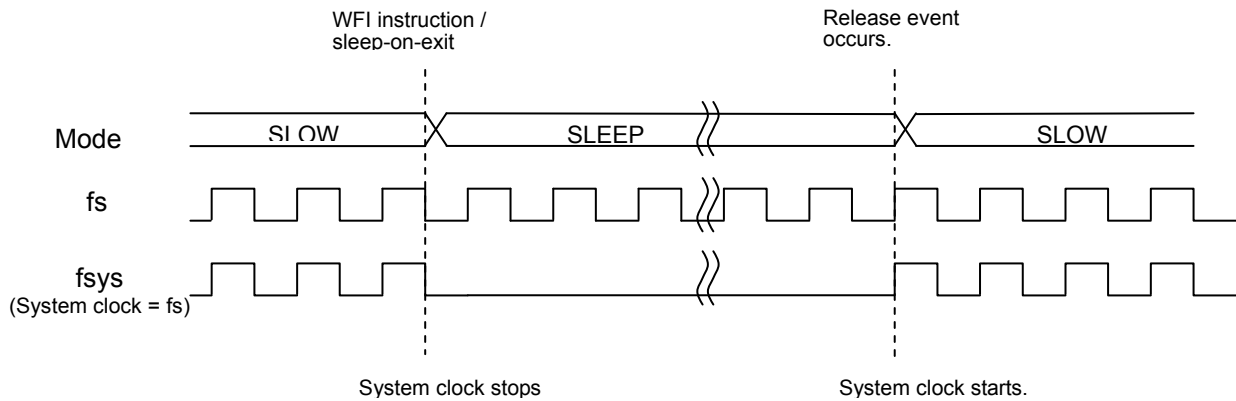
3.5 Exceptions

This chapter describes features, types and handling of exceptions.

Exceptions have close relation to the CPU core. Refer to "Cortex-M3 Technical Reference Manual" if needed.

3.5.1 Overview

An exception causes the CPU to stop the currently executing process and handle another process.

There are two types of exceptions: those that are generated when some error condition occurs or when an instruction to generate an exception is executed; and those that are generated by hardware, such as an interrupt request signal from an external pin or peripheral function.

All exceptions are handled by the Nested Vectored Interrupt Controller (NVIC) in the CPU according to the respective priority levels. When an exception occurs, the CPU stores the current state to the stack and branches to the corresponding interrupt service routine (ISR). Upon completion of the ISR, the information stored to the stack is automatically restored.

3.5.2 Exception Types

The following types of exceptions exist in the Cortex-M3.
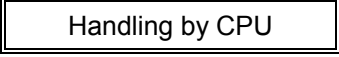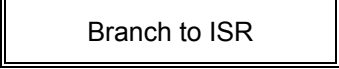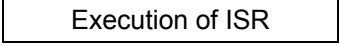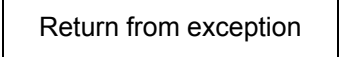
For detailed descriptions on each exception, refer to "Cortex-M3 Technical Reference Manual".

- Reset
- Non-Maskable Interrupt(NMI)
- Hard Fault
- Memory Managemet
- Bus Fault
- Usage Fault
- SVCall (Supervisor Call)
- Debug Monitor
- PendSV
- SysTick
- External Interrupt

3.5.3 Handling Flowchart

The following shows how an exception/interrupt is handled.

[▯] indicates hardware handling. [▯] Indicates software handling.

Each step is described later in this chapter.

Processing	Description	See
Detection by CG/CPU	The CG/CPU detects the exception request.	Section 3.5.3.1
Handling by CPU	The CPU handles the exception request.	Section 3.5.3.2
Branch to ISR	The CPU branches to the corresponding interrupt service routine (ISR).	
Execution of ISR	Necessary processing is executed.	Section 3.5.3.3
Return from exception	The CPU branches to another ISR or returns to the previous program.	Section 3.5.3.4

3.5.3.1   Exceptio Request and Detection

(1) Exception occurrence

Exception sources include instruction execution by the CPU, memory accesses, and interrupt requests from external interrupt pins or peripheral functions.

An exception occurs when the CPU executes an instruction that causes an exception or when an error condition occurs during instruction execution.

An exception also occurs by an instruction fetch from the Execute Never region or an access violation to the Fault region.

An interrupt is generated from an external interrupt pin or peripheral function. For interrupts that are used for releasing a standby mode, relevant settings must be made in the clock generator. For details, refer to "3.5.5 Interrupts".

(2) Exception detection

If multiple exceptions occur simultaneously, the CPU takes the exception with the highest priority.

Table 3.5.1 shows the priority of exceptions. "Configurable" means that you can assign a priority level to that exception. Memory Management, Bus Fault and Usage Fault exceptions can be enabled or disabled. If a disabled exception occurs, it is handled as Hard Fault.

Table 3.5.1 Exception Types and Priority

No.	Exception type	Priority	Description
1	Reset	-3 (highest)	Reset pin, WDT or SYSRETREQ
2	Non-Maskable Interrupt	-2	NMI pin or WDT
3	Hard Fault	-1	Fault that cannot activate because a higher-priority fault is being handled or it is disabled
4	Memory Management	Configurable	Exception from the Memory Protection Unit (MPU) (Note 1) Instruction fetch from the Execute Never (XN) region
5	Bus Fault	Configurable	Access violation to the Hard Fault region of the memory map
6	Usage Fault	Configurable	Undefined instruction execution or other faults related to instruction execution
7-10	Reserved		
11	SVCall	Configurable	System service call with SVC instruction
12	Debug Monitor	Configurable	Debug monitor when the core is not halting
13	Reserved		
14	PendSV	Configurable	Pendable system service request
15	SysTick	Configurable	Notification from system timer
16-	External Interrupt	Configurable	External interrupt pin or peripheral function (Note 2)

(Note 1)   This product does not contain the MPU.

(Note 2)   External interrupts have different sources and numbers in each product. For details, see "List of Interrupt Sources".

(3) Priority setting

Use the Interrupt Priority Registers to assign a priority to each of the external interrupts. The priority of other exceptions can be set in the System Handler Priority Registers.

The priority registers are configurable, allowing the number of bits for setting priority levels to vary between three to eight bits. Therefore, the range of priority levels that can be assigned vary with each product.

You can assign a priority level from 0 to 255 when using eight bits. Priority level 0 is the highest priority level.

If you assign the same priority level to multiple exceptions, the lowest-numbered exception has the highest priority.

(Note)	**In this product, three bits are used for assigning a priority level in the Interrupt Priority Registers and System Handler Priority Registers.**

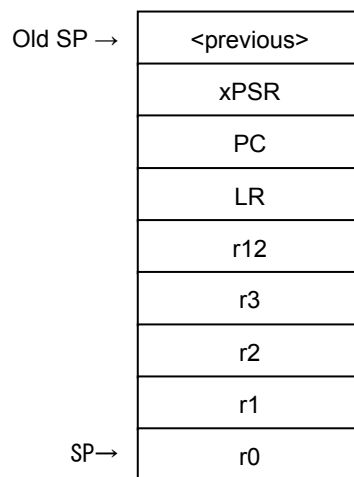### 3.5.3.2   Exceptio Handling and Branch to the Interrupt Service Routine (Pre-emption)

When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called "pre-emption".

(1) Stacking

When the CPU detects an exception, it pushes the contents of the following eight registers to the stack in the following order:

- Program Counter (PC)
- Program Status Register (xPSR)
- r0 – r3
- r12
- Link Register (LR)

The SP is decremented by eight words by the completion of the stack push. The following shows the state of the stack after the register contents have been pushed.

```
Old SP →  ┌──────────────┐
          │  <previous>  │
          ├──────────────┤
          │    xPSR      │
          ├──────────────┤
          │     PC       │
          ├──────────────┤
          │     LR       │
          ├──────────────┤
          │    r12       │
          ├──────────────┤
          │     r3       │
          ├──────────────┤
          │     r2       │
          ├──────────────┤
          │     r1       │
          ├──────────────┤
     SP→  │     r0       │
          └──────────────┘
```

(2) Fetching an ISR

At the same time as pushing the register contents to the stack, the CPU executes an instruction to fetch an ISR.

Prepare a vector table containing the top addresses of ISRs for each exception. After reset, the vector table is located at address 0x0000_0000 in the code space. By setting the Vector Table Offset Register, you can place the vector table at any address in the code or SRAM space.

The vector table should also contain the initial value of the main stack.

(3) Late-arriving

If the CPU detects a higher priority exception before executing the ISR for a previous exception, the CPU handles the higher priority exception first. This is called "late-arriving".

A late-arriving exception causes the CPU to fetch a new vector address for branching to the corresponding ISR, but the CPU does not newly push the register contens to the stack.

(4) Vector table

The vector table is configured as shown below.

You must always set the first four words (stack top address, reset ISR address, NMI ISR address, and Hard Fault ISR address). Set ISR addresses for other exceptions if necessary.

Offset	Exception	Contents	Setting
0x00	Reset	Initial value of the main stack	Required
0x04	Reset	ISR address	Required
0x08	Non-Maskable Interrupt	ISR address	Required
0x0C	Hard Fault	ISR address	Required
0x10	Memory Managemet		Optional
0x14	Bus Fault	ISR address	Optional
0x18	Usage Fault	ISR address	Optional
0x1C ∼ 0x28	Reserved		
0x2C	SVCall	ISR address	Optional
0x30	Debug Monitor	ISR address	Optional
0x34	Reserved		
0x38	PendSV	ISR address	Optional
0x3C	SysTick	ISR address	Optional
0x40	External Interrupt	ISR address	Optional

3.5.3.3   Excuting an ISR

An ISR performs necessary processing for the corresponding exception. ISRs must be prepared by the user.

An ISR may need to include code for clearing the interrupt request so that the same interrupt will not occur again upon return to normal program execution.

For details about interrupt handling, see "3.5.5 Interrupt".

If a higher priority exception occurs during ISR execution for the current exception, the CPU abandons the currently executing ISR and services the newly detected exception.

3.5.3.4   Exception exit

(1)   Execution after returning from an ISR

When returning from an ISR, the CPU takes one of the following actions:

- Tail-chaining

If a pending exception exists and there are no stacked exceptions or the pending exception has higher priority than all stacked exceptions, the CPU returns to the ISR of the pending exception.

In this case, the CPU skips the pop of eight registers and push of eight registers when exiting one ISR and entering another. This is called "tail-chaining".

- Returning to the last stacked ISR

If there are no pending exceptions or if the highest priority stacked exception is of higher priority than the highest priority pending exception, the CPU returns to the last stacked ISR.

- Returning to the previous program

When returning from an ISR, the CPU performs the following operations:

(2)   Exceptio exit sequence

When returning from an ISR, the CPU performs the following operations:

- Pop eight registers

Pops the eight registers (PC, xPSR, r0 to r3, r12 and LR) from the stack and adjust the SP.

- Load current active interrupt number

Loads the current active interrupt number from the stacked xPSR. The CPU uses this to track which interrupt to return to.

- Select SP

If returning to an exception (Handler Mode), SP is SP_main. If returning to Thread Mode, SP can be SP_main or SP_process.

3.5.4   Exceptions

3.5.4.1   Reset Exceptions

Reset exceptions are generated from the following three sources.

Use the Reset Flag (CGRSTFLG) Register of the Clock Generator to identify the source of a reset.

・External reset pin

A reset exception occurs when an external reset pin changes from "L" to "H".

・Reset exception by WDT

The watchdog timer (WDT) has a reset generating feature. For details, see the chapter on the WDT.

・Reset exception by SYSRESETREQ

A reset can be generated by setting the SYSRESETREQ bit in the NVIC's Application Interrupt and Reset Control Register.

3.5.4.2   Non-Maskable Interrupt (NMIn)

Non-maskable interrupts are generated from the following two sources.

Use the NMI Flag (CGNMIFLG) Register of the clock generator to identify the source of a non-maskable interrupt.

・External NMI pin

A non-maskable interrupt is generated when an external NMI pin changes from "H" to "L".

・Non-maskable interrupt by WDT

The watchdog timer (WDT) has a non-maskable interrupt generating feature. For details, see the chapter on the WDT.

3.5.4.3   SysTick

SysTick provides interrupt features using the CPU's system timer.

When you set a value in the SysTick Reload Value Register and enable the SysTick features in the SysTick Control and Status Register, the counter loads with the value set in the Reload Value Register and begins counting down. When the counter reaches "0", a SysTick exception occurs. You may pend exceptions and use a flag to know when the timer reaches "0".

The SysTick Calibration Value Register holds a reload value for counting 10 ms with the system timer. The count clock frequency varies with each product, and so the value set in the SysTick Calibration Value Register also varies with each product.

(Note)	In this product, the system timer counts based on a clock obtained by dividing the clock input from the X1 pin by 32.
	The SysTick Calibration Value Register is set to 0x9C4, which provides 10 ms timing when the clock input from X1 is 8 MHz.

3.5.5   Interrupts

This chapter describes routes, sources and required settings of interrupts.

The CPU is notified of interrupt requests by the interrupt signal from each interrupt source. It sets priority on interrupts and handles an interrupt request with the highest priority.

Interrupt requests for clearing a standby mode are notified to the CPU via the clock generator. Therefore, appropriate settings must be made in the clock generator.

3.5.5.1   Interrupt Source

3.5.5.1.1   Interrupt Route

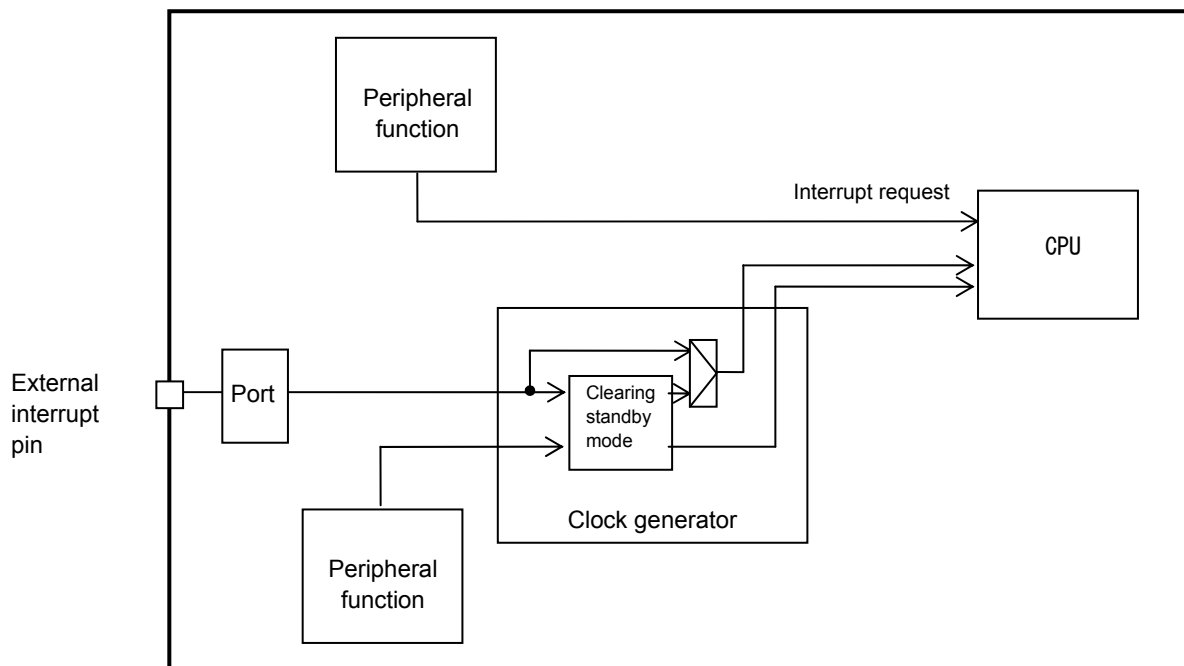Fig.3.5.1 shows an interrupt request route.

Fig.3.5.1   Interrupt Route

3.5.5.1.2   Generation

An interrupt request is generated from an external pin or peripheral function assigned as an interrupt source or by setting the NVIC's Interrupt Set-Pending Register.

・From external pin

Set the port control register so that the external pin can perform as an interrupt function pin.

・From peripheral function

Set the peripheral function to make it possible to output interrupt requests.

See the chapter of each peripheral function for details.

・By setting Interrupt Set-Pending Register (forced pending)

An interrupt request can be generated by setting the relevant bit of the Interrupt Set-Pending Register.

3.5.5.1.3   Transmission

An interrupt signal from an external pin or peripheral function is directly sent to the CPU unless it is used to clear a standby mode.

Interrupt requests from interrupt sources that can be used for clearing a standby mode are transmitted to the CPU via the clock generator. For these interrupt sources, appropriate settings must be made in the clock generator in advance. External interrupt sources not used for clearing a standby mode can be used without setting the clock generator.

3.5.5.1.4   List of Interrupt Sources

Table 3.5.2 shows the list of interrupt sources.

Table 3.5.2   List of Hardware Interrupt Sources (1/2)

No.	Interrupt Source		Active state (Clearing standby)	CG interrupt mode control register
0	INT0	Interrupt pin 0		CGIMCGA
1	INT1	Interrupt pin 1		
2	INT2	Interrupt pin 2		
3	INT3	Interrupt pin 3		
4	INT4	Interrupt pin 4		CGIMCGB
5	INT5	Interrupt pin 5		
6	INT6	Interrupt pin 6		
7	INT7	Interrupt pin 7		
8	INT8	Interrupt pin 8		CGIMCGC
9	INT9	Interrupt pin 9		
10	INTA	Interrupt pin A		
11	INTB	Interrupt pin B		
12	INTC	Interrupt pin C		CGIMCGD
13	INTD	Interrupt pin D		
14	-	Reserved		
15	-	Reserved		
16	INTRX0	Serial reception (channel.0)		
17	INTTX0	Serial transmission (channel.0)		
18	INTRX1	Serial reception (channel.1)		
19	INTTX1	Serial transmission (channel.1)		
20	INTRX2	Serial reception (channel.2)		
21	INTTX2	Serial transmission (channel.2)		
22	INTRX3	Serial reception (channel.3)		
23	INTTX3	Serial transmission (channel.3)		
24	INTRX4	Serial reception (channel.4)		
25	INTTX4	Serial transmission (channel.4)		
26	INTSBI0	Serial bus interface (channel.0)		
27	INTSBI1	Serial bus interface (channel.1)		
28	-	Reserved		CGIMCGE
29	-	Reserved	Rising edge	
30	INTRMCRX0	Remote control signal reception (channel.0)		
31	INTRMCRX1	Remote control signal reception (channel.1)		
32	INTRTC	Real time clock timer	Rising edge	CGIMCGF
33	INTKWUP	Key On Wakeup	High level	
34	INTSBI2	Serial bus interface (channel.2)		
35	INTSBI3	Serial bus interface (channel.3)		
36	INTSBI4	Serial bus interface (channel.4)		
37	INTADHP	Highest priority AD conversion complete interrupt		
38	INTADM0	AD conversion monitoring function interrupt 0		
39	INTADM1	AD conversion monitoring function interrupt 1		
40	INTTB0	16bit TMRB match detection 0		
41	INTTB1	16bit TMRB match detection 1		
42	INTTB2	16bit TMRB match detection 2		
43	INTTB3	16bit TMRB match detection 3		
44	INTTB4	16bit TMRB match detection 4		
45	INTTB5	16bit TMRB match detection 5		
46	INTTB6	16bit TMRB match detection 6		
47	INTTB7	16bit TMRB match detection 7		
48	INTTB8	16bit TMRB match detection 8		
49	INTTB9	16bit TMRB match detection 9		
50	INTTBA	16bit TMRB match detection A		
51	INTTBB	16bit TMRB match detection B		

Table 3.5.2 List of Hardware Interrupt Sources (2/2)

No.	Interrupt Source		Active state (Clearing standby)	CG interrupt mode control register
52	INTTBC	16bit TMRB match detection C		
53	INTTBD	16bit TMRB match detection D		
54	INTTBE	16bit TMRB match detection E		
55	INTTBF	16bit TMRB match detection F		
56	INTUSB	USB		
57	-	Reserved		
58	INTAD	A/D conversion completion		
59	INTSSP	Synchronous Serial Port		
60	INTRX5	Serial reception (channel.5)		
61	INTTX5	Serial transmission (channel.5)		
62	INTRX6	Serial reception (channel.6)		
63	INTTX6	Serial transmission (channel.6)		
64	INTRX7	Serial reception (channel.7)		
65	INTTX7	Serial transmission (channel.7)		
66	INTRX8	Serial reception (channel.8)		
67	INTTX8	Serial transmission (channel.8)		
68	INTRX9	Serial reception (channel.9)		
69	INTTX9	Serial transmission (channel.9)		
70	INTRX10	Serial reception (channel.10)		
71	INTTX10	Serial transmission (channel.10)		
72	INTRX11	Serial reception (channel.11)		
73	INTTX11	Serial transmission (channel.11)		
74	INTCAP10	16bit TMRB input capture 10		
75	INTCAP11	16bit TMRB input capture 11		
76	INTCAP20	16bit TMRB input capture 20		
77	INTCAP21	16bit TMRB input capture 21		
78	-	Reserved		
79	-	Reserved		
80	INTCAP50	16bit TMRB input capture 50		
81	INTCAP51	16bit TMRB input capture 51		
82	INTCAP60	16bit TMRB input capture 60		
83	INTCAP61	16bit TMRB input capture 61		
84	INTCAP70	16bit TMRB input capture 70		
85	INTCAP71	16bit TMRB input capture 71		
86	INTCAP90	16bit TMRB input capture 90		
87	INTCAP91	16bit TMRB input capture 91		
88	INTCAPA0	16bit TMRB input capture A0		
89	INTCAPA1	16bit TMRB input capture A1		
90	INTCAPB0	16bit TMRB input capture B0		
91	INTCAPB1	16bit TMRB input capture B1		
92	INTCAPD0	16bit TMRB input capture D0		
93	INTCAPD1	16bit TMRB input capture D1		
94	INTCAPE0	16bit TMRB input capture E0		
95	INTCAPE1	16bit TMRB input capture E1		
96	INTCAPF0	16bit TMRB input capture F0		
97	INTCAPF1	16bit TMRB input capture F1		
98	INTDMACERR	DMA transmission error interrupt		
99	INTDMACTC0	DMA transmission completion interrupt		

3.5.5.1.5   Active State

The active state indicates which change in signal of an interrupt source triggers an interrupt. The CPU detects an interrupt request when an interrupt signal changes from "L" to "H". Interrupt signals directly sent from peripheral functions to the CPU are configured to output "H" to indicate an interrupt request.

Interrupt requests from peripheral functions are set as rising-edge or falling-edge triggered. Interrupt requests from interrupt pins can be set as level-sensitive ("H" or "L") or edge-triggered (rising or falling).

If an interrupt source is used for clearing a standby mode, setting the relevant clock generator register is also required. Enable the CGIMCGx<INTxEN> bit and specify the active state in the CGIMCGx<EMCGx> bits. You must set the active state for interrupt requests from each peripheral function as shown in Table 3.5.2.

An interrupt request detected by the clock generator is notified to the CPU with a signal in "H" level.

Interrupt requests from interrupt pins can be used without setting the clock generator if they are not used for clearing a standby mode. However, an "H" pulse or "H"-level signal must be input so that the CPU can detect it as an interrupt request.
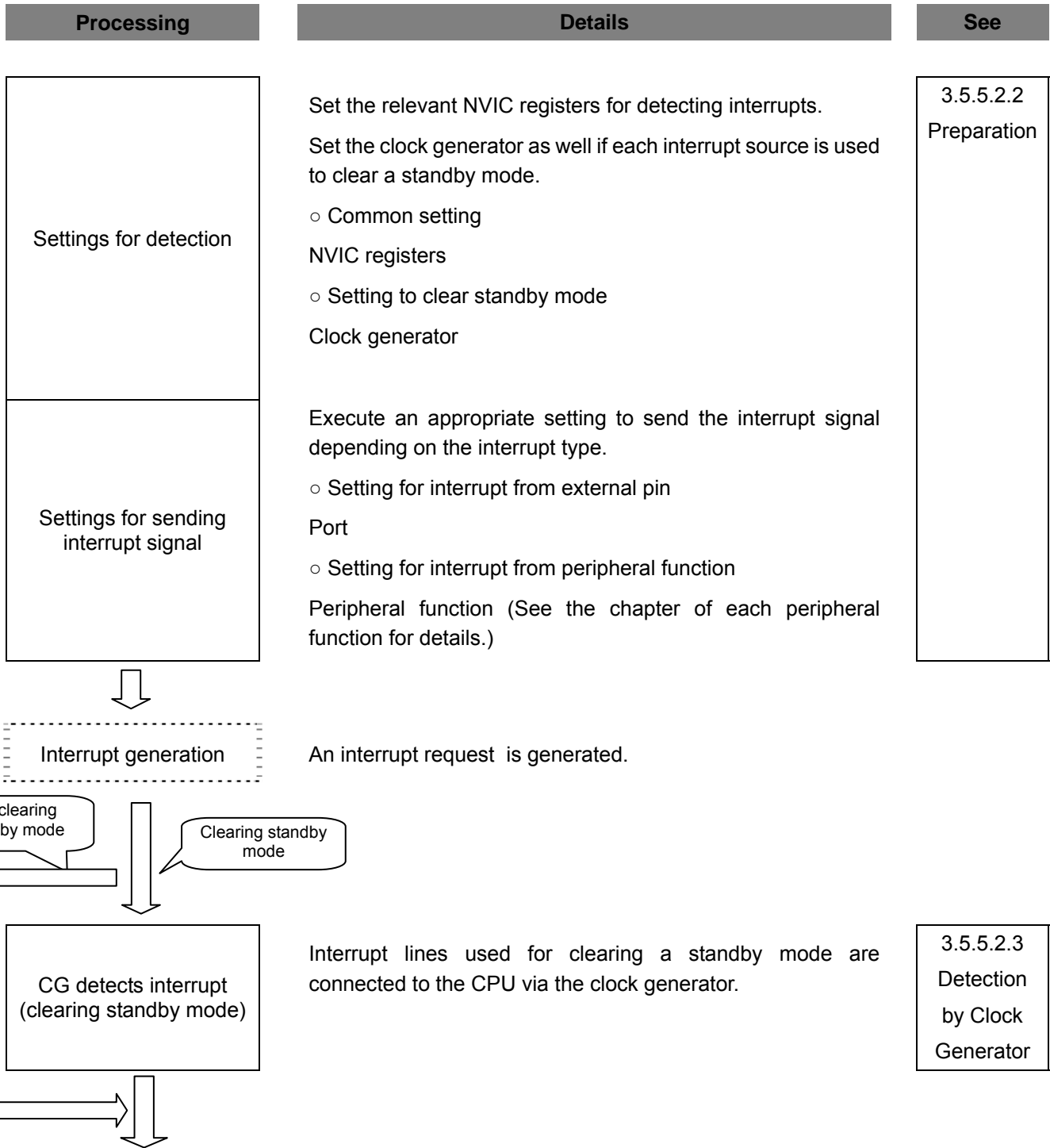
(Note)	For the remote control signal reception and real time clock timer interrupts, set the <INTxEN>bit to "1" and specify the active state as shown in Table 3.5.2, even when they are not used for clearing a standby mode.

3.5.5.2   Interrupt Handling

3.5.5.2.1   Flowchart

The following shows how an interrupt is handled.

▢ indicates hardware handling.   ▢ indicates software handling.

Processing	Details	See
Settings for detection	Set the relevant NVIC registers for detecting interrupts. Set the clock generator as well if each interrupt source is used to clear a standby mode. ○ Common setting NVIC registers ○ Setting to clear standby mode Clock generator	3.5.5.2.2 Preparation
Settings for sending interrupt signal	Execute an appropriate setting to send the interrupt signal depending on the interrupt type. ○ Setting for interrupt from external pin Port ○ Setting for interrupt from peripheral function Peripheral function (See the chapter of each peripheral function for details.)	
Interrupt generation	An interrupt request is generated.	
Not clearing standby mode / Clearing standby mode		
CG detects interrupt (clearing standby mode)	Interrupt lines used for clearing a standby mode are connected to the CPU via the clock generator.	3.5.5.2.3 Detection by Clock Generator

Processing	Details	See

⇓

| CPU detects interrupt | The CPU detects the interrupt.

If multiple interrupt requests occur simultaneously, the interrupt request with the highest priority is detected according to the priority order. | 3.5.5.2.4
Detection
by CPU |

⇓

| CPU handles interrupt | The CPU handles the interrupt.

The CPU pushes register contents to the stack before entering the ISR. | 3.5.5.2.5
CPU
processing |

⇓

| ISR execution | Program for the ISR.
Clear the interrupt source if needed. | 3.5.5.2.6
Interrupt
Service
Routire
(ISR) |

⇓

| Return to preceding program | Configure to return to the preceding program of the ISR. | |

3.5.5.2.2   Preparation

When preparing for an interrupt, you need to pay attention to the order of configuration to avoid any unexpected interrupt on the way.

Initiating an interrupt or changing its configuration must be implemented in the following order basically. Disable the interrupt by the CPU. Configure from the farthest route from the CPU. Then enable the interrupt by the CPU.

To configure the clock generator, you must follow the order indicated here not to cause any unexpected interrupt. First, configure the precondition. Secondly, clear the data related to the interrupt in the clock generator and then enable the interrupt.

The following sections are listed in the order of interrupt handling and describe how to configure them.

(1) Disabling interrupt by CPU

(2) CPU registers setting

(3) Preconfiguration 1 (Interrupt from external pin)

(4) Preconfiguration 2 (interrupt from peripheral function)

(5) Preconfiguration 3 (Interrupt Set-Pending Register)

(6) Configuring the clock generator

(7) Enabling interrupt by CPU

(1) Disabling interrupt by CPU

To make the CPU for not accepting any interrupt, write "1" to the corresponding bit of the Interrupt Clear-Enable Register. Each bit of the register, of which default setting is disabled, is assigned to a single interrupt source.

● NVIC register		
Interrupt Clear-Enable<m>	←	"1" (interrupt disabled)

(Note)   m: corresponding bit

(2) CPU registers setting

You can assign a priority level to each interrupt source in the corresponding Interrupt Priority Register of the NVIC.

Each interrupt source is provided with eight bits for assigning a priority level from 0 to 255, but the number of bits actually used varies with each product. Priority level 0 is the highest priority level. If multiple sources have the same priority, the smallest-numbered interrupt source has the highest priority.

●NVIC register		
Interrupt Priority<m>	←	"priority"

(Note)	m: corresponding bit
	This product uses three bits for assigning a priority level.

(3) Preconfiguration 1 (Interrupt from external pin)

Set "1" to the port function register of the corresponding pin. Setting PnFRx[m] allows the pin to be used as the function pin. Setting PnIE[m] allows the pin to be used as the input port.

● Port register		
PnFRx<PnmFRx>	←	「1」
PnIE<PnmIE>	←	「1」

(Note)	n: port number
	m: corresponding bit
	x: function register number
	In modes other than STOP mode, setting PnIE to enable input enables the corresponding interrupt input regardless of the PnFR setting. Be careful not to enable interrupts that are not used.

(4) Preconfiguration 2 (interrupt from peripheral function)

The setting varies depending on the peripheral function to be used. See the chapter of each peripheral function for details.

(5) Preconfiguration 3 (Interrupt Set-Pending Register)

To generate an interrupt by using the Interrupt Set-Pending Register, set "1" to the corresponding bit of this register.

● NVIC register		
Interrupt Set-Pending<m>	←	"1"

(Note)	m: corresponding bit

(6) Configuring the clock generator

For an interrupt source to be used for clearing a standby mode, you need to set the active state and enable interrupts in the CGIMCG register of the clock generator. The CGIMCG register is capable of configuring each source.

Before enabling an interrupt, clear the corresponding interrupt request already held. This can avoid unexpected interrupt. To clear corresponding interrupt request, write a value corresponding to the interrupt to be used to the CGICRCG register. See 3.5.6.3.7 CG Interrupt Requst Clear Register for each value.

Interrupt requests from external pins can be used without setting the clock generator if they are not used for clearing a standby mode. However, an "H" pulse or "H"-level signal must be input so that the CPU can detect it as an interrupt request.

● Clock generator register		
CGIMCGn<EMCGm>	←	Active state
CGICRCG<ICRCG>	←	Value corresponding to the interrupt to be used
CGIMCGn<INTmEN>	←	"1" (interrupt enabled)

(Note)   n: register number
m: number assigned to interrupt source

(7) Enabling interrupt by CPU

Enable the interrupt by the CPU as shown below.

Clear the suspended interrupt in the Interrupt Clear-Pending Register. Enable the intended interrupt with the Interrupt Set-Enable Register. Each bit of the register is assigned to a single interrupt source.

Writing "1" to the corresponding bit of the Clear-Pending Register clears the suspended interrupt. Writing "1" to the corresponding bit of the Set-Enable Register enables the intended interrupt.

If the Interrupt Set-Pending Register is used for generating an interrupt, setting of the Clear-Pending Register is not needed as this operation will cause an interrupt request to be cleared.

●NVIC register		
Interrupt Clear-Pending<m>	←	"1"
Interrupt Set-Enable<m>	←	"1"

(Note)   m: corresponding bit

3.5.5.2.3 Detection by Clock Generator

If an interrupt source is used for clearing a standby mode, an interrupt request is detected according to the active level specified in the clock generator, and is notified to the CPU.

An edge-triggered interrupt request, once detected, is held in the clock generator. A level-sensitive interrupt request must be held at the active level until it is detected, otherwise the interrupt request will cease to exist when the signal level changes from active to inactive.

When the clock generator detects an interrupt request, it keeps sending the interrupt signal in "H" level to the CPU until the interrupt request is cleared in the CG Interrupt Request Clear (CGICRCG) Register. If a standby mode is exited without clearing the interrupt request, the same interrupt will be detected again when normal operation is resumed. Be sure to clear each interrupt request in the ISR.

3.5.5.2.4 Detection by CPU

The CPU detects an interrupt request with the highest priority.

3.5.5.2.5 CPU processing

On detecting an interrupt, the CPU pushes the contents of PC, PSR, r0-r3, r12 and LR to the stack before entering the ISR.

3.5.5.2.6 Interrupt Service Routine (ISR)

An ISR requires specific programming according to the application to be used. This section describes what is recommended at the service routine programming and how the source is cleared.

(1) Pushing during ISR

An ISR normally pushes register contents to the stack and handles an interrupt as required. The Cortex-M3 core automatically pushes the contents of PC, PSR, r0-r3, r12 and LR to the stack. No extra programming is required for them.

Push the contents of other registers if needed.

Interrupt requests with higher priority and exceptions such as NMI are accepted even when an ISR is being executed. We recommend you to push the contents of general-purpose registers that might be rewritten.

(2) Clearing an interrupt source

If an interrupt source is used for clearing a standby mode, each interrupt request must be cleared with the CG Interrupt Request Clear (CGICRCG) Register.

If an interrupt source is set as level-sensitive, an interrupt request continues to exist until it is cleared at its source. Therefore, the interrupt source must be cleared. Clearing the interrupt source automatically clears the interrupt request signal from the clock generator.

If an interrupt is set as edge-sensitive, clear an interrupt request by setting the corresponding value in the CGICRCG register. When an active edge occurs again, a new interrupt request will be detected.

3.5.6 **Exception/Interrupt-Related Registers**

The CPU's NVIC registers and clock generator registers described in this chapter are shown below with their respective addresses.

3.5.6.1 Registres

●NVIC registers	
SysTick Control and Status Register	0xE000_E010
SysTick Reload Value Register	0xE000_E014
SysTick Current Value Register	0xE000_E018
SysTick Calibration Value Register	0xE000_E01C
Interrupt Set-Enable Register 1	0xE000_E100
Interrupt Set-Enable Register 2	0xE000_E104
Interrupt Set-Enable Register 3	0xE000_E108
Interrupt Set-Enable Register 4	0xE000_E10C
Interrupt Clear-Enable Register 1	0xE000_E180
Interrupt Clear-Enable Register 2	0xE000_E184
Interrupt Clear-Enable Register 3	0xE000_E188
Interrupt Clear-Enable Register 4	0xE000_E18C
Interrupt Set-Pending Register 1	0xE000_E200
Interrupt Set-Pending Register 2	0xE000_E204
Interrupt Set-Pending Register 3	0xE000_E208
Interrupt Set-Pending Register 4	0xE000_E20C
Interrupt Clear-Pending Register 1	0xE000_E280
Interrupt Clear-Pending Register 2	0xE000_E284
Interrupt Clear-Pending Register 3	0xE000_E288
Interrupt Clear-Pending Register 4	0xE000_E28C
Interrupt Priority Register	0xE000_E400-0xE000_E430
Vector Table Offset Register	0xE000_ED08
System Handler Priority Register	0xE000_ED18,0xE000_ED1C,0xE000_ED20
System Handler Control and State Register	0xE000_ED24

● Clock generator registers		
CGICRCG	CG Interrupt Request Clear Register	0x400F_4014
CGNMIFLG	NMI Flag Register	0x400F_4018
CGRSTFLG	Reset Flag Register	0x400F_401C
CGIMCGA	CG Interrupt Mode Control Register A	0x400F_4020
CGIMCGB	CG Interrupt Mode Control Register B	0x400F_4024
CGIMCGC	CG Interrupt Mode Control Register C	0x400F_4028
CGIMCGD	CG Interrupt Mode Control Register D	0x400F_402C
CGIMCGE	CG Interrupt Mode Control Register E	0x400F_4030
CGIMCGF	CG Interrupt Mode Control Register F	0x400F_4034

3.5.6.2 NVIC Register

3.5.6.2.1 SysTick Control and Status Register

	7	6	5	4	3	2	1	0
bit Symbol						CLK SOURCE	TICKINT	ENABLE
Read/Write	R					R/W	R/W	R/W
After reset	0					0	0	0
Function	"0" is read.					0: External reference clock 1: Core clock	0: Do not pend SysTick 1: Pend SysTick	0: Disable 1: Enable
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol								COUNT FLAG
Read/Write	R							R/W
After reset	0							0
Function	"0" is read.							0: Timer not counted to 0 1: Timer counted to 0
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

<bit0>　　<ENABLE>　　　　1 = The counter loads with the Reload value and then begins counting down.
　　　　　　　　　　　　　　　0 = The timer is disabled.

<bit1>　　<TICKINT>　　　　1 = SysTick exceptions are pended.
　　　　　　　　　　　　　　　0 = SysTick exceptions are not pended.

<bit2>　　<CLKSOURCE>　　0 = External reference clock
　　　　　　　　　　　　　　　1 = Core clock

<bit16>　<COUNTFLAG>　　1 =Indicates that the timer counted to 0 since last time this was read.
　　　　　　　　　　　　　　　Clears on read of any part of the SysTick Control and Status Register.

3.5.6.2.2   SysTick Reload Value Register

	7	6	5	4	3	2	1	0
bit Symbol	RELOAD							
Read/Write	R/W							
After reset	Undefined							
Function	Reload value							

	15	14	13	12	11	10	9	8
bit Symbol	RELOAD							
Read/Write	R/W							
After reset	Undefined							
Function	Reload value							

	23	22	21	20	19	18	17	16
bit Symbol	RELOAD							
Read/Write	R/W							
After reset	Undefined							
Function	Reload value							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

<bit23:0>   <RELOAD>   Set the value to load into the SysTick Current Value Register when the timer reaches "0".

(Note)	In this product, the system timer counts based on a clock obtained by dividing the clock input from the X1 pin by 32.

3.5.6.2.3 SysTick Current Value Register

	7	6	5	4	3	2	1	0
bit Symbol	CURRENT							
Read/Write	R/W							
After reset	Undefined							
Function	[Read] Current SysTick timer value [Write] Clear							

	15	14	13	12	11	10	9	8
bit Symbol	CURRENT							
Read/Write	R/W							
After reset	Undefined							
Function	[Read] Current SysTick timer value [Write] Clear							

	23	22	21	20	19	18	17	16
bit Symbol	CURRENT							
Read/Write	R/W							
After reset	Undefined							
Function	[Read] Current SysTick timer value [Write] Clear							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

<bit23:0>   <CURRENT>   [Read]   Current SysTick timer value.

[Write]   Writing to this register with any value clears it to 0. Clearing this register also clears the COUNTFLAG bit of the SysTick Control and Status Register.

3.5.6.2.4 SysTick Calibration Value Register

	7	6	5	4	3	2	1	0
bit Symbol	TENMS							
Read/Write	R							
After reset	1	1	0	0	0	1	0	0
Function	Calibration value (Note)							

	15	14	13	12	11	10	9	8
bit Symbol	TENMS							
Read/Write	R							
After reset	0	0	0	0	1	0	0	1
Function	Calibration value (Note)							

	23	22	21	20	19	18	17	16
bit Symbol	TENMS							
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Calibration value (Note)							

	31	30	29	28	27	26	25	24
bit Symbol	NOREF	SKEW						
Read/Write	R	R	R					
After reset	0	0	0					
Function	0: Reference clock provided 1: No reference clock	0: Calibration value is 10 ms. 1: Calibration value is not 10 ms.	"0" is read.					

<bit23:0>   < TENMS >   Reload value to use for 10 ms timing (0x9C4). (Note)

<bit30>   <SKEW>   1 = The calibration value is not exactly 10 ms.

<bit31>   <NOREF>   1 = The reference clock is not provided.

> **(Note)** **In this product, the system timer counts based on a clock obtained by dividing the clock input from the X1 pin by 32.**
> **The SysTick Calibration Value Register is set to a value that provides 10 ms timing when the cock input from X1 is 8 MHz.**

3.5.6.2.5 **Interrupt Set-Enable Register1**

	7	6	5	4	3	2	1	0
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 7 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 6 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 5 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 4 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 3 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 2 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 1 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 0 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

	15	14	13	12	11	10	9	8
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Reserved [Write] 1:setting prohibition [Read] "0"is read	Reserved [Write] 1:setting prohibition [Read] "0"is read	Interrupt number 13 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 12 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 11 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 10 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 9 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 8 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

	23	22	21	20	19	18	17	16
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 23 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 22 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 21 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 20 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 19 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 18 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 17 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number16 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

	31	30	29	28	27	26	25	24
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 31 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 30 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Reserved [Write] 1:setting prohibition [Read] "0"is read	Reserved [Write] 1:setting prohibition [Read] "0"is read	Interrupt number 27 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 26 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 25 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 24 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

<bit31:0>   <SETENA>   Use these bits to enable interrupts or determine which interrupts are currently enabled.

Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Disabled
1 = Enabled

(Note   **For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.**

3.5.6.2.6   Interrupt Set-Enable Register 2

	7	6	5	4	3	2	1	0
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 39 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 38 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 37 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 36 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 35 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 34 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 33 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 32 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

	15	14	13	12	11	10	9	8
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 47 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 46 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 45 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 44 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 43 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 42 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 41 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 40 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

	23	22	21	20	19	18	17	16
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 55 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 54 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 53 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 52 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 51 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 50 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 49 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 48 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

	31	30	29	28	27	26	25	24
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 63 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 62 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 61 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 60 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 59 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 58 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Reserved [Write] 1:setting prohibition [Read] "0"is read	Interrupt number 56 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

<bit31:0>   <SETENA>   Use these bits to enable interrupts or determine which interrupts are currently enabled.

Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Disabled
1 = Enabled

(Note)   For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.

3.5.6.2.7 Interrupt Set-Enable Register 3

	7	6	5	4	3	2	1	0
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 71 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 70 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 69 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 68 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 67 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 66 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 65 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 64 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

	15	14	13	12	11	10	9	8
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Reserved [Write] 1:setting prohibition [Read] "0"is read	Reserved [Write] 1:setting prohibition [Read] "0"is read	Interrupt number 77 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 76 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 75 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 74 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 73 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 72 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

	23	22	21	20	19	18	17	16
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 87 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 86 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 85 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 84 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 83 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 82 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 81 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 80 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

	31	30	29	28	27	26	25	24
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 95 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 94 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 93 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 92 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 91 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 90 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 89 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 88 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

<bit31:0>   <SETENA>   Use these bits to enable interrupts or determine which interrupts are currently enabled.

Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Disabled
1 = Enabled

(Note)   **For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.**

3.5.6.2.8   Interrupt Set-Enable Register 4

	7	6	5	4	3	2	1	0
bit Symbol					SETENA			
Read/Write			R				R/W	
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.				Interrupt number 99 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 98 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 97 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 96 [Write] 1: Enable [Read] 0: Disabled 1: Enabled
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write				R				
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write				R				
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write				R				
After reset				0				
Function	"0" is read.							

<bit3:0>   <SETENA>   Use these bits to enable interrupts or determine which interrupts are currently enabled.

Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Disabled
1 = Enabled

(Note)   For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.

3.5.6.2.9 Interrupt Clear-Enable Register1

	7	6	5	4	3	2	1	0
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 7 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 6 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 5 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 4 [Write] 1: Disable [read] 0: Disabled 1: Enabled	Interrupt number 3 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 2 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 1 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 0 [Write] 1: Disable [Read] 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Reserved [Write] 1:setting prohibition [Read] "0"is read.	Reserved [Write] 1:setting prohibition [Read] "0"is read.	Interrupt number 13 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 12 [Write] 1: Disable [read] 0: Disabled 1: Enabled	Interrupt number 11 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 10 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 9 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 8 [Write] 1: Disable [Read] 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 23 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 22 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 21 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 20 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 19 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 18 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 17 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 16 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

	31	30	29	28	27	26	25	24
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 31 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 30 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Reserved [Write] 1:setting prohibition [Read] "0"is read.	Reserved [Write] 1:setting prohibition [Read] "0"is read.	Interrupt number 27 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 26 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 25 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 24 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

<bit31:0> <CLRENA> Use these bits to disable or determine which interrupts are currently disabled.

Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Disabled
1 = Enabled

(Note) For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.

3.5.6.2.10　　Interrupt Clear-Enable Register 2

	7	6	5	4	3	2	1	0
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 39 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 38 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 37 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 36 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 35 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 34 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 33 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 32 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

	15	14	13	12	11	10	9	8
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 47 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 46 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 45 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 44 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 43 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 42 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 41 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 40 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

	23	22	21	20	19	18	17	16
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 55 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 54 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 53 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 52 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 51 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 50 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 49 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 48 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

	31	30	29	28	27	26	25	24
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 63 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 62 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 61 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 60 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 59 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 58 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Reserved [Write] 1:setting prohibition [Read] "0"is read.	Interrupt number 56 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

<bit31:0>　<CLRENA>　Use these bits to disable or determine which interrupts are currently disabled.

Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Disabled
1 = Enabled

(Note)　For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.

3.5.6.2.11 Interrupt Clear-Enable Register 3

	7	6	5	4	3	2	1	0
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 71 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 70 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 69 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 68 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 67 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 66 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 65 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 64 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

	15	14	13	12	11	10	9	8
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Reserved [Write] 1:setting prohibition [Read] "0"is read.	Reserved [Write] 1:setting prohibition [Read] "0"is read.	Interrupt number 77 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 76 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 75 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 74 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 73 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 72 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

	23	22	21	20	19	18	17	16
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 87 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 86 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 85 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 84 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 83 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 82 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 81 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 80 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

	31	30	29	28	27	26	25	24
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 95 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 94 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 93 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 92 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 91 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 90 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 89 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 88 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

<bit31:0> <CLRENA> Use these bits to disable or determine which interrupts are currently disabled.

Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Disabled
1 = Enabled

(Note) **For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.**

3.5.6.2.12   Interrupt Clear-Enable Register 4

	7	6	5	4	3	2	1	0
bit Symbol					CLRENA			
Read/Write		R			R/W			
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.				Interrupt number 99 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 98 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 97 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 96 [Write] 1: Disable [Read] 0: Disabled 1: Enabled
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write				R				
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write				R				
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write				R				
After reset								
Function	"0" is read.							

<bit3:0>   <CLRENA>   Use these bits to disable or determine which interrupts are currently disabled.

Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Disabled
1 = Enabled

(Note)   For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.

3.5.6.2.13　Interrupt Set-Pending Register 1

	7	6	5	4	3	2	1	0
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 7 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 6 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 5 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 4 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 3 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 2 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 1 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 0 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	15	14	13	12	11	10	9	8
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Reserved [Write] 1:setting prohibition [Read] "Undefined" is read.	Reserved [Write] 1:setting prohibition [Read] "Undefined" is read.	Interrupt number 13 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 12 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 11 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 10 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 9 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 8 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	23	22	21	20	19	18	17	16
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 23 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 22 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 21 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 20 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 19 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 18 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 17 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 16 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	31	30	29	28	27	26	25	24
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 31 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 30 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Reserved [Write] 1:setting prohibition [Read] "Undefined" is read.	Reserved [Write] 1:setting prohibition [Read] "Undefined" is read.	Interrupt number 27 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 26 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 25 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 24 [Write] 1: Pend [Read] 0: Not pending 1: Pending

<bit31:0>   <SETPEND>   Use these bits to force interrupts into the pending state or determine which interrupts are currently pending.

Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Not pending
1 = Pending

Each bit in this register can be cleared by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.

(Note) **For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.**

3.5.6.2.14 Interrupt Set-Pending Register 2

	7	6	5	4	3	2	1	0
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 39 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 38 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 37 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 36 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 35 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 34 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 33 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 32 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	15	14	13	12	11	10	9	8
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 47 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 46 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 45 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 44 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 43 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 42 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 41 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 40 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	23	22	21	20	19	18	17	16
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 55 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 54 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 53 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 52 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 51 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 50 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 49 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 48 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	31	30	29	28	27	26	25	24
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 63 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 62 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 61 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 60 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 59 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 58 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Reserved [Write] 1:setting prohibition [Read] "Undefined" is read.	Interrupt number 56 [Write] 1: Pend [Read] 0: Not pending 1: Pending

<bit31:0>   <SETPEND>   Use these bits to force interrupts into the pending state or determine which interrupts are currently pending.

Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Not pending
1 = Pending

Each bit in this register can be cleared by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.

(Note)   **For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.**

3.5.6.2.15   Interrupt Set-Pending Register 3

	7	6	5	4	3	2	1	0
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 71 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 70 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 69 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 68 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 67 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 66 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 65 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 64 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	15	14	13	12	11	10	9	8
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Reserved [Write] 1:setting prohibition [Read] "Undefined"is read.	Reserved [Write] 1:setting prohibition [Read] "Undefined"is read.	Interrupt number 77 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 76 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 75 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 74 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 73 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 72 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	23	22	21	20	19	18	17	16
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 87 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 86 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 85 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 84 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 83 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 82 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 81 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 80 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	31	30	29	28	27	26	25	24
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 95 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 94 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 93 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 92 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 91 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 90 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 89 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 88 [Write] 1: Pend [Read] 0: Not pending 1: Pending

<bit31:0>   <SETPEND>   Use these bits to force interrupts into the pending state or determine which interrupts are currently pending.

Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Not pending
1 = Pending

Each bit in this register can be cleared by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.

(Note) For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.

3.5.6.2.16   Interrupt Set-Pending Register 4

	7	6	5	4	3	2	1	0
bit Symbol					SETPEND			
Read/Write		R			Undefined			
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.				Interrupt number 99 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 98 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 97 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 96 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write				R				
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write				R				
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write				R				
After reset				0				
Function	"0" is read.							

<bit3:0>   <SETPEND>   Use these bits to force interrupts into the pending state or determine which interrupts are currently pending.

Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Not pending
1 = Pending

Each bit in this register can be cleared by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.

(Note)   For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.

3.5.6.2.17 Interrupt Clear-Pending Register1

	7	6	5	4	3	2	1	0
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 7 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 6 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 5 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 4 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 3 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 2 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 1 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 0 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	15	14	13	12	11	10	9	8
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Reserved [Write] 1:setting prohibition [Read] "Undefined"is read.	Reserved [Write] 1:setting prohibition [Read] "Undefined"is read.	Interrupt number 13 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 12 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 11 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 10 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 9 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 8 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	23	22	21	20	19	18	17	16
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 23 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 22 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 21 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 20 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 19 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 18 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 17 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 16 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	31	30	29	28	27	26	25	24
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 31 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 30 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Reserved [Write] 1:setting prohibition [Read] "Undefined" is read.	Reserved [Write] 1:setting prohibition [Read] "Undefined" is read.	Interrupt number 27 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 26 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 25 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 24 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

<bit31:0>   <CLRPEND>   Use these bits to clear pending interrupts or determine which interrupts are currently pending.

Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Not pending
1 = Pending

(Note)   For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.

3.5.6.2.18   Interrupt Clear-Pending Register 2

	7	6	5	4	3	2	1	0
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 39 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 38 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 37 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 36 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 35 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 34 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 33 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 32 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	15	14	13	12	11	10	9	8
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 47 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 46 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 45 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 44 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 43 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 42 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 41 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 40 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	23	22	21	20	19	18	17	16
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 55 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 54 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 53 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 52 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 51 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 50 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 49 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 48 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	31	30	29	28	27	26	25	24
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 63 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 62 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 61 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 60 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 59 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 58 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Reserved [Write] 1:setting prohibition [Read] "Undefined" is read.	Interrupt number 56 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

<bit31:0>   <CLRPEND>   Use these bits to clear pending interrupts or determine which interrupts are currently pending.

Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Not pending
1 = Pending

(Note)   **For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.**

3.5.6.2.19   Interrupt Clear-Pending Register 3

	7	6	5	4	3	2	1	0
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 71 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 70 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 69 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 68 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 67 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 66 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 65 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 64 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	15	14	13	12	11	10	9	8
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Reserved [Write] 1:setting prohibition [Read] "Undefined"is read.	Reserved [Write] 1:setting prohibition [Read] "Undefined"is read.	Interrupt number 77 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 76 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 75 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 74 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 73 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 72 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	23	22	21	20	19	18	17	16
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 87 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 86 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 85 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 84 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 83 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 82 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 81 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 80 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	31	30	29	28	27	26	25	24
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	不定							
Function	Interrupt number 95 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 94 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 93 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 92 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 91 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 90 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 89 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 88 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

<bit31:0>   <CLRPEND>   Use these bits to clear pending interrupts or determine which interrupts are currently pending.

Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Not pending
1 = Pending

(Note)   For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.

3.5.6.2.20   Interrupt Clear-Pending Register 4

	7	6	5	4	3	2	1	0
bit Symbol					CLRPEND			
Read/Write	R				Undefined			
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.				Interrupt number 99 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 98 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 97 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 96 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset								
Function	"0" is read.							

<bit3:0>   <CLRPEND>   Use these bits to clear pending interrupts or determine which interrupts are currently pending.

Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Not pending
1 = Pending

(Note)   **For descriptions of interrupts and interrupt numbers, see Section 3.5.5.1.4 List of Interrupt Sources.**

3.5.6.2.21 Interrupt Priority Registers

Each interrupt is provided with eight bits of an Interrupt Priority Register.

The following shows the addresses of the Interrupt Priority Registers corresponding to interrupt numbers.

	31 24	23 16	15 8	7 0
0xE000_E400	PRI_3	PRI_2	PRI_1	PRI_0
0xE000_E404	PRI_7	PRI_6	PRI_5	PRI_4
0xE000_E408	PRI_11	PRI_10	PRI_9	PRI_8
0xE000_E40C	Reserved	Reserved	PRI_13	PRI_12
0xE000_E410	PRI_19	PRI_18	PRI_17	PRI_16
0xE000_E414	PRI_23	PRI_22	PRI_21	PRI_20
0xE000_E418	PRI_27	PRI_26	PRI_25	PRI_24
0xE000_E41C	PRI_31	PRI_30	Reserved	Reserved
0xE000_E420	PRI_35	PRI_34	PRI_33	PRI_32
0xE000_E424	PRI_39	PRI_38	PRI_37	PRI_36
0xE000_E428	PRI_43	PRI_42	PRI_41	PRI_40
0xE000_E42C	PRI_47	PRI_46	PRI_45	PRI_44
0xE000_E430	PRI_51	PRI_50	PRI_49	PRI_48
0xE000_E424	PRI_55	PRI_54	PRI_53	PRI_52
0xE000_E428	PRI_59	PRI_58	Reserved	PRI_56
0xE000_E42C	PRI_63	PRI_62	PRI_61	PRI_60
0xE000_E424	PRI_67	PRI_66	PRI_65	PRI_64
0xE000_E428	PRI_71	PRI_70	PRI_69	PRI_68
0xE000_E42C	PRI_75	PRI_74	PRI_73	PRI_72
0xE000_E428	Reserved	Reserved	PRI_77	PRI_76
0xE000_E42C	PRI_83	PRI_82	PRI_81	PRI_80
0xE000_E424	PRI_87	PRI_86	PRI_85	PRI_84
0xE000_E428	PRI_91	PRI_90	PRI_89	PRI_88
0xE000_E42C	PRI_95	PRI_94	PRI_93	PRI_92
0xE000_E428	PRI_99	PRI_98	PRI_97	PRI_96

The number of bits to be used for assigning a priority varies with each product. This product uses three bits for assigning a priority.

The following shows the fields of the Interrupt Priority Registers for interrupt numbers 0 to 3. The Interrupt Priority Registers for all other interrupt numbers have the identical fields. Unused bits return "0" when read, and writing to unused bits has no effect.

	7	6	5	4	3	2	1	0
bit Symbol		PRI_0						
Read/Write		R/W				R		
After reset		0				0		
Function	Priority of interrupt number 0			"0" is read.				

	15	14	13	12	11	10	9	8
bit Symbol		PRI_1						
Read/Write		R/W				R		
After reset		0				0		
Function	Priority of interrupt number 1			"0" is read.				

	23	22	21	20	19	18	17	16
bit Symbol		PRI_2						
Read/Write		R/W				R		
After reset		0				0		
Function	Priority of interrupt number 2			"0" is read.				

	31	30	29	28	27	26	25	24
bit Symbol		PRI_3						
Read/Write		R/W				R		
After reset		0				0		
Function	Priority of interrupt number 3			"0" is read.				

<bit7:5>   <PRI_0>   Priority of interrupt number 0

<bit15:13>   <PRI_1>   Priority of interrupt number 1

<bit23:21>   <PRI_2>   Priority of interrupt number 2

<bit31:29>   <PRI_3>   Priority of interrupt number 3

3.5.6.2.22   Vector Table Offset Register

	7	6	5	4	3	2	1	0
bit Symbol	TBLOFF							
Read/Write	R/W	R						
After reset	0	0						
Function	Offset value	"0" is read.						
	15	14	13	12	11	10	9	8
bit Symbol	TBLOFF							
Read/Write	R/W							
After reset	0							
Function	Offset value							
	23	22	21	20	19	18	17	16
bit Symbol	TBLOFF							
Read/Write	R/W							
After reset	0							
Function	Offset value							
	31	30	29	28	27	26	25	24
bit Symbol			TBLBASE	TBLOFF				
Read/Write	R		R/W	R/W				
After reset	0		0	0				
Function	"0" is read.		Table base	Offset value				

<bit28:7>   <TBLOFF>   Set the offset value from the top of the space specified in TBLBASE.
The offset must be aligned based on the number of exceptions in the table. This means that the minimum alignment is 32 words that you can use for up to 16 interrupts. For more interrupts, you must adjust the alignment by rounding up to the next power of two.

<bit29>   <TBLBASE>   The vector table is in:
0 = Code space
1 = SRAM space

3.5.6.2.23   System Handler Priority Registers

Each exception is provided with eight bits of a System Handler Priority Register.

The following shows the addresses of the System Handler Priority Registers corresponding to each exception.

	31 24	23 16	15 8	7 0
0xE000_ED18	PRI_7	PRI_6 (Usage Fault)	PRI_5 (Bus Fault)	PRI_4 (Memory Management)
0xE000_ED1C	PRI_11 (SVCall)	PRI_10	PRI_9	PRI_8
0xE000_ED20	PRI_15 (SysTick)	PRI_14 (PendSV)	PRI_13	PRI_12 (Debug Monitor)

The number of bits to be used for assigning a priority varies with each product. This product uses three bits for assigning a priority.

The following shows the fields of the System Handler Priority Registers for Memory Management, Bus Fault and Usage Fault. The System Handler Priority Registers for all other exceptions have the identical fields. Unused bits return "0" when read, and writing to unused bits has no effect.

	7	6	5	4	3	2	1	0
bit Symbol	PRI_4							
Read/Write	R/W			R				
After reset	0			0				
Function	Priority of Memory Management			"0" is read.				

	15	14	13	12	11	10	9	8
bit Symbol	PRI_5							
Read/Write	R/W			R				
After reset	0			0				
Function	Priority of Bus Fault			"0" is read.				

	23	22	21	20	19	18	17	16
bit Symbol	PRI_6							
Read/Write	R/W			R				
After reset	0			0				
Function	Priority of Usage Fault			"0" is read.				

	31	30	29	28	27	26	25	24
bit Symbol	PRI_7							
Read/Write	R/W			R				
After reset	0			0				
Function	Reserved			"0" is read.				

3.5.6.2.24   System Handler Control and State Register

	7	6	5	4	3	2	1	0
bit Symbol	SVCALL ACT				USGFAULT ACT		BUSFAULT ACT	MEMFAULT ACT
Read/Write	R/W	R			R/W	R	R/W	R/W
After reset	0	0			0	0	0	0
Function	SVCall 0: Inactive 1: Active	"0" is read.			Usage fault 0: Inactive 1: Active	"0" is read.	Bus fault 0: Inactive 1: Active	Memory Management 0: Inactive 1: Active
	15	14	13	12	11	10	9	8
bit Symbol	SVCALL PENDED	BUSFAULT PENDED	MEMFAULT PENDED	USGFAULT PENDED	SYSTICK ACT	PENDSV ACT		MONITOR ACT
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	SVCall 0: Not pended 1: Pended	Bus Fault 0: Not pended 1: Pended	Memory Management 0: Not pended 1: Pended	Usage Fault 0: Not pended 1: Pended	SysTick 0: Inactive 1: Active	PendSV 0: Inactive 1: Active	"0" is read.	Debug Monitor 0: Inactive 1: Active
	23	22	21	20	19	18	17	16
bit Symbol						USGFAULT ENA	BUSFAULT ENA	MEMFAULT ENA
Read/Write	R					R/W	R/W	R/W
After reset	0					0	0	0
Function	"0" is read.					Usage Fault 0: Disable 1: Enable	Bus Fault 0: Disable 1: Enable	Memory Management 0: Disable 1: Enable
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

<bit0>   <MEMFAULTACT>   Reads as "1" if Memory Management is active.
<bit1>   <BUSFAULTACT>   Reads as "1" if Bus Fault is active.
<bit3>   <USGFALTACT>   Reads as "1" if Usage Fault is active.
<bit7>   <SVCALLACT>   Reads as "1" if SVCall is active.
<bit8>   <MONITORACT>   Reads as "1" if Debug Monitor is active.
<bit10>   <PENDSVACT>   Reads as "1" if PendSV is active.
<bit11>   <SYSTICKACT>   Reads as "1" if SysTick is active.
<bit12>   <USGFAULTPENDED>   Reads as "1" if Usage Fault is pended.
<bit13>   <MEMFAULTPENDED>   Reads as "1" if Memory Management is pended.
<bit14>   <BUSFAULTPENDED>   Reads as "1" if Bus Fault is pended.
<bit15>   <SVCALLPENDED>   Reads as "1" if SVCall is pended.
<bit16>   <MEMFAULTENA>   Set to "0" to disable or "1" to enable Memory Management.
<bit17>   <BUSFAULTENA>   Set to "0" to disable or "1" to enable Bus Fault.
<bit18>   <USGFAULTENA>   Set to "0" to disable or "1" to enable Usage Fault.

> **(Note)   You must clear or set the active bits with extreme caution because clearing and setting these bits does not repair stack contents.**

3.5.6.3   Clock Generaror Registers

3.5.6.3.1   CG Interrupt Mode Control Register A

CGIMCGA
(0x400F_4020)

	7	6	5	4	3	2	1	0
bit Symbol		EMCG0			EMST0			INT0EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT0 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT0 standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INT0 clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol		EMCG1			EMST1			INT1EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT1 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT1 standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INT1 clear input 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol		EMCG2			EMST2			INT2EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT2 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT2 standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INT2 clear input 0: Disable 1: Enable

	31	30	29	28	27	26	25	24
bit Symbol		EMCG3			EMST3			INT3EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT3 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT3 standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INT3 clear input 0: Disable 1: Enable

(Note 1)   Refer to EMSTx bit to know the active condition which is used for clearing standby.

(Note 2)   Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

(Note 3)   Operate only setting <EMCGx> = "100" (Both edges).

3.5.6.3.2 CG Interrupt Mode Control Register B

CGIMCGB
(0x400F_4024)

	7	6	5	4	3	2	1	0
bit Symbol		EMCG4			EMST4			INT4EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT4 standby clear request (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT4 standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INT4 clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol		EMCG5			EMST5			INT5EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT5 clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT5 standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INT5 clear input 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol		EMCG6			EMST6			INT6EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT6 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT6 standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INT6 clear input 0: Disable 1: Enable

	31	30	29	28	27	26	25	24
bit Symbol		EMCG7			EMST7			INT7EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT7 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT7 standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INT7 clear input 0: Disable 1: Enable

> **(Note 1)** Refer to EMSTx bit to know the active condition which is used for clearing standby.
>
> **(Note 2)** Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.
>
> **(Note 3)** Operate only setting <EMCGx> ="100"(Both edges).

3.5.6.3.3   CG Interrupt Mode Control Register C

CGIMCGC
(0x400F_4028)

	7	6	5	4	3	2	1	0
bit Symbol		EMCG8			EMST8			INT8EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT8 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT8 standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INT8 clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol		EMCG9			EMST9			INT9EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT9 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT9 standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INT9 clear input 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol		EMCGA			EMSTA			INTAEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INTA standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTA standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INTA clear input 0: Disable 1: Enable

	31	30	29	28	27	26	25	24
bit Symbol		EMCGB			EMSTB			INTBEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INTB standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTB standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INTB clear input 0: Disable 1: Enable

> **(Note 1)**   Refer to EMSTx bit to know the active condition which is used for clearing standby.
>
> **(Note 2)**   Please specify the bit for the edge first and then specify the bit for the \<INTxEN>. Setting them simultaneously is prohibited.
>
> **(Note 3)**   Operate only setting \<EMCGx> ="100"(Both edges).

3.5.6.3.4 CG Interrupt Mode Control Register D

CGIMCGD
(0x400F_402C)

	7	6	5	4	3	2	1	0
bit Symbol		EMCGC			EMSTC			INTCEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INTC standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTC standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INTC clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol		EMCGD			EMSTD			INTDEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INTD standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTD standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INTD clear input 0: Disable 1: Enable

(Note 1) Refer to EMSTx bit to know the active condition which is used for clearing standby.

(Note 2) Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

(Note 3) Operate only setting <EMCGx> ="100"(Both edges).

3.5.6.3.5 CG Interrupt Mode Control Register E

CGIMCGE
(0x400F_4030)

	7	6	5	4	3	2	1	0
bit Symbol								
Read/Write	R	R			R		R	R
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	"010" is read.			"00" is read.		Reserved "Undefined" is read.	"0" is read.

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R	R			R		R	R
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	"010" is read.			"00" is read.		Reserved "Undefined" is read.	"0" is read.

	23	22	21	20	19	18	17	16
bit Symbol		EMCGI			EMSTI			INTIEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INTRMCRX0 standby clear request. **Set it as shown below.** 011: Rising edge			Active state of INTRMCRX0 standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INTRMCRX0 clear input 0: Disable 1: Enable

	31	30	29	28	27	26	25	24
bit Symbol		EMCGJ			EMSTJ			INTJEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INTRMCRX1 standby clear request. **Set it as shown below.** 011: Rising edge			Active state of INTRMCRX1 standby clear request (Note3) 00: — 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INTRMCRX1 clear input 0: Disable 1: Enable

---

(Note 1) Refer to EMSTx bit to know the active condition which is used for clearing standby.

(Note 2) Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

(Note 3) Operate only setting <EMCGx> ="100"(Both edges).

3.5.6.3.6   CG Interrupt Mode Control Register F

CGIMCGF
(0x400F_4034)

	7	6	5	4	3	2	1	0
bit Symbol		EMCGK			EMSTK			INTKEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INTRTC standby clear request. **Set it as shown below.** 010: Falling edge			Active state of INTRTC standby clear request. (Note3) 00: - 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INTRTC clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol		EMCGL			EMSTL			INTLEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INTKWUP standby clear request. **Set it as shown below.** 001: "H" level			Active state of INTKWUP standby clear request. (Note3) 00: - 01: Rising edge 10: Falling edge 11: Both edges		Reserved "Undefined" is read.	INTKWUP clear input 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							

(Note 1)   Refer to EMSTx bit to know the active condition which is used for clearing standby.

(Note 2)   Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

(Note 3)   Operate only setting <EMCGx> ="100"(Both edges).

When using interrupts for waking up from STOP or IDLE mode, you must set the active state for interrupt requests from each source.

(Note 1) When using interrupts, be sure to follow the steps below to configure each interrupt:

1) If the relevant pin has an alternative function, such as general-purpose port, enable the interrupt input function.

2) Set the active state at initial configuration.

3) Clear interrupt requests.

4) Enable the interrupt.

(Note 2)   The above settings must be made while the interrupt is disabled.

(Note 3) Fifteen interrupts can be used for waking up from STOP mode: INT0 to INTD and KWUP interrupts. Whether to use these interrupts for waking up from STOP mode and the active state for each interrupt are set in the clock generator.

(Note 4) INT0 to INTD can be used without configuring them in the clock generator when used as normal interrupts.

3.5.6.3.7   CG Interrupt Request Clear Register

CGICRCG
(0x400F_4014)

	7	6	5	4	3	2	1	0
bit Symbol				ICRCG				
Read/Write	R			W				
After reset	0			0	0	0	0	0
Function	"0" is read.			Clear interrupt requests. (0_0000 : After reset default) 0_0000 : INT0 0_1000 : INT8 1_0000 : Reserved 0_0001 : INT1 0_1001 : INT9 1_0001 : Reserved 0_0010 : INT2 0_1010 : INTA 1_0010 : INTRMCRX0 0_0011 : INT3 0_1011 : INTB 1_0011 : INTRMCRX1 0_0100 : INT4 0_1100 : INTC 1_0100 : INTRTC 0_0101 : INT5 0_1101 : INTD 1_0101 : INTKWUP 0_0110 : INT6 0_1110 : Reserved 0_0111 : INT7 0_1111 : Reserved 1_0110 ～ 1_1111 : Reserved * "0" is read.				

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

3.5.6.3.8   NMI Flag Register

CGNMIFLG

(0x400F_4018)

	7	6	5	4	3	2	1	0
bit Symbol							NMIFLG1	NMIFLG0
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.						NMI source generation flag 0: not applicable 1: generated from NMI pin	NMI source generation flag 0: not applicable 1: generated from WDT

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							

(Note)   <NMIFLG1:0> are cleared to "0" when they are read.

3.5.6.3.9   Reset Flag Register

CGRSTFLG

(0x400F_401C)

	7	6	5	4	3	2	1	0
bit Symbol				DBGRSTF	BUPRSTF	WDTRSTF	PINRSTF	PONRSTF
Read/Write	R			R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	1	1
Function	"0" is read.			Debug reset flag 0: "0" is written 1: Reset from SYSRSTRQ (Note 2)	BACKUP reset flag 0: "0" is written 1: Reset from BACKUP	WDT reset flag 0: "0" is written 1: Reset from WDT	RESET pin flag 0: "0" is written 1: Reset from RESET pin	Power-on reset flag 0: "0" is written 1: Reset from power-on reset
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							

(Note 1)   **The TMPM322 has power-on reset circuit and this register is initialized only by power-on reset. Therefore, "1" is set to the <PONRSTF> bit in initial reset state right after power-on. Note that this bit is not set by the second and subsequent resets and this register is not cleared automatically. Write "0" to clear the register.**

(Note 2)   **This flag indicates a reset generated by the SYSRESETREQ bit of the Application Interrupt and Reset Control Register of the CPU's NVIC.**

3.6   Port Function

3.6.1   Port registers

PxDATA : Port register
        To read/ write port data.

PxCR   : Control register
        To control input/output
        * Need to enable the input with PxIE register even when input is set.

PxFRn   : Function register
        To set functions. An assigned function can be activated by setting "1".

PxOD   : Open drain control register
        To switch the input of a register that can be set as programmable open drain.

PxPUP   : Pull up control register
        To control program pull ups.

PxIE   : Input control enable register
        To control inputs. "0" is set as default to avoid through current. This setting prohibits
        inputs.

3.6.2 Port registers explanation

3.6.2.1 Port A

The port A is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port A performs the data bus (D0-D7)function.

Reset initializes all bits of the port A as general-purpose ports with input, output and pull-up disabled.

If this port is used for external bus function, PACR,PAFR1 and PAIE must be set to "1."

Port A register

PADATA (0x400C_0000)		7	6	5	4	3	2	1	0
	Bit Symbol	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	Read/Write	R/W							
	After reset	"0"							

Port A control register

PACR (0x400C_0004)		7	6	5	4	3	2	1	0
	Bit Symbol	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Output disabled 1: Output enabled							

Port A functions register1

PAFR1 (0x400C_0008)		7	6	5	4	3	2	1	0
	Bit Symbol	PA6F1	PA6F1	PA5F1	PA4F1	PA3F1	PA2F1	PA1F1	PA0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: D7	0: PORT 1: D6	0: PORT 1: D5	0: PORT 1: D4	0: PORT 1: D3	0: PORT 1: D2	0: PORT 1: D1	0: PORT 1: D0

Port A open drain control register

PAOD (0x400C_0028)		7	6	5	4	3	2	1	0
	Bit Symbol	PA7OD	PA6OD	PA5OD	PA4OD	PA3OD	PA2OD	PA1OD	PA0OD
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain

Port A pull-up control register

PAPUP (0x400C_002C)		7	6	5	4	3	2	1	0
	Bit Symbol	PA7UP	PA6UP	PA5UP	PA4UP	PA3UP	PA2UP	PA1UP	PA0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on

Port A input enable control register

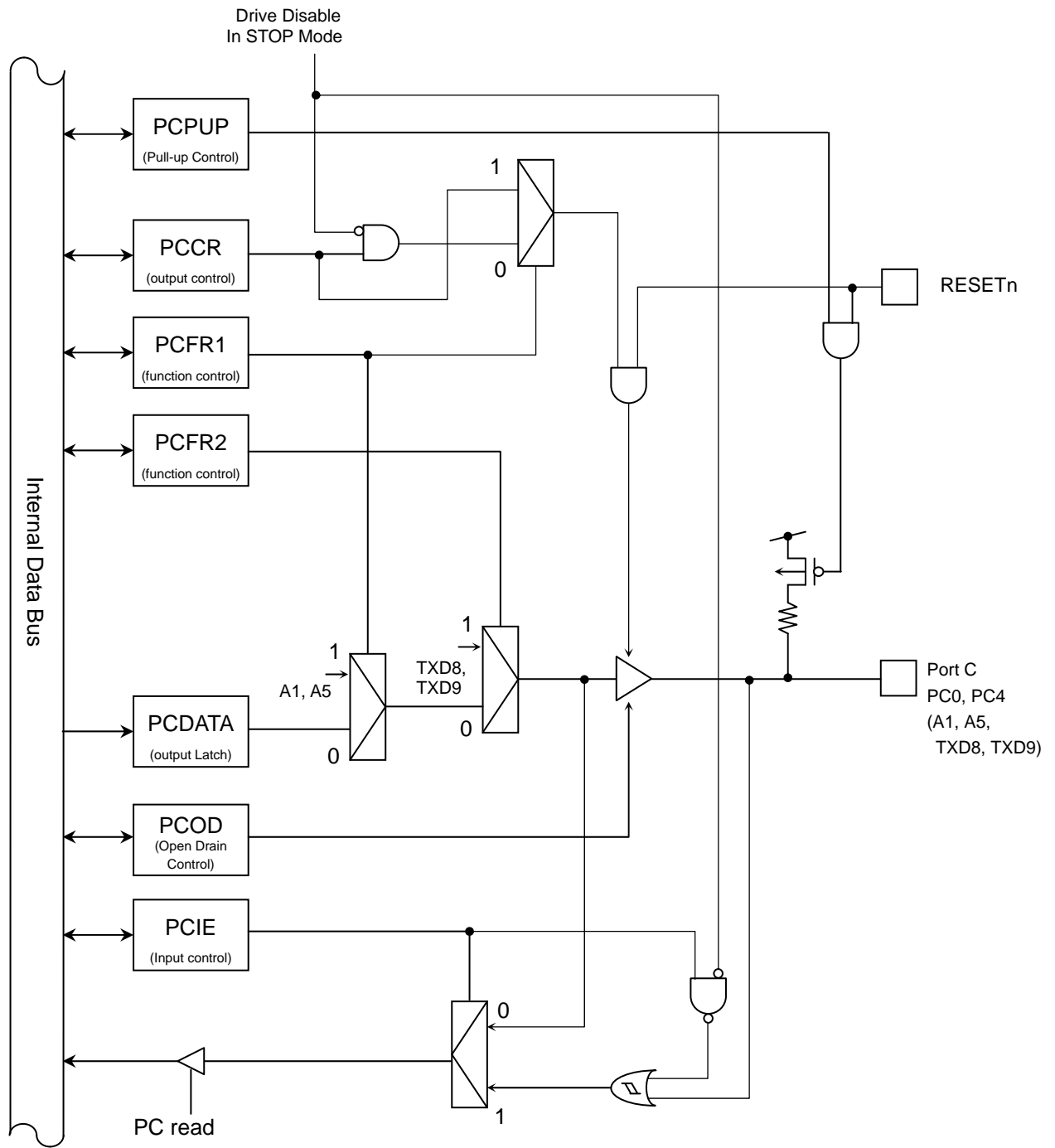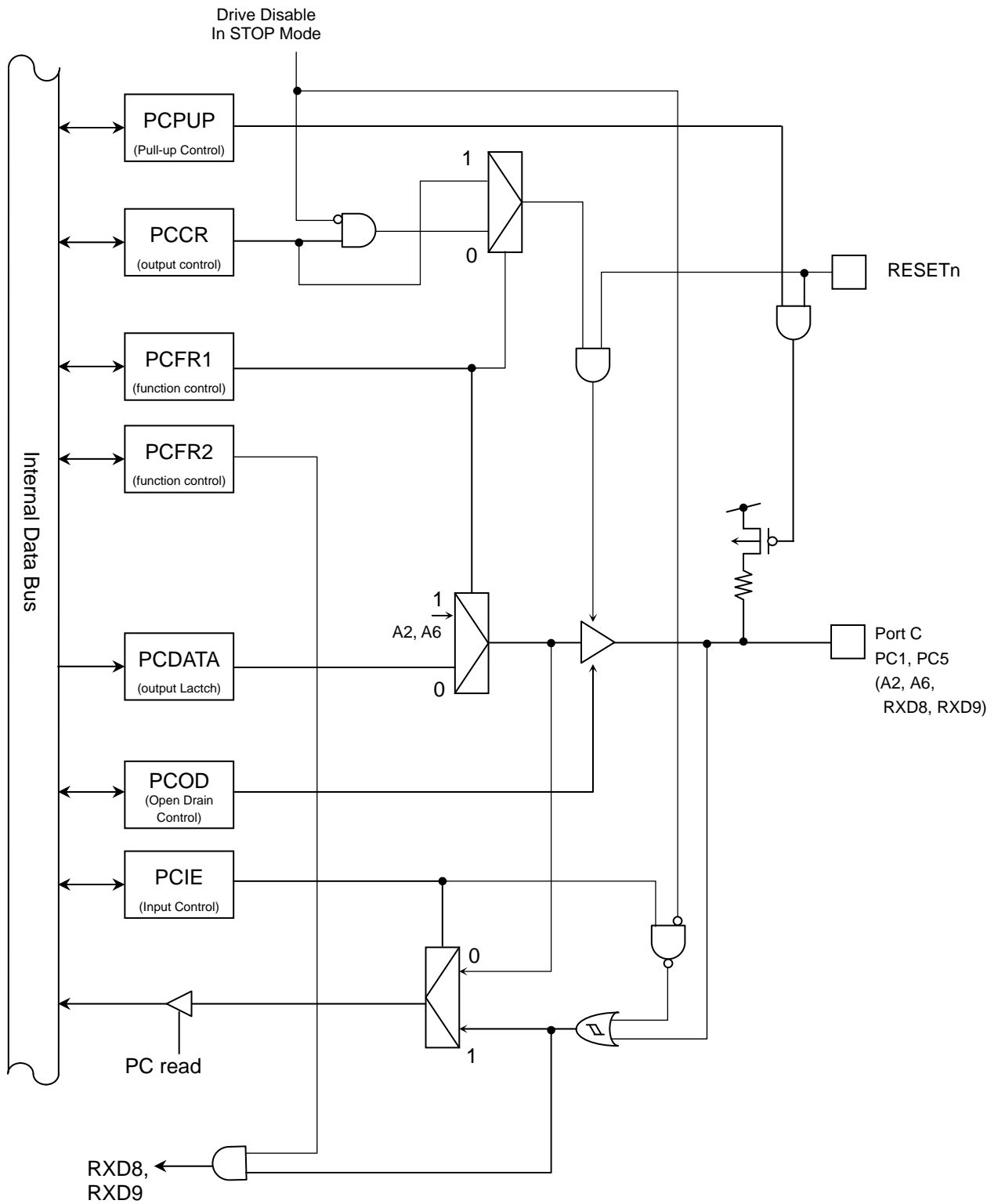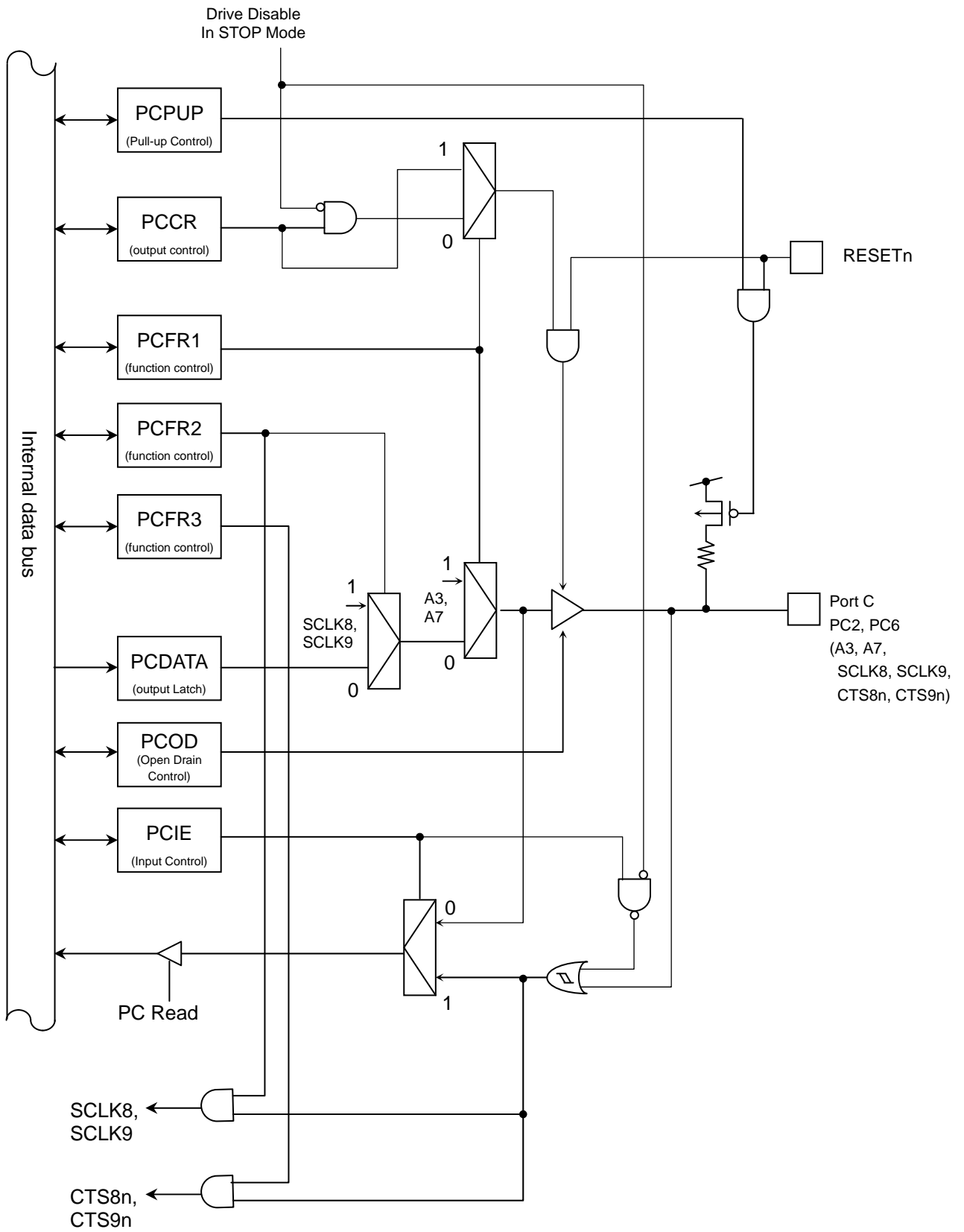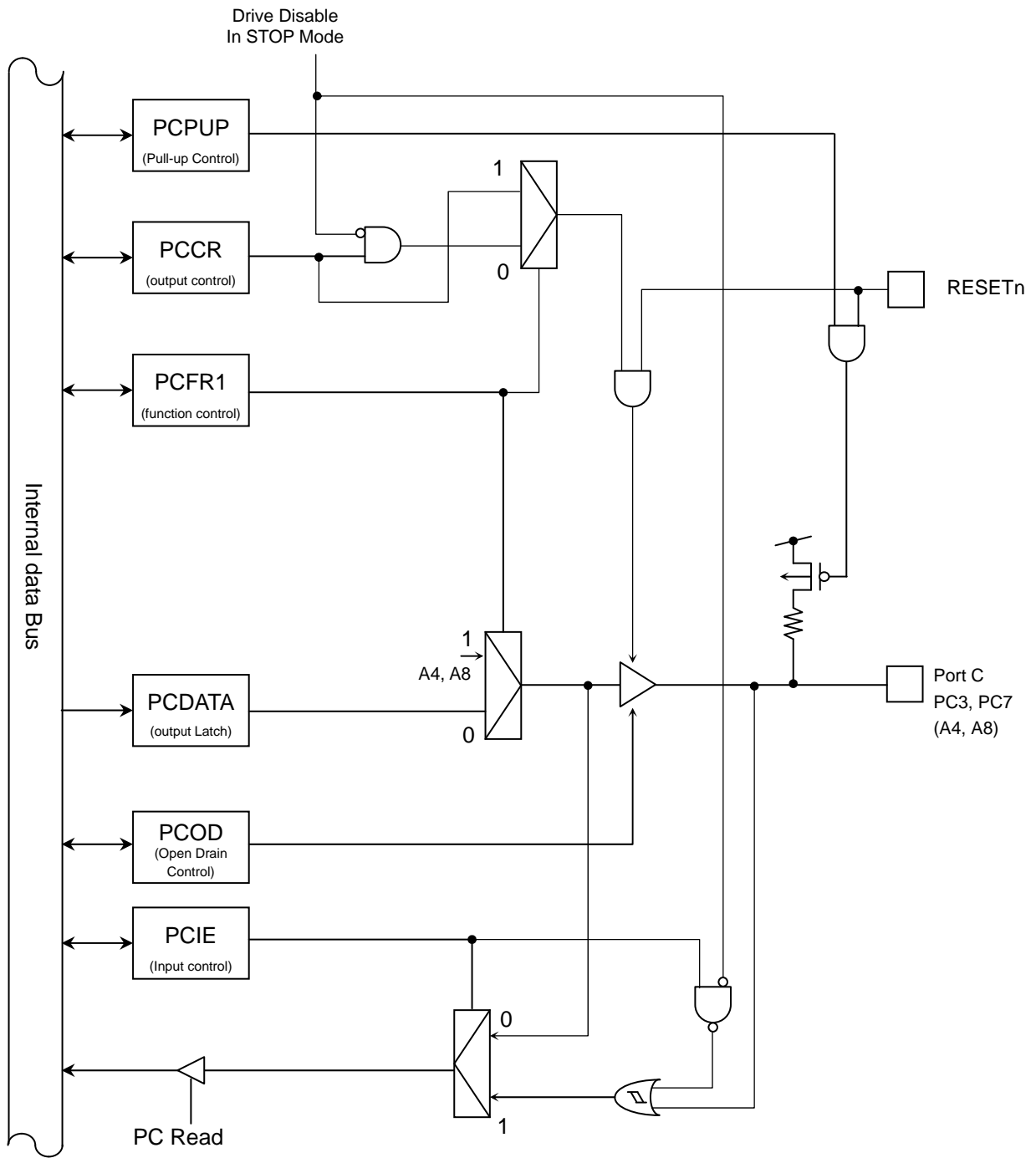| PAIE (0x400C_0038)		7	6	5	4	3	2	1	0
	Bit Symbol	PA7IE	PA6IE	PA5IE	PA4IE	PA3IE	PA2IE	PA1IE	PA0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

Fig3.6.1　PortA (PA0～PA7)

3.6.2.2 Port B

The port B is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port A performs the data bus (D8-D15)function.

Reset initializes all bits of the port B as general-purpose ports with input, output and pull-up disabled.

If this port is used for external bus function, PBCR,PBFR1 and PBIE must be set to "1."

Port B register

PBDATA (0x400C_0100)		7	6	5	4	3	2	1	0
	Bit Symbol	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
	Read/Write	R/W							
	After reset	"0"							

Port B control register

PBCR (0x400C_0104)		7	6	5	4	3	2	1	0
	Bit Symbol	PB7C	PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Output disabled 1: Output enabled							

PortB function register1

PBFR1 (0x400C_0108)		7	6	5	4	3	2	1	0
	Bit Symbol	PB7F1	PB6F1	PB5F1	PB4F1	PB3F1	PB2F1	PB1F1	PB0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: D15	0: PORT 1: D14	0: PORT 1: D13	0: PORT 1: D12	0: PORT 1: D11	0: PORT 1: D10	0: PORT 1: D9	0: PORT 1: D8

Port B open drain control register

PBOD (0x400C_0128)		7	6	5	4	3	2	1	0
	Bit Symbol	PB7 OD	PB6 OD	PB5 OD	PB4 OD	PB3 OD	PB2 OD	PB1 OD	PB0OD
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain

Port B pull-up control register

PBPUP (0x400C_012C)		7	6	5	4	3	2	1	0
	Bit Symbol	PB7UP	PB6UP	PB5UP	PB4UP	PB3UP	PB2UP	PB1UP	PB0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on

Port B input enable control register

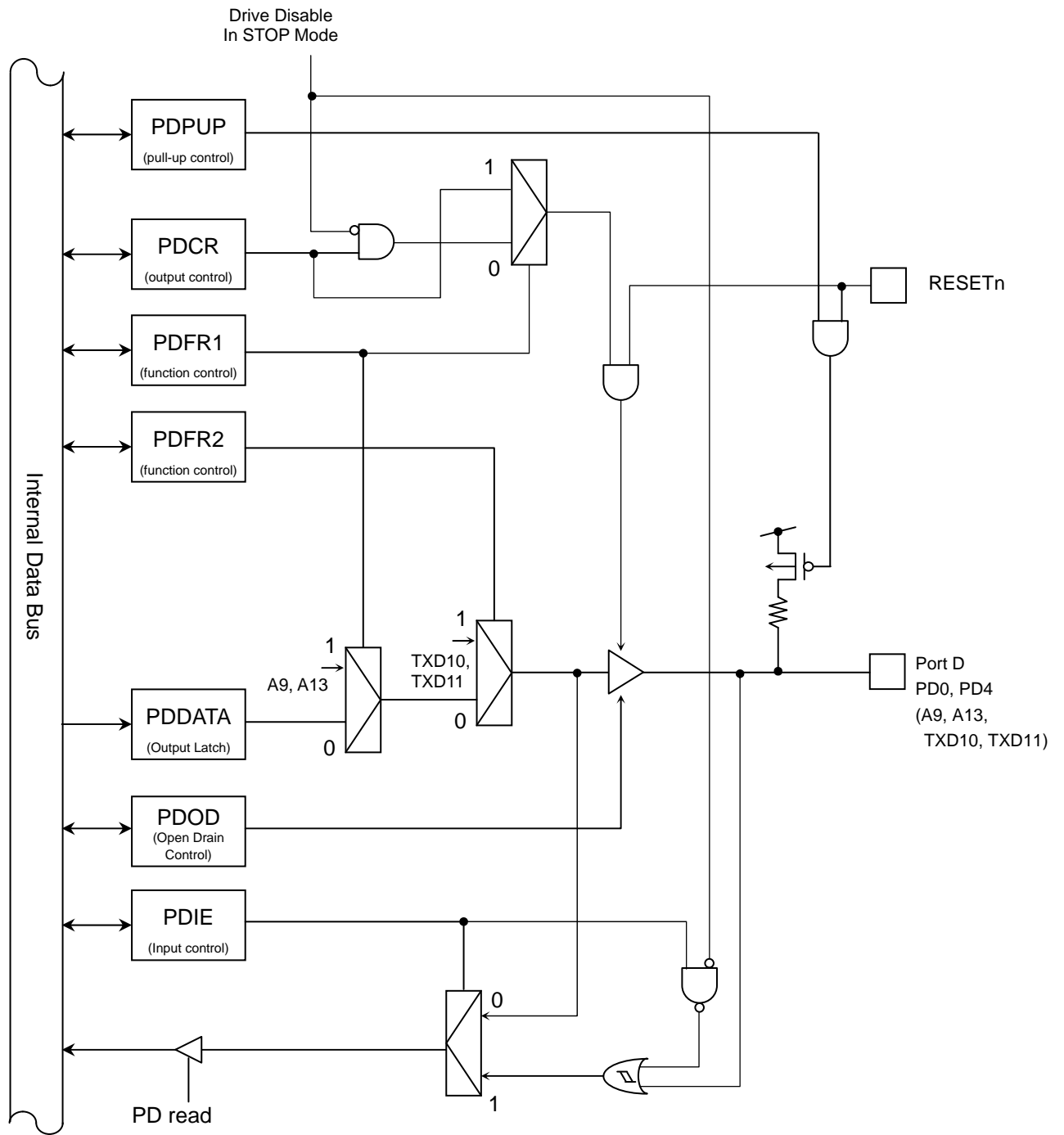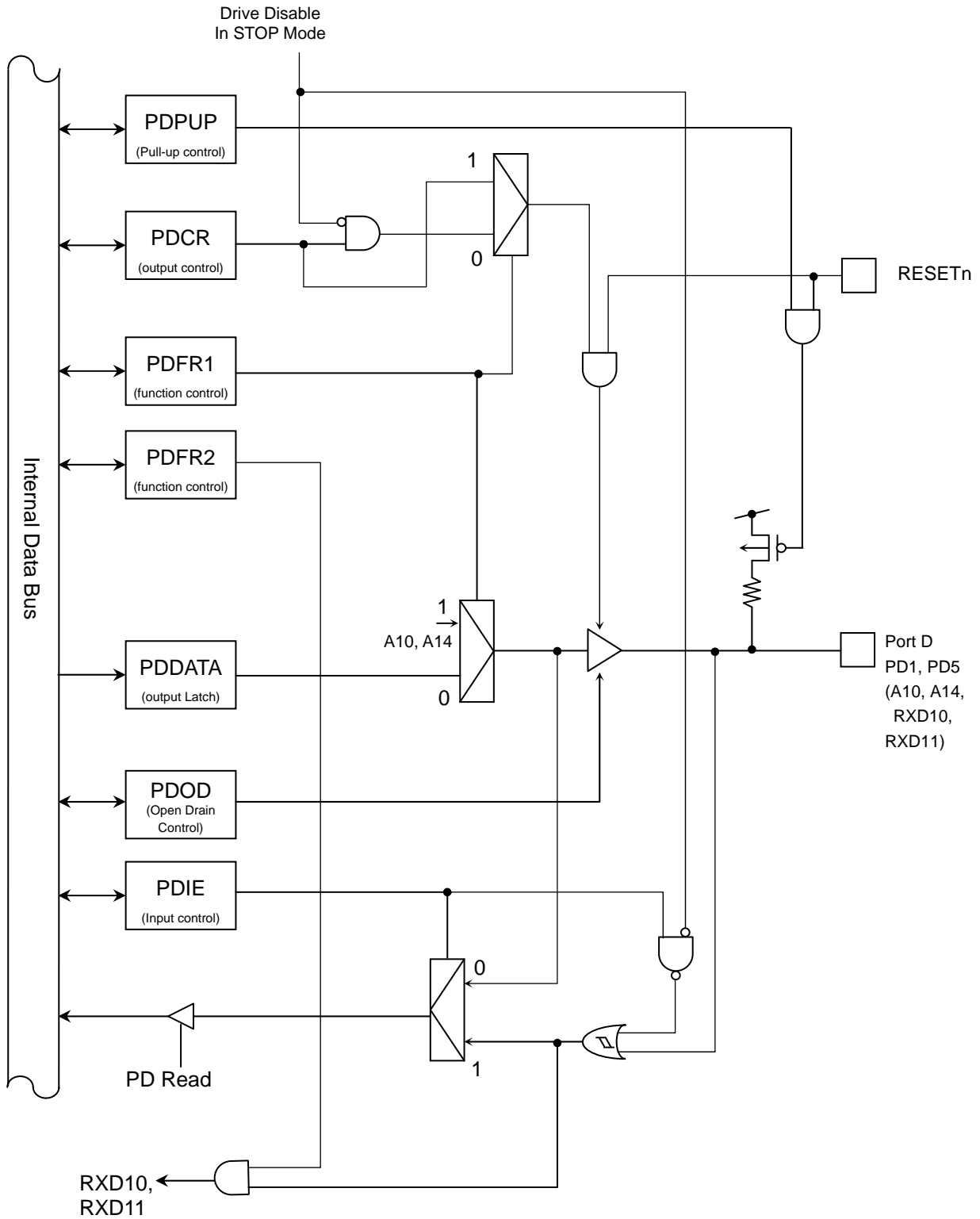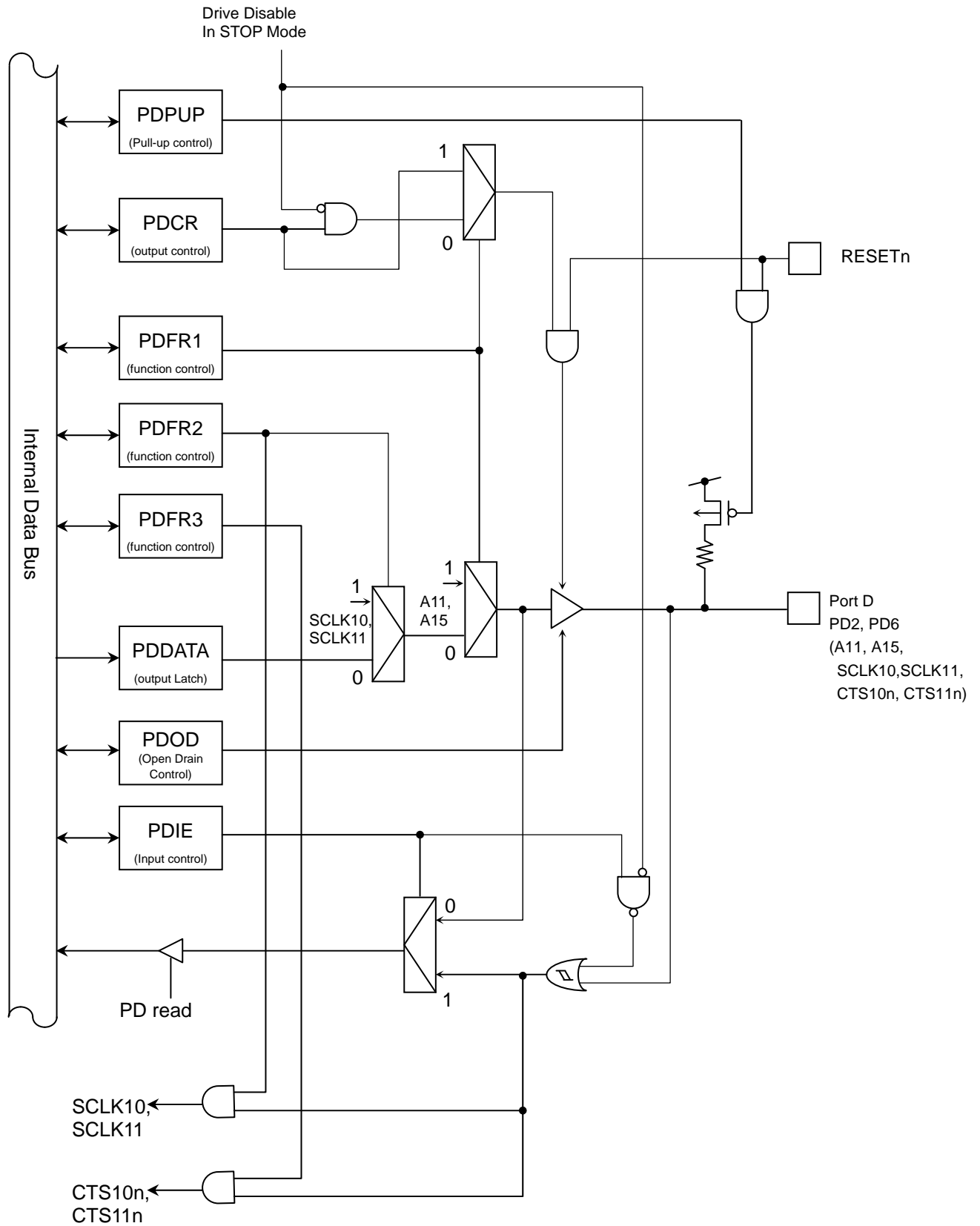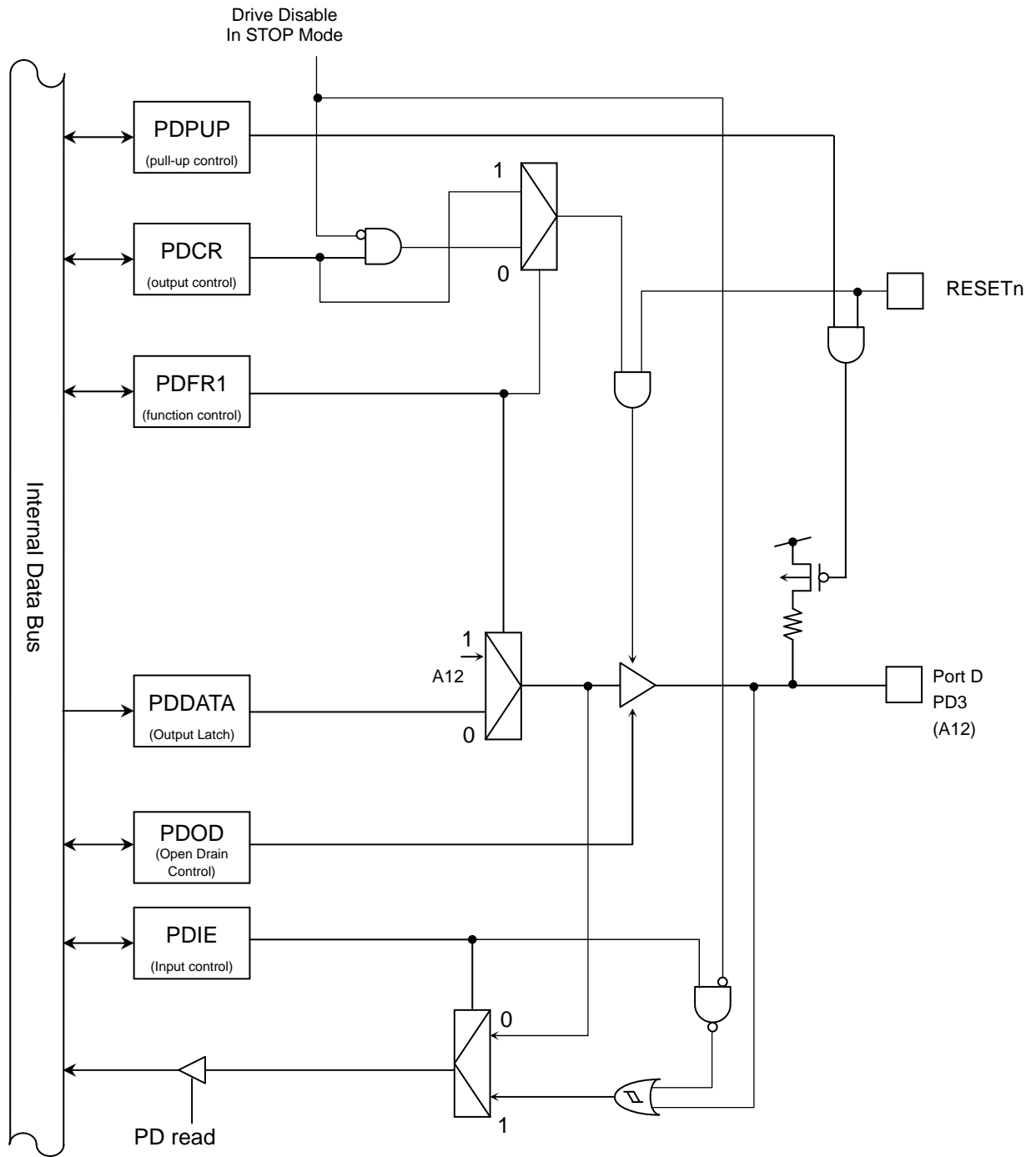| PBIE (0x400C_0138)		7	6	5	4	3	2	1	0
	Bit Symbol	PB7IE	PB6IE	PB5IE	PB4IE	PB3IE	PB2IE	PB1IE	PB0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

Fig3.6.2   PortB (PB0 to PB7)
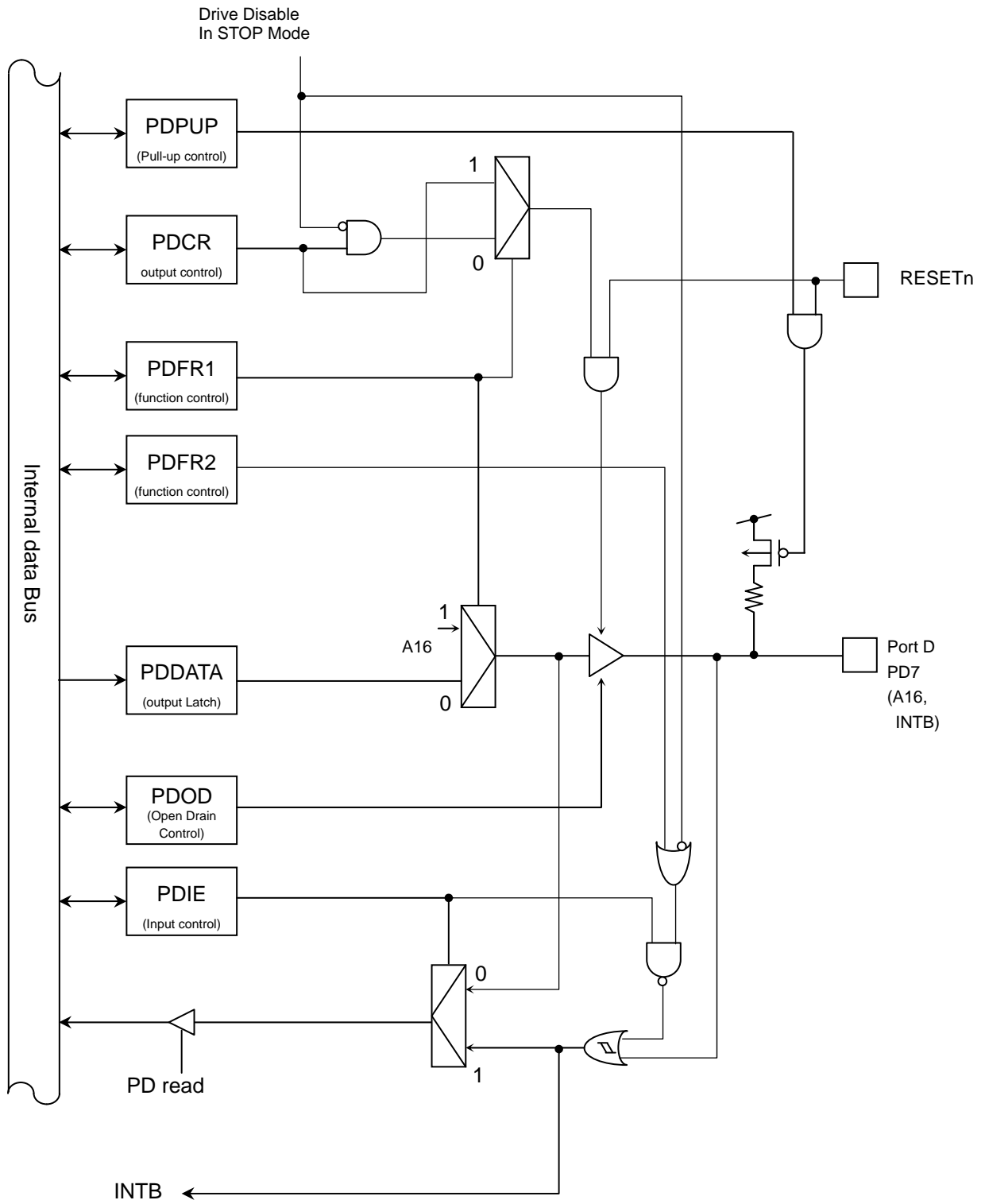
3.6.2.3 Port C

The port C is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port C performs the Adress bus(A1-A8) function and Serial interface function(UART/SIO) and the Serial bus interface function.(I2C/SIO).

Reset initializes all bits of the port C as general-purpose ports with input, output and pull-up disabled.

If this port is used for external bus function, PCCR and PCFR1 must be set to "1".

PortC register

PCDATA (0x400C_0200)		7	6	5	4	3	2	1	0
	Bit Symbol	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
	Read/Write	R/W							
	After reset	"0"							

PortC control register

PCCR (0x400C_0204)		7	6	5	4	3	2	1	0
	Bit Symbol	PC7C	PC6C	PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Output disabled 1:Output enabled							

PortC function register1

PCFR1 (0x400C_0208)		7	6	5	4	3	2	1	0
	Bit Symbol	PC6F1	PC6F1	PC5F1	PC4F1	PC3F1	PC2F1	PC1F1	PC0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: A8	0: PORT 1: A7	0: PORT 1: A6	0: PORT 1: A5	0: PORT 1: A4	0: PORT 1: A3	0: PORT 1: A2	0: PORT 1: A1

PortC function register2

PCFR2 (0x400C_020C)		7	6	5	4	3	2	1	0
	Bit Symbol	—	PC6F2	PC5F2	PC4F2	—	PC2F2	PC1F2	PC0F2
	Read/Write	R	R/W			R	R/W		
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.	0:PORT 1: SCLK9	0:PORT 1: RXD9	0:PORT 1: TXD9	"0" is read.	0:PORT 1: SCLK8	0:PORT 1: RXD8	0:PORT 1: TXD8

PortC function register3

PCFR3 (0x400C_0210)		7	6	5	4	3	2	1	0
	Bit Symbol	—	PC6F3	—	—	—	PC2F3	—	—
	Read/Write	R	R/W	R			R/W	R	
	After reset	0	0	0			0	0	
	Function	"0" is read.	0:PORT 1: CTS9n	"0" is read.			0:PORT 1: CTS8n	"0" is read.	

PortC   open drain control register

PCOD (0x400C_0228)		7	6	5	4	3	2	1	0
	Bit Symbol	PC7 OD	PC6 OD	PC5 OD	PC4 OD	PC3 OD	PC2 OD	PC1 OD	PC0OD
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain

PortC   pull-up control register

PCPUP (0x400C_022C)		7	6	5	4	3	2	1	0
	Bit Symbol	PC7UP	PC6UP	PC5UP	PC4UP	PC3UP	PC2UP	PC1UP	PC0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0:off 1:on	Pull-up 0:off 1: on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1: on	Pull-up 0:off 1: on	Pull-up 0:off 1: on	Pull-up 0:off 1: on

PortC   input enable control register

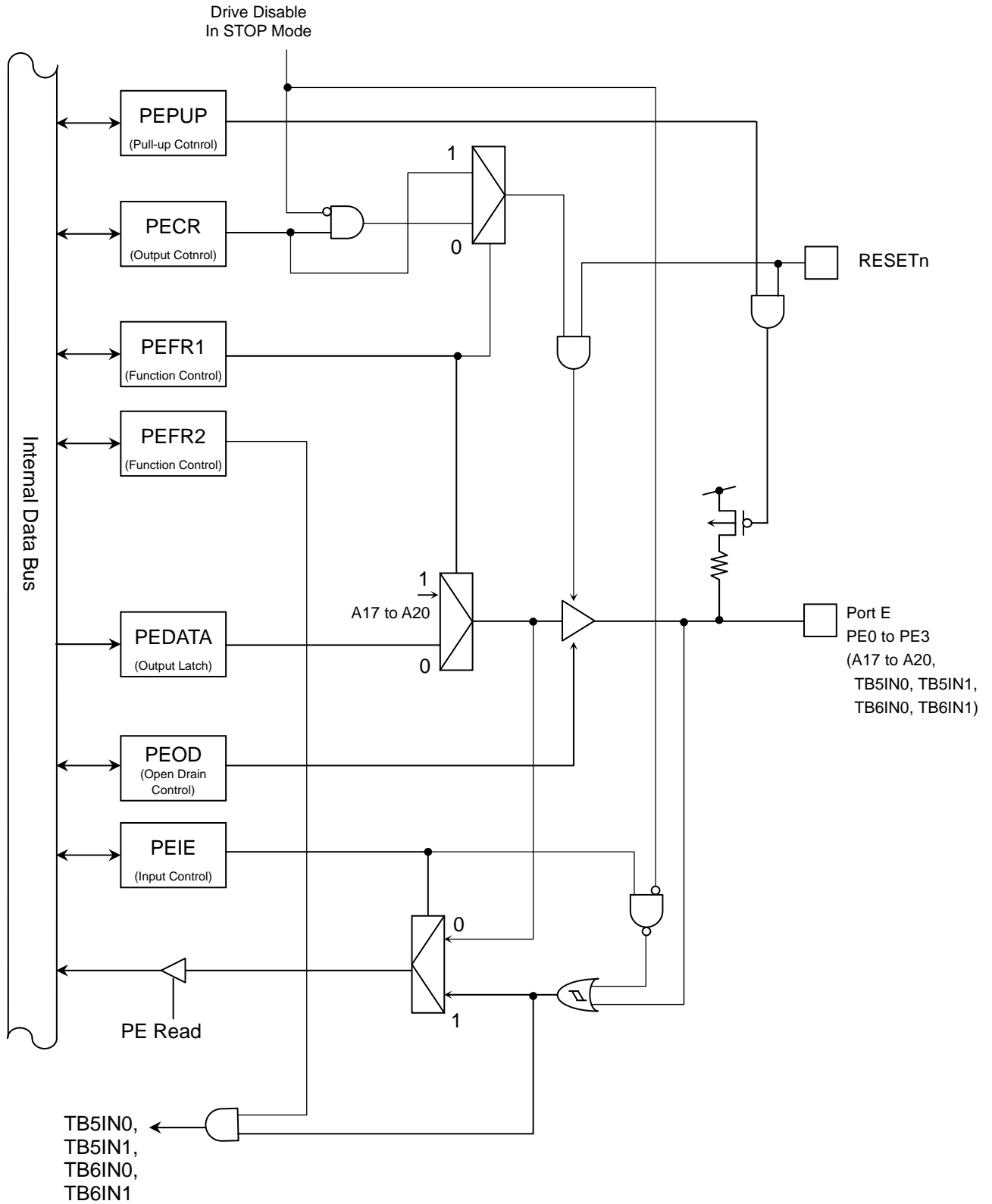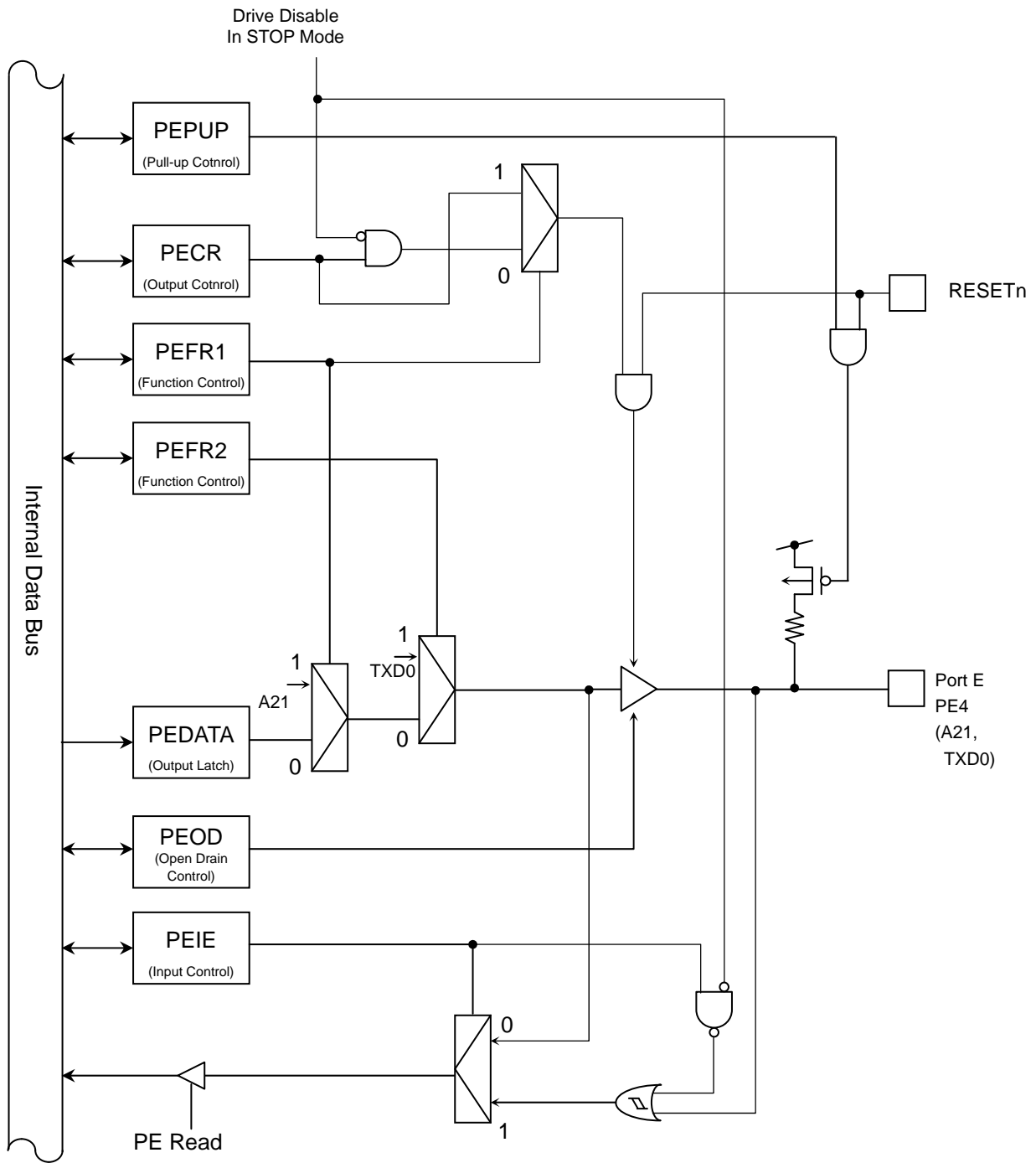| PCIE (0x400C_0238)		7	6	5	4	3	2	1	0
	Bit Symbol	PC7IE	PC6IE	PC5IE	PC4IE	PC3IE	PC2IE	PC1IE	PC0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

Fig3.6.3   PortC (PC0, PC4)

Drive Disable
In STOP Mode

PCPUP
(Pull-up Control)

PCCR
(output control)

RESETn

PCFR1
(function control)

PCFR2
(function control)

Internal Data Bus

PCDATA
(output Lactch)

A2, A6

Port C
PC1, PC5
(A2, A6,
RXD8, RXD9)

PCOD
(Open Drain
Control)

PCIE
(Input Control)

PC read

RXD8,
RXD9

Fig3.6.4   PortC (PC1, PC5)
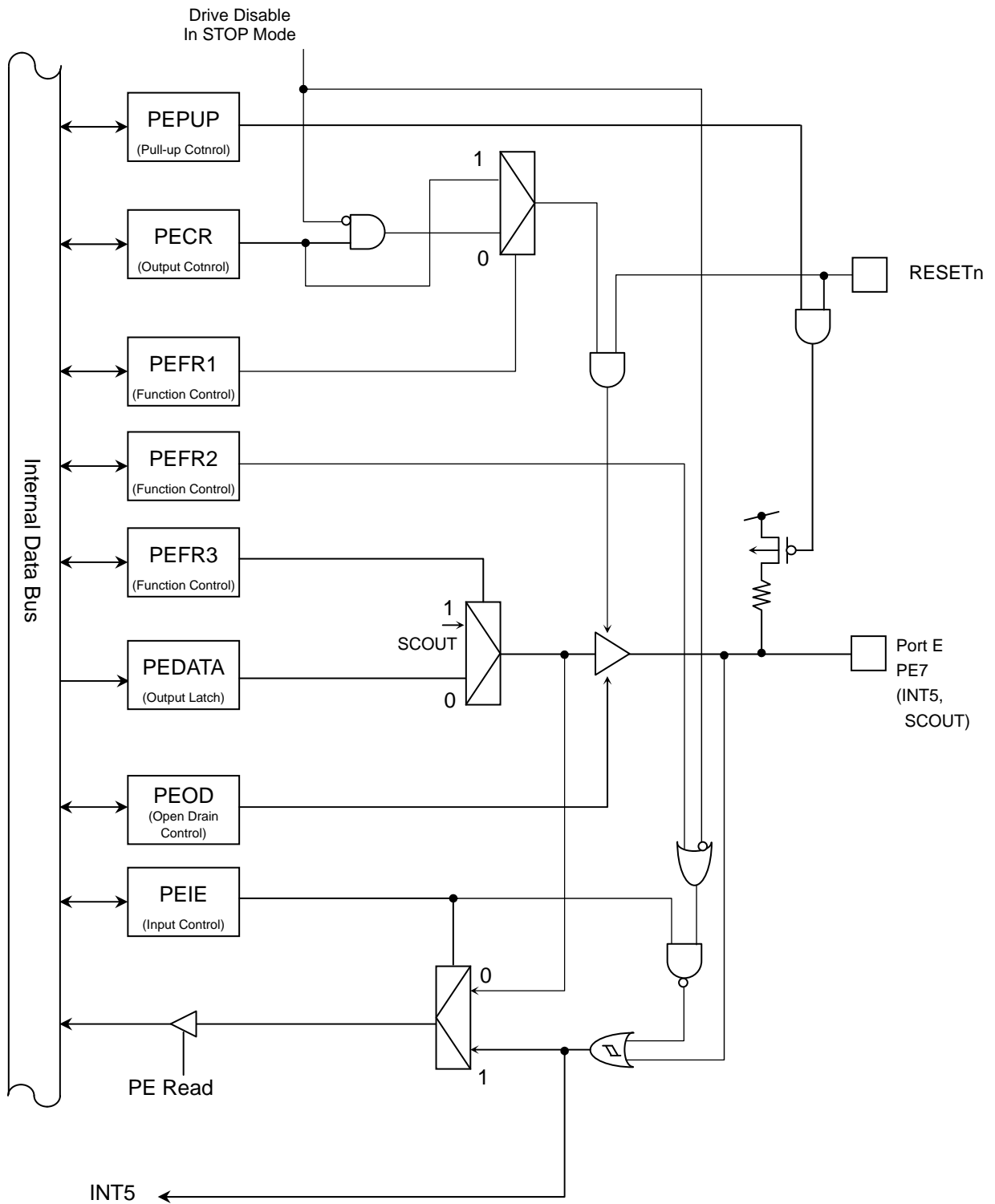
Fig3.6.5   PortC (PC2, PC6)

Fig3.6.6   PortC (PC3, PC7)

3.6.2.4 Port D

The port D is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port D performs the Adress bus(A9-A16) function and Serial interface function(UART/SIO) and the external signal interrupt function..

Reset initializes all bits of the port C as general-purpose ports with input, output and pull-up disabled. If this port is used for external bus function, PDCR and PDFR1 must be set to "1."

PortD register

PDDATA (0x400C_0300)		7	6	5	4	3	2	1	0
	Bit Symbol	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Read/Write	R/W							
	After reset	"0"							

PortD control register

PDCR (0x400C_0304)		7	6	5	4	3	2	1	0
	Bit Symbol	PD7C	PD6C	PD5C	PD4C	PD3C	PD2C	PD1C	PD0C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Output disabled 1:Output enabled							

PortD function register1

PDFR1 (0x400C_0308)		7	6	5	4	3	2	1	0
	Bit Symbol	PD7F1	PD6F1	PD5F1	PD4F1	PD3F1	PD2F1	PD1F1	PD0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: A16	0: PORT 1: A15	0: PORT 1: A14	0: PORT 1: A13	0: PORT 1: A12	0: PORT 1: A11	0: PORT 1: A10	0: PORT 1: A9

PortD function register2

PDFR2 (0x400C_030C)		7	6	5	4	3	2	1	0
	Bit Symbol	PD7F2	PD6F2	PD5F2	PD4F2	—	PD2F2	PD1F2	PD0F
	Read/Write	R/W				R	R/W		
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: INTB	0: PORT 1: SCLK11	0: PORT 1: RXD11	0: PORT 1: TXD11	"0" is read.	0: PORT 1: SCLK10	0: PORT 1: RXD10	0: PORT 1: TXD10

PortD function register3

PDFR3 (0x400C_0310)		7	6	5	4	3	2	1	0
	Bit Symbol	—	PD6F3	—	—	—	PD2F3	—	—
	Read/Write	R	R/W	R			R/W	R	
	After reset	0	0	0			0	0	
	Function	"0" is read.	0: PORT 1: CTS11n	"0" is read.			0: PORT 1: CTS10n	"0" is read.	

PortD open drain control register

PDOD (0x400C_0328)		7	6	5	4	3	2	1	0
	Bit Symbol	PD7 OD	PD6 OD	PD5 OD	PD4 OD	PD3 OD	PD2 OD	PD1 OD	PD0OD
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain

PortD pull-up control register

PDPUP (0x400C_032C)		7	6	5	4	3	2	1	0
	Bit Symbol	PD7UP	PD6UP	PD5UP	PD4UP	PD3UP	PD2UP	PD1UP	PD0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0:off 1:on	Pull-up 0:off 1: on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1: on	Pull-up 0:off 1:on	Pull-up 0:off 1:on

PortD input enable control register

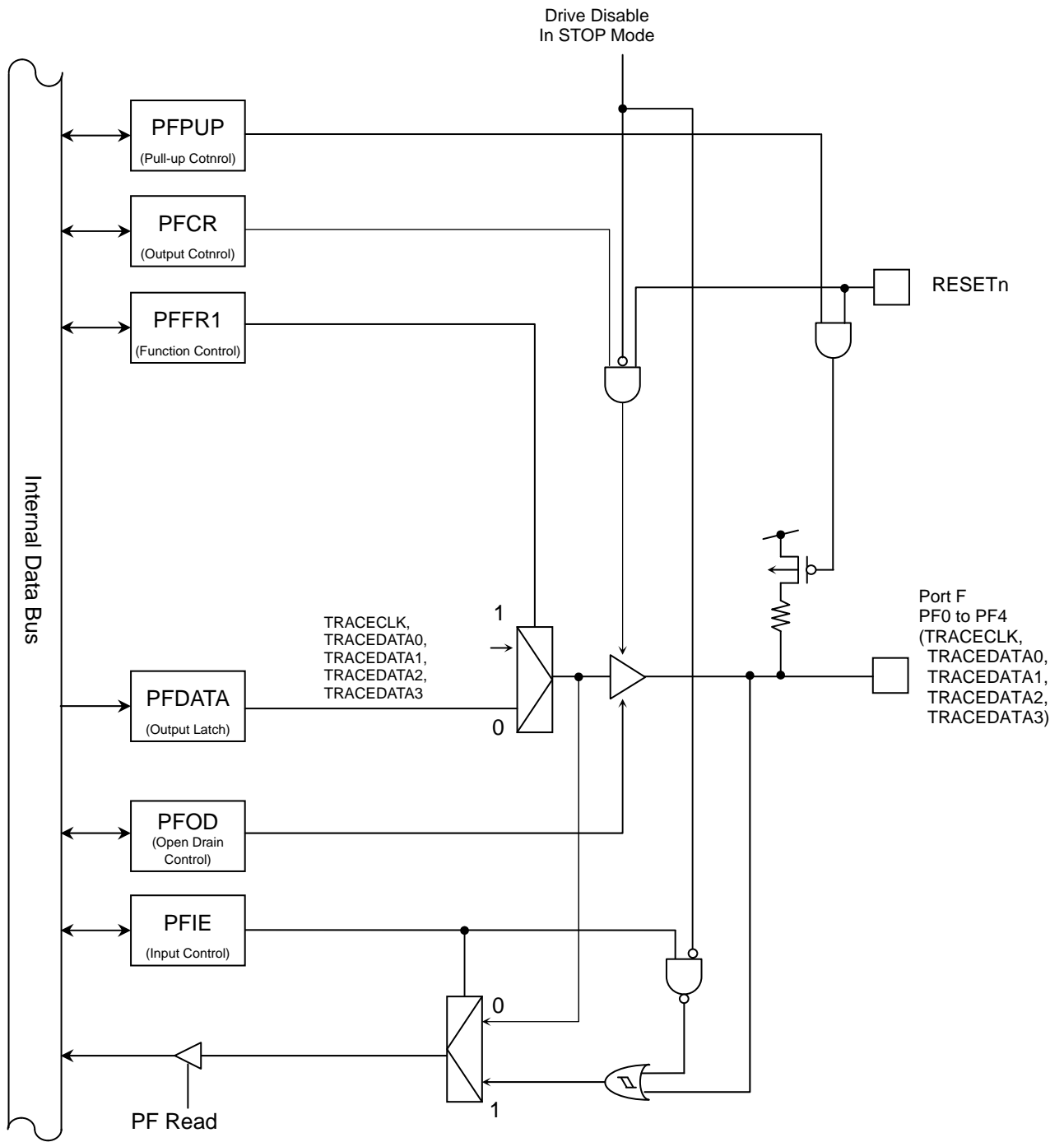| PDIE (0x400C_0338)		7	6	5	4	3	2	1	0
	Bit Symbol	PD7IE	PD6IE	PD5IE	PD4IE	PD3IE	PD2IE	PD1IE	PD0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	1	1
	Function	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

Fig3.6.7   PortD (PD0, PD4)

Fig3.6.8   PortD (PD1, PD5)

Drive Disable
In STOP Mode

PDPUP
(Pull-up control)

PDCR
(output control)

1

0

RESETn

PDFR1
(function control)

PDFR2
(function control)

PDFR3
(function control)

1

SCLK10,
SCLK11

A11,
A15

1

0

Port D
PD2, PD6
(A11, A15,
SCLK10,SCLK11,
CTS10n, CTS11n)

PDDATA
(output Latch)

0

PDOD
(Open Drain
Control)

PDIE
(Input control)

0

Internal Data Bus

PD read

1

SCLK10,
SCLK11

CTS10n,
CTS11n

Fig3.6.9   PortD (PD2, PD6)

Fig3.6.10   PortD (PD3)

Fig3.6.11   PortD (PD7)

3.6.2.5 Port E

The port E is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose port function, the port E performs the functions of the Adress bus(A17-A23) function , the serial bus interface(SIO/UART), the external interrupt input, the 16-bit timer input and the clock output.

Reset initializes all bits of the port E as general-purpose ports with input, output and pull-up disabled.

If this port is used for external bus function, PECR and PEFR1 must be set to "1."

Port E register

PEDATA (0x400C_0400)		7	6	5	4	3	2	1	0
	Bit Symbol	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
	Read/Write	R/W							
	After reset	"0"							

Port E control register

PECR (0x400C_0404)		7	6	5	4	3	2	1	0
	Bit Symbol	PE7C	PE6C	PE5C	PE4C	PE3C	PE2C	PE1C	PE0C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Output disabled 1: Output enabled							

Port E function register 1

| PEFR1 (0x400C_0408)		7	6	5	4	3	2	1	0
	Bit Symbol	PE7F1	PE6F1	PE5F1	PE4F1	PE3F1	PE2F1	PE1F1	PE0F0
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Always write "0".	0: PORT 1: A23	0: PORT 1: A22	0: PORT 1: A21	0: PORT 1: A20	0: PORT 1: A19	0: PORT 1: A18	0: PORT 1: A17

Port E function register 2

PEFR2 (0x400C_040C)		7	6	5	4	3	2	1	0
	Bit Symbol	PE7F2	PE6F2	PE5F2	PE4F2	PE3F2	PE2F2	PE1F2	PE0F2
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: INT5	0: PORT 1: SCLK0	0: PORT 1: RXD0	0: PORT 1: TXD0	0: PORT 1: TB6IN1	0: PORT 1: TB6IN0	0: PORT 1: TB5IN1	0: PORT 1: TB5IN0

Port E function register 3

PEFR3 (0x400C_0410)		7	6	5	4	3	2	1	0
	Bit Symbol	PE7F3	PE6F3	—	—	—	—	—	—
	Read/Write	R/W		R					
	After reset	0	0	0	0	0			
	Function	0: PORT 1: SCOUT	0: PORT 1: CTS0n	Always write "0".	Always write "0".	"0" is read.			

Port E open drain control register

		7	6	5	4	3	2	1	0
PEOD (0x400C_0428)	Bit Symbol	PE7 OD	PE6 OD	PE5 OD	PE4 OD	PE3 OD	PE2 OD	PE1 OD	PE0OD
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain

Port E pull-up control register

		7	6	5	4	3	2	1	0
PEPUP (0x400C_042C)	Bit Symbol	PE7UP	PE6UP	PE5UP	PE4UP	PE3UP	PE2UP	PE1UP	PE0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on

Port E input enable control register

		7	6	5	4	3	2	1	0
PEIE (0x400C_0438)	Bit Symbol	PE7IE	PE6IE	PE5IE	PE4IE	PE3IE	PE2IE	PE1IE	PE0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

Fig. 3.6.12 Port E (PE0 to PE3)

Fig. 3.6.13 Port E (PE4)
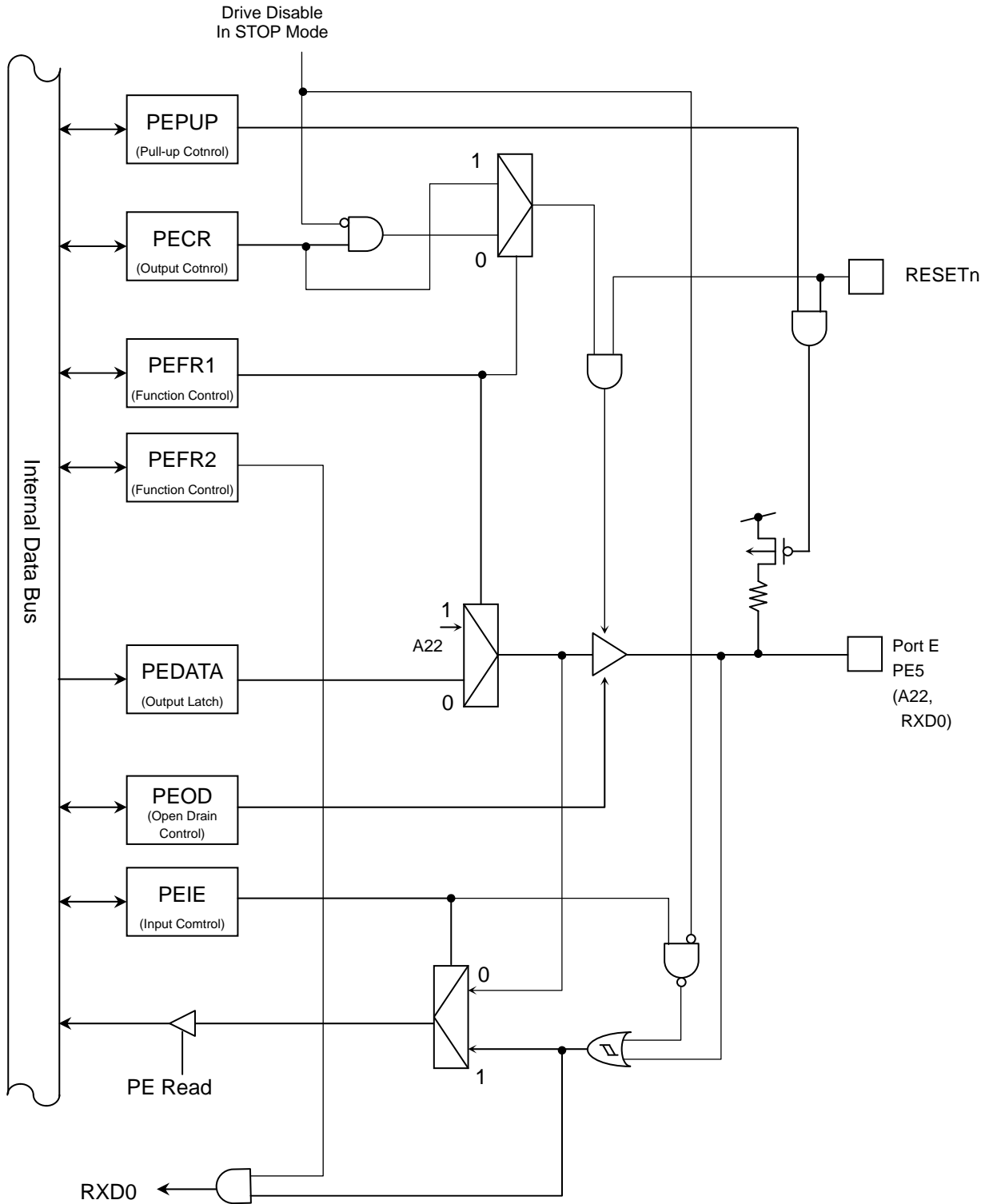
Fig. 3.6.14 Port E (PE5)
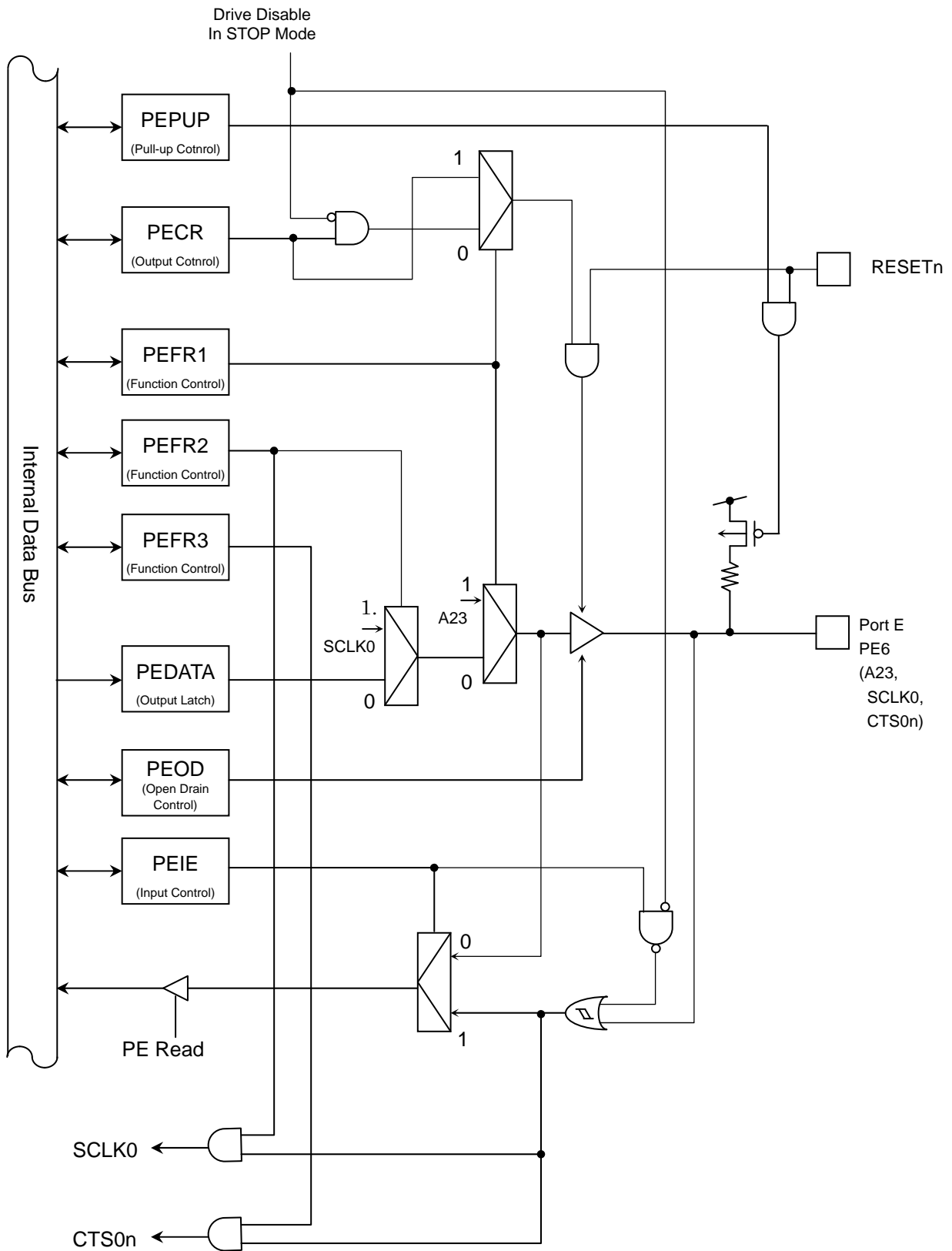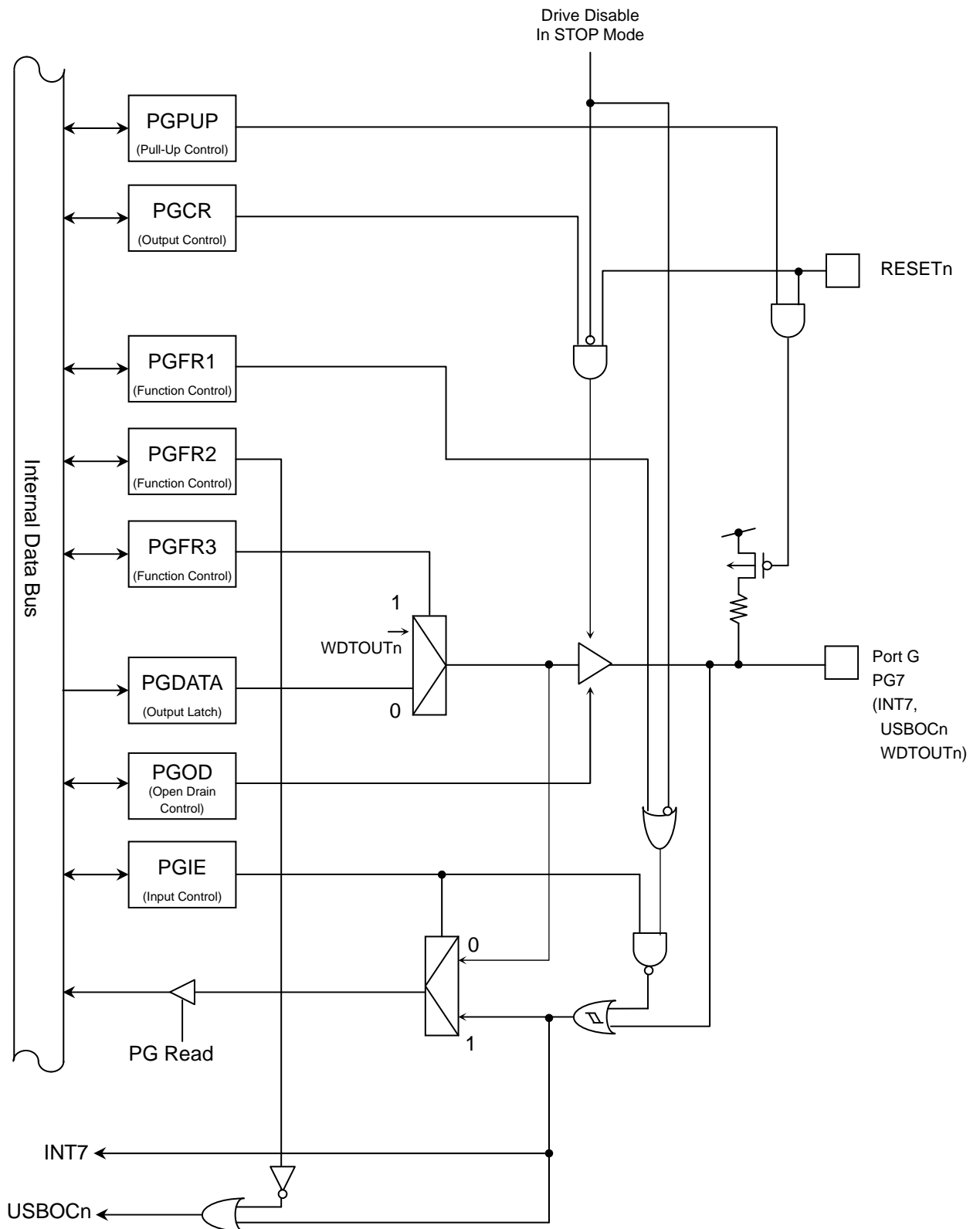
Fig. 3.6.15 Port E (PE6)

Fig. 3.6.16 Port E (PE7)

3.6.2.6 Port F

The port F is a general-purpose, 5-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port F performs the debug trace output function.

Reset initializes all bits of the port F as general-purpose ports with input, output and pull-up disabled.

Port F register

PFDATA (0x400C_0500)		7	6	5	4	3	2	1	0
	Bit Symbol	—	—	—	PF4	PF3	PF2	PF1	PF0
	Read/Write	R			R/W				
	After reset	"0" is read.			"0"				

Port F control register

PFCR (0x400C_0504)		7	6	5	4	3	2	1	0
	Bit Symbol	—	—	—	PF4C	PF3C	PF2C	PF1C	PF0C
	Read/Write	R			R/W				
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.			0: Output disabled 1: Output enabled				

Port F function register 1

PFFR1 (0x400C_0508)		7	6	5	4	3	2	1	0
	Bit Symbol	—	—	—	PF4F1	PF3F1	PF2F1	PF1F1	PF0F1
	Read/Write	R			R/W				
	After reset	0			0	0	0	0	0
	Function	"0" is read.			0: PORT 1: TRACE DATA3	0: PORT 1: TRACE DATA2	0: PORT 1: TRACE DATA1	0: PORT 1: TRACE DATA0	0: PORT 1: TRACE CLK

Port F open drain control register

PFOD (0x400C_0528)		7	6	5	4	3	2	1	0
	Bit Symbol	—	—	—	PF4 OD	PF3 OD	PF2 OD	PF1 OD	PF0OD
	Read/Write	R			R/W				
	After reset	0			0	0	0	0	0
	Function	"0" is read.			0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain

Port E pull-up control register

PFPUP (0x400C_052C)		7	6	5	4	3	2	1	0
	Bit Symbol	—	—	—	PF4UP	PF3UP	PF2UP	PF1UP	PF0UP
	Read/Write	R			R/W				
	After reset	0			0	0	0	0	0
	Function	"0" is read.			Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on

Port F input enable control register

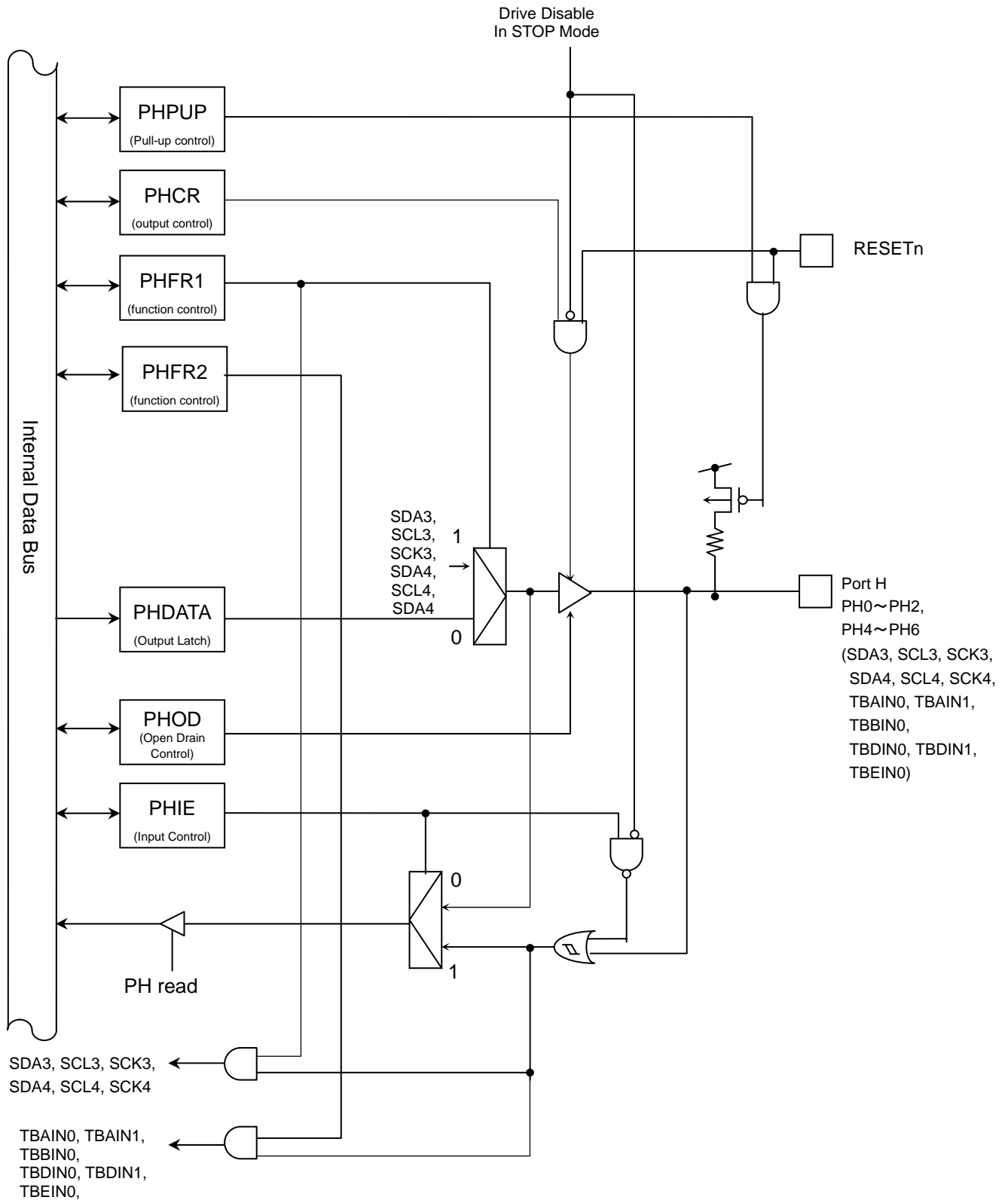| PFIE (0x400C_0538)		7	6	5	4	3	2	1	0
	Bit Symbol	—	—	—	PF4IE	PF3IE	PF2IE	PF1IE	PF0IE
	Read/Write	R			R/W				
	After reset	0			0	0	0	0	0
	Function	"0" is read.			Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

Fig. 3.6.17 Port F (PF0 to PF4)

3.6.2.7 Port G

The port G is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose port function, the port G performs the functions of the serial bus interface(I2C/SIO), the external interrupt input, the 16-bit timer input , the chip select function, the watchdog timer output, the USBPON function and the USB overcurrent frunction.

Reset initializes all bits of the port G as general-purpose ports with input, output and pull-up disabled.

Port G register

PGDATA (0x400C_0600)		7	6	5	4	3	2	1	0
	Bit Symbol	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
	Read/Write	R/W							
	After reset	"0"							

Port G control register

PGCR (0x400C_0604)		7	6	5	4	3	2	1	0
	Bit Symbol	PG7C	PG6C	PG5C	PG4C	PG3C	PG2C	PG1C	PG0C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Output disabled 1: Output enabled							

Port G function register 1

PGFR1 (0x400C_0608)		7	6	5	4	3	2	1	0
	Bit Symbol	PG7F1	PG6F1	PG5F1	PG4F1	PG3F1	PG2F1	PG1F1	PG0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: INT7	0: PORT 1: SCK2	0: PORT 1: SCL2	0: PORT 1: SDA2	0: PORT 1: INT6	0: PORT 1: SCK1	0: PORT 1: SCL1	0: PORT 1: SDA1

Port G function register 2

PGFR2 (0x400C_060C)		7	6	5	4	3	2	1	0
	Bit Symbol	PG7F2	PG6F2	PG5F2	PG4F2	—	—	PG1F2	PG0F2
	Read/Write	R/W		R/W		R		R/W	
	After reset	0	0	0	0	0		0	0
	Function	0: PORT 1: USBOCn	0: PORT 1: USBPON	0: PORT 1: TB9IN1	0: PORT 1: TB9IN0	"0" is read.		0: PORT 1: TB7IN1	0: PORT 1: TB7IN0

Port G function register 3

PGFR3 (0x400C_0610)		7	6	5	4	3	2	1	0
	Bit Symbol	PG7F3	PG6F3	—	—	PG3F3	PG2F3	—	—
	Read/Write	R/W	R/W	R		R/W	R/W	R	
	After reset	0	0	0		0	0	0	
	Function	0: PORT 1:WDTOUTn	0: PORT 1:CS3n	"0" is read.		0: PORT 1:CS1n	0: PORT 1:CS0n	"0" is read.	

Port G open drain control register

PGOD (0x400C_0628)		7	6	5	4	3	2	1	0
	Bit Symbol	PG7 OD	PG6 OD	PG5 OD	PG4 OD	PG3 OD	PG2 OD	PG1 OD	PG0OD
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain

Port G pull-up control register

PGPUP (0x400C_062C)		7	6	5	4	3	2	1	0
	Bit Symbol	PG7UP	PG6UP	PG5UP	PG4UP	PG3UP	PG2UP	PG1UP	PG0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on

Port G input enable control register

PGIE (0x400C_0638)		7	6	5	4	3	2	1	0
	Bit Symbol	PG7IE	PG6IE	PG5IE	PG4IE	PG3IE	PG2IE	PG1IE	PG0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

Fig. 3.6.18 Port G (PG0, PG1, PG4, PG5)
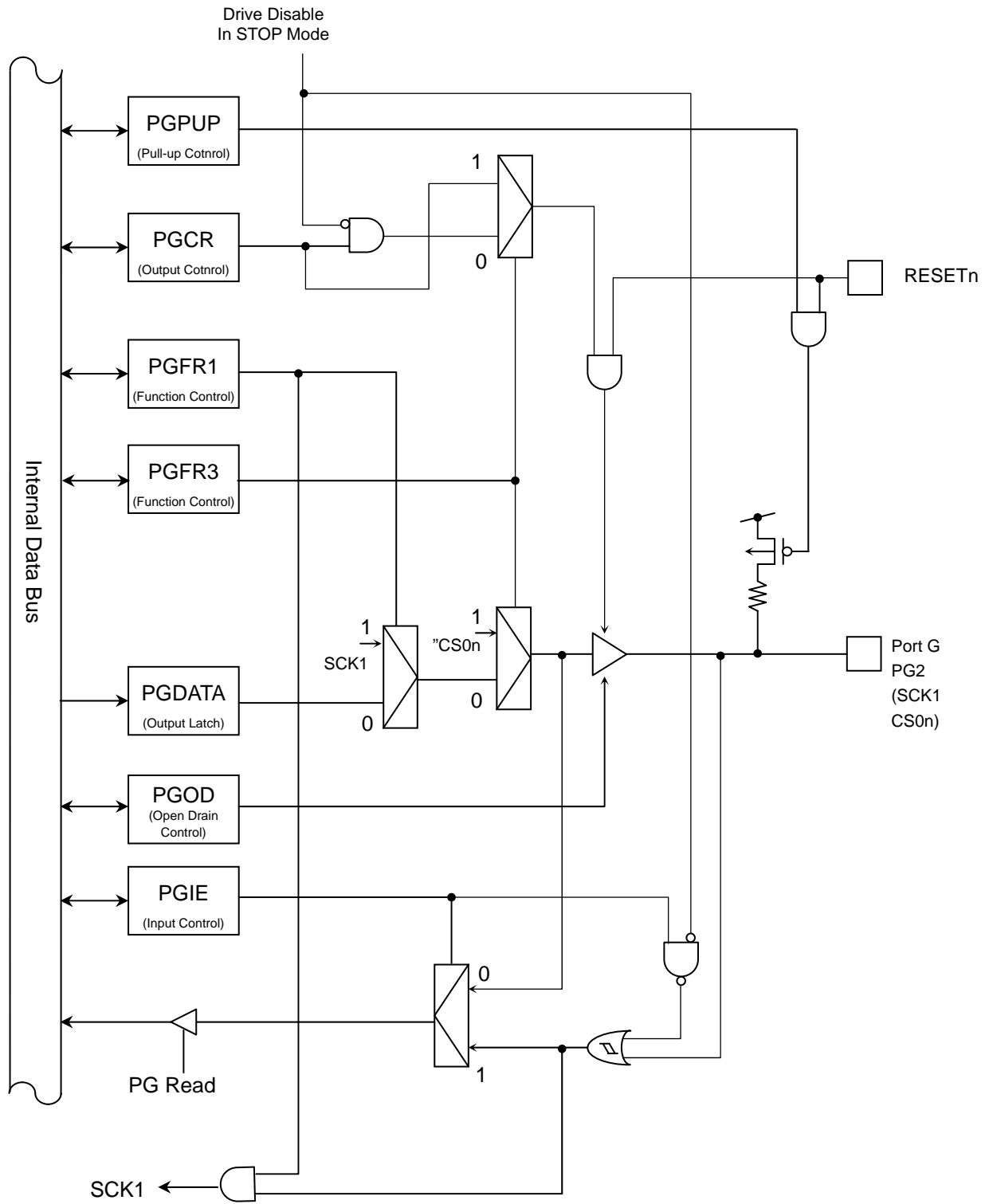
Fig. 3.6.19 Port G (PG2)
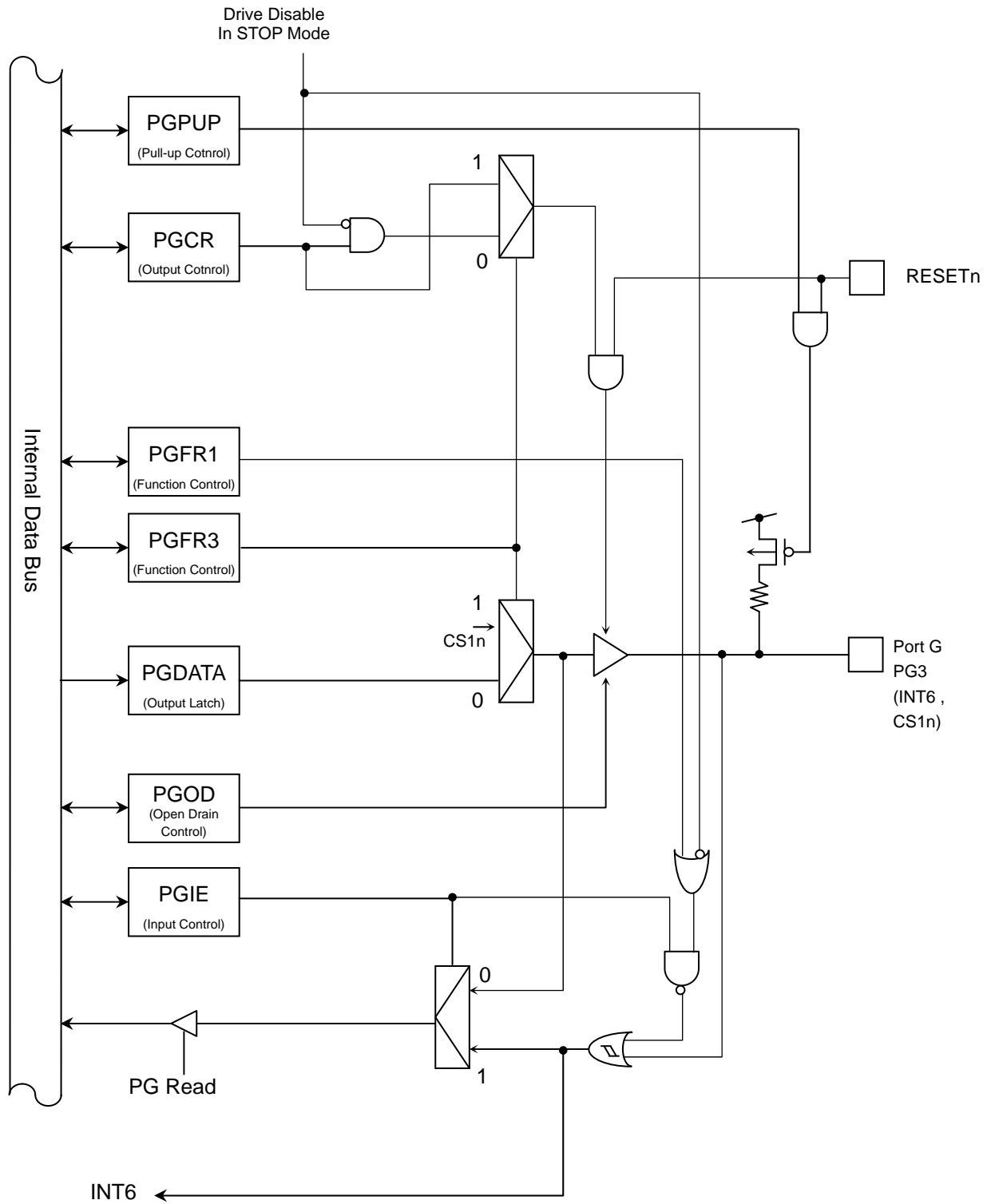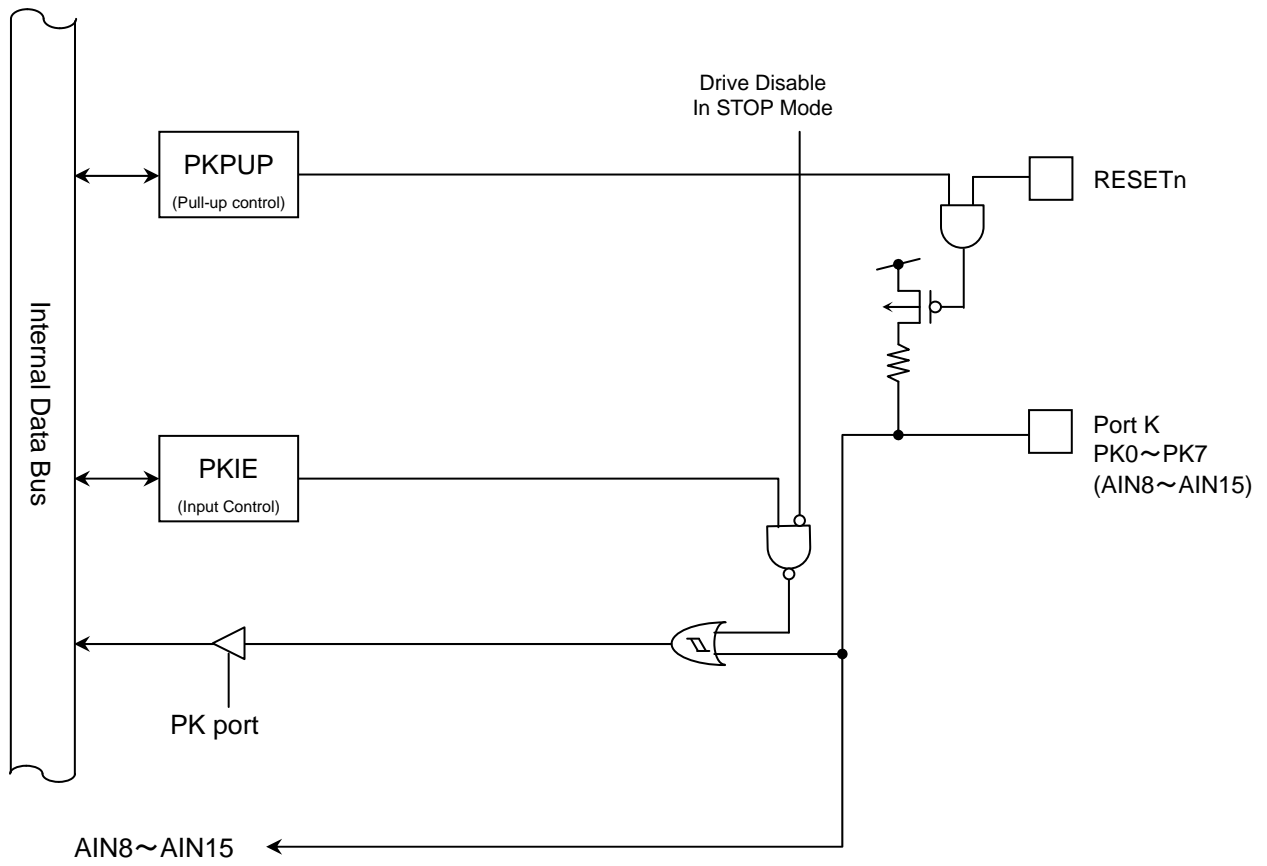
Fig. 3.6.20 Port G (PG3)
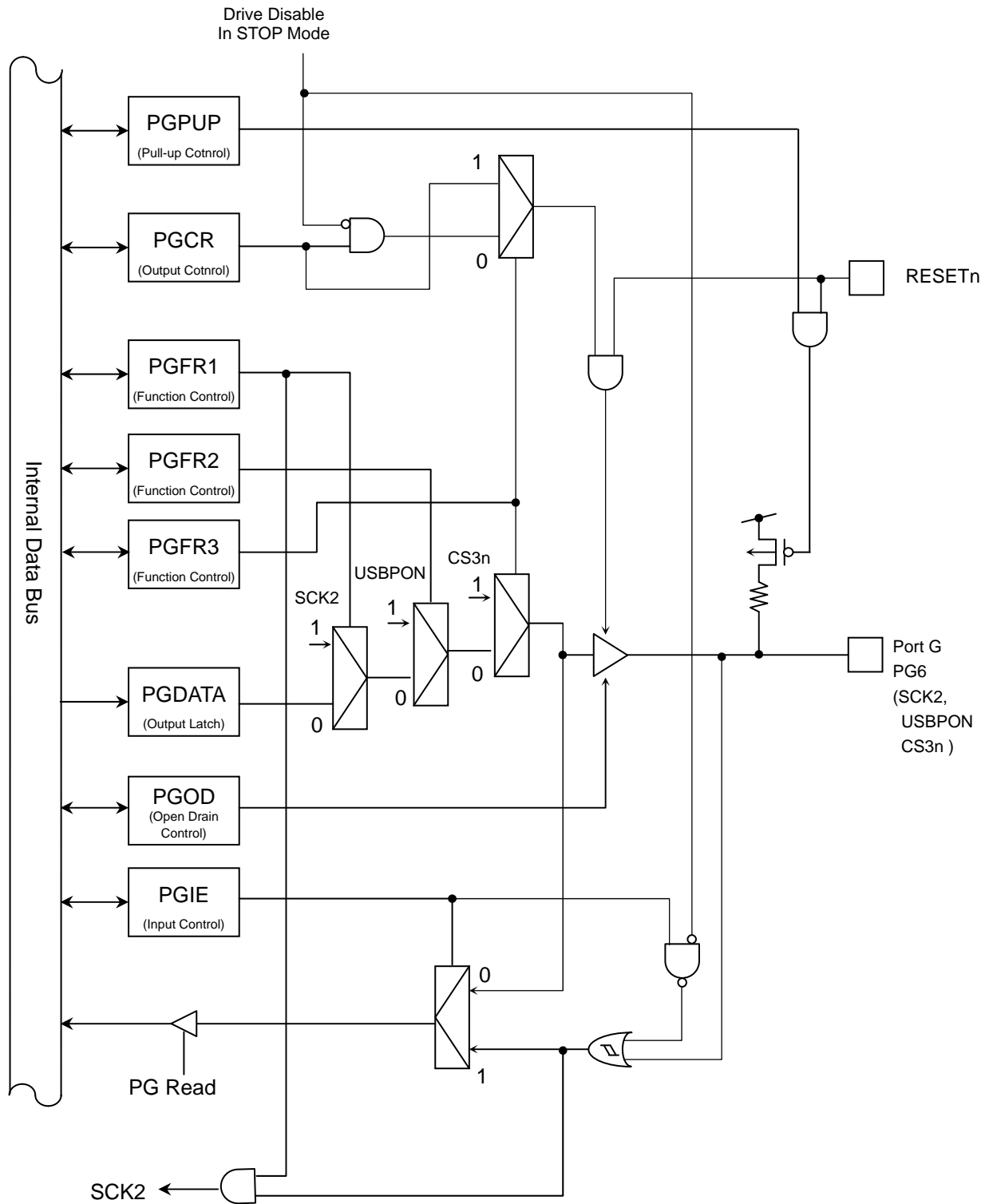
Fig. 3.6.21 Port G (PG6)

Fig. 3.6.22 Port G (PG7)

3.6.2.8 Port H

The port H is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port H performs the serial bus interface function(I2C/SIO),External signal interrupt function and 16bit timer input function.

Reset initializes all bits of the port H as general-purpose ports with input, output and pull-up disabled.

PortH register

PHDATA (0x400C_0700)		7	6	5	4	3	2	1	0
	Bit Symbol	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
	Read/Write	R/W							
	After reset	"0"							

PortH control register

PHCR (0x400C_0704)		7	6	5	4	3	2	1	0
	Bit Symbol	PH7C	PH6C	PH5C	PH4C	PH3C	PH2C	PH1C	PH0C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Output disabled 1:Output enabled							

PortH function register1

PHFR1 (0x400C_0708)		7	6	5	4	3	2	1	0
	Bit Symbol	PH7F1	PH6F1	PH5F1	PH4F1	PH3F1	PH2F1	PH1F1	PH0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: INTD	0: PORT 1: SCK4	0: PORT 1: SCL4	0: PORT 1: SDA4	0: PORT 1: INTC	0: PORT 1: SCK3	0: PORT 1: SCL3	0: PORT 1: SDA3

PortH function register2

PHFR2 (0x400C_070C)		7	6	5	4	3	2	1	0
	Bit Symbol	PH7F2	PH6F2	PH5F2	PH4F2	PH3F2	PH2F2	PH1F2	PH0F2
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: TBEIN1	0: PORT 1: TBEIN0	0: PORT 1: TBDIN1	0: PORT 1: TBDIN0	0: PORT 1: TBBIN1	0: PORT 1: TBBIN0	0: PORT 1: TBAIN1	0: PORT 1: TBAIN0

PortH   open drain control register

		7	6	5	4	3	2	1	0
PHOD (0x400C_0728)	Bit Symbol	PH7 OD	PH6 OD	PH5 OD	PH4 OD	PH3 OD	PH2 OD	PH1 OD	PH0OD
	Read/Write					R/W			
	After reset	0	0	0	0	0	0	0	0
	Function	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain

PortH   pull-up control register

		7	6	5	4	3	2	1	0
PHPUP (0x400C_072C)	Bit Symbol	PH7UP	PH6UP	PH5UP	PH4UP	PH3UP	PH2UP	PH1UP	PH0UP
	Read/Write					R/W			
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on

PortH   input enable control register

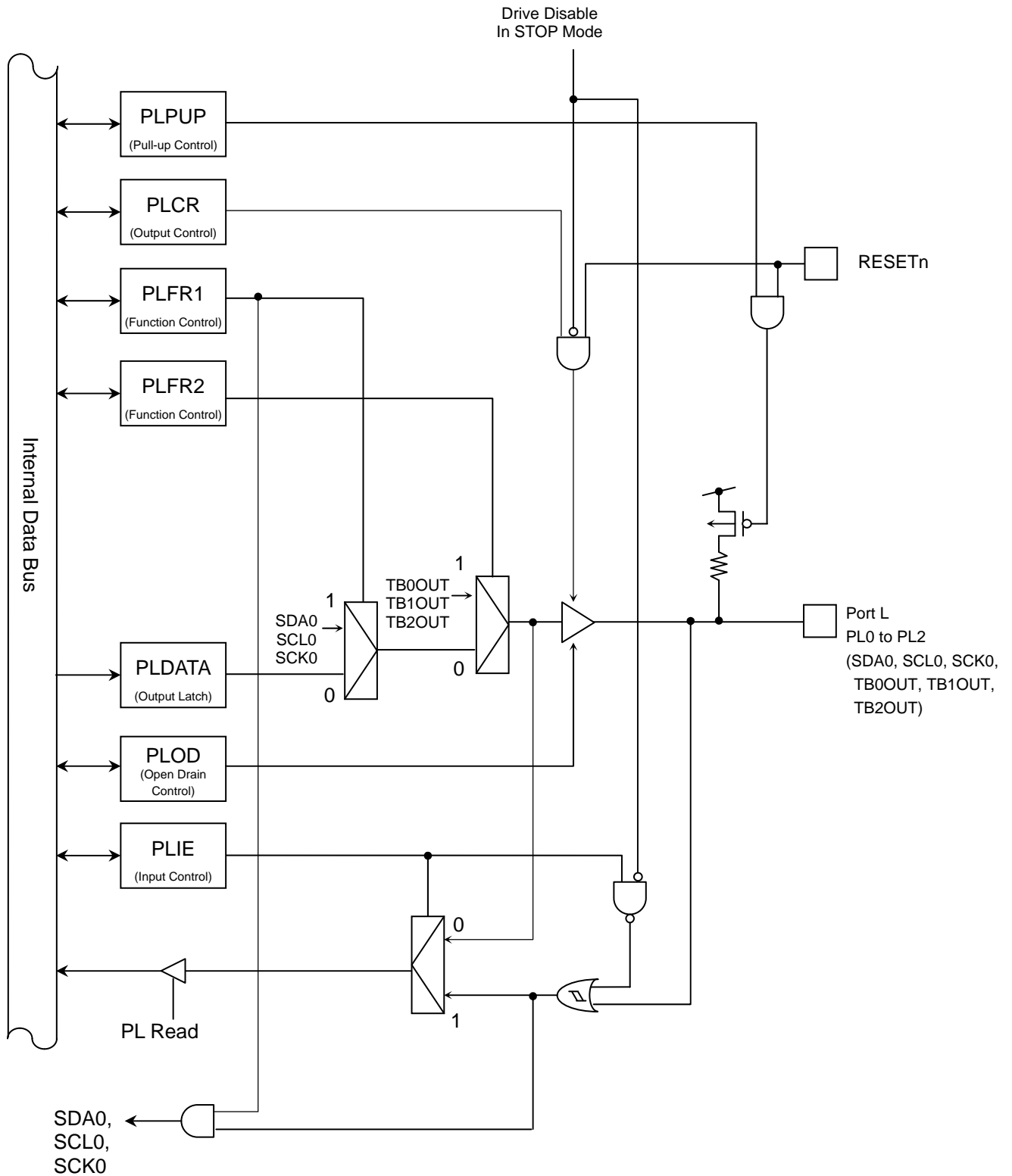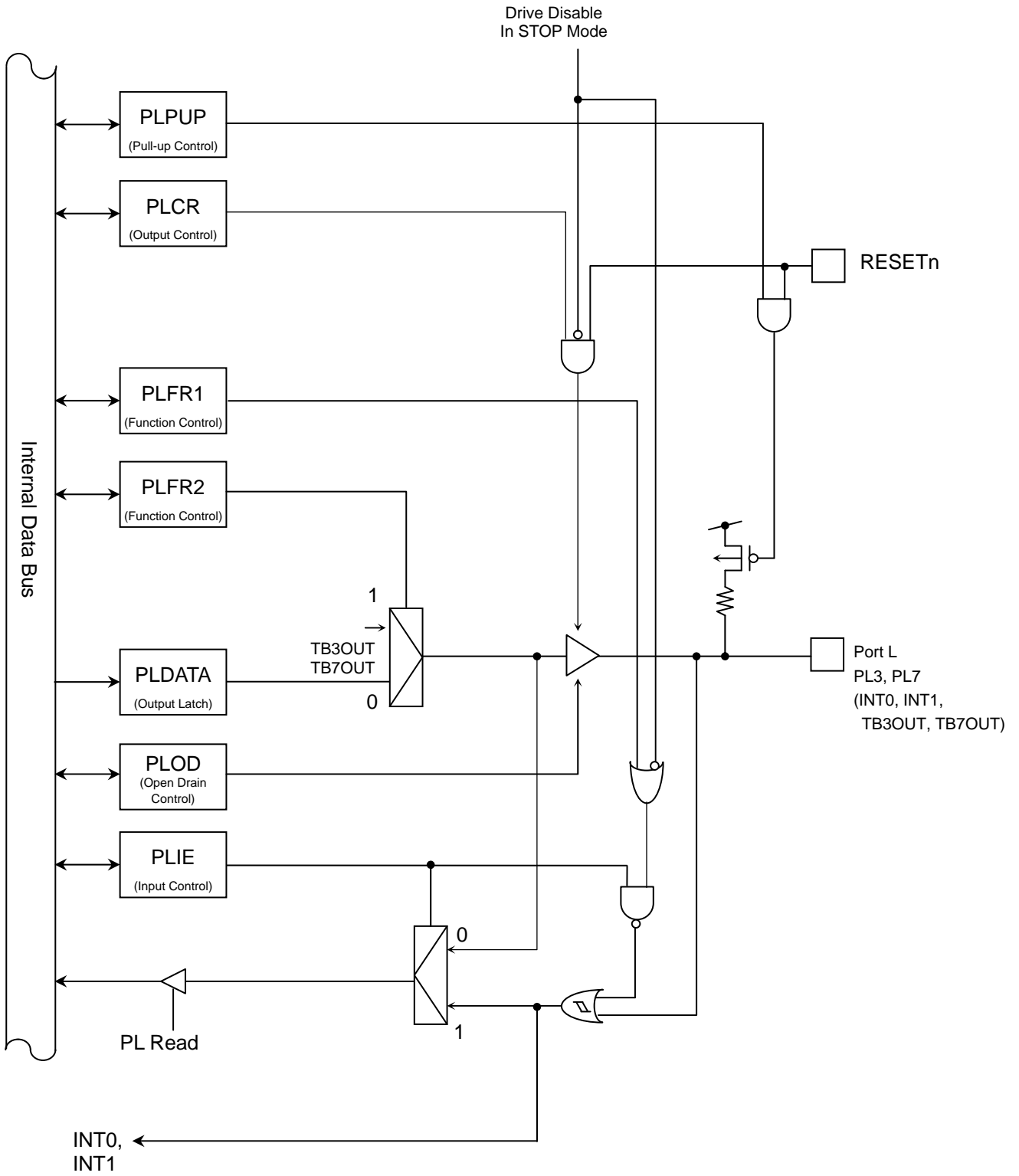|		7	6	5	4	3	2	1	0
PHIE (0x400C_0738)	Bit Symbol	PH7IE	PH6IE	PH5IE	PH4IE	PH3IE	PH2IE	PH1IE	PH0IE
	Read/Write					R/W			
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

Fig3.6.23   PortH (PH0～PF2, PH4～PH6)
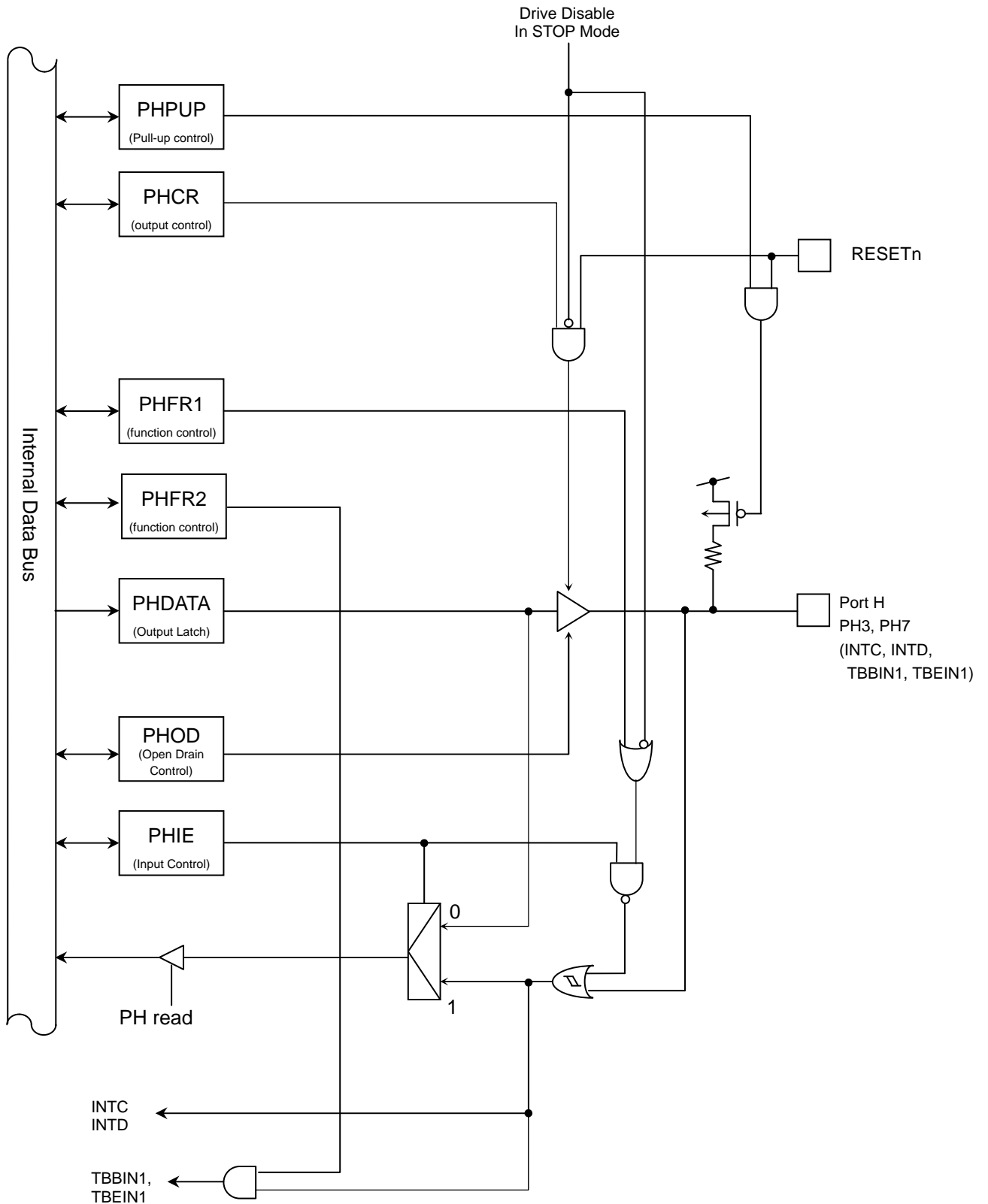
Fig3.6.24PortH (PH3, P7)

3.6.2.9 Port I

The port I is a general-purpose, 2-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose port function, the port I performs the external interrupt input.

While "0" inputs RESET pin, PI0 enables input and pull-up. During a reset, "Low level" should not be input to PI0.

Reset initializes all bits of the port E as general-purpose ports with input, output and pull-up disabled.

PI1 is the N-ch open drain state regardless PIOD register condition.

Port I register

PIDATA (0x400C_0800)		7	6	5	4	3	2	1	0
	Bit Symbol	—	—	—	—	—	—	PI1	PI0
	Read/Write	R/W						R/W	
	After reset	Always write "0".						"0"	

Port I control register

PICR (0x400C_0804)		7	6	5	4	3	2	1	0
	Bit Symbol	—	—	—	—	—	—	PI1C	PI0C
	Read/Write	R/W						R/W	
	After reset	0						0	0
	Function	Always write "0".						0: Output disabled 1: Output enabled	

Port I open drain control register

PIOD (0x400C_0828)		7	6	5	4	3	2	1	0
	Bit Symbol	—	—	—	—	—	—	—	PI0OD
	Read/Write	R/W						R	R/W
	After reset	0						0	0
	Function	Always write "0".						"0" is read. (note)	0:CMOS 1:Open drain

(Note) Port I1 is usually open drain output states, but "0" is read from PIOD<bit1>.

Port I pull-up control register

		7	6	5	4	3	2	1	0
PIPUP	Bit Symbol	—	—	—	—	—	—	—	PI0UP
(0x400C_082C)	Read/Write	R/W						R	R/W
	After reset	0						0	0
	Function	Always write "0".						"0" is read.	Pull-up 0: off 1: on

Port I input enable control register

		7	6	5	4	3	2	1	0
PIIE	Bit Symbol	—	—	—	—	—	—	PI1IE	PI0IE
(0x400C_0838)	Read/Write	R/W							
	After reset	0						0	0
	Function	Always write "0".						Input 0:disabled 1:enabled	Input 0:disabled 1:enabled
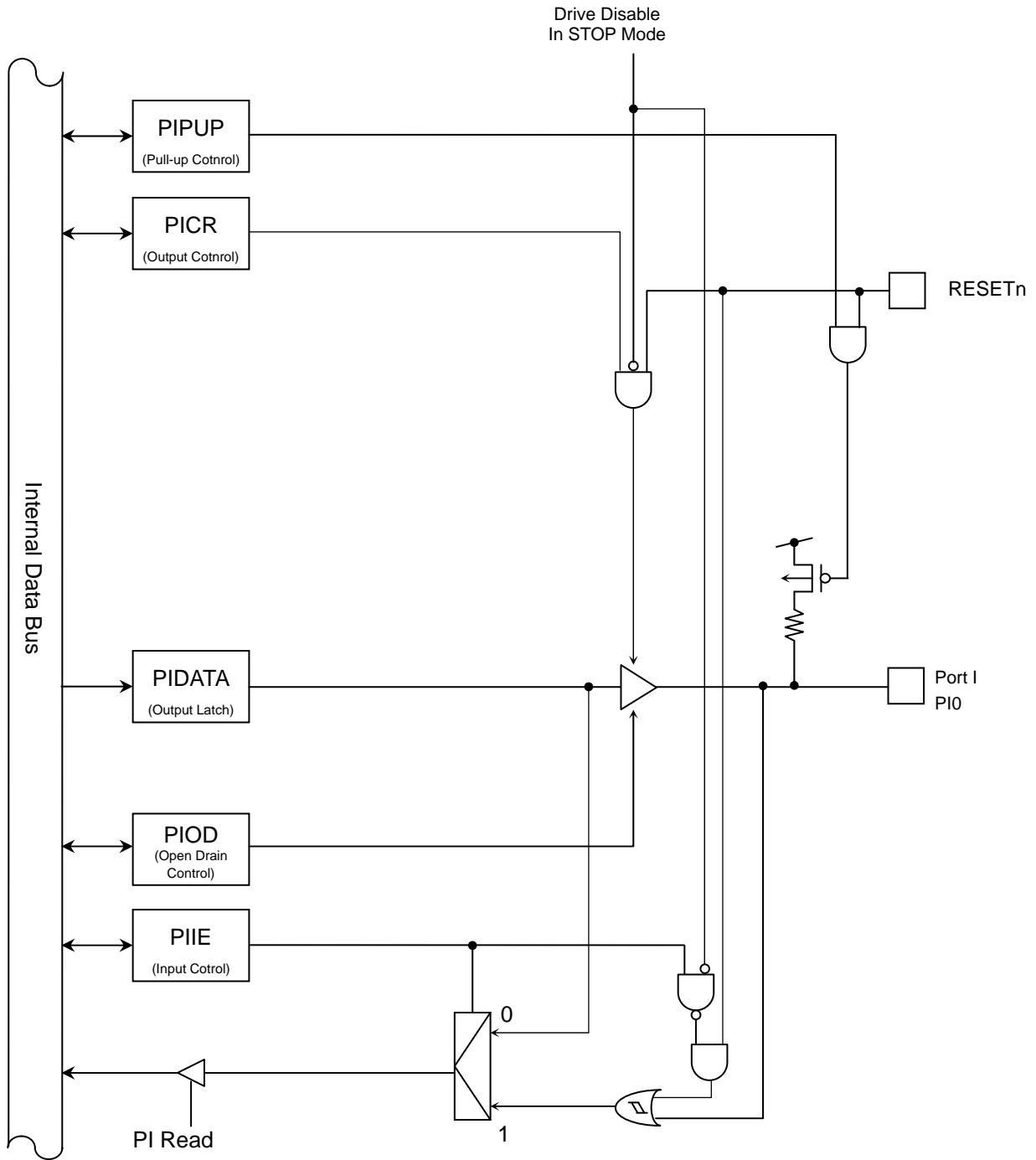
Fig. 3.6.25 Port I (PI0)
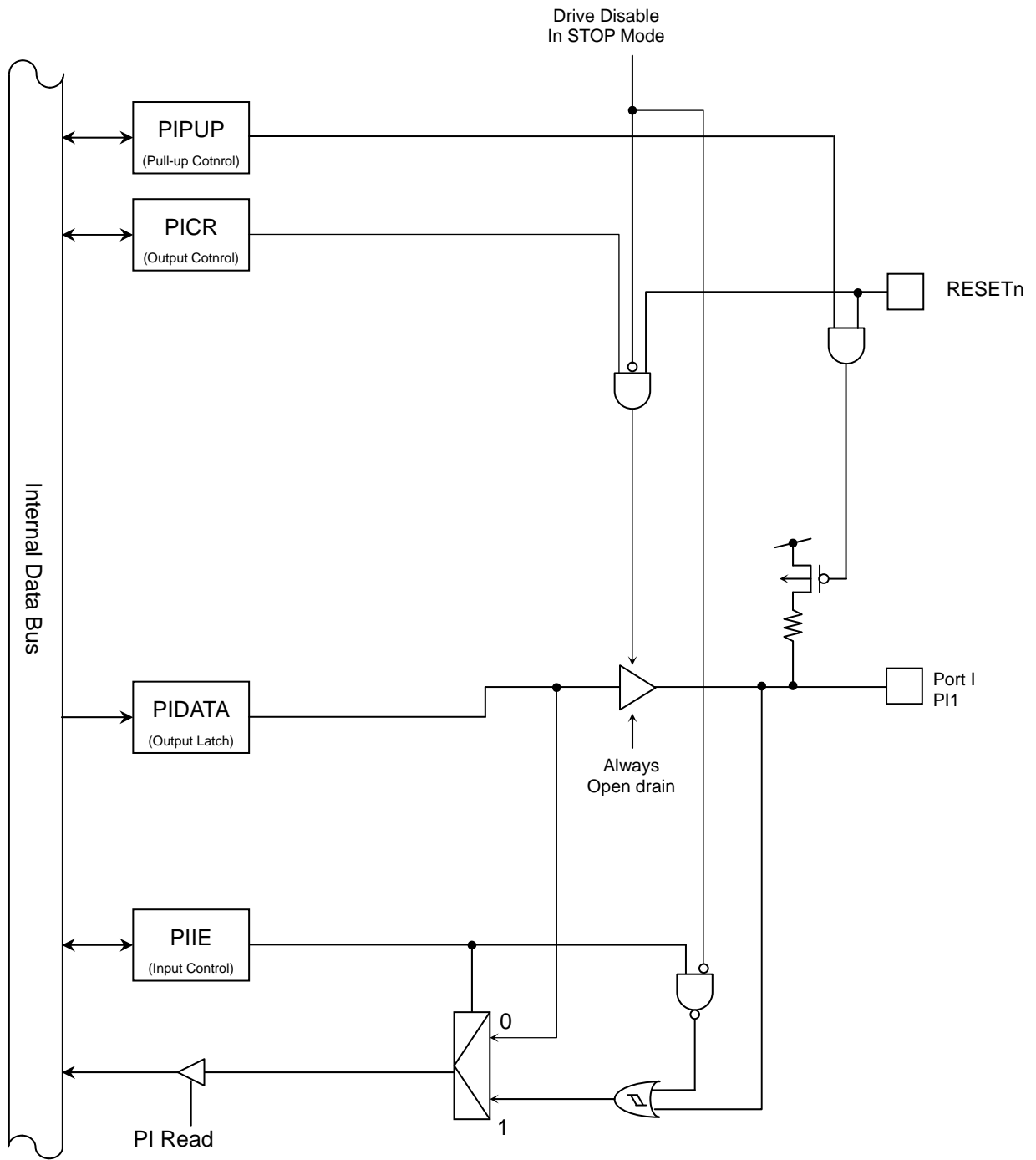
Fig. 3.6.26 Port I (PI1)

3.6.2.10   Port J

The port J is an 8-bit input port. Besides the general-purpose input function, the port J receives the analog input of the A/D converter, the key on wakeup function and A/D trigger input function.

Reset initializes all bits of the port J as general-purpose input ports with input and pull-up disabled.

Set the input enable control register when you use the port J as input ports.

(note)	Unlsee you use all the bits of Port J and Port K as analog input pins, conversion accuracy may be reduced. Be sure to verify that this causes no problem on your system.

Port J register

PJDATA (0x400C_0900)		7	6	5	4	3	2	1	0
	Bit Symbol	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
	Read/Write	R/W							
	After reset	"0"							

Port J functionl register 2

PJFR2 (0x400C_090C)		7	6	5	4	3	2	1	0
	Bit Symbol	PJ7F2	PJ6F2	PJ5F2	PJ4F2	PJ3F2	—	—	—
	Read/Write	R/W					R		
	After reset	0	0	0	0	0	0		
	Function	0: PORT 1: KWUP3	0: PORT 1: KWUP2	0: PORT 1: KWUP1	0: PORT 1: KWUP0	0: PORT 1: ADTRGn	"0" is read.		

Port J pull-up control register

PJPUP (0x400C_092C)		7	6	5	4	3	2	1	0
	Bit Symbol	PJ7UP	PJ6UP	PJ5UP	PJ4UP	PJ3UP	PJ2UP	PJ2UP	PJ0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on

Port I input enable control register

PJIE (0x400C_0938)		7	6	5	4	3	2	1	0
	Bit Symbol	PJ7IE	PJ6IE	PJ5IE	PJ4IE	PJ3IE	PJ2IE	PJ1IE	PJ0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

(Note1)   Fuctio Register1 (PJFR1) is not exist in Port J.

(Note2)   Function setting is below.
       Analog input pin   :PJ=0x00, PJFR2=0x00, PJPUP=0x00, PJIE=0x00
       Input pin   :PJ=0x00, PJFR2=0x00, PJPUP=0x00, PJIE=0xFF
       KWUP input pin   :PJ=0x00, PJFR2=0xF0, PJPUP=0x00, PJIE=0xF0
       ADTRG input pin :PJ=0x00, PJFR2=0x08, PJPUP=0x00, PJIE=0x80
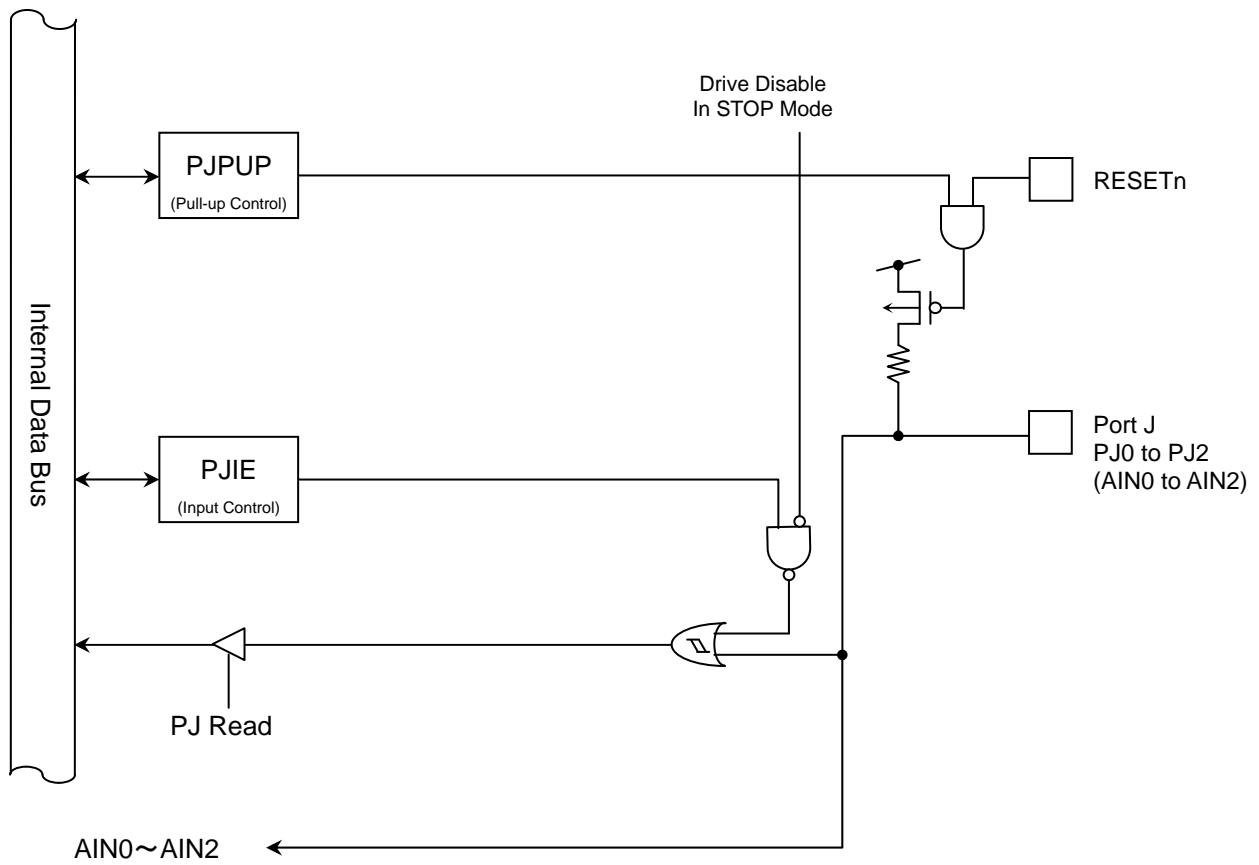
Fig. 3.6.27 Port J (PJ0 to PJ2)
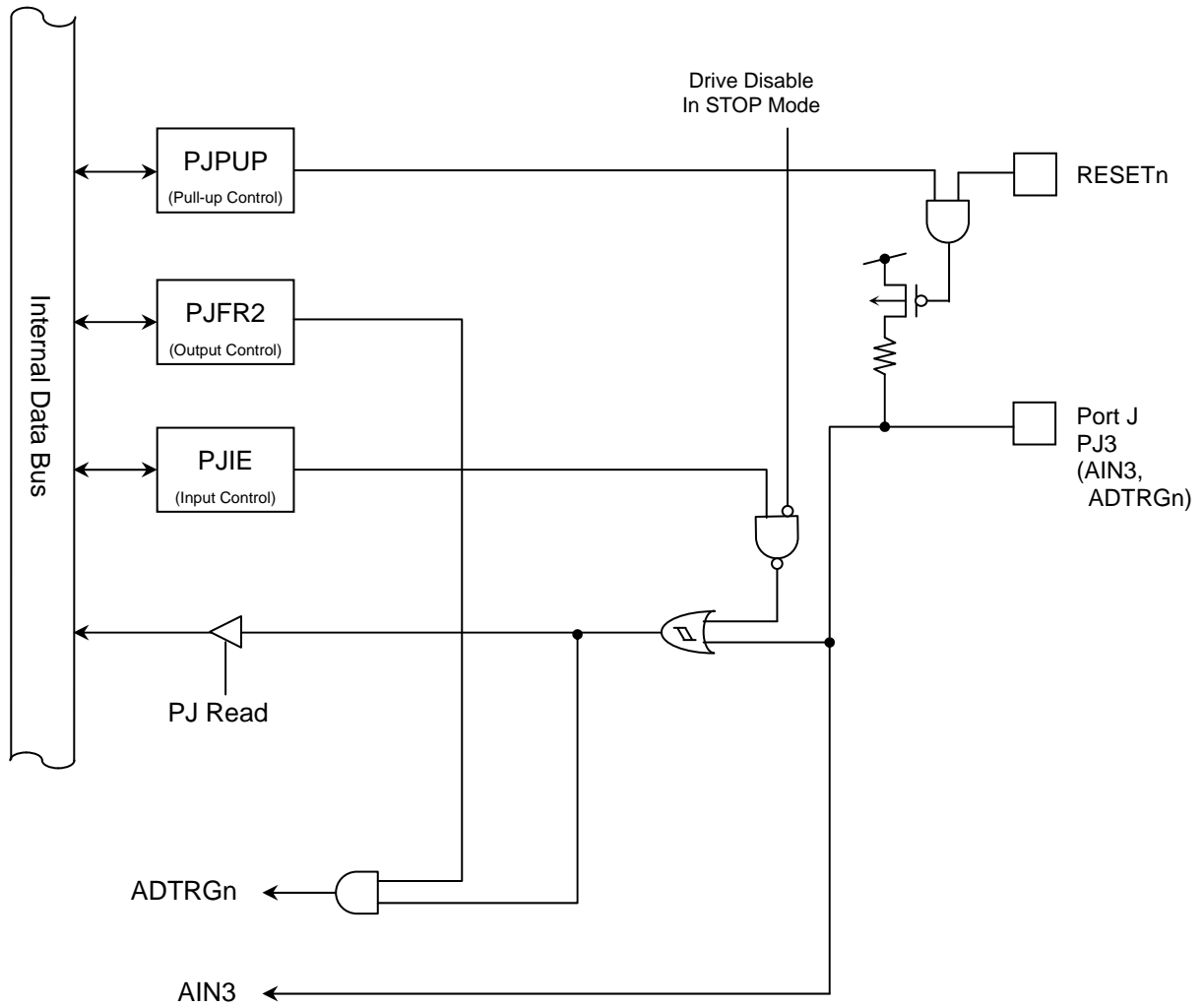
Fig. 3.6.28 Port J (PJ3)
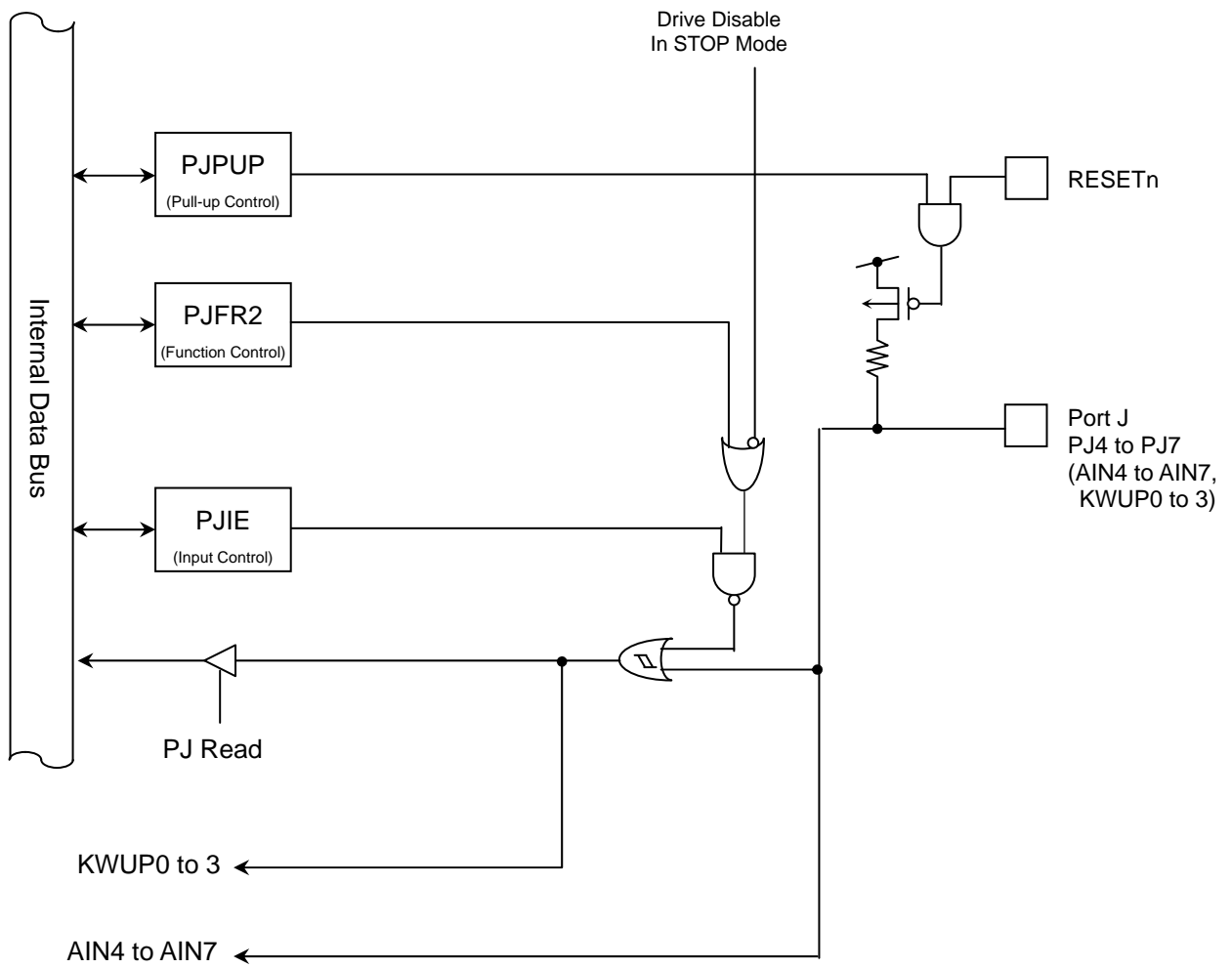
Fig. 3.6.29 Port J (PJ4 to PJ7)

3.6.2.11   Port K

The port K is an 8-bit input port. Besides the general-purpose input function, the port K receives an analog input of the A/D converter.

Reset initializes all bits of the port K as Analog input ports with input and pull-up disabled.

Set the input enable control register when you use the port K as input ports.

(note)	Unlsee you use all the bits of Port J and Port K as analog input pins, conversion accuracy may be reduced. Be sure to verify that this causes no problem on your system.

PortK register

PKDATA (0x400C_0A00)		7	6	5	4	3	2	1	0
	Bit Symbol	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
	Read/Write	R/W							
	After reset	"0"							

PortK pull-up control register

PKPUP (0x400C_0A2C)		7	6	5	4	3	2	1	0
	Bit Symbol	PK7UP	PK6UP	PK5UP	PK4UP	PK3UP	PK2UP	PK1UP	PK0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on

PortK input enable control register

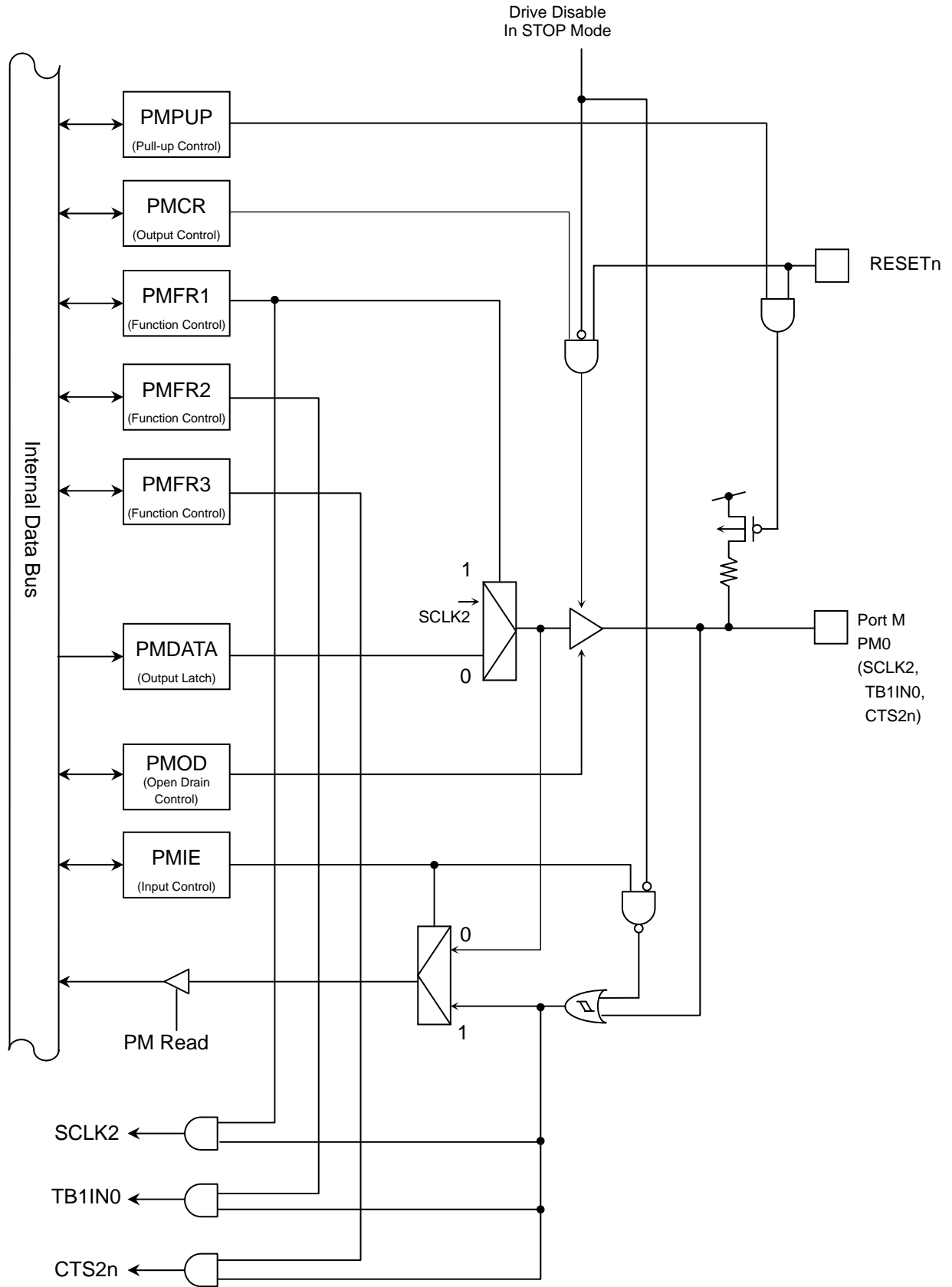| PKIE (0x400C_0A38)		7	6	5	4	3	2	1	0
	Bit Symbol	PK7IE	PK6IE	PK5IE	PK4IE	PK3IE	PK2IE	PK1IE	PK0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	1	1
	Function	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

(Note) Port J function setting is as following,
Analog input pins   : PK=0x00, PJPUP=0x00, PJIE=0x00
Input Port pins   : PK=0x00, PJPUP=0x00, PJIE=0xFF

Fig3.6.30 PortK (PK0～PK7)

3.6.2.12 Port L

The port L is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose port function, the port L performs the functions of the serial interface(SIO/UART), the functions of the serial bus interface(I2C/SIO), the external interrupt input, and the 16-bit timer output.

Reset initializes all bits of the port L as general-purpose ports with input, output and pull-up disabled.

Port L register

PLDATA (0x400C_0B00)		7	6	5	4	3	2	1	0
	Bit Symbol	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
	Read/Write	R/W							
	After reset	"0"							

Port L control register

PLCR (0x400C_0B04)		7	6	5	4	3	2	1	0
	Bit Symbol	PL7C	PL6C	PL5C	PL4C	PL3C	PL2C	PL1C	PL0C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Output disabled 1: Output enabled							

Port L function register 1

PLFR1 (0x400C_0B08)		7	6	5	4	3	2	1	0
	Bit Symbol	PL7F1	PL6F1	PL5F1	PL4F1	PL3F1	PL2F1	PL1F1	PL0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: INT1	0: PORT 1: SCLK1	0: PORT 1: RXD1	0: PORT 1: TXD1	0: PORT 1: INT0	0: PORT 1: SCK0	0: PORT 1: SCL0	0: PORT 1: SDA0

Port L function register 2

PLFR2 (0x400C_0B0C)		7	6	5	4	3	2	1	0
	Bit Symbol	PL7F2	PL6F2	PL5F2	PL4F2	PL3F2	PL2F2	PL1F2	PL0F2
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1:TB7OUT	0: PORT 1:TB6OUT	0: PORT 1:TB5OUT	0: PORT 1:TB4OUT	0: PORT 1:TB3OUT	0: PORT 1:TB2OUT	0: PORT 1:TB1OUT	0: PORT 1:TB0OUT

Port L function register 3

PLFR3 (0x400C_0B10)		7	6	5	4	3	2	1	0
	Bit Symbol	—	PL6F3	（Reserved）	（Reserved）	—	—	—	—
	Read/Write	R	R/W	R		R			
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.	0: PORT 1: CTS1n	"0" is read.		"0" is read.			

Port L open drain control register

PLOD (0x400C_0B28)		7	6	5	4	3	2	1	0
	Bit Symbol	PL7 OD	PL6 OD	PL5 OD	PL4 OD	PL3 OD	PL2 OD	PL1 OD	PL0OD
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain

Port L pull-up control register

PLPUP (0x400C_0B2C)		7	6	5	4	3	2	1	0
	Bit Symbol	PL7UP	PL6UP	PL5UP	PL4UP	PL3UP	PL2UP	PL1UP	PL0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on

Port L input enable control register

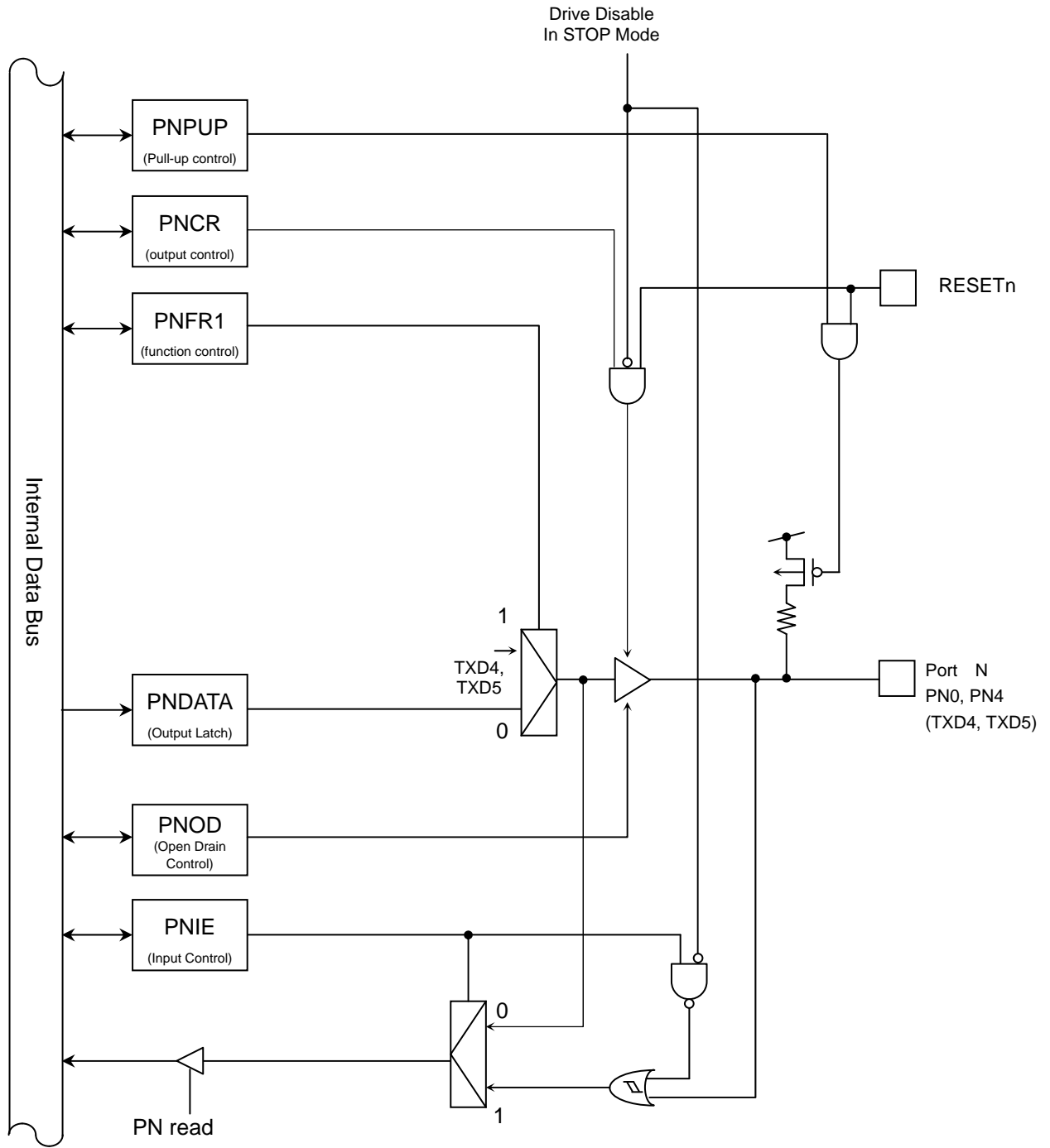| PLIE (0x400C_0B38)		7	6	5	4	3	2	1	0
	Bit Symbol	PL7IE	PL6IE	PL5IE	PL4IE	PL3IE	PL2IE	PL1IE	PL0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

Fig. 3.6.31 Port L (PL0 to PL2)

Fig. 3.6.32 Port L (PL3, PL7)
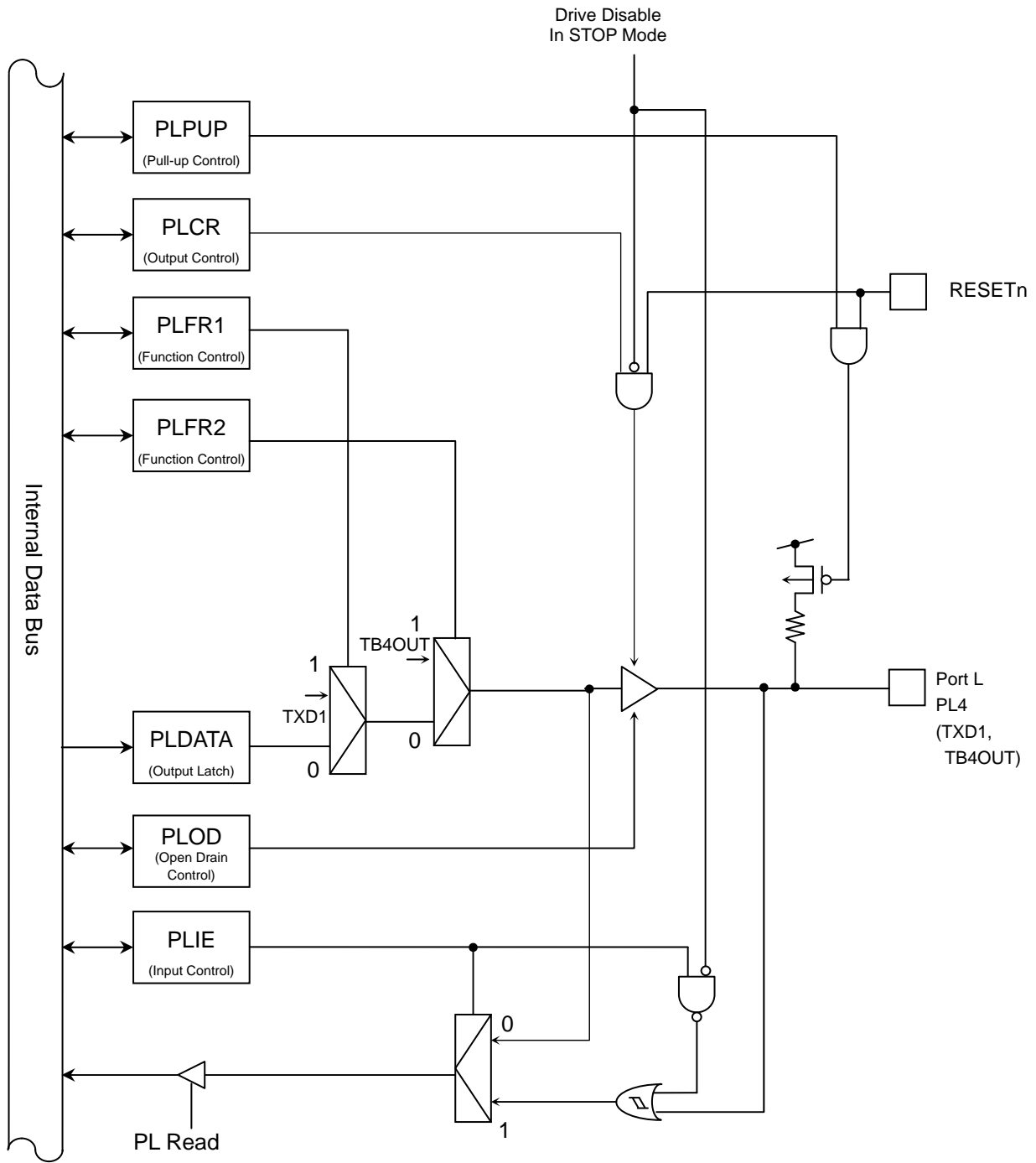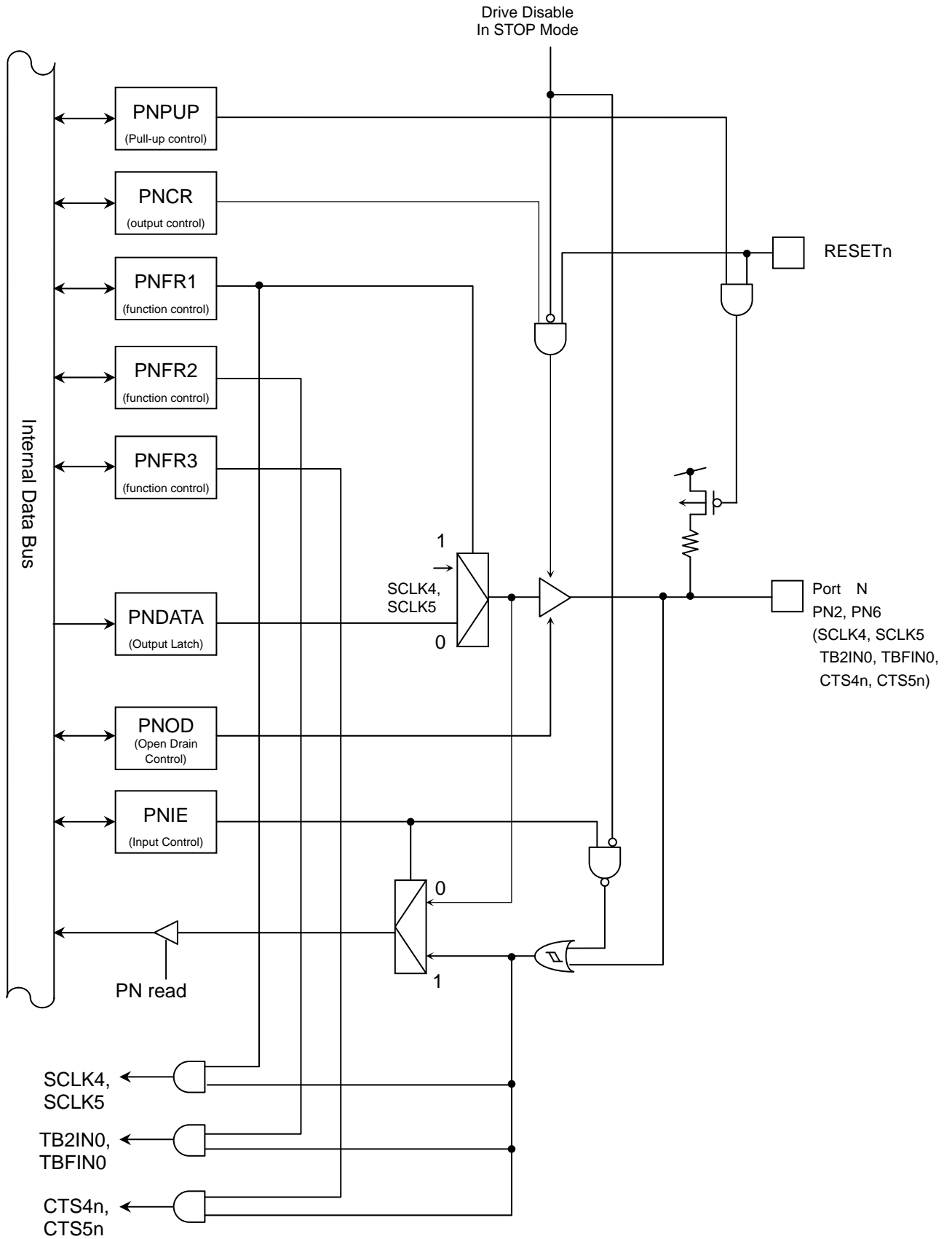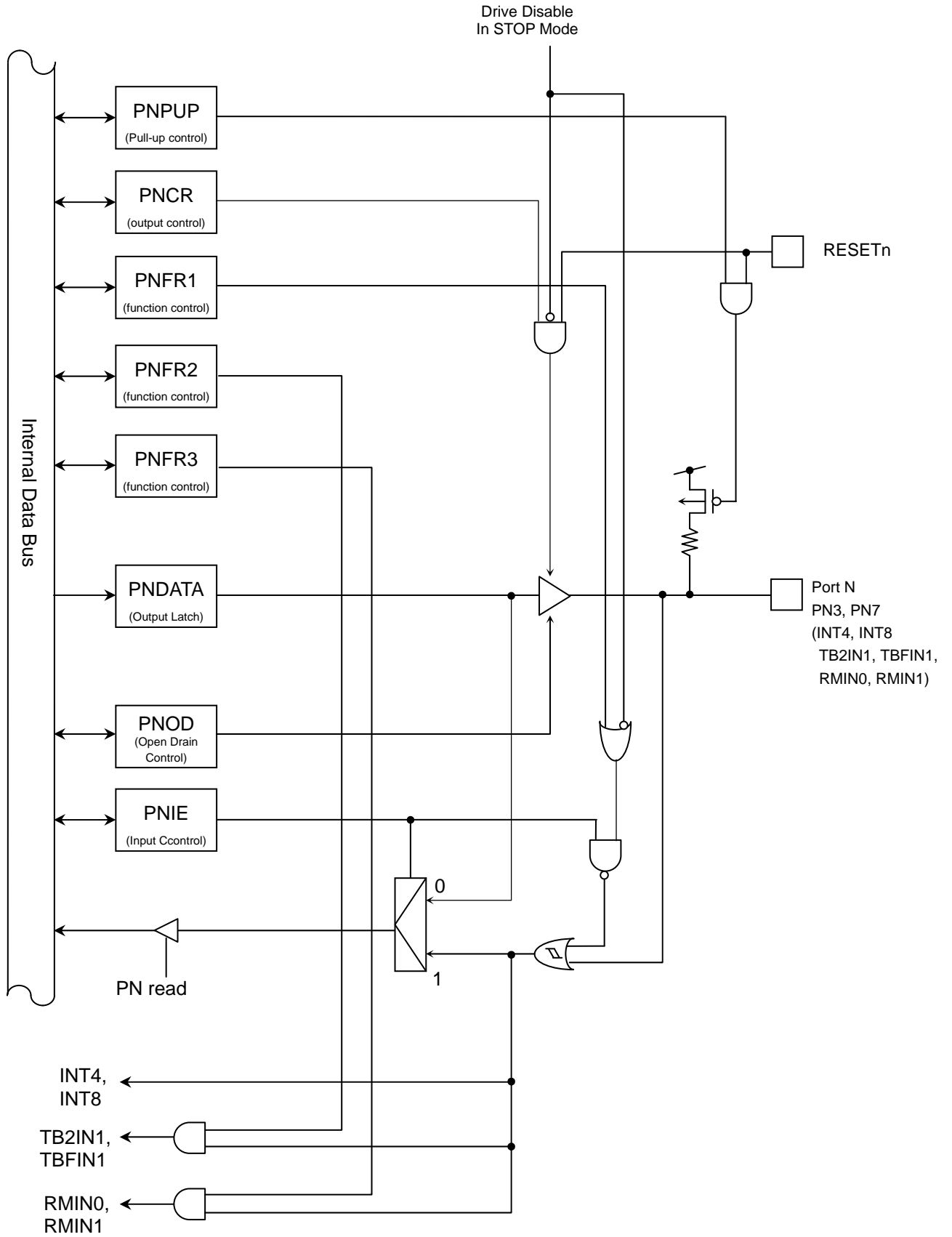
Fig. 3.6.33 Port L (PL4)
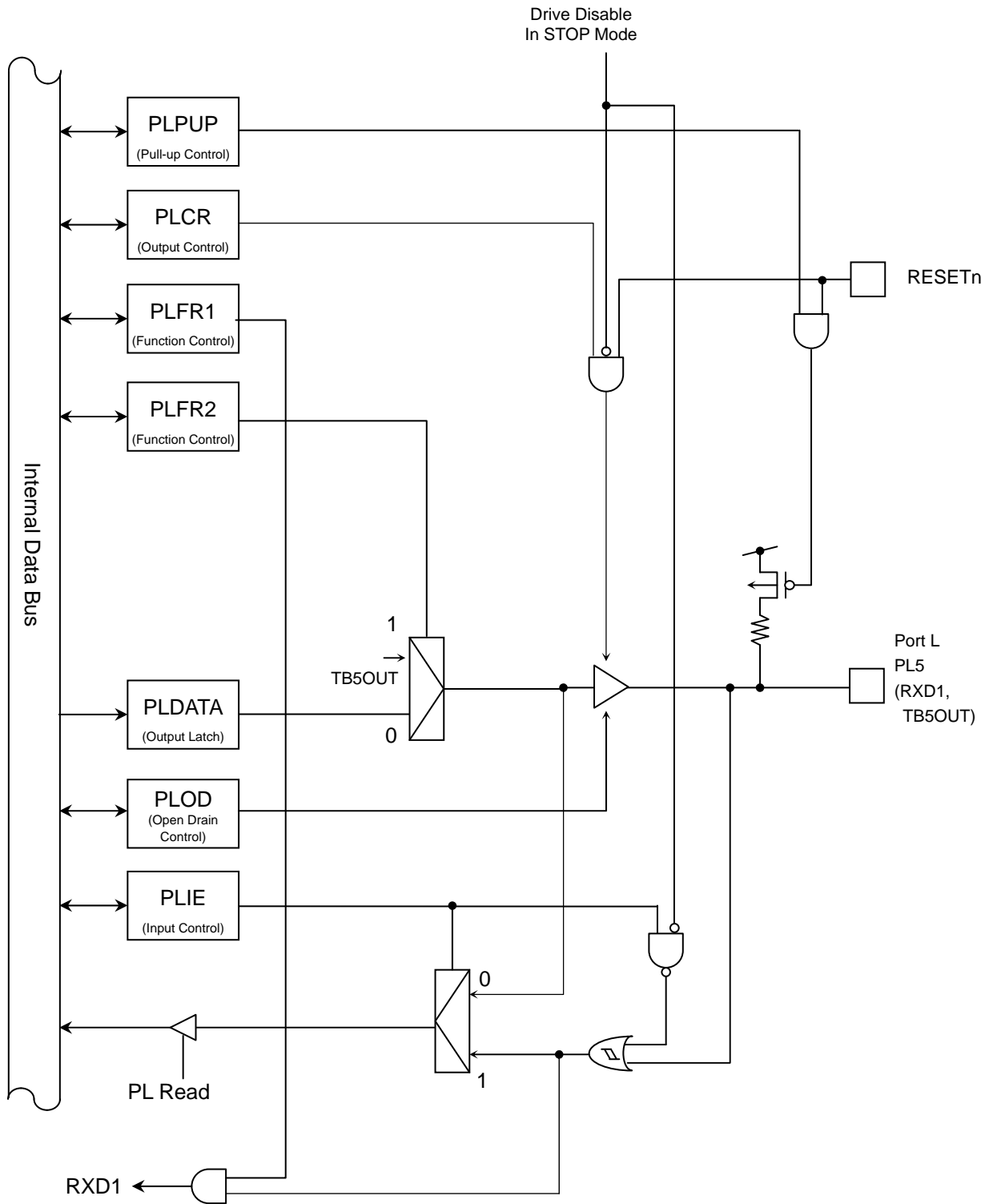
Fig. 3.6.34 Port L (PL5)
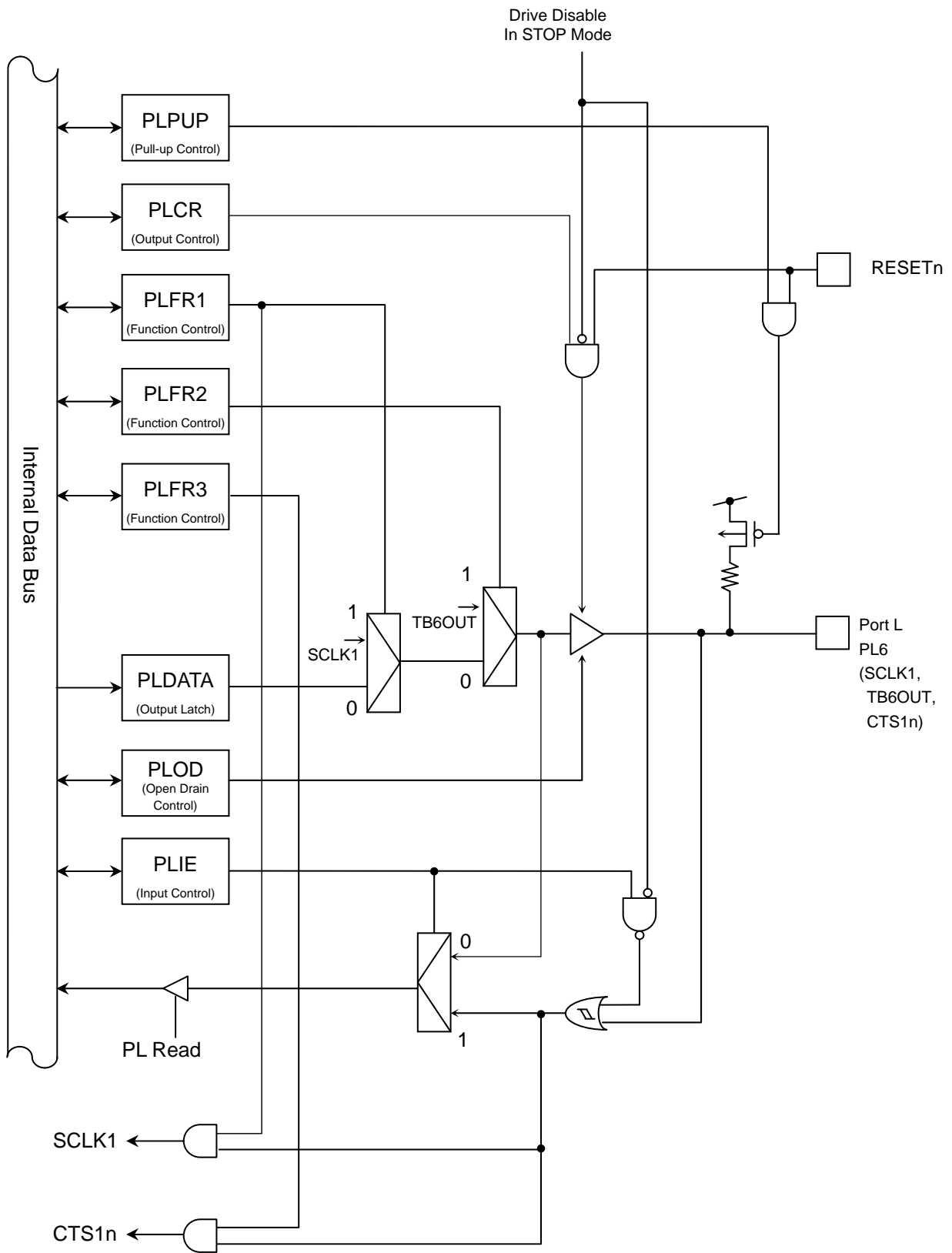
Fig. 3.6.35 Port L (PL6)

3.6.2.13 Port M

The port M is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose port function, the port M performs the functions of the serial interface(SIO/UART), the external interrupt input, the 16-bit timer intput and Alarm output.

Reset initializes all bits of the port M as general-purpose ports with input, output and pull-up disabled.

Port M register

PMDATA (0x400C_0C00)		7	6	5	4	3	2	1	0
	Bit Symbol	PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0
	Read/Write	R/W							
	After reset	"0"							

Port M control register

PMCR (0x400C_0C04)		7	6	5	4	3	2	1	0
	Bit Symbol	PM7C	PM6C	PM5C	PM4C	PM3C	PM2C	PM1C	PM0C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Output disabled 1: Output enabled							

Port M function register 1

PMFR1 (0x400C_0C08)		7	6	5	4	3	2	1	0
	Bit Symbol	PM7F1	PM6F1	PM5F1	PM4F1	PM3F1	PM2F1	PM1F1	PM0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: INT3	0: PORT 1: RXD3	0: PORT 1: TXD3	0: PORT 1: SCLK3	0: PORT 1: INT2	0: PORT 1: RXD2	0: PORT 1: TXD2	0: PORT 1: SCLK2

Port M function register 2

PMFR2 (0x400C_0C0C)		7	6	5	4	3	2	1	0
	Bit Symbol	−	−	−	-	PM3F2	PM2F2	PM1F2	PM0F2
	Read/Write	R			R	R/W			
	After reset	0			0	0	0	0	0
	Function	"0" is read.			"0" is read.	0: PORT 1: TB3OUT	0: PORT 1: ALARMn	0: PORT 1: TB1IN1	0: PORT 1: TB1IN0

Port M function register 3

PMFR3 (0x400C_0C10)		7	6	5	4	3	2	1	0
	Bit Symbol	−	−	−	PM4F3	−	−	−	PM0F3
	Read/Write	R			R/W	R			R/W
	After reset	0			0	0			0
	Function	"0" is read.			0: PORT 1: CTS3n	"0" is read.			0: PORT 1: CTS2n

Port M open drain control register

PMOD (0x400C_0C28)		7	6	5	4	3	2	1	0
	Bit Symbol	PM7 OD	PM6 OD	PM5 OD	PM4 OD	PM3 OD	PM2 OD	PM1 OD	PM0OD
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain	0:CMOS 1:Open drain

Port M pull-up control register

PMPUP (0x400C_0C2C)		7	6	5	4	3	2	1	0
	Bit Symbol	PM7UP	PM6UP	PM5UP	PM4UP	PM3UP	PM2UP	PM1UP	PM0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on	Pull-up 0: off 1: on

Port M input enable control register

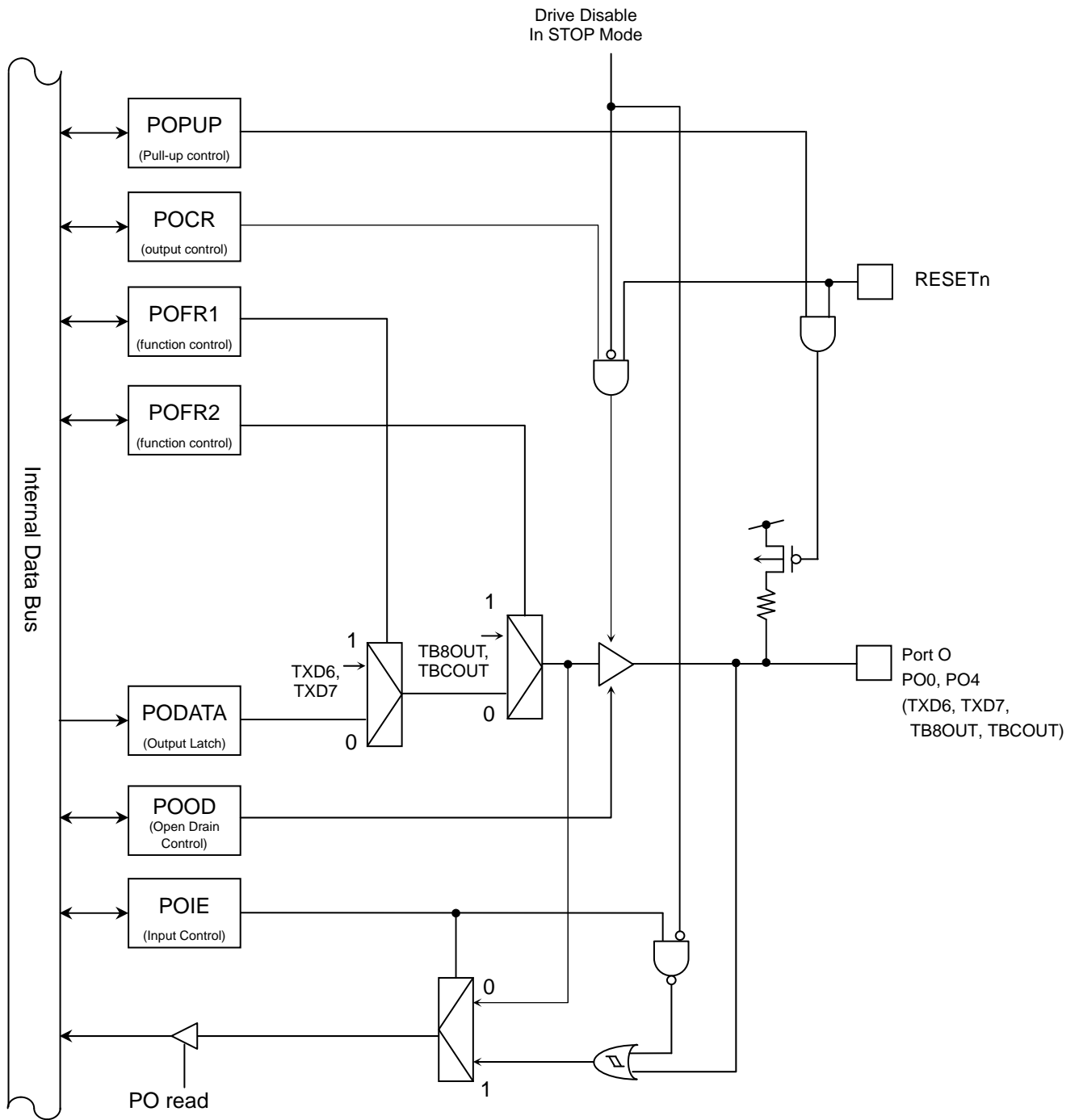| PMIE (0x400C_0C38)		7	6	5	4	3	2	1	0
	Bit Symbol	PM7IE	PM6IE	PM5IE	PM4IE	PM3IE	PM2IE	PM1IE	PM0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

Fig. 3.6.36 Port M (PM0)

Fig. 3.6.37 Port M (PM1)
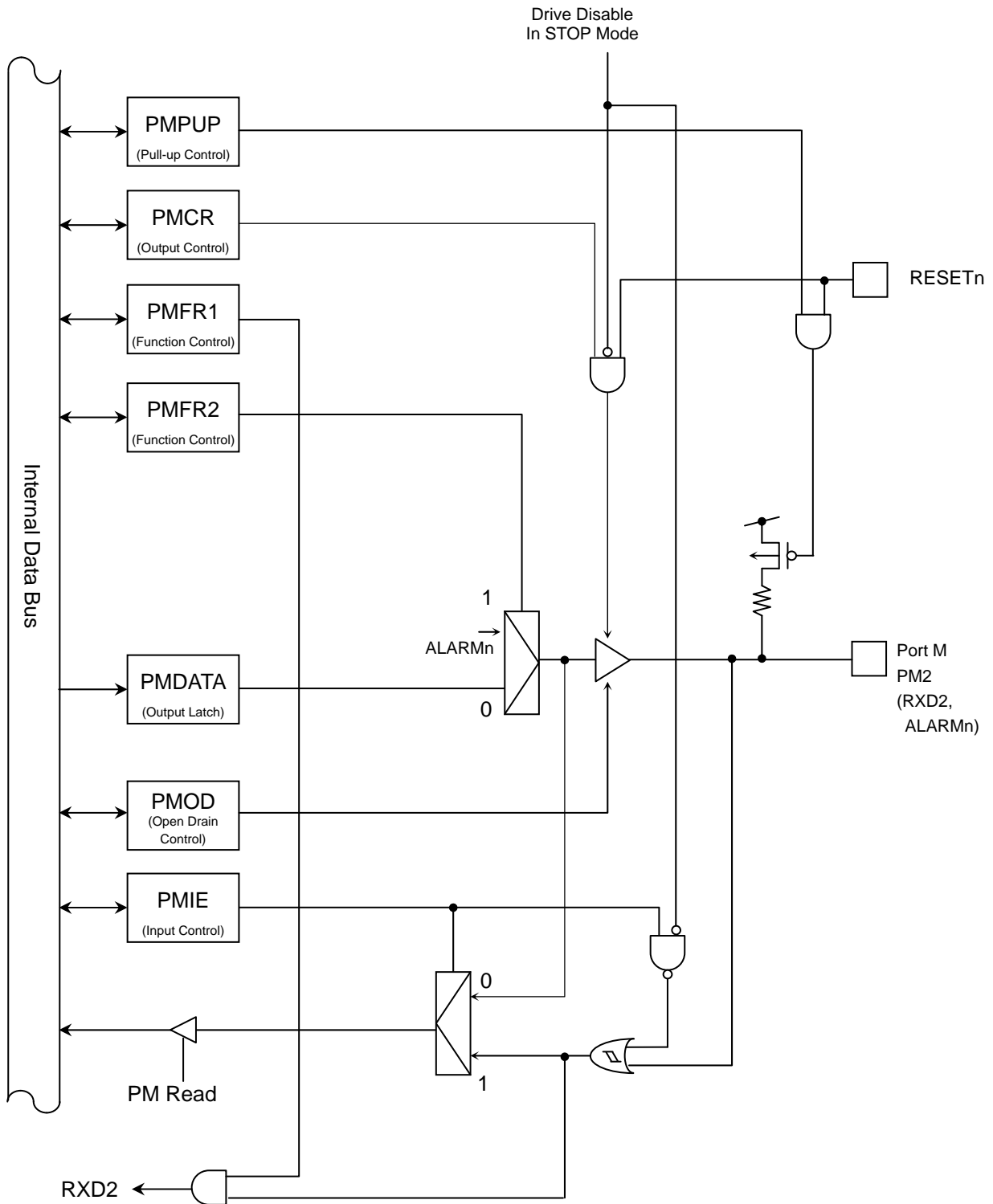
Fig. 3.6.38 Port M (PM2)
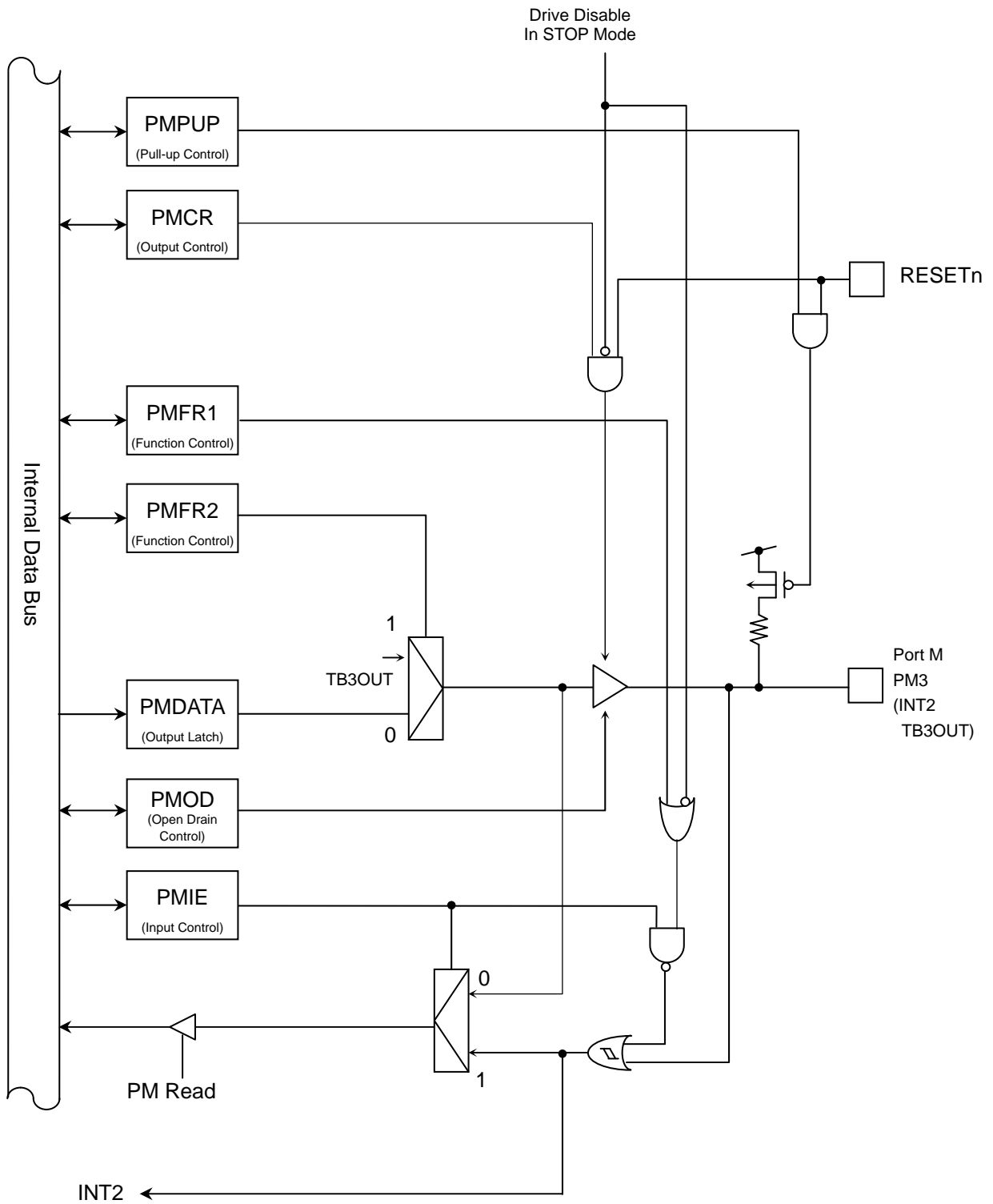
Fig. 3.6.39 Port M (PM3)
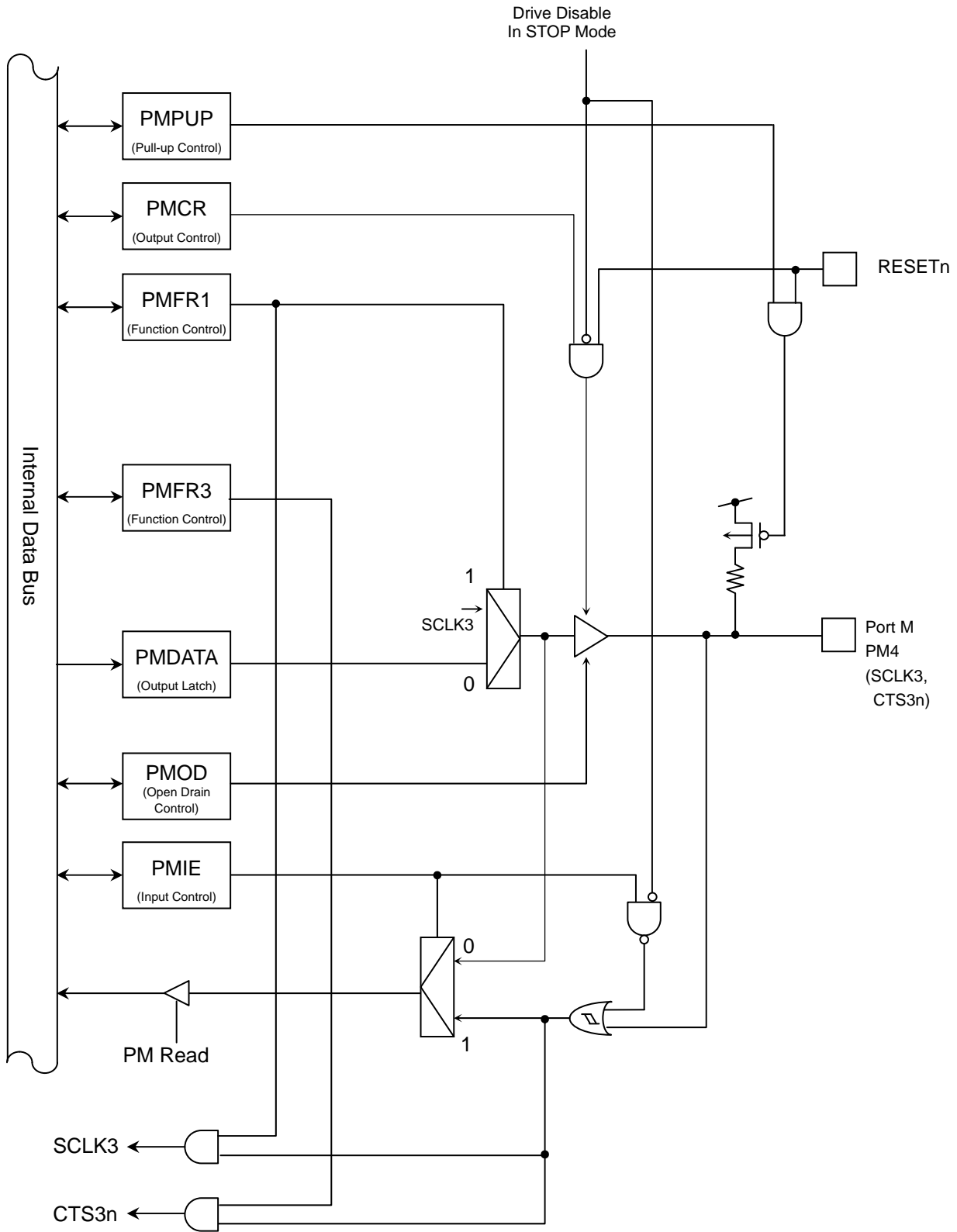
Fig. 3.6.40 Port M (PM4)
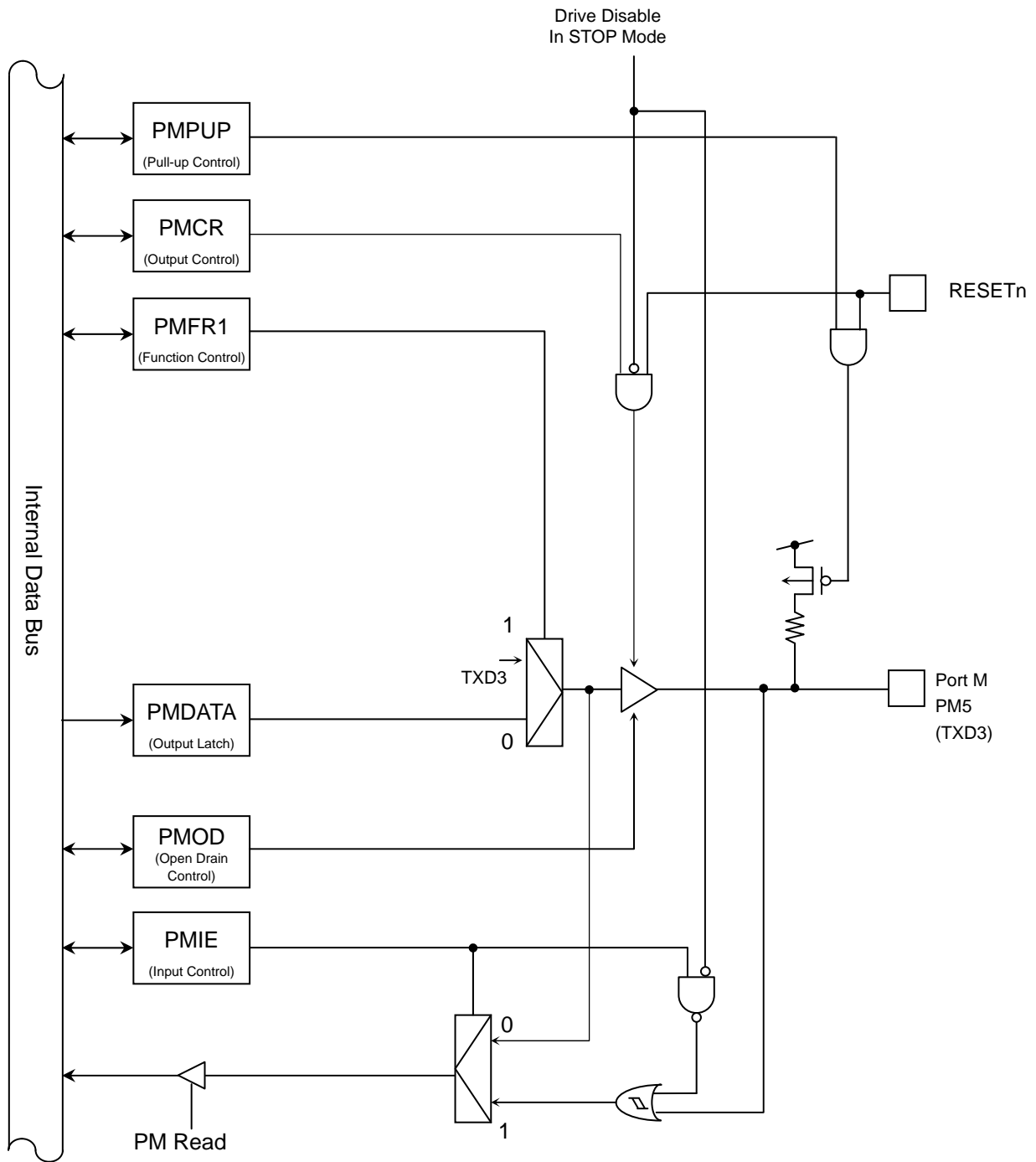
Fig. 3.6.41 Port M (PM5)
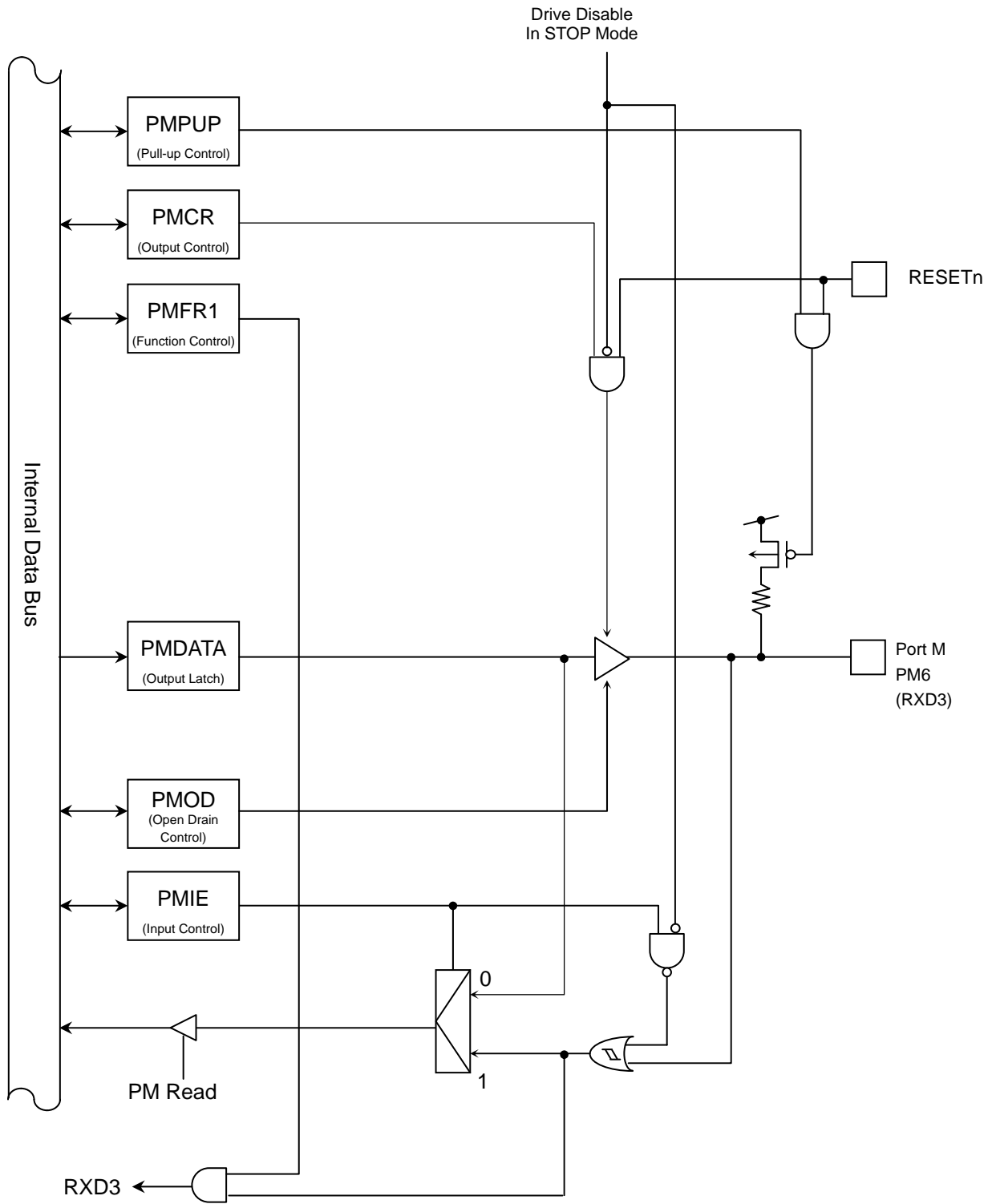
Fig. 3.6.42 Port M (PM6)
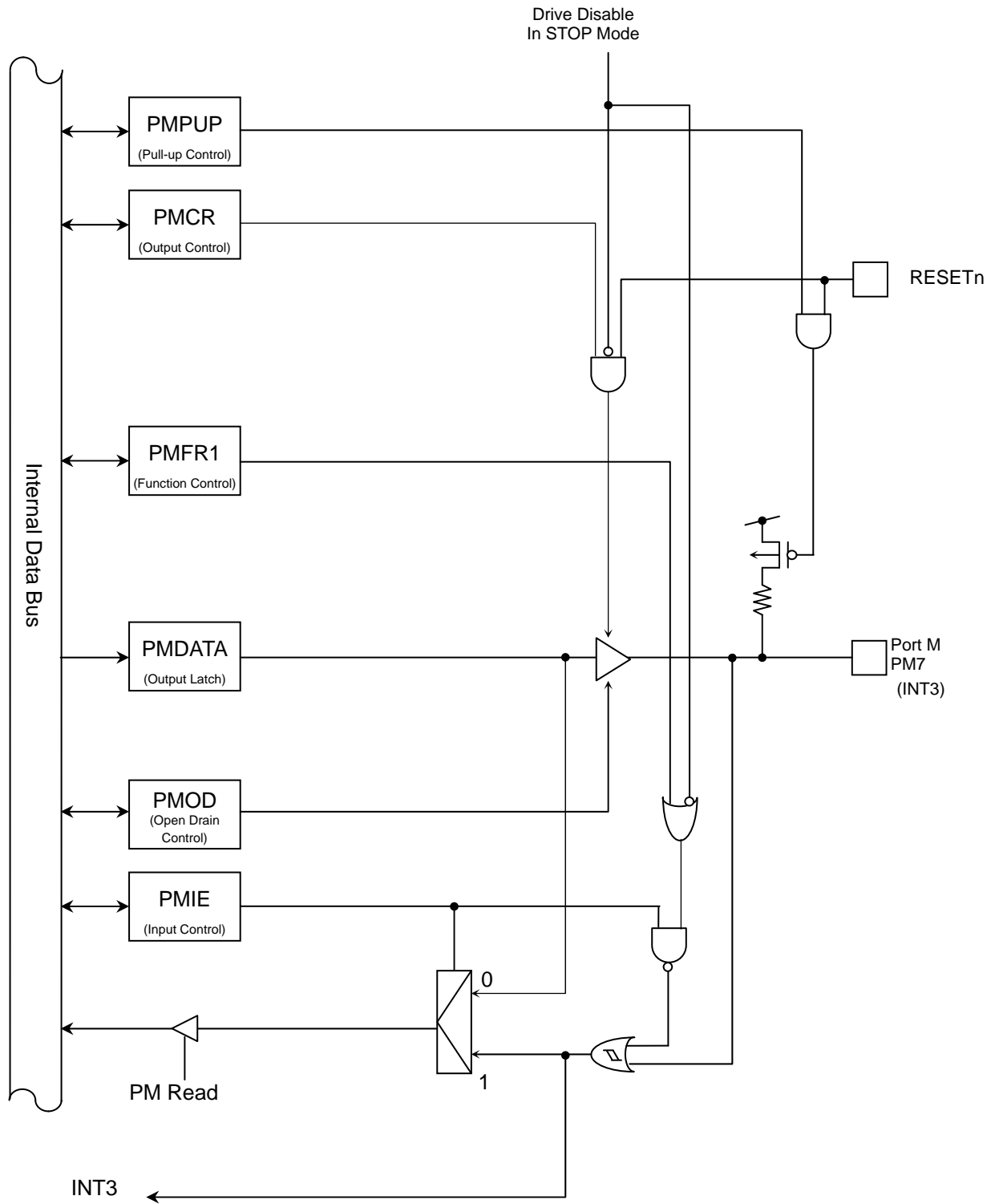
Fig. 3.6.43 Port M (PM7)

3.6.2.14   Port N

The port N is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port N performs the serial interface function(UART/SIO), External signal interrupt input ,16bit timer output function and the remote control signal preprocessor input function..

Reset initializes all bits of the port N as general-purpose ports with input, output and pull-up disabled.

PortN   register

PNDATA (0x400C_0D00)		7	6	5	4	3	2	1	0
	Bit Symbol	PN7	PN6	PN5	PN4	PN3	PN2	PN1	PN0
	Read/Write	R/W							
	After reset	"0"							

PortN   control register

PNCR (0x400C_0D04)		7	6	5	4	3	2	1	0
	Bit Symbol	PN7C	PN6C	PN5C	PN4C	PN3C	PN2C	PN1C	PN0C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Output disable 1:Output enabled							

PortN   function register1

PNFR1 (0x400C_0D08)		7	6	5	4	3	2	1	0
	Bit Symbol	PN7F1	PN6F1	PN5F1	PN4F1	PN3F1	PN2F1	PN1F1	PN0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: INT8	0: PORT 1: SCLK5	0: PORT 1: RXD5	0: PORT 1: TXD5	0: PORT 1: INT4	0: PORT 1: SCLK4	0: PORT 1: RXD4	0: PORT 1: TXD4

PortN   function register2

PNFR2 (0x400C_0D0C)		7	6	5	4	3	2	1	0
	Bit Symbol	PN7F2	PN6F2	—	—	PN3F2	PN2F2	—	—
	Read/Write	R/W	R/W	R		R/W		R	
	After reset	0	0	0		0	0	0	
	Function	0: PORT 1: TBFIN1	0: PORT 1: TBFIN0	"0" is read.		0: PORT 1: TB2IN1	0: PORT 1: TB2IN0	"0" is read.	

PortN   function register3

PNFR3 (0x400C_0D10)		7	6	5	4	3	2	1	0
	Bit Symbol	PN7F3	PN6F3	—	—	PN3F3	PN2F3	—	—
	Read/Write	R/W		R		R/W		R	
	After reset	0	0	0		0	0	0	
	Function	0: PORT 1: RMIN1	0: PORT 1: CTS5n	"0" is read.		0: PORT 1: RMIN0	0: PORT 1: CTS4n	"0" is read.	

PortN open drain control register

PNOD (0x400C_0D28)		7	6	5	4	3	2	1	0
	Bit Symbol	PN7 OD	PN6 OD	PN5 OD	PN4 OD	PN3 OD	PN2 OD	PN1 OD	PN0OD
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain

PortN pull-up control register

PNPUP (0x400C_0D2C)		7	6	5	4	3	2	1	0
	Bit Symbol	PN7UP	PN6UP	PN5UP	PN4UP	PN3UP	PN2UP	PN1UP	PN0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on

PortN input enable control register

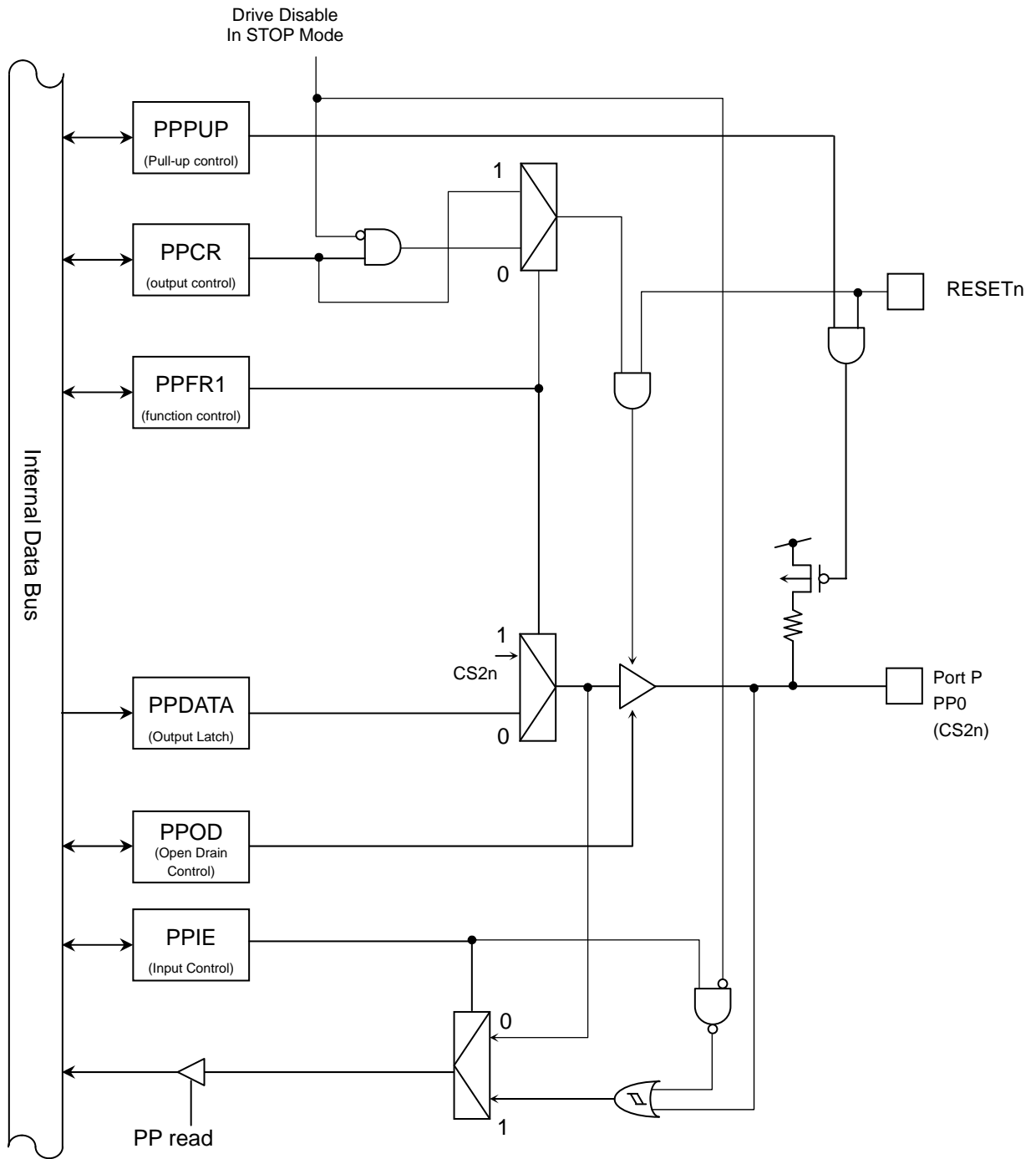| PNIE (0x400C_0D38)		7	6	5	4	3	2	1	0
	Bit Symbol	PN7IE	PN6IE	PN5IE	PN4IE	PN3IE	PN2IE	PN1IE	PN0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

Fig3.6.44   PortN (PN0, PN4)
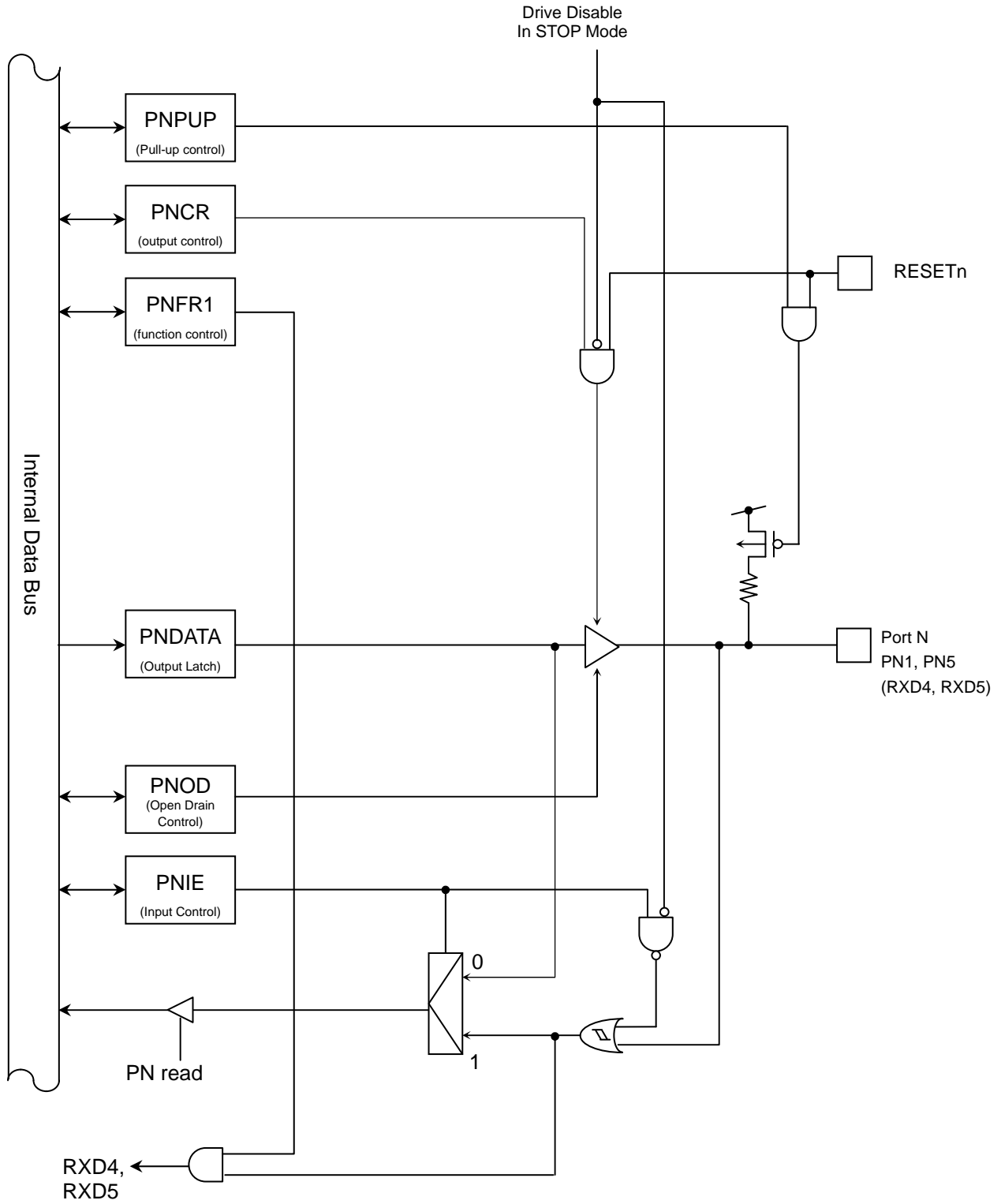
Fig3.6.45 PortN (PN1, PN5)

Fig3.6.46   PortN (PN2, PN6)

Fig3.6.47 PortN (PN3, PN7)

3.6.2.15   Port O

The port O is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port O performs the serial interface function(UART/SIO),External signal interrupt input and 16bit timer output function.

Reset initializes all bits of the port O as general-purpose ports with input, output and pull-up disabled.

PortO register

PODATA (0x400C_0E00)		7	6	5	4	3	2	1	0
	Bit Symbol	PO7	PO6	PO5	PO4	PO3	PO2	PO1	PO0
	Read/Write	R/W							
	After reset	"0"							

PortO control register

POCR (0x400C_0E04)		7	6	5	4	3	2	1	0
	Bit Symbol	PO7C	PO6C	PO5C	PO4C	PO3C	PO2C	PO1C	PO0C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Output disable 1:Output enabled							

PortO function register1

POFR1 (0x400C_0E08)		7	6	5	4	3	2	1	0
	Bit Symbol	–	PO6F1	PO5F1	PO4F1	PO3F1	PO2F1	PO1F1	PO0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: INTA	0: PORT 1: SCLK7	0: PORT 1: RXD7	0: PORT 1: TXD7	0: PORT 1: INT9	0: PORT 1: SCLK6	0: PORT 1: RXD6	0: PORT 1: TXD6

PortO function register2

POFR2 (0x400C_0E0C)		7	6	5	4	3	2	1	0
	Bit Symbol	–	PO6F2	PO5F2	PO4F2	PO3F2	PO2F2	PO1F2	PO0F2
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1:TBFOUT	0: PORT 1:TBEOUT	0: PORT 1:TBDOUT	0: PORT 1:TBCOUT	0: PORT 1:TBBOUT	0: PORT 1:TBAOUT	0: PORT 1:TB9OUT	0: PORT 1:TB8OUT

PortO function register3

POFR3 (0x400C_0E10)		7	6	5	4	3	2	1	0
	Bit Symbol	–	PO6F3	–	–	–	PO2F3	–	–
	Read/Write	R	R/W	R			R/W	R	
	After reset	0	0	0			0	0	
	Function	"0" is read.	0: PORT 1: CTSn7	"0" is read.			0: PORT 1: CTS6n	"0" is read.	

PortO open drain control register

POOD (0x400C_0E28)		7	6	5	4	3	2	1	0
	Bit Symbol	PO7 OD	PO6 OD	PO5 OD	PO4 OD	PO3 OD	PO2 OD	PO1 OD	PO0OD
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain

PortO pull-up control register

POPUP (0x400C_0E2C)		7	6	5	4	3	2	1	0
	Bit Symbol	PO7UP	PO6UP	PO5UP	PO4UP	PO3UP	PO2UP	PO1UP	PO0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0:off 1: on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on

PortO input enable control register

POIE (0x400C_0E38)		7	6	5	4	3	2	1	0
	Bit Symbol	PO7IE	PO6IE	PO5IE	PO4IE	PO3IE	PO2IE	PO1IE	PO0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

Fig3.6.48 PortO (PO0, PO4)
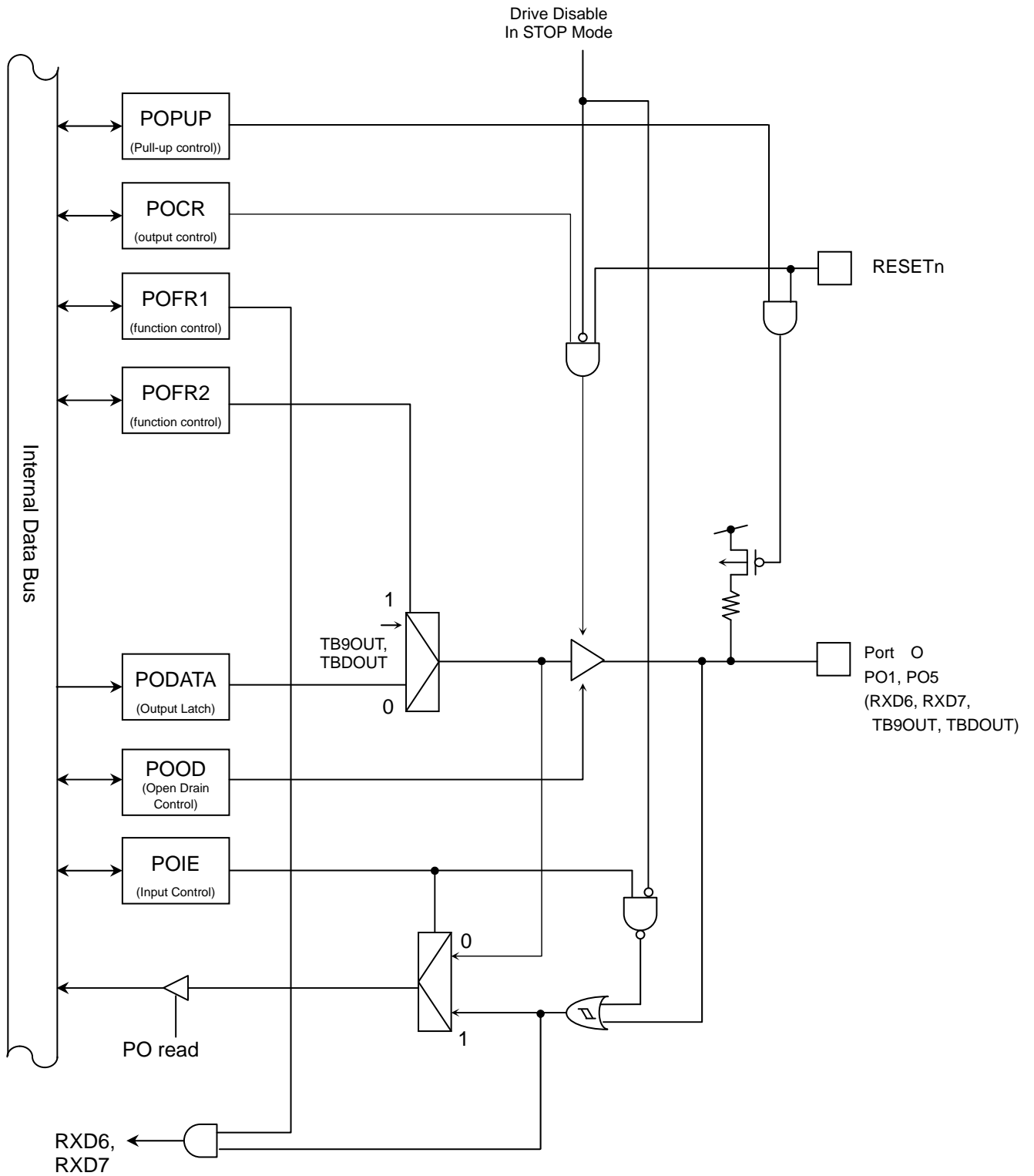
Fig3.6.49　PortO (PO1, PO5)
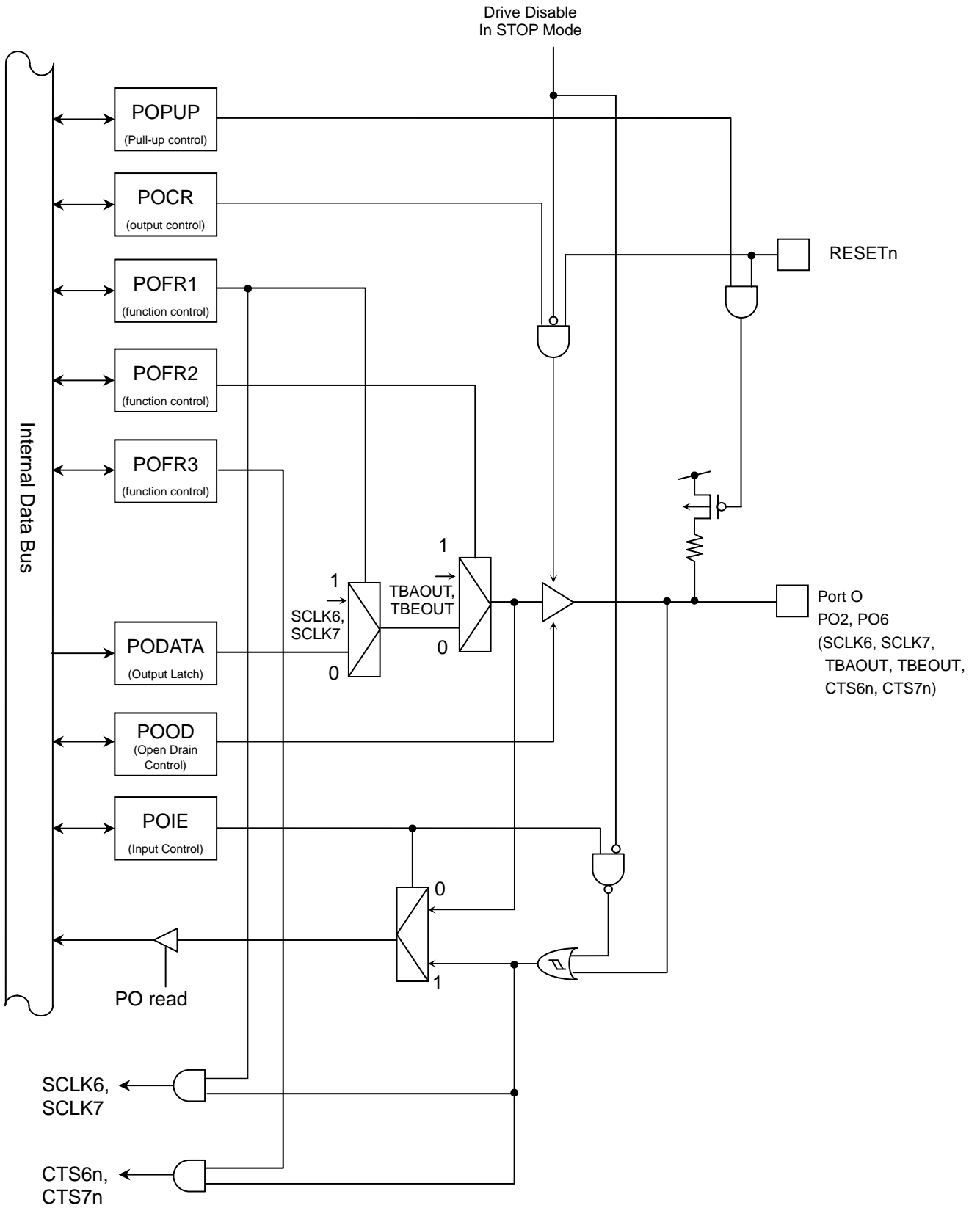
Fig3.6.50   PortO (PO2, PO6)
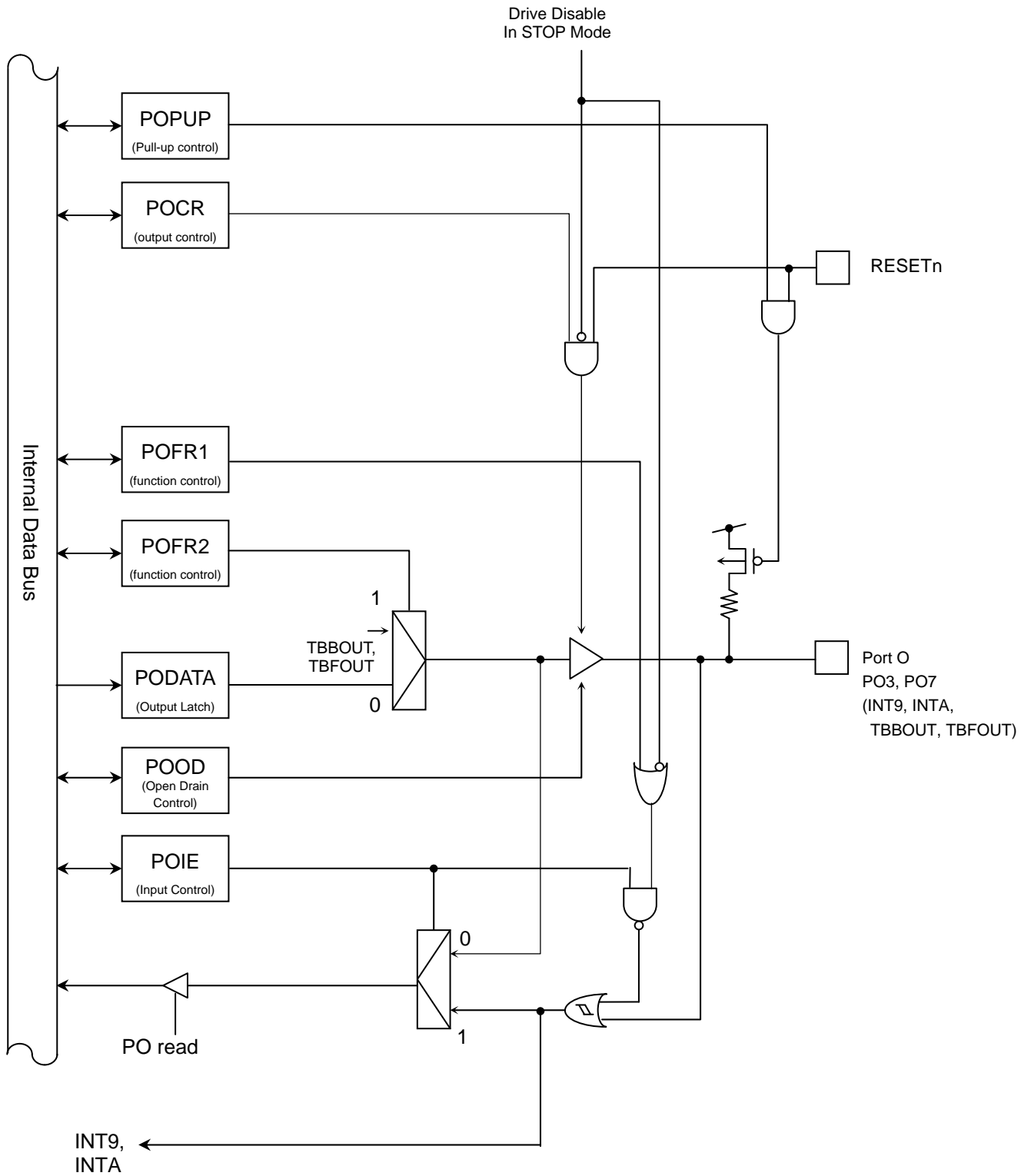
Fig3.6.51 PortO (PO3, PO7)

3.6.2.16   Port P

The port P is a general-purpose, 7-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port P performs the Extranal-Bus control function, chip-select function and , the synchronous serial interface function(SSP).

Reset initializes all bits of the port P as general-purpose ports with input, output and pull-up disabled.

PortP register

		7	6	5	4	3	2	1	0
PPDATA	Bit Symbol	—	PP6	PP5	PP4	PP3	PP2	PP1	PP0
(0x400C_0F00)	Read/Write	R	R/W						
	After reset	"0"	"0"						

PortP control register

		7	6	5	4	3	2	1	0
PPCR	Bit Symbol	—	PP6C	PP5C	PP4C	PP3C	PP2C	PP1C	PP0C
(0x400C_0F04)	Read/Write	R	R/W						
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.	0: Output disable 1:Output enabled						

PortP function register1

		7	6	5	4	3	2	1	0
PPFR1	Bit Symbol	—	—	PP5F1	PP4F1	PP3F1	PP2F1	PP1F1	PP0F1
(0x400C_0F08)	Read/Write	R	R/W						
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.	Always write "0".	0: PORT 1: OEn	0: PORT 1: Wen	0: PORT 1: BLS1n	0: PORT 1: BLS0n	Always write "0".	0: PORT 1: CS2n

PortP function register2

		7	6	5	4	3	2	1	0
PPFR2	Bit Symbol	—	—	PP5F2	PP4F2	PP3F2	PP2F2	—	—
(0x400C_0F0C)	Read/Write	R		R/W				R	
	After reset	0		0	0	0	0	0	
	Function	"0" is read.		0: PORT 1: SPFSS	0: PORT 1: SPCLK	0: PORT 1: SPDI	0: PORT 1: SPDO	"0" is read.	

PortP open drain control register

PPOD (0x400C_0F28)		7	6	5	4	3	2	1	0
	Bit Symbol	–	PP6 OD	PP5 OD	PP4 OD	PP3 OD	PP2 OD	PP1 OD	PP0OD
	Read/Write	R	R/W						
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain	0: CMOS 1:Open drain

PortP pull-up control register

PPPUP (0x400C_0F2C)		7	6	5	4	3	2	1	0
	Bit Symbol	–	PP6UP	PP5UP	PP4UP	PP3UP	PP2UP	PP1UP	PP0UP
	Read/Write	R	R/W						
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on	Pull-up 0:off 1:on

PortP input enable control register

PPIE (0x400C_0F38)		7	6	5	4	3	2	1	0
	Bit Symbol	–	PP6IE	PP5IE	PP4IE	PP3IE	PP2IE	PP1IE	PP0IE
	Read/Write	R	R/W						
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

Fig3.6.52 PortP (PP0)
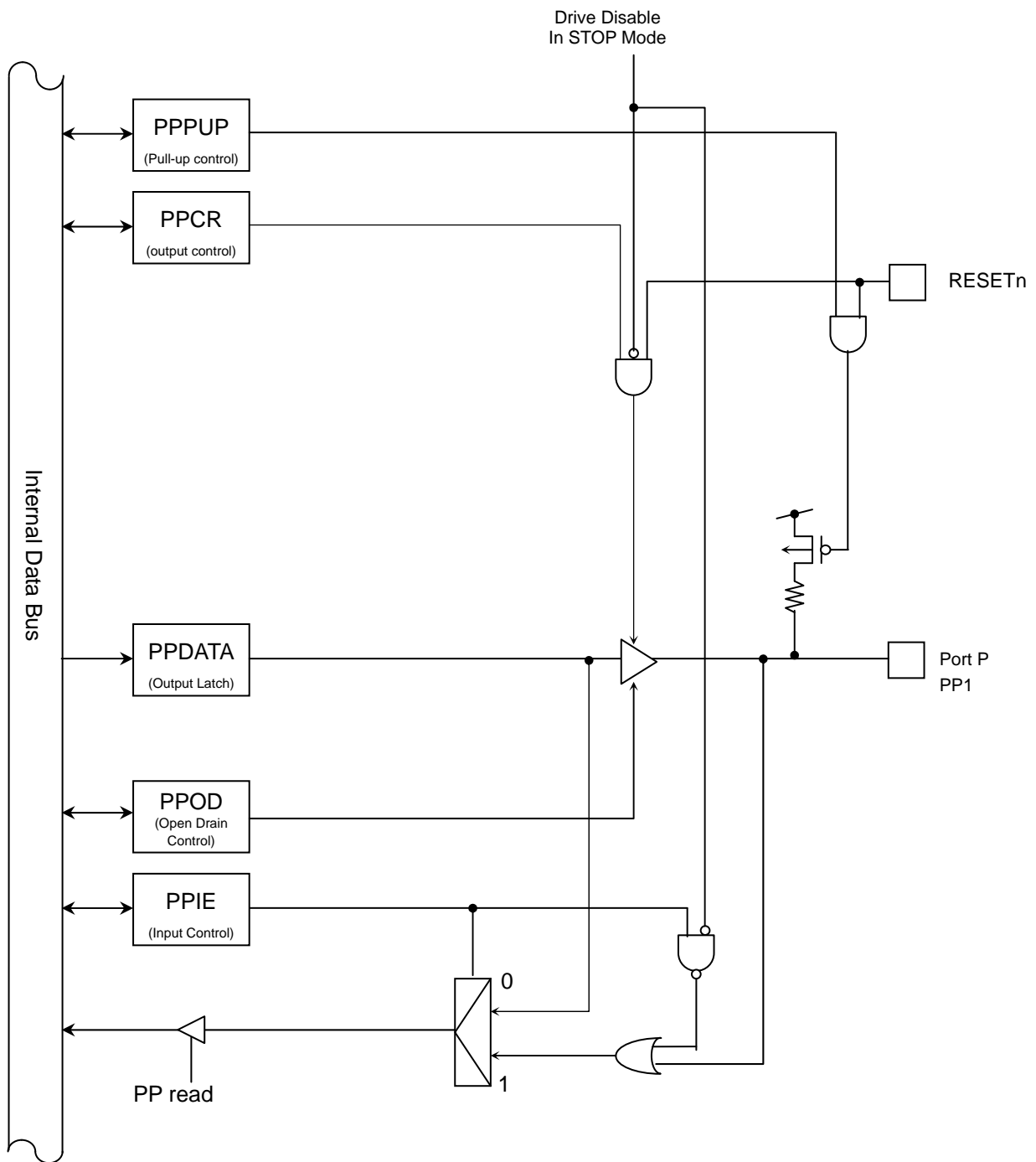
Fig3.6.53   PortP (PP1)
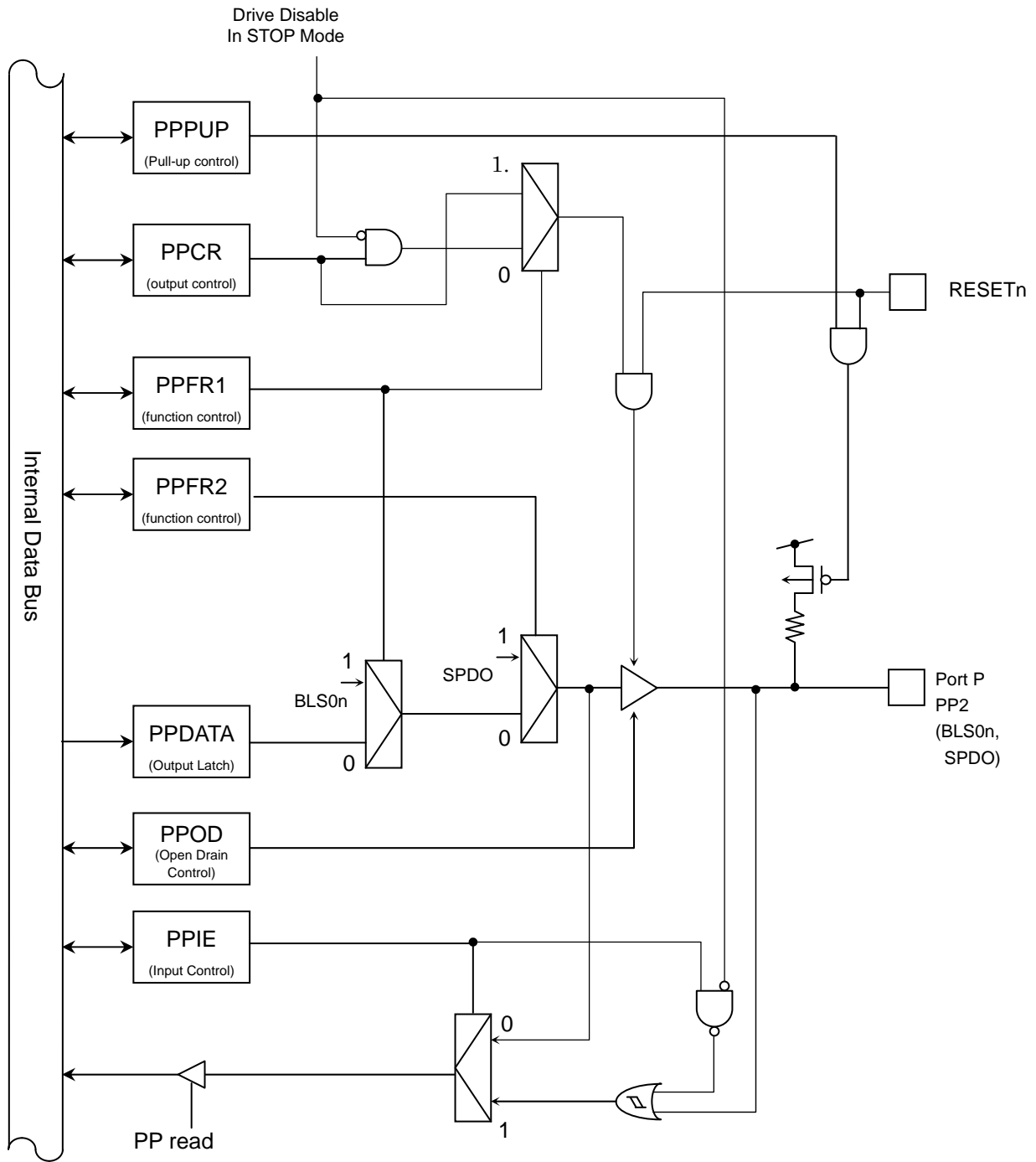
Fig3.6.54   PortP (PP2)
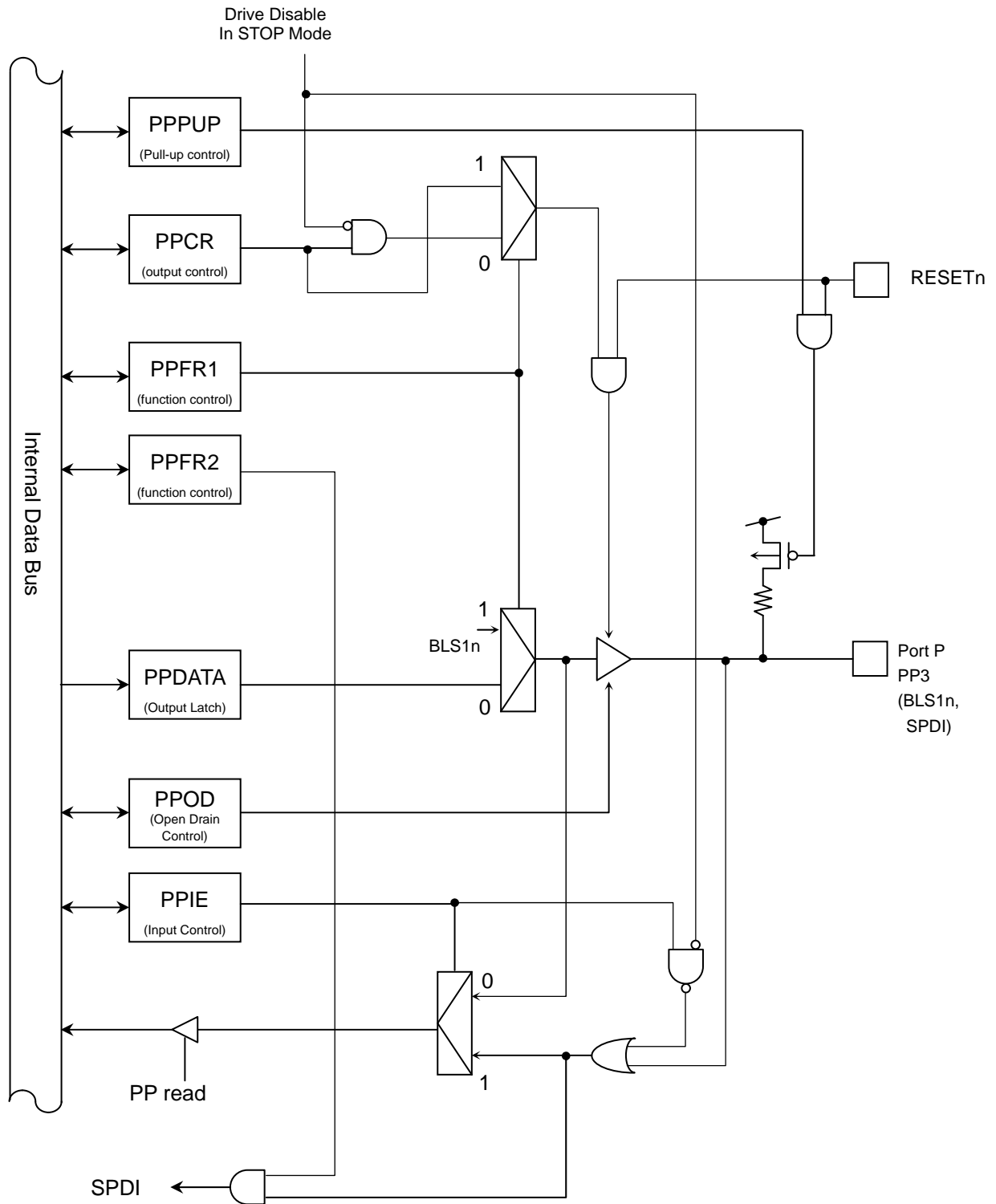
Fig3.6.55   PortP (PP3)
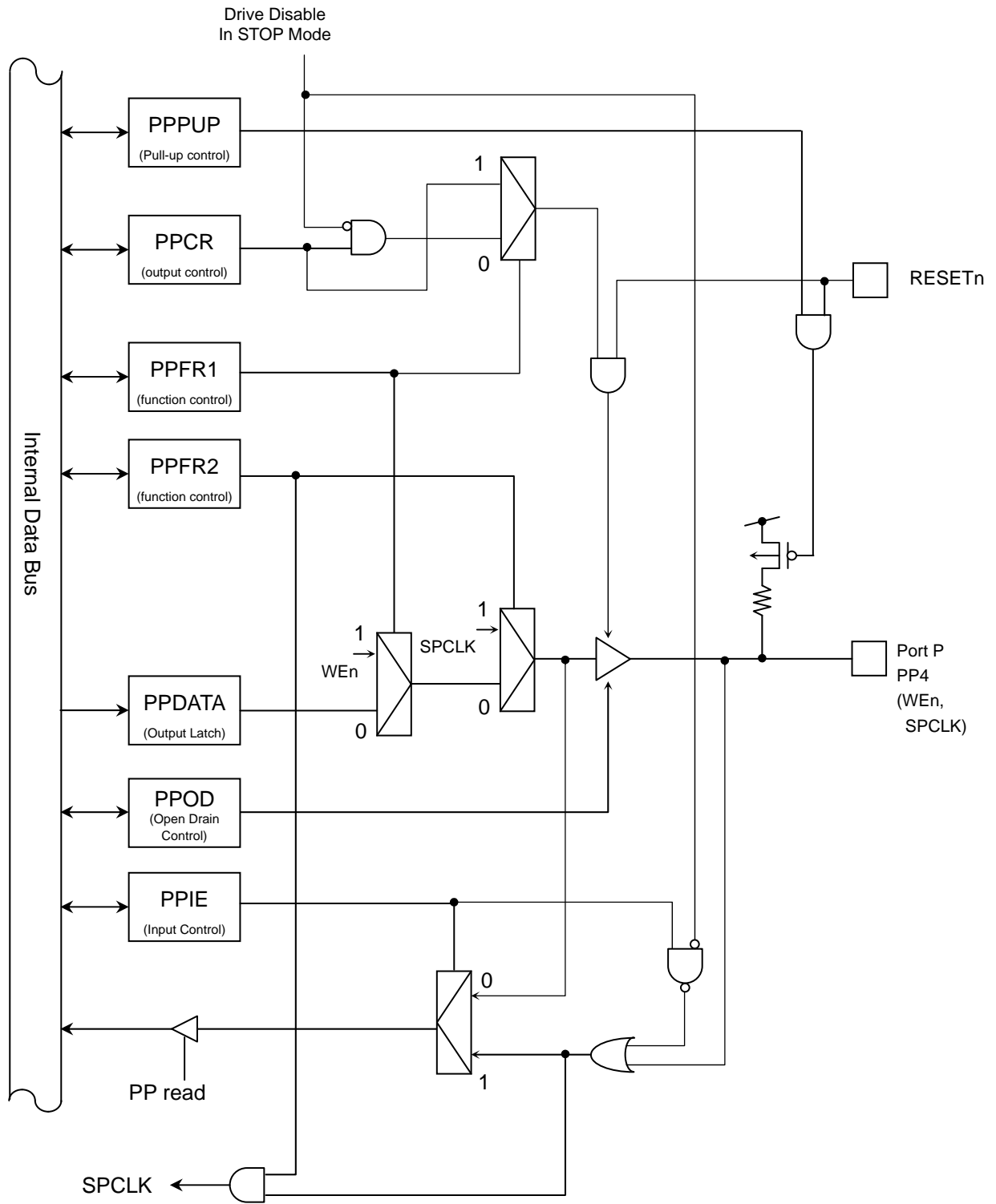
Fig3.6.56   PortP (PP4)
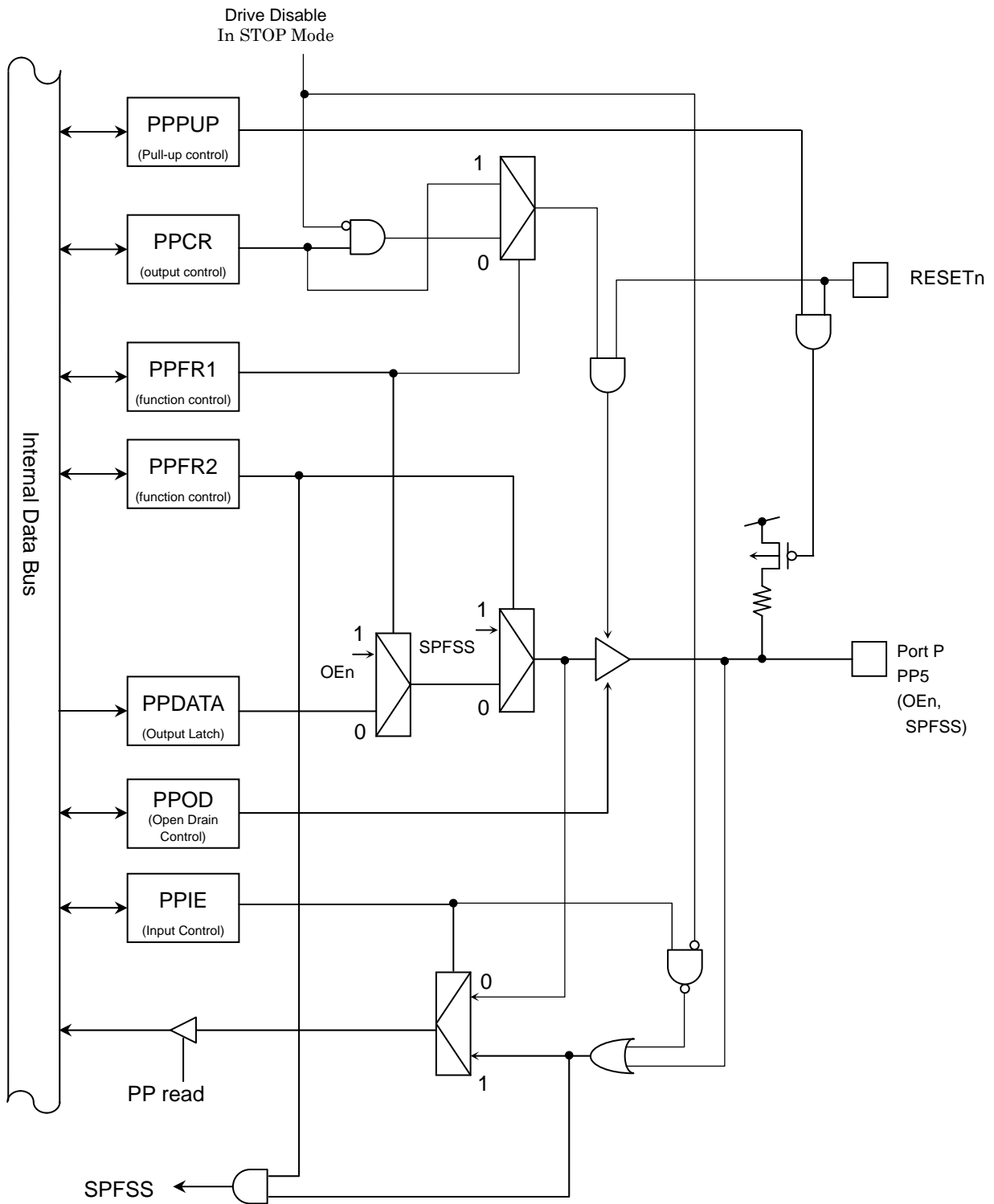
Fig3.6.57   PortP (PP5)
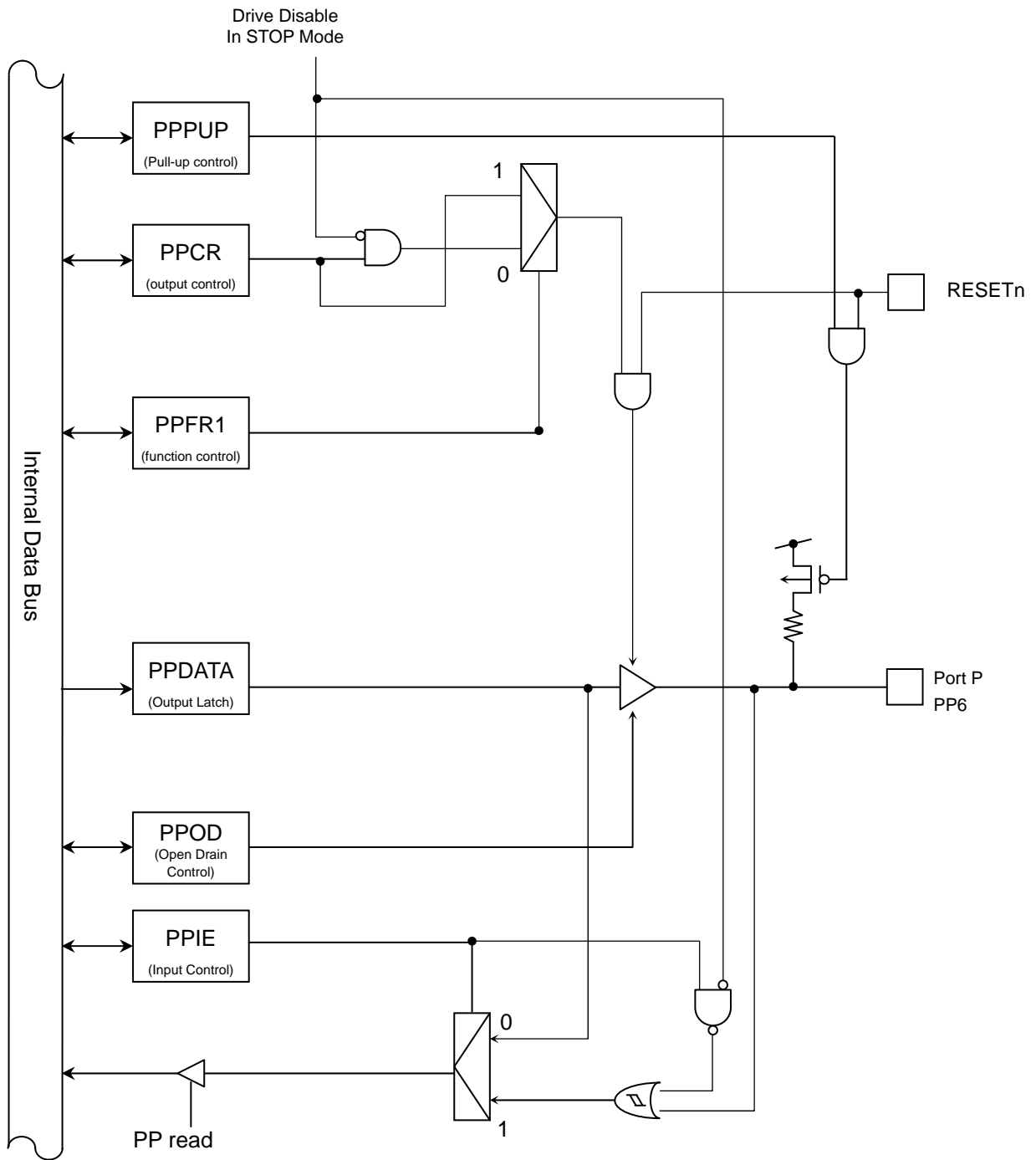
Fig3.6.58   PortP (PP6)

3.7 DMA Controller (DMAC)

3.7.1 Overview

The DMA Controller (DMAC) has the following features:

Table 3.7.1   DMAC Functional Summary

Item	Function		Description
Number of channels	2 channels		
	Hardware start		Can be used for DMA requests from peripheral IPs.
	Software start		Start by a write to the MACSoftBReq register
Bus master	32 bits × 1 (AHB)		
Priority	DMA channel 0 (high) , DMA channel 1 (low)		Hardware-fixed
FIFO	4 words × 2 channels		
Bus width	8/16/32 bits		Can be programmed independently for source and destination.
Burst size	1, 4, 8, 16, 32, 64, 128, 256		
Number of transfers	Up to 4095 times		
Address	Source address	Increment/ non-increment	It is programmable whether or not to increment the source and destination addresses after each transfer. (Address wrapping is not supported.)
	Destination address	Increment/ non-increment	
Endian	Only little endian is supported.		
Transfer type	Peripheral-to-memory (register-to-memory) Memory-to-peripheral (memory-to-register) Memory-to-memory (Note)		DMA hardware start is not supported for memory-to-memory transfers. For details, see the description of the DMACCxConfiguration register.
Interrupt function	Terminal count interrupt Error interrupt		
Special function	Scatter/gather function		

* 1 word = 32 bits

(Note)   Peripheral-to-peripheral (register-to-register) transfers are not supported.

3.7.2   DMA Transfer Types

Table 3.7.1   DMA Transfer Types

	DMA Transfer Direction	DMA Requested by	Supported DMA Request Type (Note 3)	Description
1	Memory-to-peripheral	Peripheral	Burst request	1) All transactions are conducted with burst requests. 2) For single requests, set the burst size to 1.
2	Peripheral-to-memory	Peripheral	Burst request/ Single request (Note 1)	When the data size is not a multiple of the burst size, both burst and single requests are used. If the size of remaining data $\geq$ burst size, then a burst request is used. If the size of remaining data $<$ burst size, then a single request is used.
3	Memory-to-memory	DMAC	None	Start condition: Enabling the DMAC starts data transfer with no DMAC request required. Stop condition: The specified data transfer has been completed, or the DMAC channel is disabled. (Note 2)

(Note 1) Peripheral supporting single requests: SSP

(Note 2) It is recommended to use the low-priority channel (DMAC1) for transferring large amounts of data by memory-to-memory transfer. This enables another AHB master to take ownership of the bus before the ongoing DMA transfer has completed. If the high-priority channel is used, other AHB masters have to wait until the transfer completes.

(Note 3) For the DMA request type supported for each peripheral, see Table 3.7.3 on the next page.

3.7.3   Block Diagram

AD conversion end [15]
— [14]
SSP receive [13]
SSP transmit [12]
SIO11 transmit/receive [11]
SIO10 transmit/receive [10]
SIO9 transmit/receive [9]
SIO8 transmit/receive [8]
SIO7 transmit/receive [7]
SIO6 transmit/receive [6]
SIO5 transmit/receive [5]
SIO4 transmit/receive [4]
SIO3 transmit/receive [3]
SIO2 transmit/receive [2]
SIO1 transmit/receive [1]
SIO0 transmit/receive [0]

— [15]
— [14]
SSP receive [13]
— [12]
— [11]
— [10]
— [9]
— [8]
— [7]
— [6]
— [5]
— [4]
— [3]
— [2]
— [1]
— [0]

CPU data

Burst request

Single request

AHB slave I/F

Control logic and register

DMA request and response I/F

Channel logic and register

AHB master I/F

Interrupt request

AHB

INTS[1] (DMACINTERR)

INTS[2] (DMACINTTC)

DMACCLR[15:0]

Figure3.7.1   DMAC Block Diagram

Table 3.7.1   DMA Request Number List

DMA Request Number	Corresponding Peripheral	
	Burst	Single
0	SIO0 transmission/reception	–
1	SIO1 transmission/reception	–
2	SIO2 transmission/reception	–
3	SIO3 transmission/reception	–
4	SIO4 transmission/reception	–
5	SIO5 transmission/reception	–
6	SIO6 transmission/reception	–
7	SIO7 transmission/reception	–
8	SIO8 transmission/reception	–
9	SIO9 transmission/reception	–
10	SIO10 transmission/reception	–
11	SIO11 transmission/reception	–
12	SSP transmission	–
13	SSP reception	SSP reception
14	–	–
15	AD conversion end	–

3.7.4 Description of Registers

3.7.4.1 DMAC Register List

Table 3.7.1 lists the DMAC registers.

Table 3.7.1   DMAC Register List   Base address = 0x4000_0000

RegisterName	Address (base+)	Description
DMACIntStaus	0x0000	DMAC Interrupt Status Register
DMACIntTCStatus	0x0004	DMAC Interrupt Terminal Count Status Register
DMACIntTCClear	0x0008	DMAC Interrupt Terminal Count Clear Register
DMACIntErrorStatus	0x000C	DMAC Interrupt Error Status Register
DMACIntErrClr	0x0010	DMAC Interrupt Error Clear Register
DMACRawIntTCStatus	0x0014	DMAC Raw Interrupt Terminal Count Status Register
DMACRawIntErrorStatus	0x0018	DMAC Raw Error Interrupt Status Register
DMACEnbldChns	0x001C	DMAC Enabled Channel Register
DMACSoftBReq	0x0020	DMAC Software Burst Request Register
DMACSoftSReq	0x0024	DMAC Software Single Request Register
−	0x0028	Reserved
−	0x002C	Reserved
DMACConfiguration	0x0030	DMAC Configuration Register
−	0x0034	Reserved
DMACC0SrcAddr	0x0100	DMAC Channel0 Source Address Register
DMACC0DestAddr	0x0104	DMAC Channel0 Destination Address Register
DMACC0LLI	0x0108	DMAC Channel0 Linked List Item Register
DMACC0Control	0x010C	DMAC Channel0 Control Register
DMACC0Configuration	0x0110	DMAC Channel0 Configuration Register
DMACC1SrcAddr	0x0120	DMAC Channel1 Source Address Register
DMACC1DestAddr	0x0124	DMAC Channel1 Destination Address Register
DMACC1LLI	0x0128	DMAC Channel1 Linked List Item Register
DMACC1Control	0x012C	DMAC Channel1 Control Register
DMACC1Configuration	0x0130	DMAC Channel1 Configuration Register

Note: The above registers only allow word (32-bit) accesses.

3.7.4.2 DMACIntStatus (DMAC Interrupt Status Register)

Address = (0x4000_0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	IntStatus1	RO	0y0	Status of the DMAC channel 1 interrupt 0y1: Interrupt requested 0y0: Interrupt not requested
[0]	IntStatus0	RO	0y0	Status of the DMAC channel 0 interrupt 0y1: Interrupt requested 0y0: Interrupt not requested

<IntStatus[1:0]>:

These bits indicate the status of the DMAC interrupts after passing through the terminal count interrupt enable and error interrupt enable registers. Either error or terminal count interrupt requests can generate interrupt requests.
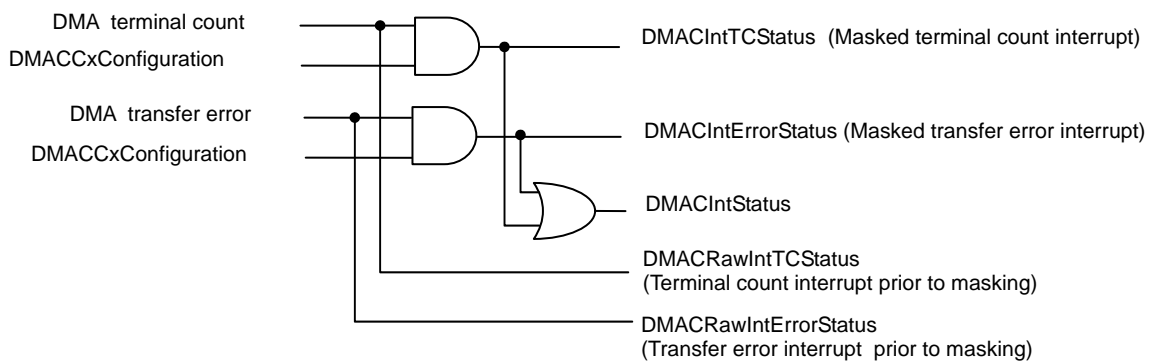
Figure 3.7.2   Block Diagram of Interrupt Logic

3.7.4.3 DMACIntTCStatus (DMAC Interrupt Terminal Count Status Register)

Address = (0x4000_0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	IntTCStatus1	RO	0y0	Status of the DMAC channel 1 terminal count interrupt 0y1: Interrupt request generated 0y0: Interrupt request not generated
[0]	IntTCStatus0	RO	0y0	Status of the DMAC channel 0 terminal count interrupt 0y1: Interrupt request generated 0y0: Interrupt request not generated

<IntTCStatus[1:0]>

These bits indicate the status of the terminal count interrupt after masking.

3.7.4.4 DMACIntTCClear (DMAC Interrupt Terminal Count Clear Register)

Address = (0x4000_0008)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	IntTCClear1	WO	0y0	Clear the DMAC channel 1 terminal count interrupt 0y1: Clear 0y0: No effect
[0]	IntTCClear0	WO	0y0	Clear the DMAC channel 0 terminal count interrupt 0y1: Clear 0y0: No effect

<IntTCClear[1:0]>

Writing a "1" to each bit in this register causes the corresponding bit in the DMACIntTCStatus register to be cleared.

3.7.4.5 DMACIntErrorStatus (DMAC Interrupt Error Status Register)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	IntErrStatus1	RO	0y0	Status of the DMAC channel 1 error interrupt 0y1: Interrupt request generated 0y0: Interrupt request not generated
[0]	IntErrStatus0	RO	0y0	Status of the DMAC channel 0 error interrupt 0y1: Interrupt request generated 0y0: Interrupt request not generated

<IntErrStatus[1:0]>

These bits indicate the status of the error interrupt after masking.

3.7.4.6 DMACIntErrClr (DMAC Interrupt Error Clear Register)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read undefined. Write as zero.
[1]	IntErrClr1	WO	0y0	Clear DMAC channel 1 error interrupt 0y1: Clear 0y0: No effect
[0]	IntErrClr0	WO	0y0	Clear DMAC channel 0 error interrupt 0y1: Clear 0y0: No effect

<IntErrClr[1:0]>

Writing a "1" to each bit in this register causes the corresponding bit in the DMACIntErrClr register to be cleared.

3.7.4.7 DMACRawIntTCStatus (DMAC Raw Interrupt Terminal Count Status Register)

Address = (0x4000_0014)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	RawIntTCS1	RO	0y0	Status of the DMAC channel 1 terminal count interrupt prior to masking 0y1: Interrupt request generated 0y0: Interrupt request not generated
[0]	RawIntTCS0	RO	0y0	Status of the DMAC channel 0 terminal count interrupt prior to masking 0y1: Interrupt request generated 0y0: Interrupt request not generated

3.7.4.8 DMACRawIntErrorStatus (DMAC Raw Error Interrupt Status Register)

Address = (0x4000_0018)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	RawIntErrS1	RO	0y0	Status of the DMAC channel 1 error interrupt prior to masking 0y1: Interrupt request generated 0y0: Interrupt request not generated
[0]	RawIntErrS0	RO	0y0	Status of the DMAC channel 0 error interrupt prior to masking 0y1: Interrupt request generated 0y0: Interrupt request not generated

3.7.4.9 DMACEnbldChns (DMAC Enabled Channel Register)

Address = (0x4000_001C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	EnabledCH1	RO	0y0	DMA channel 1 enable status 0y1: Enabled 0y0: Disabled
[0]	EnabledCH0	RO	0y0	DMA channel 0 enable status 0y1: Enabled 0y0: Disabled

<EnabledCH[1:0]>

"0": Each bit is cleared when a DMA transfer completes in the corresponding channel.

"1": The corresponding DMA channel is enabled.

3.7.4.10 DMACSoftBReq (DMAC Software Burst Request Register)

Address = (0x4000_0020)

Bit	Bit Symbol	Type	Reset Value	Description
[31:15]	–	–	Undefined	Read undefined. Write as zero.
[15]	SoftBReq15	R/W	0y0	Software DMA burst request for AD conversion end 0y1: Generate a DMA burst request 0y0: No effect (WR)
[14]	–	–	Undefined	Read undefined. Write as zero
[13]	SoftBReq13	R/W	0y0	Software DMA burst request for SSP reception 0y1: Generate a DMA burst request 0y0: No effect (WR)
[12]	SoftBReq12	R/W	0y0	Software DMA burst request for SSP transmission 0y1: Generate a DMA burst request 0y0: No effect (WR)
[11]	SoftBReq11	R/W	0y0	Software DMA burst request for SIO11 transmission/reception 0y1: Generate a DMA burst request 0y0: No effect (WR)
[10]	SoftBReq10	R/W	0y0	Software DMA burst request for SIO10 transmission/reception 0y1: Generate a DMA burst request 0y0: No effect (WR)
[9]	SoftBReq9	R/W	0y0	Software DMA burst request for SIO9 transmission/reception 0y1: Generate a DMA burst request 0y0: No effect (WR)
[8]	SoftBReq8	R/W	0y0	Software DMA burst request for SIO8 transmission/reception 0y1: Generate a DMA burst request 0y0: No effect (WR)
[7]	SoftBReq7	R/W	0y0	Software DMA burst request for SIO7 transmission/reception 0y1: Generate a DMA burst request 0y0: No effect (WR)
[6]	SoftBReq6	R/W	0y0	Software DMA burst request for SIO6 transmission/reception 0y1: Generate a DMA burst request 0y0: No effect (WR)
[5]	SoftBReq5	R/W	0y0	Software DMA burst request for SIO5 transmission/reception 0y1: Generate a DMA burst request 0y0: No effect (WR)
[4]	SoftBReq4	R/W	0y0	Software DMA burst request for SIO4 transmission/reception 0y1: Generate a DMA burst request 0y0: No effect (WR)
[3]	SoftBReq3	R/W	0y0	Software DMA burst request for SIO3 transmission/reception 0y1: Generate a DMA burst request 0y0: No effect (WR)
[1]	SoftBReq1	R/W	0y0	Software DMA burst request for SIO2 transmission/reception 0y1: Generate a DMA burst request 0y0: No effect (WR)
[1]	SoftBReq1	R/W	0y0	Software DMA burst request for SIO1 transmission/reception 0y1: Generate a DMA burst request 0y0: No effect (WR)
[0]	SoftBReq0	R/W	0y0	Software DMA burst request for SIO0 transmission/reception 0y1: Generate a DMA burst request 0y0: No effect (WR)

<SoftBReq[15:0]>

You can generate a DMA burst transfer request for each source by writing a "1" to the corresponding bit in this register. Each bit is cleared when the corresponding DMA burst transfer has completed.

(Note) Do not use software and hardware peripheral DMA requests at the same time.

3.7.4.11 DMACSoftSReq (DMAC Software Single Request Register)

Address = (0x4000_0024)

Bit	Bit Symbol	Type	Reset Value	Description
[31:15]	–	–	Undefined	Read undefined. Write as zero.
[13]	SoftSReq13	R/W	0y0	Software DMA single request for SSP reception 0y1: Generate a DMA single request 0y0: No effect (WR)
[12:0]	Reserved	–	Undefined	Read undefined. Write as zero.

<SoftSReq[13]>

You can generate a DMA single transfer request for each source by writing a "1" to the corresponding bit in this register. Each bit is cleared when the corresponding DMA single transfer has completed.

(Note) Do not use software and hardware peripheral DMA requests at the same time.

3.7.4.12 DMACConfiguration (DMAC Configuration Register)

Address = (0x4000_0030)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	M	R/W	0y0	DMA endianness configuration 0: Little endian 1: Reserved
[0]	E	R/W	0y0	DMAC enable 0: Disabled 1: Enabled

<M>

DMA Endian configuration

<E>

The DMA registers cannot be written and read unless the DMAC is enabled. To use DMA operation, the DMAC must always be enabled.

3.7.4.13 DMACC0SrcAddr (DMAC Channel0 Source Address Register)

Address = (0x4000_0100)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	SrcAddr	R/W	0x00000000	DMA transfer source address

<SrcAddr>

Program the DMACCxSrcAddr register before enabling the DMA channel. When the channel is enabled, this register is updated.

When the channel is active, the value read from this register varies with time.

Do not update DMACCxSrcAddr while the channel is active. Before changing the value of this register, you must disable the channel by programming the DMACConfiguration register.

DMACC1SrcAddr (DMAC Channel 1 Source Address Register)

The DMACC1SrcAddr register is functionally identical to the DMACC0SrcAddr register. For the bit assignements and description, see above. For the register address, see Table 3.7.1 DMAC Register List.

3.7.4.14 DMACC0DestAddr (DMAC Channel0 Destination Address Register)

Address = (0x4000_0104)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	DestAddr	R/W	0x00000000	DMA transfer destination address

<DestAddr>

Do not update the DMACC0DestAddr register while the channel is active. Before changing the value of this register, you must disable the channel by programming the DMACCxConfiguration register.

DMACC1DestAddr (DMAC Channel 1 Destination Address Register)

The DMACC1DestAddr register is functionally identical to the DMACC0DestAddr register. For the bit assignments and description, see above. For the register address, see Table 3.7.1 DMAC Register List.

3.7.4.15 DMACC0LLI (DMAC Channel0 Linked List Item Register)

Address = (0x4000_0108)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	LLI	R/W	0x00000000	Address for the next LLI
[1:0]	–	–	Undefined	Read undefined. Write as zero.

<LLI>

The <LLI> value must not exceed 0xFFFF_FFF0.

When <LLI>=0, the current LLI is the last in the chain, and the DMA channel is disabled when all DMA transfers associated with it are completed.

*For details, see "3.7.5 Special Functions".

DMACC1LLI (DMAC Channel 0 Linked List Item Register)

The DMACC1LLI register is functionally identical to the DMACC0LLI register. For the bit assignments and description, see above. For the register address, see Table 3.7.1   DMAC Register List.

3.7.4.16 DMACC0Control (DMAC Channel0 Control Register)

Address = (0x4000_010C)

Bit	Bit Symbol	Type	Reset Value	Description
[31]	I	R/W	0y0	Terminal count interrupt enable when using scatter/gather 0y0: Disable 0y1: Enable
[30:28]	−	−	Undefined	Read undefined. Write as zero.
[27]	DI	R/W	0y0	Transfer destination address increment 0y0: Fixed-address 0y1: Increment
[26]	SI	R/W	0y0	Transfer source address increment 0y0: Fixed-address 0y1: Increment
[25:24]	−	−	Undefined	Read undefined. Write as zero.
[23:21]	Dwidth[2:0]	R/W	0y000	Transfer destination bit width 0y000: Byte (8 bits) 0y001: Halfword (16 bits) 0y010: Word (32 bits) other: Reserved
[20:18]	Swidth[2:0]	R/W	0y000	Transfer source bit width 0y000: Byte (8 bits) 0y001: Halfword (16 bits) 0y010: Word (32 bits) other: Reserved
[17:15]	DBSize[2:0]	R/W	0y000	Transfer destination burst size 0y000: 1 beat 0y001: 4 beats 0y010 8 beats 0y011 16 beats 0y100 32 beats 0y101: 64 beats 0y110: 128 beats 0y111: 256 beats
[14:12]	SBSize[2:0]	R/W	0y000	Transfer source burst size 0y000: 1 beat 0y001: 4 beats 0y010: 8 beats 0y011: 16 beats 0y100: 32 beats 0y101: 64 beats 0y110: 128 beats 0y111: 256 beats
[11:0]	TransferSize	R/W	0x000	Transfer size

<I>

Enables or disables the terminal count interrupt when using scatter/gather

<DI>

When set, the destination address is incremented after each transfer.

<SI>

When set, the source address is incremented after each transfer.

<Dwidth[2:0]>

Transfer destination bit width

<Swidth[2:0]>

Transfer source bit width. The transfer size must be a multiple of the transfer source bit width.

<DBSize[2:0]>

Transfer destination burst size

(Note) The burst size specified in <DBSize> is not related to the AHB HBURST signal.

<SBSize[2:0]>

Transfer source burst size

(Note) The burst size specified in <SBsize> is not related to the AHB HBURST signal.

<TransferSize>

Transfer size. A write to this field sets the size of the transfer when the DMAC is the flow controller.

This value counts down to "0" after each transfer. A read from <TransferSize> provides the number of transfers yet to be performed.

The transfer size must be a multiple of the transfer source bit width.

For example, when <Swidth>=8 bits, the transfer size must be byte-aligned. When <Swidth>=16 bits, the transfer size must be halfword-aligned. When <Swidth>=32 bits, the transfer size must be word-aligned.

(Note)   If the transfer destination bit width is wider than the transfer source bit width, caution is required in setting the transfer size. The transfer size must be set to satisfy the following equation:

Transfer source bit width x Transfer size = Transfer destintaion bit width x N   (N: Integer)

- DMACC1Control (DMAC Channel 1 Control Register)

    The DMACC1Control register is functionally identical to the DMACC0Control register. For the bit assignments and description, see above. For the register address, see Table 3.7.1   DMAC Register List.

3.7.4.17 DMACC0Configuration (DMAC Channel0 Configuration Register)

Address = (0x4000_0110)

Bit	Bit Symbol	Type	Reset Value	Description
[31:19]	–	–	Undefined	Read undefined. Write as zero.
[18]	Halt	R/W	0y0	0y0: Enable DMA requests 0y1: Ignore DMA requests
[17]	Active	RO	0y0	0y0: No data in the FIFO of the channel 0y1: The FIFO of the channel has data.
[16]	Lock	R/W	0y0	0y0: Disable locked transfers 0y1: Enable locked transfers
[15]	ITC	R/W	0y0	Terminal count interrupt enable bit 0y0: Disable 0y1: Enable
[14]	IE	R/W	0y0	Error interrupt enable bit 0y0: Disable 0y1: Enable
[13:11]	FlowCntrl	R/W	0y000	<table><tr><td>FlowCntrl Set value</td><td>Transfer type</td></tr><tr><td>0y000</td><td>Memory-to-memory</td></tr><tr><td>0y001</td><td>Memory-to-peripheral</td></tr><tr><td>0y010</td><td>Peripheral-to-memory</td></tr><tr><td>0y011</td><td>Reserved</td></tr></table> 0y100 to 0y111: Reserved
[10]	–	–	Undefined	Read undefined. Write as zero.
[9:6]	DestPeripheral	R/W	0y000	Transfer destination peripheral (Note 1) 0y000 to 0y1111
[5]	–	–	Undefined	Read undefined. Write as zero.
[4:1]	SrcPeripheral	R/W	0y000	Transfer source peripheral (Note 1) 0y000 to 0y1111
[0]	E	R/W	0y0	Channel enable 0y0: Disable 0y1: Enable

(Note)   See Table 3.7.3 DMA Request Number List.

< Halt>

Controls whether to enable or ignore DMA requests.

< Active>

Indicates whether or not there is data in the FIFO of the channel.

<Lock>

Locked transfer enable (continuous transfer). When locked transfers are enabled, a specified number of bursts are transferred without releasing the bus. For details, see "3.7.5 Special Functions".

< ITC>

Terminal count interrupt enable

< IE>

Error interrupt enable

< FlowCntrl>

Transfer type

     0y000: Memory-to-memory

     0y001: Memory-to-peripheral

     0y010: Peripheral-to-memory

     0y011: Reserved

     0y100-0y111: Reserved

(Note) When the transfer type is set to memory-to-memory, DMA hardware start is not supported. A transfer is started by wriring a "1" to the <E> bit in the DMACC0Configuration register.

<DestPeripheral>

Transfer destination peripheral. The binary value in this field selects the DMA destination request peripheral. This field is ignored if the destination of the transfer is memory.

<SrcPeripheral>

Transfer source peripheral. The binary value in this field selects the DMA source request peripheral. This field is ignored if the source of the transfer is memory.

<E>

Channel enable. This bit can be used to enable or disable a channel. If a channel is disabled while a transfer is in progress, any data in the channel's FIFO is lost. When restarting the channel, you must fully re-initialize the channel.

If you want to temporarily disable a channel, you must set the <Halt> bit so that subsequent DMA

requests are ignored. You must then poll the <Active> bit until it reaches "0". Finally, you can clear the <E> bit.

- DMACC1Configuration (DMAC Channel 1 Configuration Register)

    The DMACC1Configuration register is functionally identical to the DMACC0Configuration register. For the bit assignments and description, see above. For the register address, see Table 3.7.3 DMAC Register List.

3.7.5   Special Functions

3.7.5.1 Scatter/gather function

If only a certain part of image data is to be transferred, the transfer data is not located at contiguous areas in memory, causing gaps in the transfer source addresses at certain intervals. Since DMA transfers can only handle data at contiguous addresses, you must re-program the transfer settings each time a gap occurs in the sequence of source addresses.

Screen image   Screen data
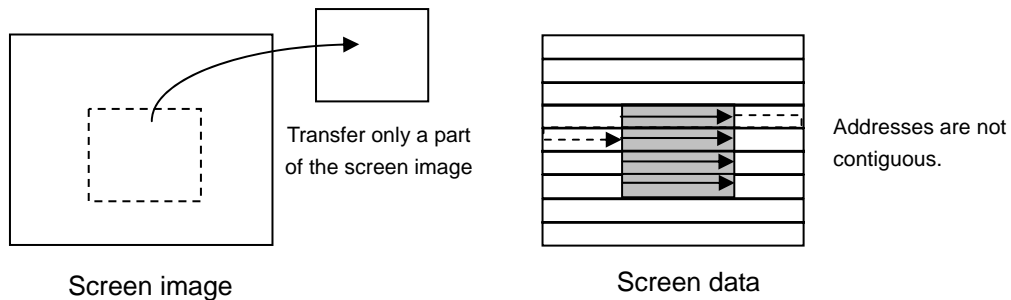
Scatter/gather is realized through the use of "linked lists (LLIs)". Each LLI contains the DMA transfer settings (source address, destination address, transfer size and transfer bus width). At the end of each LLI, another LLI can be loaded to continue the DMA operation without the intervention of the CPU.

Each linked list item contains four words:

1) DMACCxSrcAddr
2) DMACCxDestAddr
3) DMACCxLLI
4) DMACCxControl

An interrupt can be generated at the end of each LLI depending on the terminal count bit in the DMACCxControl register. This bit enables you to interrupt transfers using LLIs depending on a user-specified condition so that execution can be branched as required. To clear the interrupt, you must write a "1" to the corresponding bit in the DMACIntTCClear register.

3.7.5.2 Liked list operation

To use the scatter/gather function, you need to create a series of linked lists (LLIs) to define source and destination data areas. Each LLI controls the transfer of one block of data. Each LLI contains the settings for a normal DMA transfer. The use of LLIs allows multiple uncontiguous blocks of data to be transferred continuously. At the end of each LLI, another LLI can be loaded to continue the DMA operation (daisy-chaining).

The following shows an example of how to set linked list operation.

1. The first DMA transfer is directly programmed into the relevant DMA registers.
2. The LLI for the second or subsequent DMA transfer is written at the memory address specified by "next LLI AddressX".
3. To end the linked list operation with the Nth DMA transfer, set "nextLLIAddressX" to 0x00000000.

Transfer source memory image

Destination memory image
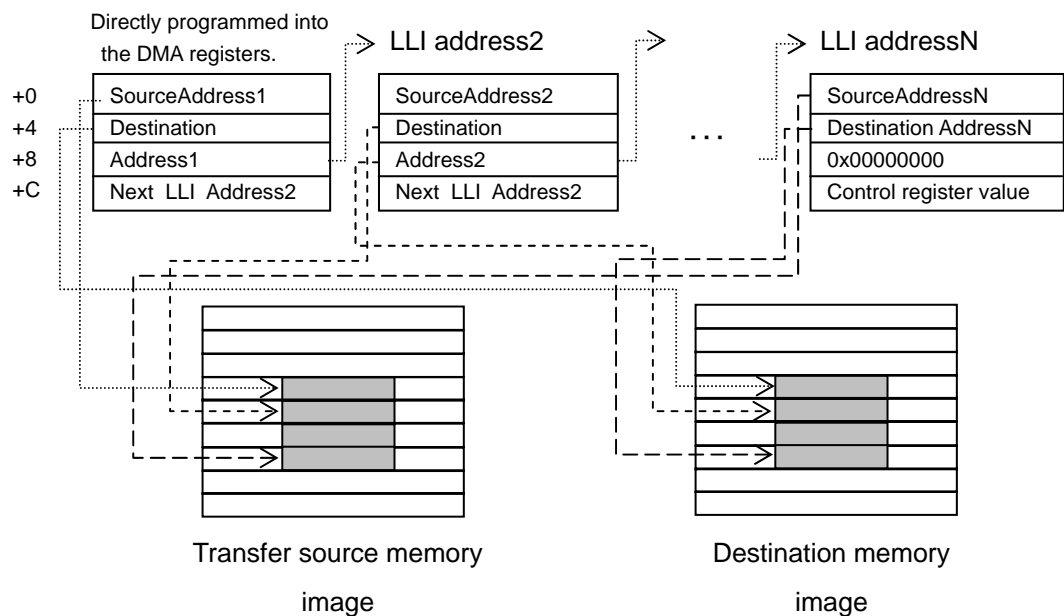
Example: The following shows an example of an LLI for transferring a rectangle of memory.
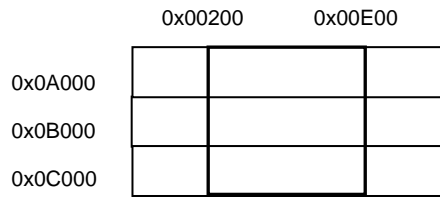
DMACCxSrcAddr: 0x0A200
DMACCxDestAddr: Destination address 1
DMACCxLLI :   0x200000
DMACCxControl :   Set the number of burst transfers, etc.

Linked List
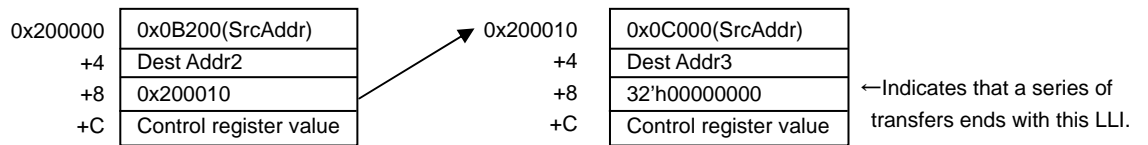
3.8 Static Memory Controller

This LSI contains static memory (NOR type Flash memory and SRAM) controller with asynchronous access.

（Note） Execute the WFI instruction after confirming the external memory access is completed

---

3.8.1 Function Overview

Outline function is showed in Table 3.8.1.

Table 3.8.1 Out line function of Static memory Controller

Item	Function
Supported Memory type	Asynchronous access memory (NOR Flash memory, SRAM, etc) It supports a separate bus.
Data bus width	16bit data bus width
Memory map	64MB access area is supported and divide into four CS signal and area. CS0：0x6000_0000～0x60FF_FFFF (16MB) CS1：0x6100_0000～0x61FF_FFFF (16MB) CS2：0x6200_0000～0x62FF_FFFF (16MB) CS3：0x6300_0000～0x63FF_FFFF (16MB)
Timing adjustment	Can be controlled AC timing by registers.
Clock	fsys x 1/2
External Contorol signals	Separate bus D0 to D15, A1 to A23, OEn, WEn, CS0n to CS3n, BLS0n, BLS1n

3.8.2   Block diagram

The following shows the block diagram of the SMC circuit.
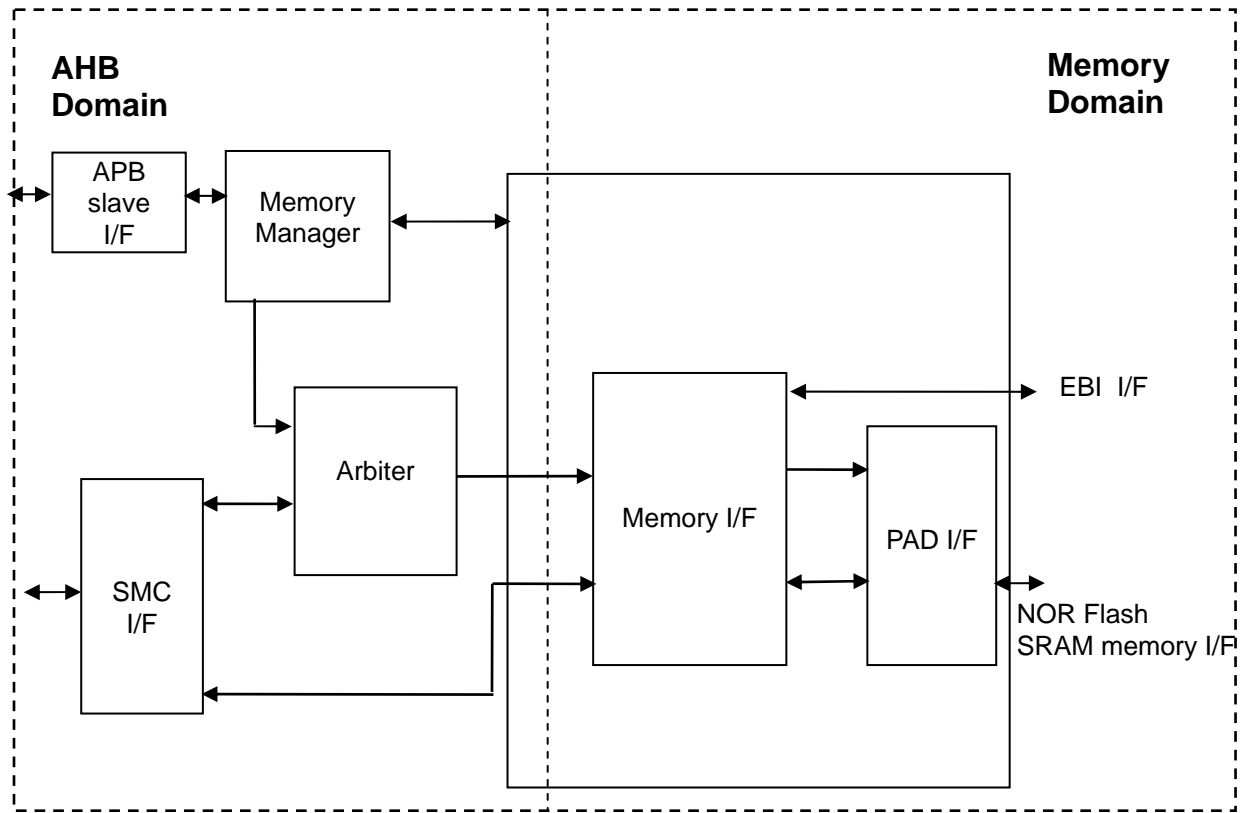
Figure 3.8.1 Block diagram

3.8.3　Description of Registers

3.8.3.1　SFR List

The following lists the SFRs and their functions
Reading and writing the register are the accesses of 32 bits (word) only.

Table 3.8.2 SFR List (1/2)　Base Address = 0x4000_1000

Register Name	Address (Base+)	Type	Description
─	0x0000	─	Reserved
smc_memif_cfg	0x0004	RO	SMC Memory Interface Configuration Register
─	0x0008	─	Reserved
─	0x000C	─	Reserved
smc_direct_cmd	0x0010	WO	SMC Direct Command Register
smc_set_cycles	0x0014	WO	SMC Set Cycles Register
smc_set_opmode	0x0018	WO	SMC Set Opmode Register
─	0x0020	─	Reserved
smc_sram_cycles0_0	0x0100	RO	SMC SRAM Cycles Registers <0>
smc_opmode0_0	0x0104	RO	SMC Opmode Registers <0>
smc_sram_cycles0_1	0x0120	RO	SMC SRAM Cycles Registers <1>
smc_opmode0_1	0x0124	RO	SMC Opmode Registers <1>
smc_sram_cycles0_2	0x0140	RO	SMC SRAM Cycles Registers <2>
smc_opmode0_2	0x0144	RO	SMC Opmode Registers <2>
smc_sram_cycles0_3	0x0160	RO	SMC SRAM Cycles Registers <3>
smc_opmode0_3	0x0164	RO	SMC Opmode Registers <3>
─	0x0200	─	Reserved
─	0x0204	─	Reserved
─	0x0E00	─	Reserved
─	0x0E04	─	Reserved
─	0x0E08	─	Reserved
─	0x0FE0	─	Reserved
─	0x0FE4	─	Reserved
─	0x0FE8	─	Reserved
─	0x0FEC	─	Reserved
─	0x0FF0	─	Reserved
─	0x0FF4	─	Reserved
─	0x0FF8	─	Reserved
─	0x0FFC	─	Reserved

Table 3.8.2 SFR List (2/2)　Base Address = 0x41FF_F000

Register Name	Address (Base+)	Type	Description
SMCMDMODE	0x0100	R/W	SMC Mode Register

Note) Access the registers by using word reads and word writes.

3.8.3.2 SMCMDMODE (SMC MODE Register)

Address = (0x41FF_F000) + (0x0100)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	R/W	Undefined	Read undefined.
[7:1]	–	R/W	0y0	Reserved
[0]	IFSMCMUXMD	R/W	0y0	SMC memory bus mode setting : 0y0 = Separate bus mode 0y1 =Reserved

< IFSMCMUXMD>　　　　　　: SMC memory bus mode setting

　　0y0　　: 　Separate bus mode

　　Others : Setting prohibition

3.8.3.3 smc_memif_cfg (SMC Memory Interface Configuration Register)

Address = (0x4000_1000) + (0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:6]	Reserved	–	Undefined	Write as zero.
[5:4]	memory_width0	RO	0y10	Maximum external SMC memory bus width: 0y00 = Reserved. 0y01 = 16 bits 0y10 = Reserved 0y11 = Reserved
[3:2]	memory_chips0	RO	0y11	The number of supported memory CS: 0y00 = 1 chip 0y01 = 2 chips 0y10 = 3 chips 0y11 = 4 chips
[1:0]	memory_type0	RO	0y01	The number of supported memory types: 0y00 = Reserved 0y01 = SRAM 0y10 = Reserved 0y11 = Reserved

[Description]

　　a.　<memory_width0>

　　　　Maximum external SMC memory bus width:
　　　　0y01 = 16 bits
　　　　Others = Reserved

　　b.　<memory_chips0>

　　　　The number of supported memory CS:
　　　　0y00 = 1 chip
　　　　0y01 = 2 chips
　　　　0y10 = 3 chips
　　　　0y11 = 4 chips

　　c.　<memory_ type0>

　　　　The number of supported memory types:
　　　　0y01 = SRAM
　　　　Others = Reserved

3.8.3.4 smc_direct_cmd (SMC Direct Command Register)

Address = (0x4000_1000) + (0x0010)

Bit	Bit Symbol	Type	Reset Value	Description
[31:26]	Reserved	WO	Undefined	Write as zero.
[25:23]	chip_select	WO	–	CS selection 0y000 = CS0 0y001 = CS1 0y010 = CS2 0y011 = CS3 0y100-0y111 = Reserved
[22:21]	cmd_type	WO	–	Current command 0y00 = Reserved 0y01 = Reserved 0y10 = UpdateRegs 0y11 = Reserved
[20:0]	Reserved	WO	Undefined	Write as zero.

After set set_opmode register and set_cycles register with suitable value, and then write <chip_select> for objective CSx and <cmd_type> for "UpdateReges".

Finally, the setting of set_opmode register and set_cycles register will be valid.

<chip_select>: Select objective Chip Select terminal

    0y000: CS0

    0y001: CS1

    0y010: CS2

    0y011: CS3

    Others: reserved

<cmd_type>: Update set_opmode register and set_cycles register values

    0y10: UpdateRegs

    Others: reserved

3.8.3.5 smc_set_cycles (SMC Set Cycles Register)

Address = (0x4000_1000) + (0x0014)

Bit	Bit Symbol	Type	Reset Value	Description
[31:23]	−	−	Undefined	Write as zero.
[22:20]	Reserved	WO	−	Write as zero.
[19:17]	Set_t5	WO	−	tTR setting value: 0y001 to 0y111
[16:14]	Set_t4	WO	−	tPC setting value: 0y001 to 0y111
[13:11]	Set_t3	WO	−	tWP setting value: 0y001 to 0y111
[10:8]	Set_t2	WO	−	tCEOE setting value: 0y001 to 0y111
[7:4]	Set_t1	WO	−	tWC setting value: 0y0011 to 0y1111
[3:0]	Set_t0	WO	−	tRC setting value: 0y0010 to 0y1111

This register is provided to adjust the access cycle of static memory and should be set to satisfy the A.C. specifications of the memory to be used. To validate SMC set cycles register settings, it is necessary to execute UpdateRegs command on smc_direct_cmd register.

<Set_t5> Set value of t_{TR} (holding register)

0y000: reserved

0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<Set_t4> Set value of t_{PC} (holding register)

0y000: reserved

0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<Set_t3> Set value of t_{WP} (holding register)

0y000: reserved

0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<Set_t2>: Set value of t_{CEOE} (holding register)

0y000: reserved

0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<Set_t1> Set value of t_{WC} (holding register)

0y0000: reserved

0y0011 to 0y1111: SMCCLK×3clk to SMCCLK×15 clk

<Set_t0> Set value of t_{RC} (holding register)

0y0000: reserved

0y0010 to 0y1111: SMCCLK×2clk to SMCCLK×15 clk

3.8.3.6   smc_set_opmode (SMC Set Opmode Register)

Address = (0x4000_1000) + (0x0018)

Bit	Bit Symbol	Type	Reset Value	Description
[31:12]	–	–	Undefined	Write as zero.
[11]	Reserved	WO	–	Write as zero.
[10:6]	Reserved	WO	Undefined	Write as zero.
[5:3]	set_rd_bl	WO	–	Read burst length 0y000 = 1 beat 0y001 = 4 beats 0y010~0y111 = Reserved
[2]	Reserved	WO	Undefined	Write as zero.
[1:0]	set_mw	WO	–	Holding register of the memory data bus width set value 0y00 = Reserved 0y01 = 16bits 0y10 = Reserved 0y11 = Reserved

To validate SMC set opmode register settings, it is necessary to execute UpdateRegs command on smc_direct_cmd register.

<set_rd_bl>：Setting bits for Burst length of data read

<set_mw>：Setting bits for Data bus width. Always set to 0y01 in this LSI

3.8.3.7   smc_sram_cycles0_0 (SMC SRAM Cycles Registers 0 <0>)

Address = (0x4000_1000) + (0x0100)

Bit	Bit Symbol	Type	Reset Value	Description
[31:20]	–	–	Undefined	Read undefined. Write as zero.
[19:17]	t_tr	RO	0y001	tTR setting value: 0y001 to 0y111
[16:14]	t_pc	RO	0y010	tPC setting value: 0y001 to 0y111
[13:11]	t_wp	RO	0y110	tWP setting value: 0y001 to 0y111
[10:8]	t_ceoe	RO	0y011	tCEOE setting value: 0y001 to 0y111
[7:4]	t_wc	RO	0y1100	tWC setting value: 0y0011 to 0y1111
[3:0]	t_rc	RO	0y1100	tRC setting value: 0y0010 to 0y1111

Current cycle setting can be read for chip select 0

<t_tr> : Turn around cycle time
   0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_pc> : Page cycle time
   0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_wp> : WEn pulse cycle time
   0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_ceoe> : delay cycle time to OEn assert
   0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_wc> : Write cycle time
   0y0011 to 0y1111: SMCCLK×3clk to SMCCLK×15 clk

<t_rc> : Read cycle time
   0y0010 to 0y1111: SMCCLK×2clk to SMCCLK×15 clk

3.8.3.8   smc_sram_cycles0_1 (SMC SRAM Cycles Registers 0 <1>)

Address = (0x4000_1000) + (0x0120)

Bit	Bit Symbol	Type	Reset Value	Description
[31:20]	–	–	Undefined	Read undefined. Write as zero.
[19:17]	t_tr	RO	0y001	tTR setting value: 0y001 to 0y111
[16:14]	t_pc	RO	0y010	tPC setting value: 0y001 to 0y111
[13:11]	t_wp	RO	0y110	tWP setting value: 0y001 to 0y111
[10:8]	t_ceoe	RO	0y011	tCEOE setting value: 0y001 to 0y111
[7:4]	t_wc	RO	0y1100	tWC setting value: 0y0011 to 0y1111
[3:0]	t_rc	RO	0y1100	tRC setting value: 0y0010 to 0y1111

Current cycle setting can be read for chip select 1

<t_tr>：Turn around cycle time
    0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_pc>：Page cycle time
    0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_wp>：WEn pulse cycle time
    0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_ceoe>：delay cycle time to OEn assert
    0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_wc>：Write cycle time
    0y0011 to 0y1111: SMCCLK×3clk to SMCCLK×15 clk

<t_rc>：Read cycle time
    0y0010 to 0y1111: SMCCLK×2clk to SMCCLK×15 clk

3.8.3.9 smc_sram_cycles0_2 (SMC SRAM Cycles Registers 0 <2>)

Address = (0x4000_1000) + (0x0140)

Bit	Bit Symbol	Type	Reset Value	Description
[31:20]	–	–	Undefined	Read undefined.
[19:17]	t_tr	RO	0y001	tTR setting value: 0y001 to 0y111
[16:14]	t_pc	RO	0y010	tPC setting value: 0y001 to 0y111
[13:11]	t_wp	RO	0y110	tWP setting value: 0y001 to 0y111
[10:8]	t_ceoe	RO	0y011	tCEOE setting value: 0y001 to 0y111
[7:4]	t_wc	RO	0y1100	tWC setting value: 0y0011 to 0y1111
[3:0]	t_rc	RO	0y1100	tRC setting value: 0y0010 to 0y1111

Current cycle setting can be read for chip select 2

<t_tr>: Turn around cycle time
    0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_pc>: Page cycle time
    0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_wp>: WEn pulse cycle time
    0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_ceoe>: delay cycle time to OEn assert
    0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_wc>: Write cycle time
    0y0011 to 0y1111: SMCCLK×3clk to SMCCLK×15 clk

<t_rc>: Read cycle time
    0y0010 to 0y1111: SMCCLK×2clk to SMCCLK×15 clk

3.8.3.10   smc_sram_cycles0_3 (SMC SRAM Cycles Registers 0 <3>)

Address = (0x4000_1000) + (0x0160)

Bit	Bit Symbol	Type	Reset Value	Description
[31:20]	–	–	Undefined	Read undefined. Write as zero.
[19:17]	t_tr	RO	0y001	tTR setting value: 0y001 to 0y111
[16:14]	t_pc	RO	0y010	tPC setting value: 0y001 to 0y111
[13:11]	t_wp	RO	0y110	tWP setting value: 0y001 to 0y111
[10:8]	t_ceoe	RO	0y011	tCEOE setting value: 0y001 to 0y111
[7:4]	t_wc	RO	0y1100	tWC setting value: 0y0011 to 0y1111
[3:0]	t_rc	RO	0y1100	tRC setting value: 0y0010 to 0y1111

Current cycle setting can be read for chip select 3

<t_tr>： Turn around cycle time
0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_pc>： Page cycle time
0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_wp>：WEn pulse cycle time
0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_ceoe>：delay cycle time to OEn assert
0y001 to 0y111: SMCCLK×1clk to SMCCLK×7 clk

<t_wc>： Write cycle time
0y0011 to 0y1111: SMCCLK×3clk to SMCCLK×15 clk

<t_rc>：Read cycle time
0y0010 to 0y1111: SMCCLK×2clk to SMCCLK×15 clk

3.8.3.11 smc_opmode0_0 (SMC Opmode Registers 0<0>)

Address = (0x4000_1000) + (0x0104)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	address_match	RO	0x60	Read as 0x60
[23:16]	Reserved	RO	0xFF	Read as 0xFF
[15:6]	Reserved	RO	Undefined	Read undefined
[5:3]	rd_bl	RO	0y000	Read burst length 0y000 = 1 beat 0y001 = 4 beats 0y010~0y111 = Reserved
[2]	Reserved	RO	Undefined	Read undefined
[1:0]	mw	RO	0y10	Holding register of the memory data bus width set value 0y00 = Reserved 0y01 = 16bits 0y10 = Reserved 0y11 = Reserved

(Note) Don't access the external memory area except set CS area

Current op_mode setting can be read for chip select 0

<address_match>: Start address of CS0 area (0x60)

<rd_bl>: Burst length of data read

<mw> : Data bus width of CS0

3.8.3.12 smc_opmode0_1 (SMC Opmode Registers 0<1>)

Address = (0x4000_1000) + (0x0124)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	address_match	RO	0x61	Read as 0x61
[23:16]	Reserved	RO	0xFF	Read as 0xFF
[15:6]	Reserved	RO	Undefined	Read undefined
[5:3]	rd_bl	RO	0y000	Read burst length 0y000 = 1 beat 0y001 = 4 beats 0y010~0y111 = Reserved
[2]	Reserved	RO	Undefined	Read undefined
[1:0]	mw	RO	0y10	Holding register of the memory data bus width set value 0y00 = Reserved 0y01 = 16bits 0y10 = Reserved 0y11 = Reserved

(Note) Don't access the external memory area except set CS area

Current op_mode setting can be read for chip select 1

<address_match>: Start address of CS1 area (0x61)

<rd_bl>: Burst length of data read

<mw> : Data bus width of CS1

3.8.3.13 smc_opmode0_2 (SMC Opmode Registers 0<2>)

Address = (0x4000_1000) + (0x0144)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	address_match	RO	0x61	Read as 0x62
[23:13]	Reserved	RO	0xFF	Read as 0xFF
[15:6]	Reserved	RO	Undefined	Read undefined
[5:3]	rd_bl	RO	0y000	Read burst length 0y000 = 1 beat 0y001 = 4 beats 0y010~0y111 = Reserved
[2]	Reserved	RO	Undefined	Read undefined
[1:0]	mw	RO	0y10	Holding register of the memory data bus width set value 0y00 = Reserved 0y01 = 16bits 0y10 = Reserved 0y11 = Reserved

(Note) Don't access the external memory area except set CS area

Current op_mode setting can be read for chip select 2

<address_match>: Start address of CS2 area (0x62)

<rd_bl>: Burst length of data read

<mw> : Data bus width of CS2

3.8.3.14   smc_opmode0_3 (SMC Opmode Registers 0<3>)

Address = (0x4000_1000) + (0x0164)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	address_match	RO	0x61	Read as 0x63
[23:16]	Reserved	RO	0xFF	Read as 0xFF
[15:6]	Reserved	RO	Undefined	Read undefined
[5:3]	rd_bl	RO	0y000	Read burst length 0y000 = 1 beat 0y001 = 4 beats 0y010~0y111 = Reserved
[2]	Reserved	RO	Undefined	Read undefined
[1:0]	mw	RO	0y10	Holding register of the memory data bus width set value 0y00 = Reserved 0y01 = 16bits 0y10 = Reserved 0y11 = Reserved

(Note) Don't access the external memory area except set CS area

Current op_mode setting can be read for chip select 3

<address_match>: Start address of CS3 area (0x63)

<rd_bl>: Burst length of data read

<mw> : Data bus width of CS3

3.8.3.15 External Bus Cycle (Separate bus Mode)

t_{RC} / t_{CEOE} registers setting example

t_{RC} =3, t_{CEOE} =1 (smc_sram_cycles0_0 = 0x0002B1C3)
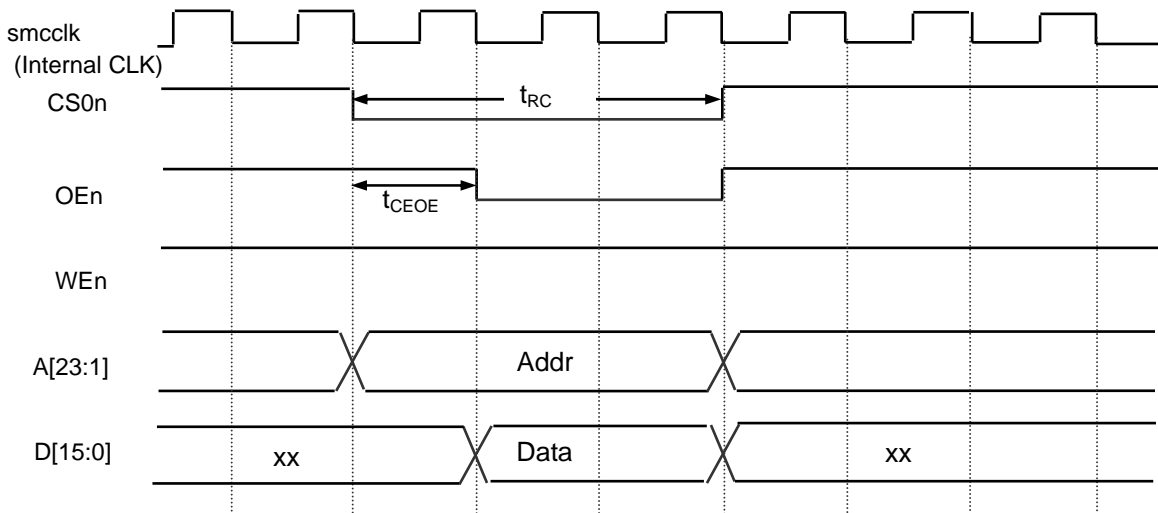
Figure 3.8.2 Asynchronous Read

t_{WC} / t_{WP} Registers setting example

t_{WC} =5, t_{WP} =2 (smc_sram_cycles0_0 = 0x0002935C)
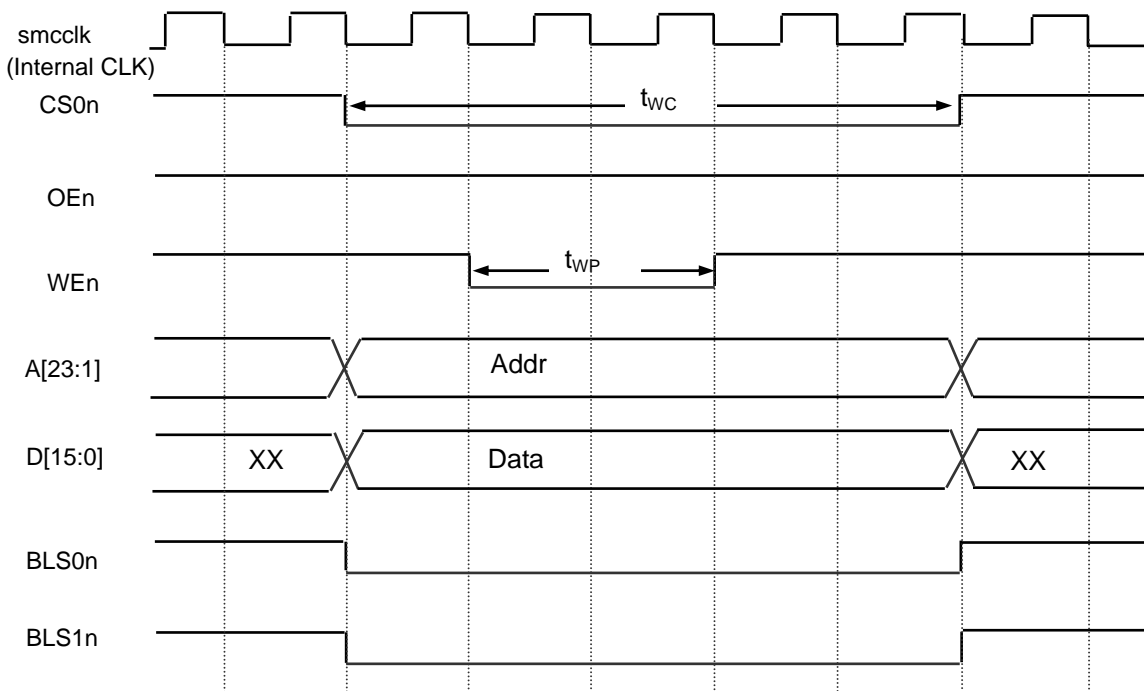
Figure 3.8.3 Asynchronous Write

t_{RC} / t_{CEOE} / t_{PC} Registers setting example

t_{RC} =3, t_{CEOE} =2, t_{PC} =1 (smc_sram_cycles0_0 = 0x000272C3)
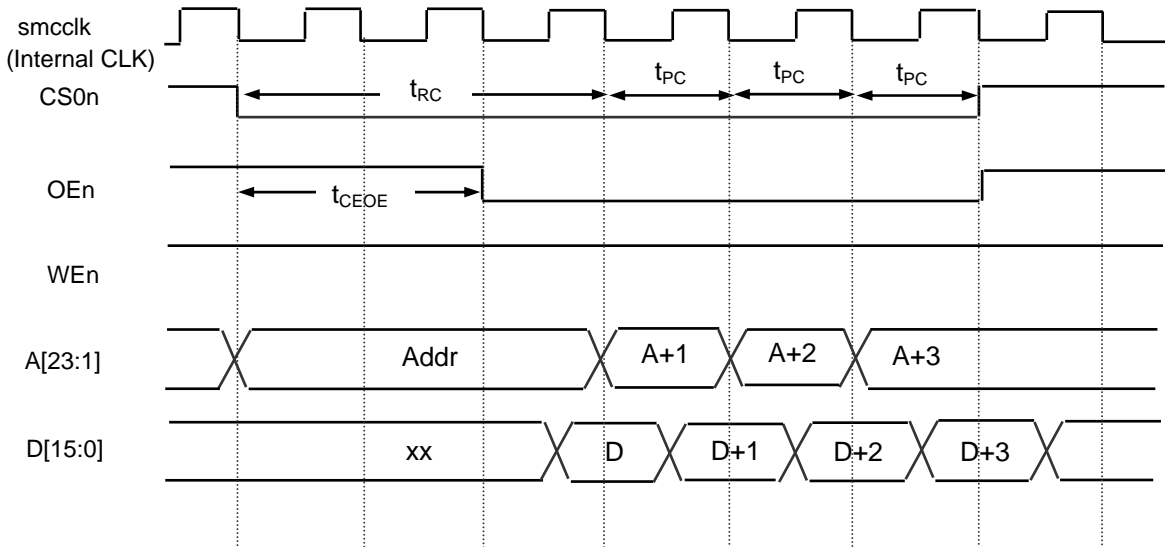
Figure 3.8.4 Asynchronous Page Read

t_{TR} Registers setting example

t_{TR} =1 (smc_sram_cycles0_0 = 0x00029133)
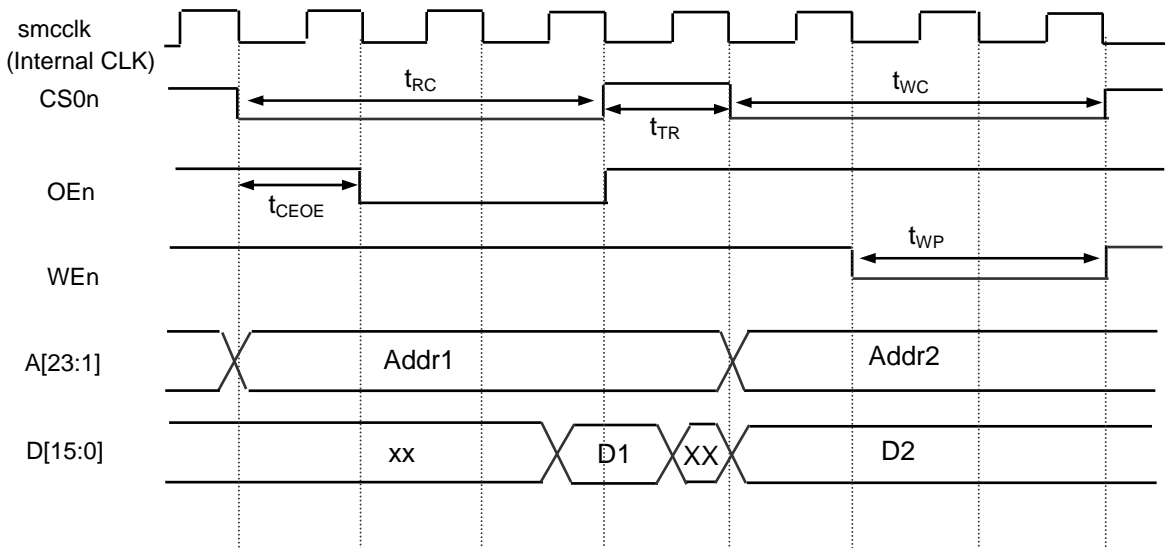
Figure 3.8.5 Asynchronous Read and Asynchronous Write

3.8.4   Connection example for external memory

Below Fig 3.8.6 shows connection example for external 16bits NOR-Flash and 16bits SRAM
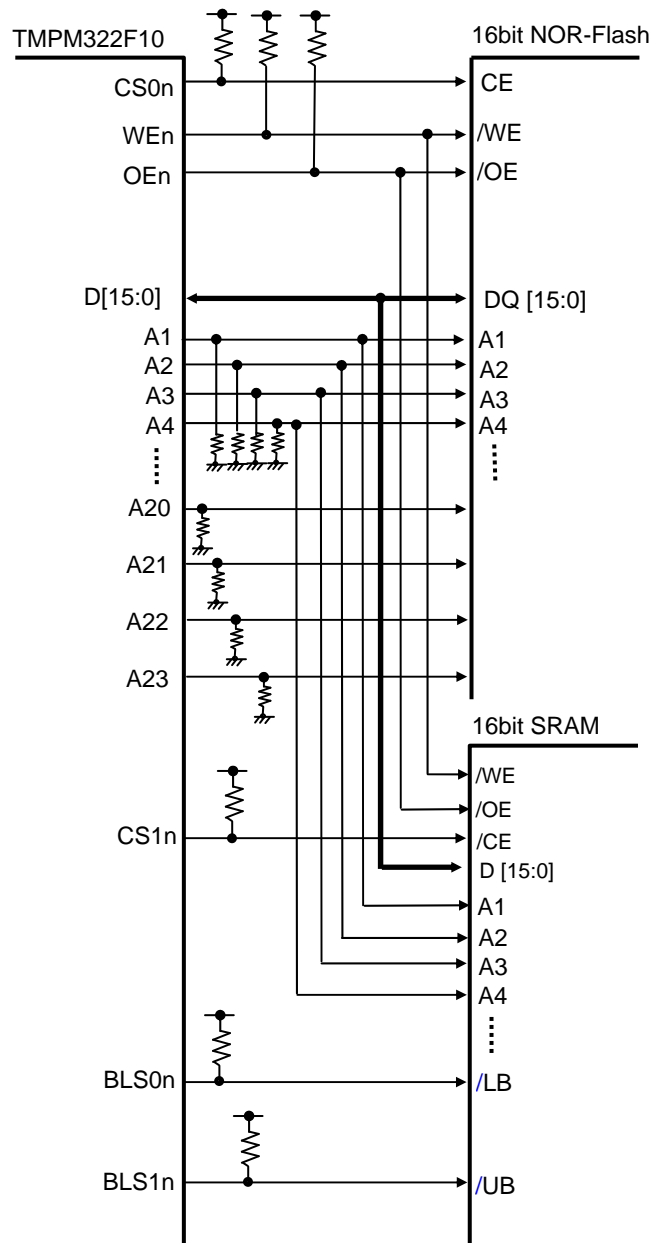
Figure 3.8.6 Connection Example for external 16bit SRAM and NOR-Flash

3.9   16-bit Timer/Event Counters(TMRBs)

3.9.1 Outline

Each of the sixteen channels (TMRB0 through TMRBF) has a multi-functional 16-bit timer/event counter. TMRBs operate in the following four operation modes:

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable square-wave output mode (PPG)
- Timer synchronous mode (capable of setting output mode for each 4ch)

The use of the capture function allows TMRBs to perform the following three measurements

- Frequency measurement
- Pulse width measurement
- Time difference measurement

3.9.2 Differences in the Specifications

Each channel (TMRB0 through TMRBF) functions independently and the channels operate in the same way, except for the differences in their specifications as shown in Table 3.9.1 and Table 3.9.2 and the two-phase pulse count function. Therefore, the operational descriptions here are only for TMRB1.

The channels shown below are used as the capture or start trigger.

(1) The flip-flop output of TMRB0, TMRB4, TMRB8 and TMRBC can be used as the capture trigger of other channels.
- TB4OUT => available for TMRB 1 through TMRB 2
- TB0OUT => available for TMRB 5 through TMRB 7
- TBCOUT => available for TMRB 9 through TMRB B
- TB8OUT => available for TMRB D through TMRB F

(2) The start trigger of the timer synchronous mode (with TBxRUN)
- TMRB0 => can start TMRB 0 through TMRB 3 synchronously
- TMRB4 => can start TMRB 4 through TMRB 7 synchronously
- TMRB8 => can start TMRB 8 through TMRB B synchronously
- TMRBC => can start TMRB C through TMRB F synchronously

Table 3.9.1 Differences in the Specifications of TMRB Modules (1)

Specification / Channel	External pins		Trigger	
	External clock/ capture trigger input pins	Timer flip-flop output pin	Timer for capture triggers	Timer for synchronous stat triggers
TMRB0	–	TB0OUT	–	TB0RUN<TBPRUN> TB0RUN <TBRUN>
TMRB1	TB1IN0 TB1IN1	TB1OUT	TB0OUT	TB0RUN <TBPRUN> TB0RUN <TBRUN>
TMRB2	TB2IN0 TB2IN1	TB2OUT	TB0OUT	TB0RUN <TBPRUN> TB0RUN <TBRUN>
TMRB3	connect to SLK3 internally	TB3OUT	TB0OUT	TB0RUN <TBPRUN> TB0RUN <TBRUN>
TMRB4	–	TB4OUT	–	TB4RUN< TBPRUN> TB4RUN <TBRUN>
TMRB5	TB5IN0 TBIN1	TB5OUT	TB4OUT	TB4RUN <TBPRUN> TB4RUN <TBRUN>
TMRB6	TB6IN0 TB6IN1	TB6OUT	TB4OUT	TB4RUN <TBPRUN> TB4RUN <TBRUN>
TMRB7	TB7IN0 TB7IN1	TB7OUT	TB4OUT	TB4RUN <TBPRUN> TB4RUN <TBRUN>
TMRB8	–	TB8OUT	–	TB8RUN <TBPRUN> TB8RUN <TBRUN>
TMRB9	TB9IN0 TB9IN1	TB9OUT	TB8OUT	TB8RUN <TBPRUN> TB8RUN <TBRUN>
TMRBA	TBAIN0 TBAIN1	TBAOUT	TB8OUT	TB8RUN <TBPRUN> TB8RUN <TBRUN>
TMRBB	TBBIN0 TBBIN1	TBBOUT *connect to RMC internally	TB8OUT	TB8RUN <TBPRUN> TB8RUN <TBRUN>
TMRBC	–	TBCOUT	–	TBCRUN <TBPRUN> TBCRUN <TBRUN>
TMRBD	TBDIN0 TBDIN1	TBDOUT	TBCOUT	TBCRUN <TBPRUN> TBCRUN <TBRUN>
TMRBE	TBEIN0 TBEIN1	TBEOUT	TBCOUT	TBCRUN <TBPRUN> TBCRUN <TBRUN>
TMRBF	TBFIN0 TBFIN1	TBFOUT	TBCOUT	TBCRUN <TBPRUN> TBCRUN <TBRUN>

Table 3.8.2 Differences in the Specifications of TMRB Modules (2)

Specification / Channel	Interrupt	
	Capture interrupt	Capture interrupt
TMRB0	−	INTTB0
TMRB1	INTCAP10 INTCAP11	INTTB1
TMRB2	INTCAP20 INTCAP21	INTTB2
TMRB3	−	INTTB3
TMRB4	−	INTTB4
TMRB5	INTCAP50 INTCAP51	INTTB5
TMRB6	INTCAP60 INTCAP61	INTTB6
TMRB7	INTCAP70 INTCAP71	INTTB7
TMRB8	−	INTTB8
TMRB9	INTCAP90 INTCAP91	INTTB9
TMRBA	INTCAPA0 INTCAPA1	INTTBA
TMRBB	INTCAPB0 INTCAPB1	INTTBB
TMRBC	−	INTTBC
TMRBD	INTCAPD0 INTCAPD1	INTTBD
TMRBE	INTCAPE0 INTCAPE1	INTTBE
TMRBF	INTCAPF0 INTCAPF1	INTTBF

3.9.3 Cofigration

Each channel consists of a 16-bit up-counter, two 16-bit timer registers (double-buffered structure), two 16-bit capture registers, two comparators, a capture input control, a timer flip-flop and its associated control circuit.

Timer operation modes and the timer flip-flop are controlled by a register.
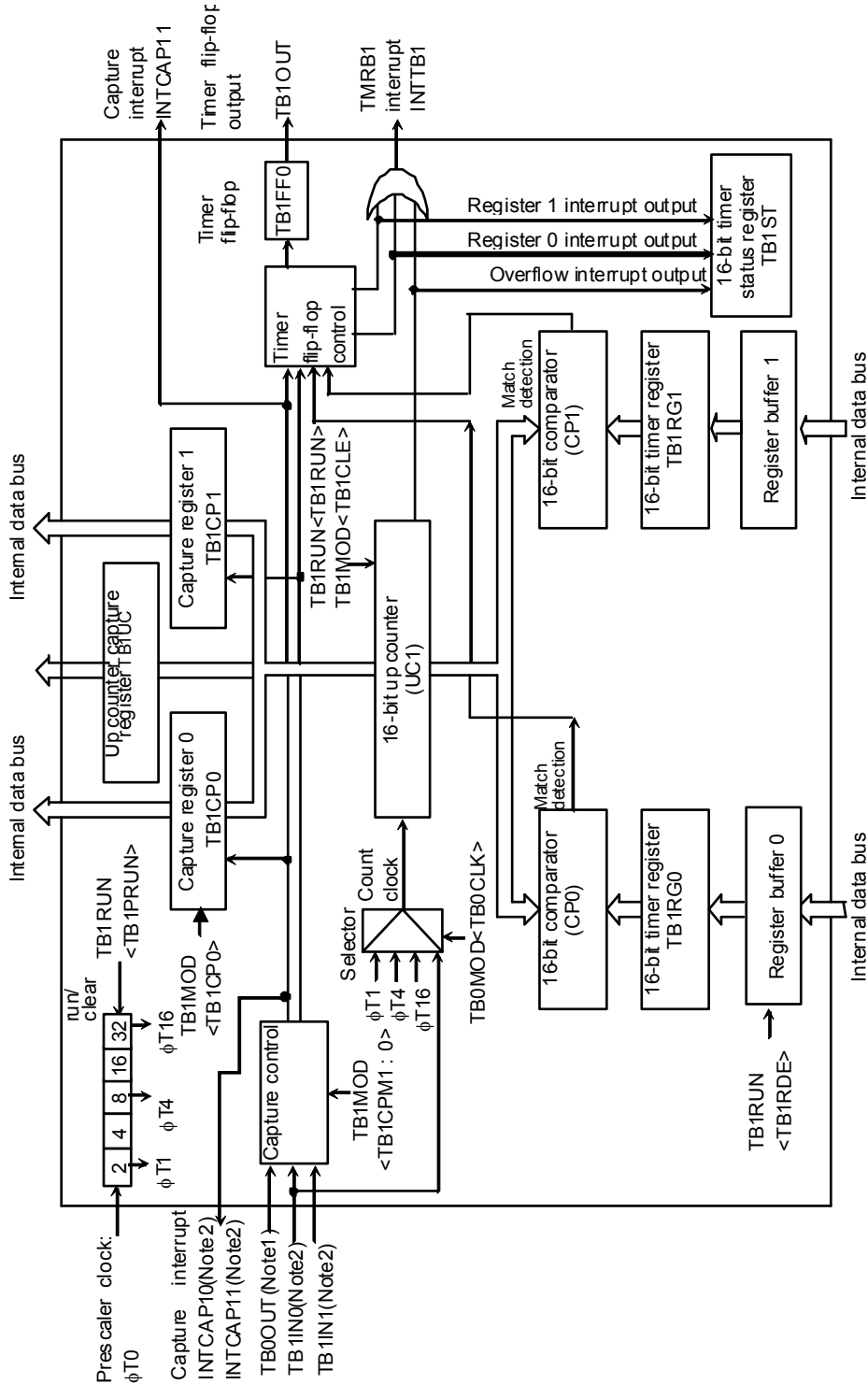
Fig. 3.9.1 TMRB1 Block Diagram (the same applies to channels 0 , 2 and 4 through F)

(Note 1) TB0OUT is input to channels 1 and 2. TB4OUT is input to channels5 through 7. TB8OUT is input to channels9 through A. TBCOUT is input to channelsD through F.

(Note 2) Please note that channels 0, 4, 8 and C :
- do not input TBIN0 and TBIN1 from an external pin.
- cannot use capture interrupt of INTCAPn0 and INTCAPn1.
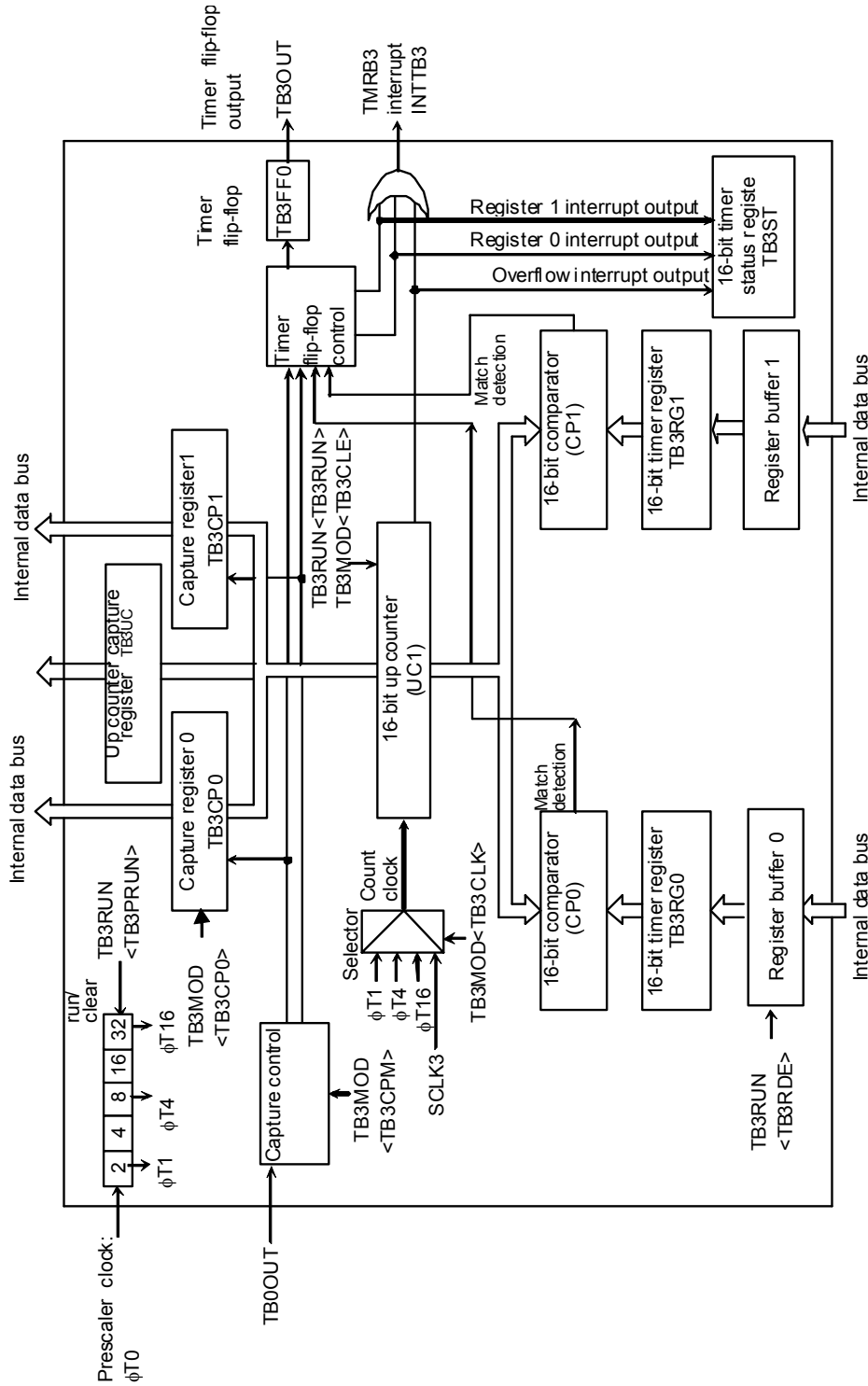
Fig. 3.8.2 TMRB3 Block Diagram

(Note) TMRB3 can use SCLK3(Serial Clock channel 3) as Timer count clock.

3.9.4 Registers

3.9.4.1   TMRB registers

Table 3.9.3 shows the register names and addresses of each channel.

Table 3.9.3 TMRB registers I

Specification \ Channel		TMRB0		TMRB1		TMRB2		TMRB3	
Register names (addresses)	Timer enable register	TB0EN	0x400D_0000	TB1EN	0x400D_0100	TB2EN	0x400D_0200	TB3EN	0x400D_0300
	Timer RUN register	TB0RUN	0x400D_0004	TB1RUN	0x400D_0104	TB2RUN	0x400D_0204	TB3RUN	0x400D_0304
	Timer control register	TB0CR	0x400D_0008	TB1CR	0x400D_0108	TB2CR	0x400D_0208	TB3CR	0x400D_0308
	Timer mode register	TB0MOD	0x400D_000C	TB1MOD	0x400D_010C	TB2MOD	0x400D_020C	TB3MOD	0x400D_030C
	Timer flip-flop control register	TB0FFCR	0x400D_0010	TB1FFCR	0x400D_0110	TB2FFCR	0x400D_0210	TB3FFCR	0x400D_0310
	Timer status register	TB0ST	0x400D_0014	TB1ST	0x400D_0114	TB2ST	0x400D_0214	TB3ST	0x400D_0314
	Interrupt mask register	TB0IM	0x400D_0018	TB1IM	0x400D_0118	TB2IM	0x400D_0218	TB3IM	0x400D_0318
	Timer up counter register	TB0UC	0x400D_001C	TB1UC	0x400D_011C	TB2UC	0x400D_021C	TB3UC	0x400D_031C
	Timer register	TB0RG0	0x400D_0020	TB1RG0	0x400D_0120	TB2RG0	0x400D_0220	TB3RG0	0x400D_0320
		TB0RG1	0x400D_0024	TB1RG1	0x400D_0124	TB2RG1	0x400D_0224	TB3RG1	0x400D_0324
	Capture register	TB0CP0	0x400D_0028	TB1CP0	0x400D_0128	TB2CP0	0x400D_0228	TB3CP0	0x400D_0328
		TB0CP1	0x400D_002C	TB1CP1	0x400D_012C	TB2CP1	0x400D_022C	TB3CP1	0x400D_032C

Specification \ Channel		TMRB4		TMRB5		TMRB6		TMRB7	
Register names (addresses)	Timer enable register	TB4EN	0x400D_0400	TB5EN	0x400D_0500	TB6EN	0x400D_0600	TB7EN	0x400D_0700
	Timer RUN register	TB4RUN	0x400D_0404	TB5RUN	0x400D_0504	TB6RUN	0x400D_0604	TB7RUN	0x400D_0704
	Timer control register	TB4CR	0x400D_0408	TB5CR	0x400D_0508	TB6CR	0x400D_0608	TB7CR	0x400D_0708
	Timer mode register	TB4MOD	0x400D_040C	TB5MOD	0x400D_050C	TB6MOD	0x400D_060C	TB7MOD	0x400D_070C
	Timer flip-flop control register	TB4FFCR	0x400D_0410	TB5FFCR	0x400D_0510	TB6FFCR	0x400D_0610	TB7FFCR	0x400D_0710
	Timer status register	TB4ST	0x400D_0414	TB5ST	0x400D_0514	TB6ST	0x400D_0614	TB7ST	0x400D_0714
	Interrupt mask register	TB4IM	0x400D_0418	TB5IM	0x400D_0518	TB6IM	0x400D_0618	TB7IM	0x400D_0718
	Timer up counter register	TB4UC	0x400D_041C	TB5UC	0x400D_051C	TB6UC	0x400D_061C	TB7UC	0x400D_071C
	Timer register	TB4RG0	0x400D_0420	TB5RG0	0x400D_0520	TB6RG0	0x400D_0620	TB7RG0	0x400D_0720
		TB4RG1	0x400D_0424	TB5RG1	0x400D_0524	TB6RG1	0x400D_0624	TB7RG1	0x400D_0724
	Capture register	TB4CP0	0x400D_0428	TB5CP0	0x400D_0528	TB6CP0	0x400D_0628	TB7CP0	0x400D_0728
		TB4CP1	0x400D_042C	TB5CP1	0x400D_052C	TB6CP1	0x400D_062C	TB7CP1	0x400D_072C

Table 3.9.3 TMRB registers 2

Specification \ Channel		TMRB8		TMRB9		TMRBA		TMRBB	
Register names (addresses)	Timer enable register	TB8EN	0x400D_0800	TB9EN	0x400D_0900	TBAEN	0x400D_0A00	TBBEN	0x400D_0B00
	Timer RUN register	TB8RUN	0x400D_0804	TB9RUN	0x400D_0904	TBARUN	0x400D_0A04	TBBRUN	0x400D_0B04
	Timer control register	TB8CR	0x400D_0808	TB9CR	0x400D_0908	TBACR	0x400D_0A08	TBBCR	0x400D_0B08
	Timer mode register	TB8MOD	0x400D_080C	TB9MOD	0x400D_090C	TBAMOD	0x400D_0A0C	TBBMOD	0x400D_0B0C
	Timer flip-flop control register	TB8FFCR	0x400D_0810	TB9FFCR	0x400D_0910	TBAFFCR	0x400D_0A10	TBBFFCR	0x400D_0B10
	Timer status register	TB8ST	0x400D_0814	TB9ST	0x400D_0914	TBAST	0x400D_0A14	TBBST	0x400D_0B14
	Interrupt mask register	TB8IM	0x400D_0818	TB9IM	0x400D_0918	TBAIM	0x400D_0A18	TBBIM	0x400D_0B18
	Timer up counter register	TB8UC	0x400D_081C	TB9UC	0x400D_091C	TBAUC	0x400D_0A1C	TBBUC	0x400D_0B1C
	Timer register	TB8RG0	0x400D_0820	TB9RG0	0x400D_0920	TBARG0	0x400D_0A20	TBBRG0	0x400D_0B20
		TB8RG1	0x400D_0824	TB9RG1	0x400D_0924	TBARG1	0x400D_0A24	TBBRG1	0x400D_0B24
	Capture register	TB8CP0	0x400D_0828	TB9CP0	0x400D_0928	TBACP0	0x400D_0A28	TBBCP0	0x400D_0B28
		TB8CP1	0x400D_082C	TB9CP1	0x400D_092C	TBACP1	0x400D_0A2C	TBBCP1	0x400D_0B2C

Specification \ Channel		TMRBC		TMRBD		TMRBE		TMRBF	
Register names (addresses)	Timer enable register	TBCEN	0x400D_0C00	TBDEN	0x400D_0D00	TBEEN	0x400D_0E00	TBFEN	0x400D_0F00
	Timer RUN register	TBCRUN	0x400D_0C04	TBDRUN	0x400D_0D04	TBERUN	0x400D_0E04	TBFRUN	0x400D_0F04
	Timer control register	TBCCR	0x400D_0C08	TBDCR	0x400D_0D08	TBECR	0x400D_0E08	TBFCR	0x400D_0F08
	Timer mode register	TBCMOD	0x400D_0C0C	TBDMOD	0x400D_0D0C	TBEMOD	0x400D_0E0C	TBFMOD	0x400D_0F0C
	Timer flip-flop control register	TBCFFCR	0x400D_0C10	TBDFFCR	0x400D_0D10	TBEFFCR	0x400D_0E10	TBFFFCR	0x400D_0F10
	Timer status register	TBCST	0x400D_0C14	TBDST	0x400D_0D14	TBEST	0x400D_0E14	TBFST	0x400D_0F14
	Interrupt mask register	TBCIM	0x400D_0C18	TBDIM	0x400D_0D18	TBEIM	0x400D_0E18	TBFIM	0x400D_0F18
	Timer up counter register	TBCUC	0x400D_0C1C	TBDUC	0x400D_0D1C	TBEUC	0x400D_0E1C	TBFUC	0x400D_0F1C
	Timer register	TBCRG0	0x400D_0C20	TBDRG0	0x400D_0D20	TBERG0	0x400D_0E20	TBFRG0	0x400D_0F20
		TBCRG1	0x400D_0C24	TBDRG1	0x400D_0D24	TBERG1	0x400D_0E24	TBFRG1	0x400D_0F24
	Capture register	TBCCP0	0x400D_0C28	TBDCP0	0x400D_0D28	TBECP0	0x400D_0E28	TBFCP0	0x400D_0F28
		TBCCP1	0x400D_0C2C	TBDCP1	0x400D_0D2C	TBECP1	0x400D_0E2C	TBFCP1	0x400D_0F2C

3.9.4.2　TMRBn **enable register (channels 0 through F)**

TMRBx enable register (x=0~F)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit Symbol	TBEN							
Read/Write	R/W	R						
After reset	0	0						
Function	TMRBn operation 0: Disabled 1: Enabled	"0" is read.						

TBxEN (0x400D_0xx0)

＜TBEN＞: Specifies the TMRB operation. When the operation is disabled, no clock is supplied to the other registers in the TMRB module. This can reduce power consumption. (This disables reading from and writing to the other registers.) To use the TMRB, enable the TMRB operation (set to "1") before programming each register in the TMRB module. If the TMRB operation is executed and then disabled, the settings will be maintained in each register.

3.9.4.3 TMRB RUN register (channels 0 through F)

TMRBx RUN register (x=0~F)

TBxRUN
(0x400D_0xx4)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	7	6	5	4	3	2	1	0
bit Symbol						TBPRUN		TBRUN
Read/Write	R					R/W	R	R/W
After reset	0					0	0	0

Function	"0" is read.					Timer Run/Stop Control
						  0: Stop & clear
						  1: Count
						* The first bit can be read as "0."		

＜TBRUN＞　:Controls the TMRBn count operation.

＜TBPRUN＞:Controls the TMRBn prescaler operation.

3.9.4.4 TMRB **control register (channels 0 through F)**

TMRBx control register (x=0~F)

TBxCR
(0x400D_0xx8)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	7	6	5	4	3	2	1	0
bit Symbol	TBWBF		TBSYNC		I2TB			
Read/Write	R/W	R/W	R/W	R	R/W	R		
After reset	0	0	0	0	0	0		
Function	Double Buffer 0: Disabled 1: Enabled	Write "0".	Timer operation mode 0:individual 1:synchronous	"0" is read.	IDLE 0:Stop 1:Operation	"0" is read.		

＜I2TB＞:Controls the operation in the IDLE mode.

＜TBSYNC＞:Controls operation mode of timers.

　　　　　　　"0": timers operate individually.

　　　　　　　"1": timers operate synchronously.

＜TBWBF＞:Controls the enabling/disabling of double buffering.

3.9.4.5 TMRB **mode register**

TMRBx mode register(x=0~2, 4~F)

TBxMOD (0x400D_0xxC)		**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
	bit Symbol								
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							
		23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	bit Symbol								
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							
		15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	bit Symbol								
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	bit Symbol			TBCP0	TBCPM		TBCLE	TBCLK	
	Read/Write	R	R/W	W	R/W				
	After reset	0	0	1	0	0	0	0	0
	Function	"0" is read.	Write "0".	Capture control by software 0: Capture by software 1: Don't care	Capture timing 00: Disable Capture timing 01: TBnIN0 ↑ TbnIN1 ↑ 10: TbnIN0 ↑ TbnIN1 ↓ 11: TBnOUT ↑ TBnOUT ↓		Up-counter control 0: Clear/disable 1: clear/enable	Selects source clock 00: TbnIN0 pin input 01: φT1 10: φT4 11: φT16	

〈TBCLK〉:Selects the TMRBn timer count clock.

〈TBCLE〉:Clears and controls the TMRBn up-counter.

"0": Disables clearing of the up-counter.

"1": Clears up-counter if there is a match with timer register 1 (TBnRG1).

〈TBnCPM1:0〉:Specifies TMRBn capture timing.

"00": Capture disable

"01": Takes count values into capture register 0 (TBnCP0) upon rising of TbnIN0 pin input. Takes count values into capture register 1 (TBnCP1) upon rising of TbnIN1 pin input.

"10": Takes count values into capture register 0 (TBnCP0) upon rising of TbnIN0 pin input. Takes count values into capture register 1 (TBnCP1) upon falling of TbnIN0 pin input.

"11":Takes count values into capture register 0 (TBnCP0) upon rising of 16-bit timer match output (TBxOUT) and into capture register 1 (TBnCP1) upon falling of TBxOUT (TMRB2 and TMRB3:TB0OUT, TMRB5 through TMRB7:TB4OUT, TMRB9 through TMRBB:TB8OUT, TMRBD through TMRBF:TBCOUT).

〈TBCP0〉:Captures count values by software and takes them into capture register 0 (TBnCP0).

(Note 1)	**The value read from bit 5 of TBnMOD is "1".**
(Note 2)	**Input from TbnIN0 and TbnIN1 is available only for channels TMRB1, 2, 5 through 7, 9 through B and D through F.**

TMRB3 mode register

TB3MOD
(0x400D_030C)

		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							
		23	22	21	20	19	18	17	16
	bit Symbol								
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							
		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							
		7	6	5	4	3	2	1	0
	bit Symbol						TB3CLE	TB3CLK1	TB3CLK0
	Read/Write	R	R/W	W	R/W		R/W		
	After reset	0	0	1	0	0	0	0	0
	Function	"0" is read.	Write "0".	Write "1".	Write "00".		Up-counter control 0: Clear/disable 1: clear/enable	Selects source clock 00: SCLK3 pin input 01: φT1 10: φT4 11: φT16	

＜TB3CLK＞:Selects the TMRB3 timer count clock.

＜TB3CLE＞:Clears and controls the TMRB3 up-counter.

　　　　　　"0": Disables clearing of the up-counter.

　　　　　　"1": Clears up-counter if there is a match with timer register 1 (TB3RG1).

(Note 1)　　　　The value read from bit 5 of TB3MOD is "1".

3.9.4.6 TMRB **flip-flop control register (channels 0 through** F)

TMRBx flip-flop control register (x=0~F)

TBxFFCR (0x400D_0xx0)		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							
		23	22	21	20	19	18	17	16
	bit Symbol								
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							
		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							
		7	6	5	4	3	2	1	0
	bit Symbol			TBC1T1	TBC0T1	TBE1T1	TBE0T1	TBFF0C	
	Read/Write	R		R/W				R/W	
	After reset	1	1	0	0	0	0	1	1
	Function	"11" is read.		TBnFF0 reverse trigger 0: Disable trigger 1: Enable trigger				TBnFF0 control 00: Invert 01: Set 10: Clear 11: Don't care * This is always read as "11."	
				When the up-counter value is taken into TBnCP1	When the up-counter value is taken into TBnCP0	When the up-counter matches TBnRG1	When the up-counter matches TBnRG0		

＜TBFF0C＞:Controls the timer flip-flop.

　　　　"00": Reverses the value of TBnFF0 (reverse by using software).

　　　　"01": Sets TBnFF0 to "1".

　　　　"10": Clears TBnFF0 to "0".

　　　　"11":Don't care

＜TBE1:0＞:Reverses the timer flip-flop when the up-counter matches the timer register 0,1 (TBnRG0,1).

＜TBC1:0＞:Reverses the timer flip-flop when the up-counter value is taken into the capture register 0,1 (TBnCP0,1).

3.9.4.7 TMRB **status register (channels 0 through F)**

TMRBx status register (x=0~F)

TBxST (0x400D_0xx4)		**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
	bit Symbol								
	Read/Write					R			
	After reset	0	0	0	0	0	0	0	0
	Function				"0" is read.				
		23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	bit Symbol								
	Read/Write					R			
	After reset	0	0	0	0	0	0	0	0
	Function				"0" is read.				
		15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	bit Symbol								
	Read/Write					R			
	After reset	0	0	0	0	0	0	0	0
	Function				"0" is read.				
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	bit Symbol						INTTBOF	INTTB1	INTTB0
	Read/Write				R			R	
	After reset				0		0	0	0
	Function			"0" is read.			0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated

＜INTTB0＞:Interrupt generated if there is a match with timer register 0 (TBnRG0)

＜INTTB1＞:Interrupt generated if there is a match with timer register 1 (TBnRG1)

＜INTTBOF＞:Interrupt generated if an up-counter overflow occurs

(Note)	**If any interrupt is generated, the flag that corresponds to the interrupt is set to TBnST and the generation of interrupt is notified to the CPU. The flag is cleared by reading the TBnST register.**

3.9.4.8 TMRB interrupt mask register (channels 0 through F)

TMRBx interrupt mask register (x=0~F)

TBxIM
(0x400D_0xx8)

		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read							
		23	22	21	20	19	18	17	16
	bit Symbol								
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read							
		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read							
		7	6	5	4	3	2	1	0
	bit Symbol						TBIMOF	TBIM1	TBIM0
	Read/Write	R					R		
	After reset	0					0	0	0
	Function	"0" is read					1: Interrupt is masked	1: Interrupt is masked	1: Interrupt is masked

<TBIM0>: Interrupt is masked if there is a match with timer register 0 (TBnRG0)

<TBIM1>:Interrupt is masked if there is a match with timer register 1 (TBnRG1).

<TBIMOF>:Interrupt is masked if an up-and-down counter overflow occurs.

3.9.4.9 TMRB **read capture register (channels 0 through F)**

TbxUC read capture register (x=0~F)

TBxUC
(0x400D_0xxC)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	TBUC							
Read/Write	R							
After reset	0							
Function	Data obtained by read capture: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	TBUC							
Read/Write	R							
After reset	0							
Function	Data obtained by read capture: 7-0 bit							

3.9.4.10 TMRB **timer register (channels 0 through F)**

TBxRG0 timer register (x=0~F)

TBxRG0
(0x400D_0xx0)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	TBRG0							
Read/Write	R/W							
After reset	0							
Function	Timer count value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	TBRG0							
Read/Write	R/W							
After reset	0							
Function	Timer count value: 7-0 bit data							

TBxRG1 timer register (x=0~F)

TBxRG1
(0x400D_0xx4)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	TBRG1							
Read/Write	R/W							
After reset	0							
Function	Timer count value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	TBRG1							
Read/Write	R/W							
After reset	0							
Function	Timer count value: 7-0 bit data							

3.9.4.11　TMRB **capture register (channels 0 through F)**

TBxCP0 capture register (x=0~F)

TBxCP0
(0x400D_0xx8)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	TBCP0							
Read/Write	R							
After reset	Undefined							
Function	Timer capture value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	TBCP0							
Read/Write	R							
After reset	Undefined							
Function	Timer capture value: 7-0 bit data							

TBxCP1 capture register (x=0~F)

TBxCP1
(0x400D_0xxC)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	TBCP1							
Read/Write	R							
After reset	Undefined							
Function	Timer capture value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	TBCP1							
Read/Write	R							
After reset	Undefined							
Function	Timer capture value: 7-0 bit data							

3.9.5 Description of Operations for Each Circuit

The channels operate in the same way, except for the differences in their specifications as shown in Table 3.9.1 and Table 3.9.2. Therefore, the operational descriptions here are only for channel 1.

3.9.5.1 Prescaler

There is a 4-bit prescaler to generate the source clock for up-counter UC1. The prescaler input clock ϕT0 is fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16 or fperiph/32 selected by CGSYSCR<PRCK> in the CG. The peripheral clock, fperiph, is either fgear, a clock selected by CGSYSCR<FPSEL> in the CG, or fc, which is a clock before it is divided by the clock gear.

The operation or the stoppage of a prescaler is set with TB1RUN<TBPRUN> where writing "1" starts counting and writing "0" clears and stops counting. Table 3.9.4 show prescaler output clock resolutions.

Table 3.9.4 Prescaler Output Clock Resolutions @fc = 48MHz

Release peripheral clock <FPSEL>	Clock gear value <GEAR2:0> (fsys)	Select prescaler clock <PRCK2:0>	Prescaler output clock resolutions φT1	φT4	φT16
0 (fgear)	000 (fc)	000(fperiph/1)	—	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)
		001(fperiph/2)	$fc/2^2$(0.0833μs)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)
		010(fperiph/4)	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)
		011(fperiph/8)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)
		100(fperiph/16)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		101(fperiph/32)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
	100(fc/2)	000(fperiph/1)	$fc/2^2$(0.0833μs)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)
		001(fperiph/2)	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)
		010(fperiph/4)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)
		011(fperiph/8)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		100(fperiph/16)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
		101(fperiph/32)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)
	101(fc/4)	000(fperiph/1)	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)
		001(fperiph/2)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)
		010(fperiph/4)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		011(fperiph/8)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
		100(fperiph/16)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)
		101(fperiph/32)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)	$fc/2^{12}$(85.33μs)
	110(fc/8)	000(fperiph/1)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)
		001(fperiph/2)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		010(fperiph/4)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
		011(fperiph/8)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)
		100(fperiph/16)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)	$fc/2^{12}$(85.33μs)
		101(fperiph/32)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)	$fc/2^{13}$(170.67μs)
1 (fc) meet below conditions fsys＞φTn	000 (fc)	000(fperiph/1)	—	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)
		001(fperiph/2)	$fc/2^2$(0.0833μs)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)
		010(fperiph/4)	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)
		011(fperiph/8)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)
		100(fperiph/16)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		101(fperiph/32)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
	100(fc/2)	000(fperiph/1)	—	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)
		001(fperiph/2)	—	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)
		010(fperiph/4)	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)
		011(fperiph/8)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)
		100(fperiph/16)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		101(fperiph/32)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
	101(fc/4)	000(fperiph/1)	—	—	$fc/2^5$(0.67μs)
		001(fperiph/2)	—	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)
		010(fperiph/4)	—	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)
		011(fperiph/8)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)
		100(fperiph/16)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		101(fperiph/32)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
	110(fc/8)	000(fperiph/1)	—	—	$fc/2^5$(0.67μs)
		001(fperiph/2)	—	—	$fc/2^6$(1.33μs)
		010(fperiph/4)	—	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)
		011(fperiph/8)	—	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)
		100(fperiph/16)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		101(fperiph/32)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)

(Note 1) The prescaler output clock φTn must be selected so that φTn<fsys/2 is satisfied (so that φTn is slower than fsys/2).

(Note 2) Do not change the clock gear while the timer is operating.

(Note 3) "—" denotes a setting prohibited.

3.9.5.2   Up-counter (UC1)

UC1 is a 16-bit binary counter.

- Source clock

    UC0 source clock, specified by TB1MOD<TB1CLK1:0>, can be selected from either three types - ϕT1, ϕT4 and ϕT16 – of prescaler output clock or the external clock of the TB1IN0 pin.

- Count start/ stop

    Counter operation is specified by TB1RUN<TBRUN>. UC1 starts counting if <TBRUN> = "1", and stops counting and clears counter value if <TBRUN> = "0".

- Timing to clear UC1

    1) When a match is detected

    By setting TB1MOD<TBCLE> = "1", UC1 is cleared if when the comparator detects a match between counter value and the value set in TB1RG1. UC1 operates as a free-running counter if TB1MOD<TBCLE> = "0".

    2) When UC1 stops

    UC1 stops counting and clears counter value if TB1RUN <TBRUN> = "0".

- UC1 overflow

    If UC1 overflow occurs, the INTTB1 overflow interrupt is generated.

3.9.5.3 Timer registers (TB1RG0, TB1RG1)

TB1RG0 and TB1RG1 are registers for setting values to compare with up-counter values and two registers are built into each channel. If the comparator detects a match between a value set in this timer register and that in a UC1 up-counter, it outputs the match detection signal.

- Configuration

TB1RG0 and TB1RG1 of this timer registers are paired with register buffer 0 – a double-buffered configuration. The two registers use TB1CR<TBWBF> to control the enabling/disabling of double buffering. If <TBWBF> = "0", double buffering is disabled and if <TBWBF> = "1", it is enabled. If double buffering is enabled, data is transferred from register buffer 0 to the TB1RG0 and TB1RG1 timer registers when there is a match between UC1 and TB1RG1.

- Default setting

The values of TB1RG0 and TB1RG1 become undefined after a reset. A reset disables the double buffer.

- Register setting

1) When not using double-buffering

To write data to the timer registers, either a 2-byte data transfer instruction or a 1-byte data transfer instruction written twice in the order of low-order 8 bits followed by high-order 8 bits can be used.

2) When using double-buffering

TB1RG0/ TB1RG1 and the register buffers are assigned to the same address. If <TBWBF> = "0," the same value is written to TB1RG0, TB1RG1 and each register buffer; if <TBWBF> = "1," the value is only written to each register buffer. To write an initial value to the timer register, therefore, the register buffers must be set to "disable". Then set <TBWBF> = "1"and write the following data to the register.

3.9.5.4 Capture

This is a circuit that controls the timing of latching values from the UC1 up-counter into the TB1CP0 and TB1CP1 capture registers. The timing with which to latch data is specified by TB1MOD <TBCPM1:0>.

Software can also be used to import values from the UC1 up-counter into the capture register; specifically, UC1 values are taken into the TB1CP0 capture register each time "0" is written to TB1MOD<TBCP0>. To use this capability, the prescaler must be running (TB1RUN<TBPRUN> = "1").

3.9.5.5 Capture Registers (TB1CP0, TB1CP1)

These are 16-bit registers for latching values from the UC1 up-counter. To read data from the capture register, use a 16-bit data transfer instruction or read in the order of low-order bits followed by high-order bits.

3.9.5.6 Up-counter capture register (TB1UC)

Other than the capturing functions shown above, the current count value of the UC1 can be captured by reading the TB1UC registers.

3.9.5.7 Comparators (CP0, CP1)

These are 16-bit comparators for detecting a match by comparing set values of the UC1 up-counter with set values of the TB1RG0 and TB1RG1 timer registers. If a match is detected, INTTB1 is generated.

3.9.5.8 Timer Flip-flop (TB1FF0)

The timer flip-flop (TB1FF0) is reversed by a match signal from the comparator and a latch signal to the capture registers. It can be enabled or disabled to reverse by setting the TB1FFCR<TB1C1T1, TB1C0T1, TB1E1T1, TB1E0T1>.

The value of TB1FF0 becomes undefined after a reset. The flip-flop can be reversed by writing "00" to TB1FFCR<TBFF0C[1:0]>. It can be set to "1" by writing "01," and can be cleared to "0" by writing "10."

The value of TB1FF0 can be output to the timer output pin, TB1OUT (shared with PL1). To enable timer output, the port L related registers PLCR and PLFR must be programmed beforehand.

3.9.5.9 Capture Interrupt (INTCAP00, INTCAP01)

Interrupts INTCAP00 and INTCAP01 can be generated at the timing of latching values from the UC1 up-counter into the TB1CP0 and TB1CP1 capture registers. The interrupt timing is specified by the CPU.

3.9.6 Description of Operations for Each Mode

3.9.6.1 16-bit interval Timer Mode

-Generating interrupts at periodic cycles

To generate the INTTB1 interrupt, specify a time interval in the TB1RG1 timer register.

3.9.6.2 16-bit Event Counter Mode

It is possible to make it the event counter by using an input clock as an external clock (TB1IN0 pin input).

The up-counter counts up on the rising edge of TB1IN0 pin input. It is possible to read the count value by capturing value using software and reading the captured value.

To use it as an event counter, put the prescaler in a "RUN" state (TB1RUN<TB1PRUN> = "1").

3.9.6.3 16-bit Programmable Square Wave Output Mode (PPG)

Square waves with any frequency and any duty (programmable square waves) can be output. The output pulse can be either low-active or high-active.

Programmable square waves can be output from the TB1OUT pin by triggering the timer flip-flop (TB1FF) to reverse when the set value of the up-counter (UC1) matches the set values of the timer registers (TB1RG0 and TB1RG1). Note that the set values of TB1RG0 and TB1RG1 must satisfy the following requirement:

(Set value of TB1RG0) < (Set value of TB1RG1)
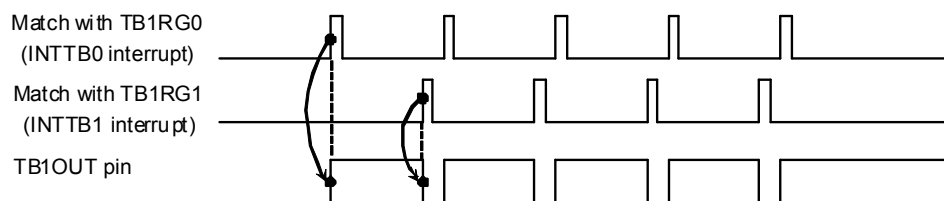
Fig. 3.9.3 Example of Output of Programmable Square Wave (PPG)

In this mode, by enabling the double buffering of TB1RG0, the value of register buffer 0 is shifted into TB1RG0 when the set value of the up-counter matches the set value of TB1RG1. This facilitates handling of small duties.
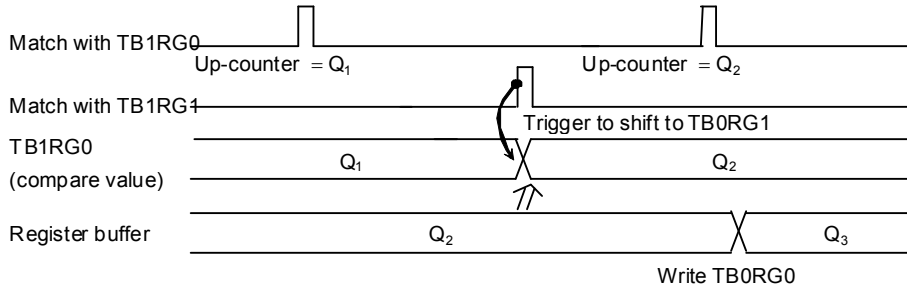
Fig. 3.9.4 Register Buffer Operation

The block diagram of this mode is shown below.
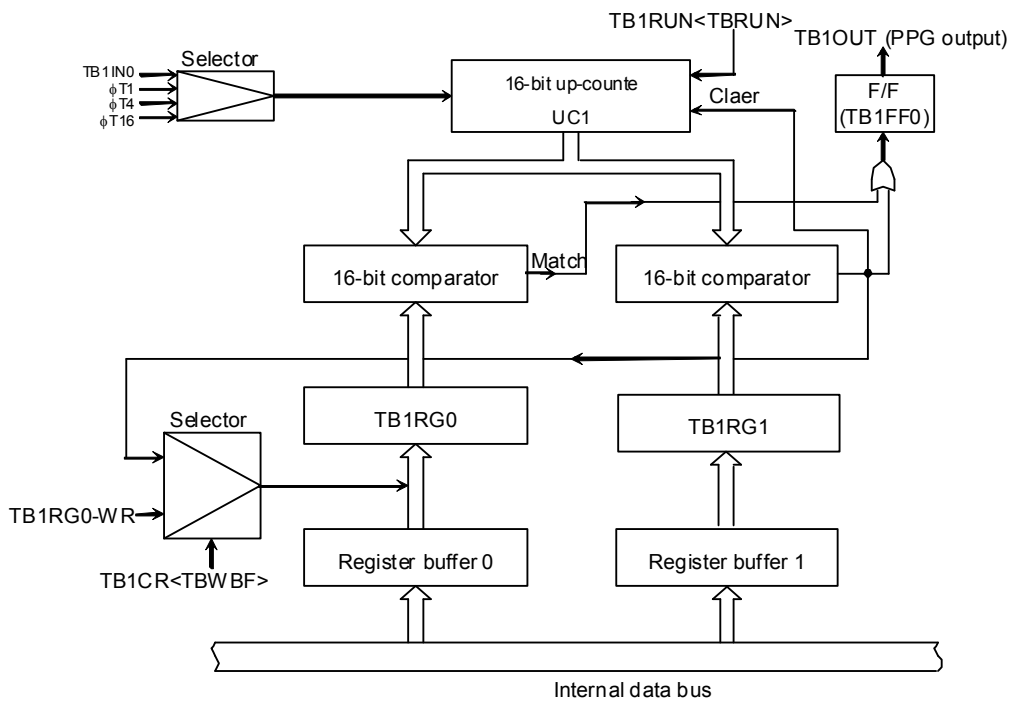
Fig. 3.9.5 Block Diagram of 16-bit PPG Mode

Each register in the 16-bit PPG output mode must be programmed as listed below.

```
                7  6  5  4  3  2  1  0
TB1EN     ←     1  X  X  X  X  X  X  X        Starts the TMRB0 module.
TB1RUN    ←     X  X  X  X  X  0  X  0        Stops the TMRB0 module.

TB1RG0    ←     *  *  *  *  *  *  *  *        Specifies a duty. (16 bits *32-bits for register)
          ←     *  *  *  *  *  *  *  *
TB1RG1    ←     *  *  *  *  *  *  *  *        Specifies a cycle. (16 bits *32-bits for register)
          ←     *  *  *  *  *  *  *  *
TB1CR     ←     1  0  X  0  0  0  0  0        Enables the TB1RG0 double buffering.
                                             (Changes the duty/cycle when the INTTB1 interrupt is
                                             generated)

TB1FFCR   ←     X  X  0  0  1  1  1  0        Specifies to trigger TB1FF0 to reverse
                                             when a match with TB1RG0 or TB1RG1 is detected,
                                             and sets the initial value of TB1FF0 to "0."

TB1MOD    ←     0  0  1  0  0  1  *  *        Designates the prescaler output clock as the input clock,
                             (** = 01, 10, 11)   and disables the capture function.
PICR      ←     −  −  −  −  −  −  −  1
PIFR1     ←     −  −  −  −  −  −  −  1        Assigns PI0 to TB1OUT
TB1RUN    ←     *  *  *  *  *  1  X  1        Starts TMRB0
```

X; Don't care   −; no change

3.9.7 Timer synchronous mode

This mode enables the timers to start synchronously.

If the mode is used with PPG output, the output can be applied to drive a motor.

Use of the timer synchronous mode is specified in TBnCR<TBSYNC>.

<TBSYNC> ="0": Timers operates individually.

<TBSYNC> ="1": Timers operate synchronously.

The channels are in four segments: channels TMRB0 through 3, channels TMRB4 through 7, channels TMRB8 through B and channels TMRBC through F.

If <TBSYNC> ="1" is set, the start timing is synchronized with TMRB0, TMRB4, TMRB8 and TMRBC. The start timing of each channel, TBnRUN <TBPRUN,TBRUN> ="1,1", is ignored.

(Note 1) The channels designated for synchronous output must be started by TBnRUN<TBPRUN,TBRUN>="1,1" before the start triggered by TMRB0, TMRB4, TMRB8 and TMRBC.

(Note 2) Set TBnCR<TBSYNC> to "0" unless the timer synchronous mode is used. The timer synchronous mode keeps the other channels operation waiting until TMRB0, TMRB4, TMRB8 and TMRBC start operation.

(Note 3) TMRB0, TMRB4, TMRB8 and TMRBC are the master clocks of the timer synchronous mode. Therefore, their TBSYNC bit must be set to "0".

	7	6	5	4	3	2	1	0
bit Symbol	TBWBF		TBSYNC		I2TB			
Read/Write	R/W	R/W	R/W	R	R/W	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Double Buffer 0:Disabled 1:Enabled	Write "0".	Timer operation **0:Individual**	"0" is read.	IDLE 0:Stop 1:Operation	"0" is read.	"0" is read.	"0" is read.

TBnCR (0x400D_0xx8)

Set the TBSYNC bit of the timers, which are used as the slave clocks, to "1".

	7	6	5	4	3	2	1	0
bit Symbol	TBWBF		TBSYNC		I2TB			
Read/Write	R/W	R/W	R/W	R	R/W	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Double Buffer 0:Disabled 1:Enabled	Write "0".	Timer operation **1:Synchronous**	"0" is read.	IDLE 0:Stop 1:Operation	"0" is read.	"0" is read.	"0" is read.

TBnCR (0x400D_0xx8)

3.9.8 Applications using the Capture Function

The capture function can be used to develop many applications, including those described below:

① One-shot pulse output triggered by an external pulse

② Frequency measurement

③ Pulse width measurement

④ Time difference measurement

① One-shot pulse output triggered by an external pulse

One-shot pulse output triggered by an external pulse is carried out as follows:

The 16-bit up-counter is made to count up by putting it in a free-running state using the prescaler output clock. An external pulse is input through the TB5IN0 pin. A trigger is generated at the rising of the external pulse by using the capture function and the value of the up-counter is taken into the capture registers (TB5CP0).

The CPU must be programmed so that an interrupt INTCAP50 is generated at the rising of an external trigger pulse. This interrupt is used to set the timer registers (TB5RG0) to the sum of the TB5CP0 value (c) and the delay time (d), (c + d), and set the timer registers (TB5RG1) to the sum of the TB5RG0 values and the pulse width (p) of one-shot pulse, (c + d + p).

TB5RG1 change must be completed before the next match.

In addition, the timer flip-flop control registers (TB5FFCR<TBE1T1, TBE0T1>) must be set to "11." This enables triggering the timer flip-flop (TB5FF0) to reverse when UC5 matches TB5RG0 and TB5RG1. This trigger is disabled by the INTTB5 interrupt after a one-shot pulse is output.

Symbols (c), (d) and (p) used in the text correspond to symbols c, d and p in "Fig3.9.6 One-shot Pulse Output (With Delay)."
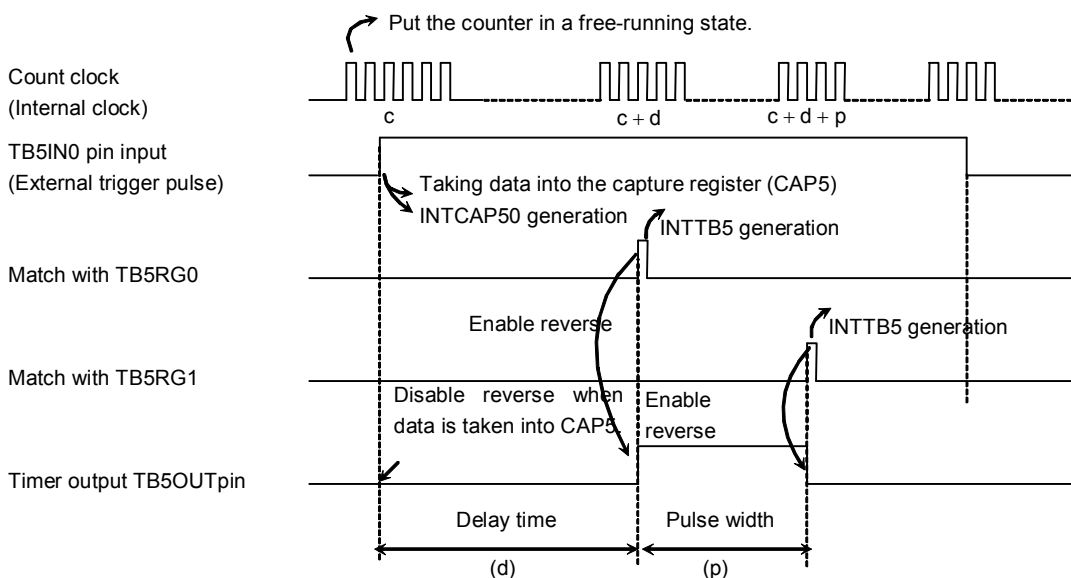
Fig. 3.9.6 One-shot Pulse Output (With Delay)

If a delay is not required, TB5FF0 is reversed when data is taken into TB5CP0, and TB5RG1 is set to the sum of the TB5CPO value (c) and the one-shot pulse width (p), (c + p), by generating the INT0 interrupt. TB5RG1 change must be completed before the next match. TB5FF0 is enabled to reverse when UC5 matches with TB5RG1, and is disabled by generating the INTTB5 interrupt.
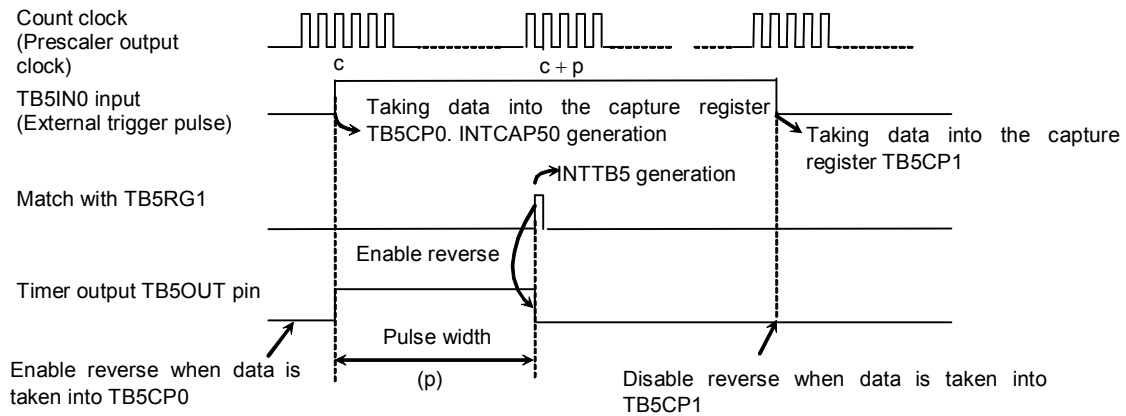
Fig. 3.9.7 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

② Frequency measurement

The frequency of an external clock can be measured by using the capture function.

To measure frequency, another 16-bit timer (TMRB0) is used in combination with the 16-bit event counter mode. As an example, we explain with TMRB2 and TMRB6. TB6OUT of the 16-bit timer TMRB6 is used to specify the measurement time.

The TB2IN0 pin input is selected as the TMRB2 count clock to perform the count operation using an external input clock. TB2MOD<TBCPM[1:0]> is set to "11." This setting allows a count value of the 16-bit up-counter UC2 to be taken into the capture register (TB2CP0) upon rising of a timer flip-flop output (TB6OUT) of the 16-bit timer (TMRB6), and an UC2 counter value to be taken into the capture register (TB2CP1) upon falling of TB6OUT of the 16-bit timer (TMRB6).

A frequency is then obtained from the difference between TB2CP0 and TB2CP1 based on the measurement, by generating the INTTB6 16-bit timer interrupt.
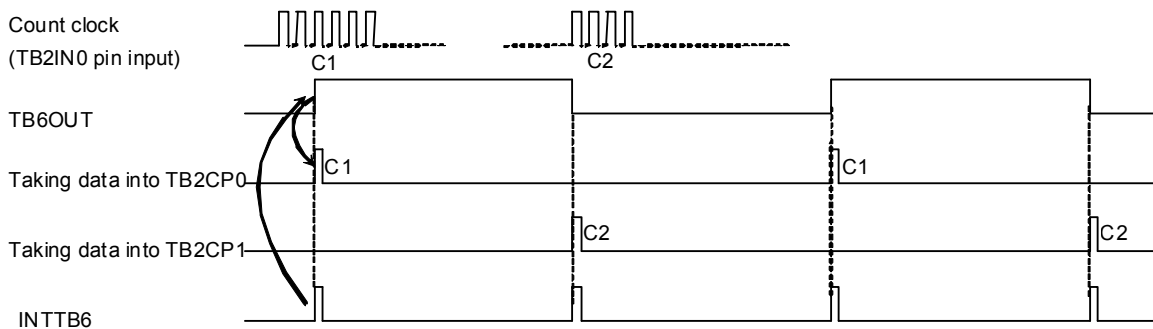
Fig. 3.9.8 Frequency Measurement

For example, if the difference between TB2CP0 and TB2CP1 is 100 and the level width setting value of TB6OUT is 0.5s, the frequency is 200Hz (100 / 0.5 s = 200 Hz).

③   Pulse width measurement

By using the capture function, the "H" level width of an external pulse can be measured. Specifically, by putting it in a free-running state using the prescaler output clock, an external pulse is input through the TB5IN0 pin and the up-counter (UC5) is made to count up. A trigger is generated at each rising and falling edge of the external pulse by using the capture function and the value of the up-counter is taken into the capture registers (TB5CP0, TB5CP1). The CPU must be programmed so that INTCAP51 is generated at the falling edge of an external pulse input through the TB5IN0 pin.

The "H" level pulse width can be calculated by multiplying the difference between TB5CP0 and TB5CP1 by the clock cycle of an internal clock.

For example, if the difference between TB5CP0 and TB5CP1 is 100 and the cycle of the prescaler output clock is 0.5 us, the pulse width is 100×0.5 us = 50 us.

Caution must be exercised when measuring pulse widths exceeding the UC5 maximum count time which is dependant upon the source clock used. The measurement of such pulse widths must be made using software.
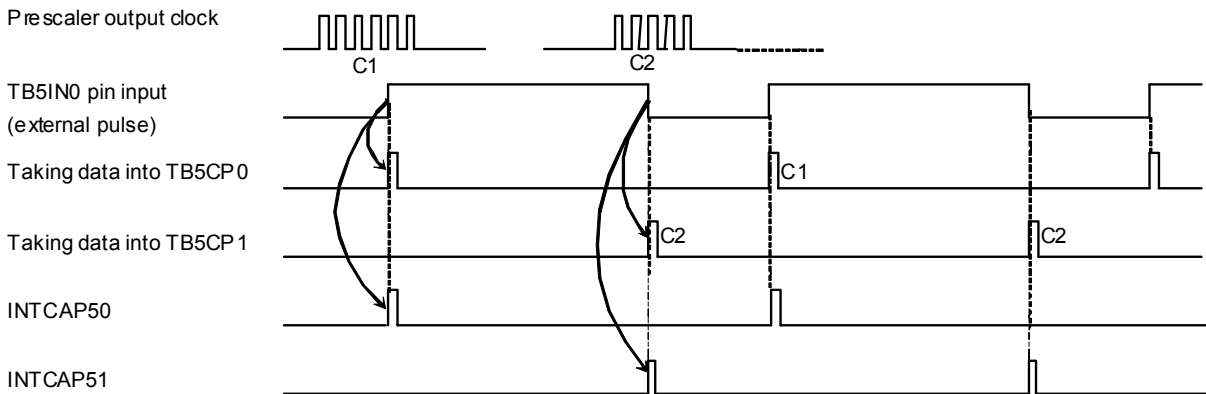
Fig. 3.9.9 Pulse Width Measurement

The "L" level width of an external pulse can also be measured. In such cases, the difference between C2 generated the first time and C1 generated the second time is initially obtained by performing the second stage of INTCAP50 interrupt processing as shown in "Fig. 3.9.9 Pulse Width Measurement" and this difference is multiplied by the cycle of the prescaler output clock to obtain the "L" level width.

④   Time Difference Measurement

The up-counter (UC5) is made to count up by putting it in a free-running state using the prescaler output clock. The value of UC5 is taken into the capture register (TB5CP0) at the rising edge of the TB5IN0 pin input pulse. The CPU must be programmed to generate INTCAP50 interrupt at this time.

The value of UC5 is taken into the capture register TB5CP1 at the rising edge of the TB5IN1 pin input pulse. The CPU must be programmed to generate INTCAP51 interrupt at this time.

The time difference can be calculated by multiplying the difference between TB5CP1 and TB5CP0 by the clock cycle of an internal clock.
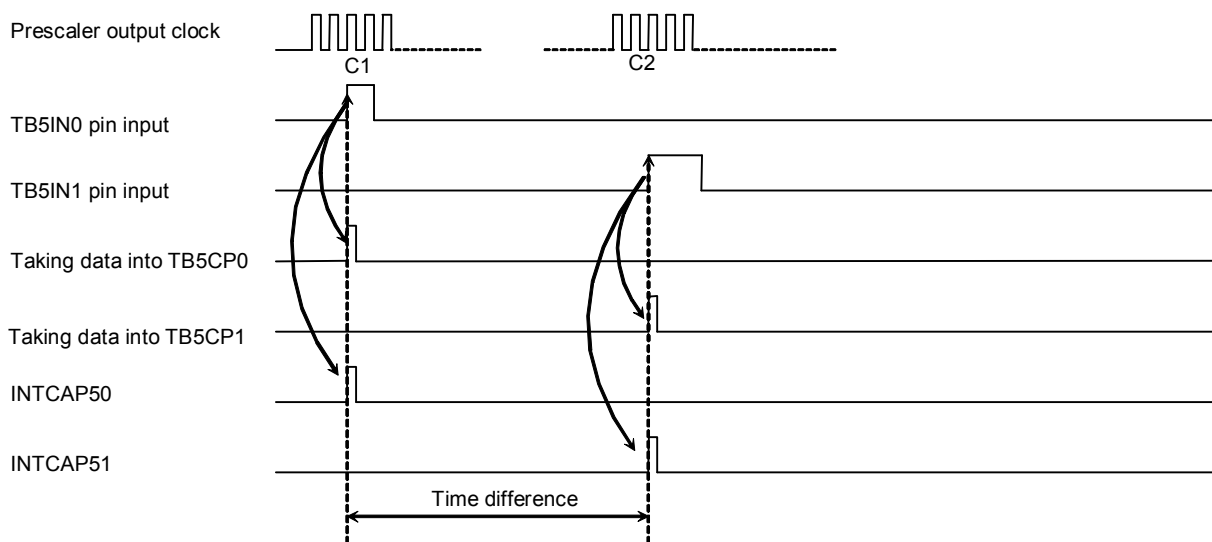
Fig. 3.9.10 Time Difference Measurement

3.10 Serial Channel (SIO/UART)

3.10.1 Features

This device has twelve serial I/O channels: SIO0 to SIO11. Each channel operates in either the UART mode (asynchronous communication) or the I/O interface mode (synchronous communication) which is selected by the user.

I/O interface mode data	—— Mode 0:	This is the mode to transmit and receive I/O and associated synchronization signals (SCLK) to extend I/O.
Asynchronous (UART) mode:	⌐ Mode 1:	TX/RX Data Length: 7 bits
	├ Mode 2:	TX/RX Data Length: 8 bits
	⌐ Mode 3:	TX/RX Data Length: 9 bits

In the above modes 1 and 2, parity bits can be added. The mode 3 has a wakeup function in which the master controller can start up slave controllers via the serial link (multi-controller system). Fig.3.10.2 shows the block diagram of SIO0.

Each channel consists of a prescaler, a serial clock generation circuit, a receive buffer, its control circuit, a transmit buffer and its control circuit. Each channel functions independently.

As the SIOs 0 to 11 operate in the same way, only SIO0 is described here.

Table 3.10.1 Difference in the Specifications of SIO Module (1/4)

		Channel 0		Channel 1		Channel 2	
Pin name		TXD0 (PE4) RXD0 (PE5) CTS0n/SCLK0 (PE5)		TXD1 (PL4) RXD1 (PL5) CTS1n/SCLK1 (PL5)		TXD2 (PM1) RXD2 (PM2) CTS2n/SCLK2 (PM0)	
Interrupt		INTRX0 INTTX0		INTRX1 INTTX1		INTRX2 INTTX2	
Register name (address)	Enable register	SC0EN	0x400E_1000	SC1EN	0x400E_1100	SC2EN	0x400E_1200
	Transmit/ receive buffer register	SC0BUF	0x400E_1004	SC1BUF	0x400E_1104	SC2BUF	0x400E_1204
	Control register	SC0CR	0x400E_1008	SC1CR	0x400E_1108	SC2CR	0x400E_1208
	Mode control register 0	SC0MOD0	0x400E_100C	SC1MOD0	0x400E_110C	SC2MOD0	0x400E_120C
	Baud rate generator control	SC0BRCR	0x400E_1010	SC1BRCR	0x400E_1110	SC2BRCR	0x400E_1210
	Baud rate generator control 2	SC0BRADD	0x400E_1014	SC1BRADD	0x400E_1114	SC2BRADD	0x400E_1214
	Mode control register 1	SC0MOD1	0x400E_1018	SC1MOD1	0x400E_1118	SC2MOD1	0x400E_1218
	Mode control register 2	SC0MOD2	0x400E_101C	SC1MOD2	0x400E_111C	SC2MOD2	0x400E_121C
	Receive FIFO configuration register	SC0RFC	0x400E_1020	SC1RFC	0x400E_1120	SC2RFC	0x400E_1220
	Transmit FIFO configuration register	SC0TFC	0x400E_1024	SC1TFC	0x400E_1124	SC2TFC	0x400E_1224
	Receive FIFO status register	SC0RST	0x400E_1028	SC1RST	0x400E_1128	SC2RST	0x400E_1228
	Transmit FIFO status register	SC0TST	0x400E_102C	SC1TST	0x400E_112C	SC2TST	0x400E_122C
	FIFO configuration register	SC0FCNF	0x400E_1030	SC1FCNF	0x400E_1130	SC2FCNF	0x400E_1230

Table 3.10.1 Difference in the Specifications of SIO Module (2/4)

		Channel 3		Channel 4		Channel 5	
Pin name		TXD3 (PM5) RXD3 (PM6) CTS3n/SCLK3 (PM4)		TXD4 (PN0) RXD4 (PN1) CTS4n/SCLK4 (PN2)		TXD5 (PN4) RXD5 (PN5) CTS5n/SCLK5 (PN6)	
Interrupt		INTRX3 INTTX3		INTRX4 INTTX4		INTRX5 INTTX5	
Register name (address)	Enable register	SC3EN	0x400E_1300	SC4EN	0x400E_1400	SC5EN	0x400E_1500
	Transmit/ receive buffer register	SC3BUF	0x400E_1304	SC4BUF	0x400E_1404	SC5BUF	0x400E_1504
	Control register	SC3CR	0x400E_1308	SC4CR	0x400E_1408	SC5CR	0x400E_1508
	Mode control register 0	SC3MOD0	0x400E_130C	SC4MOD0	0x400E_140C	SC5MOD0	0x400E_150C
	Baud rate generator control	SC3BRCR	0x400E_1310	SC4BRCR	0x400E_1410	SC5BRCR	0x400E_1510
	Baud rate generator control 2	SC3BRADD	0x400E_1314	SC4BRADD	0x400E_1414	SC5BRADD	0x400E_1514
	Mode control register 1	SC3MOD1	0x400E_1318	SC4MOD1	0x400E_1418	SC5MOD1	0x400E_1518
	Mode control register 2	SC3MOD2	0x400E_131C	SC4MOD2	0x400E_141C	SC5MOD2	0x400E_151C
	Receive FIFO configuration register	SC3RFC	0x400E_1320	SC4RFC	0x400E_1420	SC5RFC	0x400E_1520
	Transmit FIFO configuration register	SC3TFC	0x400E_1324	SC4TFC	0x400E_1424	SC5TFC	0x400E_1524
	Receive FIFO status register	SC3RST	0x400E_1328	SC4RST	0x400E_1428	SC5RST	0x400E_1528
	Transmit FIFO status register	SC3TST	0x400E_132C	SC4TST	0x400E_142C	SC5TST	0x400E_152C
	FIFO configuration register	SC3FCNF	0x400E_1330	SC4FCNF	0x400E_1430	SC5FCNF	0x400E_1530

Table 3.10.1 Difference in the Specifications of SIO Module (3/4)

		Channel 6		Channel 7		Channel 8	
Pin name		TXD6 (PO0) RXD6 (PO1) CTS6n/SCLK6 (PO2)		TXD7 (PO4) RXD7 (PO5) CTS7n/SCLK7 (PO6)		TXD8 (PC0) RXD8 (PC1) CTS8n/SCLK8 (PC2)	
Interrupt		INTRX6 INTTX6		INTRX7 INTTX7		INTRX8 INTTX8	
Register name (address)	Enable register	SC6EN	0x400E_1600	SC7EN	0x400E_1700	SC8EN	0x400E_1800
	Transmit/ receive buffer register	SC6BUF	0x400E_1604	SC7BUF	0x400E_1704	SC8BUF	0x400E_1804
	Control register	SC6CR	0x400E_1608	SC7CR	0x400E_1708	SC8CR	0x400E_1808
	Mode control register 0	SC6MOD0	0x400E_160C	SC7MOD0	0x400E_170C	SC8MOD0	0x400E_180C
	Baud rate generator control	SC6BRCR	0x400E_1610	SC7BRCR	0x400E_1710	SC8BRCR	0x400E_1810
	Baud rate generator control 2	SC6BRADD	0x400E_1614	SC7BRADD	0x400E_1714	SC8BRADD	0x400E_1814
	Mode control register 1	SC6MOD1	0x400E_1618	SC7MOD1	0x400E_1718	SC8MOD1	0x400E_1818
	Mode control register 2	SC6MOD2	0x400E_161C	SC7MOD2	0x400E_171C	SC8MOD2	0x400E_181C
	Receive FIFO configuration register	SC6RFC	0x400E_1620	SC7RFC	0x400E_1720	SC8RFC	0x400E_1820
	Transmit FIFO configuration register	SC6TFC	0x400E_1624	SC7TFC	0x400E_1724	SC8TFC	0x400E_1824
	Receive FIFO status register	SC6RST	0x400E_1628	SC7RST	0x400E_1728	SC8RST	0x400E_1828
	Transmit FIFO status register	SC6TST	0x400E_162C	SC7TST	0x400E_172C	SC8TST	0x400E_182C
	FIFO configuration register	SC6FCNF	0x400E_1630	SC7FCNF	0x400E_1730	SC8FCNF	0x400E_1830

Table 3.10.1 Difference in the Specifications of SIO Module (4/4)

		Channel 9		Channel 10		Channel11	
Pin name		TXD9 (PC4) RXD9 (PC5) CTS9n/SCLK9 (PC6)		TXD10 (PD0) RXD10 (PD1) CTS10n/SCLK10 (PD2)		TXD11 (PD4) RXD11 (PD5) CTS11n/SCLK11 (PD6)	
Interrupt		INTRX9 INTTX9		INTRX10 INTTX10		INTRX11 INTTX11	
Register name (address)	Enable register	SC9EN	0x400E_1900	SC10EN	0x400E_1A00	SC11EN	0x400E_1B00
	Transmit/ receive buffer register	SC9BUF	0x400E_1904	SC10BUF	0x400E_1A04	SC11BUF	0x400E_1B04
	Control register	SC9CR	0x400E_1908	SC10CR	0x400E_1A08	SC11CR	0x400E_1B08
	Mode control register 0	SC9MOD0	0x400E_190C	SC10MOD0	0x400E_1A0C	SC11MOD0	0x400E_1B0C
	Baud rate generator control	SC9BRCR	0x400E_1910	SC10BRCR	0x400E_1A10	SC11BRCR	0x400E_1B10
	Baud rate generator control 2	SC9BRADD	0x400E_1914	SC10BRADD	0x400E_1A14	SC11BRADD	0x400E_1B14
	Mode control register 1	SC9MOD1	0x400E_1918	SC10MOD1	0x400E_1A18	SC11MOD1	0x400E_1B18
	Mode control register 2	SC9MOD2	0x400E_191C	SC10MOD2	0x400E_1A1C	SC11MOD2	0x400E_1B1C
	Receive FIFO configuration register	SC9RFC	0x400E_1920	SC10RFC	0x400E_1A20	SC11RFC	0x400E_1B20
	Transmit FIFO configuration register	SC9TFC	0x400E_1924	SC10TFC	0x400E_1A24	SC11TFC	0x400E_1B24
	Receive FIFO status register	SC9RST	0x400E_1928	SC10RST	0x400E_1A28	SC11RST	0x400E_1B28
	Transmit FIFO status register	SC9TST	0x400E_192C	SC10TST	0x400E_1A2C	SC11TST	0x400E_1B2C
	FIFO configuration register	SC9FCNF	0x400E_1930	SC10FCNF	0x400E_1A30	SC11FCNF	0x400E_1B30

- Mode 0 (I/O Interface mode) /LSB first

← Transmission direction

- Mode 0 (I/O Interface mode) /MSB first

← Transmission direction

- Mode 1 (7 bits UART mode)

Without parity

With parity

- Mode 2 (8 bits UART mode)

Without parity

With parity

- Mode 3 (9 bits UART mode)

Ilf bit 8=1, represents address (select code)
Ilf bit 8=0, represents data.
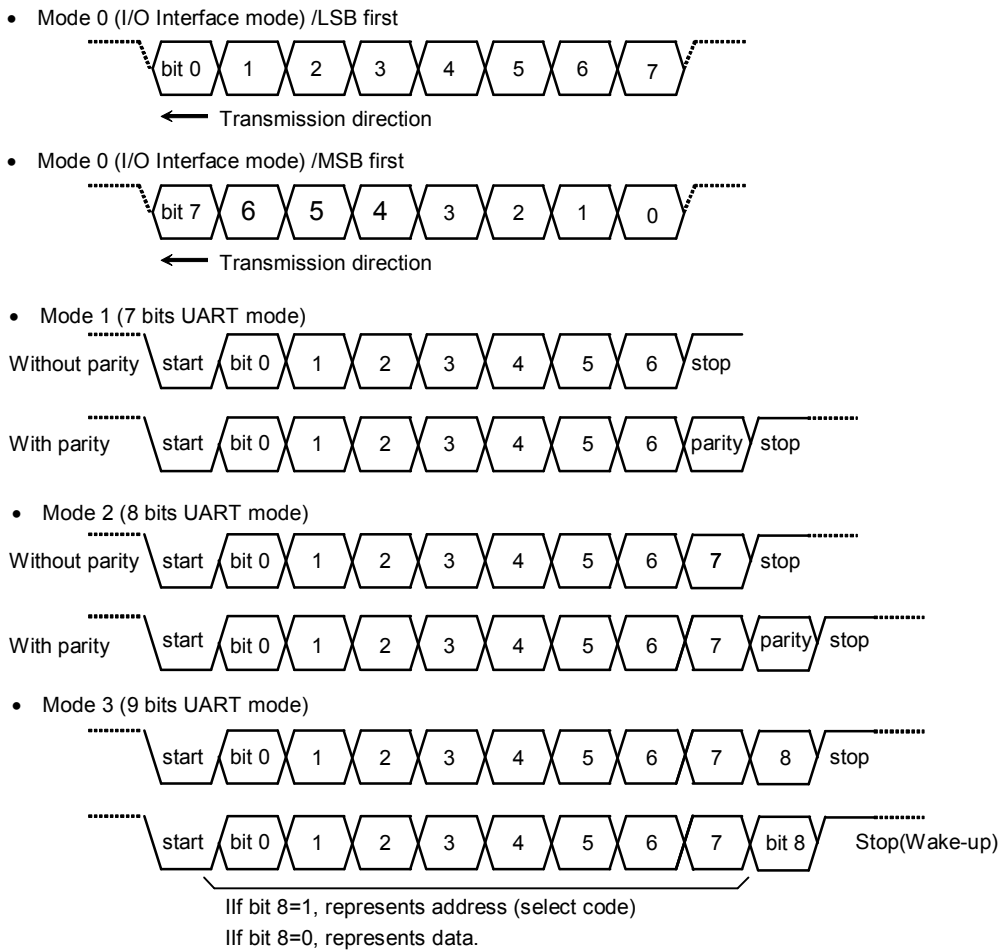
Fig. 3.10.1 Data Format

3.10.2 Block Diagram (Channel 0)
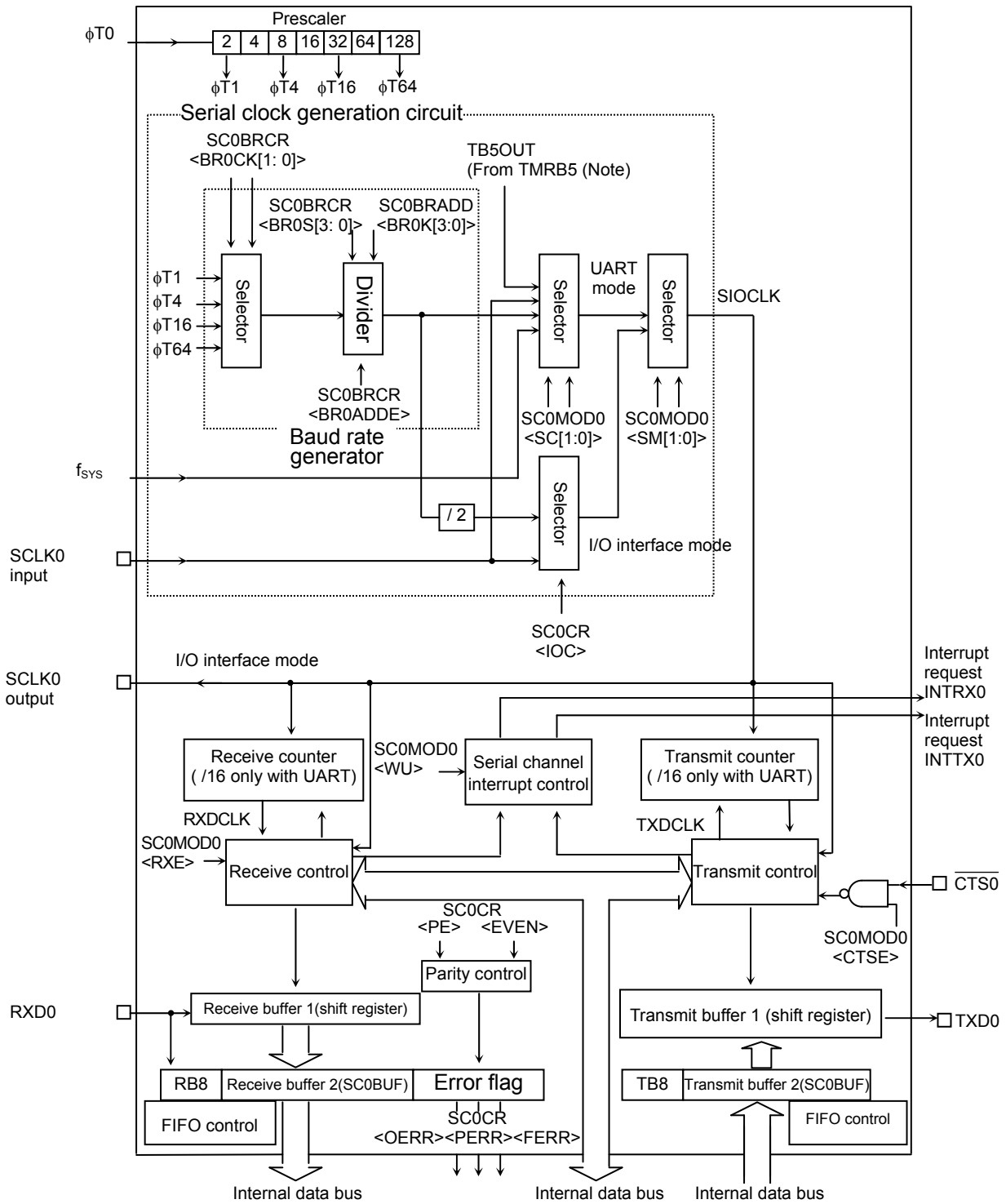
Fig 3.10.2 SIO0Block Diagram

(Note) When timer output is used as source clock, SIO0 to 3 use TMR5(TBOUT5),
SIO4 to 7 uses TMR6(TB6OUT), SIO8 to 11 uses TMR9(TB9OUT),.

3.10.3 Operation of Each Circuit (Channel 0)

The operation of SIO0 is described here. The difference between SIO0 and the other channel is listed in Table 3.10.1

3.10.3.1 Prescaler

The device includes a 7-bit prescaler to generate necessary clocks to drive SIO0. The input clock $\phi T0$ to the prescaler is selected by CGSYSCR <PRCK> to provide the frequency of either fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16 or fperiph/32.

The clock frequency fperiph is either the clock "fgear," to be selected by CGSYSCR<FPSEL> of CG, or the clock "fc" before it is divided by the clock gear.

The prescaler becomes active only when the baud rate generator is selected for generating the serial transfer clock. Table 3.10.2 list the prescaler output clock resolution.

Table3.10.2Clock Resolution to the Baud Rate Generator @ = 48MHz

Clear peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Prescaler clock selection <PRCK2:0>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
0 (fgear)	000 (fc)	000(fperiph/1)	$fc/2^1$(0.0417μs)	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.66μs)
		001(fperiph/2)	$fc/2^2$(0.0833μs)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)
		010(fperiph/4)	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		011(fperiph/8)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
		100(fperiph/16)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)
		101(fperiph/32)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)	$fc/2^{12}$(85.33μs)
	100(fc/2)	000(fperiph/1)	$fc/2^2$(0.0833μs)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)
		001(fperiph/2)	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		010(fperiph/4)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
		011(fperiph/8)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)
		100(fperiph/16)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)	$fc/2^{12}$(85.33μs)
		101(fperiph/32)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)	$fc/2^{13}$(170.67μs)
	101(fc/4)	000(fperiph/1)	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		001(fperiph/2)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
		010(fperiph/4)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)
		011(fperiph/8)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)	$fc/2^{12}$(85.33μs)
		100(fperiph/16)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)	$fc/2^{13}$(170.67μs)
		101(fperiph/32)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)	$fc/2^{12}$(85.33μs)	$fc/2^{14}$(341.33μs)
	110(fc/8)	000(fperiph/1)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
		001(fperiph/2)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)
		010(fperiph/4)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)	$fc/2^{12}$(85.33μs)
		011(fperiph/8)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)	$fc/2^{13}$(170.67μs)
		100(fperiph/16)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)	$fc/2^{12}$(85.33μs)	$fc/2^{14}$(341.33μs)
		101(fperiph/32)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)	$fc/2^{13}$(170.67μs)	$fc/2^{15}$(682.67μs)
1 (fc) meet below conditions fsys>φTn	000 (fc)	000(fperiph/1)	$fc/2^1$(0.0417μs)	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.66μs)
		001(fperiph/2)	$fc/2^2$(0.0833μs)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)
		010(fperiph/4)	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		011(fperiph/8)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
		100(fperiph/16)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)
		101(fperiph/32)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)	$fc/2^{12}$(85.33μs)
	100(fc/2)	000(fperiph/1)	—	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.66μs)
		001(fperiph/2)	$fc/2^2$(0.0833μs)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)
		010(fperiph/4)	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		011(fperiph/8)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
		100(fperiph/16)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)
		101(fperiph/32)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)	$fc/2^{12}$(85.33μs)
	101(fc/4)	000(fperiph/1)	—	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.66μs)
		001(fperiph/2)	—	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)
		010(fperiph/4)	$fc/2^3$(0.167μs)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		011(fperiph/8)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
		100(fperiph/16)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)
		101(fperiph/32)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)	$fc/2^{12}$(85.33μs)
	110(fc/8)	000(fperiph/1)	—	—	$fc/2^5$(0.67μs)	$fc/2^7$(2.66μs)
		001(fperiph/2)	—	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)
		010(fperiph/4)	—	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)
		011(fperiph/8)	$fc/2^4$(0.33μs)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)
		100(fperiph/16)	$fc/2^5$(0.67μs)	$fc/2^7$(2.67μs)	$fc/2^9$(10.67μs)	$fc/2^{11}$(42.67μs)
		101(fperiph/32)	$fc/2^6$(1.33μs)	$fc/2^8$(5.33μs)	$fc/2^{10}$(21.33μs)	$fc/2^{12}$(85.33μs)

(Note 1) The prescaler output clock φTn must be selected so that φTn<fsys/2 is satisfied (so that φTn is slower than fsys/2).

(Note 2) Do not change the clock gear while the SIO is operating.

(Note 3) "—" denotes a setting prohibited.

The serial interface baud rate generator uses four different clocks, i.e., φT1, φT4, φT16 and φT64, supplied from the prescaler output clock.

3.10.3.2 Baud Rate Generator

The baud rate generator generates transmit and receive clocks to determine the serial channel transfer rate.

The baud rate generator uses either the ϕT1, ϕT4, ϕT16 or ϕT64 clock supplied from the 7-bit prescaler. This input clock selection is made by setting the baud rate generator control register, SC0BRCR <BR0CK[1:0]>.

The baud rate generator contains built-in dividers for divide by 1, N + m/16 (N=2~15, m=0~15), and 16. The division is performed according to the settings of the baud rate generator control registers SC0BRCR <BR0ADDE> <BR0S[3:0]> and SC0BRADD <BR0K[3:0]> to determine the resulting transfer rate.

- UART Mode

  1) If SC0BRCR <BRADDE> = 0,

     The setting of SC0BRADD <BRK[3:0]> is ignored and the counter is divided by N where N is the value set to SC0BRCR <BRS[3:0]>. (N = 1 to 16).

  2) If SC0BRCR <BRADDE> = 1,

     The N + (16 - K)/16 division function is enabled and the division is made by using the values N (set in SC0BRCR <BRS[3:0]>) and K (set in SC0BRADD<BRK[3:0]>). (N = 2 to 15, K = 1 to 15)

     > **(Note) For the N values of 1 and 16, the above N+(16-K)/16 division function is inhibited. So, be sure to set SC0BRCR<BRADDE> to "0."**

- I/O interface Mode

  The N + (16 - K)/16 division function cannot be used in the I/O interface mode. Be sure to divide by N, by setting SC0BRCR <BRADDE> to "0".

- Baud rate calculation to use the baud rate generator

  1) UART mode

     $$\text{Baud rate} = \frac{\text{Baud rated generator input clock}}{\text{Frequency divided by the divide ratio}} /16$$

     The highest baud rate out of the baud rate generator is 1.5 Mbps when ϕT1 is 24 MHz.

     The fsys frequency, which is independent of the baud rate generator, can be used as the serial clock. In this case, the highest baud rate will be 3.0 Mbps when fsys is 48 MHz.

  2) I/O interface mode

     $$\text{Baud rate} = \frac{\text{Baud rated generator input clock}}{\text{Frequency divided by the divide ratio}} /2$$

     The highest baud rate will be generated when ϕT1 is 24 MHz. The divide ratio can be set to 1 if double buffer is used and the resulting output baud rate will be 12 Mbps. (If double buffering is not used, the highest baud rate will be 6.0 Mbps applying the divide ratio of "2").

- Example baud rate setting

1) Division by an integer (divide by N):

Selecting fc = 39.321 MHz for fperiph, setting ϕT0 to fperiph/16, using the baud rate generator input clock ϕT1, setting the divide ratio N (SC0BRCR<BRS[3:0]>) = 4, and setting SC0BRCR<BRADDE> = "0," the resulting baud rate in the UART mode is calculated as follows:

* Clocking conditions   System clock   :   High-speed (fc)

High speed clock gear :   x 1 (fc)

Prescaler clock   : f_{periph}/16 (f_{periph} = f_{sys})

Baud rate = $\dfrac{fc/32}{4}$ /16

$= 39.321 \times 10^6 / 32 / 4 / 16 \fallingdotseq 19200$ (bps)

> **(Note) The divide by (N + (16-K)/16) function is inhibited and thus SC0BRADD <BR0K3:0> is ignored.**

2) For divide by N + (16-K)/16 (only for UART mode):

Selecting fc = 9.6 MHz for fperiph, setting ϕT0 to fperiph/8, using the baud rate generator input clock ϕT1, setting the divide ratio N (SC0BRCR<BRS[3:0]>) = 7, setting K (SC0BRADD<BRK[3:0]>) = 3, and selecting SC0BRCR<BRADDE> = 1, the resulting baud rate is calculated as follows:

* Clocking conditions   System clock   :   High-speed (fc)

High-speed clock gear :   x 1 (fc)

Prescaler clock   : f_{periph}/4 (f_{periph} = f_{sys})

Baud rate = $\dfrac{fc/16}{7 + \dfrac{(16 - 3)}{16}}$ /16

$= 9.6 \times 10^6 / 16 / (7 + \dfrac{13}{16}) / 16 \fallingdotseq 4800$ (bps)

Also, an external clock input may be used as the serial clock. The resulting baud rate calculation is shown below:

- Baud rate calculation for an external clock input

    1) UART mode

    Baud Rate = external clock input / 16

    In this, the period of the external clock input must be equal to or greater than 2/fsys.

    If fsys = 48 MHz, the highest baud rate will be 48 / 2 / 16=1.5 (Mbps).

    2) I/O interface mode

    Baud Rate = external clock input

    When double buffering is used, it is necessary to satisfy the following relationship:

    External clock input period > 6/fsys

    Therefore, when fsys = 48 MHz, the baud rate must be set to a rate lower than

    48 / 6=8.0 (Mbps).

    When double buffering is not used, it is necessary to satisfy the following relationship:

    External clock input period > 8/fsys

    Therefore, when fsys = 48 MHz, the baud rate must be set to a rate lower than

    48 / 8=6.0 (Mbps).

    The baud rate examples for the UART mode are shown in Table 3.10.3 and Table 3.10.4.

Table 3.10.3 Selection of UART Baud Rate

(Using the baud rate generator with SC0BRCR <BRADDE> = 0)   Unit (kbps)

fc [MHz]	Divide ratio N (Set to SC0BRCR <BR0S[3:0]>) Input clock	ϕT1 (fc/4)	ϕT4 (fc/16)	ϕT16 (fc/64)	ϕT64 (fc/256)
9.830400	2	76.800	19.200	4.800	1.200
↑	4	38.400	9.600	2.400	0.600
↑	8	19.200	4.800	1.200	0.300
↑	0	9.600	2.400	0.600	0.150

(Note) This table shows the case where the system clock is set to fc, the clock gear is set to fc/1, and the prescaler clock is set to f_{periph}/2.

Table 3.10.4 Selection of UART Baud Rate

(The TMRBx timer output (internal TBxOUT) is used with the timer input clock set to ϕT1.)

Unit (kbps)

TB0RG / fc	48 MHz	32 MHz	9.8304 MHz	8 MHz
1H	375	250	76.8	62.5
2H	187.5	125	38.4	31.25
3H			25.6	
4H	93.75	62.5	19.2	15.625
5H	75	50	15.36	12.5
6H			12.8	
8H		31.25	9.6	
AH	37.5	25	7.68	6.25
10H		15.625	4.8	
14H	18.75	12.5	3.84	3.125

Baud rate calculation to use the TMRBx timer:

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by CGSYSCR<PRCK>}}{(\text{TBxRG} \times \underline{2}) \times \underline{2} \times 16}$$

└ In the case the timer prescaler clock ϕT1 (2division ratio)is selected.

└ One clock cycle is a period that the timer frip-flop is inverted twice.

(Note 1) In the I/O interface mode, the TMRB5 timer output signal cannot be used internally as the transfer clock.

(Note 2) This table shows the case where the system clock is set to fc, the clock gear is set to fc, and the prescaler clock is set to f_{periph}/4.

3.10.3.3 Serial Clock Generation Circuit

This circuit generates basic transmit and receive clocks.

- I/O interface mode

In the SCLK output mode with the SC0CR <IOC> serial control register set to "0," the output of the previously mentioned baud rate generator is divided by 2 to generate the basic clock.

In the SCLK input mode with SC0CR <IOC> set to "1," rising and falling edges are detected according to the SC0CR <SCLKS> setting to generate the basic clock.

- Asynchronous (UART) mode :

According to the settings of the serial control mode register SC0MOD0 <SC[1:0]>, either the clock from the baud rate register, the system clock (f_{SYS}), the internal output signal of the TMRB5 timer, or the external clock (SCLKO pin) is selected to generate the basic clock, SIOCLK.

3.10.3.4 Receive Counter

The receive counter is a 4-bit binary counter used in the asynchronous (UART) mode and is up-counted by SIOCLK. Sixteen SIOCLK clock pulses are used in receiving a single data bit while the data symbol is sampled at the seventh, eighth, and ninth pulses. From these three samples, majority logic is applied to decide the received data.

3.10.3.5 Receive Control Unit

- I/O interface mode:

In the SCLK output mode with SC0CR <IOC> set to "0," the RXD0 pin is sampled on the rising edge of the shift clock output to the SCLK0 pin.

In the SCLK input mode with SC0CR <IOC> set to "1," the serial receive data RXD0 pin is sampled on the rising or falling edge of SCLK input depending on the SC0CR <SCLKS> setting.

- Asynchronous (UART) mode:

The receive control unit has a start bit detection circuit, which is used to initiate receive operation when a normal start bit is detected.

3.10.3.6 Receive Buffer

The receive buffer is of a dual structure to prevent overrun errors. The first receive buffer (a shift register) stores the received data bit-by-bit. When a complete set of bits have been stored, they are moved to the second receive buffer (SC0BUF). At the same time, the receive buffer full flag (SC0MOD2 "RBFLL") is set to "1" to indicate that valid data is stored in the second receive buffer. However, if the receive FIFO is set enabled, the receive data is moved to the receive FIFO and this flag is immediately cleared.

If the receive FIFO has been disabled (SCOFCNF <CNFG> = 0 and SC0MOD1<FDPX[1:0]> =01), the INTRX0 interrupt is generated at the same time. If the receive FIFO has been enabled (SCNFCNF <CNFG> = 1 and SC0MOD1<FDPX[1:0]> = 01/11), an interrupt will be generated according to the SC0RFC <RIL1:0> setting.

The CPU will read the data from either the second receive buffer (SC0BUF) or from the receive FIFO (the address is the same as that of the receive buffer). If the receive FIFO has not been enabled, the receive buffer full flag <RBFLL> is cleared to "0" by the read operation. The next data received can be stored in the first receive buffer even if the CPU has not read the previous data from the second receive buffer (SC0BUF) or the receive FIFO.

If SCLK is set to generate clock output in the I/O interface mode, the double buffer control bit SC0MOD2 <WBUF> can be programmed to enable or disable the operation of the second receive buffer (SCOBUF).

By disabling the second receive buffer (i.e., the double buffer function) and also disabling the receive FIFO (SCOFCNF <CNFG> = 0 and <FDPX[1:0]> = 01), handshaking with the other side of communication can be enabled and the SCLK output stops each time one frame of data is transferred. In this setting, the CPU reads data from the first receive buffer. By the read operation of CPU, the SCLK output resumes.

If the second receive buffer (i.e., double buffering) is enabled but the receive FIFO is not enabled, the SCLK output is stopped when the first receive data is moved from the first receive buffer to the second receive buffer and the next data is stored in the first buffer filling both buffers with valid data. When the second receive buffer is read, the data of the first receive buffer is moved to the second receive buffer and the SCLK output is resumed upon generation of the receive interrupt INTRX. Therefore, no buffer overrun error will be caused in the I/O interface SCLK output mode regardless of the setting of the double buffer control bit SC0MOD2 <WBUF>.

If the second receive buffer (double buffering) is enabled and the receive FIFO is also enabled (SCNFCNF<CNFG> = 1 and <FDPX[1:0]> = 01/11), the SCLK output will be stopped when the receive FIFO is full (according to the setting of SCOFNCF <RFST>) and both the first and second receive buffers contain valid data. Also in this case, if SCOFCNF <RXTXCNT> has been set to "1," the receive control bit RXE will be automatically cleared upon suspension of the SCLK output. If it is set to "0," automatic clearing will not be performed.

> **(Note) In this mode, the SC0CR<OEER> flag is insignificant and the operation is undefined. Therefore, before switching from the SCLK output mode to another mode, the SC0CR register must be read to initialize this flag.**

In other operating modes, the operation of the second receive buffer is always valid, thus improving the performance of continuous data transfer. If the receive FIFO is not enabled, an overrun error occurs when the data in the second receive buffer (SC0BUF) has not been read before the first receive buffer is full with the next receive data. If an overrun error occurs, data in the first receive buffer will be lost while data in the second receive buffer and the contents of SC0CR <RB8> remain intact. If the receive FIFO is enabled, the FIFO must be read before the FIFO is full and the second receive buffer is written by the next data through the first buffer. Otherwise, an overrun error will be generated and the receive FIFO overrun error flag will be set. Even in this case, the data already in the receive FIFO remains intact.

The parity bit to be added in the 8-bit UART mode as well as the most significant bit in the 9-bit UART mode will be stored in SC0CR <RB8>.

In the 9-bit UART mode, the slave controller can be operated in the wake-up mode by setting the wake-up function SC0MOD0 <WU> to "1." In this case, the interrupt INTRX0 will be generated only when SC0CR <RB8> is set to "1."

3.10.3.7   Receive FIFO Buffer

In addition to the double buffer function already described, data may be stored using the receive FIFO buffer. By setting <CNFG> of the SC0FCNF register and <FDPX[1:0]> of the SC0MOD1 register, the 4-byte receive buffer can be enabled. Also, in the UART mode or I/O interface mode, data may be stored up to a predefined fill level. When the receive FIFO buffer is to be used, be sure to enable the double buffer function.

If data with parity bit is to be received in the UART mode, parity check must be performed each time a data frame is received.

3.10.3.8   Receive FIFO Operation

① I/O interface mode with SCLK output:

The following example describes the case a 4-byte data stream is received in the half duplex mode:

Data transmission can be initiated by setting the transfer mode to half duplex.

SC0FCNF <4:0>=10111:   Automatically inhibits continued reception after reaching the fill level. The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.

SC0RFC<RIL[1:0]>=00:   Sets the interrupt to be generated at fill level 4.

SC0RFC<RFCS,RFIS>=01:   Clears receive FIFO and sets the condition of interrupt generation.

In this condition, 4-byte data reception may be initiated by writing "1" to the RXE bit. After receiving 4 bytes, the RXE bit is automatically cleared and the receive operation is stopped (SCLK is stopped).
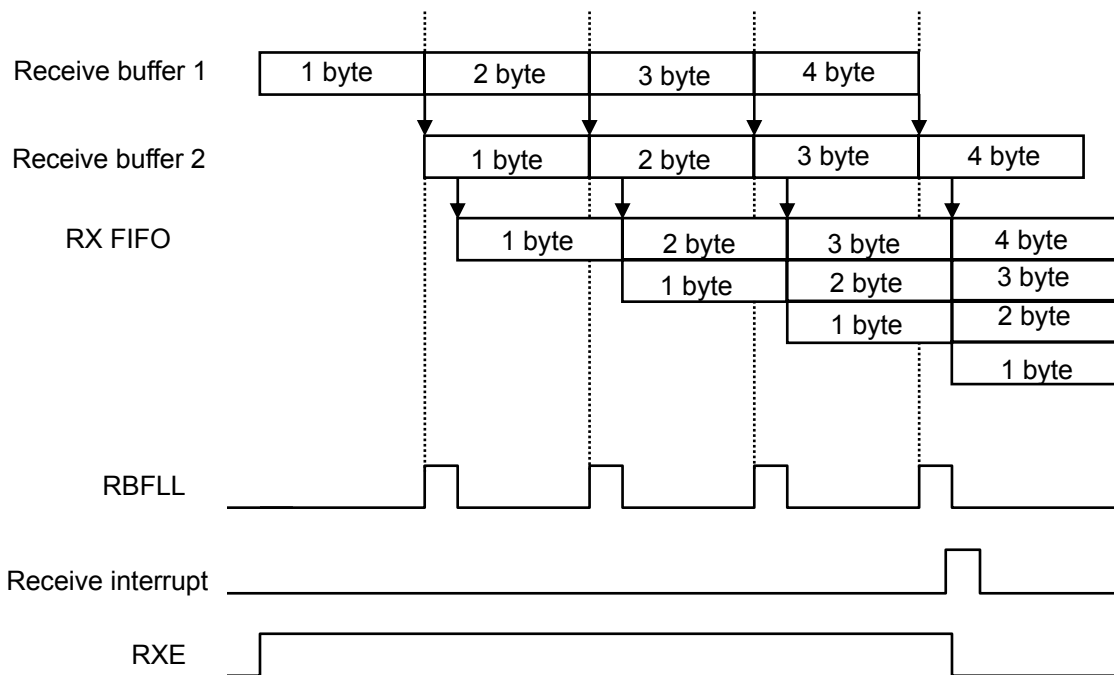
Fig. 3.10.2 Receive FIFO Operation

② I/O interface mode with SCLK input:

The following example describes the case a 4-byte data stream is received:

Data transmission can be initiated by setting the transfer mode to half duplex.

SC0FCNF <4:0> = 10101:   Automatically allows continued reception after reaching the fill
                           level. The number of bytes to be used in the receive FIFO is
                           the maximum allowable number.

SC0RFC <RIL[1:0]> = 00:   Sets the interrupt to be generated at fill level 4.

SC0RFC <RFCS,RFIS> = 10:   Clears receive FIFO and sets the condition of interrupt
                           generation

In this condition, 4-byte data reception may be initiated by writing "1" to the RXE bit. After receiving 4 bytes, receive FIFO interrupt is generated. This setting enables the next data reception as well. The next 4 bytes can be received before all the data is read from FIFO.
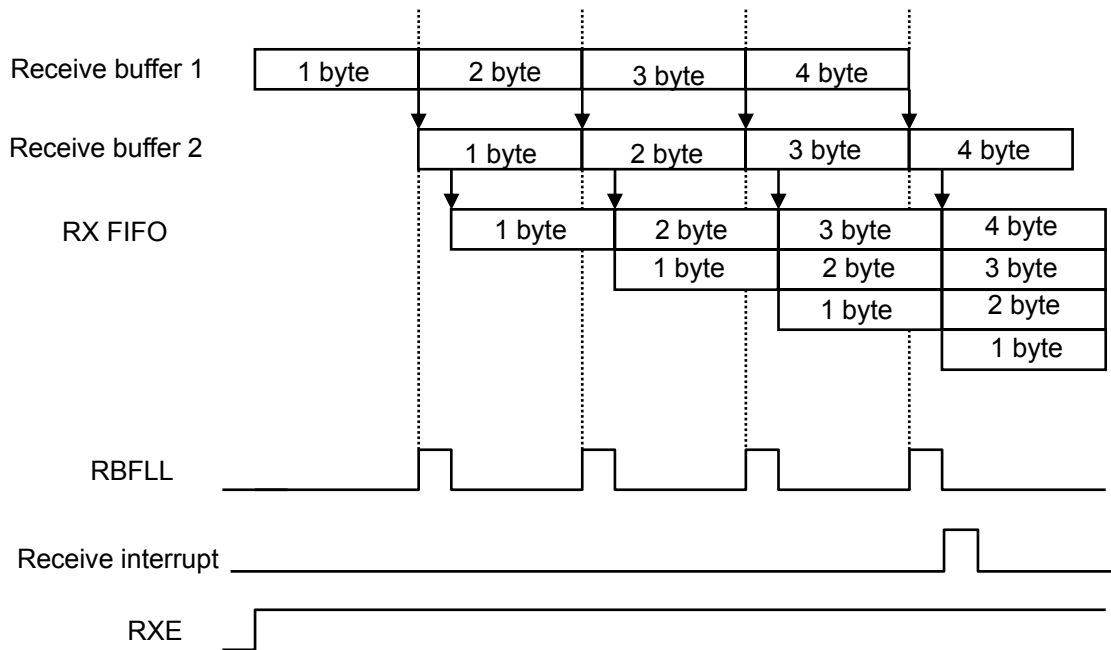
Fig. 3.10.4 Receive FIFO Operation

3.10.3.9   Transmit Counter

The transmit counter is a 4-bit binary counter used in the asynchronous communication (UART) mode. It is counted by SIOCLK as in the case of the receive counter and generates a transmit clock (TXDCLK) on every 16th clock pulse.
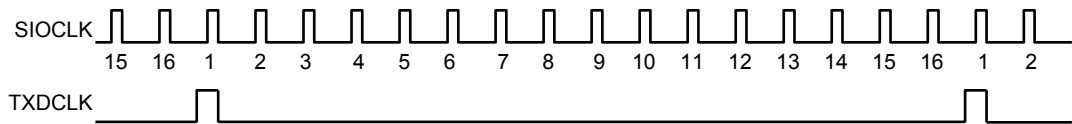
Fig. 3.10.5   Transmit Clock Generation

3.10.3.10   Transmit Control Uit

- I/O interface mode:

    In the SCLK output mode with SC0CR <IOC> set to "0," each bit of data in the transmit buffer is output to the TXD0 pin on the rising edge of the shift clock output from the SCLK0 pin.

    In the SCLK input mode with SC0CR <IOC> set to "1," each bit of data in the transmit buffer is output to the TXD0 pin on the rising or falling edge of the input SCLK signal according to the SC0CR <SCLKS> setting.

- Asynchronous (UART) mode:

    When the CPU writes data to the transmit buffer, data transmission is initiated on the rising edge of the next TXDCLK and the transmit shift clock (TXDSFT) is also generated.

- Handshake function

    The CTS0n pin enables frame by frame data transmission so that overrun errors can be prevented. This function can be enabled or disabled by SC0MOD0 <CTSE>.

    When the CTS0n pin is set to the "H" level, the current data transmission can be completed but the next data transmission is suspended until the CTS0n pin returns to the "L" level. However in this case, the INTTX0 interrupt is generated, the next transmit data is requested to the CPU, data is written to the transmit buffer, and it waits until it is ready to transmit data.

    Although no RTS0n pin is provided, a handshake control function can be easily implemented by assigning a port for the RTS0n function. By setting the port to "H" level upon completion of data reception (in the receive interrupt routine), the transmit side can be requested to suspend data transmission.
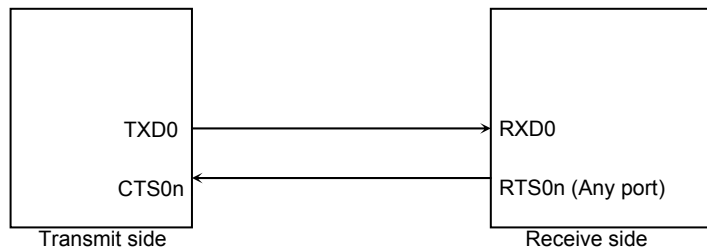
Fig. 3.10.6 Handshake Function
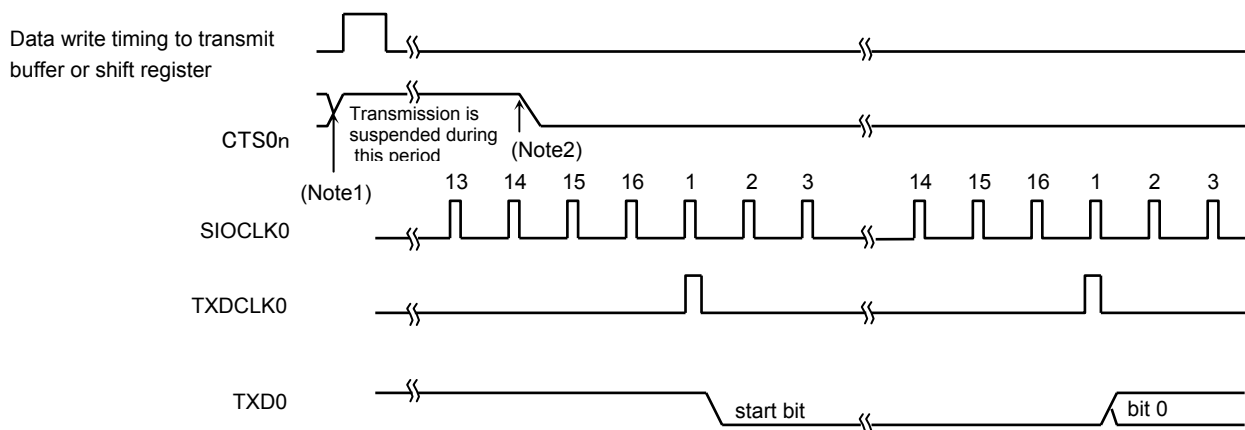
Fig. 3.10.7 CTSn(Clear to send) Signal Timing

(Note 1)   If the CTS0n signal is set to "H" during transmission, the next data transmission is suspended after the current transmission is completed.

(Note 2)   Data transmission starts on the first falling edge of the TXDCLK clock after CTS0n is set to "L."

3.10.3.11 Transmit Buffer

The transmit buffer (SC0BUF) is in a dual structure. The double buffering function may be enabled or disabled by setting the double buffer control bit <WBUF> in serial mode control register 2 (SC0MOD2). If double buffering is enabled, data written to Transmit Buffer 2 (SCOBUF) is moved to Transmit Buffer 1 (shift register).

If the transmit FIFO has been disabled (SCOFCNF <CNFG> = 0 or 1 and SC0MOD1＜FDPX[1:0]＞＝01), the INTTX0 interrupt is generated at the same time and the transmit buffer empty flag <TBEMP> of SC0MOD2 is set to "1." This flag indicates that Transmit Buffer 2 is now empty and that the next transmit data can be written. When the next data is written to Transmit Buffer 2, the <TBEMP> flag is cleared to "0."

If the transmit FIFO has been enabled (SCNFCNF <CNFG> = 1 and SC0MOD1＜FDPX[1:0]＞＝10/11), any data in the transmit FIFO is moved to the Transmit Buffer 2 and <TBEMP> flag is immediately cleared to "0." The CPU writes data to Transmit Buffer 2 or to the transmit FIFO.

If the transmit FIFO is disabled in the I/O interface SCLK input mode and if no data is set in Transmit Buffer 2 before the next frame clock input, which occurs upon completion of data transmission from Transmit Buffer 1, an under-run error occurs and a serial control register (SC0CR) <PERR> parity/under-run flag is set.

If the transmit FIFO is enabled in the I/O interface SCLK input mode, when data transmission from Transmit Buffer 1 is completed, the Transmit Buffer 2 data is moved to Transmit Buffer 1 and any data in transmit FIFO is moved to Transmit Buffer 2 at the same time.

If the transmit FIFO is disabled in the I/O interface SCLK output mode, when data in Transmit Buffer 2 is moved to Transmit Buffer 1 and the data transmission is completed, the SCLK output stops. So, no under-run errors can be generated.

If the transmit FIFO is enabled in the I/O interface SCLK output mode, the SCLK output stops upon completion of data transmission from Transmit Buffer 1 if there is no valid data in the transmit FIFO.

Note) In the I/O interface SCLK output mode, the SC0CR <PEER> flag is insignificant. In this case, the operation is undefined. Therefore, to switch from the SCLK output mode to another mode, SC0CR must be read in advance to initialize the flag.

If double buffering is disabled, the CPU writes data only to Transmit Buffer 1 and the transmit interrupt INTTX0 is generated upon completion of data transmission.

If handshaking with the other side is necessary, set the double buffer control bit <WBUF> to "0" (disable) to disable Transmit Buffer 2; any setting for the transmit FIFO should not be performed.

3.10.3.12 Transmit FIFO Buffer

In addition to the double buffer function already described, data may be stored using the transmit FIFO buffer. By setting <CNFG> of the SC0FCNF register and <FDPX1:0> of the SC0MOD1 register, the 4-byte transmit buffer can be enabled. In the UART mode or I/O interface mode, up to 4 bytes of data may be stored.

If data is to be transmitted with a parity bit in the UART mode, parity check must be performed on the receive side each time a data frame is received.

> **Note) When using transmit FIFO buffer, transmit FIFO buffer should be clear after setting the transfer mode to half duplex or full duplex and enable FIFO(setting SC0FCNF<CNFG>="1")**

3.10.3.13 Transmit FIFO Operation

① I/O interface mode with SCLK output (normal mode):

The following example describes the case a 4-byte data stream is transmitted:

Data transmission can be initiated by setting the transfer mode to half duplex.

SC0FCNF <4:0> = 01011:	Inhibits continued transmission after reaching the fill level.
SC0TFC <TIL[1:0]> = 00:	Sets the interrupt to be generated at fill level 0.
SC0TFC <TFCS, TFIS> = 01:	Clears transmit FIFO and sets the condition of interrupt generation

In this condition, data transmission can be initiated by writing 4 bytes of data to the transmit FIFO, and setting the <TXE> bit to "1." When the last transmit data is moved to the transmit buffer, the transmit FIFO interrupt is generated. When transmission of the last data is completed, the clock is stopped and the transmission sequence is terminated.
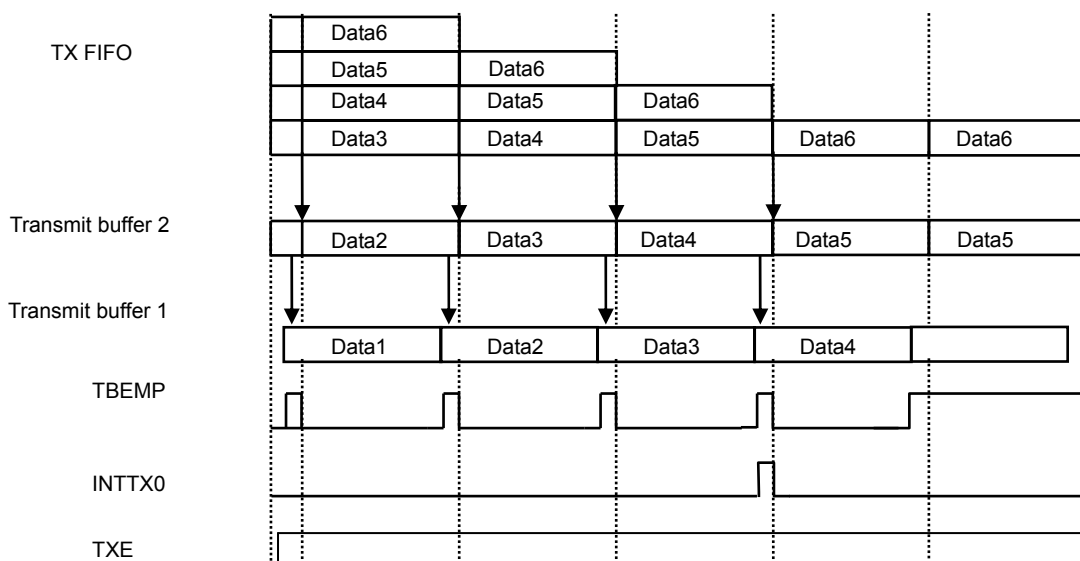
Fig. 3.10.8 Transmit FIFO Operation

② I/O interface mode with SCLK input (normal mode):

The following example describes the case a 4-byte data stream is transmitted:

Data transmission can be initiated by setting the transfer mode to half duplex.

SC0FCNF <4:0> = 01001:   Allows continued transmission after reaching the fill level.

SC0TFC <TIL[1:0]> = 01:   Clears the transmit FIFO and sets the condition of interrupt generation.

SC0TFC <TFCS,TFIS> = 000000:   Sets the interrupt to be generated at fill level 0.

In this condition, data transmission can be initiated along with the input clock by setting the transfer mode to half duplex, writing 4 bytes of data to the transmit FIFO, and setting the <TXE> bit to "1." When the last transmit data is moved to the transmit buffer, the transmit FIFO interrupt is generated.
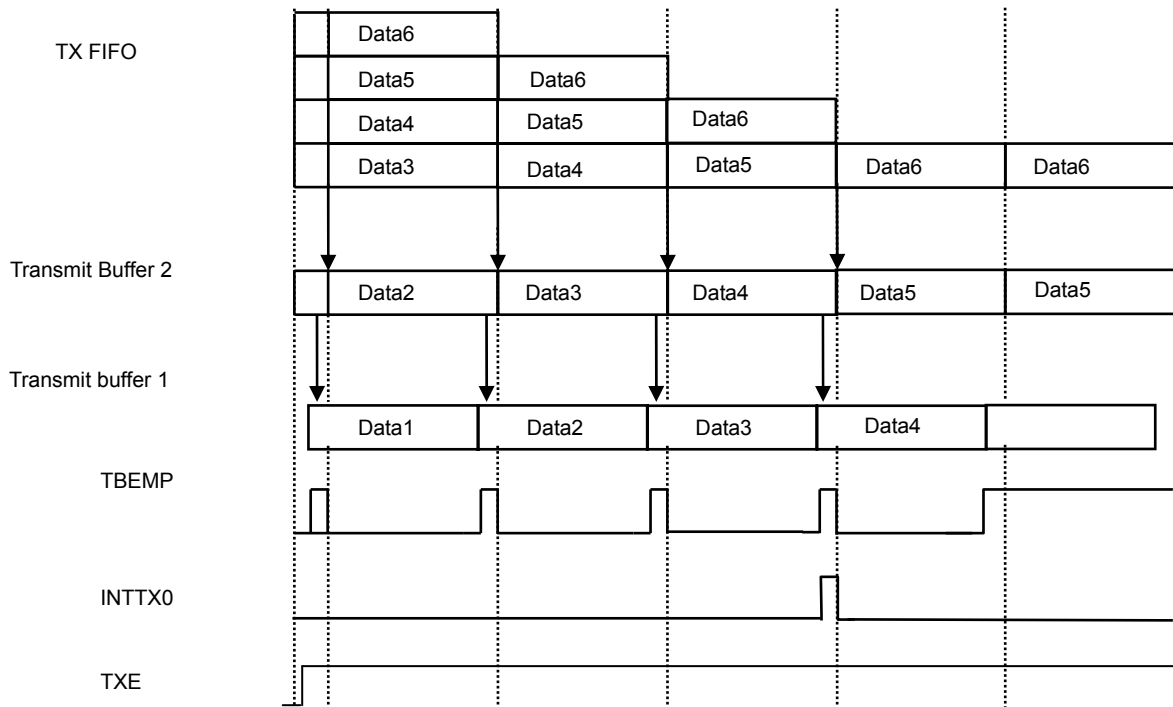
Fig. 3.10.9 Transmit FIFO Operation

3.10.3.14 Parity Control Circuit

If the parity addition bit <PE> of the serial control register SC0CR is set to "1," data is sent with the parity bit. Note that the parity bit may be used only in the 7- or 8-bit UART mode. The <EVEN> bit of SC0CR selects either even or odd parity.

Upon data transmission, the parity control circuit automatically generates the parity with the data written to the transmit buffer (SC0BUF). After data transmission is complete, the parity bit will be stored in SC0BUF bit 7 <TB7> in the 7-bit UART mode and in bit 7 <TB8> in the serial mode control register SC0MOD in the 8-bit UART mode. The <PE> and <EVEN> settings must be completed before data is written to the transmit buffer.

Upon data reception, the parity bit for the received data is automatically generated while the data is shifted to receive buffer 1 and moved to receive buffer 2 (SC0BUF). In the 7-bit UART mode, the parity generated is compared with the parity stored in SC0BUF <RB7>, while in the 8-bit UART mode, it is compared with the bit 7 <RB8> of the SC0CR register. If there is any difference, a parity error occurs and the <PERR> flag of the SC0CR register is set.

In the I/O interface mode, the SC0CR <PERR> flag functions as an under-run error flag, not as a parity flag.

3.10.3.15 Error Flag

Three error flags are provided to inprove the reliability of received data.

1. Overrun error <OERR>: Bit 4 of the serial control register SC0CR

   In both UART and I/O interface modes, this bit is set to "1" when an error is generated by completing the reception of the next frame receive data before the receive buffer has been read. If the receive FIFO is enabled, the received data is automatically moved to the receive FIFO and no overrun error will be generated until the receive FIFO is full (or until the usable bytes are fully occupied). This flag is set to "0" when it is read. In the I/O interface SCLK output mode, no overrun error is generated and therefore, this flag is inoperative and the operation is undefined.

2. Parity error/under-run error <PERR>: Bit 3 of the SC0CR register

   In the UART mode, this bit is set to "1" when a parity error is generated. A parity error is generated when the parity generated from the received data is different from the parity received. This flag is set to "0" when it is read.

   In the I/O interface mode, this bit indicates an under-run error. When the double buffer control bit <WBUF> of the serial mode control register SC0MOD2 is set to "1" in the SCLK input mode, if no data is set to the transmit double buffer before the next data transfer clock after completing the transmission from the transmit shift register, this error flag is set to "1" indicating an under-run error. If the transmit FIFO is enabled, any data content in the transmit FIFO will be moved to the buffer. When the transmit FIFO and the double buffer are both empty, an under-run error will be generated. Because no under-run errors can be generated in the SCLK output mode, this flag is inoperative and the operation is undefined. If Transmit Buffer 2 is disabled, the under-run flag <PERR> will not be set. This flag is set to "0" when it is read.

3. Framing error <FERR>: Bit 2 of the SC0CR register

In the UART mode, this bit is set to "1" when a framing error is generated. This flag is set to "0" when it is read. A framing error is generated if the corresponding stop bit is determined to be "0" by sampling the bit at around the center. Regardless of the <SBLEN> (stop bit length) setting of the serial mode control register 2, SC0MOD2, the stop bit status is determined by only 1 bit on the receive side.

Operation mode	Error flag	Function
UART	OERR	Overrun error flag
	PERR	Parity error flag
	FERR	Framing error flag
I/O Interface (SCLK input)	OERR	Overrun error flag
	PERR	Underrun error flag (WBUF = 1)
		Fixed to 0 (WBUF = 0)
	FERR	Fixed to 0
I/O Interface (SCLK output)	OERR	Operation undefined
	PERR	Operation undefined
	FERR	Fixed to 0

3.10.3.16 Direction of Data Transfer

In the I/O interface mode, the direction of data transfer can be switched between "MSB first" and "LSB first" by the data transfer direction setting bit <DRCHG> of the SC0MOD2 serial mode control register 2. Don't switch the direction when data is being transferred.

3.10.3.17 Stop Bit Length

In the UART transmission mode, the stop bit length can be set to either 1 or 2 bits by bit 4 <SBLEN> of the SC0MOD2 register.

3.10.3.18 Status Flag

If the double buffer function is enabled (SC0MOD2 <WBUF> = "1"), the bit 6 flag <RBFLL> of the SC0MOD2 register indicates the condition of receive buffer full. When one frame of data has been received and transferred from buffer 1 to buffer 2, this bit is set to "1" to show that buffer 2 is full (data is stored in buffer 2). When the receive buffer is read by CPU/DMAC, it is cleared to "0." If <WBUF> is set to "0," this bit is insignificant and must not be used as a status flag. When double buffering is enabled (SC0MOD2 <WBUF> = "1"), the bit 7 flag <TBEMP> of the SC0MOD2 register indicates that Transmit Buffer 2 is empty. When data is moved from Transmit Buffer 2 to Transmit Buffer 1 (shift register), this bit is set to "1" indicating that Transmit Buffer 2 is now empty. When data is set to the transmit buffer by CPU/DMAC, the bit is cleared to "0." If <WBUF> is set to "0," this bit is insignificant and must not be used as a status flag.

3.10.3.19 **Configurations of Transmit/Receive Buffer**

		<WBUF> = 0	<WBUF> = 1
UART	Transmit buffer	Single	Double
	Receive buffer	Double	Double
I/O Interface (SCLK input)	Transmit buffer	Single	Double
	Receive buffer	Double	Double
I/O Interface (SCLK output)	Transmit buffer	Single	Double
	Receive buffer	Single	Double

3.10.3.20 Software reset

Software reset is generated by writing the bits 1 and 0 of SC0MOD2 <SWRST1:0> as "10" followed by "01". As a result, SC0MOD0<RXE>, SC0MOD1<TXE>, SC0MOD2<TBEMP>,<RBFLL>,<TXRUN> of mode registers and SC0CR<OERR>, <PERR>, <FERR> of control registers and internal circuit is initialized. Other states are maintained.

3.10.3.21 Signal Generation Timing

① UART Mode:

Receive Side

Mode	9-bit	8-bit with parity	8-bit, 7-bit, and 7-bit with parity
Interrupt generation timing	Around the center of the 1st stop bit	Around the center of the 1st stop bit	Around the center of the 1st stop bit
Framing error generation timing	Around the center of the stop bit	Around the center of the stop bit	Around the center of the stop bit
Parity error generation timing	—	Around the center of the last (parity) bit	Around the center of the last (parity) bit
Overrun error generation timing	Around the center of the stop bit	Around the center of the stop bit	Around the center of the stop bit

Transmit Side

Mode	9-bit	8-bit with parity	8-bit, 7-bit, and 7-bit with parity
Interrupt generation timing (<WBUF> = 0)	Just before the stop bit is sent	Just before the stop bit is sent	Just before the stop bit is sent
Interrupt generation timing (<WBUF> = 1)	Immediately after data is moved to transmit buffer 1 (just before start bit transmission)	Immediately after data is moved to transmit buffer 1 (just before start bit transmission).	Immediately after data is moved to transmit buffer 1 (just before start bit transmission)

② I/O interface mode:

Receive Side

Interrupt generation timing (<WBUF> = 0)	SCLK output mode	Immediately after the rising edge of the last SCLK
	SCLK input mode	Immediately after the rising or falling edge of the last SCLK (for rising or falling edge mode, respectively).
Interrupt generation timing (<WBUF> = 1)	SCLK output mode	Immediately after the rising edge of the last SCLK (just after data transfer to receive buffer 2) or just after receive buffer 2 is read.
	SCLK input mode	Immediately after the rising edge or falling edge of the last SCLK (right after data is moved to receive buffer 2).
Overrun error generation timing	SCLK input mode	Immediately after the rising or falling edge of the last SCLK (for rising or falling edge mode, respectively)

Transmit Side

Interrupt generation timing (<WBUF> = 0)	SCLK output mode	Immediately after the rising edge of the last SCLK
	SCLK input mode	Immediately after the rising or falling edge of the last SCLK (for rising or falling edge mode, respectively)
Interrupt generation timing (<WBUF> = 1)	SCLK output mode	Immediately after the rising edge of the last SCLK or just after data is moved to Transmit Buffer 1
	SCLK input mode	Immediately after the rising or falling edge of the last SCLK or just after data is moved to Transmit Buffer 1
Under-run error generation timing	SCLK input mode	Immediately after the falling or rising edge of the next SCLK

> **(Note 1)** Do not modify any control register when data is being sent or received (in a state ready to transmit or receive).
> **(Note 2)** Do not stop the receive operation (by setting SC0MOD0 <RXE> = "0") when data is being received.
> **(Note 3)** Do not stop the transmit operation (by setting SC0MOD1 <TXE> = "0") when data is being transmitted.

3.10.4   Register Description (Only for Channel 0)

The channel 0 registers are described here. Each register for all the channels operates in the same way.

3.10.4.1   Enable register

SC0EN

	7	6	5	4	3	2	1	0
bit Symbol								SIOE
Read/Write	R							R/W
After reset	0							0
Function	"0" is read.							SIO operation 0: disabled 1: enabled

<SIOE>:   Specified the SIO operation.
To use the SIO, enable the SIO operation.
When the operation is disabled, no clock is supplied to the other registers in the SIO module. This can reduce the power consumption.
If the SIO operation is executed and then disabled, the settings will be maintained in each register except SC0TFC<TIL[1:0]>.

(Note) When transmitting DMA by Transmit/Receive interrupt of SIO, firstly reset by setting SC0MOD2<SWRST[1:0]> register, next enable DMAC(DMA request waiting state), start SIO.

3.10.4.2   Buffer register

SC0BUF works as a transmit buffer for WR operation and as a receive buffer for RD operation.

SC0BUF

	7	6	5	4	3	2	1	0
bit Symbol	TB/RB							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	TB : Transmit buffer/FIFO RB : Receive buffer/FIFO							

<TB7:0>   Transmit buffer (at WR operation).

<RB7:0>   Receive buffer (at RD operation).

3.10.4.3 Control register

SC0CR		7	6	5	4	3	2	1	0
	bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
	Read/Write	R	R/W		R (Cleared to "0" when read)			R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	Receive data bit 8 **(For UART)**	Parity **(For UART)** 0: Odd 1: Even	Add parity **(For UART)** 0: Disabled 1: Enabled	0: Normal operation 1: Error			0: SCLK0 1: SCLK0	**(For I/O interface)** 0: Baud rate generator 1: SCLK0 pin input
					Overrun	Parity/ underrun	Framing		

<RB8>: 9th bit of the received data in the 9 bits UART mode.

<EVEN>: Selects even or odd parity.
"0": odd parity.
"1": even parity.
The parity bit may be used only in the 7- or 8-bit UART mode.

<PE>: Controls enabling/ disabling parity.
The parity bit may be used only in the 7- or 8-bit UART mode.

<OERR>: Error flag **(see note)**
<PERR>: Indicate overrun error, parity error, underrun error and framing error.
<FERR>:

<SCLKS>: Selects edge for data transmission and reception.
"0": Data transmit/receive at rising edges of SCLK0
"1": Data transmit/receive at falling edges of SCLK0

<IOC>: Selects input clock in the I/O interface mode.
"0": baud rate generator
"1": SCLK0 pin input

(Note) **Any error flag is cleared when read.**

3.10.4.4 Mode Control register 0

SC0MOD0

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE	RXE	WU	SM		SC	
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transmit data bit 8	Handshake function control 0: CTS diable 1: CTS enable	Receive control 0: Reception disabled 1: Reception enabled	Wake-up function 0: Reception disabled 1: Reception enabled	Serial transfer mode 00: I/O interface mode 01: 7-bit length UART mode 10: 8-bit length UART mode 11: 9-bit length UART mode		Serial transfer clock **(for UART)** 00: Timer TB5OUT 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clock (SCLK0 input)	

<TB8>: Writes the 9th bit of transmit data in the 9 bits UART mode.

<CTSE>: ontrols handshake function.
Setting "1" enables handshake function using CTSn pin.

<RXE>: Controls reception **(see note).**
Set <RXE> after setting each mode register (SC0MOD0, SC0MOD1 and SC0MOD2).

<WU>: Controls wake-up function.
This function is available only at 9-bit UART mode.

	9-bit UART mode	Other modes
0	Interrupt when received	don't care
1	Interrupt only when RB9=1	

<SM1 : 0>: Specifies transfer mode.

<SC1 : 0>: Selects the serial transfer clock in the UART mode.
As for the I/O interface mode, the serial transfer clock can be set in the control register SC0CR.

(Note) **With <RXE> set to "0," set each mode register (SC0MOD0, SC0MOD1 and SC0MOD2). Then set <RXE> to "1."**

3.10.4.5 Mode Control register 1

SC0MOD1

	7	6	5	4	3	2	1	0
bit Symbol	I2SC	FDPX		TXE	SINT			－
Read/Write	R/W	R/W		R/W	R/W			R/W
After reset	0	0	0	0	0	0	0	0
Function	IDLE 0: Stop 1: Start	Transfer mode setting 00: Transfer prohibited 01: Half duplex(RX) 10:Half duplex(TX) 11:Full duplex		Transmit control 0:Disabled 1: Enabled	Interval time of continuous transmission **(for I/O interface)** 000: None 100:8SCLK 001:1SCLK 101:16SCLK 010:2SCLK 110:32SCLK 011:4SCLK 111:64SCLK			Write "0".

<I2SC> :　　　Specifies the IDLE mode operation.

<FDPX[1:0]> :　Configures the transfer mode in the I/O interface mode. Also configures the FIFO if it is enabled. In the UART mode, it is used only to specify the FIFO configuration.

<TXE> :　　　This bit enables transmission and is valid for all the transfer modes (**see note**). If disabled while transmission is in progress, transmission is inhibited only after the current frame of data is completed for transmission.

<SINT[2:0]> :　Specifies the interval time of continuous transmission when double buffering or FIFO is enabled in the I/O interface mode. This parameter is valid only for the I/O interface mode when SCLK0 pin input is not selected.

(Note)	**Specify the mode first and then specify the <TXE> bit.**

(Note) When usung Full duplex transmission of SIO, can not use DMA transmit.

3.10.4.6 Mode Control register 2

SC0MOD2

	7	6	5	4	3	2	1	0
bit Symbol	TBEMP	RBFLL	TXRUN	SBLEN	DRCHG	WBUF	SWRST	
Read/Write	R			R/W				
After reset	1	0	0	0	0	0	0	0
Function	Transmit buffer empty flag 0: full 1: Empty	Receive Buffer full flag 0: Empty 1: full	In transmission flag 0: Stop 1: Start	STOP bit **(for UART)** 0: 1-bit 1: 2-bit	Setting transfer direction 0: LSB first 1: MSB first	W-buffer 0: Disabled 1: Enabled	SOFT RESET Overwrite "01" on "10"to reset.	

<TBEMP>: This flag shows that the transmit double buffers are empty. When data in the transmit double buffers is moved to the transmit shift register and the double buffers are empty, this bit is set to "1." Writing data again to the double buffers sets this bit to "0."
If double buffering is disabled, this flag is insignificant.

<RBFLL>: This is a flag to show that the receive double buffers are full. When a receive operation is completed and received data is moved from the receive shift register to the receive double buffers, this bit changes to "1" while reading this bit changes it to "0."
If double buffering is disabled, this flag is insignificant.

<TXRUN>: This is a status flag to show that data transmission is in progress.
<TXRUN> and <TBEMP> bits indicate the following status.

<TXRUN>	<TBEMP>	Status
1	-	Transmission in progress
0	1	Transmission completed
	0	Wait state with data in TX buffer

<SBLEN>: This specifies the length of stop bit transmission in the UART mode. On the receive side, the decision is made using only a single bit regardless of the <SBLEN> setting.

<DRCHG>: Specifies the direction of data transfer in the I/O interface mode. In the UART mode, it is fixed to LSB first.

<WBUF>: This parameter enables or disables the transmit/receive buffers to transmit (in both SCLK output/input modes) and receive (in SCLK output mode) data in the I/O interface mode and to transmit data in the UART. When receiving data in the I/O interface mode (I SCLK input) and UART mode, double buffering is enabled in both cases that 0 or 1 is set to <WBUF> bit.

<SWRST[1:0]>: Overwriting "01" in place of "10" generates a software reset. When this software reset is executed, the following bits and their internal circuits are initialized **(see note 1, 2 and 3)**.

Register name	Bit
SC0MOD0	RXE
SC0MOD1	TXE
SC0MOD2	TBEMP,RBFLL,TXRUN,
SC0CR	OERR,PERR,FERR

(Note 1)	**While data transmission is in progress, any software reset operation must be executed twice in succession.**
(Note 2)	**A software reset requires 2 clocks-duration at the time between the end of recognition and the start of execution of software reset instruction.**
(Note 3)	**A software reset initializes other bits. Resetting a mode register and a control register are needed.**

3.10.4.7 Baud rate generator control register(SC0BRCR)
Baud rate generator control register 2(SC0BRADD)

SC0BRCR

	7	6	5	4	3	2	1	0
bit Symbol	−	BRADDE	BRCK		BRS			
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Write "0".	N + (16 − K)/ 16 divider function 0: disabled 1: enabled	Select input clock to the baud rate generator 00: ϕT1 01: ϕT4 10: ϕT16 11: ϕT64		Division ratio "N" 0000: 16 0001: 1 0010: 2 : 1111: 15			

SC0BRADD

	7	6	5	4	3	2	1	0
bit Symbol					BRK			
Read/Write	R				R/W			
After reset	0				0	0	0	0
Function	"0" is read.				Specify K for the "N + (16 − K)/16" division 0000: Prohibited 0001: K=1 0010: K=2 : 1111: K=15			

<RBADDE>:   Specifies N + (16-K)/16 division function.
                N + (16-K)/16 division function can only be used in the UART mode.

<RBCK[1:0]>:   Specifies the baud rate generator input clock.

<RBS[3:0]>:   Specifies division ratio "N".

<RBK[3:0]>:   Specifies K for the "N+(16-K)/16" division.

The division ratio of the baud rate generator can be specified in the registers shown above.
Table 3.10.5 lists the settings of baud rate generator division ratio.

Tabel 3.10.5 Setting division ratio

	BRADDE=0	BRADDE=1 **(Note 1)** (Only UART)
BRS	Specify "N" **(Note 2) (Note 3)**	
BRK	No setting required	Setting "K" **(Note 4)**
Division ratio	Divide by N	$N + \frac{(16-K)}{16}$ division

(Note 1)   To use the "N + (16 - K)/16" division function, be sure to set BR0K <BR0ADDE> to "1" after setting the K value to BR0K. The "N + (16 - K)/16" division function can only be used in the UART mode.

(Note 2)   The division ratio "1" of the baud rate generator can be specified only when
　・the "N + (16 - K)/16" division function is not used in the UART mode.
　・double buffering is used in the I/O interface mode.

(Note 3)   As a division ratio, 1 ("0001") or 16 ("0000") cannot be applied to N when using the "N + (16 - K)/16" division function.

(Note 4)   Specifying "K = 0" is prohibited.

3.10.4.8 FIFO configuration register

	7	6	5	4	3	2	1	0
bit Symbol	Reserved	Reserved	Reserved	RFST	TFIE	RFIE	RXTXCNT	CNFG
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	**Be sure to write "000".**			Bytes used in RX FIFO 0: Maximum 1:Same as FILL level of RX FIFO	TX interrupt for TX FIFO 0: Disabled 1: Enabled	RX interrupt for RX FIFO 0:Disabled 1: Enabled	Automatic disable of RXE/TXE 0:None 1:Auto disable	FIFO enable 0: Disabled 1: Enabled

SC0FCNF

<RFST>:   When RX FIFO is enabled, the number of RX FIFO bytes to be used is selected **(see note)**.
0: The maximum number of bytes of the FIFO configured (see also <CNFG>).
1: Same as the fill level for receive interrupt generation specified by SC0RFC <RIL[1:0]>.

<TFIE>:   When TX FIFO is enabled, transmit interrupts are enabled or disabled by this parameter.

<RFIE>:   When RX FIFO is enabled, receive interrupts are enabled or disabled by this parameter.

<RXTXCNT>:   Controls automatic disabling of transmission and reception.
The mode control register SCOMOD1 <FDPX[1:0]> is used to set the types of TX/RX.
Setting "1" enables to operate as follows.

Half duplex RX	When the RX FIFO is filled up to the specified number of valid bytes, SC0MOD0<RXE> is automatically set to "0" to inhibit further reception.
Half duplex TX	When the TX FIFO is empty, SC0MOD1<TXE> is automatically set to "0" to inhibit further transmission.
Full duplex	When either of the above two conditions is satisfied, TXE/RXE are automatically set to "0" to inhibit further transmission and reception.

<CNFG>:   Enables FIFO.
If enabled, the SCOMOD1 <FDPX[1:0]> setting automatically configures FIFO as follows:
(The type of TX/RX can be specified in the mode control register 1 SC0MOD1<FDPX1:0>).

Half duplex RX	RX FIFO 4byte
Half duplex TX	TX FIFO 4byte
Full duplex	RX FIFO 2byte＋TX FIFO 2byt

(Note1)	**Regarding TX FIFO, the maximum number of bytes being configured is always available. The available number of bytes is the bytes already written to the TX FIFO.**
(Note2)	**(Note) When transmitting DMA by Transmit/Receive interrupt of SIO, can not use FIFO.**

3.10.4.9   RX FIFO configuration register

	7	6	5	4	3	2	1	0
bit Symbol	RFCS	RFIS					RIL	
Read/Write	W	R/W		R			R/W	
After reset	0	0		0			0	0
Function	RX FIFO clear 1: Clear "0" is read.	Select interrupt generatio n condition 0: when the data reaches to the specified fill level. 1: when the data reaches to the specified fill level or the data exceeds the specified fill level at the time data is read.	"0" is read.				FIFO fill level to generate RX interrupts 00:4byte(2 Byte at full duplex) 01:1byte 10:2byte 11:3byte	

SC0RFC

<RFCS>:   Clears RX FIFO
            Setting "1" clears RX FIFO and "0" is always read.

<RFIS>:   Specifies the condition of interrupt generation.
            0: An interrupt is generated when it reaches to the specified fill level.
            An interrupt is generated when it is reaches to the specified fill level or if it exceeds the
            specified fill level at the time data is read.

<RIL[1:0]>:   Specifies FIFO fill level**(see note).**

	Other than full duplex	Full duplex
00	4byte	2byte
01	1byte	1byte
10	2byte	2byte
11	3byte	1byte

(Note1)   RIL1 is ignored when FDPX1:0 = 11 (full duplex)

(Note2)   DMA transmit does not start by interrupt generating by fill level of FIFO.

3.10.4.10 TX FIFO configuration register

SC0TFC

	7	6	5	4	3	2	1	0
bit Symbol	TFCS	TFIS					TIL	
Read/Write	w	R/W			R		R/W	
After reset	0	0			0		0	0
Function	TX FIFO clear 1:Clear Always reads "0".	Select interrupt generation condition 0: when the data reaches to the specified fill level. 1: when the data reaches to the specified fill level or the data cannot reach the specified fill level at the time new data is read.	"0" is read.				FIFO fill level to generate TX interrupts. 00:Empty 01:1byte 10:2byte 11:3byte Note: TIL1 is ignored when FDPX1:0=11 (full duplex).	

<TFCS>: Clears TX FIFO.
Setting "1" clears TX FIFO and "0" is always read.

<TFIS>: Selects interrupt generation condition.
0: An interrupt is generated when the data reaches to the specified fill level.
1: An interrupt is generated when the data reaches to the specified fill level or the data cannot reach the
specified fill level at the time new data is read.

<TIL[1:0]>: Selects FIFO fill level **(see note).**

	Other than full duplex	Full duplex
00	Empty	Empty
01	1byte	1byte
10	2byte	Empty
11	3byte	1byte

(Note1) **TIL1 is ignored when FDPX1:0 = 11 (full duplex).**

(Note2) **When setting SC0EN<0>=0 (SIO function disable, clock stop) or shifting standby mode(IDLE,SLEEP,STOP)(function disable in standby mode, clock stop), setting SC0TFC register again.**

(Note3) **DMA transmit does not start by interrupt generating by fill level of FIFO.**

3.10.4.11 RX FIFO status register

	7	6	5	4	3	2	1	0
bit Symbol	ROR						RLVL	
Read/Write	R			R			R	
After reset	0			0		0	0	0
Function	RX FIFO Overrun 1: Generated	"0" is read.				Status of RX FIFO fill level 000:Empty 001:1Byte 010:2Byte 011:3Byte 100:4Byte		

SC0RST

<ROR>:   Flags for RX FIFO overrun.
                 When the overrun occurs, these bits are set to "1" **(see note).**
<RLVL[2:0]>:   Shows the fill level of RX FIFO.

(Note)	**The <ROR> bit is cleared to "0" when receive data is read from the SC0BUF register.**

3.10.4.12 TX FIFO status register

	7	6	5	4	3	2	1	0
bit Symbol	TUR						TLVL	
Read/Write	R			R			R	
After reset	0			0		0	0	0
Function	TX FIFO Under run 1:Generated Cleared by writing FIFO	"0" is read.				Status of TX FIFO fill level 000:Empty 001:1Byte 010:2Byte 011:3Byte 100:4Byte		

SC0TST

<TUR>:   Flags for TX FIFO underrun. When the underrun occurs, these bits are set to "1" **(see note).**

<TLVL[2:0]>:   Shows the fill level of TX FIFO.

(Note)	**The <TUR> bit is cleared to "0" when transmit data is written to the SC0BUF register.**

3.10.5 Operation in Each Mode

3.10.5.1 Mode 0 (I/O interface mode)

Mode 0 consists of two modes, the "SCLK output" mode to output synchronous clock and the "SCLK input" mode to accept synchronous clock from an external source. The following operational descriptions are for the case use of FIFO is disabled. For details of FIFO operation, refer to the previous sections describing receive/transmit FIFO functions.

①   Transmitting data

SCLK output mode

In the SCLK output mode, if SC0MOD2<WBUF> is set to "0" and the transmit double buffers are disabled, 8 bits of data are output from the TXD0 pin and the synchronous clock is output from the SCLK0 pin each time the CPU writes data to the transmit buffer. When all data is output, the INTTX0 interrupt is generated.

If SC0MOD2 <WBUF> is set to "1" and the transmit double buffers are enabled, data is moved from Transmit Buffer 2 to Transmit Buffer 1 when the CPU writes data to Transmit Buffer 2 while data transmission is halted or when data transmission from Transmit Buffer 1 (shift register) is completed. When data is moved from Transmit Buffer 2 to Transmit Buffer 1, the transmit buffer empty flag SC0MOD2 <TBEMP> is set to "1," and the INTTX0 interrupt is generated. If Transmit Buffer 2 has no data to be moved to Transmit Buffer 1, the INTTX0 interrupt is not generated and the SCLK0 output stops.

<WBUF>="0" (if double buffering is disabled)

<WBUF>="1" (if double buffering is enabled and there is data in buffer 2)

<WBUF>="1" (if double buffering is enabled and there is no data in buffer 2)
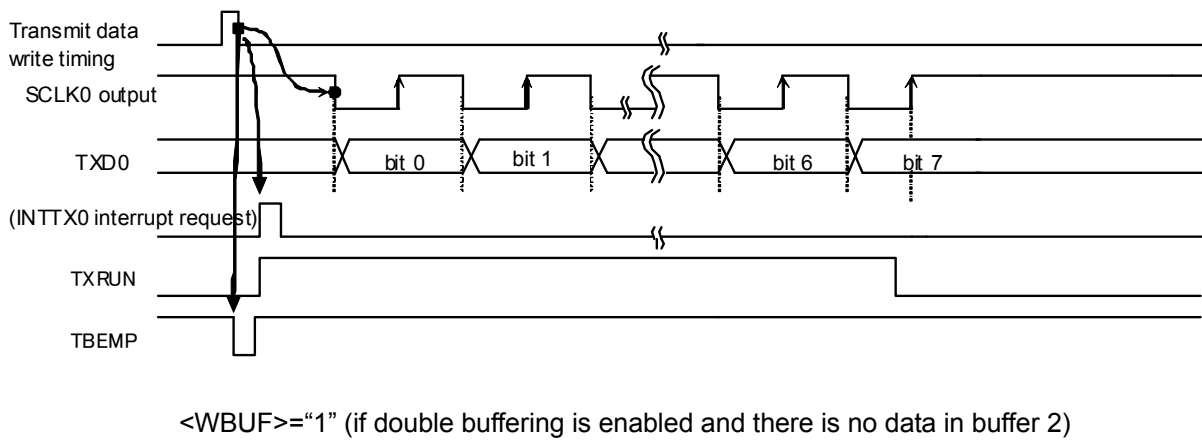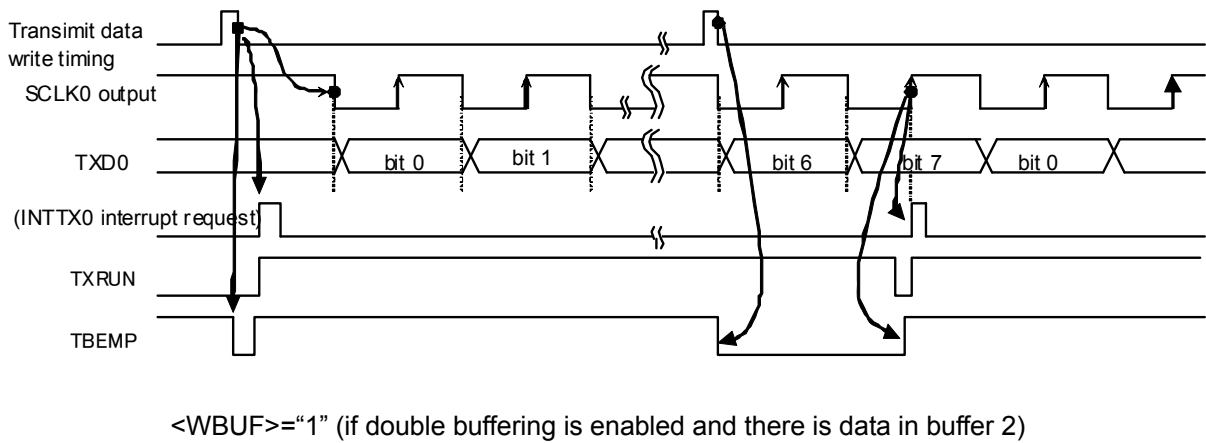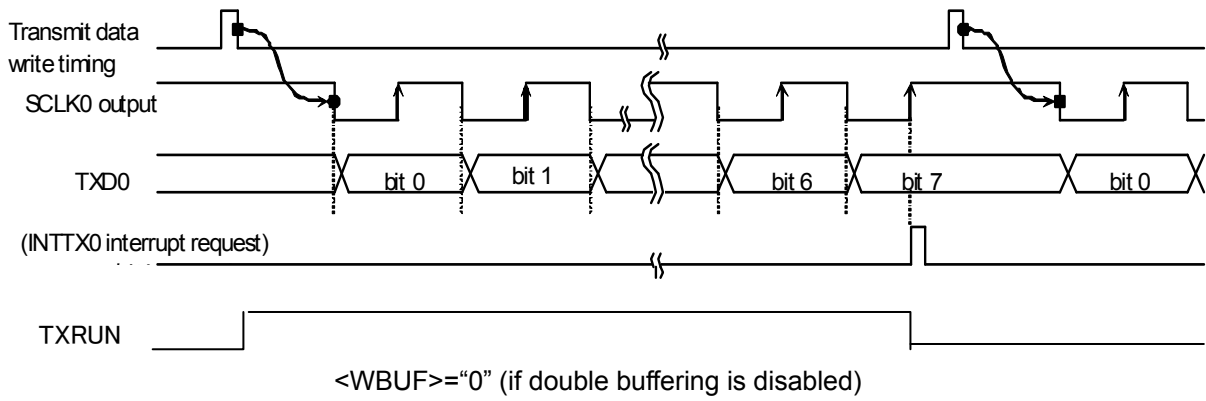
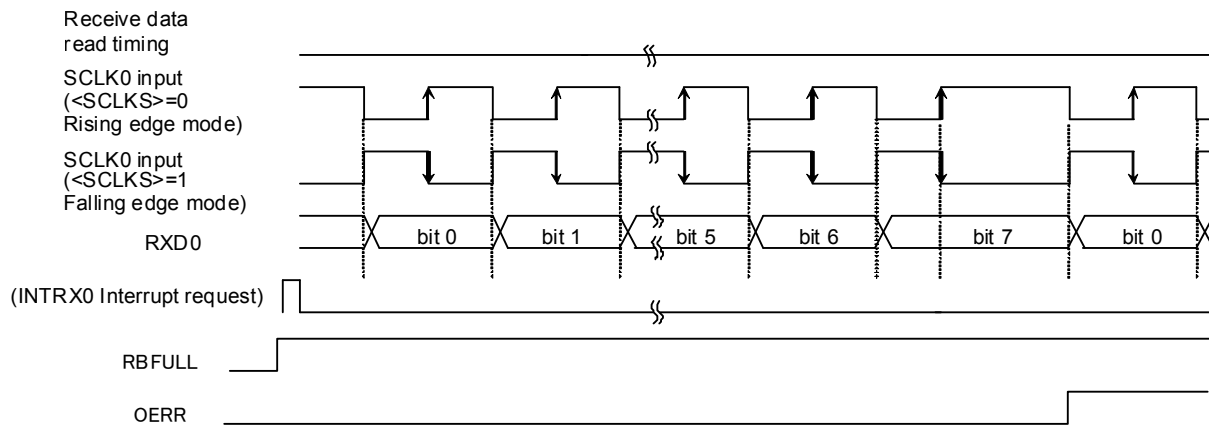Fig. 3.10.10 Transmit Operation in the I/O Interface Mode (SCLK0 Output Mode)

SCLK input mode

In the SCLK input mode, if SC0MOD2 <WBUF> is set to "0" and the transmit double buffers are disabled, 8-bit data that has been written in the transmit buffer is output from the TXD0 pin when the SCLK0 input becomes active. When all 8 bits are sent, the INTTX0 interrupt is generated. The next transmit data must be written before the timing point "A" as shown in Fig. 3.10.11.

If SC0MOD2 <WBUF> is set to "1" and the transmit double buffers are enabled, data is moved from Transmit Buffer 2 to Transmit Buffer 1 when the CPU writes data to Transmit Buffer 2 before the SCLK0 becomes active or when data transmission from Transmit Buffer 1 (shift register) is completed. As data is moved from Transmit Buffer 2 to Transmit Buffer 1, the transmit buffer empty flag SC0MOD2 <TBEMP> is set to "1" and the INTTX0 interrupt is generated. If the SCLK0 input becomes active while no data is in Transmit Buffer 2, although the internal bit counter is started, an under-run error occurs and 8-bit dummy data (FFh) is sent.
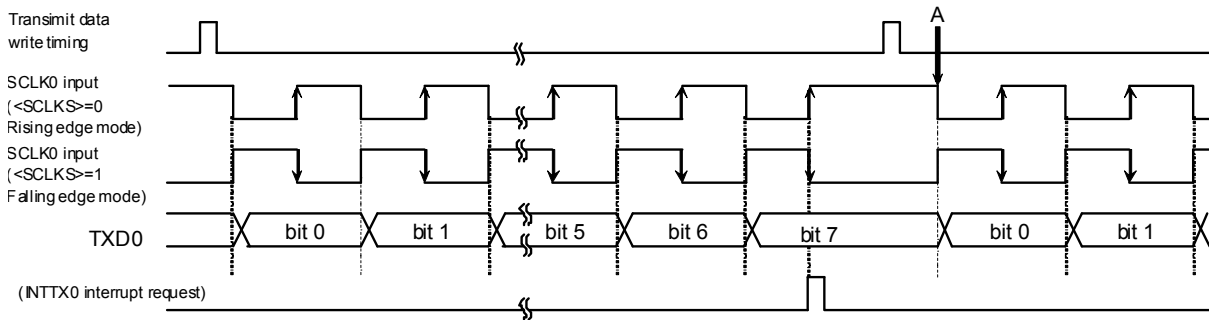
<WBUF>="0" (if double buffering is disabled
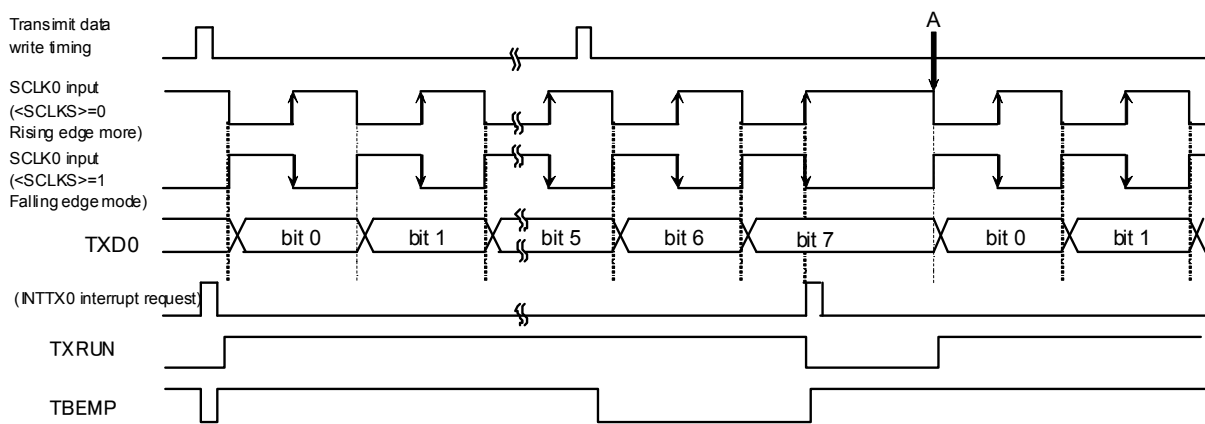
<WBUF>="1" (if double buffering is enabled and there is data in buffer 2)
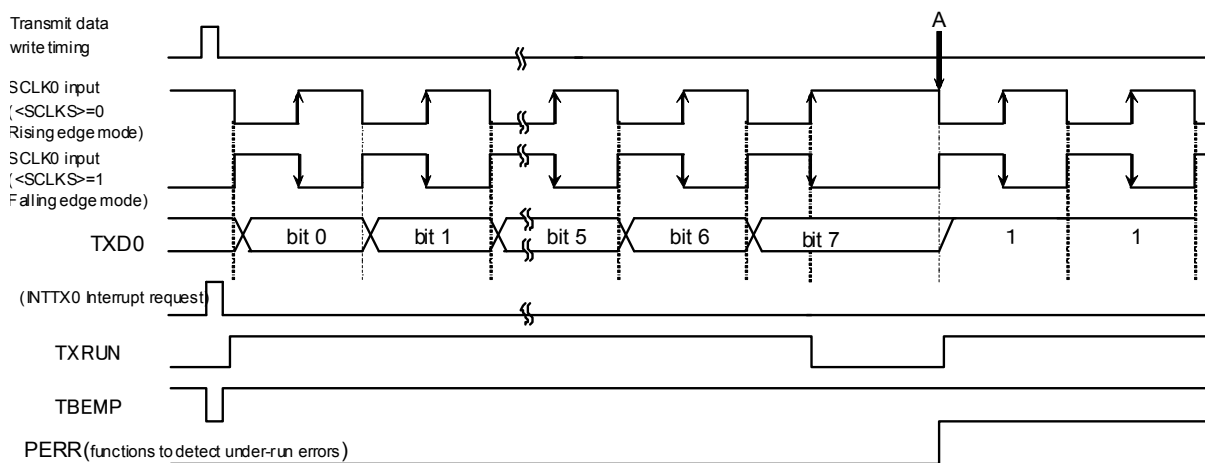
<WBUF>="1" (if double buffering is enabled and there is no data in buffer 2)

Fig. 3.10.11 Transmit Operation in the I/O Interface Mode (SCLK0 Input Mode)

② Receiving data
SCLK output mode

In the SCLK output mode, if SC0MOD2 <WBUF> = "0" and receive double buffering is disabled, a synchronous clock pulse is output from the SCLK0 pin and the next data is shifted into receive buffer 1 each time the CPU reads received data. When all the 8 bits are received, the INTRX0 interrupt is generated.

The first SCLK output can be started by setting the receive enable bit SC0MOD0 <RXE> to "1." If the receive double buffering is enabled with SC0MOD2 <WBUF> set to "1," the first frame received is moved to receive buffer 2 and receive buffer 1 can receive the next frame successively. As data is moved from receive buffer 1 to receive buffer 2, the receive buffer full flag SC0MOD2 <RBFULL> is set to "1" and the INTRX0 interrupt is generated.

While data is in receive buffer 2, if CPU/DMAC cannot read data from receive buffer 2 before completing reception of the next 8 bits, the INTRX0 interrupt is not generated and the SCLK0 clock stops. In this state, reading data from receive buffer 2 allows data in receive buffer 1 to move to receive buffer 2 and thus the INTRX0 interrupt is generated and data reception resumes.
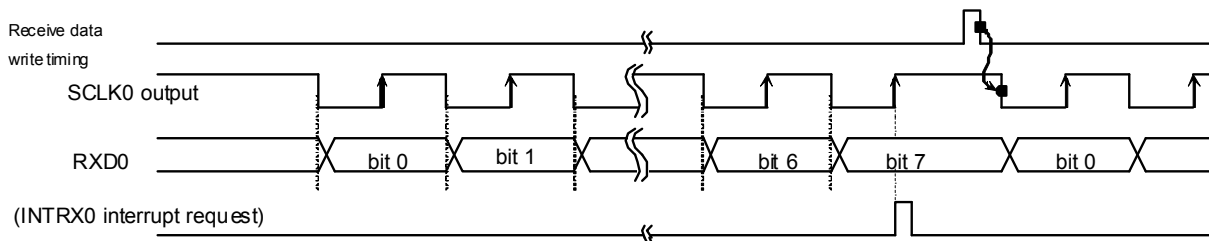
<WBUF>="0" (if double buffering is disabled)
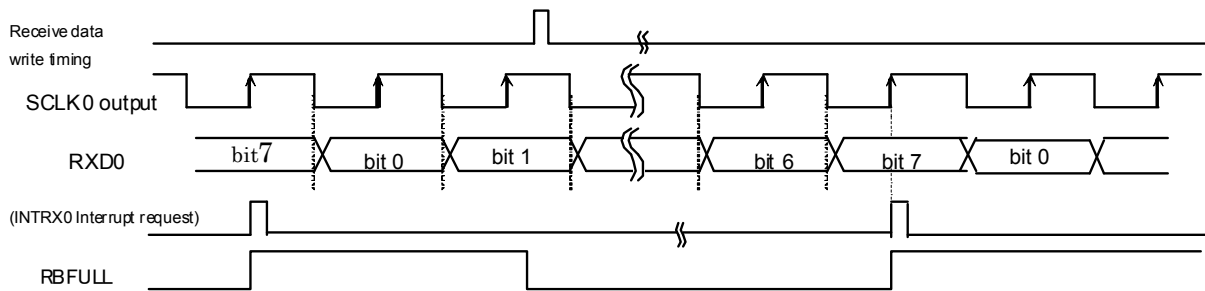
<WBUF>="1" (if double buffering is enabled and data is read from buffer 2)
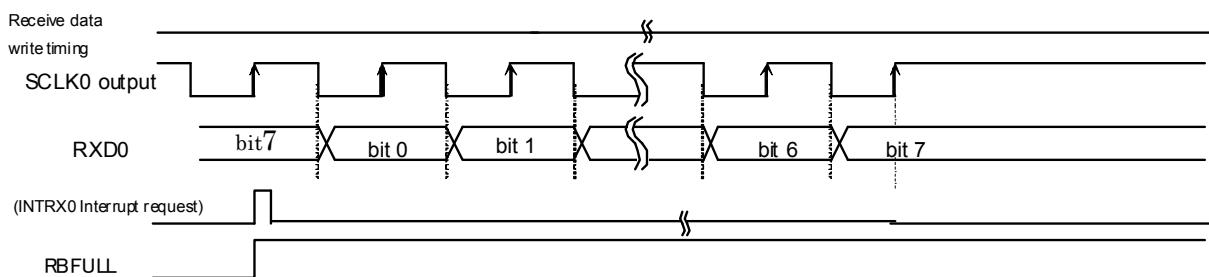
<WBUF>="1" (if double buffering is enabled and data cannot be read from buffer 2)

Fig. 3.10.12 Receive Operation in the I/O Interface Mode (SCLK0 Output Mode)

SCLK input mode

In the SCLK input mode, receiving double buffering is always enabled, the received frame can be moved to receive buffer 2 and receive buffer 1 can receive the next frame successively.

The INTRX receive interrupt is generated each time received data is moved to received buffer 2.
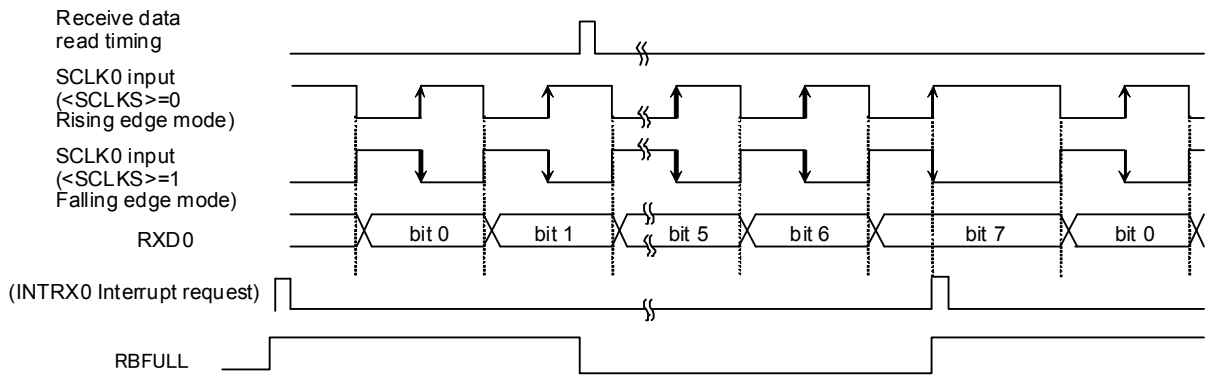
If data is read from buffer 2

If data cannot be read from buffer 2

Fig. 3.10.13 Receive Operation in the I/O Interface Mode (SCLK0 Input Mode)

> **(Note)** To receive data, SC0MOD <RXE> must always be set to "1" (receive enable) in the SCLK output / SCLK input mode.

③   Transmit and receive (full-duplex)

The full-duplex mode is enabled by setting bit 5 and 6<FDPX[1:0]> of the serial mode control register 1 (SC0MOD1) to "11".

<u>SCLK output mode</u>

In the SCLK output mode, if SC0MOD2 <WBUF> is set to "0" and both the transmit and receive double buffers are disabled, SCLK is output when the CPU writes data to the transmit buffer. Subsequently, 8 bits of data are shifted into receive buffer 1 and the INTRX0 receive interrupt is generated. Concurrently, 8 bits of data written to the transmit buffer are output from the TXD0 pin, the INTTX0 transmit interrupt is generated when transmission of all data bits has been completed. Then, the SCLK output stops. In this, the next round of data transmission and reception starts when the data is read from the receive buffer and the next transmit data is written to the transmit buffer by the CPU. The order of reading the receive buffer and writing to the transmit buffer can be freely determined. Data transmission is resumed only when both conditions are satisfied.

If SC0MOD2 <WBUF> = "1" and double buffering is enabled for both transmission and reception, SCLK is output when the CPU writes data to the transmit buffer. Subsequently, 8 bits of data are shifted into receive buffer 1, moved to receive buffer 2, and the INTRX0 interrupt is generated. While 8 bits of data is received, 8 bits of transmit data is output from the TXD0 pin. When all data bits are sent out, the INTTX0 interrupt is generated and the next data is moved from the Transmit Buffer 2 to Transmit Buffer 1. If Transmit Buffer 2 has no data to be moved to Transmit Buffer 1 (SC0MOD2 <TBEMP> = 1) or when receive buffer 2 is full (SC0MOD2 <RBFULL> = 1), the SCLK clock is stopped. When both conditions, receive data is read and transmit data is written, are satisfied, the SCLK output is resumed and the next round of data transmission is started.
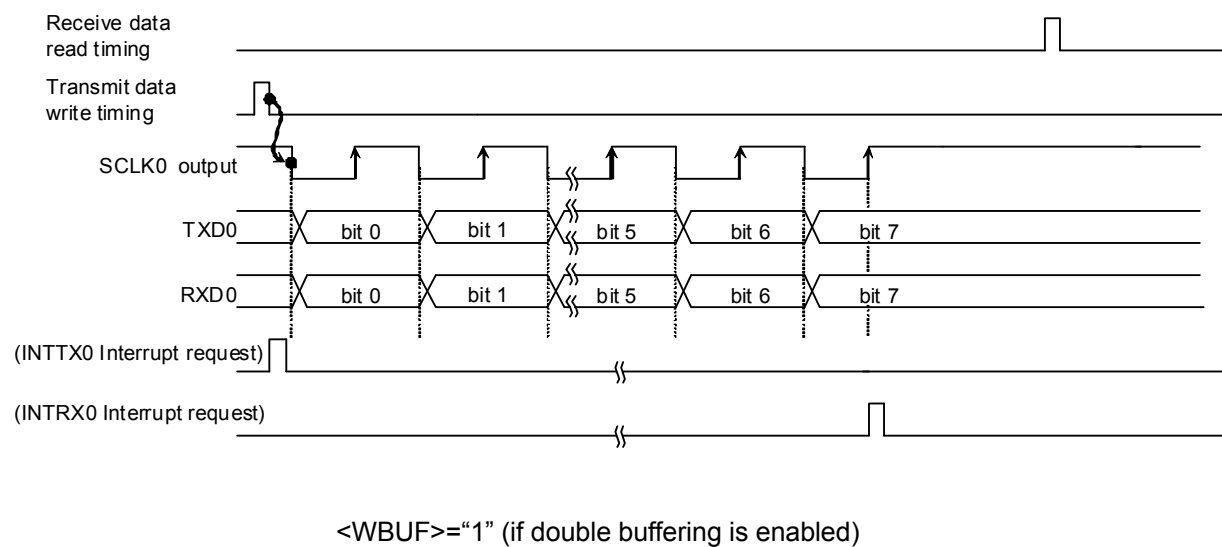
<WBUF>="0" (if double buffering is disabled)

<WBUF>="1" (if double buffering is enabled)
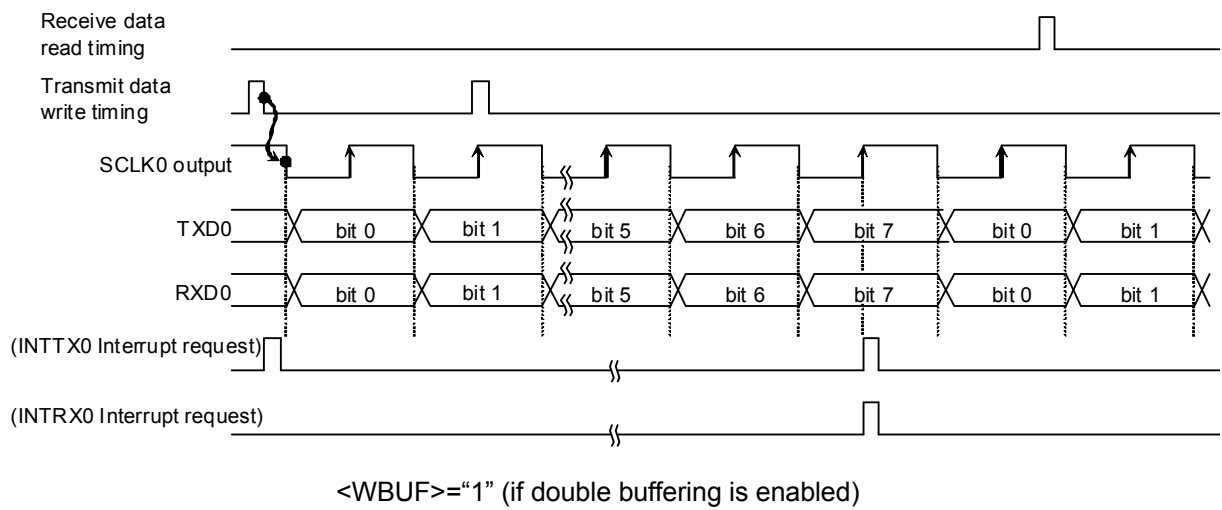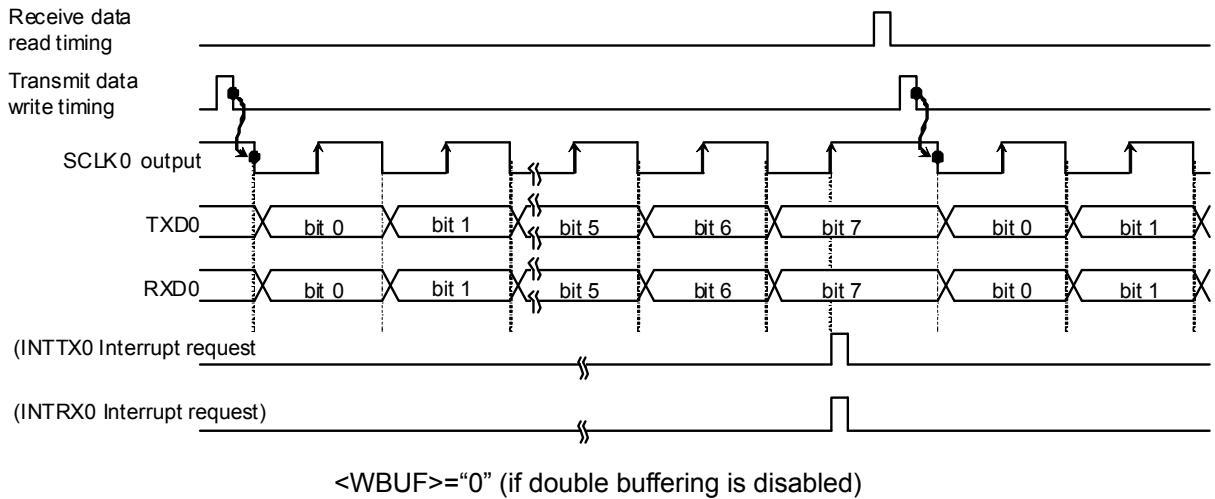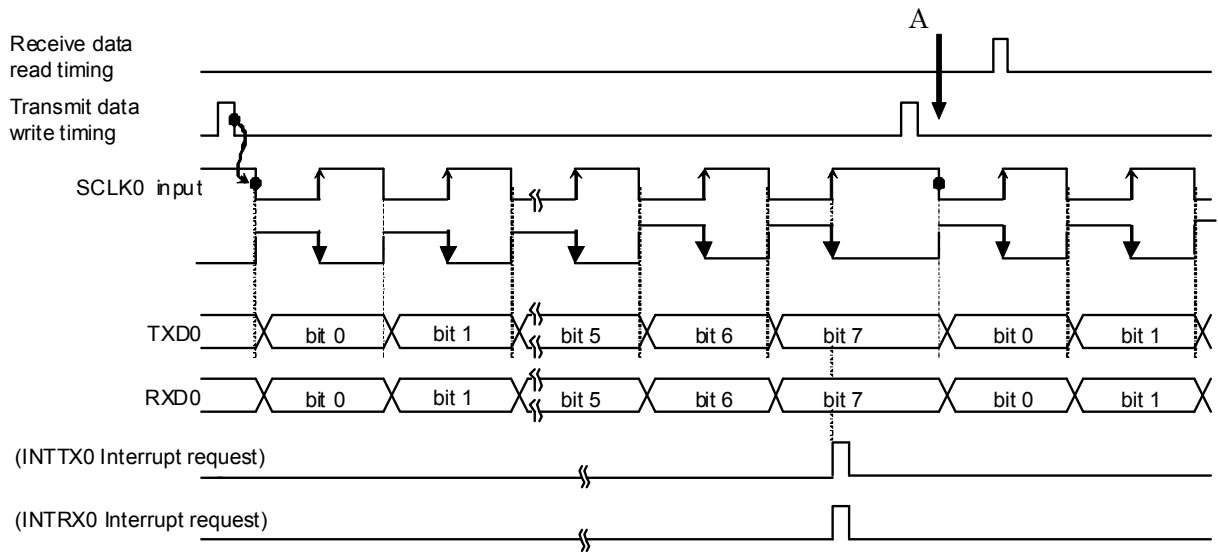

<WBUF>="1" (if double buffering is enabled)

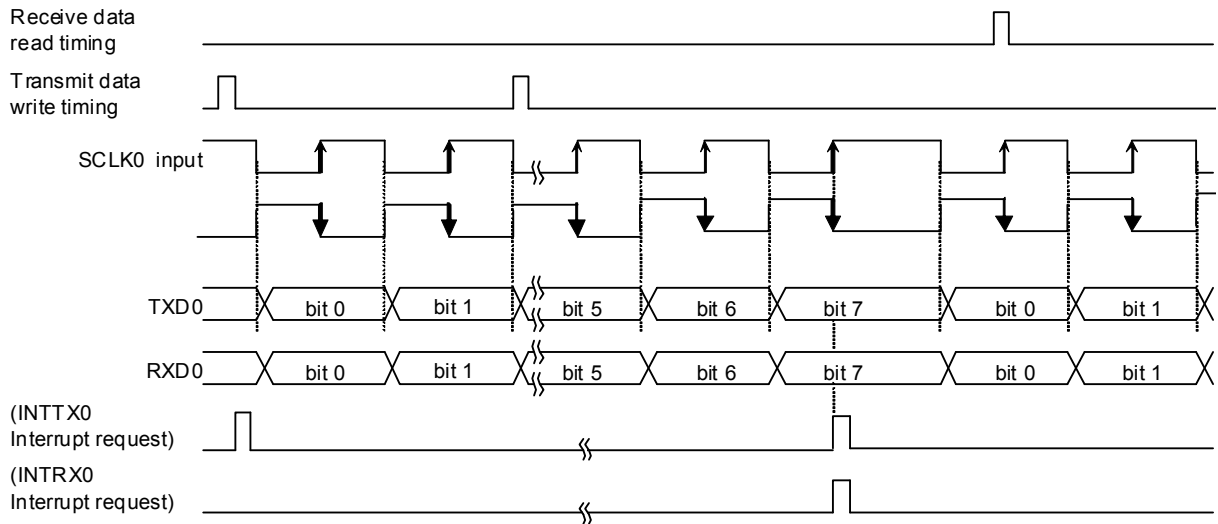Fig. 3.10.14 Transmit/Receive Operation in the I/O Interface Mode (SCLK0 Output Mode)

SCLK input mode

In the SCLK input mode with SC0MOD2 <WBUF> set to "0" and the transmit double buffers are disabled (double buffering is always enabled for the receive side), 8-bit data written in the transmit buffer is output from the TXD0 pin and 8 bits of data is shifted into the receive buffer when the SCLK input becomes active. The INTTX0 interrupt is generated upon completion of data transmission and the INTRX0 interrupt is generated at the instant the received data is moved from receive buffer 1 to receive buffer 2. Note that transmit data must be written into the transmit buffer before the SCLK input for the next frame (data must be written before the point A in Fig 3.10.15). As double buffering is enabled for data reception, data must be read before completing reception of the next frame data.

If SC0MOD2 <WBUF> = "1" and double buffering is enabled for both transmission and reception, the interrupt INTRX0 is generated at the timing Transmit Buffer 2 data is moved to Transmit Buffer 1 after completing data transmission from Transmit Buffer 1. At the same time, the 8 bits of data received is shifted to buffer 1, it is moved to receive buffer 2, and the INTRX0 interrupt is generated. Upon the SCLK input for the next frame, transmission from Transmit Buffer 1 (in which data has been moved from Transmit Buffer 2) is started while receive data is shifted into receive buffer 1 simultaneously. If data in receive buffer 2 has not been read when the last bit of the frame is received, an overrun error occurs. Similarly, if there is no data written to Transmit Buffer 2 when SCLK for the next frame is input, an under-run error occurs.

<WBUF>="0" (if double buffering is disabled)

<WBUF>="1" (if double buffering is enabled with no errors)
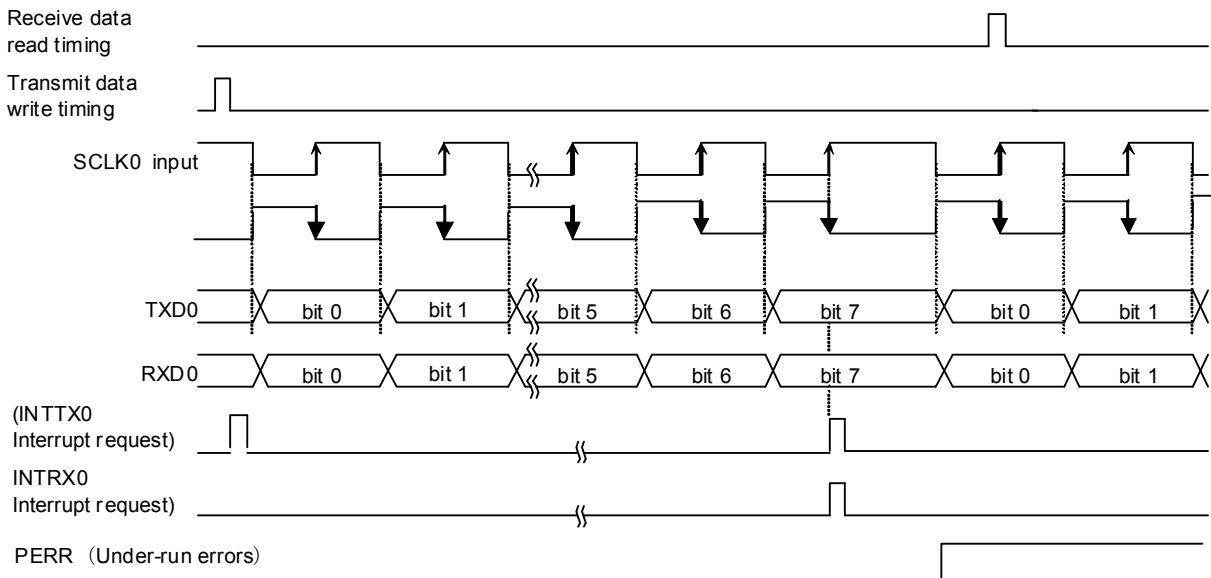
<WBUF>="1" (if double buffering is enabled with error generation)

Fig. 3.10.15 Transmit/Receive Operation in the I/O Interface Mode (SCLK0 Input Mode)

3.10.5.2 Mode 1 (7-bit UART Mode)

The 7-bit UART mode can be selected by setting the serial mode control register (SC0MOD <SM[1:0]>) to "01".

In this mode, parity bits can be added to the transmit data stream; the serial mode control register (SC0CR <PE>) controls the parity enable/disable setting. When <PE> is set to "1" (enable), either even or odd parity may be selected using the SC0CR <EVEN> bit. The length of the stop bit can be specified using SC0MOD2<SBLEN>.

The following table shows the control register settings for transmitting in the following data format.
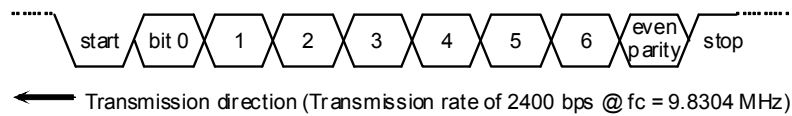
Transmission direction (Transmission rate of 2400 bps @ fc = 9.8304 MHz)

* Clocking conditions

System clock : high- speed (fc)

High-speed clock gear : x1 (fc)

Prescaler clock : $f_{periph/2}$ ($f_{periph} = f_{sys}$)

3.10.5.3   Mode 2 (8-bit UART Mode)

The 8-bit UART mode can be selected by setting SC0MOD0 <SM[1:0]> to "10." In this mode, parity bits can be added and parity enable/disable is controlled using SC0CR <PE>. If <PE> = "1" (enabled), either even or odd parity can be selected using SC0CR <EVEN>.

The control register settings for receiving data in the following format are as follows:

Transmission direction (transmission rate of 9600 bps @ fc = 9.8304 MHz)

| * Clocking conditions	System clock	:	High-speed (fc)
	High-speed clock gear	:	x1 (fc)
	Prescaler clock	:	$f_{periph}/4$ ($f_{periph} = f_{sys}$)

3.10.5.4   Mode 3 (9-bit UART Mode)

The 9-bit UART mode can be selected by setting SC0MOD0 <SM[1:0]> to "11." In this mode, parity bits must be disabled (SC0CR <PE> = "0").

The most significant bit (9th bit) is written to bit 7 <TB8> of the serial mode control register 0 (SC0MOD0) for transmitting data. The data is stored in bit 7 <RB8> of the serial control register SC0CR. When writing or reading data to/from the buffers, the most significant bit must be written or read first before writing or reading to/from SC0BUF. The stop bit length can be specified using SC0MOD2 <SBLEN>.

Wakeup function

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting the wake-up function control bit SC0MOD0 <WU> to "1." In this case, the interrupt INTRX0 will be generated only when SC0CR <RB8> is set to "1".
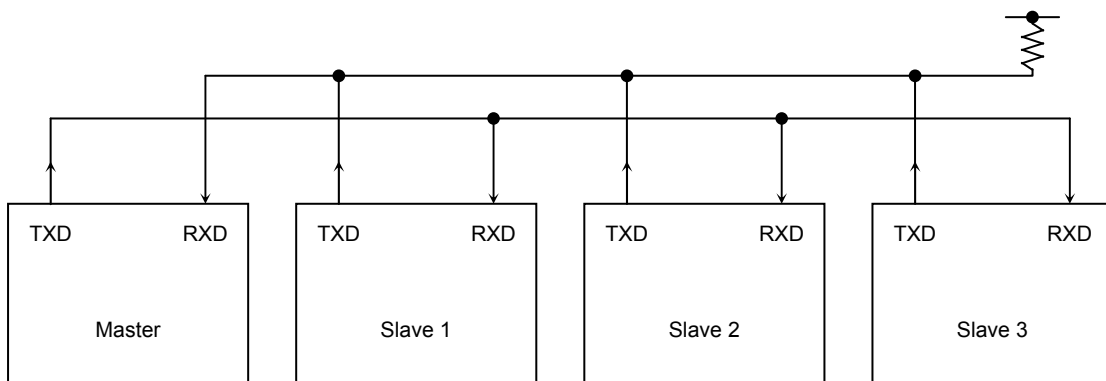

| (Note)The TXD pin of the slave controller must be set to the open drain output mode using the ODE register. |

Fig. 3.10.16 Serial Links to Use Wake-up Function

Protocol

① Select the 9-bit UART mode for the master and slave controllers.

② Set SC0MOD <WU> to "1" for the slave controllers to make them ready to receive data.

③ The master controller is to transmit a single frame of data that includes the slave controller select code (8 bits). In this, the most significant bit (bit 8) <TB8> must be set to "1".
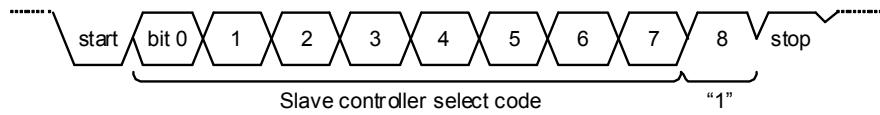

④ Each slave controller receives the above data frame; if the code received matches with the controller's own select code, it clears the WU bit to "0".

⑤ The master controller transmits data to the designated slave controller (the controller of which SC0MOD <WU> bit is cleared to "0"). In this, the most significant bit (bit 8) <TB8> must be set to "0".
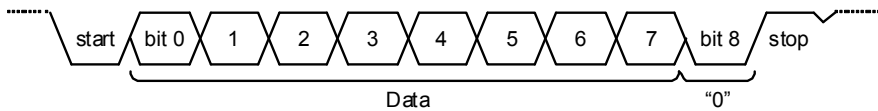
⑥ The slave controllers with the <WU> bit set to "1" ignore the receive data because the most significant bit (bit 8) <RB8> is set to "0" and thus no interrupt (INTRX0) is generated.

Also, the slave controller with the <WU> bit set to "0" can transmit data to the master controller to inform that the data has been successfully received.

An example: Using the internal clock f_{SYS} as the transfer clock, two slave controllers are serially linked as follows.
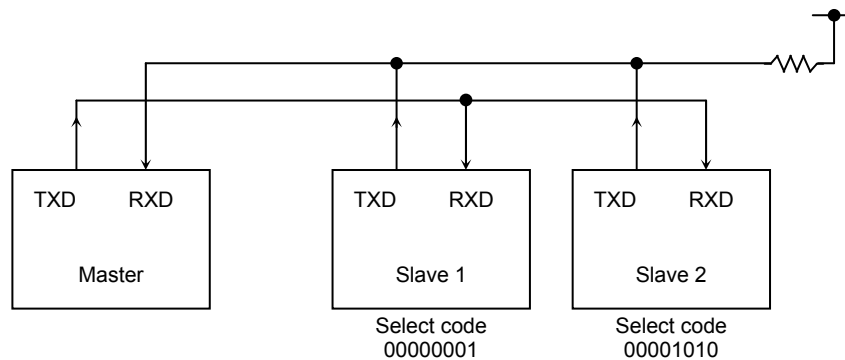
3.11 SSP (Synchronous Serial Port)

3.11.1 Overview

This LSI contains a one-channel synchronous serial port (SSP).

Key features of the SSP are summarized below.

|	Channel 0
Communication protocol	3 types of synchronous serial communication protocols, including SPI - Motorora SPI (SPI) frame format - TI synchronous (SSI) frame format - National Microwire (Microwire) frame format
Operation mode	Master mode, slave mode
Transmit FIFO	16-bit wide, 8 locations deep
Receive FIFO	16-bit wide, 8 locations deep
Transmit/receive data size	4 to 16 bits
Interrupt type	Transmit interrupt Receive interrupt Receive overrun interrupt Timeout interrupt
Baud rate	Master mode: f_{sys} (48 MHz) / 4 (up to 12 Mbps)
	Slave mode: f_{sys} (48 MHz) /12 (up to 4 Mbps) (Note) For slave mode, set the clock prescale as follows: SSPCR0<SCR>=0x02, SSPCPSR=0x04
DMA	Supported
On-chip test function	Internal loopback test mode available
Control pins	Channel 0
	SPCLK SPFSS SPDO SPDI

3.11.2 Block Diagram
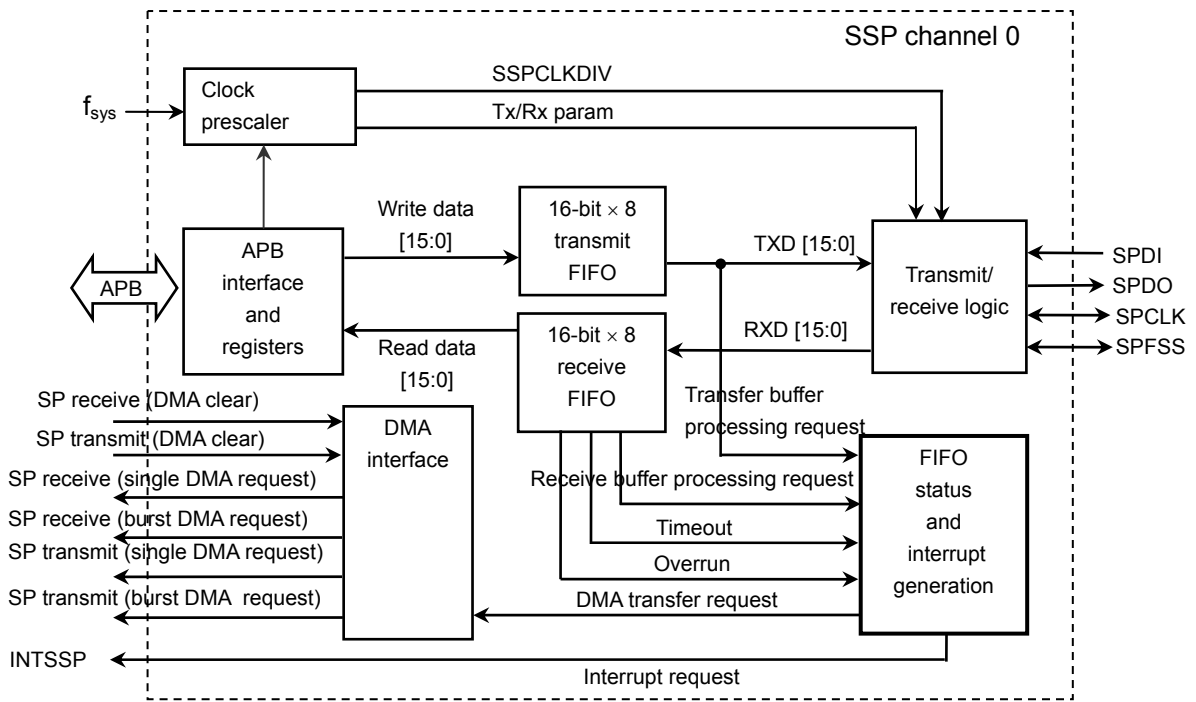
Figure 3.11.1 Block Diagram of SSP

3.11.3   SSP Overview

The SSP is an interface for serial communication with peripheral devices that have three types of synchronous serial interfaces.

The SSP performs serial-to-parallel conversion on data received from a peripheral device. The transmit and receive paths are buffered with a 16-bit wide, 8 locations deep independent transmit and receive FIFOs in transmit and receive modes. Serial data is transmitted on SPDO and received on SPDI.

The SSP contains a programmable prescaler to generate the serial output clock SPCLK from the input clock f_{sys}. The SSP operating mode, frame format and data size are programmed through the control registers SSPCR0 and SSPCR1.

3.11.3.1   Clock prescaler

When configured as a master, a clock prescaler comprising two serially linked free-running counters is used to provide the serial output clock SPCLK.

This clock prescaler can be programmed, thorugh the SSPCPSR register, to divide f_{sys} by a factor of 2 to 254 in steps of two. By not using the least significant bit of the SSPCPSR register, division by an odd number cannot be programmed.

The output of the prescaler is further divided by a factor of 1 to 256, obtained by adding one to the value programmed in the SSPCR0 control register, to give the master output clock SPCLK.

$$\text{Bit rate} = f_{sys} / (\text{CPSDVSR} \times (1+\text{SCR}))$$
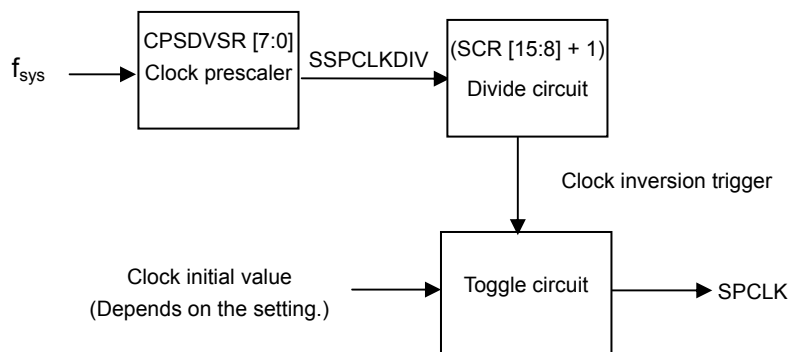
3.11.3.2   Transmit FIFO

The transmit FIFO buffer is 16-bit wide, 8 locations deep and is shared between master and slave modes.

3.11.3.3   Receive FIFO

The receive FIFO buffer is 16-bit wide, 8 locations deep and is shared between master and slave modes.

3.11.3.4 Interrupt generation logic

Four individual maskable active HIGH interrupts are generated by the SSP. A combined interrupt output is also generated as an OR function of the individual interrupt requests.

- Transmit interrupt: Indicates that TxFIFO is more than half empty.
  (Number of valid entries in TxFIFO ≤ 4)
- Receive interrupt: Indicates that RxFIFO is more than half full.
  (Number of valid entries in RxFIFO ≥4)
- Timeout interrupt: Indicates that data is present in RxFIFO and has not been read before a timeout period expires.
- Receive overrun interrupt: Indicates that the data is written to RxFIFO when it is full.

INTSSP0 is asserted if any of the above four interrupts is asserted.

(a) Transmit interrupt

The transmit interrupt is asserted when there are four or less valid entries in the transmit FIFO. The transmit interrupt is generated even when SSP operation is disabled (SSPCR1<SSE>=0).

The initial transmit data can be written into the transmit FIFO by using this interrupt.

(b) Receive interrupt

The receive interrupt is asserted when there are four or more valid entries in the receive FIFO.

(c) Timeout interrupt

The receive timeout interrupt is asserted when the receive FIFO is not empty and the SSP has remained idle for a fixed duration of 32-bit period (bit rate). This mechanism ensures that the user is aware that data is still present in the receive FIFO and requires servicing. The timeout interrupt is generated both in master and slave modes. When the timeout interrupt is generated, read all the data in the receive FIFO. Data can be transmitted/received without reading all the data in the receive FIFO provided that the receive FIFO has empty space for receiving the data to be transmitted. The timeout interrupt is cleared when a transfer is started. If a transfer is performed when the receive FIFO is full, the timeout interrupt is cleared and the overrun interrupt is generated.
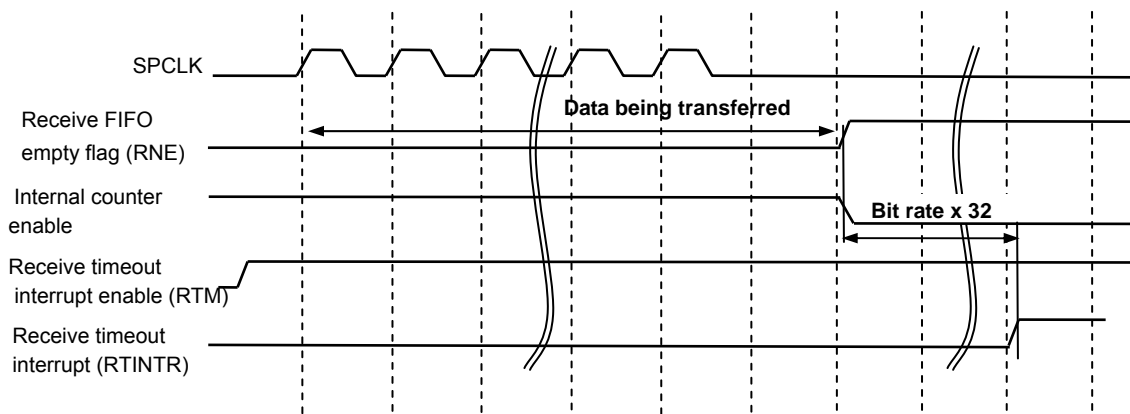
(d) Receive overrun interrupt

When the receive FIFO is already full and an additional (9th) data frame is received, the receive overrun interrupt is asserted immediately after the completion of the current transfer. Once the receive overrun error occurs, any subsequent data received (including the 9th data frame) is invalid and discarded. However, if the data in the receive FIFO is read while the 9th data frame is being received (before the receive overrun interrupt occurs), the 9th data frame is written into the receive FIFO as valid data. To perform proper transfer operation after the receive overrun error occurred, write a 1 to the receive overrun interrupt clear register and then read all the data in the receive FIFO. Data can be transmitted/received without reading all the data in the receive FIFO provided that the receive FIFO has empty space for receiving the data to be transmitted. If the receive FIFO is not read (if it is not empty) for a fixed duration of 32-bit period (bit rate) after the receive overrun interrupt has been cleared, the timeout interrupt is generated.

(e) Combined interrupt

The individual masked sources of the above four interrupts are also combined into a single interrupt. The combined interrupt INTSSP is asserted if any of the four interrupts is asserted.
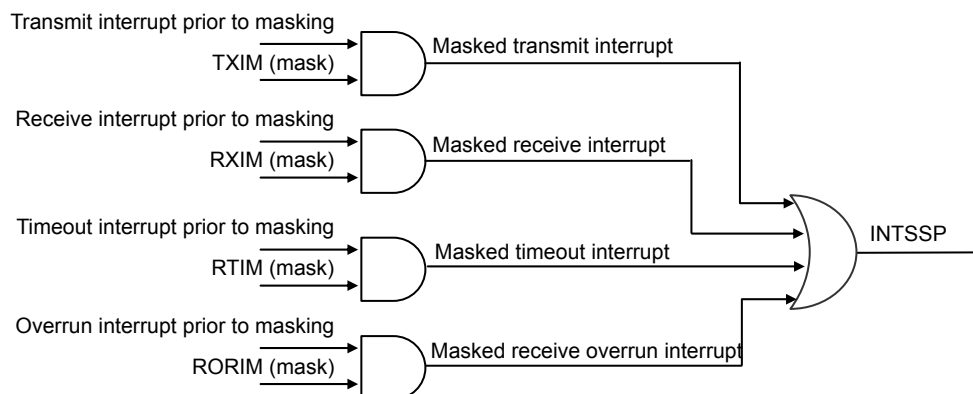
3.11.3.5 DMA Interface

The SSP provides an interface to connect to a DMA controller.

3.11.4   SSP Operation

3.11.4.1   Configuring the SSP

The SSP communication protocol must be configured while the SSP is disabled.

Select the frame format to be used in the control register SSPCR0 and select master or slave mode in the control register SSPCR1. The communication rate need also be set by programming the prescale register SSPCPSR and SSPCR0<SCR>.

This SSP supports the following frame formats:

- SPI
- SSI
- Microwire

3.11.4.2   Enabling the SSP

Transmission of data begins when SSP operation is enabled after transmit data has been written into the transmit FIFO or when transmit data is written into the transmit FIFO after SSP operation has been enabled.

However, if the transmit FIFO has four entries or less when SSP operation is enabled, the transmit interrupt will be generated. It is possible to use this interrupt to write the initial transmit data.

Note :   When using the SPI slave mode without using the FFS pin, be sure to write 1 byte or more of data into the transmit FIFO before enabling SSP operation. If SSP operation is enabled while the transmit FIFO is empty, transfer data cannot be output properly.

3.11.4.3   Clock Ratios

The f_{sys} frequency setting must satisfy the following conditions:

[Master mode]

f_{SP0CLK} (max) = f_{sys} / 4

f_{SP0CLK} (min) = f_{sys} / (254 x 256)

[Slave mode]

f_{SP0CLK} (max) = f_{sys} / 12

f_{SP0CLK} (min) = f_{sys} / (254 x 256)

3.11.4.4   Frame Format

Each frame format is between 4 to 16 bits long depending on the size of data programmed, and is transmitted starting with the MSB.

• Serial clock (SPCLK)

For SSI and Microwire frame formats, the serial clock (SPCLK) is held LOW while the SSP is idle. For SPI frame format, the serial clock (SPCLK) is held inactive while the SSP is idle. SPCLK is output at the specified bit rate only while data is being transmitted.

• Serial frame (SPFSS)

For SPI and Microwire frame formats, the serial frame (SPFSS) pin is active LOW, and is asserted during the entire transmission of the frame.

For SSI frame format, the SPFSS pin is asserted for one bit rate period prior to the transmission of each frame. For this frame format, output data is transmitted on the rising edge of SPCLK, and input data is received on the falling edge.

3.11.4.5 SSI frame format

In this mode, SPCLK and SPFSS are forced LOW and the transmit data line SPDO is put in the Hi-Z state whenever the SSP is idle. When data is written into the transmit FIFO, the master pulses the SPFSS line HIGH for one SPCLK period. The transmit data is transferred from the transmit FIFO to the transmit serial shift register. On the next rising edge of SPCLK, the MSB of the 4 to 16-bit data frame is shifted out on the SPDO pin.

Likewise, the received data is shifted onto the SPDI pin from the MSB on the falling edge of SPCLK. The received data is transferred from the serial shift register to the receive FIFO on the first rising edge of SPCLK after the LSB has been latched.
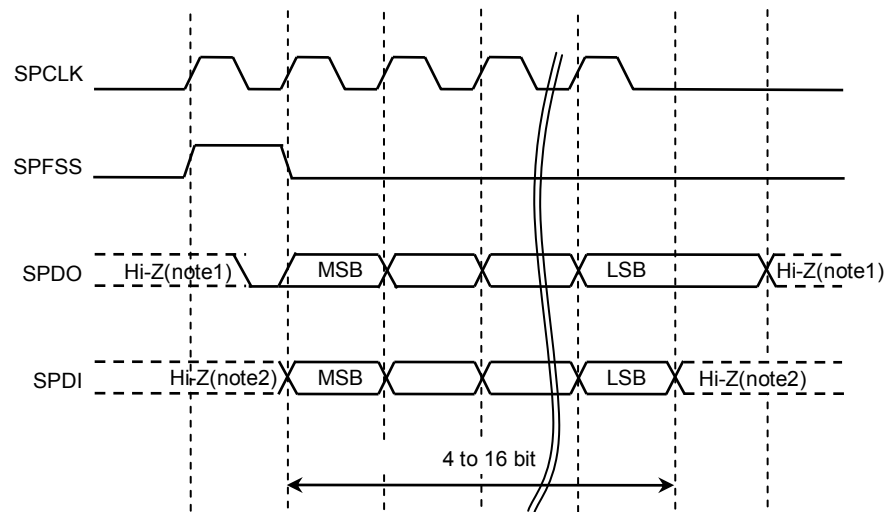
(Note1) Output from SPDO pin is stopped when the transmission stops and becomes Hi-Z. Therefore, fix the level with a pull-up or a pull-down register according to the system you use.

(Note2) Since SPDI pin is a continuous input, fix the level with a pull-up or a pull-down register in case that the SPDI pin stops output when the transmission stops.

Figure 3.11.2   SSI Frame Format (Single Transfer – Transmit/Receive)
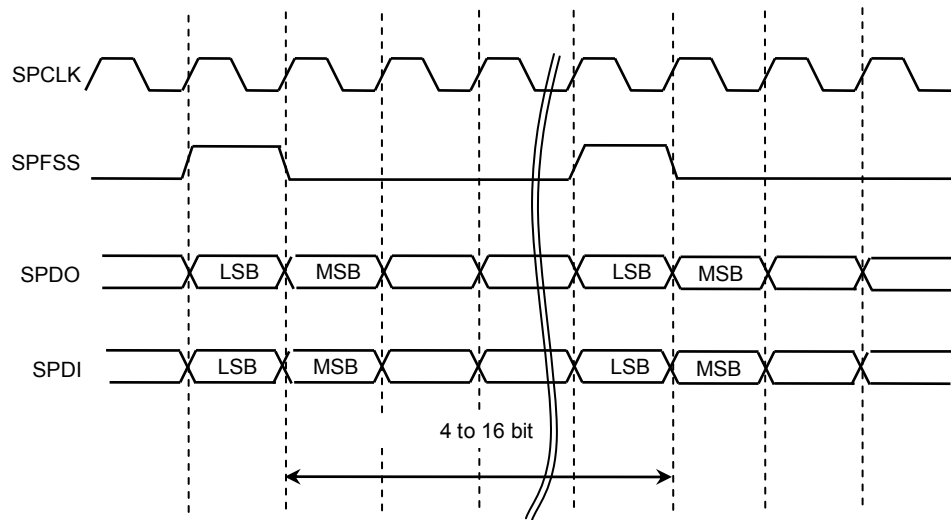
(Note1) Output from SPDO pin is stopped when the transmission stops and becomes Hi-Z. Therefore, fix the level with a pull-up or a pull-down register according to the system you use.

(Note2) Since SPDI pin is a continuous input, fix the level with a pull-up or a pull-down register in case that the SPDI pin stops output when the transmission stops.

Figure 3.11.3   SSI Frame Format (Continuous Transfer – Transmit/Receive)

3.11.4.6   SPI frame format

The SPI interface is a four-wire interface where the SPFSS signal behaves as a slave select. The main feature of the SPI format is that the operation timing of SPCLK is programmable through the <SPO> and <SPH> bits in the SSPCR0 control register.

SSPCR0<SPO>

SSPCR0<SPO> specifies the SPCLK level during idle periods.

<SPO>=1: SPCLK is held HIGH.

<SPO>=0: SPCLK is held LOW.

SSPCR0<SPH>

SSPCR0<SPH> selects the clock edge for latching data.

SSPCR0<SPH>=0: Data is latched on the first clock edge.

SSPCR0<SPH>=1: Data is latched on the second clock edge.
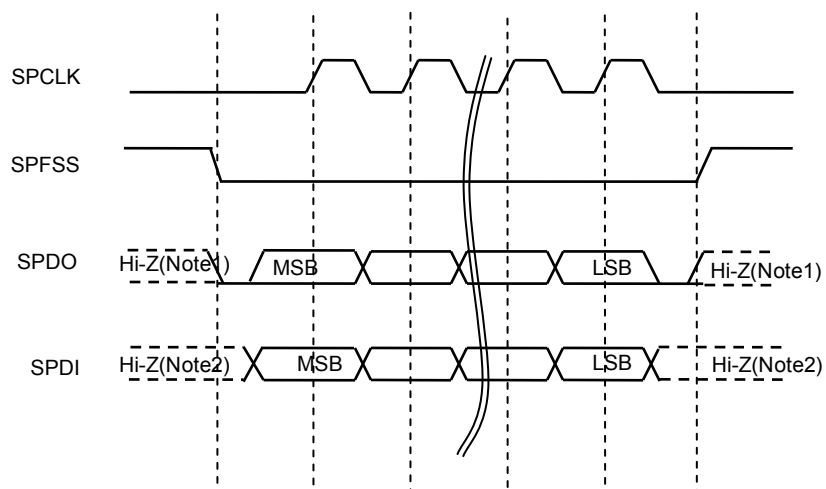
(Note1) Output from SPDO pin is stopped when the transmission stops and becomes Hi-Z. Therefore, fix the level with a pull-up or a pull-down register according to the system you use.

(Note2) Since SPDI pin is a continuous input, fix the level with a pull-up or a pull-down register in case that the SPDI pin stops output when the transmission stops.

Figure 3.11.4   SPI Frame Format (Single Transfer with <SPO>=0 and <SPH>=0)
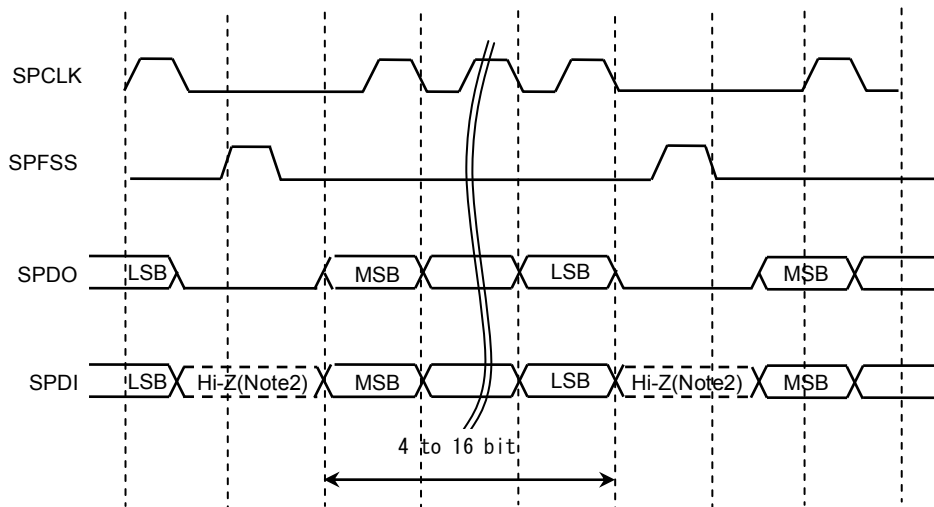
(Note2) Since SPDI pin is a continuous input, fix the level with a pull-up or a pull-down register in case that the SPDI pin stops output when the transmission stops.

Figure 3.11.5   SPI Frame Format (Continuous Transfer with <SPO>=0 and <SPH>=0)

In this configuration, during idle periods:

• the SPCLK signal is forced LOW.

• SPFSS is forced HIGH.

• the transmit data line SPDO is arbitrarily forced LOW.

If the SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SPFSS master signal being driven LOW. This causes slave data to be enabled onto the SPDI input line of the master.

One half SPCLK period later, valid master data is transferred to the SPDO pin. Now that both the master and slave data have been set, the SPCLK master clock pin goes HIGH after one further half SPCLK period. The data is now captured on the rising and propagated on the falling edges of the SPCLK signal. In the case of a single word transmission, after all bits of the data word have been transferred, the SPFSS line is returned to its idle HIGH state one SPCLK period after the last bit has been captured. However, in the case of continuous back-to-back transmission, the SPFSS signal must be pulsed HIGH between each data word transfer. This is because the slave select pin freezes the data in its serial peripheral register and does not allow it to be altered if the <SPH> bit is logic zero. Therefore the master device must raise the SPFSS pin of the slave device between each data transfer to enable the serial peripheral data write. On completion of the continuous transfer, the SPFSS pin is returned to its idle state one SPCLK period after the last bit has been captured.

3.11.4.7   Microwire frame format

• Microwire frame format

The Microwire format uses a special master-slave messaging technique, which operates at half-duplex. In this mode, when a frame begins, an 8-bit control message is transmitted to the slave. During this transmission, no incoming data is received by the SSP. After the message has been sent, the slave decodes it and, after waiting one serial clock period after the last bit of the 8-bit control message has been sent, responds with the requested data. The returned data can be 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits.

The details of each frame format are described below.
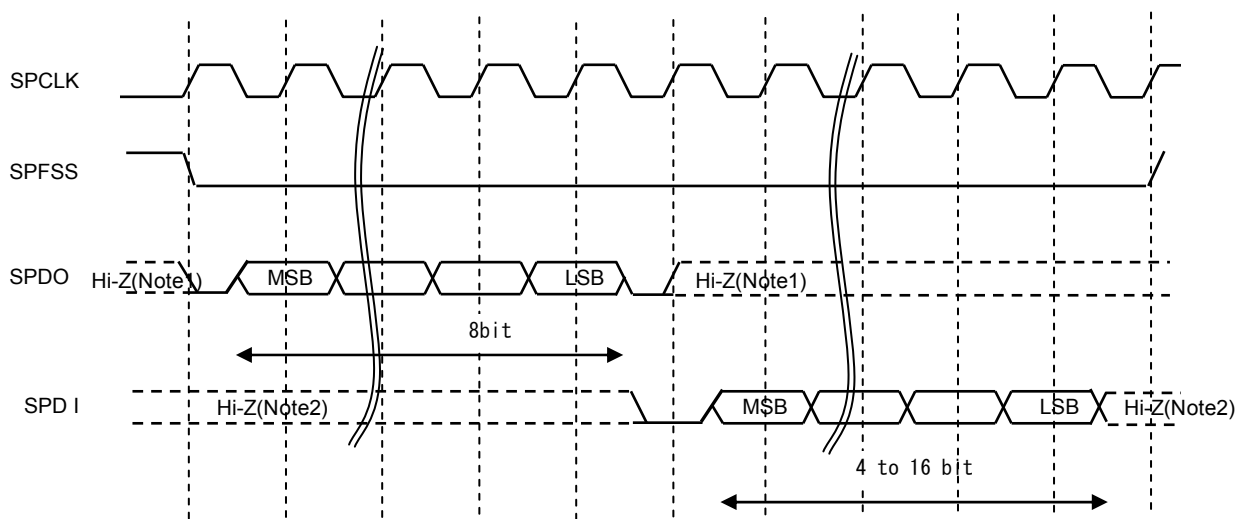
(Note1) Output from SPDO pin is stopped when the transmission stops and becomes Hi-Z. Therefore, fix the level with a pull-up or a pull-down register according to the system you use.

(Note2) Since SPDI pin is a continuous input, fix the level with a pull-up or a pull-down register in case that the SPDI pin stops output when the transmission stops.

Figure 3.11.6   Microwire Frame Format (Single Transfer)

Microwire format is very similar to SPI format, except that transmission is half-duplex instead of full-duplex, using a master-slave message passing technique. Each serial transmission begins with an 8-bit control word that is transmitted from the SSP to the off-chip slave device. During this transmission, no incoming data is received by the SSP. After the message has been sent, the off-chip slave decodes it and, after waiting one serial clock after the last bit of the 8-bit control message has been sent, responds with the required data. The returned data is 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits.

In this configuration, during idle periods:

• the SPCLK signal is forced LOW

• SPFSS is forced HIGH

• the transmit data line SPDO is arbitrarily forced LOW.

A transmission is triggered by writing a control byte to the transmit FIFO. The falling edge

of SPFSS causes the value contained in the bottom entry of the transmit FIFO to be transferred to the serial shift register of the transmit logic, and the MSB of the 8-bit control frame to be shifted out onto the SPDO pin. SPFSS remains LOW for the duration of the frame transmission. The SPDI pin remains tristated during this transmission. The off-chip serial slave device latches each control bit into its serial shifter on the rising edge of each SPCLK. After the last bit is latched by the slave device, the control byte is decoded during a one clock wait-state, and the slave responds by transmitting data back to the SSP. Each bit is driven onto the SPDI line on the falling edge of SPCKL. The SSP in turn latches each bit on the rising edge of SPCLK. At the end of the frame, for single transfers, the SPFSS signal is pulled HIGH one clock period after the last bit has been latched in the receive serial shifter. This causes the data to be transferred to the receive FIFO.

Note: The off-chip slave device can tristate the receive line either on the falling edge of SPCLK after the LSB has been latched by the receive shifter, or when the SPFSS pin goes HIGH.
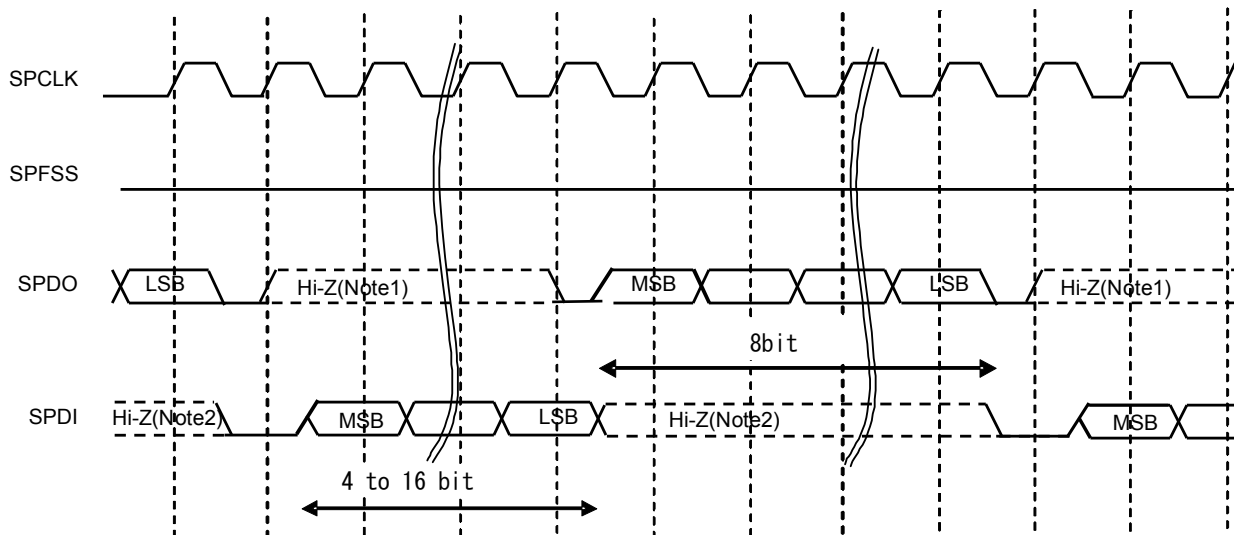
(Note1) Output from SPDO pin is stopped when the transmission stops and becomes Hi-Z. Therefore, fix the level with a pull-up or a pull-down register according to the system you use.

(Note2) Since SPDI pin is a continuous input, fix the level with a pull-up or a pull-down register in case that the SPDI pin stops output when the transmission stops.

Figure 3.11.7   Microwire Frame Format (Continuous Transfer)

For continuous transfers, data transmission begins and ends in the same manner as a single transfer. However, the SPFSS line is continuously asserted (held LOW) and transmission of data occurs back to back. The control byte of the next frame follows directly after the LSB of the received data from the current frame. Each of the received values is transferred from the receive shifter on the falling edge of SPCLK after the LSB of the frame has been latched into the SSP.

Note: The SSP does not support dynamic switching between master and slave in a system. Each instance is configured and connected either as a master or slave.

3.11.4.8   DMA Interface

The DMA operation of the SSP is controlled through the DMA control register SPDMACR.

When the amount of data contained in the receive FIFO exceeds the watermark level (1/2 of FIFO), the receive DMA request is asserted.

When the amount of data contained in the transmit FIFO is less than the wartermark level (1/2 of FIFO), the transmit DMA request is asserted.

The DMA interface has input pins for the transmit/receive DMA request clear signals, to be asserted by the DMA controller, for clearing the transmit/receive DMA request signals. The DMA burst length should be set to 4 words.

* For the remaining three characters, the SSP does not assert the burst request.

Each request signal remains asserted until the relevant DMA clear signal is asserted. After the request clear signal is deasserted, a request signal can become active again, depending on the conditions described above. All request signals are deasserted if the SSP is disabled or the DMA enable signal is cleared.

The following table shows the trigger points for DMABREQ, for both the transmit and receive FIFOs.

	Burst length	
Watermark level	Transmit (number of empty locations)	Receive (number of filled locations)
1/2	4	4

3.11.5   Description of Registers

The SSP has the following registers.

3.11.5.1   SSP Registers

Base Address= 0x4004_0000

Register Name	Address (base+)	Description
SSPCR0	0x0000	Control register 0
SSPCR1	0x0004	Control register 1
SSPDR	0x0008	SSP data register
SSPSR	0x000C	Status register
SSPCPSR	0x0010	Clock prescale register
SSPIMSC	0x0014	Interrupt mask set and clear register
SSPRIS	0x0018	Raw interrupt status register
SSPMIS	0x001C	Masked interrupt status register
SSPICR	0x0020	Interrupt clear register
SSPDMACR	0x0024	DMA control register
−	0x0028 - 0xFFC	Reserved

(Note) The above registers only allow word (32-bit) accesses.

3.11.5.2   SSPCR0 (SSP control register 0)

Address = (0x4004_0000) + 0x0000

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read undefined. Write as zero.
[15:8]	SCR	R/W	0y0	Parameter for setting the serial clock rate: 0x00 to 0xFF (See the description below.)
[7]	SPH	R/W	0y0	SPCLK phase (Applicable to Motorola SPI frame format only. See the description of "SPI frame format".)
[6]	SPO	R/W	0y0	SPCLK polarity (Applicable to Motorola SPI frame format only. See the description of "SPI frame format".)
[5:4]	FRF	R/W	0y00	Frame format 0y00: Motorola SPI frame format 0y01: TI synchronous serial frame format 0y10: National Microwire frame format 0y11: Reserved, undefined operation
[3:0]	DSS	R/W	0y0000	Data size select 0y0000: Reserved, undefined operation 0y0001: Reserved, undefined operation 0y0010: Reserved, undefined operation 0y0011: 4-bit data 0y0100: 5-bit data 0y0101: 6-bit data 0y0110: 7-bit data 0y0111: 8-bit data 0y1000: 9 -bit data 0y1001: 10-bit data 0y1010: 11-bit data 0y1011: 12-bit data 0y1100: 13-bit data 0y1101: 14-bit data 0y1110: 15-bit data 0y1111: 16-bit data

<SCR>: The <SCR> value is used to generate the transmit and receive bit rate of the SSP.

The bit rate is:

$f_{sys} / (CPSDVSR \times (1+SCR))$

where CPSDVSR is an even value from 2 to 254, programmed through the SSPCPSR register and SCR is a value from 0 to 255.

3.11.5.3   SSPCR1 (SSP control register 1)

Address = (0x4004_0000) + 0x0004

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read undefined. Write as zero.
[3]	SOD	R/W	0y0	Slave mode SPDO output control 0y0: Enable 0y1: Disable
[2]	MS	R/W	0y0	Master or slave mode select 0y0: Device configured as a master 0y1: Device configured as a slave
[1]	SSE	R/W	0y0	SSP0 enable 0y0: Disable 0y1: Enable
[0]	LBM	R/W	0y0	Loop back mode 0y0: Normal serial port operation enabled 0y1: Output of transmit serial shifter is connected to input of receive serial shifter internally.

<SOD>: Slave mode output disable. This bit is relevant only in the slave mode (<MS>=1).

3.11.5.4   SSPDR (SSP data register)

Address = (0x4004_0000) + 0x0008

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read undefined. Write as zero.
[15:0]	DATA	R/W	0x0000	Transmit/receive FIFO data: 0x00 - 0xFF

<DATA>:

Read: Receive FIFO

Write: Transmit FIFO

You must right-justify data when the SSP is programmed for a data size that is less than 16 bits. Unused bits at the top are ignored by the transmit logic. The receive logic automatically right-justifies.

3.11.5.5   SSPSR (SSP status register)

Address = (0x4004_0000) + 0x000C

Bit	Bit Symbol	Type	Reset Value	Description
[31:5]	–	–	Undefined	Read undefined. Write as zero.
[4]	BSY	RO	0y0	Busy flag 0y0: SSP is idle. 0y1: SSP is busy.
[3]	RFF	RO	0y0	Receive FIFO full flag 0y0: Receive FIFO is not full. 0y1: Receive FIFO is full.
[2]	RNE	RO	0y0	Receive FIFO empty flag 0y0: Receive FIFO is empty. 0y1: Receive FIFO is not empty.
[1]	TNF	RO	0y1	Transmit FIFO full flag 0y0: Transmit FIFO is full. 0y1: Transmit FIFO is not full.
[0]	TFE	RO	0y1	Transmit FIFO empty flag: 0y0: Transmit FIFO is not empty. 0y1: Transmit FIFO is empty.

<BSY>: <BSY>="1" indicates that the SSP is currently transmitting and/or receiving a frame or the transmit FIFO is not empty.

3.11.5.6   SSPCPSR (SSP clock prescale register)

Address = (0x4004_0000) + 0x0010

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read undefined. Write as zero.
[7:0]	CPSDVSR	R/W	0x0000	Clock prescale divisor: Must be an even number from 2 to 254.

<CPSDVSR>:   Clock prescale divisor. Must be an even number from 2 to 254, depending on the frequency of f_{sys}. The least significant bit always returns zero on reads.

3.11.5.7 SSPIMSC (SSP interrupt mask set or clear register)

Address = (0x4004_0000) + 0x0014

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read undefined. Write as zero.
[3]	TXIM	R/W	0y0	Transmit FIFO interrupt enable 0y0: Disable 0y1: Enable
[2]	RXIM	R/W	0y0	Receive FIFO interrupt enable 0y0: Disable 0y1: Enable
[1]	RTIM	R/W	0y0	Receive timeout interrupt enable 0y0: Disable 0y1: Enable
[0]	RORIM	R/W	0y0	Receive overrun interrupt enable 0y0: Disable 0y1: Enable

<TXIM>: Enables or disables the transmit interrupt.

<RXIM>: Enables or disables the receive interrupt.

<RTIM>: Enables or disables the timeout interrupt.

<RORIM>: Enables or disables the receive overrun interrupt.

3.11.5.8   SSPRIS (SSP raw interrupt status register)

Address = (0x4004_0000) + 0x0018

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read undefined. Write as zero.
[3]	TXRIS	RO	0y1	Transmit interrupt status prior to enable gate 0y0: No interrupt 0y1: Interrupt requested
[2]	RXRIS	RO	0y0	Receive interrupt status prior to enable gate 0y0: No interrupt 0y1: Interrupt requested
[1]	RTRIS	RO	0y0	Receive timeout interrupt status prior to enable gate 0y0: No interrupt 0y1: Interrupt requested
[0]	RORRIS	RO	0y0	Receive overrun interrupt status prior to enable gate 0y0: No interrupt 0y1: Interrupt requested

3.11.5.9   SSPMIS (SSP masked interrupt status register)

Address = (0x4004_0000) + 0x001C

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read undefined. Write as zero.
[3]	TXMIS	RO	0y0	Transmit interrupt status after enable gate 0y0: No interrupt 0y1: Interrupt requested
[2]	RXMIS	RO	0y0	Receive interrupt status after enable gate 0y0: No interrupt 0y1: Interrupt requested
[1]	RTMIS	RO	0y0	Receive timeout interrupt status after enable gate 0y0: No interrupt 0y1: Interrupt requested
[0]	RORMIS	RO	0y0	Receive overrun interrupt status after enable gate 0y0: No interrupt 0y1: Interrupt requested

3.11.5.10   SSPICR (SSP interrupt clear register)

Address = (0x4004_0000) + 0x0020

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	RTIC	WO	Undefined	Clears the receive timeout interrupt flag. 0y0: No effect 0y1: Clear
[0]	RORIC	WO	Undefined	Clears the receive overrun interrupt flag. 0y0: No effect 0y1: Clear

3.11.5.11   SSPDMACR (SSPDMA control register)

Address = (0x4004_0000) + 0x0024

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	TXDMAE	R/W	0y0	Transmit FIFO DMA enable 0y0: Disable 0y1: Enable
[0]	RXDMAE	R/W	0y0	Receive FIFO DMA enable 0y0: Disable 0y1: Enable

3.12 Serial Bus Interface (SBI)

3.12.1 Outline

The TMPM322 contains five Serial Bus Interface (SBI) channels, in which the following two operating modes are included:

- I^{2}C bus mode (with multi-master capability)
- Clock-synchronous 8-bit SIO mode

In the I^{2}C bus mode, the SBI is connected to external devices via SCL and SDA. In the clock-synchronous 8-bit SIO mode, the SBI is connected to external devices via SCK, SI and SO.

The following table shows the programming required to put the SBI in each operating mode.

		Pin name (PIN No.)	Port Function Register	Port Control Register	Port Input Register	Port Open Drain Output
Channel 0	I^{2}C bus mode	SCL: PL1 (8) SDA: PL0 (7)	PLFR1[1:0] = 11	PLCR[1:0] = 11	PLIE[1:0] = 11	PLOD[1:0] = 11
	Clock-synchronous 8-bit SIO mode	SCK: PL2 (9) SI : PL1 (8) SO : PL0 (7)	PLFR1[2:0] = 111	PLCR[2:0] = 101	PLIE[2:0] = 110	PLOD[2:0] = xxx
Channel 1	I^{2}C bus mode	SCL: PG1 (71) SDA: PG0 (70)	PGFR1[1:0] = 11	PGCR[1:0] = 11	PGIE[1:0] = 11	PGOD[1:0] = 11
	Clock-synchronous 8-bit SIO mode	SCK: PG2 (72) SI : PG1 (71) SO : PG0 (70)	PGFR1[2:0] = 111	PGCR[2:0] = 101	PGIE[2:0] = 110	PGOD[2:0] = xxx
Channel 2	I^{2}C bus mode	SCL: PG5 (75) SDA: PG4 (74)	PGFR1[5:4] = 11	PGCR[5:4] = 11	PGIE[5:4] = 11	PGOD[5:4] = 11
	Clock-synchronous 8-bit SIO mode	SCK: PG6 (76) SI : PG5 (75) SO : PG4 (74)	PGFR1[6:4] = 111	PGCR[6:4] = 101	PGIE[6:4] = 110	PGOD[6:4] = xxx
Channel 3	I^{2}C bus mode	SCL: PH1 (107) SDA: PH0 (106)	PHFR1[1:0] = 11	PHCR[1:0] = 11	PHIE[1:0] = 11	PHOD[1:0] = 11
	Clock-synchronous 8-bit SIO mode	SCK: PH2 (108) SI : PH1 (107) SO : PH0 (106)	PHFR1[2:0] = 111	PHCR[2:0] = 101	PHIE[2:0] = 110	PHOD[2:0] = xxx
Channel 4	I^{2}C bus mode	SCL: PH5 (111) SDA: PH4 (110)	PHFR1[5:4] = 11	PHCR[5:4] = 11	PHIE[5:4] = 11	PHOD[5:4] = 11
	Clock-synchronous 8-bit SIO mode	SCK: PH6 (112) SI : PH5 (111) SO : PH4 (110)	PHFR1[6:4] = 111	PHCR[6:4] = 101	PHIE[6:4] = 110	PHOD[6:4] = xxx

x: Don't care

3.12.2 Configuration

The configuration is shown in Fig. 3.12.1.
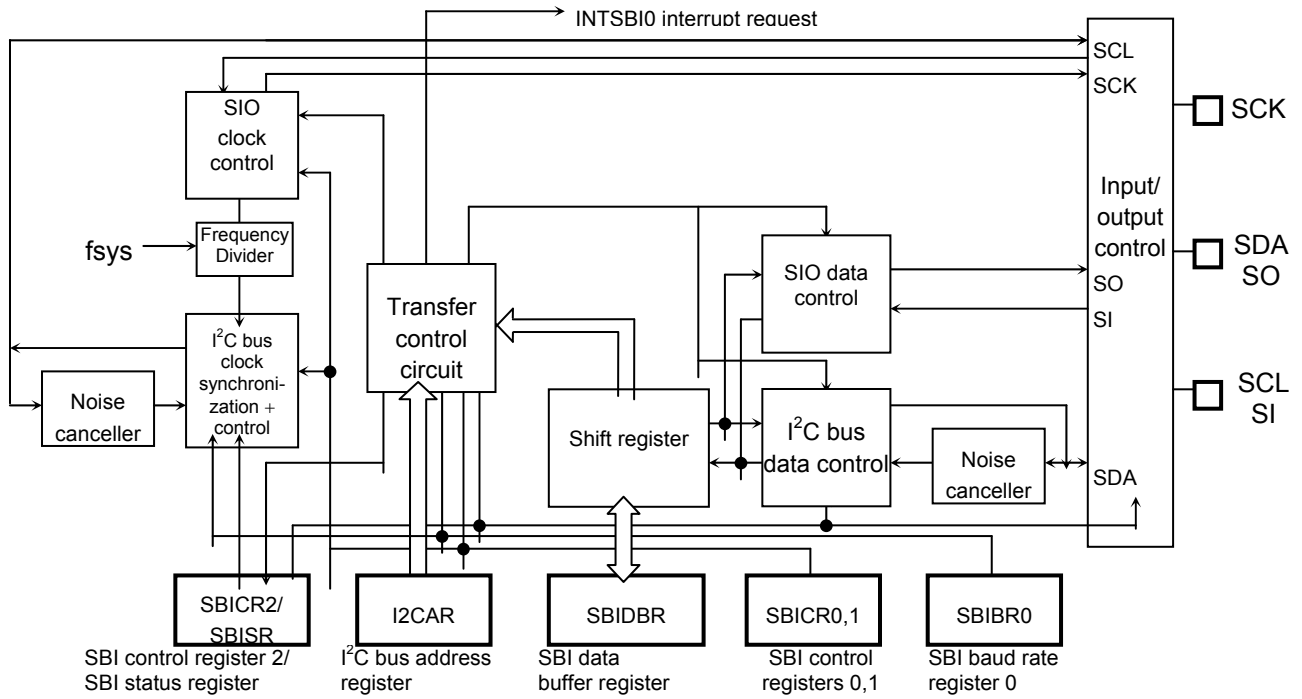
Fig. 3.12.1 SBI Block Diagram

3.12.3 I^{2}C Bus Mode Data Formats

Fig 3.12.2 shows the data formats used in the I^{2}C bus mode.

(a) Addressing format
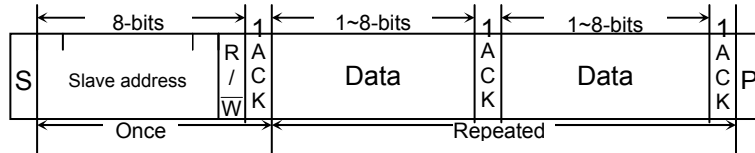
(b) Addressing format (with repeated start condition)
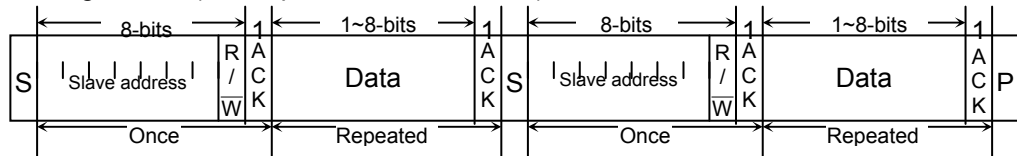
(c) Free data format (master-transmitter to slave-receiver)
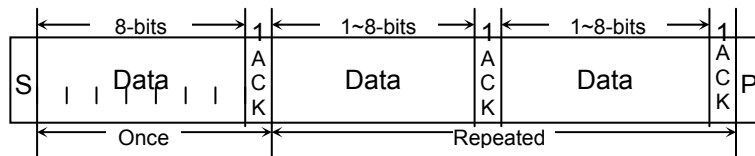
Note)   S: Start condition
        R/$\overline{\text{W}}$: Direction bit
        ACK: Acknowledge bit
        P: Stop condition

Fig. 3.12.2 I^{2}C Bus Mode Data Formats

3.12.4 Control register

The following registers control the serial bus interface and provide its status information for monitoring.

- Serial bus interface control registers 0 (SBIxCR0)

- Serial bus interface control registers 1 (SBIxCR1)

- Serial bus interface control registers 2 (SBIxCR2)

- Serial bus interface buffer registers (SBIxDBR)

- I^{2}C bus address register (SBIxI2CAR)

- Serial bus interface status registers (SBIxSR)

- Serial bus interface baud rate registers 0 (SBIxBR0)

The functions of these registers vary, depending on the mode in which the SBI is operating. For a detailed description of the registers, refer to "3.12.6 Control in the I^{2}C Bus Mode" and "3.12.8 Control in the Clock-synchronous 8-bit SIO Mode".

The addresses of each register are shown below.

Register name (address)		Channel 0		Channel 1		Channel 2	
	Serial bus interface control register 0	SBI0CR0	0x400E_0000	SBI1CR0	0x400E_0100	SBI2CR0	0x400E_0200
	Serial bus interface control register 1	SBI0CR1	0x400E_0004	SBI1CR1	0x400E_0104	SBI2CR1	0x400E_0204
	Serial bus interface control register 2	SBI0CR2 (reading)	0x400E_0010	SBI1CR2 (reading)	0x400E_0110	SBI2CR2 (reading)	0x400E_0210
	Serial bus interface status register	SBI0SR (writing)		SBI1SR (writing)		SBI2SR (writing)	
	Serial bus interface baud rate register 0	SBI0BR0	0x400E_0014	SBI1BR0	0x400E_0114	SBI2BR0	0x400E_0214
	Serial bus interface data buffer register	SBI0DBR	0x400E_0008	SBI1DBR	0x400E_0108	SBI2DBR	0x400E_0208
	I^{2}C bus address register	SBI0I2CAR	0x400E_000C	SBI1I2CAR	0x400E_010C	SBI2I2CAR	0x400E_020C

Register name (address)		Channel 3		Channel 4	
	Serial bus interface control register 0	SBI3CR0	0x400E_0300	SBI4CR0	0x400E_0400
	Serial bus interface control register 1	SBI3CR1	0x400E_0304	SBI4CR1	0x400E_0404
	Serial bus interface control register 2	SBI3CR2 (reading)	0x400E_0310	SBI4CR2 (reading)	0x400E_0410
	Serial bus interface status register	SBI3SR (writing)		SBI4SR (writing)	
	Serial bus interface baud rate register 0	SBI3BR0	0x400E_0314	SBI4BR0	0x400E_0414
	Serial bus interface data buffer register	SBI3DBR	0x400E_0308	SBI4DBR	0x400E_0408
	I^{2}C bus address register	SBI3I2CAR	0x400E_030C	SBI4I2CAR	0x400E_040C

3.12.5 Control Registers in the I2C Bus Mode

The following registers control the serial bus interface (SBI) in the I^{2}C bus mode and provide its status information for monitoring.

Serial bus control register 0

SBIxCR0		**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	bit Symbol	SBIEN							
	Read/Write	R/W	R						
	After reset	0	0						
	Function	SBI operation 0: Disable 1: Enable	"0" is read.						
		15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							
		23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							
		31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							

<SBIEN>:   To use the SBI, enable the SBI operation ("1") before setting each register in the SBI module.

Serial bus control register 1

SBIxCR1		7	6	5	4	3	2	1	0
	bit Symbol	BC			ACK		SCK2	SCK1	SCK0/ SWRMON
	Read/Write	R/W			R/W	R	R/W		R/W
	After reset	0	0	0	0	1	0	0	1
	Function	Select the number of bits per transfer (Note 1)			Acknowledgment clock 0: Not generate 1: Generate	"1" is read.	Select internal SCL output clock frequency (Note 2) and reset monitor.		
		15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							
		23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							
		31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							

<Bit 2：0><SCK2:0>: Select internal SCL output clock frequency

On writing <SCK2:0>: Select internal SCL output clock frequency

000	n=5	462 kHz	
001	n=6	353 kHz	System clock: fsys
010	n=7	240 kHz	(=48 MHz)
011	n=8	146 kHz	Clock gear : fc/1
100	n=9	82 kHz	
101	n=10	44 kHz	$\text{Frequency} = \dfrac{f_{sys}}{2^n + 72}$ [Hz]
110	n=11	23 kHz	
111		reserved	

<Bit 0>< SWRMON：0>: Software reset status monitor

On reading <SWRMON>: Software reset status monitor

0	Software reset operation is in progress.
1	Software reset operation is not in progress.

<Bit 7：5><BC[2：0]>   : Select the number of bits per transfer

Select the number of bits per transfer

<BC[2:0]>	When <ACK> = 0		When <ACK> = 1	
	Number of clock cycles	Data length	Number of clock cycles	Data length
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

(Note 1)	Clear <BC[2:0]> to "000" before switching the operation mode to the clock-synchronous 8-bit SIO mode.
(Note 2)	For details on the SCL line clock frequency, refer to "3.12.6.3 Serial Clock."
(Note 3)	After a reset, the <SCK0/SWRMON> bit is read as "1." However, if the SIO mode is selected at the SBIxCR2 register, the initial value of the <SCK0> bit is "0."

Serial bus control register 2

SBIxCR2

	7	6	5	4	3	2	1	0
bit Symbol	MST	TRX	BB	PIN	SBIM		SWRST	
Read/Write	W				W (Note2)		W (Note1)	
After reset	0	0	0	1	0	0	0	0
Function	Select master/slave 0: Slave 1: Master	Select transmit/ receive 0: Receive 1: Transmit	Start/stop condition generation 0: Stop condition generated 1: Start condition generated	Clear INTSBIn interrupt request 0: – 1: Clear interrupt request	Select serial bus interface operating mode (Note 2) 00: Port mode 01: SIO mode 10: I²C bus mode 11: (Reserved))		Software reset generation Write "10" followed by "01" to generate a reset.	

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

<Bit 1 : 0><SWRST[1:0]>   : Write "10" followed by "01" to generate a reset.
<Bit 3 : 2><SBIM[1:0]>   : Select serial bus interface operating mode

Select serial bus interface operating mode (Note 2)

00	Port mode (disables serial bus interface output)
01	Clock –synchronous 8-bit SIO mode
10	I²C bus mode
11	(Reserved)

<Bit 4><PIN>   : Clear INTSBIn interrupt request
<Bit 5><BB>   : Start/stop condition generation
<Bit 6><TRX>   : Select transmit/ receive
<Bit 7><MST>   : Select master/slave

(Note 1)   **Reading this register causes it to function as the SBInSR register.**

(Note 2)   **Ensure that the bus is free before switching the operating mode to the port mode. Ensure that the port is at the "H" level before switching the operating mode from the port mode to the I²C bus or clock-synchronous 8-bit SIO mode.**

(Note 3)   **Ensure that serial transfer is completed before switching the mode.**

Table 3.12.1 Base Clock Resolution

@fsys = 48 MHz

Clock gear value <GEAR2:0>	Base clock resolution
000 (fc)	$fsys/2^2$ (0.083 μs)
100 (fc/2)	$fsys/2^3$ (0.167μs)
101 (fc/4)	$fsys/2^4$ (0.33 μs)
110 (fc/8)	$fsys/2^5$ (0.67 μs)
111 (fc/16)	$fsys/2^6$ (1.33 μs)

Serial bus interface status register

SBIxSR		7	6	5	4	3	2	1	0
	bit Symbol	MST	TRX	BB	PIN	AL	AAS	ADO	LRB
	Read/Write				R				
	After reset	0	0	0	1	0	0	0	0
	Function	Master/ slave selection monitor 0: Slave 1: Master	Transmit/ receive selection monitor 0: Receive 1: Transmit	I^{2}C bus state monitor 0: Free 1: Busy	INTSBIn interrupt request monitor 0: Interrupt request generated 1: Interrupt request cleared	Arbitration lost detection 0: − 1: Detected	Slave address match detection 0: − 1: Detected	General call detection 0: − 1: Detected	Last received bit monitor 0: "0" 1: "1"
		15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	bit Symbol								
	Read/Write				R				
	After reset				0				
	Function				"0" is read.				
		23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	bit Symbol								
	Read/Write				R				
	After reset				0				
	Function				"0" is read.				
		31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
	bit Symbol								
	Read/Write				R				
	After reset				0				
	Function				"0" is read.				

(Note) Writing to this register causes it to function as SBI0CR2.

<Bit 0><LRB>   : Last received bit monitor

<Bit 1><ADO>   : General call detection

<Bit 2><AAS>   : Slave address match detection

<Bit 3><AL>   : Arbitration lost detection

 <Bit 4><PIN>   : INTSBIn interrupt request monitor

<Bit 5><BB>   : I^{2}C bus state monitor

<Bit 6><TRX>   : Transmit/ receive selection monitor

<Bit 7><MST>   : Master/ slave selection monitor

Serial bus interface baud rate register 0

SBIxBR0		7	6	5	4	3	2	1	0
	bit Symbol		I2SBI						
	Read/Write	R	R/W	R					R/W
	After reset	1	0	1					0
	Function	"1" is read.	IDLE 0: Stop 1:Operate	"1" is read.					Be sure to write "0." (Note)
		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							
		23	22	21	20	19	18	17	16
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							
		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							

<Bit 6><I2SBI>   : Operation at the IDLE mode

(Note)   This is read as "1" at the SIO mode.

Serial bus interface data buffer register

SBIxDBR		7	6	5	4	3	2	1	0
	bit Symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	Read/Write	R (Receive)/W (Transmit)							
	After reset	0							
	Function	RX data/ TX data.							
		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							
		23	22	21	20	19	18	17	16
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							
		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							

(Note) The transmission data must be written in to the register from the MSB (bit 7).

I²C bus address register

SBIxI2CAR		7	6	5	4	3	2	1	0
	bit Symbol	SA							ALS
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Set the slave address when the SBI acts as a slave device.							Specify address recognition mode
		15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							
		23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							
		31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function	"0" is read.							

<Bit 0><ALS>   : Specify address recognition mode

(Note)   **Please set the bit 0 <ALS> of I2C bus address register SBInI2CAR to "0", except when you use a free data format. It operates as a free data format when setting it to "1". Selecting the master fixes to transmission. Selecting the slave fixes to reception.**

3.12.6 Control in the I²C Bus Mode

3.12.6.1 Setting the Acknowledgement Mode

Setting SBIxCR1<ACK> to "1" selects the acknowledge mode. When operating as a master, the SBI adds one clock for acknowledgment signals. As a transmitter, the SBI releases the SDA pin during this clock cycle to receive acknowledgment signals from the receiver. As a receiver, the SBI pulls the SDA pin to the "L" level during this clock cycle and generates acknowledgment signals.

By setting <ACK> to "0", the non-acknowledgment mode is activated. When operating as a master, the SBI does not generate clock for acknowledgement signals.

3.12.6.2 Setting the Number of Bits per Transfer

SBIxCR1 <BC[2:0]> specifies the number of bits of the next data to be transmitted or received.

Under the start condition, <BC[2:0]> is set to "000," causing a slave address and the direction bit to be transferred in a packet of eight bits. At other times, <BC[2:0]> keeps a previously programmed value.

3.12.6.3 Serial Clock

① Clock source

SBIxCR1 <SCK2:0> specifies the maximum frequency of the serial clock to be output from the SCL pin in the master mode.

$t_{LOW} = 2^{n-1}/f_{sys} + 58/f_{sys}$

$t_{HIGH} = 2^{n-1}/f_{sys} + 14/f_{sys}$

$f_{scl} = 1/(t_{Low} + t_{HIGH})$

$\quad = \dfrac{f_{sys}}{2^n + 72}$

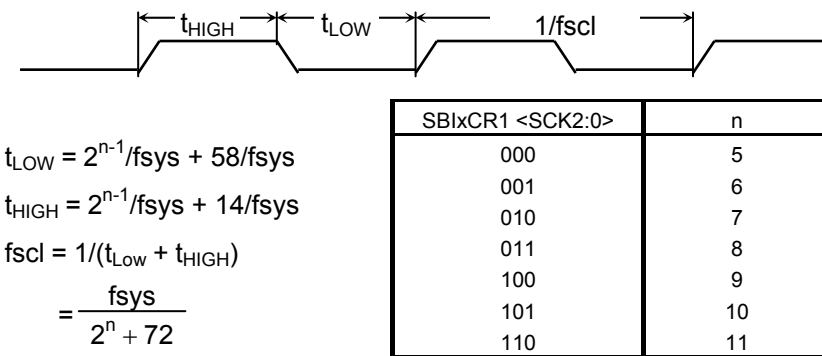| SBIxCR1 <SCK2:0>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Fig. 3.12.3 Clock Source

(Note)	**The highest speeds in the standard and high-speed modes are specified to 100KHz and 400KHz respectively following the communications standards. Note that the internal SCL clock frequency is determined by the fsys used and the calculation formula shown above.**

② Clock Synchronization

The I^{2}C bus is driven by using the wired-AND connection due to its pin structure. The first master that pulls its clock line to the "L" level overrides other masters producing the "H" level on their clock lines. This must be detected and responded by the masters producing the "H" level.

Clock synchronization assures correct data transfer on a bus that has two or more master.

For example, the clock synchronization procedure for a bus with two masters is shown below.
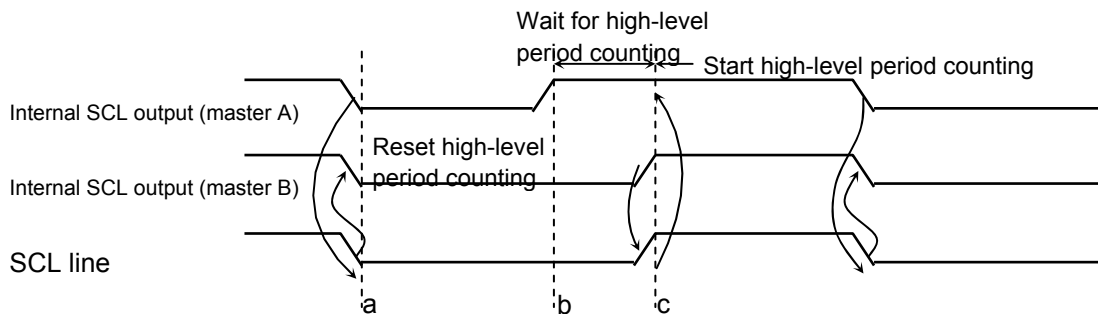
Fig. 3.12.4 Example of Clock Synchronization

At the point a, Master A pulls its internal SCL output to the "L" level, bringing the SCL bus line to the "L" level. Master B detects this transition, resets its "H" level period counter, and pulls its internal SCL output level to the "L" level.

Master A completes counting of its "L" level period at the point b, and brings its internal SCL output to the "H" level. However, Master B still keeps the SCL bus line at the "L" level, and Master A stops counting of its "H" level period counting. After Master A detects that Master B brings its internal SCL output to the "H" level and brings the SCL bus line to the "H" level at the point c, it starts counting of its "H" level period.

This way, the clock on the bus is determined by the master with the shortest "H" level period and the master with the longest "L" level period among those connected to the bus.

3.12.6.4 Slave Addressing and Address Recognition Mode

When the SBI is configured to operate as a slave device, the slave address <SA[6:0]> and <ALS> must be set at SBIxI2CAR. Setting <ALS> to "0" selects the address recognition mode.

3.12.6.5 Configuring the SBI as a Master or a Slave

Setting SBIxCR2<MST> to "1" configures the SBI to operate as a master device.

Setting <MST> to "0" configures the SBI as a slave device. <MST> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

3.12.6.6 Configuring the SBI as a Transmitter or a Receiver

Setting SBIxCR2 <TRX> to "1" configures the SBI as a transmitter. Setting <TRX> to "0" configures the SBI as a receiver.

At the slave mode, the SBI receives the direction bit ($R/\overline{W}$) from the master device on the following occasions:

- when data is transmitted in the addressing format
- when the received slave address matches the value specified at I2CCR
- when a general-call address is received; i.e., the eight bits following the start condition are all zeros

If the value of the direction bit ($R/\overline{W}$) is "1," <TRX> is set to "1" by the hardware. If the bit is "0," <TRX> is set to "0".

As a master device, the SBI receives acknowledgement from a slave device. If the direction bit of "1" is transmitted, <TRX> is set to "0" by the hardware. If the direction bit is "0," <TRX> changes to "1." If the SBI does not receive acknowledgement, <TRX> retains the previous value.

<TRX> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

3.12.6.7 Generating Start and Stop Conditions

When SBIxSR<BB> is "0," writing "1" to SBIxCR2 <MST, TRX, BB, PIN> causes the SBI to generate the start condition on the bus and output 8-bit data. <ACK> must be set to "1" in advance.
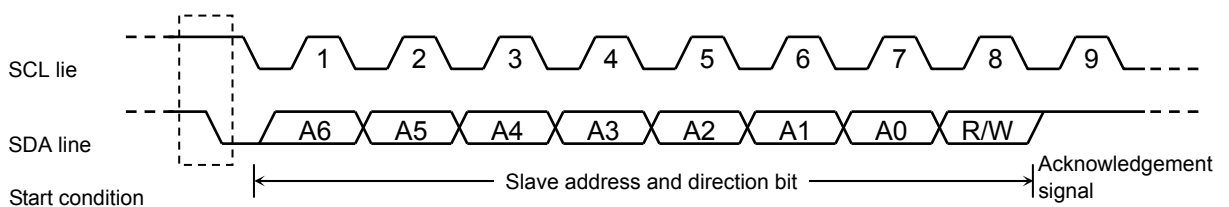
Fig. 3.12.5 Generating the Start Condition and a Slave Address

When <BB> is "1," writing "1" to <MST, TRX, PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus. The contents of <MST, TRX, BB, PIN> should not be altered until the stop condition appears on the bus.
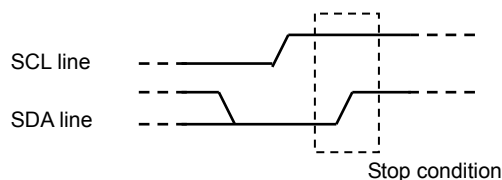
Fig. 3.12.6 Generating the Stop Condition

SBIxSR<BB> can be read to check the bus state. <BB> is set to "1" when the start condition is detected on the bus (the bus is busy), and set to "0" when the stop condition is detected (the bus is free).

3.12.6.8 Interrupt Service Request and Release

When a serial bus interface interrupt request (INTSBI0) is generated, SBIxCR2 <PIN> is cleared to "0." While <PIN> is "0," the SBI pulls the SCL line to the "L" level.

After transmission or reception of one data word, <PIN> is cleared to "0." It is set to "1" when data is written to or read from SBIxDBR. It takes a period of t_{LOW} for the SCL line to be released after <PIN> is set to "1."

In the address recognition mode (<ALS> = "0"), <PIN> is cleared to "0" when the received slave address matches the value specified at SBIxI2CAR or when a general-call address is received; i.e., the eight bits following the start condition are all zeros. When the program writes "1" to SBIxCR2<PIN>, it is set to "1." However, writing "0" does clear this bit to "0".

3.12.6.9 Serial Bus Interface Operating Modes

SBIxCR2 <SBIM[1:0]> selects an operating mode of the serial bus interface. <SBIM[1:0]> must be set to "10" to configure the SBI for the I^{2}C bus mode. Make sure that the bus is free before switching the operating mode to the port mode.

3.12.6.10   Lost-arbitration Detection Monitor

The I^{2}C bus has the multi-master capability (there are two or more masters on a bus), and requires the bus arbitration procedure to ensure correct data transfer.

A master that attempts to generate the start condition while the bus is busy loses bus arbitration, with no start condition occurring on the SDA and SCL lines. The I^{2}C-bus arbitration takes place on the SDA line.

The arbitration procedure for two masters on a bus is shown below. Up until the point a, Master A and Master B output the same data. At the point a, Master A outputs the "L" level and Master B outputs the "H" level. Then Master A pulls the SDA bus line to the "L" level because the line has the wired-AND connection. When the SCL line goes high at the point b, the slave device reads the SDA line data, i.e., data transmitted by Master A. At this time, data transmitted by Master B becomes invalid. This condition of Master B is called "Lost Arbitration". Master B releases its SDA pin, so that it does not affect the data transfer initiated by another master. If two or more masters have transmitted exactly the same first data word, the arbitration procedure continues with the second data word.
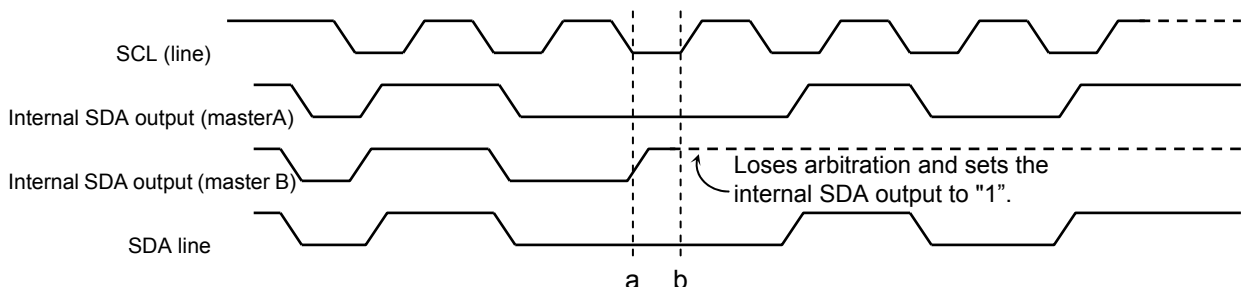
Fig. 3.12.7 Lost Arbitration

A master compares the SDA bus line level and the internal SDA output level at the rising of the SCL line. If there is a difference between these two values, Arbitration Lost occurs

and SBIxSR <AL> is set to "1".

When <AL> is set to "1," SBIxSR <MST, TRX> are cleared to "0," causing the SBI to operate as a slave receiver. <AL> is cleared to "0" when data is written to or read from SBIxDBR or data is written to SBIxCR2.
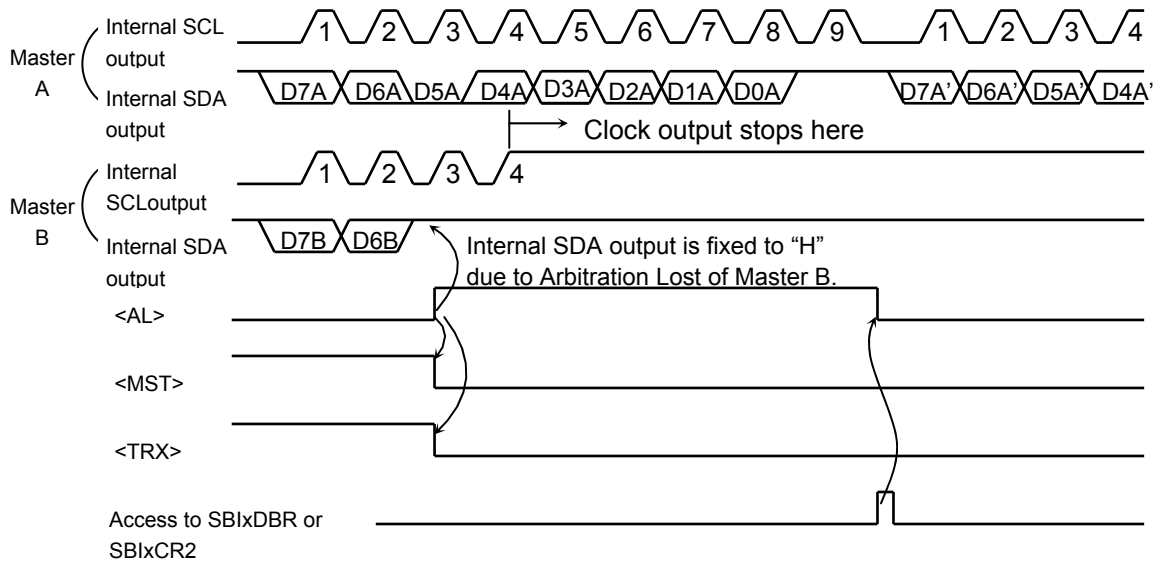
Fig. 3.12.8 Example of Master B Lost Arbitration (D7A = D7B, D6A = D6B)

3.12.6.11 Slave Address Match Detection Monitor

When the SBI operates as a slave device in the address recognition mode (SBIxI2CAR <ALS> ("0"), SBIxSR <AAS> is set to "1" on receiving the general-call address or the slave address that matches the value specified at SBIxI2CAR. When <ALS> is "1," <AAS> is set to "1" when the first data word has been received. <AAS> is cleared to "0" when data is written to or read from SBIxDBR.

3.12.6.12 General-call Detection Monitor

When the SBI operates as a slave device, SBIxSR <AD0> is set to "1" when it receives the general-call address; i.e., the eight bits following the start condition are all zeros. <AD0> is cleared to "0" when the start or stop condition is detected on the bus.

3.12.6.13 Last Received Bit Monitor

SBIxSR <LRB> is set to the SDA line value that was read at the rising of the SCL line. In the acknowledgment mode, reading SBIxSR <LRB> immediately after generation of the INTSBIx interrupt request causes ACK signal to be read.

3.12.6.14   Software Reset

If the serial bus interface circuit locks up due to external noise, it can be initialized by using a software reset.

Writing "10" followed by "01" to SBIxCR2 <SWRST[1:0]> generates a reset signal that initializes the serial bus interface circuit. After a reset, all control registers and status flags are initialized to their reset values. When the serial bus interface is initialized, <SWRST> is automatically cleared to "0".

(Note)	**A software reset causes the SBI operating mode to switch from the I^{2}C mode to the port mode.**

3.12.6.15   Serial Bus Interface Data Buffer Register (SBIxDBR)

Reading or writing SBIxDBR initiates reading received data or writing transmitted data. When the SBI is acting as a master, setting a slave address and a direction bit to this register generates the start condition.

3.12.6.16   I^{2}C Bus Address Register (SBIxI2CAR)

When the SBI is configured as a slave device, the SBIxI2CAR<SA[6:0]> bit is used to specify a slave address. If I2CAR <ALS> is set to "0," the SBI recognizes a slave address transmitted by the master device and receives data in the addressing format. If <ALS> is set to "1," the SBI does not recognize a slave address and receives data in the free data format.

3.12.6.17   IDLE Setting Register (SBIxBR0)

The SBIxBR0<I2SBI> register determines if the SBI operates or not when it enters the IDLE mode. This register must be programmed before executing an instruction to switch to the standby mode.

3.12.7 Data Transfer Procedure in the I²C Bus Mode

3.12.7.1 Device Initialization

First, program SBIxCR1<ACK, SCK2:0> by writing "0" to bits 7 to 5 and bit 3 in SBIxCR1.

Next, program SBIxI2CAR by specifying a slave address at <SA[6:0]> and an address recognition mode at <ALS>. (<ALS> must be set to"0" when using the addressing format).

Then program SBIxCR2 to initially configure the SBI in the slave receiver mode by writing "0" to <MST, TRX, BB> , "1" to <PIN> , "10" to <SBIM[1:0]> and "0" to bits 1 and 0.

```
                7 6 5 4 3 2 1 0
SBIxCR1  ← 0 0 0 X 0 X X X       Specifies ACK and SCL clock.
SBIxI2CA ← X X X X X X X X       Specifies a slave address and an address recognition
R                               mode.
SBIxCR2  ← 0 0 0 1 1 0 0 0       Configures the SBI as a slave receiver.
(Note) X: Don't care
```

3.12.7.2 Generating the Start Condition and a Slave Address

① Master mode

In the master mode, the following steps are required to generate the start condition and a slave address.

First, ensure that the bus is free (<BB> = "0"). Then, write "1" to SBIxCR1 <ACK> to select the acknowledgment mode. Write to SBIxDBR a slave address and a direction bit to be transmitted.

When <BB> = "0," writing "1111" to SBIxCR2 <MST, TRX, BB, PIN> generates the start condition on the bus. Following the start condition, the SBI generates nine clocks from the SCL pin. The SBI outputs the slave address and the direction bit specified at SBIxDBR with the first eight clocks, and releases the SDA line in the ninth clock to receive an acknowledgment signal from the slave device.

The INTSBIx interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0." In the master mode, the SBI holds the SCL line at the "L" level while <PIN> is "0." <TRX> changes its value according to the transmitted direction bit at generation of the INTSBIx interrupt request, provided that an acknowledgment signal has been returned from the slave device.

Settings in main routine
```
                7 6 5 4 3 2 1 0
→ Reg.        ← SBIxSR
  Reg.        ← Reg. e 0x20
  if Reg.     ≠ 0x00                Ensures that the bus is free.
└ Then
  SBIxCR1  ← X X X 1 0 X X X        Selects the acknowledgement mode.
  SBIxDR1  ← X X X X X X X X        Specifies the desired slave address and direction.
  SBIxCR2  ← 1 1 1 1 1 0 0 0        Generates the start condition.
```

Example of INTSBI0 interrupt routine

Clears the interrupt request.
Processing
End of interrupt

② Slave mode

In the slave mode, the SBI receives the start condition and a slave address.

After receiving the start condition from the master device, the SBI receives a slave address and a direction bit from the master device during the first eight clocks on the SCL line. If the received address matches its slave address specified at SBIxI2CAR or is equal to the general-call address, the SBI pulls the SDA line to the "L" level during the ninth clock and outputs an acknowledgment signal.

The INTSBIx interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0." In the slave mode, the SBI holds the SCL line at the "L" level while <PIN> is "0".
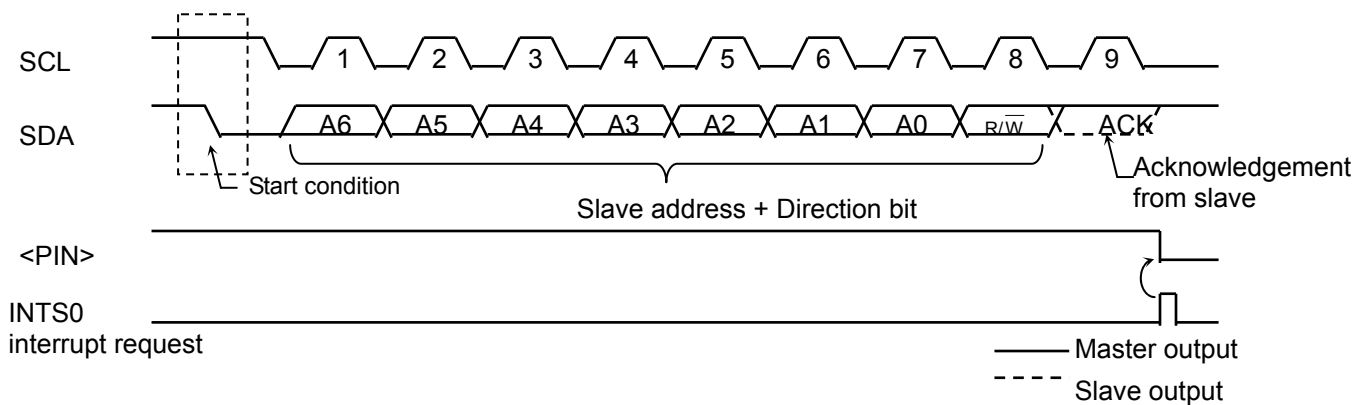
Fig. 3.12.9 Generation of the Start Condition and a Slave Address

3.12.7.3 Trasferring a Data Word

At the end of a data word transfer, the INTSBIx interrupt is generated to test <MST> to determine whether the SBI is in the master or slave mode.

① Master mode (<MST> = "1")

Test <TRX> to determine whether the SBI is configured as a transmitter or a receiver.

Transmitter mode (<TRX> = "1")

Test <LRB>. If <LRB> is "1," that means the receiver requires no further data. The master then generates the stop condition as described later to stop transmission.

If <LRB> is "0," that means the receiver requires further data. If the next data to be transmitted has eight bits, the data is written into SBIxDBR. If the data has different length, <BC[2:0]> and <ACK> are programmed and the transmit data is written into SBIxDBR. Writing the data makes <PIN> to"1," causing the SCL pin to generate a serial clock for transferring a next data word, and the SDA pin to transfer the data word. After the transfer is completed, the INTSBIx interrupt request is generated, <PIN> is set to "0," and the SCL pin is pulled to the "L" level. To transmit more data words, test <LRB> again and repeat the above procedure.

INTSBIx interrupt

if MST = 0
Then go to the slave-mode processing
if TRX = 0
Then go to the receiver-mode processing
if LRB = 0
Then go to processing for generating the stop condition
SBIxCR1 ← X X X X 0 X X X   Specifies the number of bits to be transmitted and specify whether ACK is required.

SBIxDBR ← X X X X X X X X   Writes the transmit data.
End of interrupt processing
(Note)   X: Don't care
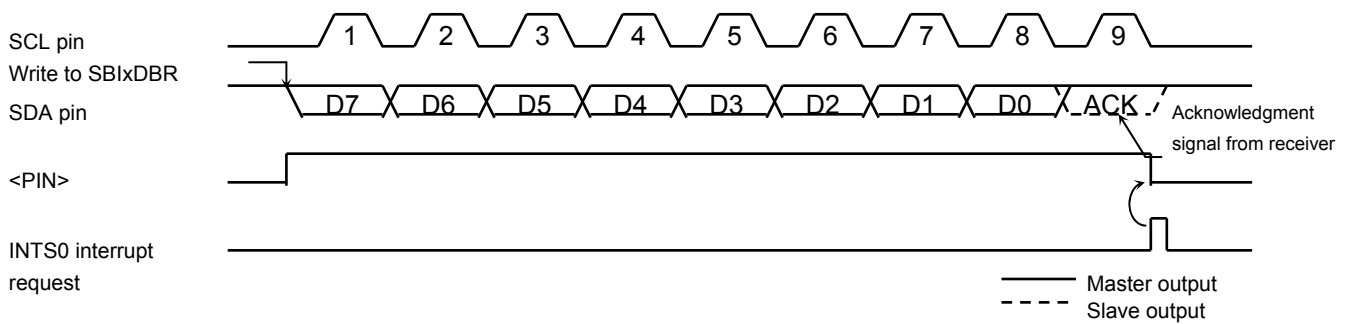
Fig. 3.12.10 <BC[2:0]> = "000" and <ACK> = "1" (Transmitter Mode)

Receiver mode (<TRX> = "0")

If the next data to be transmitted has eight bits, the transmit data is written into SBIxDBR. If the data has different length, <BC[2:0]> and <ACK> are programmed and the received data is read from SBIxDBR to release the SCL line. (The data read immediately after transmission of a slave address is undefined.) On reading the data, <PIN> is set to "1," and the serial clock is output to the SCL pin to transfer the next data word. In the last bit, when the acknowledgment signal becomes the "L" level, "0" is output to the SDA pin.

After that, the INTSBIx interrupt request is generated, and <PIN> is cleared to "0," pulling the SCL pin to the "L" level. Each time the received data is read from SBIxDBR, one-word transfer clock and an acknowledgement signal are output.
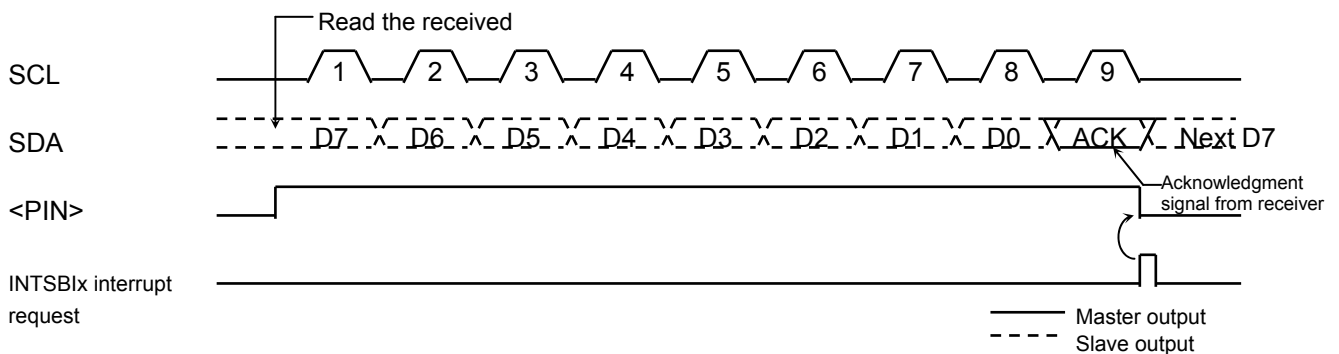
Fig. 3.12.11 <BC[2:0]> = "000" and <ACK> = "1" (Receiver Mode)

To terminate the data transmission from the transmitter, <ACK> must be set to "0" immediately before reading the data word second to last. This disables generation of an acknowledgment clock for the last data word. When the transfer is completed, an interrupt request is generated. After the interrupt processing, <BC[2:0]> must be set to "001" and the data must be read so that a clock is generated for 1-bit transfer. At this time, the master receiver holds the SDA bus line at the "H" level, which signals the end of transfer to the transmitter as an acknowledgment signal.

In the interrupt processing for terminating the reception of 1-bit data, the stop condition is generated to terminate the data transfer.
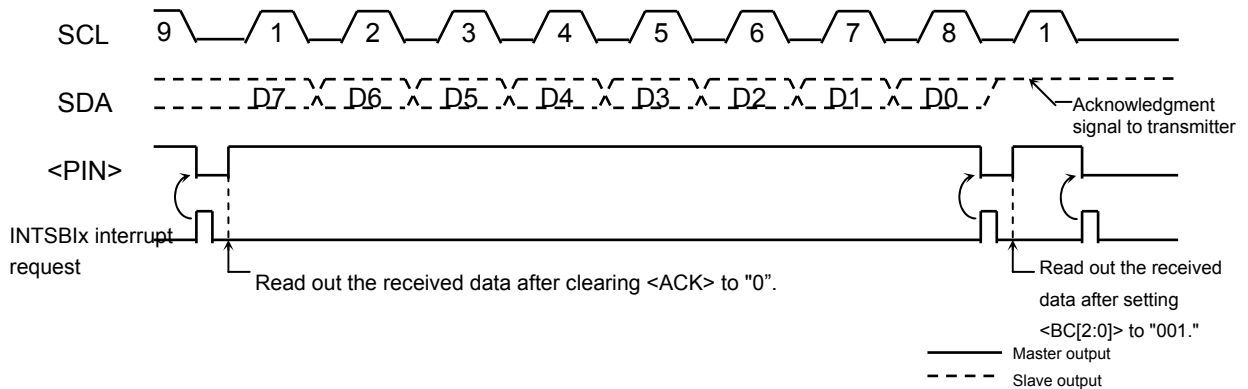
Fig. 3.12.12 Terminating Data Transmission in the Master Receiver Mode

Example: When receiving N data word

INTSBIx interrupt (after data transmission)

```
               7 6 5 4 3 2 1 0
SBIxCR1  ← X X X X 0 X X X        Sets the number of bits of data to be received and
                                  specify whether ACK is required.
Reg.      ← SBIxDBR               Reads dummy data.
End of interrupt
```

INTSBIx interrupt (first to (N-2)th data reception)

```
               7 6 5 4 3 2 1 0
Reg.      ← SBIxDBR               Reads the first to (N-2)th data words.
End of interrupt
```

INTSBIx interrupt ((N-1)th data reception)

```
               7 6 5 4 3 2 1 0
SBIxCR1  ← X X X 0 0 X X X        Disables generation of acknowledgement clock.
Reg.      ← SBIxDBR               Reads the (N-1)th data word.
End of interrupt
```

INTSBIx interrupt (Nth data reception)

```
               7 6 5 4 3 2 1 0
SBIxCR1  ← 0 0 1 0 0 X X X        Disables generation of acknowledgement clock.
Reg.      ← SBIxDBR               Reads the Nth data word.
End of interrupt
```

INTSBIx interrupt (after completing data reception)

```
Processing to generate the stop condition.   Terminates the data transmission.
End of interrupt
```

```
(Note)              X: Don't care
```

② Slave mode (<MST> = "0")

In the slave mode, the SBI generates the INTSBIx interrupt request on four occasions: 1) when the SBI has received any slave address from the master, 2) when the SBI has received a general-call address, 3) when the received slave address matches its own address, and 4) when a data transfer has been completed in response to a general-call. Also, if the SBI detects Arbitration Lost in the master mode, it switches to the slave mode. Upon the completion of data word transfer in which Arbitration Lost is detected, the INTSBIx interrupt request is generated, <PIN> is cleared to "0," and the SCL pin is pulled to the "L" level. When data is written to or read from SBIxDBR or when <PIN> is set to "1," the SCL pin is released after a period of t_{LOW}.

In the slave mode, the normal slave mode processing or the processing as a result of Arbitration Lost is carried out.

SBIxSR <AL>, <TRX>, <AAS> and <AD0> are tested to determine the processing required.

Table 3.12.2 shows the slave mode states and required processing.

Example: When the received slave address matches the SBI's own address and the direction bit is "1" in the slave receiver mode.

INTSBIx interrupt

```
if TRX = 0
Then go to other processing
if AL = 1
Then go to other processing
if AAS = 0
Then go to other processing
SBIxCR1  ← X X X 1 0 X X X        Sets the number of bits to be transmitted.
SBIxDBR  ← X X X X 0 X X X        Sets the transmit data.

(Note) X: Don't care
```

Table 3.12.2 Processing in Slave Mode

\<TRX\>	\<AL\>	\<AAS\>	\<AD0\>	State	Processing
1	1	1	0	Arbitration Lost is detected while the slave address was being transmitted and the SBI received a slave address with the direction bit "1" transmitted by another master.	Set the number of bits in a data word to \<BC[2:0]\> and write the transmit data into SBIxDBR.
	0	1	0	In the slave receiver mode, the SBI received a slave address with the direction bit "1" transmitted by the master.	
	0	0	0	In the slave transmitter mode, the SBI has completed a transmission of one data word.	Test LRB. If it has been set to "1," that means the receiver does not require further data. Set \<PIN\> to 1 and reset \<TRX\> to 0 to release the bus. If \<LRB\> has been reset to "0," that means the receiver requires further data. Set the number of bits in the data word to \<BC[2:0]\> and write the transmit data to the SBIxDBR.
0	1	1	1/0	Arbitration Lost is detected while a slave address is being transmitted, and the SBI receives either a slave address with the direction bit "0" or a general-call address transmitted by another master.	Read the SBIxDBR (a dummy read) to set \<PIN\> to 1, or write "1" to \<PIN\>.
		0	0	Arbitration Lost is detected while a slave address or a data word is being transmitted, and the transfer is terminated.	
	0	1	1/0	In the slave receiver mode, the SBI received either a slave address with the direction bit "0" or a general-call address transmitted by the master.	
		0	1/0	In the slave receiver mode, the SBI has completed a reception of a data word.	Set the number of bits in the data word to \<BC[2:0]\> and read the received data from SBIxDBR.

3.12.7.4 Generating the Stop Condition

When SBIxSR <BB> is "1," writing "1" to SBIxCR2 <MST, TRX, PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus. Do not alter the contents of <MST, TRX, BB, PIN> until the stop condition appears on the bus.

If another device is holding down the SCL bus line, the SBI waits until the SCL line is released. After that, the SDA pin goes high, causing the stop condition to be generated.

```
              7 6 5 4 3 2 1 0
SBIxCR2 ← 1 1 0 1 1 0 0 0          Generates the stop condition.
```

"1" → <MST>
"1" → <TRX>
"0" → <BB>
"1" → <PIN>

Stop condition

SCL pin

SDA pin

<PIN>

<BB> (read)
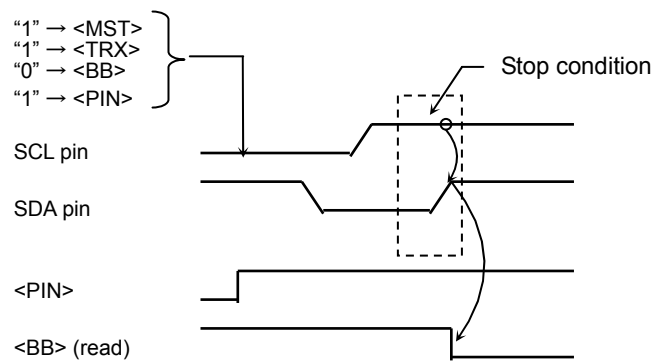
Fig. 3.12.13 Generating the Stop Condition

3.12.7.5 Restart Procedure

Restart is used when a master device changes the data transfer direction without terminating the transfer to a slave device. The procedure of generating a restart in the master mode is described below.

First, set SBIxCR2 <MST, TRX, BB> to "0" and write "1" to <PIN> to release the bus. At this time, the SDA pin is held at the "H" level and the SCL pin is released. Because no stop condition is generated on the bus, other devices recognize that the bus is busy. Then, test SBIxSR <BB> and wait until it becomes "0" to ensure that the SCL pin is released. Next, test <LRB> and wait until it becomes "1" to ensure that no other device is pulling the SCL bus line to the "L" level. Once the bus is determined to be free this way, use the above-mentioned steps 3.12.7.2 to generate the start condition.

To satisfy the setup time of restart, at least 4.7-μs wait period (in the standard mode) must be created by the software after the bus is determined to be free.

```
                7 6 5 4 3 2 1 0
  SBIxCR2  ←  0 0 0 1 1 0 0 0          Releases the bus.
  if SBISR<BB> ≠ 0                     Checks that the SCL pin is released.
  Then
  if SBIxSR<LRB> ≠ 1                   Checks that no other device is pulling the SCL pin to the
                                       "L" level.

  Then
  4.7 μs Wait
  SBIxCR1  ←  X X X 1 0 X X X          Selects the acknowledgment mode.
  SBIxDBR  ←  X X X X X X X X          Sets the desired slave address and direction.
  SBIxCR2  ←  1 1 1 1 1 0 0 0          Generates the start condition.
```

(Note) X: Don't care
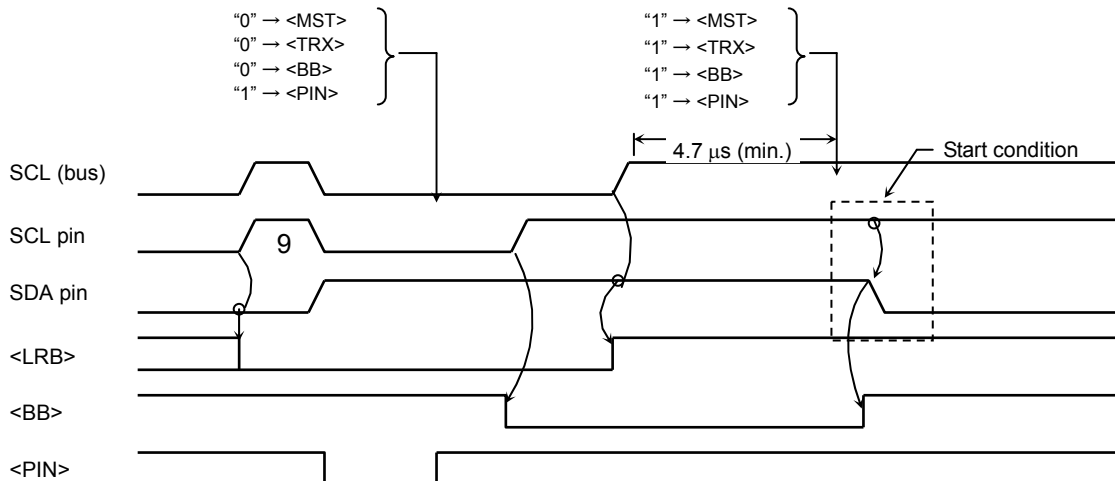
| (Note) | Do not write <MST> to "0" when it is "0." (Restart cannot be initiated.) |

Fig.3.12.14 Timing Chart of Generating a Restart

3.12.8 Control in the Clock-synchronous 8-bit SIO Mode

3.12.8.1 Control Registers in the Clock-synchronous SIO Mode

The following registers control the serial bus interface in the clock-synchronous 8-bit SIO mode and provide its status information for monitoring.

Serial bus interface control register 0

SBIxCR0

	7	6	5	4	3	2	1	0
bit Symbol	SBIEN							
Read/Write	R/W	R						
After reset	0	0						
Function	SBI operation 0: Disable 1: Enable	"0" is read.						

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

<SBIEN>: To use the SBI, enable the SBI operation ("1") before setting each register of SBI module.

Serial bus interface control register 1

SBIxCR1

	7	6	5	4	3	2	1	0
bit Symbol	SIOS	SIOINH	SIOM			SCK2	SCK1	SCK0
Read/Write	W				R	W		R/W
After reset	0	0	0	0	1	0	0	0
Function	Start transfer 0: Stop 1: Start	Transfer 0: Continue 1:Forced termination	Select transfer mode 00: Transmit mode 01: (Reserved) 10: Transmit/receive mode 11: Receive mode		"1" is read.	Select serial clock frequency		

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

On writing <SCK2:0>: Select serial clock frequency

000	n = 3	3 MHz
001	n = 4	1.5 MHz
010	n = 5	750 MHz
011	n = 6	375 kHz
100	n = 7	186 kHz
101	n = 8	93.6 kHz
110	n = 9	46.9 kHz
111	—	External clock

System clock : fsys
(=48 MHz)
Clock gear : fc/1
Frequency = [Hz]

(Note) Set <SIOS> to "0" and <SIOINH> to "1" before programming the transfer mode and the serial clock.

Serial bus interface data buffer register

SBIxDBR

	7	6	5	4	3	2	1	0
bit Symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Receive)/W (Transmit)							
After reset	Undefined							
Function	RX data/ TX data							
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

Serial bus interface control register 2

SBIxCR2

	7	6	5	4	3	2	1	0
bit Symbol					SBIM1	SBIM0		
Read/Write	R				W		R	
After reset	1				0	0	1	
Function	"1" is read.				Select serial bus interface operating mode 00: Port mode 01: Clock-synchronous 8-bit SIO mode 10: I^{2}C bus mode 11: (Reserved)		"1" is read.	
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

Serial bus interface register

SBIxSR		**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	bit Symbol					SIOF	SEF		
	Read/Write			R			R		R
	After reset			1		0	0		1
	Function			"1" is read.		Serial transfer status monitor 0: Completed 1: In progress	Shift operation status monitor 0: Completed 1: In progress		"1" is read.
		15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	bit Symbol								
	Read/Write					R			
	After reset					0			
	Function					"0" is read.			
		23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	bit Symbol								
	Read/Write					R			
	After reset					0			
	Function					"0" is read.			
		31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
	bit Symbol								
	Read/Write					R			
	After reset					0			
	Function					"0" is read.			

Serial bus interface baud rate register 0

SBIxBR0		**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	bit Symbol		I2SBI						
	Read/Write	R	R/W			R			W
	After reset	1	0			1			0
	Function	"1" is read.	IDLE 0: Stop 1: Operate	"1" is read.					Make sure to write "0."
		15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	bit Symbol								
	Read/Write					R			
	After reset					0			
	Function					"0" is read.			
		23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	bit Symbol								
	Read/Write					R			
	After reset					0			
	Function					"0" is read.			
		31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
	bit Symbol								
	Read/Write					R			
	After reset					0			
	Function					"0" is read.			

3.12.8.2 Serial Clock

①   Clock source

Internal or external clocks can be selected by programming SBIxCR1 <SCK2:0>.

Internal clocks

In the internal clock mode, one of the seven frequencies can be selected as a serial clock, which is output to the outside through the SCK pin. At the beginning of a transfer, the SCK pin output becomes the "H" level.

If the program cannot keep up with this serial clock rate in writing the transmit data or reading the received data, the SBI automatically enters a wait period. During this period, the serial clock is stopped automatically and the next shift operation is suspended until the processing is completed.
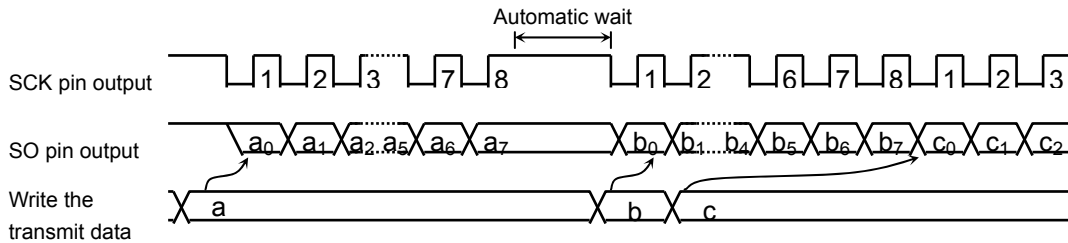
Fig. 3.12.15 Automatic Wait

External clock (<SCK2:0> = "111")

The SBI uses an external clock supplied from the outside to the SCK pin as a serial clock. For proper shift operations, the serial clock at the "H" and "L" levels must have the pulse widths as shown below.

t_{SCKL}, $t_{SCKH} > 4/fsys$
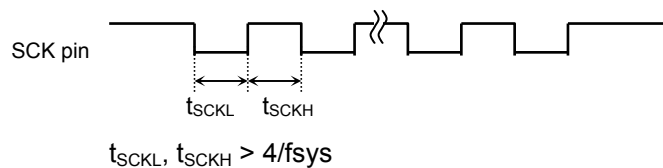
Fig. 3.12.16 Maximum Transfer Frequency of External Clock Input

②   Shift Edge

Leading-edge shift is used in transmission. Trailing-edge shift is used in reception.

Leading-edge shift

Data is shifted at the leading edge of the serial clock (or the falling edge of the SCK pin input/output).

Trailing-edge shift

Data is shifted at the trailing edge of the serial clock (or the rising edge of the SCK pin input/output).

(a) Leading-edge shift

(b) Trailing-edge shift

(Note) ＊; Don't care
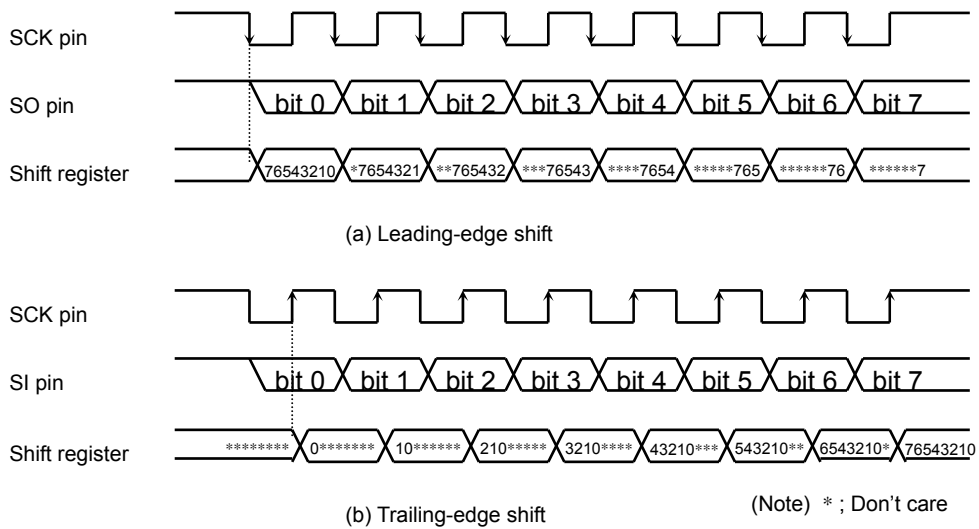
Fig. 3.12.17 Shift Edge

3.12.8.3 Transfer Modes

The transmit mode, the receive mode or the transmit/receive mode can be selected by programming SBIxCR1 <SIOM[1:0]>.

① 8-bit transmit mode

Set the control register to the transmit mode and write the transmit data to SBIxDBR.

After writing the transmit data, writing "1" to SBIxCR1 <SIOS> starts the transmission. The transmit data is moved from SBIxDBR to a shift register and output to the SO pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the transmit data is transferred to the shift register, SBIxDBR becomes empty, and the INTSBIx (buffer-empty) interrupt is generated, requesting the next transmit data.

In the internal clock mode, the serial clock will be stopped and automatically enter the wait state, if next data is not loaded after the 8-bit data has been fully transmitted. The wait state will be cleared when SBIxDBR is loaded with the next transmit data.

In the external clock mode, SBIxDBR must be loaded with data before the next data shift operation is started. Therefore, the data transfer rate varies depending on the maximum latency between when the interrupt request is generated and when SBIxDBR is loaded with data in the interrupt service program.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting SBIxSR <SIOF> to "1" to the falling edge of SCK.

Transmission can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, remaining data is output before transmission ends. The program checks SBIxSR <SIOF> to determine whether transmission has come to an end. <SIOF> is cleared to "0" at the end of transmission. If <SIOINH> is set to "1," the transmission is aborted immediately and <SIOF> is cleared to "0".

In the external clock mode, <SIOS> must be set to "0" before the next transmit data shift operation is started. Otherwise, operation will stop after dummy data is transmitted.

```
              7 6 5 4 3 2 1 0
SBIxCR1  ← 0 1 0 0 0 X X X          Selects the transmit mode.

SBIxDBR  ← X X X X X X X X          Writes the transmit data.
SBIxCR1  ← 1 0 0 0 0 X X X          Starts transmission.
```

INTSBIx interrupt
```
SBIxDBR  ← X X X X X X X X          Writes the transmit data.
```

<SIOS> is cleared.

<SIOS>

<SIOF>

<SEF>

SCK pin (output)

SO pin $*$ a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7

INTSBIx
interrupt request

SBIxDBR a b

↑   ↑
Write the transmit data   (a) Internal clock

<SIOS> is cleared.

<SIOS>

<SIOF>

<SEF>

SCK pin (input)

SO pin $*$ a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7

INTSBIx
interrupt request

SBIxDBR a b

↑   ↑
Write the transmit data   (b) External clock
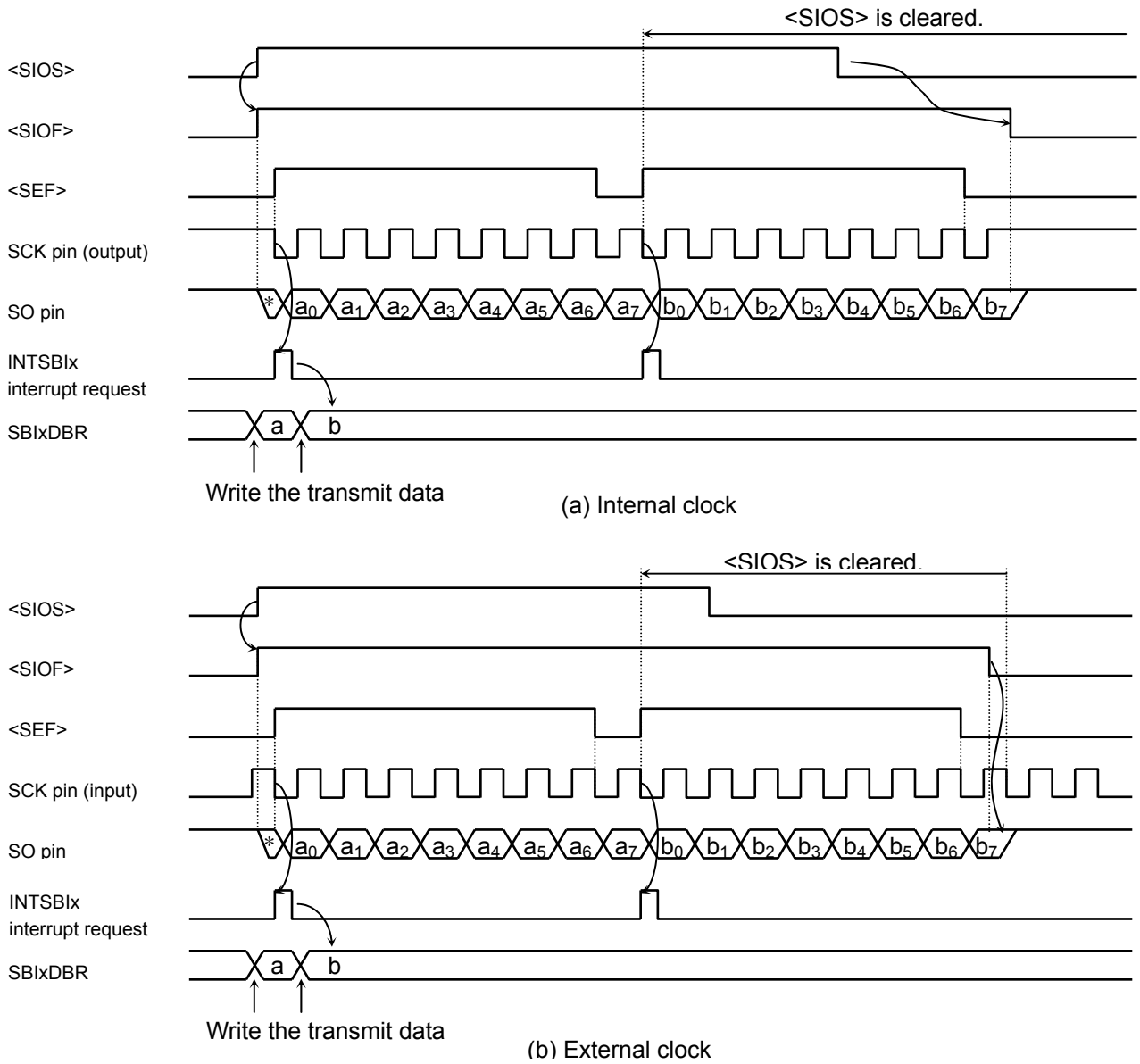
Fig. 3.12.18 Transfer Mode

Example: Example of programming (external clock) to terminate transmission by <SIO>

```
                7 6 5 4 3 2 1 0
  ⌐→  if SBISR<SIOF> ≠ 0              Recognizes the completion of the transmission.
  └   Then
  ⌐→  if SCK ≠ 1                      Recognizes "1" is set to the SCK pin by monitoring the port.
  └   Then
      SBIxCR1  ← 0 0 0 0 0 1 1 1      Completes the transmission by setting <SIOS> = 0.
```

② 8-bit receive mode

Set the control register to the receive mode. Then writing "1" to SBIxCR1 <SIOS> enables reception. Data is taken into the shift register from the SI pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIxDBR and the INTSBIx (buffer-full) interrupt request is generated to request reading the received data. The interrupt service program then reads the received data from SBIxDBR.

In the internal clock mode, the serial clock will be stopped and automatically be in the wait state until the received data is read from SBIxDBR.

In the external clock mode, shift operations are executed in synchronization with the external clock. The maximum data transfer rate varies, depending on the maximum latency between generating the interrupt request and reading the received data.

Reception can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, reception continues until all the bits of received data are written to SBIxDBR. The program checks SBIxSR <SIOF> to determine whether reception has come to an end. <SIOF> is cleared to "0" at the end of reception. After confirming the completion of the reception, last received data is read. If <SIOINH> is set to "1," the reception is aborted immediately and <SIOF> is cleared to "0." (The received data becomes invalid, and there is no need to read it out.)

(Note) **The contents of SBIxDBR will not be retained after the transfer mode is changed. The ongoing reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.**

```
                7 6 5 4 3 2 1 0
SBIxCR1  ←  0 1 1 1 0 X X X        Selects the receive mode.

SBIxCR1  ←  1 0 1 1 0 0 0 0        Starts reception.
```

INTSBIx interrupt

    Reg.   ← SBIxDBR   Reads the received data.
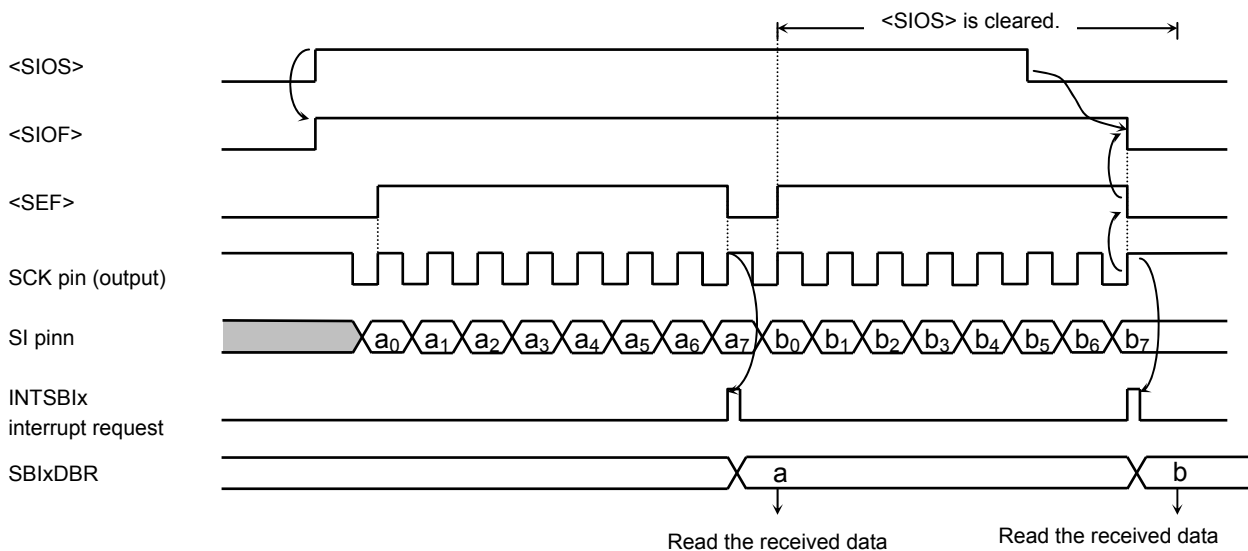
Fig. 3.12.19 Receive Mode (Example: Internal Clock)

③　8-bit transmit/receive mode

Set the control register to the transfer/receive mode. Then writing the transmit data to SBIxDBR and setting SBIxCR1 <SIOS> to "1" enables transmission and reception. The transmit data is output through the SO pin at the falling of the serial clock, and the received data is taken in through the SI pin at the rising of the serial clock, with the least-significant bit (LSB) first. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIxDBR and the INTSBIx interrupt request is generated. The interrupt service program reads the received data from the data buffer register and writes the next transmit data. Because SBIxDBR is shared between transmit and receive operations, the received data must be read before the next transmit data is written.

In the internal clock operation, the serial clock will be automatically in the wait state until the received data is read and the next transmit data is written.

In the external clock mode, shift operations are executed in synchronization with the external serial clock. Therefore, the received data must be read and the next transmit data must be written before the next shift operation is started. The maximum data transfer rate for the external clock operation varies depending on the maximum latency between when the interrupt request is generated and when the transmit data is written.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting <SIOF> to "1" to the falling edge of SCK.

Transmission and reception can be terminated by clearing <SIOS> to "0" or setting SBIxCR1 <SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, transmission and reception continue until the received data is fully transferred to SBIxDBR. The program checks SBIxSR <SIOF> to determine whether transmission and reception have come to an end. <SIOF> is cleared to "0" at the end of transmission and reception. If <SIOINH> is set, the transmission and reception are aborted immediately and <SIOF> is cleared to "0."

(Note) **The contents of SBIxDBR will not be retained after the transfer mode is changed. The ongoing transmission and reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.**
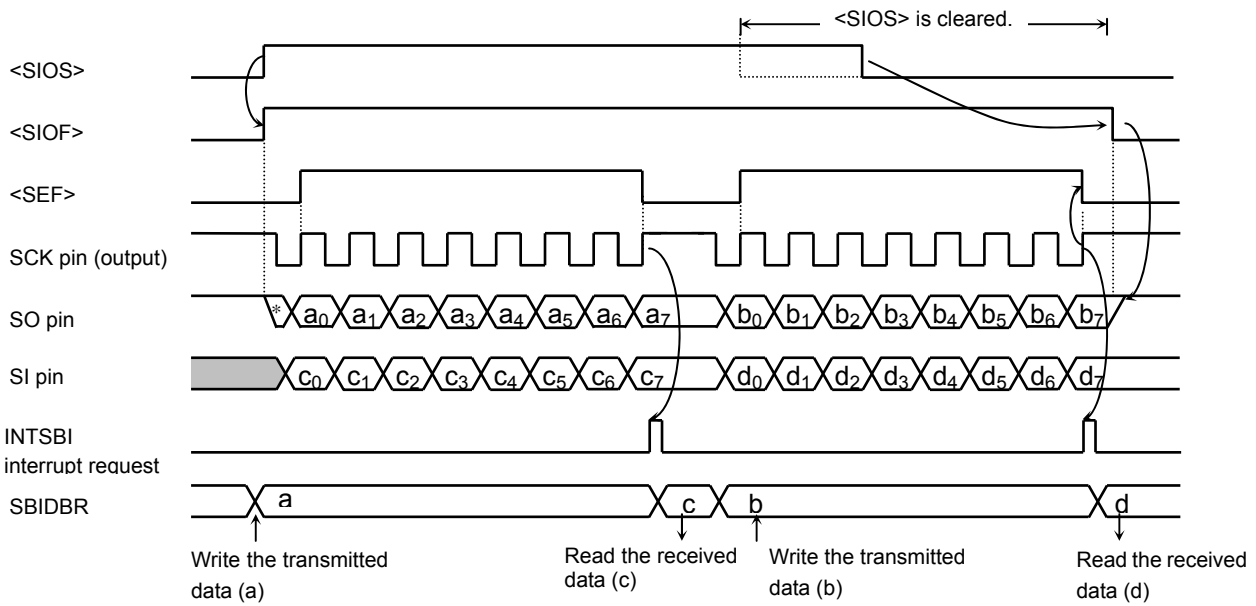
Fig. 3.12.20 Transmit/Receive Mode (Example: Internal Clock)

```
                7 6 5 4 3 2 1 0
SBIxCR1  ← 0 1 1 0 0 X X X          Selects the transmit mode.

SBIxDBR  ← X X X X X X X X          Writes the transmit data.
SBIxCR1  ← 1 0 1 0 0 X X X          Starts reception/transmission.
```

INTSBIx interrupt

```
Reg.        ← SBIxDBR               Reads the received data.
SBIxDBR  ← X X X X X X X X          Writes the transmit data.
```

3.13 Remote control signal preprocessor (RMC)

3.13.1 Basic operation

Remote control signal preprocessor (hereafter referred to as RMC) receives a remote control signal of which carrier is removed.

3.13.1.1 Reception of Remote Control Signal

- Enables to set as sampling clock Low-Speed oscillator (32kHz) or Timer Output
- Noise canceller
- Leader detection
- Batch reception up to 72bit of data

3.13.2 Registers

3.13.2.1 Register Map

Addresses and names of RMC control registers are shown below.

Register		Address	
		Channel 0	Channel 1
Remote Control Enable Register	RMCxEN	0x400E_3000	0x400E_3100
Remote Control Receive Enable Register	RMCxREN	0x400E_3004	0x400E_3104
Remote Control Receive Data Buffer Register 1	RMCxRBUF1	0x400E_3008	0x400E_3108
Remote Control Receive Data Buffer Register 2	RMCxRBUF2	0x400E_300C	0x400E_310C
Remote Control Receive Data Buffer Register 3	RMCxRBUF3	0x400E_3010	0x400E_3110
Remote Control Receive Control Register 1	RMCxRCR1	0x400E_3014	0x400E_3114
Remote Control Receive Control Register 2	RMCxRCR2	0x400E_3018	0x400E_3118
Remote Control Receive Control Register 3	RMCxRCR3	0x400E_301C	0x400E_311C
Remote Control Receive Control Register 4	RMCxRCR4	0x400E_3020	0x400E_3120
Remote Control Receive Status Register	RMCxRSTAT	0x400E_3024	0x400E_3124
Remote Control Receive End Bit Number Register 1	RMCxEND1	0x400E_3028	0x400E_3128
Remote Control Receive End Bit Number Register 2	RMCxEND2	0x400E_302C	0x400E_312C
Remote Control Receive End Bit Number Register 3	RMCxEND3	0x400E_3030	0x400E_3130
Remote Control Source Clock Select Register	RMCxFSSEL	0x400E_3034	0x400E_3134

3.13.2.2 Remote Control Enable Register [RMCEN]

	7	6	5	4	3	2	1	0
bit Symbol	—						I2RMC	RMCEN
Read/Write	R						R/W	R/W
After reset	0						0	0
Function	"0" is read.						RMC in IDLE mode 0: Disabled 1: Enabled	RMC operation 0: Disabled 1: Enabled

〈I2RMC〉: Controls RMC operation in IDLE mode.
Set "1" to enable RMC in IDLE Mode.
This bit and the <RMCEN> bit can be set simultaneously.

〈RMCEN〉: Controls RMC operation.
To allow RMC to function, enable the RMCEN bit first. If the operation is disabled, all the clocks for RMC except for the enable register are stopped, and it can reduce power consumption.
If RMC is enabled and then disabled, the settings in each register remain intact.

3.13.2.3 Remote Control Receive Enable Register [RMCREN]

	7	6	5	4	3	2	1	0
bit Symbol	—							RMCREN
Read/Write	R							R/W
After reset	0							0
Function	"0" is read.							Reception 0: Disabled 1: Enabled

〈RMCREN〉: Controls reception of RMC.
Setting this bit to "1" enables reception.

(Note)	**Enable the <RMCREN> bit after setting the RMC0RCR1, RMC0RCR2 and RMC0RCR3.**

3.13.2.4 Remote Control Receive Data Buffer Register 1 [RMCRBUF1]

	31	30	29	28	27	26	25	24
bit Symbol	RMCRBUF[31:24]							
Read/Write	R							
After reset	0							
Function	Received data							

	23	22	21	20	19	18	17	16
bit Symbol	RMCRBUF[23:16]							
Read/Write	R							
After reset	0							
Function	Received data							

	15	14	13	12	11	10	9	8
bit Symbol	RMCRBUF[15:8]							
Read/Write	R							
After reset	0							
Function	Received data							

	7	6	5	4	3	2	1	0
bit Symbol	RMCRBUF[7:0]							
Read/Write	R							
After reset	0							
Function	Received data							

〈RMCRBUF31:0〉:   Reads 4 bytes of received data.

3.13.2.5   Remote Control Receive Data Buffer Register 2 [RMCRBUF2]

	31	30	29	28	27	26	25	24
bit Symbol	RMCRBUF [63:56]							
Read/Write	R							
After reset	0							
Function	Received data							
	23	22	21	20	19	18	17	16
bit Symbol	RMCRBUF [55:48]							
Read/Write	R							
After reset	0							
Function	Received data							
	15	14	13	12	11	10	9	8
bit Symbol	RMCRBUF [47:38]							
Read/Write	R							
After reset	0							
Function	Received data							
	7	6	5	4	3	2	1	0
bit Symbol	RMCRBUF [39:32]							
Read/Write	R							
After reset	0							
Function	Received data							

⟨RMCRBUF63:32⟩:   Reads 4 bytes of received data.

3.13.2.6   Remote Control Receive Data Buffer Register 3 [RMCRBUF3]

	7	6	5	4	3	2	1	0
bit Symbol	RMCRBUF [71:64]							
Read/Write	R							
After reset	0							
Function	Received data							

⟨RMCRBUF71:64⟩:   Reads a byte of received data.

(Note1)	**Received data is stored from RMCRBUF1 <RMCRBUF0> to RMCRBUF3 <RMCRBUF71> in sequence.**
(Note2)	**The first received bit is stored in the MSB. The last received bit is stored in the LSB (bit 0). (The RMCRBUF0 is continued the data.)**
	If the remote control signal is received in the LSB first algorithm, the received data is stored in reverse sequence.

3.13.2.7 Remote Control Receive Control Register 1 [RMCRCR1]

	31	30	29	28	27	26	25	24
bit Symbol	RMCLCMAX							
Read/Write	R/W							
After reset	0							
Function	Maximum cycle of leader detection: RMCLCMAX×4/fs[s]							
	23	22	21	20	19	18	17	16
bit Symbol	RMCLCMIN							
Read/Write	R/W							
After reset	0							
Function	Minimum cycle of leader detection: RMCLCMIN×4/fs[s]							
	15	14	13	12	11	10	9	8
bit Symbol	RMCLLMAX							
Read/Write	R/W							
After reset	0							
Function	Maximum low width of leader detection: RMCLLMAX×4/fs[s]							
	7	6	5	4	3	2	1	0
bit Symbol	RMCLLMIN							
Read/Write	R/W							
After reset	0							
Function	Minimum low width of leader detection: RMCLLMIN×4/fs[s]							

⟨RMCLCMAX[7:0]⟩: Specifies a maximum cycle of leader detection.
Calculating formula of the maximum cycle: RMCLCMAX×4/fs[s].
RMC detects the first cycle as a leader if it is within the maximum cycle.

⟨RMCLCMIN[7:0]⟩: Specifies a minimum cycle of leader detection.
Calculating formula of the minimum cycle: RMCLCMIN×4/fs[s].
RMC detects the first cycle as a leader if it exceeds the minimum cycle.

⟨RMCLLMAX[7:0]⟩: Specifies a maximum low width of leader detection.
Calculating formula of the maximum low width: RMCLLMAX×4/fs[s]
RMC detects the first cycle as a leader if its low width is within the maximum low width.

⟨RMCLLMIN[7:0]⟩: Specifies a minimum low width of leader detection.
Calculating formula of the minimum low width: RMCLLMIN×4/fs[s]
RMC detects the first cycle as a leader if its low width exceeds the minimum low width.
If RMCRCR2<RMCLD> = 1, a value less than the specified is determined as data.

(Note)   When you configure the register, you must follow the rule shown below.

Leader	Rules
Low width + high width	<RMCLCMAX[7:0]> > <RMCLCMIN[7:0]> <RMCLLMAX[7:0]> > <RMCLLMIN[7:0]> <RMCLCMIN[7:0]> > <RMCLLMAX[7:0]>
Only with high width	<RMCLCMAX[7:0]> > <RMCLCMIN[7:0]> <RMCLLMAX[7:0]> = 0x00000000 <RMCLLMIN[7:0]> = don't care
No leader	<RMCLCMAX[7:0]> = 0x00000000 <RMCLCMIN[7:0]> = don't care <RMCLLMAX[7:0]> = don't care <RMCLLMIN[7:0]> = don't care

3.13.2.8 Remote Control Receive Control Register 2 [RMCRCR2]

	31	30	29	28	27	26	25	24
bit Symbol	RMCLIEN	RMCEDIEN	—	—	—	—	RMCLD	RMCPHM
Read/Write	R/W	R/W	R				R/W	R/W
After reset	0	0	0				0	0
Function	Leader detection interrupt 0: Not generated 1: Generated	Remote control input falling edge interrupt 0: Not generated 1: Generated	"0" is read.				Receiving remote control signal with or without leader 0: Disabled 1: Enabled	Receive a remote control signal in phase method? 0: No (receive in cycle method) 1:Yes

	23	22	21	20	19	18	17	16
bit Symbol	—	—	—	—	—	—	—	—
Read/Write	R							
After reset	0							
Function	"0" is read.							

	15	14	13	12	11	10	9	8
bit Symbol	RMCLL							
Read/Write	R/W							
After reset	1							
Function	Excess low width that triggers reception completion and interrupt generation 00000000~11111110:RMCLL×1/fs[s] 11111111:not to use as the trigger							

	7	6	5	4	3	2	1	0
bit Symbol	RMCDMAX							
Read/Write	R/W							
After reset	1							
Function	Maximum data bit cycle that triggers reception completion and interrupt generation 00000000~11111110:RMCDMAX×1/fs[s] 11111111: not to use as the trigger							

<RMCLIEN>:   Enables to generate a leader detection interrupt by detecting a leader.

<RMCEDIEN>:   Enables to generate a remote control input falling edge Interrupt.

<RMCLD>:   Enables RMC to receive signals with or without a leader.

<RMCPHM>:   Specifies data reception mode of a phase method. If you use the phase method of which signal cycle is fixed, set "1".

<RMCLL7:0>:   Specifies an excess low width. If an excess low width is detected, reception is completed and an interrupt is generated. The low width is not detected if <RMCLL[7:0]> = 11111111b. Calculating formula of an excess low width: RMCLLx1/fs[s].

<RMCDMAX7:0>:   Specifies a threshold for detecting a maximum data bit cycle. It is detected when a data bit cycle exceeds the threshold. It is not detected when <RMCMAX[7:0]> = 11111111b. Calculating formula of the threshold: RMCDMAX x 1/fs[s].

3.13.2.9 Remote Control Receive Control Register 3 [RMCRCR3]

	15	14	13	12	11	10	9	8
bit Symbol	—	RMCDATH						
Read/Write	R	W						
After reset	0	0						
Function	"0" is read.	Larger threshold to determine a signal pattern in a phase method RMCDATH×1/fs[s]						

	7	6	5	4	3	2	1	0
bit Symbol	—	RMCDATL						
Read/Write	R	W						
After reset	0	0						
Function	"0" is read.	Threshold to determine 0 or 1/ smaller threshold to determine a signal pattern in a phase method RMCDATL×1/fs[s]						

〈RMCDATH[6:0]〉: Specifies a larger threshold (within a range of 1.5T and 2T) to determine a pattern of remote control signal in a phase method. If the measured cycle exceeds the threshold, the bit is determined as "10". If not, the bit is determined as "01". Calculating formula of the threshold: RMCDATHx1/fs[s].

〈RMCDATL[6:0]〉: Specifies two kinds of thresholds: a threshold to determine whether a data bit is 0 or 1; a smaller threshold (within a range of 1T and 1.5T) to determine a pattern of remote control signal in a phase method.
As for the determination of data bit, if the measured cycle exceeds the threshold, the bit is determined as "1". If not, the bit is determined as "0". Calculating formula of the threshold: RMCDATL×1/fs[s].
As for the determination of a remote control signal pattern in a phase method, if the measured cycle exceeds the threshold, the bit is determined as "01". If not, the bit is determined as "00". Calculating formula of the threshold to determine 0 or 1: RMCDATLx1/fs[s].

(Note)	If the <RMCPHM> bit of the Remote Control Receive Control Register 2 is "0", <RMCDATH[6:0]> are not enabled. The bits are enabled when <RMCPHM> is "1".

3.13.2.10 Remote Control Receive Control Register 4 [RMCRCR4]

	7	6	5	4	3	2	1	0
bit Symbol	RMCP0	—	—	—	RMCNC			
Read/Write	R/W	R			R/W			
After reset	0	0			0			
Function	Remote control input signal 0: Not reversed 1: Reversed	"0" is read.			Noise cancellation time 0000: No cancellation 0001~1111:RMCNC×1/fs[s]			

⟨RMCP0⟩:   Specifies whether a remote control input signal is reversed or not.

⟨RMCNC[3:0]⟩:   Specifies time noises are cancelled by a noise canceller. If <RMCNC[3:0]> = 0000b, noises are not cancelled. Calculating formula of noise cancellation time: RMCNC x 1/fs[s].

3.13.2.11　Remote Control Receive Status Register [RMCRSTAT]

	15	14	13	12	11	10	9	8
bit Symbol	RMCRLIF	RMCLOIF	RMCDMAXIF	RMCEDIF	—	—	—	—
Read/Write	R	R	R	R	R			
After reset	0	0	0	0	0			
Function	Leader detection is interrupt factor? 0: No 1: Yes	Low width detection is interrupt factor? 0: No 1: Yes	Maximum data bit cycle detection is interrupt factor? 0: No 1: Yes	Remote control input falling edge interrupt is interrupt factor? 0: No 1: Yes	"0" is read.			

	7	6	5	4	3	2	1	0
bit Symbol	RMCRLDR	RMCRNUM						
Read/Write	R	R						
After reset	0	0						
Function	Leader detection 0: No 1: Yes	The number of received data bit 0000000:no data bit (only with leader) 0000001~1001000:1~72bit 1001001~1111111:73bit and more						

〈RMCRLIF〉:　　　Indicates that leader detection is the interrupt factor.

〈RMCLOIF〉:　　　Indicates that low width detection is the interrupt factor.

〈RMCDMAXIF〉:　Indicates that maximum data bit cycle detection is the interrupt factor.

〈RMCEDIF〉:　　　Indicates that a remote control input falling edge interrupt is the interrupt factor.

〈RMCRLDR〉:　　　Detects a leader of a received remote control signal.

〈RMCRNUM[6:0]〉:　Indicates the number of bits received as remote control signal data. The number cannot be monitored during reception. On completion of reception, the number is stored.

(Note 1)　**This register is updated every time an interrupt is generated.**
　　　　　Writing to this register is ignored.

(Note 2)　**RMC keeps receiving 73 bit or more data unless reception is completed by detecting the maximum data bit cycle or the excess low width. If so, the received data in the data buffer may not be correct.**

3.13.2.12 Remote Control Receive End Bit Number Register 1 [RMCxEND1]

	7	6	5	4	3	2	1	0
bit Symbol	—	RMCEND1						
Read/Write	R	R/W						
After reset	0	0						
Function	"0" is read.	Specifies that the number of receive data bit 0000000: No specifically the receive data bit 0000001 to 1001000: 1bit to 72bit 1001001 to 1111111: Don't set the value						

<RMCEND1[6:0]>: Specifies that the number of receive data bit

3.13.2.13 Remote Control Receive End Bit Number Register 2 [RMCxEND2]

	7	6	5	4	3	2	1	0
bit Symbol	—	RMCEND2						
Read/Write	R	R/W						
After reset	0	0						
Function	"0" is read.	Specifies that the number of receive data bit 0000000: No specifically the receive data bit 0000001 to 1001000: 1bit to 72bit 1001001 to 1111111: Don't set the value						

<RMCEND2[6:0]>: Specifies that the number of receive data bit

3.13.2.14 Remote Control Receive End Bit Number Register 3 [RMCxEND3]

	7	6	5	4	3	2	1	0
bit Symbol	—	RMCEND3						
Read/Write	R	R/W						
After reset	0	0						
Function	"0" is read.	Specifies that the number of receive data bit 0000000: No specifically the receive data bit 0000001 to 1001000: 1bit to 72bit 1001001 to 1111111: Don't set the value						

<RMCEND3[6:0]>: Specifies that the number of receive data bit

(Note 1) As specified to RMCxEND[3:1], it is able to set three kinds of the receive data bit. **(Note 2) To use the RMCxEND[3:1] is in combination with the maximum data bit cycle.**

3.13.2.15 Remote Control Source Clock Select Register [RMCxFSSEL]

	7	6	5	4	3	2	1	0
bit Symbol	—							RMCCLK
Read/Write	R							R/W
After reset	0							0
Function	"0" is read.							RMC Sampling Clock 0 : Low-Speed Oscillator (32kHz) 1 : TBBOUT

<RMCCLK>:　　　　Specifies a sampling clock of RMC function.
　　　　　　　　For the Sampling of RMC function ,It is able to set the Low Frequency Clock(32KHz) or Timer output （TBBOUT）.
　　　　　　　　The Setting range of Timer output by TBBOUT is from 30KHz to 34KHz.

（**Note**）	**To Change the sampling clock by RMCxFSSEL, at first Stopping RMC operation by RMCxEN<bit0> ,** **After resetting enable operation , set the RMCxFSSEL before the other RMC registers.**

3.13.3 Operation description

3.13.3.1   Reception of Remote Control Signal

3.13.3.1.1 Sampling Clock

A remote control signal is sampled by low-speed clock (fs).

3.13.3.1.2 Basic Operation

RMC starts to receive a data bit if a leader is detected while RMC is waiting for a leader. Based on a falling edge cycle, the data bit is determined as 0 or 1. By detecting a leader while RMC is waiting for a leader, a leader detection interrupt is generated, and the data bit reception starts. The data bit is determined as 0 or 1 based on a falling edge cycle. RMC is capable of receiving data up to 72bit. Reception is completed by detecting either a maximum data bit cycle or the excess low width. On completion of reception, RMC is waiting for the next leader, and the Remote Control Receive Data Buffer Registers and the Remote Control Receive Status Register are updated.
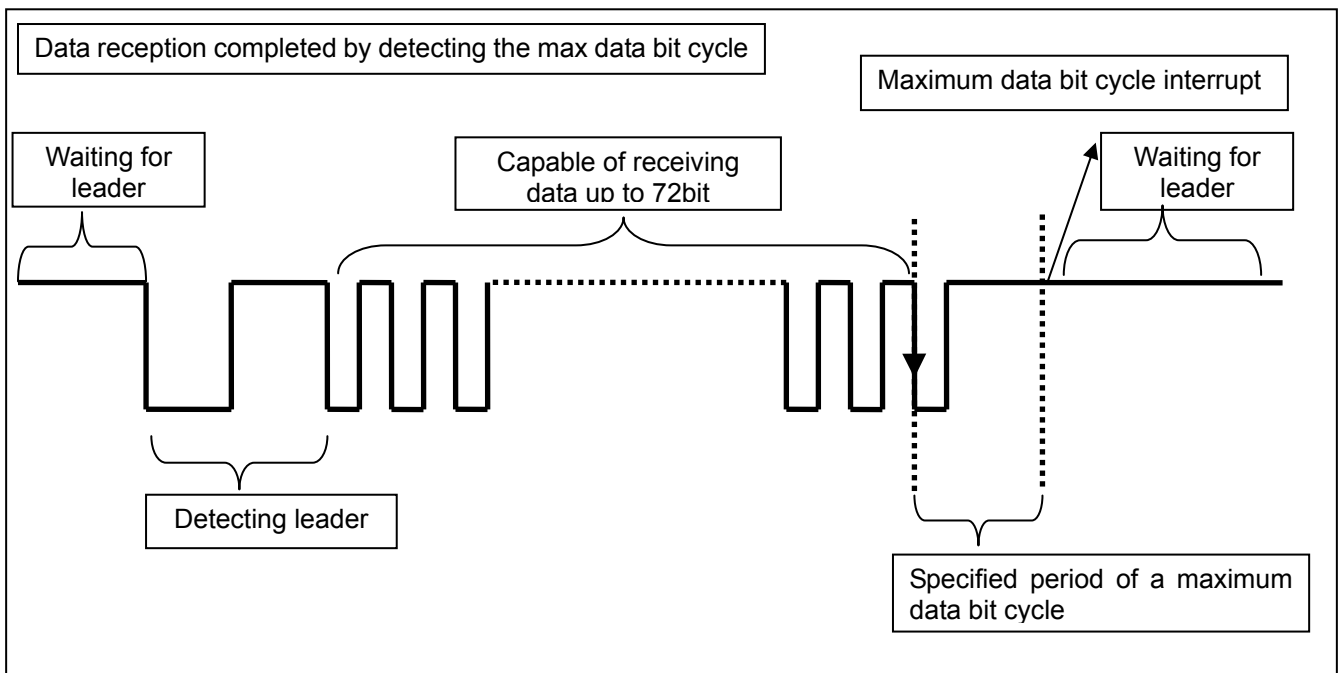
3.13.3.1.3 Preparation

Configure reception operation of a remote control signal with the Remote Control Signal Receive Control Registers (RMCRCR1, RMCRCR 2 and RMCRCR3) before reception.

(1) Setting of Noise Canceling Time

Configure noise cancelling time with the RMCRCR4 <RMCNC[3:0]> bit.

RMC monitors a remote control signal in each rising edge of a sampling clock. If "1" is monitored, RMC recognizes that the signal was changed to "0" after monitoring cycles of "0"s specified in RMCNC. If "0" is monitored, RMC recognizes that the signal was changed to "1" after monitoring cycles of "1"s specified in RMCNC.

The following figure shows how RMC operates according to the noise cancel setting of RMCNC [3:0] = 0011 (3 cycles). Subsequent to noise cancellation, the signal is changed from "1" to "0" upon monitoring 3 cycles of "0" s, and the signal is changed from "0" to "1" upon monitoring 3 cycles of "1" s.
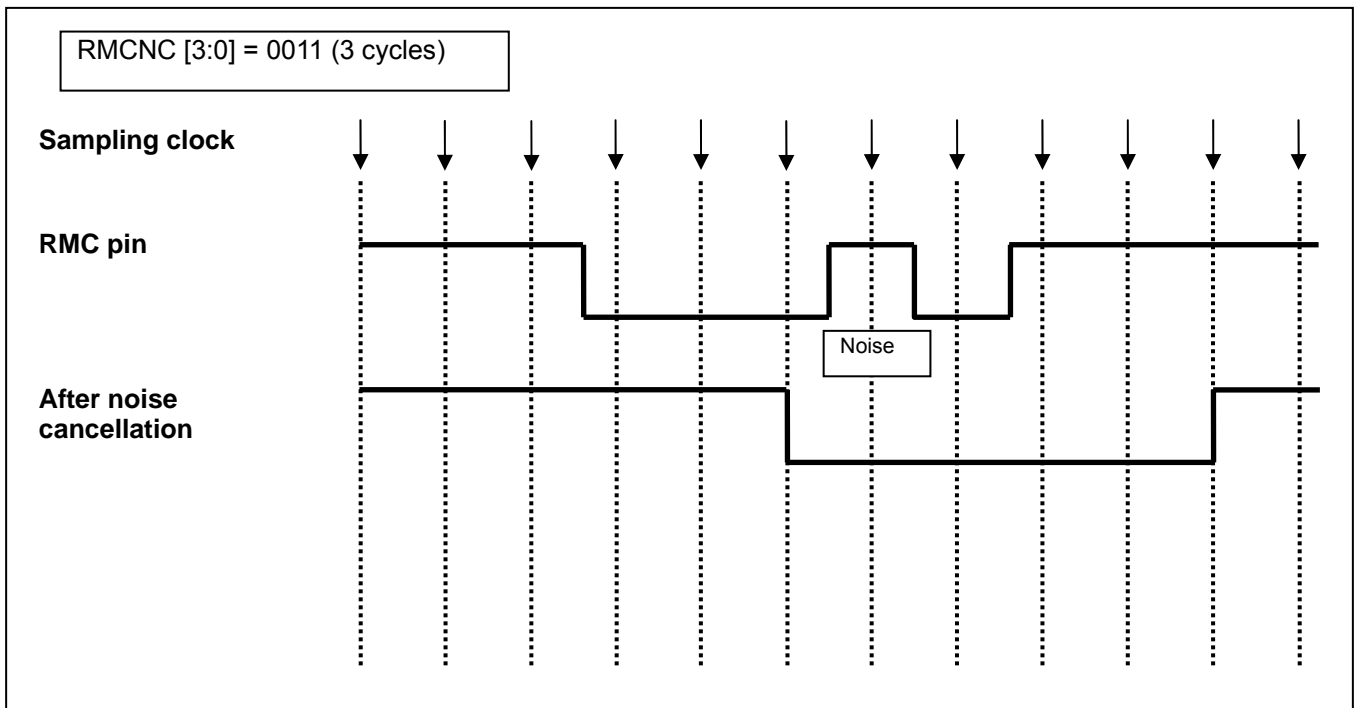
(2) Setting of Detecting Leader

To detect a leader, configure cycle and low width of the leader with the RMCRCR1 <RMCLLMIN[7:0]> <RMCLLMAX[7:0]> <RMCLCMIN[7:0]> <RMCLCMAX[7:0]> bits. When you configure the register, you must follow the rule shown below.

Leader	Rules
Low width + High width	<RMCLCMAX[7:0]> > <RMCLCMIN[7:0]> <RMCLLMAX[7:0]> > <RMCLLMIN[7:0]>
Only with high width	<RMCLCMIN[7:0]> > <RMCLLMAX[7:0]> <RMCLCMAX[7:0]> > <RMCLCMIN[7:0]> <RMCLLMAX[7:0]> = 0x00000000 <RMCLLMIN[7:0]> = don't care
No leader	<RMCLCMAX[7:0]> = 0x00000000 <RMCLCMIN[7:0]> = don't care <RMCLLMAX[7:0]> = don't care <RMCLLMIN[7:0]> = don't care

The following shows a leader waveform and the RMCRCR1 register settings.
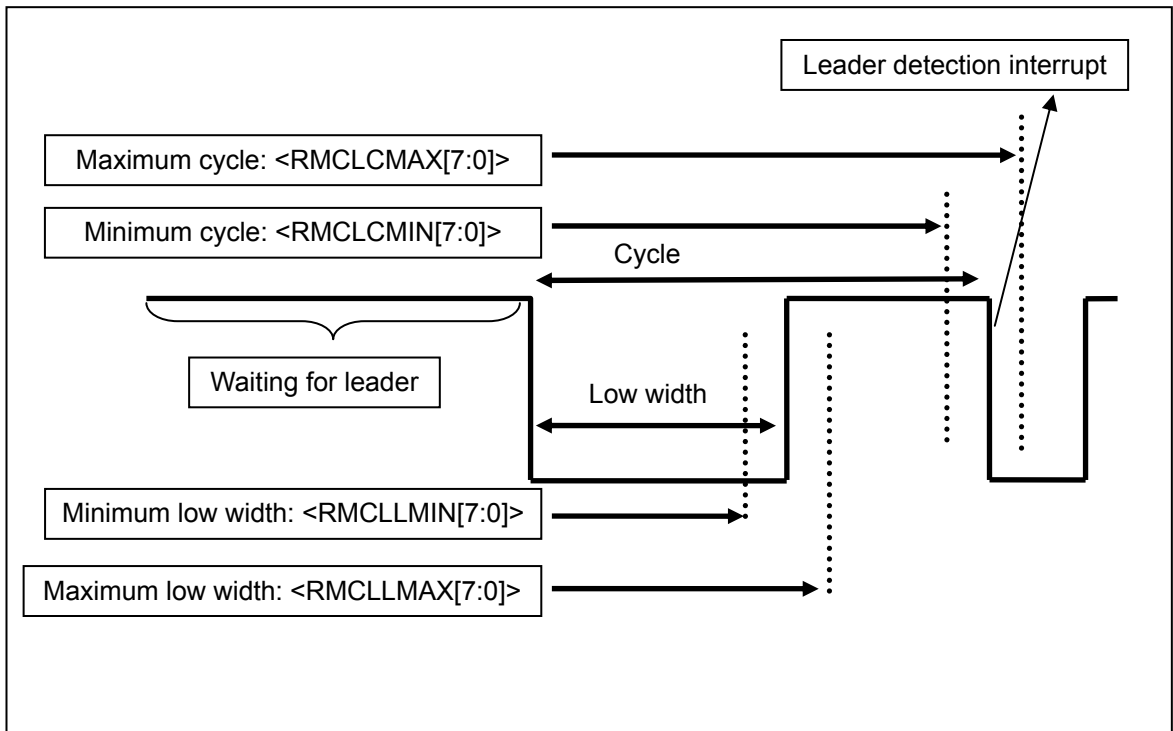
If you want to generate an interrupt when detecting a leader, configure the RMCRCR2 <RMCLIEN> bit. A remote control signal without a leader cannot generate a leader detection interrupt.

(3) Setting of Data Bit Determination

Based on a falling edge cycle, the data bit is determined as 0 or 1.
Configure a threshold of the determination with the RMCRCR3 <RMCDATL[6:0]> bit. If the cycle exceeds the threshold, the bit is determined as "1". If not, the bit is determined as "0".
By setting "1" to the RMCRCR2 <RMCEDIEN> bit, a remote control signal input falling edge interrupt can be generated in each falling edge of the data bit. Using this interrupt together with a 16-bit timer enables the determination to be done by software.
The following shows how the data bit is determined as "0" or "1".

Example: Threshold of 0/1 determination is set to 2.5T with the <RMCDATL[6:0]> bit.
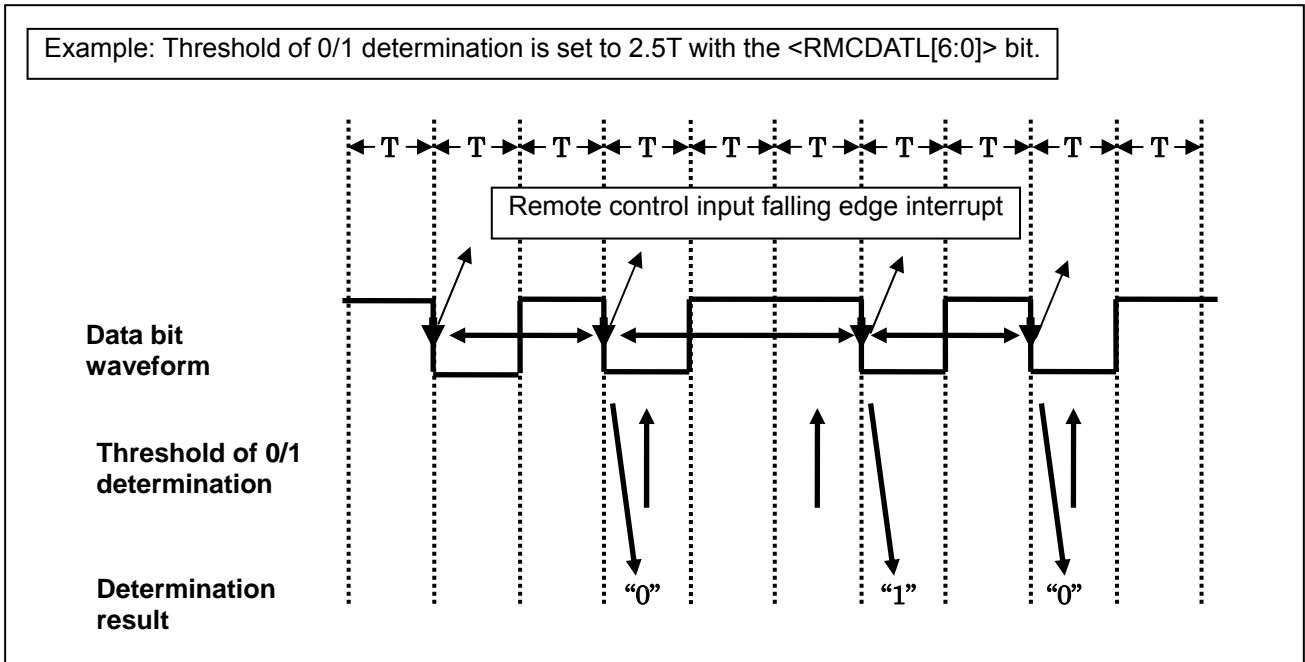
As for data bit determination of a remote control signal in a phase method, see 3.13.3.10 "Receiving a Remote Control Signal in a Phase Method".

(4) Setting of Reception Completion

To complete data reception, settings of detecting the maximum data bit cycle and excess low width are required. If multiple factors are specified, reception is completed by the factor detected first. Make sure to configure the reception completion settings.

1) Completed by a maximum data bit cycle

To complete reception by detecting a maximum data bit cycle, you need to configure the RMCRCR2 <RMCDMAX[7:0]> bits. If the falling edge of the data bit cycle isn't monitored after time specified as threshold in the <RMCDMAX[7:0]> bits, a maximum data bit cycle is detected. The detection completes reception and generates an interrupt.

As specified the <RMCEND1[6:0]>, <RMCEND2[6:0]> and <RMCEND3[6:0]> of RMCEND[3:1] registers, to be able to complete reception is by specified the number of receive data bit. In this case, only the number of bit that being generated at maximum data bit cycle received and when the number of receptions bit specified by the <RMCEND1[6:0]>, <RMCEND2[6:0]> and <RMCEND3[6:0]> of RMCxEND[3:1] register is corresponding, the maximum data bit cycle interrupt is generated. As specified to the <RMCEND1[6:0]>, <RMCEND2[6:0]> and <RMCEND3[6:0]> of RMCxEND[3:1] registers, it is able to set three kinds of the receive data bit.

When not corresponding to the number of receptions bit specified to the <RMCEND1[6:0]>, <RMCEND2[6:0]> and <RMCEND3[6:0]> of RMCxEND[3:1] registers and the number of bit that being generated at maximum data bit cycle received, it's still be leader or ready for reception.

2) Completed by excess low width

To complete reception by detecting the low width, you need to configure the RMCRCR2 <RMCLL[7:0]> bits. After the falling edge of the data bit is detected, if the signal stays low longer than specified, excess low width is detected. The detection completes reception and generates an interrupt.

3.13.3.1.1 Enabling Reception

By enabling the RMCREN <RMCREN> bit after configuring the RMCRCR1, RMCRCR2, RMCRCR3 and RMCRCR4 registers, RMC is ready for reception. Detecting a leader initiates reception.

(Note)	Changing the configurations of the RMCRCR1, RMCRCR2, RMCRCR3 and RMCRCR4 registers during reception may harm their proper operation. Be careful if you change them during reception.

3.13.3.1.2 Reception

Detecting a leader sets the RMCRSTAT <RMCRLDR> bit. Simultaneously, a leader detection interrupt is generated if the RMCRCR2 <RMCLIEN> bit is set. When the interrupt is generated, the RMCRSTAT <RMCRLIF> bit is set.

Next to the leader detection, each data bit is determined as 0 or 1. The results are stored in the RMCRBUF1, RMCRBUF 2 and RMCRBUF 3 registers up to 72bit. By setting "1" to the RMCRCR2 <RMCEDIEN> bit, a remote control signal input falling edge interrupt can be generated in each falling edge of the data bit. When the interrupt is generated, the RMCRSTAT <RMCEDIF> bit is set.

Detecting the maximum data bit cycle or the excess low width completes reception and generates an interrupt. In case of when the number of receptions bit specified by the <RMCEND1[6:0]>, <RMCEND2[6:0]> and <RMCEND3[6:0]> of RMCxEND[3:1] registers, the completion of reception and the interrupt is generated but only measured up the number of received bit until the detecting maximum data bit cycle.

To check the status of RMC after reception is completed, read the Remote Control Receive Status Register.

On completion of reception, RMC is waiting for the next leader.

By setting RMC to receive a signal without a leader, RMC recognizes the received is data and starts reception without detecting a leader.

If the next data reception is completed before reading the preceding received data, the preceding data is overwritten by the next one.

3.13.3.1.3 Reception Completion

1) Completed by a maximum data bit cycle

Detecting a maximum data bit cycle completes reception and generates an interrupt. After the interrupt is generated, the RMCRSTAT <RMCDMAXIF> bit is set to "1".

Maximum data bit cycle interrupt

Threshold: <RMCDMAX[7:0]>
If the falling edge of the data bit cycle isn't monitored after time specified as threshold, a maximum data bit cycle is detected.
The detection completes reception and generates an interrupt.
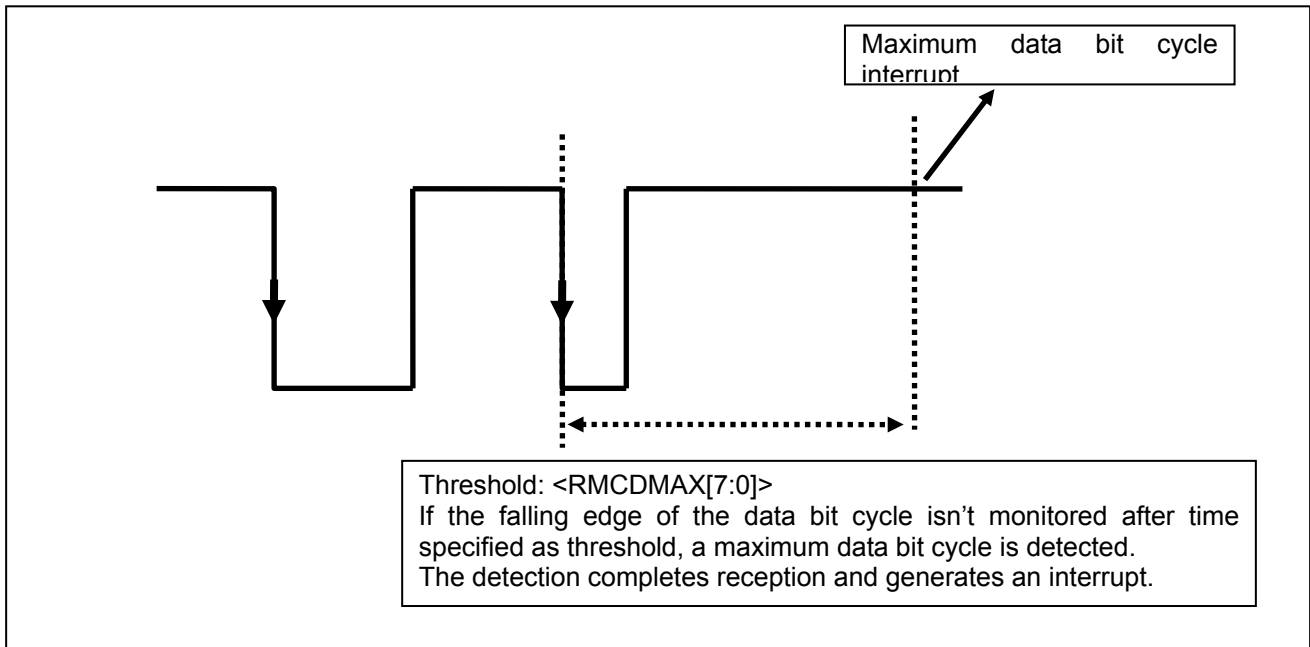
2) Completed by excess low width

Detecting excess low width completes reception and generates an interrupt. After the interrupt is generated, the RMCRSTAT <RMCLOIF> bit is set to "1".

Excess low width detection interrupt

Threshold: < RMCLL[7:0]>
Excess low width is detected when signal stay low longer than specified.
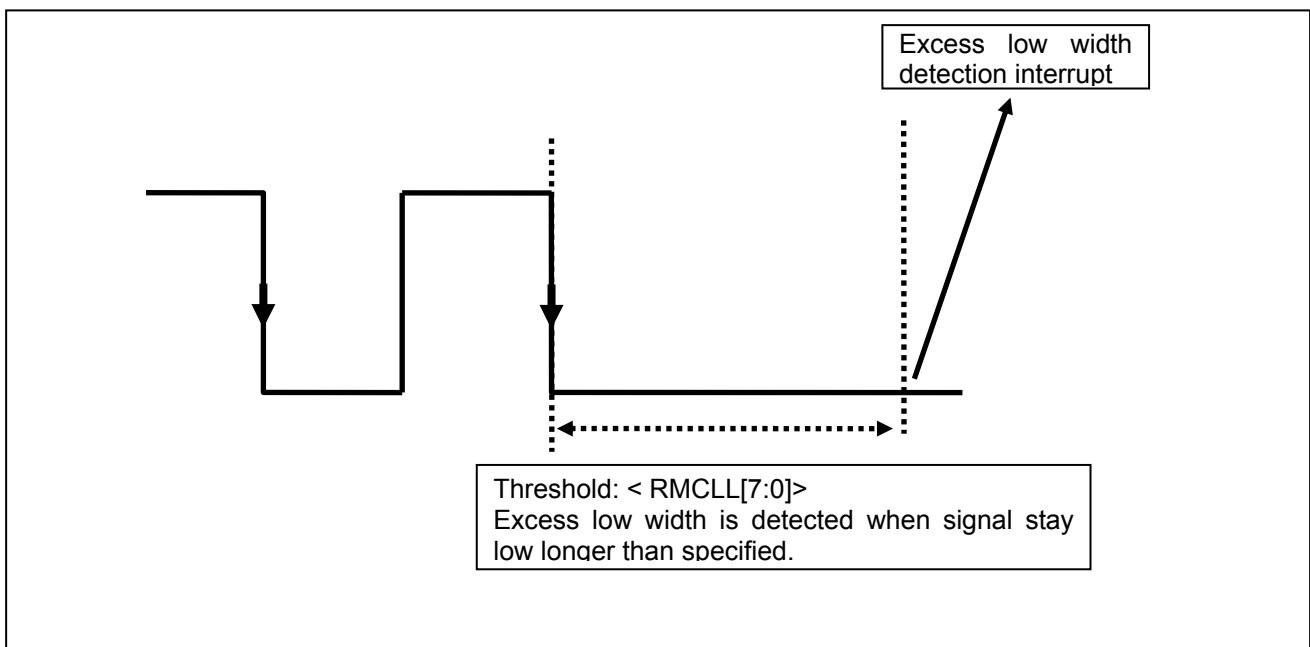
RMC keeps receiving 73 bit or more data unless reception is completed by detecting the maximum data bit cycle or the excess low width. If so, the received data in the data buffer may not be correct.

To check the status of RMC after reception is completed, read the Remote Control Receive Status Register. The status of RMC that each bit type indicates is shown below.

<RMCRLDR>	<RMCRNUM6:0>	RMC Status
0	0000001～1001000	Receiving remote control signal without a leader (Data bits: 1~72bit)
0	1001001～1111111	Receiving remote control signal without a leader (Data bits: 73bit and more)
1	0000000	Only with a leader
1	0000001～1001000	Receiving remote control signal with a leader (Data bits: 1~72bit)
1	1001001～1111111	Receiving remote control signal without a leader (Data bits: 73bit and more)

3.13.3.1.4 Stopping Reception

RMC stops reception by clearing the RMCREN <RMCREN> bit to "0" (reception disabled).
Clearing this bit during reception stops reception immediately and the received data is discarded.

3.13.3.1.5 Receiving Remote Control Signal without Leader

Setting RMCRCR2 <RMCLD> enables RMC to receive signals with or without a leader. By setting RMCRCR2 <RMCLD>, RMC starts receiving data if it recognizes a signal of which low width is shorter than a maximum low width of leader detection specified in the RMCRCR1 <RMCLLMAX[7:0]> bits. RMC keeps receiving data until the final data bit is received.

If RMCRCR2 <RMCLD> is enabled, the same settings of error detection, reception completion and data bit determination of 0 or 1 are applied regardless of whether a signal has a leader or not. Thus receivable remote control signals are limited.

RMCRCR2 register <RMCLD>=1

Waiting for leader

Leader waveform

A waveform of which low width is shorter than specified is determined as the beginning of data. RMC starts data reception.

Maximum data bit cycle is detected if a signal stays low shorter than specified and longer than a maximum data bit cycle.

Minimum low width <RMCLLMIN[7:0]>

Maximum data bit cycle < RMCDMAX[7:0]>
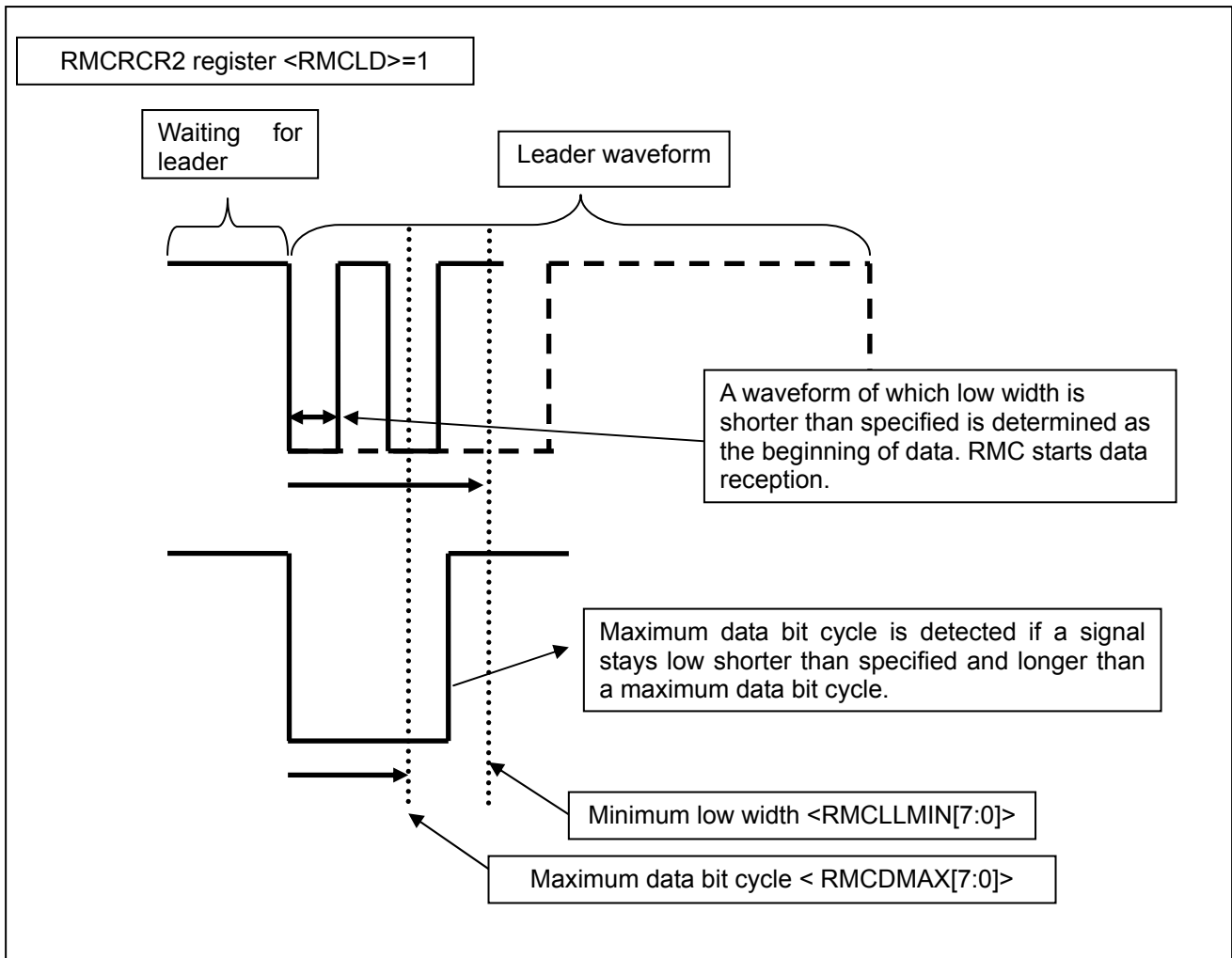
3.13.3.1.6 A Leader only with Low Width

The figure shown below illustrates a remote control signal that starts with a leader of which waveform only has low width. This signal starts with a leader that only has low width and a data bit cycle starts from the rising edge. To enable the signal, it must be sent after being reversed by setting the RMCRCR4 <RMCPO> bit to "1". This is because RMC is configured to detect a data bit cycle from the falling edge.

A leader is detected by the low width. When you configure the RMCRCR1 register, you must follow the rule shown below.

<RMCLLMAX[7:0]> = 0x00000000
<RMCLHMAX[7:0]> > <RMCLHMIN[7:0]>

If the rules are applied, RMC does not care about the value of <RMCLLMIN[7:0]>.

To determine the data bit as 0 or 1, configure a threshold of the determination with the RMCRCR3 <RMCDATL[6:0]> bit.

Configure a maximum data bit cycle with the <RMCDMAX7:0> bits of the Remote Control Receive Control Register 2.

To complete reception by detecting the maximum data bit cycle, you need to configure the RMCRCR2 <RMCDMAX[7:0]> bits. To complete reception by detecting the low width, you need to configure the RMCRCR2 <RMCLL[7:0]> bits. Detecting the maximum data bit cycle or the excess low width completes reception and generates an interrupt. RMC waits for the next leader.
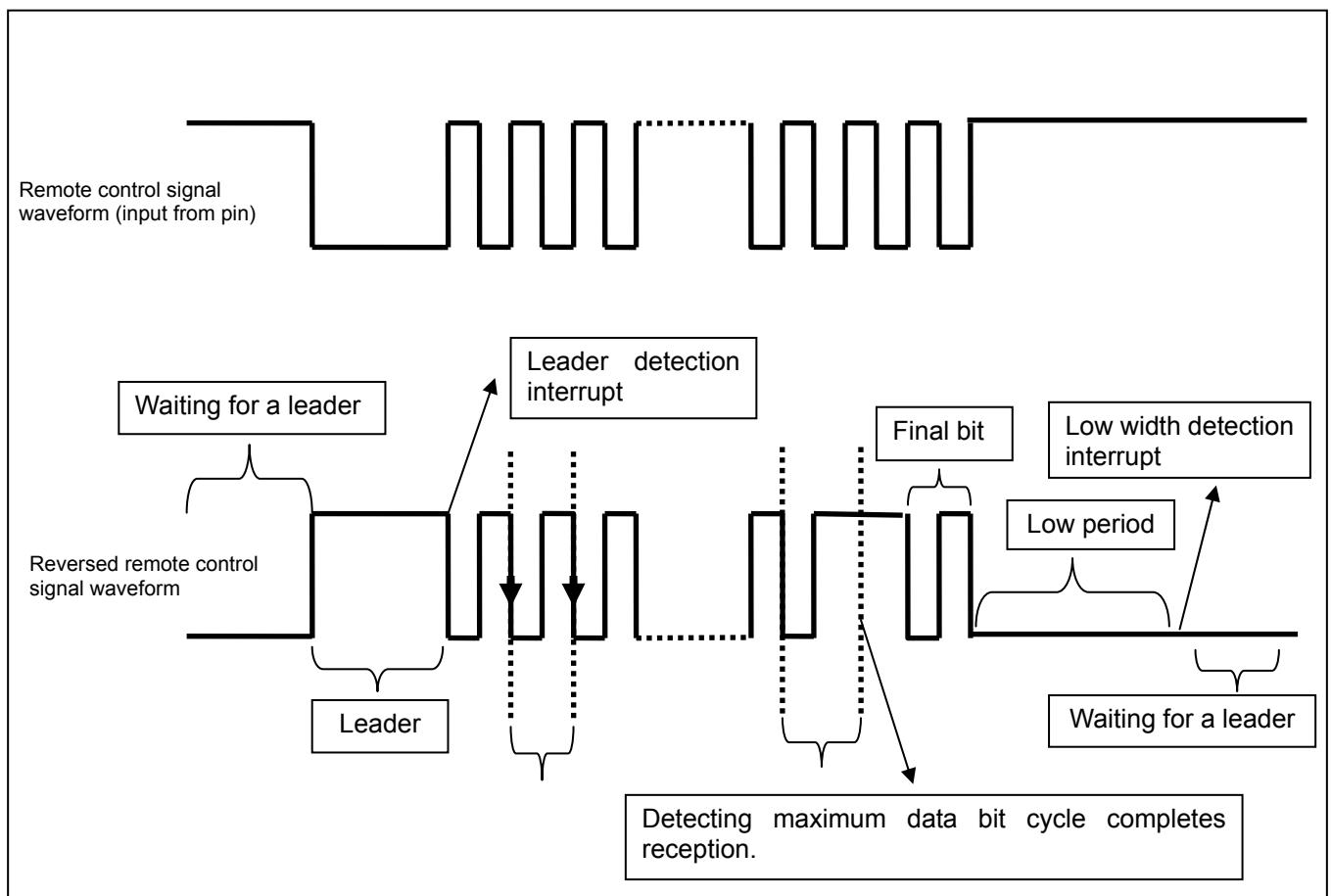
3.13.3.1.7 Receiving a Remote Control Signal in a Phase Method

   RMC is capable of receiving a remote control signal in a phase method of which signal cycle is fixed. A signal in the phase method has three waveform patterns (see the figure shown below). By setting two thresholds a remote control signal pattern is determined. RMC converts the signal into data "0" or "1". On completion of reception, received data "0" and "1" are stored in the RMCRBUF1, RMCRBUF 2 and RMCRBUF3.
   By setting RMCRCR2<RMCPHM>="1", RMC enables to receive a remote control signal in the phase method. Each threshold can be configured with the RMCRCR4 <RMCDATL[6:0]> bits and <RMCDATH[6:0]> bits. Two thresholds are used to distinguish three waveform patterns. On condition that a cycle between two falling edges is "T", three patterns show cycles of 1T, 1.5T and 2T. Details of the two thresholds are shown below.
   To determine a remote control signal in the phase method, three patterns of data waveform and preceding data are required. In addition, the signal needs to start from data "1".

	Determined by	Threshold	Register bits to set
Threshold 1	Pattern 1 & pattern 2	1T~1.5T	RMCRCR2 register <RMCDATL[6:0]>
Threshold 2	Pattern 2 & pattern 3	1.5T~2T	RMCRCR2 register <RMCDATH[6:0]>

Waveform pattern in phase method

Cycle T

Pattern 1

Pattern 2

Pattern 3

Remote control signal data in phase method

Data "1"   Data "0"

A pulse shape in a cycle indicates whether it is data "0" or data "1".

Remote control signal in phase method

Remote control signal

The first two bits of data need to be "11".
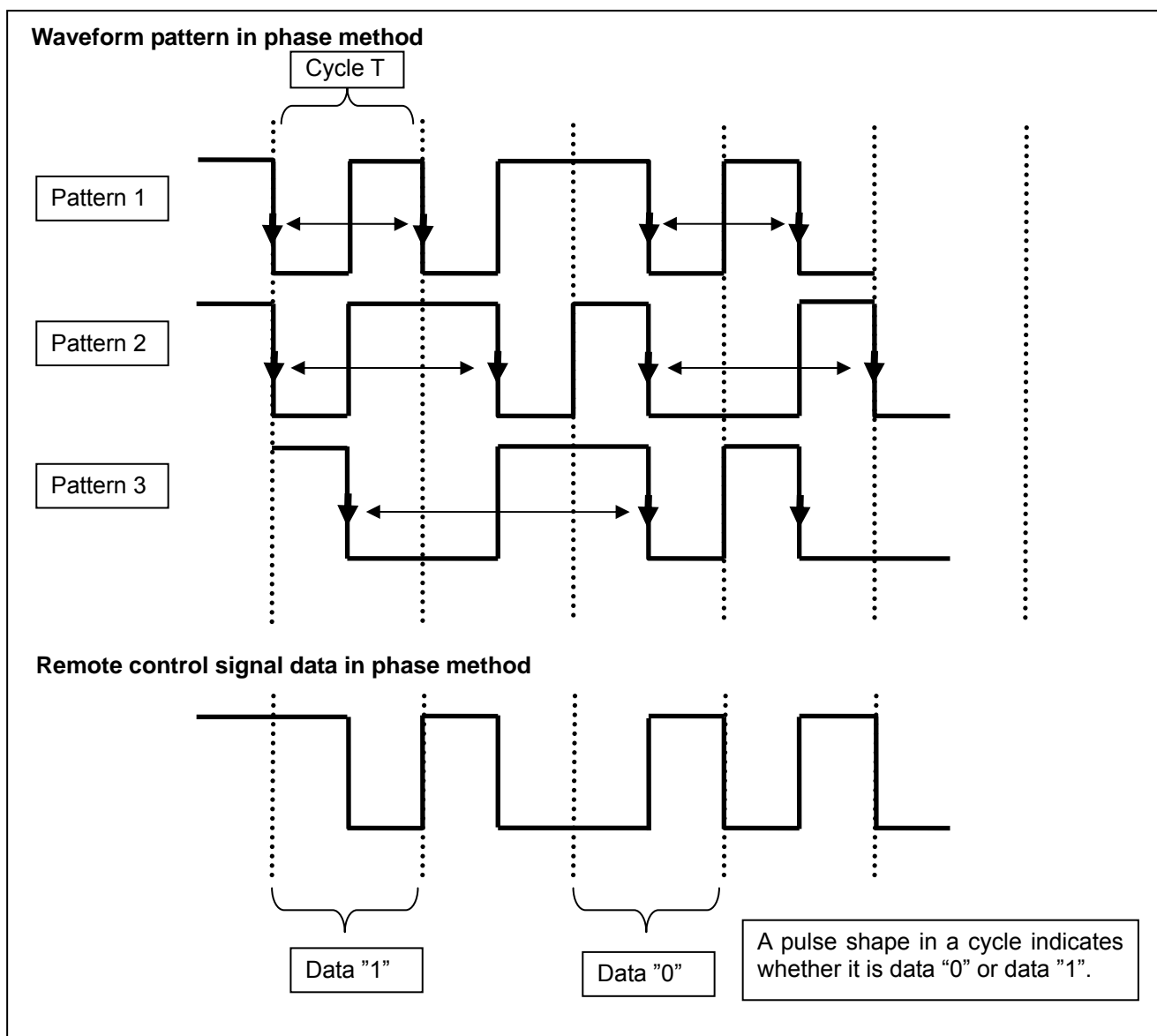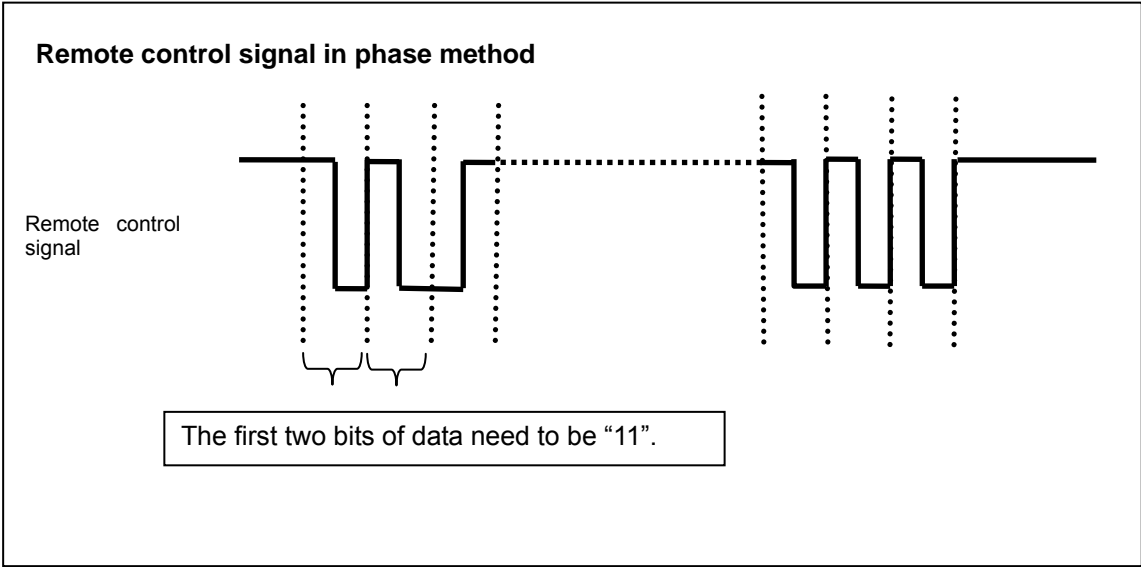
3.14 USB Host Controller

The USB Host Controller (USBHC) is compliant with the USB Specification Revision 2.0 and the Open HCI Specification Release 1.0a, and supports USB transfers at 12 Mbps (full-speed). The USBHC is connected to the CPU via bus bridge logic.

The USBHC is subject to some restrictions. For details, see section 3.14.8.

3.14.1 System Overview

The key features of the USBHC are as follows:

(1) Supports full-speed (12 Mbps) USB devices. Low-speed (1.5 Mbps) USB devices are not supported.

(2) Supports control, bulk, interrupt and isochronous transfers.

(There are some restrictions. Refer to section 3.14.8)

(3) Contains two 16-byte FIFO buffers (IN and OUT) in the bus bridge logic for connecting with the CPU, allowing up to 16 bytes of burst transfers.

(4) Supports data transfers between the FIFO buffers in the bus bridge logic and the on-chip SRAM. Accessible SRAM is RAM0(0x2000_0000 to 0x2000_1FFF) and RAM1(0x2000_2000 to 0x2000_3FFF).

(Note)If the CPU/DMA accesses the RAM0 or the RAM1 when the USBHC is already

accessing it, the CPU/DMA has a higher priority over the USBHC for access.

Refer to 3.14.7.5 for details.

3.14.2 System Configuration

The USBHC consists of the following three blocks:

(1) USBHC core (OHCI)

(2) USB transceiver

(3) CPU bus bridge logic
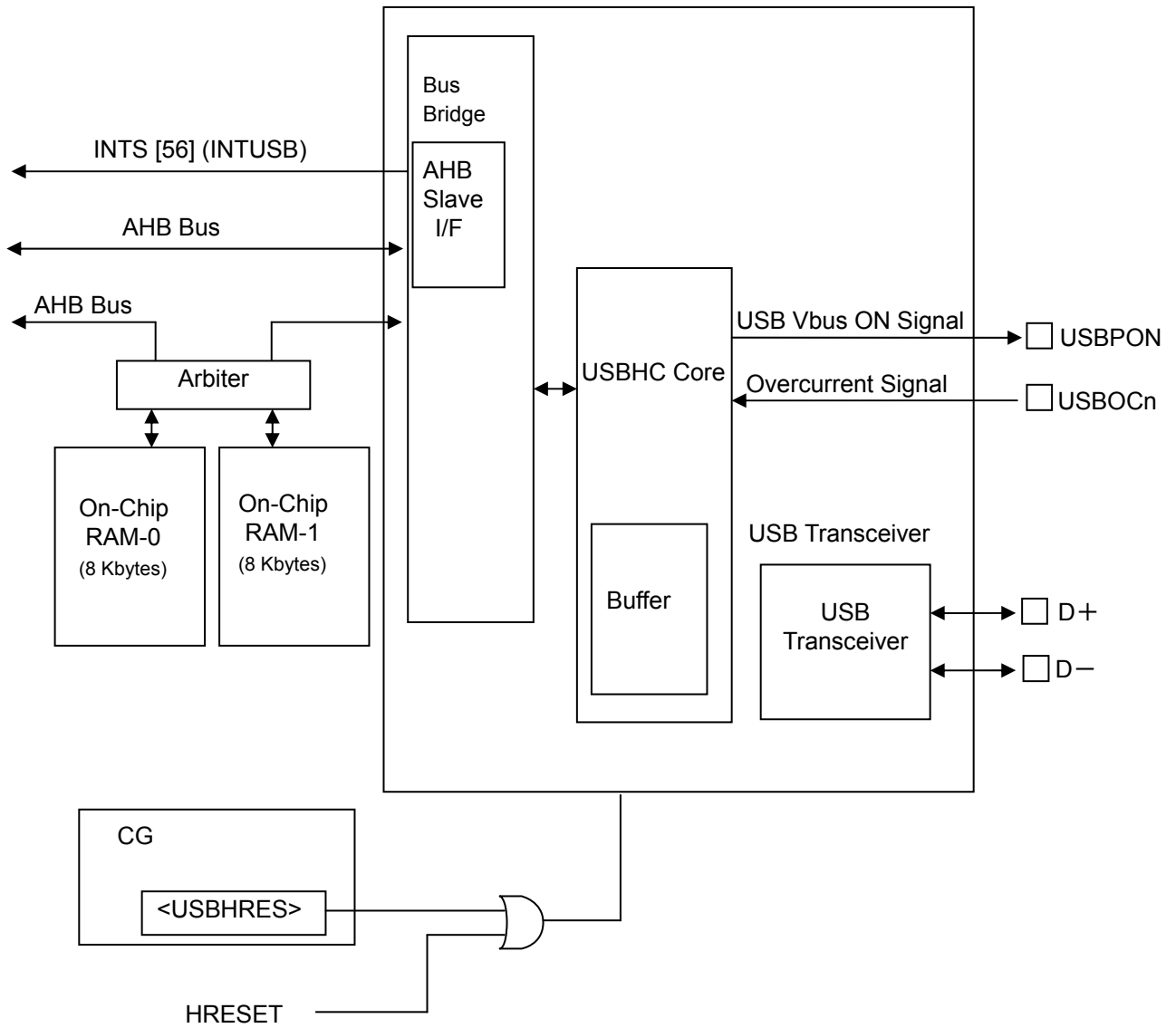
Figure 3.14.1   USB Host Controller

3.14.3 Interrupts

The USBHC generates the following interrupts:

- Scheduling Overrun
- HcDoneHead Write back
- Start of Frame
- Resume Detect
- Unrecoverable Error
- Frame Number Overflow
- Root Hub Status Change
- Ownership Change

When an event that causes an interrupt occurs, the USBHC sets the corresponding bit in the HcInterruptStatus register. At this time, if the MasterInterruptEnable bit is enabled and the corresponding bit in the HcInterruptEnable register is enabled, a USB interrupt (INTSUB) is generated.

The USBHS driver software can clear each bit in the HcInterruptStatus register by writing a 1 to it. (The driver software cannot set these bits, and the USBHC cannot clear these bits.)

3.14.4 Reset

The USBHC is initialized by a hardware or software reset.

3.14.4.1 Hardware Reset

A hardware reset is generated via an external reset pin or internally (WDT reset, USBHC reset or PCM reset).

- All registers are initialized.
- The reset signal is output on the USB bus by an external pull-down resistor ($D+=D-=0$). (The USB transceiver is in the SUSPEND state.)
- The USB state changes to the USBRESET state.
- List processing and SOF token generation are disabled.
- The FrameNumber field of the HcFmNumber register is not incremented.

3.14.4.2 Software Reset

A software reset is generated when a 1 is written to the HostControllerReset bit in the HcCommandStatus register.

- All OHCI registers are initialized except the following:
  - The RemoteWakeupConnected and InterruptRouting bits in the HcControl register remain the same.
  - The HcBCR0 register is not initialized.
- The USBHC outputs the reset signal on the USB bus ($D+=D-=0$).
- The USB state changes to the USBSUSPEND state.

(The FunctionalState bit in the HcControl register is set to 0x03 to transition to the USBSUSPEND state.)

3.14.5 Bus Power Control

The USBHC has a control signal for the external Vbus power IC. This signal is controlled by the USBPON (PG6) pin.

To use PG6 as the USBPON pin, the port G control register (PGFR) must be set appropriately. Then, setting the LPSC bit in the HcRhStatus register to 1 makes the USBPON pin to output high level.

The USBOC (PG7) pin is used to detect overcurrent conditions. When low level is detected on this pin, the USBHC sets the OCI bit in the HcRhStatus register to 1. (To use PG7 as the USBOC pin, the port G control register (PGFR) must be set appropriately.)

3.14.6 Registers

The USBHC contains a set of control registers compliant with the Open HCI Specification which are mapped into the memory space. The bus bridge logic for connecting with the CPU also includes control registers. These registers are directly accessible from the CPU via a 32-bit bus.

Table 3.14.1 Registers Compliant with the USB Open HCI Specification

Base address = 0x4000_3000

Register Name	Address (base+)	Description
HcRevision	0x0000	Hc Revision Register
HcControl	0x0004	Hc Control Register
HcCommandStatus	0x0008	Hc Command Status Register
HcInterruptStatus	0x000C	Hc Interrupt Status Register
HcInterruptEnable	0x0010	Hc Interrupt Enable Register
HcInterruptDisable	0x0014	Hc Interrupt Disable Register
HcHCCA	0x0018	Hc Host Controller Communication Area Register
HcPeriodCurrentED	0x001C	Hc Period Current Endpoint Descriptor Register
HcControlHeadED	0x0020	Hc Control Head Endpoint Descriptor Register
HcControlCurrentED	0x0024	Hc Control Current ndpoint Descriptor Register
HcBulkHeadED	0x0028	Hc Bulk Head Endpoint Descriptor Register
HcBulkCurrentED	0x002C	Hc Bulk Current Endpoint Descriptor Register
HcDoneHead	0x0030	Hc Done Head Register
HcFmInterval	0x0034	Hc Frame Interval Register
HcFmRemaining	0x0038	Hc Frame Remaining Register
HcFmNumber	0x003C	Hc Frame Number Register
HcPeriodStart	0x0040	Hc Period Start Register
HcLSThreshold	0x0044	Hc Low Speed Threshold Register
HcRhDescriptorA	0x0048	Hc Root hub Descriptor A Register
HcRhDescripterB	0x004C	Hc Root hub Descriptor B Register
HcRhStatus	0x0050	Hc Root hub Status Register
HcRhPortStatus1	0x0054	Hc Root hub Port Status Register
HcBCR0	0x0080	Hc BCR0 Resister

Note 1: The address listed in Table 3.14.1 are those mapped on the CPU registers.

Note 2: The Open HCI Specification Release 1.0a specifies the FrameRemaining (FR) and FrameRemainingToggle (FRT) bits in the HcFmRemaining register and the FramNumber (FN) bit in the HcFmNumber register as read-only to the Host Control Driver (HDC). However, the USBHC allows write accesses to these registers by the HCD for debug purposes. If the HCD writes to these registers, undefined results will occur. These bits must not be written by the HCD.

3.14.6.1 HcRevision Register

Address = (0x4000_3000) + (0x0000)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol								Reserved								
Read/Write (HCD)																
Read/Write (HC)																
Reset Value																
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol				Reserved								Rev				
Read/Write (HCD)												R				
Read/Write (HC)												R				
Reset Value									0	0	0	1	0	0	0	0

Bit	Mnemonic	Field name	Description
[31:8]	–	Reserved	
[7:0]	REV	Revision	This read-only field contains the BCD representation of the version of the HCI specification that is implemented by this HC. For example, a value of 0x11 corresponds to version 1.1. All of the HC implementations that are compliant with this specification will have a value of 0x10.

3.14.6.2 HcControl Register

The HcControl register defines the operating modes for the Host Controller. Most of the fields in this register are modified only by the Host Controller Driver, except HostControllerFunctionalState and RemoteWakeupConnected.

Address = (0x4000_3000) + (0x0004)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	Reserved															
Read/Write (HCD)																
Read/Write (HC)																
Reset Value																
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	Reserved					RWE	RWC	IR	HCFS		BLE	CLE	IE	PLE	CBSR	
Read/Write (HCD)						R/W										
Read/Write (HC)						R	R/W	R	R/W		R	R	R	R	R	
Reset Value						0	0	0	0	0	0	0	0	0	0	0

Bit	Mnemonic	Field name	Description
[31:11]		Reserved	
[10]	RWE	RemoteWakeup Enable	This bit is used by HCD to enable or disable the remote wakeup feature upon the detection of upstream resume signaling. When this bit is set and the ResumeDetected bit in HcInterruptStatus is set, a remote wakeup is signaled to the host system. Setting this bit has no impact on the generation of hardware interrupt.
[9]	RWC	RemoteWakeup Connected	This bit indicates whether HC supports remote wakeup signaling. If remote wakeup is supported and used by the system, it is the responsibility of system firmware to set this bit during POST. HC clears the bit upon a hardware reset but does not alter it upon a software reset.
[8]	IR	Interrupt Routing	This bit determines the routing of interrupts generated by events registered in HcInterruptStatus. If clear, all interrupts are routed to the normal host bus interrupt mechanism. If set, interrupts are routed to the System Management Interrupt. HCD clears this bit upon a hardware reset, but it does not alter this bit upon a software reset. HCD uses this bit as a tag to indicate the ownership of HC.

Bit	Mnemonic	Field name	Description
[7:6]	HCFS	HostController FunctionalState ForUSB	0y00: USBRESET 0y01: USBRESUME 0y10: USBOPERATIONAL 0y11: USBSUSPEND A transition to USBOPERATIONAL from another state causes SOF generation to begin 1 ms later. HCD may determine whether HC has begun sending SOFs by reading the StartofFrame field of HcInterruptStatus. This field may be changed by HC only when in the USBSUSPEND state. HC may move from the USBSUSPEND state to the USBRESUME state after detecting the resume signaling from a downstream port. HC enters USBSUSPEND after a software reset, whereas it enters USBRESET after a hardware reset. The latter also resets the Root Hub and asserts subsequent reset signaling to downstream ports.
[5]	BLE	BulkListEnable	This bit is set to enable the processing of the Bulk list in the next Frame. If cleared by HCD, processing of the Bulk list does not occur after the next SOF. HC checks this bit whenever it determines to process the list. When disabled, HCD may modify the list. If HcBulkCurrentED is pointing to an ED to be removed, HCD must advance the pointer by updating HcBulkCurrentED before re-enabling processing of the list.
[4]	CLE	ControlList Enable	This bit is set to enable the processing of the Control list in the next Frame. If cleared by HCD, processing of the Control list does not occur after the next SOF. HC must check this bit whenever it determines to process the list. When disabled, HCD may modify the list. If HcControlCurrentED is pointing to an ED to be removed, HCD must advance the pointer by updating HcControlCurrentED before re-enabling processing of the list.
[3]	IE	Isochronous Enable	This bit is used by HCD to enable/disable processing of isochronous EDs. While processing the periodic list in a Frame, HC checks the status of this bit when it finds an Isochronous ED (F=1). If set (enabled), HC continues processing the EDs. If cleared (disabled), HC halts processing of the periodic list (which now contains only isochronous EDs) and begins processing the Bulk/Control lists. Setting this bit is guaranteed to take effect in the next Frame (not the current Frame). * This product has some restrictions on isochronous transfers.
[2]	PLE	PeriodicList Enable	This bit is set to enable the processing of the periodic list in the next Frame. If cleared by HCD, processing of the periodic list does not occur after the next SOF. HC must check this bit before it starts processing the list.
[1:0]	CBSR	ControlBulk ServiceRatio	This specifies the service ratio between Control and Bulk EDs. Before processing any of the nonperiodic lists, HC must compare the ratio specified with its internal count on how many nonempty Control EDs have been processed, in determining whether to continue serving another Control ED or switching to Bulk EDs. The internal count will be retained when crossing the frame boundary. In case of reset, HCD is responsible for restoring this value. CBSR No. of Control EDs over Bulk EDs served 00 1:1 01 2:1 10 3:1 11 4:1

3.14.6.3 HcCommandStatus Register

The HcCommandStatus register is used by the Host Controller to receive commands issued by the Host Controller Driver, as well as reflecting the current status of the Host Controller. To the Host Controller Driver, it appears to be a "write to set" register. The Host Controller must ensure that bits written as '1' become set in the register while bits written as '0' remain unchanged in the register. The Host Controller Driver may issue multiple distinct commands to the Host Controller without concern for corrupting previously issued commands. The Host Controller Driver has normal read access to all bits.

The SchedulingOverrunCount field indicates the number of frames with which the Host Controller has detected the scheduling overrun error. This occurs when the Periodic list does not complete before EOF. When a scheduling overrun error is detected, the Host Controller increments the counter and sets the SchedulingOverrun field in the HcInterruptStatus register.

Address = (0x4000_3000) + (0x0008)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	Reserved														SOC	
Read/Write (HCD)															R	
Read/Write (HC)															R/W	
Reset Value															0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved												OCR	BLF	CLF	HCR
Read/Write (HCD)													R/W			
Read/Write (HC)													R/W			
Reset Value													0	0	0	0

Bit	Mnemonic	Field name	Description
[31:18]	–	Reserved	
[17:16]	SOC	Scheduling OverrunCount	These bits are incremented on each scheduling overrun error. It is initialized to 00b and wraps around at 11b. This will be incremented when a scheduling overrun is detected even if SchedulingOverrun in HcInterruptStatus has already been set. This is used by HCD to monitor any persistent scheduling problems.
[15:4]	–	Reserved	
[3]	OCR	Ownership ChangeRequest	This bit is set by an OS HCD to request a change of control of the HC. When set HC will set the OwnershipChange field in HcInterruptStatus. After the changeover, this bit is cleared and remains so until the next request from OS HCD.

Bit	Mnemonic	Field name	Description
[2]	BLF	BulkListFilled	This bit is used to indicate whether there are any TDs on the Bulk list. It is set by HCD whenever it adds a TD to an ED in the Bulk list. When HC begins to process the head of the Bulk list, it checks BF. As long as BulkListFilled is 0, HC will not start processing the Bulk list. If BulkListFilled is 1, HC will start processing the Bulk list and will set BF to 0. If HC finds a TD on the list, then HC will set BulkListFilled to 1 causing the Bulk list processing to continue. If no TD is found on the Bulk list, and if HCD does not set BulkListFilled, then BulkListFilled will still be 0 when HC completes processing the Bulk list and Bulk list processing will stop.
[1]	CLF	ControlListFilled	This bit is used to indicate whether there are any TDs on the Control list. It is set by HCD whenever it adds a TD to an ED in the Control list. When HC begins to process the head of the Control list, it checks CLF. As long as ControlListFilled is 0, HC will not start processing the Control list. If CF is 1, HC will start processing the Control list and will set ControlListFilled to 0. If HC finds a TD on the list, then HC will set ControlListFilled to 1 causing the Control list processing to continue. If no TD is found on the Control list, and if the HCD does not set ControlListFilled, then ControlListFilled will still be 0 when HC completes processing the Control list and Control list processing will stop.
[0]	HCR	HostController Reset	This bit is set by HCD to initiate a software reset of HC. Regardless of the functional state of HC, it moves to the USBSUSPEND state in which most of the operational registers are reset except those stated otherwise; e.g., the InterruptRouting field of HcControl, and no Host bus accesses are allowed. This bit is cleared by HC upon the completion of the reset operation. The reset operation must be completed within 10 μs. This bit, when set, should not cause a reset to the Root Hub and no subsequent reset signaling should be asserted to its downstream ports.

3.14.6.4 HcInterruptStatus Register

This register provides status on various events that cause hardware interrupts. When an event occurs, the Host Controller sets the corresponding bit in this register. When a bit becomes set, a hardware interrupt is generated if the interrupt is enabled in the HcInterruptEnable register and the MasterInterruptEnable bit is set. The Host Controller Driver may clear specific bits in this register by writing '1' to bit positions to be cleared. The Host Controller Driver may not set any of these bits. The Host Controller will never clear the bit.

Address = (0x4000_3000) + (0x000C)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	Reserved	OC	Reserved													
Read/Write (HCD)		R/W														
Read/Write (HC)		R/W														
Reset Value		0														

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved									RHSC	FNO	UE	RD	SF	WDH	SO
Read/Write (HCD)										R/W						
Read/Write (HC)										R/W						
Reset Value										0	0	0	0	0	0	0

Bit	Mnemonic	Field name	Description
[31]	–	Reserved	
[30]	OC	Ownership Change	This bit is set by HC when HCD sets the OwnershipChangeRequest field in HcCommandStatus. This event, when unmasked, will always generate an System Management Interrupt (SMI) immediately. This bit is tied to 0 when the SMI pin is not implemented.
[29:7]	–	Reserved	
[6]	RHSC	RootHubStatus Change	This bit is set when the content of HcRhStatus or the content of any of HcRhPortStatus[NumberofDownstreamPort] has changed.
[5]	FNO	FrameNumber Overflow	This bit is set when the MSb of HcFmNumber (bit 15) changes value, from 0 to 1 or from 1 to 0, and after HccaFrameNumber has been updated.
[4]	UE	Unrecoverable Error	This bit is set when HC detects a system error not related to USB. HC should not proceed with any processing nor signaling before the system error has been corrected. HCD clears this bit after HC has been reset.

Bit	Mnemonic	Field name	Description
[3]	RD	ResumeDetected	This bit is set when HC detects that a device on the USB is asserting resume signaling. It is the transition from no resume signaling to resume signaling causing this bit to be set. This bit is not set when HCD sets the USBRESUME state.
[2]	SF	StartofFrame	This bit is set by HC at each start of a frame and after the update of HccaFrameNumber. HC also generates a SOF token at the same time.
[1]	WDH	WritebackDone Head	This bit is set immediately after HC has written HcDoneHead to HccaDoneHead. Further updates of the HccaDoneHead will not occur until this bit has been cleared. HCD should only clear this bit after it has saved the content of HccaDoneHead.
[0]	SO	Scheduling Overrun	This bit is set when the USB schedule for the current Frame overruns and after the update of HccaFrameNumber. A scheduling overrun will also cause the SchedulingOverrunCount of HcCommandStatus to be incremented.

3.14.6.5 HcInterruptEnable Register

Each enable bit in the HcInterruptEnable register corresponds to an associated interrupt bit in the HcInterruptStatus register. The HcInterruptEnable register is used to control which events generate a hardware interrupt. When a bit is set in the HcInterruptStatus register AND the corresponding bit in the HcInterruptEnable register is set AND the MasterInterruptEnable bit is set, then a hardware interrupt is requested on the host bus.

Writing a '1' to a bit in this register sets the corresponding bit, whereas writing a '0' to a bit in this register leaves the corresponding bit unchanged. On read, the current value of this register is returned.

Address = (0x4000_3000) + (0x0010)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	MIE	OC	Reserved													
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
Reset Value	0	0														

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	Reserved									RHSC	FNO	UE	RD	SF	WDH	SO
Read/Write (HCD)										R/W						
Read/Write (HC)										R						
Reset Value										0	0	0	0	0	0	0

Bit	Mnemonic	Field name	Description
[31]	MIE	MasterInterrupt Enable	A '0' written to this field is ignored by HC. A '1' written to this field enables interrupt generation due to events specified in the other bits of this register. This is used by HCD as a Master Interrupt Enable.
[30]	OC	Ownership Change	0y0: Ignore 0y1: Enable interrupt generation due to Ownership Change.
[29:7]	−	Reserved	
[6]	RHSC	RootHubStatus Change	0y0: Ignore 0y1: Enable interrupt generation due to Root Hub Status Change.
[5]	FNO	FrameNumber Overflow	0y0: Ignore 0y1: Enable interrupt generation due to Frame Number Overflow .
[4]	UE	Unrecoverable Error	0y0: Ignore 0y1: Enable interrupt generation due to Unrecoverable Error .
[3]	RD	ResumeDetected	0y0: Ignore 0y1: Enable interrupt generation due to Resume Detected.
[2]	SF	StartofFrame	0y0: Ignore 0y1: Enable interrupt generation due to Start of Frame.
[1]	WDH	WritebackDone Head	0y0: Ignore 0y1: Enable interrupt generation due to HcDoneHeadWriteback.
[0]	SO	Scheduling Overrun	0y0: Ignore 0y1: Enable interrupt generation due to Scheduling Overrun.

3.14.6.6 HcInterruptDisable Register

Each disable bit in the HcInterruptDisable register corresponds to an associated interrupt bit in the HcInterruptStatus register. The HcInterruptDisable register is coupled with the HcInterruptEnable register. Thus, writing a '1' to a bit in this register clears the corresponding bit in the HcInterruptEnable register, whereas writing a '0' to a bit in this register leaves the corresponding bit in the HcInterruptEnable register unchanged. On read, the current value of the HcInterruptEnable register is returned.

Address = (0x4000_3000) + (0x0014)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	MIE	OC	Reserved													
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
Reset Value	0	0														
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved									RHSC	FNO	UE	RD	SF	WDH	SO
Read/Write (HCD)										R/W						
Read/Write (HC)										R						
Reset Value										0	0	0	0	0	0	0

Bit	Mnemonic	Field name	Function
[31]	MIE	MasterInterrupt Enable	A '0' written to this field is ignored by HC. A '1' written to this field disables interrupt generation due to events specified in the other bits of this register. This field is set after a hardware or software reset.
[30]	OC	Ownership Change	0y0: Ignore 0y1: Disable interrupt generation due to Ownership Change.
[29:7]		Reserved	
[6]	RHSC	RootHubStatus Change	0y0: Ignore 0y1: Disable interrupt generation due to Root Hub Status Change.
[5]	FNO	FrameNumber Overflow	0y0: Ignore 0y1: Disable interrupt generation due to Frame Number Overflow.
[4]	UE	Unrecoverable Error	0y0: Ignore 0y1: Disable interrupt generation due to Unrecoverable Error.
[3]	RD	ResumeDetected	0y0: Ignore 0y1: Disable interrupt generation due to Resume Detected.
[2]	SF	StartofFrame	0y0: Ignore 0y1: Disable interrupt generation due to Start of Frame.
[1]	WDH	WritebackDone Head	0y0: Ignore 0y1: Disable interrupt generation due to HcDoneHeadWriteback.
[0]	SO	Scheduling Overrun	0y0: Ignore 0y1: Disable interrupt generation due to SchedulingOverrun.

3.14.6.7 HcHCCA Register

The HcHCCA register contains the physical address of the Host Controller Communication Area. The Host Controller Driver determines the alignment restrictions by writing all 1s to HcHCCA and reading the content of HcHCCA. The alignment is evaluated by examining the number of zeroes in the lower order bits. The minimum alignment is 256 bytes; therefore, bits 0 through 7 must always return '0' when read. This area is used to hold the control structures and the Interrupt table that are accessed by both the Host Controller and the Host Controller Driver.

Address = (0x4000_3000) + (0x0018)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	HCCA															
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	HCCA								Reserved							
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
Reset Value	0	0	0	0	0	0	0	0								

Bit	Mnemonic	Field name	Description
[31:8]	HCCA	HostController Communication Area	This is the base address of the Host Controller Communication Area.
[7:0]	−	Reserved	

3.14.6.8 HcPeriodCurrentED Register

The HcPeriodCurrentED register contains the physical address of the current Isochronous or Interrupt Endpoint Descriptor.

Address = (0x4000_3000) + (0x001C)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	PCED															
Read/Write (HCD)	R															
Read/Write (HC)	R/W															
After Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	PCED												Reserved			
Read/Write (HCD)	R															
Read/Write (HC)	R/W															
After Reset	0	0	0	0	0	0	0	0	0	0	0	0				

Bit	Mnemonic	Field name	Description
[31:4]	PCED	PeriodCurrentED	This is used by HC to point to the head of one of the Periodic lists which will be processed in the current Frame. The content of this register is updated by HC after a periodic ED has been processed.
[3:0]	–	Reserved	

3.14.6.9 HcControlHeadED Register

The HcControlHeadED register contains the physical address of the first Endpoint Descriptor of the Control list.

Address = (0x4000_3000) + (0x0020)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	CHED															
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
After Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	CHED												Reserved			
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
After Reset	0	0	0	0	0	0	0	0	0	0	0	0				

Bit	Mnemonic	Field name	Description
[31:4]	CHED	ControlHeadED	HC traverses the Control list starting with the HcControlHeadED pointer. The content is loaded from HCCA during the initialization of HC.
[3:0]	−	Reserved	

3.14.6.10 HcControlCurrentED Register

The HcControlCurrentED register contains the physical address of the current Endpoint Descriptor of the Control list.

Address = (0x4000_3000) + (0x0024)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	CCED															
Read/Write (HCD)	R/W															
Read/Write (HC)	R/W															
After Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	CCED												Reserved			
Read/Write (HCD)	R/W															
Read/Write (HC)	R/W															
After Reset	0	0	0	0	0	0	0	0	0	0	0	0				

Bit	Mnemonic	Field name	Description
[31:4]	CCED	ControlCurrentED	This pointer is advanced to the next ED after serving the present one. HC will continue processing the list from where it left off in the last Frame. When it reaches the end of the Control list, HC checks the ControlListFilled of HcCommandStatus. If set, it copies the content of HcControlHeadED to HcControlCurrentED and clears the bit. If not set, it does nothing. HCD is allowed to modify this register only when the ControlListEnable of HcControl is cleared. When set, HCD only reads the instantaneous value of this register. Initially, this is set to zero to indicate the end of the Control list.
[3:0]	–	Reserved	

3.14.6.11 HcBulkHeadED Register

The HcBulkHeadED register contains the physical address of the first Endpoint Descriptor of the Bulk list.

Address = (0x4000_3000) + (0x0028)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	BHED															
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
After Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	BHED												Reserved			
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
After Reset	0	0	0	0	0	0	0	0	0	0	0	0				

Bit	Mnemonic	Field name	Description
[31:4]	BHED	BulkHeadED	HC traverses the Bulk list starting with the HcBulkHeadED pointer. The content is loaded from HCCA during the initialization of HC.
[3:0]	–	Reserved	

3.14.6.12 HcBulkCurrentED Register

The HcBulkCurrentED register contains the physical address of the current endpoint of the Bulk list. As the Bulk list will be served in a round-robin fashion, the endpoints will be ordered according to their insertion to the list.

Address = (0x4000_3000) + (0x002C)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	BCED															
Read/Write (HCD)	R/W															
Read/Write (HC)	R/W															
After Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	BCED												Reserved			
Read/Write (HCD)	R/W															
Read/Write (HC)	R/W															
After Reset	0	0	0	0	0	0	0	0	0	0	0	0				

Bit	Mnemonic	Field name	Description
[31:4]	BCED	BulkCurrentED	This is advanced to the next ED after the HC has served the present one. HC continues processing the list from where it left off in the last Frame. When it reaches the end of the Bulk list, HC checks the ControlListFilled of HcControl. If set, it copies the content of HcBulkHeadED to HcBulkCurrentED and clears the bit. If it is not set, it does nothing. HCD is only allowed to modify this register when the BulkListEnable of HcControl is cleared. When set, the HCD only reads the instantaneous value of this register. This is initially set to zero to indicate the end of the Bulk list.
[3:0]	–	Reserved	

3.14.6.13 HcDoneHead Register

The HcDoneHead register contains the physical address of the last completed Transfer Descriptor that was added to the Done queue. In normal operation, the Host Controller Driver should not need to read this register as its content is periodically written to the HCCA.

Address = (0x4000_3000) + (0x0030)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	DH															
Read/Write (HCD)	R															
Read/Write (HC)	R/W															
After Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	DH												Reserved			
Read/Write (HCD)	R															
Read/Write (HC)	R/W															
After Reset	0	0	0	0	0	0	0	0	0	0	0	0				

Bit	Mnemonic	Field name	Description
[31:4]	DH	DoneHead	When a TD is completed, HC writes the content of HcDoneHead to the NextTD field of the TD. HC then overwrites the content of HcDoneHead with the address of this TD.
			This is set to zero whenever HC writes the content of this register to HCCA. It also sets the WritebackDoneHead of HcInterruptStatus.
[3:0]	–	Reserved	

3.14.6.14 HcFmInterval Register

The HcFmInterval register contains a 14-bit value which indicates the bit time interval in a Frame, (i.e., between two consecutive SOFs), and a 15-bit value indicating the Full Speed maximum packet size that the Host Controller may transmit or receive without causing scheduling overrun. The Host Controller Driver may carry out minor adjustment on the FrameInterval by writing a new value over the present one at each SOF. This provides the programmability necessary for the Host Controller to synchronize with an external clocking resource and to adjust any unknown local clock offset.

Address = (0x4000_3000) + (0x0034)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	FIT	FSMPS														
Read/Write (HCD)	R/W	R/W														
Read/Write (HC)	R	R														
After Reset	0	TBD														

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	Reserved		FI													
Read/Write (HCD)			R/W													
Read/Write (HC)			R													
After Reset			1	0	1	1	1	0	1	1	0	1	1	1	1	1

Bit	Mnemonic	Field name	Description
[31]	FIT	FrameInterval Toggle	HCD toggles this bit whenever it loads a new value to FrameInterval.
[30:16]	FSMPS	FSLargestData Packet	This field specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame. The counter value represents the largest amount of data in bits which can be sent or received by the HC in a single transaction at any given time without causing scheduling overrun. The field value is calculated by the HCD.
[15:14]	−	Reserved	
[13:0]	FI	FrameInterval	This specifies the interval between two consecutive SOFs in bit times. The nominal value is set to be 11,999. HCD should store the current value of this field before resetting HC. By setting the HostControllerReset field of HcCommandStatus as this will cause the HC to reset this field to its nominal value. HCD may choose to restore the stored value upon the completion of the Reset sequence.

3.14.6.15 HcFmRemaining Register

The HcFmRemaining register is a 14-bit down counter showing the bit time remaining in the current Frame.

Address = (0x4000_3000) + (0x0038)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	FRT	Reserved														
Read/Write (HCD)	R															
Read/Write (HC)	R/W															
Reset Value	0															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved		FR													
Read/Write (HCD)			R													
Read/Write (HC)			R/W													
Reset Value			0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Mnemonic	Field name	Description
[31]	FRT	FrameRemaining Toggle	This bit is loaded from the FrameIntervalToggle field of HcFmInterval whenever FrameRemaining reaches 0. This bit is used by HCD for the synchronization between FrameInterval and FrameRemaining.
[30:14]	–	Reserved	
[13:0]	FR	FrameRemaining	This counter is decremented at each bit time. When it reaches zero, it is reset by loading the FrameInterval value specified in HcFmInterval at the next bit time boundary. When entering the USBOPERATIONAL state, HC re-loads the content with the FrameInterval of HcFmInterval and uses the updated value from the next SOF.

3.14.6.16 HcFmNumber Register

The HcFmNumber register is a 16-bit counter. It provides a timing reference among events happening in the Host Controller and the Host Controller Driver. The Host Controller Driver may use the 16-bit value specified in this register and generate a 32-bit frame number without requiring frequent access to the register.

Address = (0x4000_3000) + (0x003C)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	Reserved															
Read/Write (HCD)																
Read/Write (HC)																
Reset Value																
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	FN															
Read/Write (HCD)	R															
Read/Write (HC)	R/W															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Mnemonic	Field name	Description
[31:16]	–	Reserved	
[15:0]	FN	FrameNumber	This is incremented when HcFmRemaining is re-loaded. It will be rolled over to 0h after ffffh. When entering the USBOPERATIONAL state, this will be incremented automatically. The content will be written to HCCA after HC has incremented the FrameNumber at each frame boundary and sent a SOF but before HC reads the first ED in that Frame. After writing to HCCA, HC will set the StartofFrame in HcInterruptStatus.

3.14.6.17 HcPeriodicStart Register

The HcPeriodicStart register has a 14-bit programmable value which determines when is the earliest time HC should start processing the periodic list

Address = (0x4000_3000) + (0x0040)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	Reserved															
Read/Write (HCD)																
Read/Write (HC)																
Reset Value																
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	Reserved		PS													
Read/Write (HCD)			R/W													
Read/Write (HC)			R													
Reset Value			0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Mnemonic	Field name	Description
[31:14]	−	Reserved	
[13:0]	PS	PeriodicStart	After a hardware reset, this field is cleared. This is then set by HCD during the HC initialization. The value is calculated roughly as 10% off from HcFmInterval. A typical value will be 3E67h. When HcFmRemaining reaches the value specified, processing of the periodic lists will have priority over Control/Bulk processing. HC will therefore start processing the Interrupt list after completing the current Control or Bulk transaction that is in progress.

3.14.6.18 HcLSThreshold Register

The HcLSThreshold register contains an 11-bit value used by the Host Controller to determine whether to commit to the transfer of a maximum of 8-byte LS packet before EOF. Neither the Host Controller nor the Host Controller Driver are allowed to change this value.

Address = (0x4000_3000) + (0x0044)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	Reserved															
Read/Write (HCD)																
Read/Write (HC)																
Reset Value																
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	Reserved				LST											
Read/Write (HCD)					R/W											
Read/Write (HC)					R											
Reset Value					0	1	1	0	0	0	1	0	1	0	0	0

Bit	Mnemonic	Field name	Description
[31:12]	–	Reserved	
[11:0]	LST	LSThreshold	This field contains a value which is compared to the FrameRemaining field prior to initiating a Low Speed transaction. The transaction is started only if FrameRemaining ≥ this field. The value is calculated by HCD with the consideration of transmission and setup overhead.

3.14.6.19 HcRhDescriptorA Register

The HcRhDescriptorA register is the first register of two describing the characteristics of the Root Hub. Reset values are implementation-specific. The descriptor length, descriptor type, and hub controller current fields of the hub Class Descriptor are emulated by the HCD. All other fields are located in the HcRhDescriptorA and HcRhDescriptorB registers.

Address = (0x4000_3000) + (0x0048)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	POTPGT								Reserved							
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
Reset Value	0	0	0	0	0	0	1	0								
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	Reserved			NOCP	OCPM	DT	NPS	PSM	NDP							
Read/Write (HCD)				R/W	R/W	R	R/W	R/W	R							
Read/Write (HC)				R	R	R	R	R	R							
Reset Value				0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Mnemonic	Field name	Description
[31:24]	POTPGT	PowerOnTo PowerGoodTime	This byte specifies the duration HCD has to wait before accessing a powered-on port of the Root Hub. It is implementation-specific. The unit of time is 2 ms. The duration is calculated as POTPGT * 2 ms.
[23:13]		Reserved	
[12]	NOCP	NoOverCurrent Protection	This bit describes how the overcurrent status for the Root Hub ports are reported. When this bit is cleared, the OverCurrentProtectionMode field specifies global or per-port reporting. 0y0: Over-current status is reported collectively for all downstream ports 0y1: No overcurrent protection supported
[11]	OCPM	OverCurrent ProtectionMode	This bit describes how the overcurrent status for the Root Hub ports are reported. At reset, this fields should reflect the same mode as PowerSwitchingMode. This field is valid only if the NoOverCurrentProtection field is cleared. 0y0: Over-current status is reported collectively for all downstream ports 0y1: Over-current status is reported on a per-port basis
[10]	DT	DeviceType	This bit specifies that the Root Hub is not a compound device. The Root Hub is not permitted to be a compound device. This field should always read/write 0.
[9]	NPS	NoPower Switching	These bits are used to specify whether power switching is supported or ports are always powered. It is implementation- specific. When this bit is cleared, the PowerSwitchingMode specifies global or per-port switching. 0y0: Ports are power switched 0y1: Ports are always powered on when the HC is powered on
[8]	PSM	PowerSwitching Mode	This bit is used to specify how the power switching of the Root Hub ports is controlled. It is implementation-specific. This field is only valid if the NoPowerSwitching field is cleared. 0y0: All ports are powered at the same time. 0y1: Each port is powered individually. This mode allows port power to be controlled by either the global switch or per-port switching. If the PortPowerControlMask bit is set, the port responds only to port power commands (Set/ClearPortPower). If the port mask is cleared, then the port is controlled only by the global power switch (Set/ClearGlobalPower).
[7:0]	NDP	Number DownstreamPorts	These bits specify the number of downstream ports supported by the Root Hub. It is implementation-specific. This module has one port, so "0x01" is read.

3.14.6.20 HcRhDescriptorB Register

The HcRhDescriptorB register is the second register of two describing the characteristics of the Root Hub. These fields are written during initialization to correspond with the system implementation. Reset values are implementation-specific.

Address = (0x4000_3000) + (0x004C)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	PPCM															
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	DR															
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Mnemonic	Field name	Description
[31:16]	PPCM	PortPower ControlMask	Each bit indicates if a port is affected by a global power control command when PowerSwitchingMode is set. When set, the port's power state is only affected by per-port power control (Set/ClearPortPower). When cleared, the port is controlled by the global power switch (Set/ClearGlobalPower). If the device is configured to global switching mode (PowerSwitchingMode=0), this field is not valid. bit 0: Reserved bit 1: Ganged-power mask on Port #1 bit 2: Ganged-power mask on Port #2 … bit 15: Ganged-power mask on Port #15 (note) Write "0" to the corresponding bit. Because this host controller doesn't have Port#2 - Port#15.
[15:0]	DR	Device Removable	Each bit is dedicated to a port of the Root Hub. When cleared, the attached device is removable. When set, the attached device is not removable. bit 0: Reserved bit 1: Device attached to Port #1 bit 2: Device attached to Port #2 … bit 15: Device attached to Port #15 (note) Write "0" to the corresponding bit. Because this host controller doesn't have Port#2 - Port#15.

3.14.6.21 HcRhStatus Register

The HcRhStatus register is divided into two parts. The lower word of a Dword represents the Hub Status field and the upper word represents the Hub Status Change field. Reserved bits should always be written '0'.

Address = (0x4000_3000) + (0x0050)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	CRWE	Reserved													OCIC	LPSC
Read/Write (HCD)	W														R/W	R/W
Read/Write (HC)	R														R/W	R
Reset Value	—														0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	DRWE	Reserved													OCI	LPS
Read/Write (HCD)	R/W														R	R/W
Read/Write (HC)	R														R/W	R
Reset Value	0														0	0

Bit	Mnemonic	Field name	Description
[31]	CRWE	ClearRemote WakeupEnable	Writing a '1' clears DeviceRemoveWakeupEnable. Writing a '0' has no effect.
[30:18]	−	Reserved	
[17]	OCIC	OverCurrent Indicator Change	This bit is set by hardware when a change has occurred to the OCI field of this register. The HCD clears this bit by writing a '1'. Writing a '0' has no effect.
[16]	LPSC	LocalPower StatusChange	(read) LocalPowerStatusChange The Root Hub does not support the local power status feature; thus, this bit is always read as '0'. (write) SetGlobalPower In global power mode (PowerSwitchingMode=0), This bit is written to '1' to turn on power to all ports (clear PortPowerStatus). In per-port power mode, it sets PortPowerStatus only on ports whose PortPowerControlMask bit is not set. Writing a '0' has no effect.

[15]	DRWE	DeviceRemote WakeupEnable	(read) DeviceRemoteWakeupEnable
			This bit enables a ConnectStatusChange bit as a resume event, causing a USBSUSPEND to USBRESUME state transition and setting the ResumeDetected interrupt.
			    0y0 = ConnectStatusChange is not a remote wakeup event.
			    0y1 = ConnectStatusChange is a remote wakeup event.
			
			(write) SetRemoteWakeupEnable
			Writing a '1' sets DeviceRemoveWakeupEnable. Writing a '0' has no effect.
			 (refer to section 3.14.8)

[14:2]	−	Reserved	
[1]	OCI	OverCurrent Indicator	This bit reports overcurrent conditions when the global reporting is implemented. When set, an overcurrent condition exists. When cleared, all power operations are normal. If per-port overcurrent protection is implemented this bit is always '0'
[0]	LPS	LocalPower Status	(read) LocalPowerStatus The Root Hub does not support the local power status feature; thus, this bit is always read as '0'. (write) ClearGlobalPower In global power mode (PowerSwitchingMode=0), this bit is written to '1' to turn off power to all ports (clear PortPowerStatus). In per-port power mode, it clears PortPowerStatus only on ports whose PortPowerControlMask bit is not set. Writing a '0' has no effect.

3.14.6.22 HcRhPortStatus1 Register

The HcRhPortStatus1 register is used to control and report port events on a per-port basis. NumberDownstreamPorts represents the number of HcRhPortStatus registers that are implemented in hardware. The lower word is used to reflect the port status, whereas the upper word reflects the status change bits. Some status bits are implemented with special write behavior (see below). If a transaction (token through handshake) is in progress when a write to change port status occurs, the resulting port status change must be postponed until the transaction completes. Reserved bits should always be written '0'.

Address = (0x4000_3000) + (0x0054)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	Reserved											PRSC	OCIC	PSSC	PESC	CSC
Read/Write (HCD)												R/W	R/W	R/W	R/W	R/W
Read/Write (HC)												R/W	R/W	R/W	R/W	R/W
Reset Value												0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	Reserved						LSDA	PPS	Reserved			PRS	POCI	PSS	PES	CCS
Read/Write (HCD)							R/W	R/W				R/W	R/W	R/W	R/W	R/W
Read/Write (HC)							R/W	R/W				R/W	R/W	R/W	R/W	R/W
Reset Value							X	0				0	0	0	0	0

Bit	Mnemonic	Field name	Description
[31:21]	−	Reserved	
[20]	PRSC	PortResetStatus Change	This bit is set at the end of the 10-ms port reset signal. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. 0y0 = port reset is not complete 0y1 = port reset is complete
[19]	OCIC	PortOverCurrent IndicatorChange	This bit is valid only if overcurrent conditions are reported on a per-port basis. This bit is set when Root Hub changes the PortOverCurrentIndicator bit. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. 0y0 = no change in PortOverCurrentIndicator 0y1 = PortOverCurrentIndicator has changed
[18]	PSSC	PortSuspend StatusChange	This bit is set when the full resume sequence has been completed. This sequence includes the 20-s resume pulse, LS EOP, and 3-ms resynchronization delay. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. This bit is also cleared when ResetStatusChange is set. 0y0 = resume is not completed 0y1 = resume completed

Bit	Mnemonic	Field name	Description
[17]	PESC	PortEnable StatusChange	This bit is set when hardware events cause the PortEnableStatus bit to be cleared. Changes from HCD writes do not set this bit. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. 　　　　0y0 = no change in PortEnableStatus 　　　　0y1 = change in PortEnableStatus
[16]	CSC	ConnectStatus Change	This bit is set whenever a connect or disconnect event occurs. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. If CurrentConnectStatus is cleared when a SetPortReset, SetPortEnable, or SetPortSuspend write occurs, this bit is set to force the driver to re-evaluate the connection status since these writes should not occur if the port is disconnected. 　　　　0y0 = no change in CurrentConnectStatus 　　　　0y1 = change in CurrentConnectStatus Note: If the DeviceRemovable[NDP] bit is set, this bit is set only after a Root Hub reset to inform the system that the device is attached.
[15:10]		Reserved	
[9]	LSDA	LowSpeed DeviceAttached	(read) LowSpeedDeviceAttached This bit indicates the speed of the device attached to this port. When set, a Low Speed device is attached to this port. When clear, a Full Speed device is attached to this port. This field is valid only when the CurrentConnectStatus is set. 　　　　0y0 = full speed device attached 　　　　0y1 = low speed device attached (write) ClearPortPower The HCD clears the PortPowerStatus bit by writing a '1' to this bit. Writing a '0' has no effect.

Bit	Mnemonic	Field name	Description
[8]	PPS	PortPower Status	(read) PortPowerStatus This bit reflects the port's power status, regardless of the type of power switching implemented. This bit is cleared if an overcurrent condition is detected. HCD sets this bit by writing SetPortPower or SetGlobalPower. HCD clears this bit by writing ClearPortPower or ClearGlobalPower. Which power control switches are enabled is determined by PowerSwitchingMode and PortPortControlMask[NDP]. In global switching mode (PowerSwitchingMode=0), only Set/ClearGlobalPower controls this bit. In per-port power switching (PowerSwitchingMode=1), if the PortPowerControlMask[NDP] bit for the port is set, only Set/ClearPortPower commands are enabled. If the mask is not set, only Set/ClearGlobalPower commands are enabled. When port power is disabled, CurrentConnectStatus, PortEnableStatus, PortSuspendStatus, and PortResetStatus should be reset. 0y0 = port power is off 0y1 = port power is on (write) SetPortPower The HCD writes a '1' to set the PortPowerStatus bit. Writing a '0' has no effect. Note: This bit is always reads '1b' if power switching is not supported.
[7:5]	−	Reserved	
[4]	PRS	PortReset Status	(read) PortResetStatus When this bit is set by a write to SetPortReset, port reset signaling is asserted. When reset is completed, this bit is cleared when PortResetStatusChange is set. This bit cannot be set if CurrentConnectStatus is cleared. 0y0 = port reset signal is not active 0y1 = port reset signal is active (write) SetPortReset The HCD sets the port reset signaling by writing a '1' to this bit. Writing a '0' has no effect. If CurrentConnectStatus is cleared, this write does not set PortResetStatus, but instead sets ConnectStatusChange. This informs the driver that it attempted to reset a disconnected port.

Bit	Mnemonic	Field name	Description
[3]	POCI	PortOverCurrent Indicator	(read) PortOverCurrentIndicator This bit is only valid when the Root Hub is configured in such a way that overcurrent conditions are reported on a per-port basis. If per-port overcurrent reporting is not supported, this bit is set to 0. If cleared, all power operations are normal for this port. If set, an overcurrent condition exists on this port. This bit always reflects the overcurrent input signal 0y0 = no overcurrent condition. 0y1 = overcurrent condition detected. (write) ClearSuspendStatus The HCD writes a '1' to initiate a resume. Writing a '0' has no effect. A resume is initiated only if PortSuspendStatus is set.
[2]	PSS	PortSuspend Status	(read) PortSuspendStatus This bit indicates the port is suspended or in the resume sequence. It is set by a SetSuspendState write and cleared when PortSuspendStatusChange is set at the end of the resume interval. This bit cannot be set if CurrentConnectStatus is cleared. This bit is also cleared when PortResetStatusChange is set at the end of the port reset or when the HC is placed in the USBRESUME state. If an upstream resume is in progress, it should propagate to the HC. 0y0 = port is not suspended 0y1 = port is suspended (write) SetPortSuspend The HCD sets the PortSuspendStatus bit by writing a '1' to this bit. Writing a '0' has no effect. If CurrentConnectStatus is cleared, this write does not set PortSuspendStatus; instead it sets ConnectStatusChange. This informs the driver that it attempted to suspend a disconnected port.

Bit	Mnemonic	Field name	Description
[1]	PES	PortEnable Status	(read) PortEnableStatus This bit indicates whether the port is enabled or disabled. The Root Hub may clear this bit when an overcurrent condition, disconnect event, switched-off power, or operational bus error such as babble is detected. This change also causes PortEnabledStatusChange to be set. HCD sets this bit by writing SetPortEnable and clears it by writing ClearPortEnable. This bit cannot be set when CurrentConnectStatus is cleared. This bit is also set, if not already, at the completion of a port reset when ResetStatusChange is set or port suspend when SuspendStatusChange is set. 0y0 = port is disabled 0y1 = port is enabled (write) SetPortEnable The HCD sets PortEnableStatus by writing a '1'. Writing a '0' has no effect. If CurrentConnectStatus is cleared, this write does not set PortEnableStatus, but instead sets ConnectStatusChange. This informs the driver that it attempted to enable a disconnected port.
[0]	CCS	CurrentConnect Status	(read) CurrentConnectStatus This bit reflects the current state of the downstream port. 0y0 = no device connected 0y1 = device connected (write) ClearPortEnable The HCD writes a '1' to this bit to clear the PortEnableStatus bit. Writing a '0' has no effect. The CurrentConnectStatus is not affected by any write. Note: This bit is always read '1b' when the attached device is nonremoveable (DeviceRemoveable[NDP]).

3.14.6.23 HcBCR0 Register

The HcBCR0 register controls over current input of enable or disable to the USBHC and the SUSPEND state of the USB transceiver.

The USBHC is not active in SLOW mode or Low Power Consumption Modes. Therefore, before entering SLOW mode or Low Power Consumption Modes, place the USBHC and the USB transceiver in the SUSPEND state.

Address = (0x4000_3000) + (0x0080)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Symbol	Reserved	TRNS SUSP	OVCE	Reserved												
Read/Write	R	R/W	R/W	R												
Reset Value	0	1	0	0												
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Symbol	Reserved															
Read/Write	R															
Reset Value	0															

Bit	Mnemonic	Field name	Description
[31]	−	Reserved	
[30]	TRNS_SUSP	Transceiver Suspend	This bit controls the SUSPEND state of the USB transceiver. To enter SLOW mode or Low power Consumption Modes with the USB transceiver in the SUSPEND state, set this bit to 1. 0: - (Controlled by the USB Host Controller) 1: Suspend
[29]	OVCE	USB Host Over Current Input Enable	USB Host over current input enable 0y0: Enable 0y1: Disable
[28:0]	−	Reserved	

3.14.7 Notes on Using the USB Host Controller

3.14.7.1 Setting the USB Clock

The PLL is not active after reset release. Therefore, before accessing the USB Host, the <PLLSEL> bit in the CGPLLSEL register must be set to 1 in the clock controller.

3.14.7.2 Oscillator Recommendation

To generate a clock for the USBHC, we recommend using a 12-MHz crystal oscillator with an accuracy of ±50 ppm or less to comply with the USB specification.

Note that a USB clock generated by the on-chip PLL may not satisfy the requirements of the USB specification depending on the implementation environment, conditions and variations.

3.14.7.3 Entering SLOW Mode and Low Power Consumption Modes

Before entering SLOW Mode and Low Power Consumption Modes, the USBHC must be placed in the SUSPEND state. Then trun off Vbus.

< Example Software Setting >
1- Disable Interrupt
2- HcCommandStatus.HCR[0] = "0y1" write
: Software Reset of the USB Host Controller
3- HcControl.HCFS[7:6] = "0y11" read
: Check the transition to the SUSPENDstate.
4- HcBCR0.TRANS_SUSP[30] = "0y1"write
: Change the state of the D+ and D- pin to the SUSPEND state.
5- Output "0" from PG6   : Turn off Vbus
6- Setup to shifting Low Power Consumtion and others
(Stop of peripheral function, port setting, warm up setting and so on. About shifting BACKUP mode, refer to that section)
7- Stop PLL
8- Execute WFI Instruction

3.14.7.4 When Not Using USB

The D+ and D- pins must be pulled down.

3.14.7.5 Competing access to the RAM0 and the RAM1

If the CPU/DMA accesses the RAM0 or the RAM1 when the USBHC is already accessing it, the USBHC connection is discontinued and the CPU/DMA gets higher priority for access. The USBHC is reconnected when the CPU/DMA access is completed. If the disconnection of the USBHC caused by the CPU/DMA takes long, it may cause problems in USB transfer. Caution should be used. Specifically, in case IN transfer, permit the CPU/DMA access to complete in the period that the 64-byte FIFO contained in the USBHC becomes full (within 42.68μS).

3.14.8 Restrictions on Using the USB Host Controller

1. For an isochronous transfer, a Frame number to be transferred is defined in an Isochronous Transfer Descriptor (ITD). However, Frame numbers are not synchronized between the Host and software. If a descriptor to be executed in a previous Frame is scheduled later, the Host determines that a time error has occurred and writes back DATAOVERRUN to the CC field of the ITD. At this time, if the following conditions are met, the Host will write back inappropriate status (NOERROR).

   <Conditions>
   The above problem occurs if transfers are scheduled in a way the following two conditions both hold true:
   1. ITD.FC[2:0] = R[2:0]
   2. ITD.FC[2:0] < R[15:0]

   where ITD.FC indicates the number of times an ITD is executed and
   R = HcFmNumber (current Frame number) – ITD.SF (transfer start Frame number).

   Make sure that each ITD is synchronized to the current Frame number. If not, this ITD should not be linked.

2. If a fatal error occurs on the USB system and the Host detects this error, the OHCI core sets HcInterrutStatus.UE[4].

   At this time, if the HcInterruptEnable.UE[4] register has been set additionally, a hardware interrupt is generated.

   After this interrupt is detected, a software reset (HcCommandStatus.HCR[0] = 0y1) is required to recover from the Unrecoverable Error state and the Host then moves to the SUSPEND state.

3. After the software reset, OHCI registers are initialized. If a remote wake-up from the device occurs, the Host remains in the SUSPEND state. When the remote wake-up function is used, a program for recovering from the SUSPEND state must be prepared.

4. When an overcurrent error occurs, if the HcRhDescriptorA.NPS[9] register has been set to 0y1, the HcRhPortStatus1.PRS[4] and HcRhPortStatus1.PSS[2] bits are not cleared. Therefore, do not set the HcRhDescriptorA.NPS[9] to 0y1 and the HcRhDescriptorB.DR[Port No] to 0y1.

5. When the HcRhStatus.DRWE [15] is set to 0y1, a remote wake-up event does not cause a USBSUSPEND to USBRESUME transition.
   When remote wake-up events are used, causing state transition by monitoring to the HcInterruptStatus.RD[3] by the HCD.
   And when remote wake-up events are not used, do not set the HcRhStatus.DRWE [15] to 0y1.

6. In a system supporting overcurrent conditions, set the HcRhDescriptorA.NOCP [12] to 0y0.

7. When Port Reset is executed while generating Babble, Port keep disable state.
    Port Reset sequence is below.

< Example >
    1- HcRhPortStatus1.PRS[4] = "0y1" write   : Execute Port Reset.
    2- Detect of HW Interrupt.
    3- HcRhPortStatus1.PRSC[20] = "0y1", HcRhPortStatus1.PES[1] = "0y0",
        HcRhPortStatus1.CCS[0] = "0y1"read   : Port is an invalid state.
    4- Clear of HW Interrupt
    5- HcRhPortStatus1.PRS[4] = "0y1" write : Retry Port Reset.
    6- Detect of HW Interrupt.
    7- HcRhPortStatus1.PRSC[20] = "0y1", HcRhPortStatus1.PES[1] = "0y1",
        HcRhPortStatus1.CCS[0] = "0y1" read   : Port is a valid state.

8. When Scheduling Overrun is occurred while executing Full-Speed Isochronous transfer,
    It may be generated the Unrecoverable Error. When Unrecoverable Error was generated,
    recover from Unrecoverable Error by software reset.

3.14.9 Connection Example
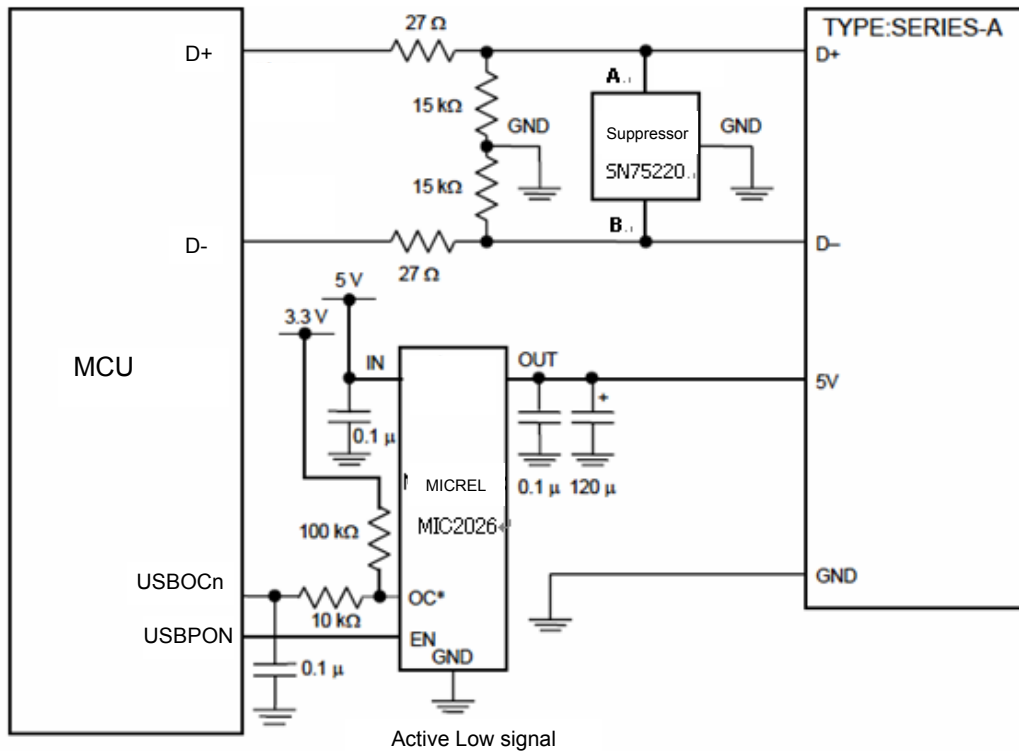
Bus power switch device: TPS2052 (Texas Instruments)
                             MIC2026-1BM (MICREL)
                             MIC2026-1BN (MICREL), etc.
 Transient voltage suppressor device: SN75240 (Texas Instruments), etc.
Resistance precision: 5%
Resistance rating: 1/2 W for 27 ohms recommended
Capacitor: Low ESR type 120 uF capacitor (OS-CON, etc.) recommended

Note 1:   Do not apply voltages exceeding the absolute maximum rating.
Note 2:   When designing your board, make sure that the D+ and D- pins are placed at the equal distance from the
          USB A receptacle.
Note 3:   No suppressor device is required in the USB specification.
Note 4:   After reset release, the USBPON and USBOCn pins are initialized as input ports. Process the pin so as not
          to influence the external control circuit.

3.15   Analog/Digital Converter

A 10-bit, sequential-conversion analog/digital converter (A/D converter) is built into the TMPM322F10. This A/D converter is equipped with 16 analog input channels.

Fig.3.16.1 shows the block diagram of this A/D converter.

These 16 analog input channels (pins AN0 through AN15) are also used as input/ output ports.

(Note) **If it is necessary to reduce a power current by operating the TMPM322F10 in IDLE or STOP mode and if either case shown below is applicable, you must first stop the A/D converter and then execute the instruction to put the TMPM322 into standby mode:**

1) The TMPM322 must be put into IDLE mode when ADMOD1<I2AD> is "0."
2) The TMPM322 must be put into STOP mode.
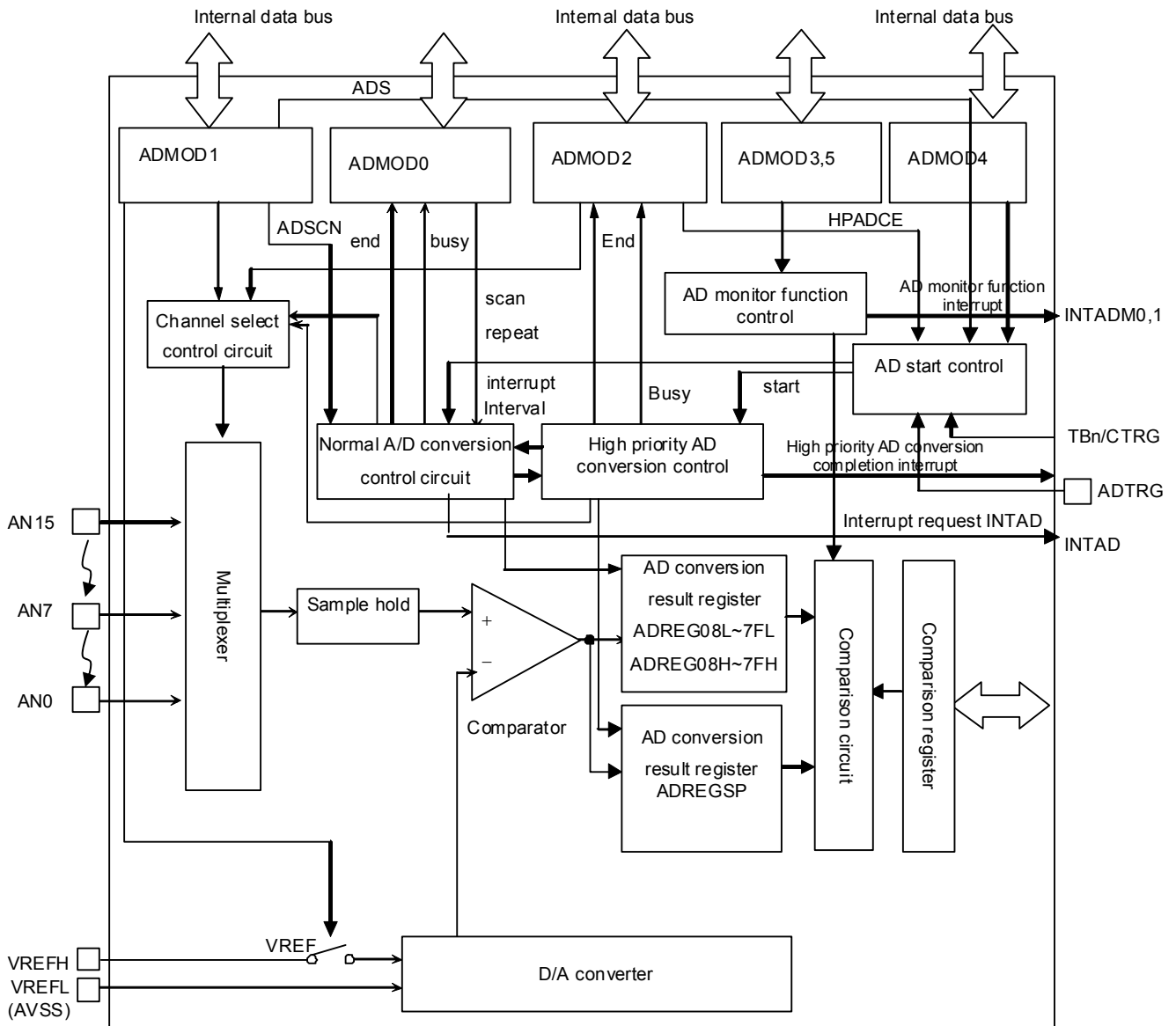
Fig. 3. 16. 1   A/D Converter Block Diagram

3.15.1 Registers

The control registers and addresses of the A/D converter are as follows.

Register name		Address
A/D Conversion Clock Setting Register	ADCLK	0x400F_0000
A/D Mode Control Register 0	ADMOD0	0x400F_0004
A/D Mode Control Register 1	ADMOD1	0x400F_0008
A/D Mode Control Register 2	ADMOD2	0x400F_000C
A/D Mode Control Register 3	ADMOD3	0x400F_0010
A/D Mode Control Register 4	ADMOD4	0x400F_0014
A/D Mode Control Register 5	ADMOD5	0x400F_0018
A/D Conversion Accuracy Setting Register (Note)	ADCBAS	0x400F_0020
A/D Conversion Result Register 08	ADREG08	0x400F_0030
A/D Conversion Result Register 19	ADREG19	0x400F_0034
A/D Conversion Result Register 2A	ADREG2A	0x400F_0038
A/D Conversion Result Register 3B	ADREG3B	0x400F_003C
A/D Conversion Result Register 4C	ADREG4C	0x400F_0040
A/D Conversion Result Register 5D	ADREG5D	0x400F_0044
A/D Conversion Result Register 6E	ADREG6E	0x400F_0048
A/D Conversion Result Register 7F	ADREG7F	0x400F_004C
A/D Conversion Result Register SP	ADREGSP	0x400F_0050
A/D Conversion Result Comparison Register 0	ADCMP0	0x400F_0054
A/D Conversion Result Comparison Register 1	ADCMP1	0x400F_0058

Note) To assure conversion accuracy, the ADCBAS register must be configured as follows.

0x400F_0020 ＝ 0x58

ADCBAS

	7	6	5	4	3	2	1	0
bit Symbol								
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	1	1	1	0	0	0
Function	Write "0".	Write "1".	Write "0".	Write "1".	Write "1".	Write "0".	Write "0".	Write "0".

3.15.2   Register Description

The A/D converter is controlled by A/D mode control registers (ADMOD0, ADMOD1, ADMOD2, ADMOD3, ADMOD4 and ADMOD5). Results of A/D conversion are stored in 8 A/D conversion result registers ADREG08 through ADREG7F. Results of top-priority conversion are stored in ADREGSP.

Here are the descriptions of the registers.

A/D Mode Control Register 0

ADMOD0

	7	6	5	4	3	2	1	0
bit Symbol	EOCFN	ADBFN		ITM		REPEAT	SCAN	ADS
Read/Write	R		R	R/W				
After reset	0	0	0	0	0	0	0	0
Function	Normal A/D conversion completion flag 0: Before or during conversion 1: Completion	Normal A/D conversion BUSY flag 0: Conversion stop 1: During conversion	"0" is read.	Specify interrupt in fixed channel repeat conversion mode	Specify interrupt in fixed channel repeat conversion mode.	Specify repeat mode 0: Single conversion mode 1: Repeat conversion mode	Specify scan mode 0: Fixed channel mode 1: Channel scan mode	Start A/D conversion 0: Don't care 1: Start conversion "0" is always read.

Specify A/D conversion interrupt in fixed channel repeat conversion mode

	Fixed channel repeat conversion mode <SCAN> = "0", <REPEAT> = "1"
00	Generate interrupt once every single conversion
01	Generate interrupt once every 4 conversions
10	Generate interrupt once every 8 conversions
11	Setting prohibited

(Note 1) Please specify the mode first and then specify the <ADS> bit.

(Note 2) Before using the AD conversion completion interrupt for starting DMA transfer, apply a software reset on the AD converter through bits 1:0 of the ADMOD4 register. Then, enable the DMAC (for receiving DMA requests) and then set (start) the ADC.

A/D Mode Control Register 1

ADMOD1		7	6	5	4	3	2	1	0
	bit Symbol	VREFON	I2AD	ADSCN	−	ADCH3	ADCH2	ADCH1	ADCH0
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	VREF application control 0 :OFF 1 :ON	IDLE 0 : Stop 1 : Operation	Specify operation mode for channel scanning 0: 4ch scan 1: 8ch scan	Write "0."	Select analog input channel			

Select analog input channel

<SCAN> <ADCH [3:0]>	0 Fixed channel	1 Channel scan （ADSCN=0）	1 Channel scan （ADSCN= 1 ）
0000	AN0	AN0	AN0
0001	AN1	AN0〜AN1	AN0〜AN1
0010	AN2	AN0〜AN2	AN0〜AN2
0011	AN3	AN0〜AN3	AN0〜AN3
0100	AN4	AN4	AN0〜AN4
0101	AN5	AN4〜AN5	AN0〜AN5
0110	AN6	AN4〜AN6	AN0〜AN6
0111	AN7	AN4〜AN7	AN0〜AN7
1000	AN8	AN8	AN8
1001	AN9	AN8〜AN9	AN8〜AN9
1010	AN10	AN8〜AN10	AN8〜AN10
1011	AN11	AN8〜AN11	AN8〜AN11
1100	AN12	AN12	AN8〜AN12
1101	AN13	AN12〜AN13	AN8〜AN13
1110	AN14	AN12〜AN14	AN8〜AN14
1111	AN15	AN12〜AN15	AN8〜AN15

(Note 1) Before starting AD conversion, write "1" to the <VREFON> bit, wait for 3 μs during which time the internal reference voltage should stabilize, and then write "1" to the ADMOD0<ADS> bit.

(Note 2) To go into standby mode upon completion of AD conversion, set <VREFON> to "0."

A/D Mode Control Register 2

ADMOD2

	7	6	5	4	3	2	1	0
bit Symbol	EOCFHP	ADBFHP	HPADCE	―	HPADCH			
Read/Write	R	R	R/W					
After reset	0	0	0	0	0	0	0	0
Function	Top-priority AD conversion completion flag 0:Before or during conversion 1: Upon completion	Top-priority AD conversion BUSY flag 0: During conversion halts 1:During conversion	Activate top-priority conversion 0: Don't care 1: Start conversion. "0" is always read.	Write "0".	Select analog input channel when activating top-priority conversion.			

<HPADCH4,3.2, 1, 0>	Analog input channel when executing top-priority conversion
0000	AN0
0001	AN1
0010	AN2
0011	AN3
0100	AN4
0101	AN5
0110	AN6
0111	AN7
1000	AN8
1001	AN9
1010	AN10
1011	AN11
1100	AN12
1101	AN13
1110	AN14
1111	AN15

(Note) Set the analog input channels first, and then set the <HPADCE> bit.

A/D Mode Control Register 3

ADMOD3

	7	6	5	4	3	2	1	0
bit Symbol			ADOBIC0	ADREGS0				ADOBSV0
Read/Write	R/W	R	R/W					
After reset	0	0	0	0	0	0	0	0
Function	Write "0".	"0" can be read.	Make AD monitor function interrupt setting 0: Smaller than comparison Reg. 1: Larger than comparison Reg.	BIT for selecting the AD conversion result storage Reg. that is to be compared with the comparison Reg. if the AD monitor function is enabled				AD monitor function 0 : Disable 1 : Enable

<ADREGS[2:1]>	AD conversion result storage Reg. to be compared
0000	ADREG08
0001	ADREG19
0010	ADREG2A
0011	ADREG3B
0100	ADREG4C
0101	ADREG5D
0110	ADREG6E
0111	ADREG7F
1XXX	ADREGSP

A/D Mode Control Register 5

	7	6	5	4	3	2	1	0
bit Symbol			ADOBIC1	ADREGS1				ADOBSV1
Read/Write	R		R/W					
After Reset	0	0	0	0	0	0	0	0
Function	"0" can be read.		Make AD monitor function interrupt setting Smaller than comparison Reg. Larger than comparison Reg.	BIT for selecting the AD conversion result storage Reg. that is to be compared with the comparison Reg. if the AD monitor function is enabled .				AD monitor function 0 : Disable 1 : Enable

ADMOD5

<ADREGS[2:0]>	AD conversion result storage Reg. to be compared
0000	ADREG08
0001	ADREG19
0010	ADREG2A
0011	ADREG3B
0100	ADREG4C
0101	ADREG5D
0110	ADREG6E
0111	ADREG7F
1XXX	ADREGSP

A/D Mode Control Register 4

ADMOD4

	7	6	5	4	3	2	1	0
bit Symbol	HADHS	HADHTG	ADHS	ADHTG			ADRST	
Read/Write	R/W				R		W	W
After reset	0	0	0	0	0		—	—
Function	HW source for activating top-priority A/D conversion 0:External TRG 1:Match with TB5RG0	HW for activating top-priority A/D conversion 0: Disable 1: Enable	HW source for activating normal A/D conversion 0:External TRG 1:Match with TB6RG0	HW for activating normal A/D conversion 0: Disable 1: Enable	"0" can be read.		Overwriting 10 with 01 allows ADC to be software reset.	

(Note 1)   If AD conversion is executed with the match triggers <ADHTG> and <HADHTG> of a 16-bit timer set to "1" by using a source for triggering H/W, A/D conversion can be activated at specified intervals by performing four steps shown below when the timer is idle:

    1. Stop the timer.
    2. Select a source for triggering HW: <ADHS>, <HADHS>
    3. Enable H/W activation of AD conversion: <ADHTG>, <HADHTG>
    4. Start the timer.

(Note 2)   Do not make a top-priority AD conversion setting and a normal AD conversion setting simultaneously.

(Note 3)   The external trigger cannot be used for H/W activation of AD conversion when it is used for H/W activation of top priority AD conversion.

(Note 4)   A software reset initializes other bits. Resetting a mode register is needed.

A/D Conversion Result Register 08

ADREG08		15	14	13	12	11	10	9	8
	bit Symbol	ADR0[9:2]							
	Read/Write	R							
	After reset	0							
	Function	Store upper 8 bits of A/D conversion result							
		7	6	5	4	3	2	1	0
	bit Symbol	ADR0[1:0]						OVR0	ADR0RF
	Read/Write	R		R				R	R
	After reset	0		0				0	0
	Function	Store lower 2 bits of A/D conversion result		"0" can be read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1:Presence of conversion result

A/D Conversion Result Register 19

ADREG19		15	14	13	12	11	10	9	8
	bit Symbol	ADR1[9:2]							
	Read/Write	R							
	After reset	0							
	Function	Store upper 8 bits of A/D conversion result							
		7	6	5	4	3	2	1	0
	bit Symbol	ADR1[1:0]						OVR1	ADR1RF
	Read/Write	R		R				R	R
	After reset	0		0				0	0
	Function	Store lower 2 bits of A/D conversion result		"0" can be read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1:Presence of conversion result

- Bit 0 of ADREG08/ADREG19 is the A/D conversion result storage flag <ADRxRF>. This bit is set to "1" after an A/D converted value is stored. A read of register (ADRx[1:0]) will set this bit to "0".
- Bit 1 of ADREG08/ADREG19 is the over RUN flag <OVRx>. This bit is set to "1" if a conversion result is overwritten before both conversion result storage register (ADRx[9:0]) is read. A read of a flag will clear this bit to "0."
- When reading conversion result storage registers on a byte-by-byte basis, first read upper registers and then read lower registers.
- Access to this register must be a half word or word access.

A/D Conversion Result Register 2A

ADREG2A

	15	14	13	12	11	10	9	8
bit Symbol	ADR2[9:2]							
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							
	7	6	5	4	3	2	1	0
bit Symbol	ADR2[1:0]						OVR2	ADR2RF
Read/Write	R		R				R	R
After reset	0		0				0	0
Function	Store lower 2 bits of A/D conversion result		"0" can be read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1:Presence of conversion result

A/D Conversion Result Register 3B

ADREG3B

	15	14	13	12	11	10	9	8
bit Symbol	ADR3[9:2]							
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							
	7	6	5	4	3	2	1	0
bit Symbol	ADR3[1:0]						OVR3	ADR3RF
Read/Write	R		R				R	R
After reset	0		0				0	0
Function	Store lower 2 bits of A/D conversion result		"0" can be read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1:Presence of conversion result

- Bit 0 of ADREG2A/ADREG3B is the A/D conversion result storage flag <ADRxRF>. This bit is set to "1" after an A/D converted value is stored. A read of register (ADRx[1:0]) will set this bit to "0".
- Bit 1 of ADREG2A/ADREG3B is the over RUN flag <OVRx>. This bit is set to "1" if a conversion result is overwritten before both conversion result storage register (ADRx[9:0]) is read. A read of a flag will clear this bit to "0."
- When reading conversion result storage registers on a byte-by-byte basis, first read upper registers and then read lower registers.
- Access to this register must be a half word or word access.

A/D Conversion Result Register 4C

ADREG4C		15	14	13	12	11	10	9	8
	bit Symbol	ADR4[9:2]							
	Read/Write	R							
	After reset	0							
	Function	Store upper 8 bits of A/D conversion result							
		7	6	5	4	3	2	1	0
	bit Symbol	ADR4[1:0]						OVR4	ADR4RF
	Read/Write	R		R				R	R
	After reset	0		0				0	0
	Function	Store lower 2 bits of A/D conversion result		"0" can be read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1:Presence of conversion result

A/D Conversion Result Register 5D

ADREG5D		15	14	13	12	11	10	9	8
	bit Symbol	ADR5[9:2]							
	Read/Write	R							
	After reset	0							
	Function	Store upper 8 bits of A/D conversion result							
		7	6	5	4	3	2	1	0
	bit Symbol	ADR5[1:0]						OVR5	ADR5RF
	Read/Write	R		R				R	R
	After reset	0		0				0	0
	Function	Store lower 2 bits of A/D conversion result		"0" can be read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1:Presence of conversion result

- Bit 0 of ADREG4C/ADREG5D is the A/D conversion result storage flag <ADRxRF>. This bit is set to "1" after an A/D converted value is stored. A read of register (ADRx[1:0]) will set this bit to "0".
- Bit 1 of ADREG4C/ADREG5D is the over RUN flag <OVRx>. This bit is set to "1" if a conversion result is overwritten before both conversion result storage register (ADRx[9:0]) is read. A read of a flag will clear this bit to "0."
- When reading conversion result storage registers on a byte-by-byte basis, first read upper registers and then read lower registers.
- Access to this register must be a half word or word access.

A/D Conversion Result Register 6E

ADREG6E

	15	14	13	12	11	10	9	8
bit Symbol	ADR6[9:2]							
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							
	7	6	5	4	3	2	1	0
bit Symbol	ADR6[1:0]						OVR6	ADR6RF
Read/Write	R		R				R	R
After reset	0		0				0	0
Function	Store lower 2 bits of A/D conversion result		"0" can be read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1:Presence of conversion result

A/D Conversion Result Register 7F

ADREG7F

	15	14	13	12	11	10	9	8
bit Symbol	ADR7[9:2]							
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							
	7	6	5	4	3	2	1	0
bit Symbol	ADR7[1:0]						OVR7	ADR7RF
Read/Write	R		R				R	R
After reset	0		0				0	0
Function	Store lower 2 bits of A/D conversion result		"0" can be read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1:Presence of conversion result

- Bit 0 of ADREG6E/ADREG7F is the A/D conversion result storage flag <ADRxRF>. This bit is set to "1" after an A/D converted value is stored. A read of register (ADRx[1:0]) will set this bit to "0".
- Bit 1 of ADREG6E/ADREG7F is the over RUN flag <OVRx>. This bit is set to "1" if a conversion result is overwritten before both conversion result storage register (ADRx[9:0]) is read. A read of a flag will clear this bit to "0."
- When reading conversion result storage registers on a byte-by-byte basis, first read upper registers and then read lower registers.
- Access to this register must be a half word or word access.

A/D Conversion Result Register SP

ADREGSP		15	14	13	12	11	10	9	8
	bit Symbol	ADRSP[9:2]							
	Read/Write	R							
	After reset	0							
	Function	Store upper 8 bits of A/D conversion result							
		7	6	5	4	3	2	1	0
	bit Symbol	ADRSP[1:0]						OVRSP	ADRSPRF
	Read/Write	R		R				R	R
	After reset	0		0				0	0
	Function	Store lower 2 bits of A/D conversion result		"0" can be read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1:Presence of conversion result

- Bit 0 of the ADREGSP is the A/D conversion result storage flag <ADRxRF>. It is set to "1" after an A/D converted value is stored. A read of register (ADRx[1:0]) will set this bit to "0."
- Bit 1 of the ADREGSP is the over RUN flag <OVRx>. It is set to "1" if a conversion result is overwritten before both conversion result storage register (ADRx[9:0]) is read. A read of a flag will clear this bit to "0".
- When reading conversion result storage registers on a byte-by-byte basis, first read upper registers and then read lower registers.
- Access to this register must be a half word or word access.

A/D Conversion Result Comparison Register 0

ADCMP0

	15	14	13	12	11	10	9	8
bit Symbol	ADCOM0[9:2]							
Read/Write	R/W							
After reset	0							
Function	Store upper 8 bits of A/D conversion result comparison							
	7	6	5	4	3	2	1	0
bit Symbol	ADCMO0[1:0]							
Read/Write	R/W		R					
After reset	0		0					
Function	Store lower 2 bits of A/D conversion result comparison		"0" is read.					

A/D Conversion Result Comparison Register 1

ADCMP1

	15	14	13	12	11	10	9	8
bit Symbol	ADCOM1[9:2]							
Read/Write	R/W							
After reset	0							
Function	Store upper 8 bits of A/D conversion result comparison							
	7	6	5	4	3	2	1	0
bit Symbol	ADCOM1[1:0]							
Read/Write	R/W		R					
After reset	0		0					
Function	Store lower 2 bits of A/D conversion result comparison		"0" is read.					

(Note) To set or change a value in this register, the AD monitor function must be disabled (ADMOD3, 5 <ADOBSVx>="0").

3.15.3 Conversion Clock

● The conversion time is calculated by the 46 conversion clock.

A/D Conversion Clock Setting Register

ADCLK

|		7	6	5	4	3	2	1	0
	bit Symbol	TSH					ADCLK		
	Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
	After reset	1	0	0	0	0	0	0	0
	Function	Select the A/D sample hold time 1000: 8 conversion clock 1001: 16 conversion clock 1010: 24 conversion clock 1011: 32 conversion clock 0011: 64 conversion clock 1100: 128 conversion clock 1101: 512 conversion clock The setup other than those above: reserved				"0" is read.	Select the A/D prescaler output (Note) 000: fc 001: fc／2 010: fc／4 011: fc／8 100: fc／16 101～111: Reserved		
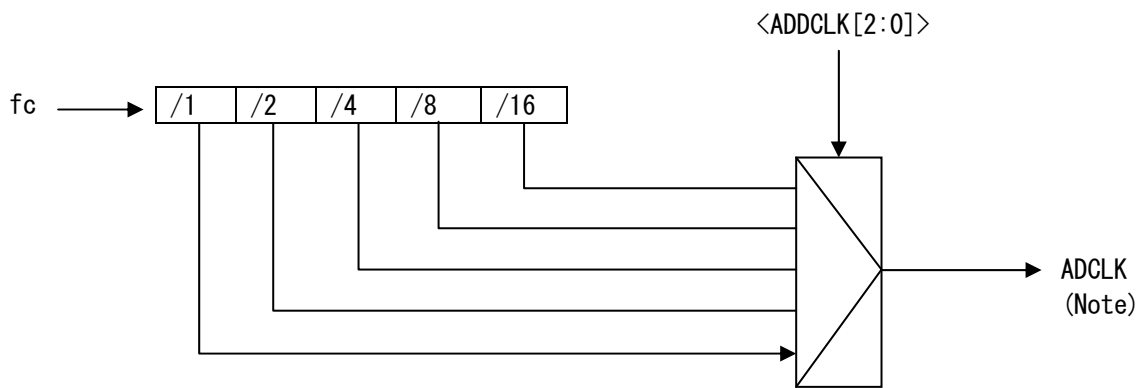
Fig.3.16.2　A/D Conversion clock(ADCLK）

(Note) **ADCLK must not exceed 40 MHz.**
For example, if fosc=12MHz and PLL=4x, then fc=48MHz. In this case, ADCLK<ADCLK[2:0]> must be set to a value other than "000".

Table 3.16.1   A/D Conversion Time   (minimum conversion time : 46clock)

Prescalar [ADCLK2:0]	S/H time	tconv. (conversion time) fc=48MHz
000 (fc)	Conversion clk*8	Setting prohibition
	Conversion clk*16	
	Conversion clk*24	
	Conversion clk*32	
	Conversion clk*64	
	Conversion clk*128	
	Conversion clk*512	
001 (fc/2)	Conversion clk*8	1.92μs
	Conversion clk*16	2.25μs
	Conversion clk*24	2.58μs
	Conversion clk*32	2.92μs
	Conversion clk*64	4.25μs
	Conversion clk*128	6.92μs
	Conversion clk*512	22.92μs
010 (fc/4)	Conversion clk*8	3.83μs
	Conversion clk*16	4.50μs
	Conversion clk*24	5.17μs
	Conversion clk*32	5.83μs
	Conversion clk*64	8.50μs
	Conversion clk*128	13.83μs
	Conversion clk*512	45.83μs
011 (fc/8)	Conversion clk*8	7.67μs
	Conversion clk*16	9.00μs
	Conversion clk*24	10.33μs
	Conversion clk*32	11.67μs
	Conversion clk*64	17.00μs
	Conversion clk*128	27.67μs
	Conversion clk*512	91.67μs
100 (fc/16)	Conversion clk*8	15.33μs
	Conversion clk*16	18.00μs
	Conversion clk*24	20.67μs
	Conversion clk*32	23.33μs
	Conversion clk*64	34.00μs
	Conversion clk*128	55.33μs
	Conversion clk*512	183.33μs

(Note) Please do not change the analog to digital conversion clock setting during the analog to digital conversion.

3.15.4　Description of Operation

3.15.4.1　Analog Reference Voltage

The "H" level of the analog reference voltage shall be applied to the VREFH pin, and the "L" level shall be applied to the VREFL pin. By writing "0" to the ADMOD1<VREFON> bit, a switched-on state of VREFH-VREFL can be turned into a switched-off state. To start AD conversion, make sure that you first write "1" to the <VREFON> bit, wait for 3 µs during which time the internal reference voltage should stabilize, and then write "1" to the ADMOD0<ADS> bit.

3.15.4.2　Selecting the Analog Iput Channel

How the analog input channel is selected is different depending on A/D converter operation mode used.

(1) Normal AD conversion mode

- If the analog input channel is used in a fixed state (ADMOD0<SCAN>="0"):

One channel is selected from analog input pins AIN0 through AIN15 by setting ADMOD1<ADCH[3:0]> to an appropriate setting.

- If the analog input channel is used in a scan state (ADMOD0<SCAN>="1"):

One scan mode is selected from 16 scan modes by setting ADMOD1<ADCH[3:0]> and ADSCN to appropriate settings.

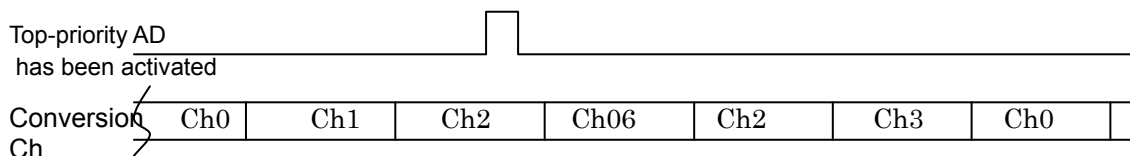（2）Top-priority AD conversion mode

One channel is selected from analog input pins AIN0 through AIN15 by setting ADMOD2<HPADCH[3:0]> to an appropriate setting.

After a reset, ADMOD0<SCAN> is initialized to "0" and ADMOD1<ADCH[3:0]> is initialized to "0000." This initialization works as a trigger to select a fixed channel input through the AN0 pin. The pins that are not used as analog input channels can be used as ordinary input ports.

If top-priority AD conversion is activated during normal AD conversion, normal AD conversion is discontinued, top-priority AD conversion is executed and completed, and then normal AD conversion is resumed.

Example:

A case in which repeat-scan conversion is ongoing at channels AIN0 through AIN3 with ADMOD0<REPEAT:SCAN> set to "11" and ADMOD1<ADCH[3:0]> set to 0011, and top-priority AD conversion has been activated at AIN6 with ADMOD2<HPADCH[3:0]>=0110.

Top-priority AD has been activated							
Conversion Ch	Ch0	Ch1	Ch2	Ch06	Ch2	Ch3	Ch0

3.15.4.3 Starting A/D Conversion

Two types of A/D conversion are supported: normal AD conversion and top-priority AD conversion. Normal AD conversion is software activated by setting ADMOD0<ADS> to "1." Top-priority AD conversion is software activated by setting ADMOD2<HPADCE> to "1." 4 operation modes are made available to normal AD conversion. In performing normal AD conversion, one of these operation modes must be selected by setting ADMOD0<2:1> to an appropriate setting. For top-priority AD conversion, only one operation mode can be used: fixed channel single conversion mode. Normal AD conversion can be activated using the HW activation source selected by ADMOD3<ADHS>, and top-priority AD conversion can be activated using the HW activation source selected by ADMOD3<HADHS>. If this bit is "0," normal and top-priority AD conversions are activated in response to the input of a falling edge through the ADTRGn pin. If this bit is "1," normal AD conversion is activated in response to TB6RG0 generated by the 16-bit timer 6, and top-priority AD conversion is activated in response to TB5RG0 generated by the 16-bit timer 5. Software activation is still valid even after H/W activation has been authorized.

(Note)	**When an external trigger is used for the HW activation source of a top priority analog to digital conversion, an external trigger cannot usually be set for activating analog to digital conversion HW.**

(Note)	**The TMPM322 disables the external trigger used for H/W activation. Therefore "0" cannot be set to <HADHS> and <ADHS>.**

When normal A/D conversion starts, the A/D conversion Busy flag (ADMOD0<ADBF>) showing that A/D conversion is under way is set to "1." When top-priority A/D conversion starts, the A/D conversion Busy flag (ADMOD2<ADBFHP>) showing that A/D conversion is under way is set to "1." At that time, the value of the Busy flag for normal A/D conversion before the start of top-priority A/D conversion is retained. The value of the conversion completion flag EOCFN for normal A/D conversion before the start of top-priority A/D conversion can also be retained.

(Note)	**Normal A/D conversion must not be activated when top-priority A/D conversion is under way. In that case, the top-priority A/D conversion completion flag cannot be set, and the flag for previous normal A/D conversion cannot be cleared.**

To reactivate normal A/D conversion while the conversion is under way, a software reset (ADMOD4<ADRST[1:0]>) must be performed before starting A/D conversion. The HW activation method must not be used to reactivate normal A/D conversion.

If ADMOD2<HPADCE> is set to "1" during normal A/D conversion, ongoing A/D conversion is discontinued and top-priority A/D conversion starts; specifically, A/D conversion (fixed channel single conversion) is executed for a channel designated by ADMOD2<3:0>. After the result of this top-priority A/D conversion is stored in the storage register ADREGSP, normal A/D conversion is resumed.

If HW activation of top-priority A/D conversion is authorized during normal A/D conversion, ongoing A/D conversion is discontinued when requirements for activation using a resource are met, and top-priority A/D conversion (fixed channel single conversion) starts for a channel designated by ADMOD2<3:0>. After the result of this top-priority A/D conversion is stored in the storage register ADREGSP, normal A/D conversion is resumed.

3.15.4.4   A/D Conversion Modes and A/D Conversion Completion Interrupts

For A/D conversion, the following four operation modes are supported. For normal A/D conversion, an operation mode can be selected by setting ADMOD0<2:1> to an appropriate setting. For top-priority A/D conversion, the fixed channel single conversion mode is automatically selected, irrespective of the ADMOD0<2:1> setting.

- Fixed channel single conversion mode
- Channel scan single conversion mode
- Fixed channel repeat conversion mode
- Channel scan repeat conversion mode

（1）Normal A/D conversion

An operation mode is selected with ADMOD0<REPEAT, SCAN>. As A/D conversion starts, ADMOD0<ADBFN> is set to "1." When specified A/D conversion is completed, the A/D conversion completion interrupt (INTAD) is generated, and ADMOD0<EOCF> showing the completion of A/D conversion is set to "1." If <REPEAT>="0," <ADBFN> returns to "0" concurrently with the setting of EOCF. If <REPEAT> is set to "1," <ADBFN> remains at "1" and A/D conversion continues.

1. Fixed channel single conversion mode

If ADMOD0 <REPEAT, SCAN> is set to "00," A/D conversion is performed in the fixed channel single conversion mode.

In this mode, A/D conversion is performed once for one channel selected. After A/D conversion is completed, ADMOD0<EOCF> is set to "1," ADMOD0<ADBF> is cleared to "0," and the interrupt request INTAD is generated. <EOCF> is cleared to "0" upon read.

2. Channel scan single conversion mode

If ADMOD0 <REPET,SCAN> is set to "01," A/D conversion is performed in the channel scan single conversion mode.

In this mode, A/D conversion is performed once for each scan channel selected. After A/D scan conversion is completed, ADMOD0<EOCF> is set to "1," ADMOD0<ADBF> is cleared to "0," and the interrupt request INTAD is generated. <EOCF> is cleared to "0" upon read.

3. Fixed channel repeat conversion mode

If ADMOD0<REPEAT,SCAN> is set to "10," A/D conversion is performed in fixed channel repeat conversion mode.

In this mode, A/D conversion is performed repeatedly for one channel selected. After A/D conversion is completed, ADMOD <EOCF> is set to "1." ADMOD0 <ADBF> is not cleared to "0." It remains at "1." The timing with which the interrupt request INTAD is generated can be selected by setting ADMOD0 <ITM[1:0]> to an appropriate setting. <EOCF> is set with the same timing as this interrupt INTAD is generated.

<EOCF> is cleared to "0" upon read.

With <ITM[1:0]> set to "00," an interrupt request is generated each time one A/D conversion is completed. In this case, the conversion results are always stored in the storage register ADREG08. After the conversion result is stored, EOCF changes to "1."

With <ITM[1:0]> set to "01," an interrupt request is generated each time four A/D conversion are completed. In this case, the conversion results are sequentially stored in storage registers ADREG08 through ADREG3B. After the conversion results are stored in ADREG3B, <EOCF> is set to "1," and the storage of subsequent conversion results starts from ADREG08. <EOCF> is cleared to "0" upon read.

With <ITM[1:0]> set to "10," an interrupt request is generated each time eight A/D conversions are completed. In this case, the conversion results are sequentially stored in storage registers ADREG08 through ADREG7F. After the conversion results are stored in ADREG7F, <EOCF> is set to "1," and the storage of subsequent conversion results starts from ADREG08.

<EOCF> is cleared to "0" upon read.

4. Channel scan repeat conversion mode

If ADMOD0 <REPEAT, SCAN> is set to "11," A/D conversion is performed in the channel scan repeat conversion mode.

In this mode, A/D conversion is performed repeatedly for a scan channel selected. Each time one A/D scan conversion is completed, ADMOD0 <EOCF> is set to "1," and the interrupt request INTAD is generated. ADMOD0 <ADBF> is not cleared to "0." It remains at "1." <EOCF> is cleared to "0" upon read.

To stop the A/D conversion operation in the repeat conversion mode (modes described in 3. and 4. above), write "0" to ADMOD0 <REPEAT>. When ongoing A/D conversion is completed, the repeat conversion mode terminates, and ADMOD0 <ADBF> is set to "0."

Before switching from one mode to standby mode (such standby modes as IDLE, STOP, etc.), check that A/D conversion is not being executed. If A/D conversion is under way, you must stop it or wait until it is completed.

（2）Top-priority A/D conversion

Top-priority A/D conversion is performed only in fixed channel single conversion mode. The ADMOD0<REPEAT, SCAN> setting has no relevance to the top-priority A/D conversion operations or preparations. As activation requirements are met, A/D conversion is performed only once for a channel designated by ADMOD2<HPADCH[3:0]>. After the A/D conversion is completed, the top-priority A/D conversion completion interrupt is generated, ADMOD2<EOCFHP> is set to "1," and <ADBFHP> returns to "0." The EOCFHP Flag is cleared upon read.

Table 3. 16. 3 Relationships among A/D Conversion Modes, Interrupt Generation Timings and Flag Operations

Conversion mode	Interrupt generation timing	EOCF setting timing (see Note)	ADBF (after the interrupt is generated)	ADMOD0		
				ITM[1:0]	REPEAT	SCAN
Fixed channel single conversion	After conversion is completed	After conversion is completed	0	−	0	0
Fixed channel repeat conversion	Each time one conversion is completed	After one conversion is completed	1	00	1	0
	Each time four conversions are completed	After four conversions are completed	1	01		
	Each time eight conversions are completed	After eight conversions are completed	1	10		
Channel scan single conversion	After scan conversion is completed	After scan conversion is completed	0	−	0	1
Channel scan repeat conversion	Each time one scan conversion is completed	After one scan conversion is completed	1	−	1	1

(Note)   EOCF is cleared upon read.

3.15.4.5 High-priority Conversion Mode

By interrupting ongoing normal A/D conversion, top-priority A/D conversion can be performed. Top-priority A/D conversion can be software activated by setting ADMOD2<HPADCE> to "1" or it can be activated using the HW resource by setting ADMOD3<7:6> to an appropriate setting. If top-priority A/D conversion has been activated during normal A/D conversion, ongoing normal A/D conversion is interrupted, and single conversion is performed for a channel designated by ADMOD2<3:0>. The result of single conversion is stored in ADREGSP, and the top-priority A/D conversion interrupt is generated. After top-priority A/D conversion is completed, normal A/D conversion is resumed; the status of normal A/D conversion immediately before being interrupted is maintained. Top-priority A/D conversion activated while top-priority A/D conversion is under way is ignored.

For example, if channel repeat conversion is activated for channels AN0 through AN7 and if <HPADCE> is set to "1" during AN3 conversion, AN3 conversion is suspended, and conversion is performed for a channel designated by <HPADC3:0>. After the result of conversion is stored in ADREGSP, channel repeat conversion is resumed, starting from AN3.

> (Note) The top-priority AD conversion completion interrupt (INTADHP) cannot be used for generating a DMA transfer request. To generate a DMA transfer request, use the AD conversion completion interrupt (INTAD).

3.15.4.6 A/D Monitor Function

If ADMOD3,5<ADOBSVx> is set to "1," the A/D monitor function is enabled. If the value of the conversion result storage register specified by REGSx<3:0> becomes larger or smaller ("larger" or "smaller" to be designated by ADOBIC) than the value of a comparison register, the A/D monitor function interrupt is generated. This comparison operation is performed each time a result is stored in a corresponding conversion result storage register, and the interrupt is generated if the conditions are met. Because storage registers assigned to perform the A/D monitor function are usually not read by software, overrun flag <OVRx> is always set and the conversion result storage flag <ADRxRF> is also set. To use the A/D monitor function, therefore, a flag of a corresponding conversion result storage register must not be used.

3.15.4.7 Storing and Reading A/D Conversio Results

A/D conversion results are stored in upper and lower A/D conversion result registers for normal A/D conversion (ADREG08 through ADRG7F).

In fixed channel repeat conversion mode, A/D conversion results are sequentially stored in ADREG08 through ADREG7F. If <ITM[1:0[> is so set as to generate the interrupt each time one A/D conversion is completed, conversion results are stored only in ADREG08. If <ITM[1:0]> is so set as to generate the interrupt each time four A/D conversions are completed, conversion results are sequentially stored in ADREG08 through ADREG3B.

Table 3.16.4 shows analog input channels and related A/D conversion result registers.

Table 3.16.4 Analog Input Channels and Related A/D Conversion Result Registers

ADMOD1 <ADCH[3:0]>	Fixed channel repeat conversion mode (every one conversion)		Channel Scan Mode			
			ADMMOD1<ADSCN>="0" Fixed channel repeat conversion mode (every four conversions)		ADMMOD1<ADSCN>="1" Fixed channel repeat conversion mode (every eight conversions)	
	Stored Register	ch	Stored Register	ch	Stored Register	ch
0000	ADREG08	0ch	ADREG08	0ch	ADREG08	0ch
0001	ADREG19	1ch	ADREG08 ～ ADREG19	0ch～1ch	ADREG08 ～ ADREG19	0ch～1ch
0010	ADREG2A	2ch	ADREG08 ～ ADREG2A	0ch～2ch	ADREG08 ～ ADREG2A	0ch～2ch
0011	ADREG3B	3ch	ADREG08 ～ ADREG3B	0ch～3ch	ADREG08 ～ ADREG3B	0ch～3ch
0100	ADREG4C	4ch	ADREG4C	4ch	ADREG08 ～ ADREG4C	0ch～4ch
0101	ADREG5D	5ch	ADREG4C ～ ADREG5D	4ch～5ch	ADREG08 ～ ADREG5D	0ch～5ch
0110	ADREG6E	6ch	ADREG4C ～ ADREG6E	4ch～6ch	ADREG08 ～ ADREG6E	0ch～6ch
0111	ADREG7F	7ch	ADREG4C ～ ADREG7F	4ch～7ch	ADREG08 ～ ADREG7F	0ch～7ch
1000	ADREG08	8ch	ADREG08	8ch	ADREG08	8ch
1001	ADREG19	9ch	ADREG08 ～ ADREG19	8ch～9ch	ADREG08 ～ ADREG19	8ch～9ch
1010	ADREG2A	10ch	ADREG08 ～ ADREG2A	8ch～10ch	ADREG08 ～ ADREG2A	8ch～10ch
1011	ADREG3B	11ch	ADREG08 ～ ADREG3B	8ch～11ch	ADREG08 ～ ADREG3B	8ch～11ch
1100	ADREG4C	12ch	ADREG4C	12ch	ADREG08 ～ ADREG4C	8ch～12ch
1101	ADREG5D	13ch	ADREG4C ～ ADREG5D	12ch～13ch	ADREG08 ～ ADREG5D	8ch～13ch
1110	ADREG6E	14ch	ADREG4C ～ ADREG6E	12ch～14ch	ADREG08 ～ ADREG6E	8ch～14ch
1111	ADREG7F	15ch	ADREG4C ～ ADREG7F	12ch～15ch	ADREG08 ～ ADREG7F	8ch～15ch

3.15.4.8 Data Polling

To process A/D conversion results without using interrupts, ADMOD0<EOCF> must be polled. If this flag is set, conversion results are stored in a specified A/D conversion result register. After confirming that this flag is set, read that conversion result storage register. In reading the register, make sure that you first read upper bits and then lower bits to detect an overrun. If OVRx is "0" and ADRxRF is "1" in lower bits, a correct conversion result has been obtained.

Operating precautions for the AD converter

AD conversion results may vary because of a change in the supply voltage or surrounding noise effects.
The AD conversion accuracy may be lower if any of the following conditions occur during AD conversion:
  An input to a port which is also used for AD input changes.
  An output from a port which is also used for AD input changes.
  Output current of a port configured as an output port changes.
As a way to counter this issue, get the average value from the results of AD conversions performed for multiple times.

3.16   Real Time Clock (RTC)

3.16.1   Functions

1) Clock (hour, minute and second)
2) Calendar (month, week, date and leap year)
3) Selectable 12 (am/ pm) and 24 hour display
4) Time adjustment $+$ or $-$ 30 seconds (by software)
5) Alarm (alarm output)
6) Alarm interrupt

3.16.2   Block Diagram
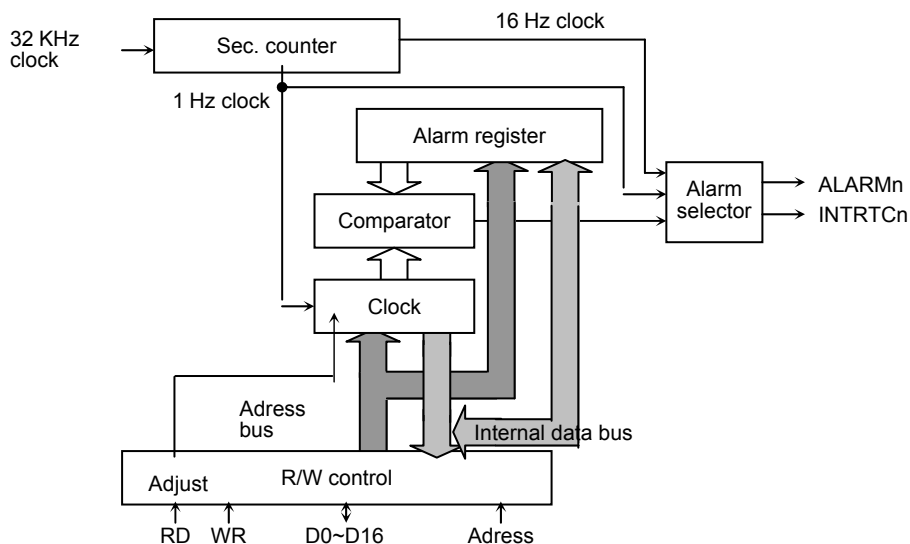
Fig. 3.16.1 Block Diagram

(Note 1) Western calendar year column:
This product uses only the final two digits of the year. The year following 99 is 00 years. Please take into account the first two digits when handling years in the western calendar.

(Note 2) Leap year:
A leap year is divisible by 4 excluding a year divisible by 100; the year divisible by 100 is not considered to be a leap year. Any year divisible by 400 is a leap year. This product is considered the year divisible by 4 to be a leap year and does not take into account the above exceptions. It needs adjustments for the exceptions.

3.16.3 Control Registers

Reset operation initializes the following registers:

● RTCPAGER<PAGE>,<ADJUST>,<INTENA>
● RTCRESTR<RSTALM>,<RSTTMR>,<DIS16HZ>,<DIS1HZ>

Other clock-related registers are not initialized by reset operation.

Before starting the RTC, set the time, month, day, day of the week, year and leap year in the relevant registers.

Caution is required in setting clock data, adjusting seconds or resetting the clock. Refer to "3.16.5.3 Entering the Low Power Consumption Mode".

Table 3.16.1   PAGE0 (clock function) register

Symbol	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function	Read/Write
RTCSECR	0x400F_3000		40 sec	20 sec	10 sec	8 sec	4 sec	2 sec	1 sec	Second column	R/W
RTCMINR	0x400F_3001		40 min	20 min	10 min	8 min	4 min	2 min	1 min	Minute column	R/W
RTCHOURR	0x400F_3002			20 hours /PM/AM	10 hours	8 hours	4 hours	2 hours	1 hours	Hour column	R/W
RTCDAYR	0x400F_3004						W2	W1	W0	Day of the week column	R/W
RTCDATER	0x400F_3005			Day 20	Day 10	Day 8	Day 4	Day 2	Day 1	Day column	R/W
RTCMONTHR	0x400F_3006				Month 10	Month 8	Month 4	Month 2	Month 1	Month column	R/W
RTCYEARR	0x400F_3007	Year 80	Year 40	Year 20	Year 10	Year 8	Year 4	Year 2	Year 1	Year column (lower two columns)	R/W
RTCPAGER	0x400F_3008	Interrupt enable			Adjustment function	Clock enable	Alarm enable		PAGE setting	PAGE register	W, R/W
RTCRESTR	0x400F_300C	1Hz enable	16Hz enable	Clock reset	Alarm reset	Always write "0".				Reset register	W only

(Note) Reading RTCSECR, RTCMINR, RTCHOURR, RTCDAYR, RTCMONTHR, RTCYEARR of PAGE0 captures the current state.

Table 3.16.2 PAGE1 (alarm function) registers

Symbol	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function	Read/Write
RTCSECR	0x400F_3000										
RTCMINR	0x400F_3001		40 min	20 min	10 min	8 min	4 min	2 min	1 min	Minute column	R/W
RTCHOURR	0x400F_3002			20 hours /PM/AM	10 hours	8 hours	4 hours	2 hours	1 hour	Hour column	R/W
RTCDAYR	0x400F_3004						W2	W1	W0	Day of the week column	R/W
RTCDATER	0x400F_3005			Day 20	Day 10	Day 8	Day 4	Day 2	Day 1	Day column	R/W
RTCMONTHR	0x400F_3006							24/12		24-hour clock mode	R/W
RTCYEARR	0x400F_3007							Leap-year setting		Leap-year mode	R/W
RTCPAGER	0x400F_3008	Interrupt enable			Adjustment function	Clock enable	Alarm enable		PAGE setting	PAGE register	W,R/W
RTCRESTR	0x400F_300C	1Hz enable	16Hz enable	Clock reset	Alarm reset	Always write "0".				Reset register	W only

> **(Note 1) Reading RTCMINR, RTCHOURR, RTCDAYR, RTCMONTHR, YEARR of PAGE1 captures the current state.**
>
> **(Note 2) RTCSECR, RTCMINR, RTCHOURR, RTCDAYR, RTCDATER, RTCMONTHR, RTCYEARR of PAGE0 and RTCYEARR of PAGE1 (for leap year) must be read twice and compare the data captured.**

3.16.4 Detailed Description of Control Register

3.16.4.1 Secod column register (for PAGE0 only)

		7	6	5	4	3	2	1	0
RTCSECR	Bit symbol	—	SE						
	Read/Write	R	R/W						
	After reset	0	Undefined						
	Function	"0" is read.	40 sec. column	20 sec. column	10 sec. column	8 sec. column	4 sec. column	2 sec. column	1 sec. column

<u>Seconds Register Settings</u>

0	0	0	0	0	0	0	0 sec
0	0	0	0	0	0	1	1 sec
0	0	0	0	0	1	0	2 sec
0	0	0	0	0	1	1	3 sec
0	0	0	0	1	0	0	4 sec
0	0	0	0	1	0	1	5 sec
0	0	0	0	1	1	0	6 sec
0	0	0	0	1	1	1	7 sec
0	0	0	1	0	0	0	8 sec
0	0	0	1	0	0	1	9 sec
0	0	1	0	0	0	0	10 sec
			:				
0	0	1	1	0	0	1	19 sec
0	1	0	0	0	0	0	20 sec
			:				
0	1	0	1	0	0	1	29 sec
0	1	1	0	0	0	0	30 sec
			:				
0	1	1	1	0	0	1	39 sec
1	0	0	0	0	0	0	40 sec
			:				
1	0	0	1	0	0	1	49 sec
1	0	1	0	0	0	0	50 sec
			:				
1	0	1	1	0	0	1	59 sec

Note) The setting other than listed above is prohibited.

3.16.4.2 Minute column register(for PAGE0/1)

RTCMINR		7	6	5	4	3	2	1	0
	Bit symbol	—	MI						
	Read/Write	R	R/W						
	After reset	0	Undefined						
	Function	"0" is read	40 min. column	20 min. column	10 min. column	8 min. column	4 min. column	2 min. column	1 min. column

Minutes Register Settings

0	0	0	0	0	0	0	0 min
0	0	0	0	0	0	1	1 min
0	0	0	0	0	1	0	2 min
0	0	0	0	0	1	1	3 min
0	0	0	0	1	0	0	4 min
0	0	0	0	1	0	1	5 min
0	0	0	0	1	1	0	6 min
0	0	0	0	1	1	1	7 min
0	0	0	1	0	0	0	8 min
0	0	0	1	0	0	1	9 min
0	0	1	0	0	0	0	10 min

:

0	0	1	1	0	0	1	19 min
0	1	0	0	0	0	0	20 min

:

0	1	0	1	0	0	1	29 min
0	1	1	0	0	0	0	30 min

:

0	1	1	1	0	0	1	39 min
1	0	0	0	0	0	0	40 min

:

1	0	0	1	0	0	1	49 min
1	0	1	0	0	0	0	50 min

:

1	0	1	1	0	0	1	59 min

Note)   The setting other than listed above is prohibited.

3.16.4.3 Hour column register(for PAGE0/1)

1. 24-hour clock mode (RTCMONTHR<MO0> ="1")

RTCHOURR		7	6	5	4	3	2	1	0
	Bit symbol	—		HO					
	Read/Write	R		R/W					
	After reset	0		Undefined					
	Function	"0"is read.		20 hour column	10 hour column	8 hour column	4 hour column	2 hour column	1 hour column

Hours Register Settings

0	0	0	0	0	0	0 o' clock
0	0	0	0	0	1	1 o' clock
0	0	0	0	1	0	2 o' clock
			:			
0	0	1	0	0	0	8 o' clock
0	0	1	0	0	1	9 o' clock
0	1	0	0	0	0	10 o' clock
			:			
0	1	1	0	0	1	19 o' clock
1	0	0	0	0	0	20 o' clock
			:			
1	0	0	0	1	1	23 o' clock

Note) The setting other than listed above is prohibited.

2. 12-hour clock mode(RTCMONTHR<MO0> ="0")

RTCHOURR		7	6	5	4	3	2	1	0
	Bit symbol	—		HO					
	Read/Write	R		R/W					
	After reset	0		Undefined					
	Function	"0"is read.		PM/AM	10 hour column	8 hour column	4 hour column	2 hour column	1 hour column

Hours Register Settings

0	0	0	0	0	0	0 o' clock (AM)
0	0	0	0	0	1	1 o' clock
0	0	0	0	1	0	2 o' clock
			:			
0	0	1	0	0	1	9 o' clock
0	1	0	0	0	0	10 o' clock
0	1	0	0	0	1	11 o' clock
1	0	0	0	0	0	0 o' clock (PM)
1	0	0	0	0	1	1 o' clock

Note) The setting other than listed above is prohibited.

3.16.4.4 Day of thr week column register(for PAGE0/1）

RTCDAYR		7	6	5	4	3	2	1	0
	Bit symbol	—					WE		
	Read/Write	R					R/W		
	After reset	0					Undefined		
	Function	"0" is read.					W2	W1	W0

Day of the Week Register Settings

0	0	0	Sunday
0	0	1	Monday
0	1	0	Tuesday
0	1	1	Wednesday
1	0	0	Thursday
1	0	1	Friday
1	1	0	Saturday

Note)   The setting other than listed above is
prohibited.

3.16.4.5 Day column register(for PAGE0/1）

RTCDATER		7	6	5	4	3	2	1	0
	Bit symbol	—		DA					
	Read/Write	R		R/W					
	After reset	0		Undefined					
	Function	"0" is read.		Day 20	Day 10	Day 8	Day 4	Day 2	Day 1

Date Register Settings

0	0	0	0	0	0	0
0	0	0	0	0	1	1st day
0	0	0	0	1	0	2nd day
0	0	0	0	1	1	3rd day
0	0	0	1	0	0	4th day
:						
0	0	1	0	0	1	9th day
0	1	0	0	0	0	10th day
0	1	0	0	0	1	11th day
:						
0	1	1	0	0	1	19th day
1	0	0	0	0	0	20th day
:						
1	0	1	0	0	1	29th day
1	1	0	0	0	0	30th day
1	1	0	0	0	1	31st day

Note 1)   The setting other than listed above is prohibited.

Note 2)   Do not set for non-existent days (e.g.: 30th Feb)

3.16.4.6 Month column register(for PAGE0 only)

	7	6	5	4	3	2	1	0
RTCMONTHR Bit symbol	—			MO				
Read/Write	R			R/W				
After reset	0			Undefined				
Function	"0" is read.			10 months	8 months	4 months	2 months	1 month

Month Register Settings

0	0	0	0	1	January
0	0	0	1	0	February
0	0	0	1	1	March
0	0	1	0	0	April
0	0	1	0	1	May
0	0	1	1	0	June
0	0	1	1	1	July
0	1	0	0	0	August
0	1	0	0	1	September
1	0	0	0	0	October
1	0	0	0	1	November
1	0	0	1	0	December

Note)   The setting other than listed above is prohibited.

3.16.4.7 Selection of 24-hour clock or 12-hour clock (for PAGE1 only)

	7	6	5	4	3	2	1	0
RTCMONTHR Bit symbol	—							MO0
Read/Write	R							R/W
After reset	0							Undefined
Function	"0" is read.							1: 24-hour 0: 12-hour

（**Note**）**Do not change the RTCMONTHR<MO0> bit while the RTC is in operation (RTCPAGER<ENATMR>="1").**

3.16.4.8　Year column register(for PAGE0 only）

	7	6	5	4	3	2	1	0
Bit symbol	YE							
Read/Write	R/W							
After reset	Undefined							
Function	80 years	40 years	20 years	10 years	8 years	4 years	2 years	1 year

RTCYEARR

Year Register Settings

0	0	0	0	0	0	0	0	00
0	0	0	0	0	0	0	1	01 years
0	0	0	0	0	0	1	0	02 years
0	0	0	0	0	0	1	1	03 years
0	0	0	0	0	1	0	0	04 years
0	0	0	0	0	1	0	1	05 years
:								
1	0	0	1	1	0	0	1	99 years

Note)　The setting other than listed above is prohibited.

3.16.4.9　Leap year register(for PAGE1 only）

	7	6	5	4	3	2	1	0
Bit symbol	—						LEAP	
Read/Write	R						R/W	
After reset	0						Undefined	
Function	"0" is read.						00: leap year 01: one year after leap year 10: two years after leap year 11: three years after leap year	

RTCYEARR

0	0	Current year is a leap-year.
0	1	Current year is the year following a leap-year.
1	0	Current year is two years after a leap year.
1	1	Current year is three years after a leap year

3.16.4.10   PAGE register (for PAGE0/1）

RTCPAGER		7	6	5	4	3	2	1	0
	Bit symbol	INTENA	—		ADJUST	ENATMR	ENAALM	—	PAGE
	Read/Write	R/W	R		R/W	R/W		R	R/W
	After reset	0	0		0	Undefined		0	0
A read-modify-write operation cannot be performed.	Function	INTRTC 0: Disabled 1: Enabled	"0" is read.		[Write] 0: Don't care 1: Sets ADJUST request [Read] 0: No ADJUST request 1: ADJUST requested	Clock 0: Disabled 1: Enabled	ALARM 0: Disabled 1: Enabled	"0" is read.	PAGE selection

(Note) Keep the setting order of <ENATMR>, <ENAAML> and <INTENA> as shown in the example below. Ensure an interval of time between Clock/Alarm and interrupt.

Example: Clock setting/Alarm setting

```
            7 6 5 4 3 2 1 0
RTCPAGER  ← 0 0 0 0 1 1 0 0    Enables Clock and alarm
RTCPAGER  ← 1 0 0 0 1 1 0 0    Enables interrupt
```

PAGE	0	Selects Page0
	1	Selects Page1

ADJUST	0	Don't care
	1	Adjusts seconds. The request is sampled when the sec. counter counts up. If the time elapsed is between 0 and 29 seconds, the sec. counter is cleared to "0". If the time elapsed is between 30 and 59 seconds, the min. counter is carried and sec. counter is cleared to "0". Reading this bit shows if ADJUST is requested or not.

3.16.4.11　Reset register(for PAGE0/1）

RTCRESTR		7	6	5	4	3	2	1	0
	Bit symbol	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	—	—	—	—
	Read/Write	R/W				R	R/W		
	After reset	1	1	0	0	0	1	1	1
A read-modify-write operation cannot be performed.	Function	1 Hz 0: Enabled 1: Disabled	16 Hz 0: Enabled 1: Disabled	[Write] 0:　Don't care 1:　Clock reset [Read] 0:　No RESET request 1:　RESET requested	0:　Don't care 1:　Alarm reset	"0" is read.	Write to "1".		

	0	Unused
RSTALM	1	Initializes alarm registers (Minute Column, Hour Column, Day Column and Day of the week Column) as follows. Minute: 00, Hour: 00, Day: 01, Day of the week: Sunday

	0	Unused
RSTTMR	1	Resets sec counter. Reading this bit shows if RESET is requested or not. The request is sampled using low-speed clock.

<DIS1HZ>	<DIS16HZ>	PAGER<ENAALM>	Interrupt source signal
1	1	1	Alarm
0	1	0	1Hz
1	0	0	16Hz
Others			Outputs "0".

3.16.5   Operation Description

The RTC incorporates a sec. counter that generates an 1Hz signal from a 32.768 KHz signal. The sec. counter operation must be taken into account when using the RTC.

3.16.5.1   Reading clock data

1. Using 1Hz interrupt

    The 1Hz interrupt is generated being synchronized with counting up of the sec. counter. Data can be read correctly if reading data after 1Hz interrupt occurred.

2. Using pair reading

    There is a possibility that the clock data may be read incorrectly if the internal counter operates carry during reading. To ensure correct data reading, read the clock data twice as shown below. A pair of data read successively needs to match.
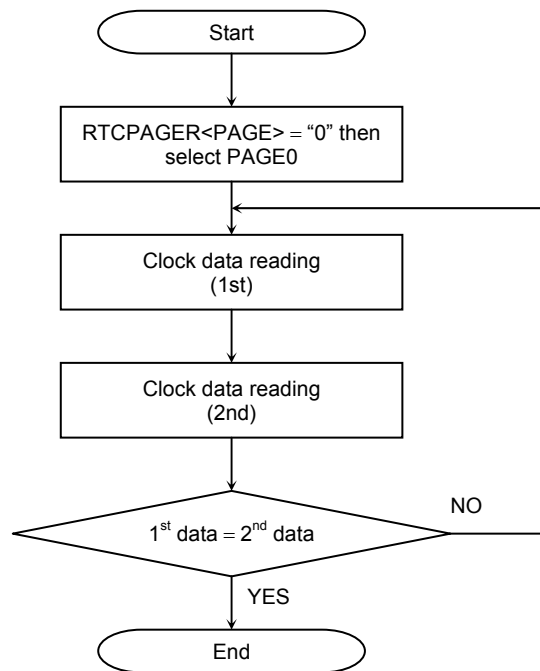
Fig. 3.16.2 Flowchart of the clock data reading

3.16.5.2 Writing clock data

A carry during writing ruins correct data writing. The following procedure ensures the correct data writing.

1. Using 1Hz interrupt

    The 1Hz interrupt is generated being synchronized with counting up of the sec. counter. If data is written in the time between 1Hz interrupt and subsequent one second count, it completes correctly.

2. Resetting counter

    Write data after resetting the sec. counter.

    The 1Hz-interrupt is generated one second after enabling the interrupt subsequent to counter reset. The time must be set within one second after the interrupt.

```
                    ┌─────────────────┐
                    │      Start      │
                    └─────────────────┘
                             │
                             ▼
              ┌──────────────────────────────┐
              │ RTCPAGER<PAGE> = "0"  then    │
              │        select PAGE0           │
              └──────────────────────────────┘
                             │
                             ▼
              ┌──────────────────────────────┐
              │ RTCRESTR<RSTTMR> = "1"  then  │
              │         reset counter         │
              └──────────────────────────────┘
                             │
                             ▼
              ┌──────────────────────────────┐
              │ RTCRESTR<DIS1HZ> = "0"  then  │
              │      enable 1Hz interrupt     │
              └──────────────────────────────┘
                             │
                             ▼
                    ╱───────────────╲          NO
                   ╱  First interrupt ╲ ────────────┐
                   ╲   (After 1S)     ╱             │
                    ╲───────────────╱ ─────────────┘
                             │ YES
                             ▼
              ┌──────────────────────────────┐
              │        Time setting           │
              └──────────────────────────────┘
                             │
                             ▼
                    ┌─────────────────┐
                    │       End       │
                    └─────────────────┘
```
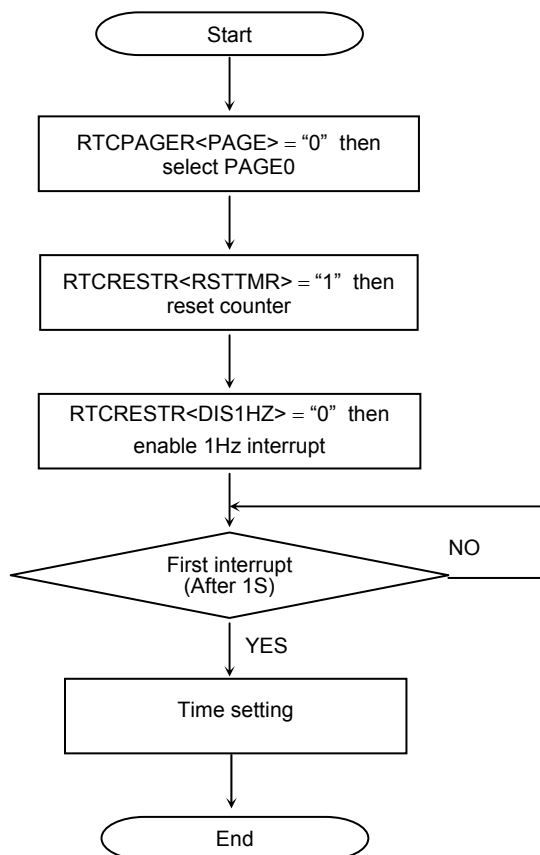
Fig. 3.16.3 Flowchart of the clock data writing

3. Disabling the clock

Writing "0" to RTCPAGER<ENATMR> disables clock operation including a carry.

Stop the clock after the 1Hz-interrupt. The sec. counter keeps counting. Set the clock again and enable the clock within one second before next 1Hz-interrupt.

```
        ┌─────────────────────┐
        │        Start        │
        └──────────┬──────────┘
                   │
        ┌──────────▼──────────┐
        │   Disabling clock   │
        └──────────┬──────────┘
                   │
        ┌──────────▼──────────┐
        │ Writing the clock data │
        └──────────┬──────────┘
                   │
        ┌──────────▼──────────┐
        │  Enabling the clock │
        └──────────┬──────────┘
                   │
        ┌──────────▼──────────┐
        │         End         │
        └─────────────────────┘
```
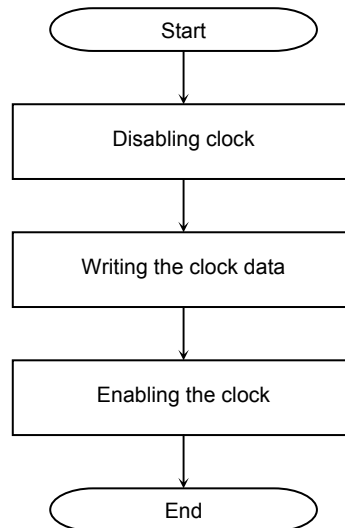
Fig. 3.16.4 Flowchart of the disabling clock

3.16.5.3   Entering the Low Power Consumption Mode

To enter SLEEP mode, in which the system clock stops, after changing clock data, adjusting seconds or resetting the clock, be sure to observe one of the following procedures:

1. After changing the clock setting registers, setting the RTCPAGER<ADJUST> bit or setting the RTCRESTR<RSTTMR> bit, wait for one second for an interrupt to be generated.
2. After changing the clock setting registers, setting the RTCPAGER<ADJUST> bit or setting the RTCRESTR<RSTTMR> bit, read the corresponding clock register values, <ADJUST> or <RSTTMR> to make sure that the setting you have made is reflected.

3.16.6   Ararm Function

By writing "1" to RTCPAGER<PAGE>, the alarm function of the PAGE1 registers is enabled. One of the following three signals is output to the ARARMn pin.

(1) "0" pulse (when the alarm register corresponds with the clock)

(2) 1Hz cycle "0" pulse

(3) 16Hz cycle "0" pulse

In any cases shown above, the INTRTCn outputs one cycle pulse of low-speed clock. It outputs the INTRTCn interrupt request simultaneously.

The INTRTCn interrupt signal is falling edge triggered. Specify the falling edge as the active state in the CG Interrupt Mode Control Register.

(1) "0" pulse (when the alarm register corresponds with the clock)

"0" pulse is output to the ARARMn pin when the values of the PAGE0 clock register and the PAGE1 alarm register correspond. The INTRTC interrupt is generated and the alarm is triggered.

<u>The alarm settings</u>

Initialize the alarm with alarm prohibited. Write "1" to RTCRESTR<RSTALM>. It makes the alarm setting to be 00 minute, 00 hour, 01 day and Sunday.

Setting alarm for min., hour, date and day is done by writing data to the relevant PAGE1 register. Enable the alarm with the RTCPAGER <ENAALM> bit. Enable the interrupt with the RTCPAGER <INTENA> bit.

The following is an example program for outputting an alarm from the ALARMn pin at noon (PM12:00) on Monday 5th.

```
                      7 6 5 4 3 2 1 0
    RTCPAGER  ←  0 0 0 0 1 0 0 1      Disables alarm, sets PAGE1
    RTCRESTR  ←  1 1 0 1 0 0 0 0      Initializes alarm
    RTCDAYR   ←  0 0 0 0 0 0 0 1      Monday
    RTCDATAR  ←  0 0 0 0 0 1 0 1      5th day
    RTCHOURR  ←  0 0 0 1 0 0 1 0      Sets 12 o'clock
    RTCMINR   ←  0 0 0 0 0 0 0 0      Sets 00 min.
    RTCPAGER  ←  0 0 0 0 1 1 0 0      Enables alarm
    RTCPAGER  ←  1 0 0 0 1 1 0 0      Enables interrupt
```

The above alarm works in synchronization with the low-speed clock. When the CPU is operating at high frequency oscillation, a maximum of one clock delay at 32 kHz (about 30µs) may occur for the time register setting to become valid.

(Note)   To make the alarm work repeatedly (e.g. every Wednesday at 12:00), next alarm must be set during the INTRTCn interrupt routine that is generated when the time set for the alarm matches the RTC count.

---

(2) 1Hz cycle "0" pulse

   The RTC outputs a "0" pulse cycle of low-speed 1Hz clock to the ALARMn pin by setting
RTCPAGER<INTENA>=1   after   setting   RTCPAGER<ENAALM>=   "0",
RTCRESTR<DIS1HZ>= "0" and <DIS16HZ>= "1". It generates an INTRTCn interrupt
simultaneously.

(3) 16Hz cycle "0" pulse

   The RTC outputs a "0" pulse cycle of low-speed 16Hz clock to the ALARMn pin by setting
RTCPAGER<INTENA>=1   after   setting   RTCPAGER<ENAALM>=   "0",
RTCRESTR<DIS1HZ>= "1" and <DIS16HZ>= "0". It generates an INTRTCn interrupt
simultaneously.

3.17 Watchdog Timer (WDT)

The watchdog timer (WDT) is for detecting malfunctions (runaways) of the CPU caused by noises or other disturbances and remedying them to return the CPU to normal operation. If the timer detects a runaway, it generates a non-maskable interrupt to notify the CPU and outputs "0" from the output pin of the watchdog timer (WDTOUTn) to notify peripherals.

The watchdog timer can reset the CPU by connecting the WDTOUT pin to internal reset.

3.17.1 Configration

Fig.3.17.1 shows the block diagram of the watchdog timer
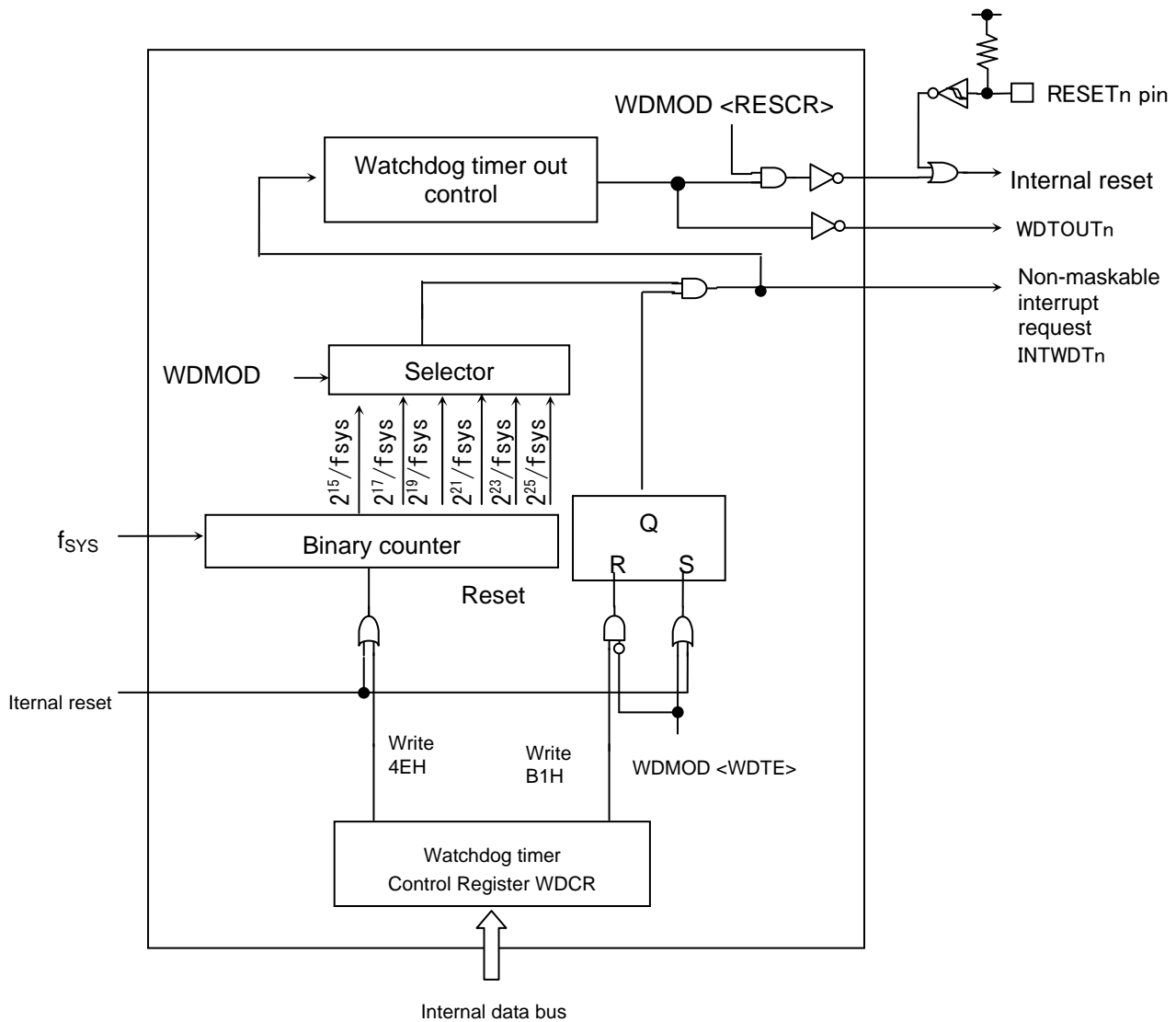
Fig. 3.17.1   Block Diagram of the Watchdog Timer

3.17.2 Watchdog Timer Interrupt

The watchdog timer consists of the binary counters and work using the f_{SYS} system clock as an input clock. The outputs produced by these binary counters are 2^{15}, 2^{17}, 2^{19}, 2^{21}, 2^{23} are 2^{25}. By selecting one of these outputs with WDMOD <WDTP>, a watchdog timer interrupt INTWDTn can be generated and the WDTOUTn is output when an overflow occurs, as shown in Fig. 3.17.2.

The INTWDTn interrupt is one of the non-maskable interrupt factors. The INTWDTn interrupt can be identified with the CGNMIFLG <NMIFLG0> bit in the clock generator

The output pin of the watchdog timer can reset the peripherals by outputting "0" caused by an overflow. The output is set to "1" if the watchdog timer is cleared (if the clear code 4EH is written to the WDCR register). The WDTOUTn pin outputs "0" at normal mode unless the clear code is written to WDCR register.
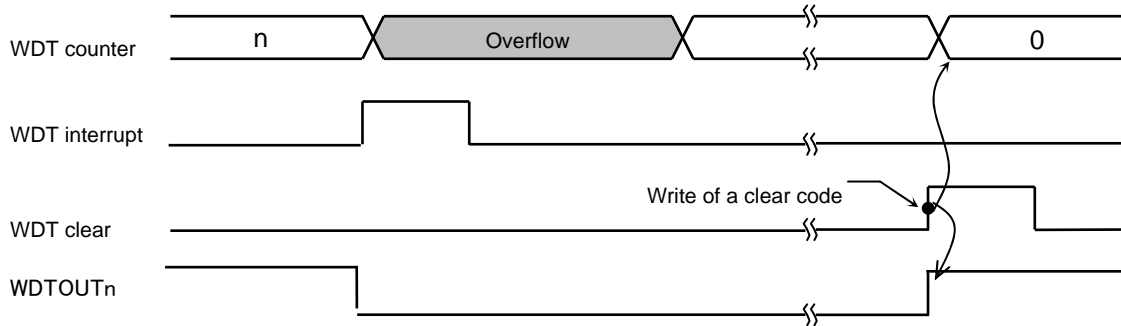
Fig. 3.17.2   Normal Mode

When an overflow occurs, resetting the chip itself is an option to choose. If the chip is reset, a reset is affected for a 32-state time, as shown in Fig. 3.17.3. If this reset is affected, the clock f_{SYS} that the clock gear generates by dividing the clock f_C of the high-speed oscillator by 1 is used as an input clock f_{SYS}.

32-state
(2.67 μs @ fosc = 12 MHz, f_C = fsys = 12 MHz)
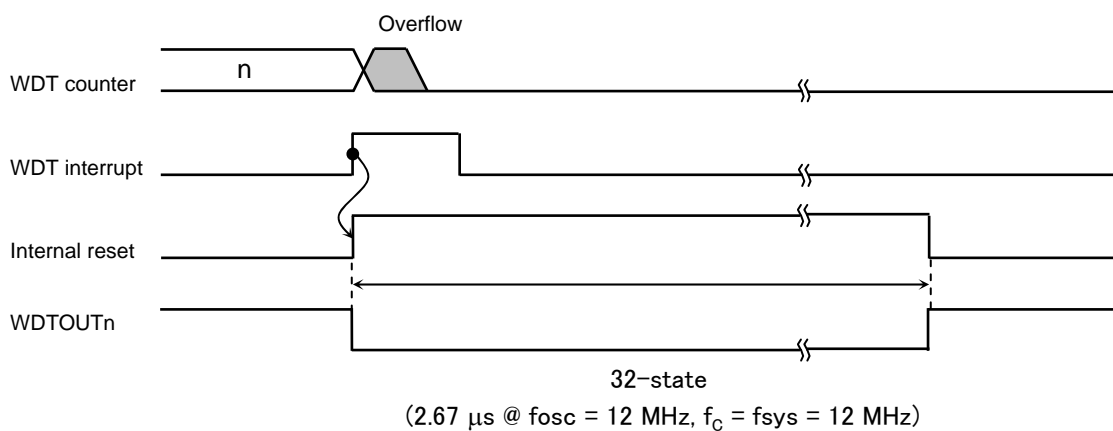
Fig. 3.17.3   Reset Mode

3.17.3 Control Registers

The watchdog timer (WDT) is controlled by two control registers WDMOD and WDCR.

3.17.3.1   Watchdog Timer Mode Register(WDMOD)

1. Specifying the detection time of the watchdog timer <WDTP[2:0]>

   This is a 2-bit register for specifying the watchdog timer interrupt time for runaway detection. When a reset is effected, this register is initialized to WDMOD <WDTP[2:0]> = "00." WDMOD<WDTP[2:0]> description shows the detection time of the watchdog timer.

2. Enabling/disabling the watchdog timer <WDTE>

   When resetting, WDMOD <WDTE> is initialized to "1" and the watchdog timer is enabled.

   To disable the watchdog timer, this bit must be set to "0" and, at the same time, the disable code (B1H) must be written to the WDCR register. This dual setting is intended to minimize the probability that the watchdog timer may inadvertently be disabled if a runaway occurs.

   To change the status of the watchdog timer from "disable" to "enable," set the <WDTE> bit to "1".

3. Watchdog timer out reset connection <RESCR>

   Setting this bit to "1" enables the watch dog timer to be reset when a runaway is detected. Since a reset initializes this bit to "1," a counter overflow causes a reset.

3.17.3.2   Watchdog Timer Control Register(WDCR)

This is a register for disabling the watchdog timer function and controlling the clearing function of the binary counter.

- Disabling control

   By writing the disable code (B1H) to this WDCR register after setting WDMOD <WDTE> to "0," the watchdog timer can be disabled.

WDMOD	← 0 − − − − − − −	Clears WDTE to "0."
WDCR	← 1 0 1 1 0 0 0 1	Writes the disable code (B1H).

- Enablong control

   Set WDMOD <WDTE> to "1".

- Watchdog timer cleaning control

   Writing the clear code (4EH) to the WDCR register clears the binary counter and allows it to resume counting.

WDCR	← 0 1 0 0 1 1 1 0	Writes the clear code (4EH)

> **(Note) Writing the disable code (BIH) clears the binary counter**.

Watchdog Timer Mode Register

WDMOD
(0x400F_2000)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP				I2WDT	RESCR	
Read/Write	R/W	R/W			R	R/W		R/W
After reset	1	0	0	0		0	1	0
Function	WDT control 1: enable	Selects WDT detection time 000: 2^{15}/f$_{SYS}$ 001: 2^{17}/f$_{SYS}$ 010: 2^{19}/f$_{SYS}$ 011: 2^{21}/f$_{SYS}$ 100: 2^{23}/f$_{SYS}$ 101: 2^{25}/f$_{SYS}$ 110: Setting prohibited 111: Setting prohibited			"0" is read.	IDLE 0: Stop 1: Start	0: Generates NMI interrupt 1: Connects WDTOUT to reset	Write "0"

Watchdog timer out control

0	Generates NMI interrupt
1	Connects WDT out to reset

Detection time of watchdog timer   @ fc = 48MHz

CGSYSCR1 clock gear value <GEAR>	Detection time of watchdog timer					
	WDMOD<WDTP>					
	000	001	010	011	100	101
000 （fc）	0.68 ms	2.73 ms	10.92 ms	43.69 ms	174.76 ms	0.70 s
100 (fc/2)	1.37 ms	5.46 ms	21.85 ms	87.38 ms	349.53 ms	1.40 s
101 (fc/4)	2.73 ms	10.92 ms	43.69 ms	174.76 ms	699.05 ms	2.80 s
110 (fc/8)	5.46 ms	21.85 ms	87.38 ms	349.53 ms	1.40 s	5.59 s
111 (fc/16)	10.92 ms	43.69 ms	174.76 ms	699.05 ms	2.80 s	11.18 s

Enable/disable control of the watchdog timer

0	Disable
1	Enable

Watchdog Timer Control Register

WDCR
(0x400F_2004)

	7	6	5	4	3	2	1	0
bit Symbol	WDCR							
Read/Write	W							
After reset	—							
Function	B1H : WDT disable code 4EH : WDT clear code							

Disable & clear of WDT

B1H	WDT disable code
4EH	WDT clear code
Others	—

3.17.4 Description of Operation

The watchdog timer generates the INTWDTn interrupt after a lapse of the detection time specified by the WDMOD <WDTP> register and outputs a signal at low level from the output pin of the watchdog timer (WDTOUTn). Before generating the INTWDTn interrupt, the binary counter for the watchdog timer must be cleared to "0" using software (instruction). If the CPU malfunctions (runaways) due to noise or other disturbances and cannot execute the instruction to clear the binary counter, the binary counter overflows and the non-maskable interrupt is generated by the INTWDTn interrupt. The CPU is able to recognize the occurrence of a malfunction (runaway) by identifying the non-maskable interrupt and to restore the faulty condition to normal by using a malfunction (runaway) countermeasure program. Additionally, it is possible to resolve the problem of a malfunction (runaway) of the CPU by connecting the watchdog timer out pin to reset pins of peripheral devices.

The watchdog timer begins operation immediately after a reset is cleared.

In STOP mode, the watchdog timer is reset and in an idle state. In IDLE mode, its operation depends on the WDMOD <I2WDT> setting. Before putting it in IDLE mode, WDMOD <I2WDT> must be set to an appropriate setting, as required.

Example:

1. To clear the binary counter

                    7 6 5 4 3 2 1 0
    WDCR ← 0 1 0 0 1 1 1 0   Writes the clear code (4EH)

2. To set the detection time of the watchdog timer to $2^{19}/f_{SYS}$.

                    7 6 5 4 3 2 1 0
    WDMOD ← 1 0 1 − − − − −

3. To disable the watchdog timer.

                    7 6 5 4 3 2 1 0
    WDMOD ← 0 − − − − − − −   Clears WDTE to "0"
    WDCR ← 1 0 1 1 0 0 0 1   Writes the disable code (B1H)

(Note 1) **If the watchdog timer is operated when the high-frequency oscillator is idle, the system reset operation initiated by the watchdog timer becomes erratic due to the unstable oscillation of the high-frequency oscillator. Therefore, do not operate the watchdog timer when the high-frequency oscillator is idle.**

(Note 2) **The counter of the watchdog timer stops at the debug mode.**

3.18   Key-On Wake-Up Operation

3.18.1 Overview

- The TMPM322F10 has four key inputs, KWUP0 to KWUP3. They can be used as interrupts for waking up from STOP mode or normal external interrupts. The four inputs share a single interrupt source (INTKWUP), which needs to be set in the clock generator (CG) block. KWUP0 to KWUP3 can be enabled or disabled individually in the <KEYEN> bit of the KWUP control register (KWUPCRn).

- The active state of each KWUP input can be selected in the KWUPCRn<KEY> bits from rising edge, falling edge, both edges, High level and Low level.

- All interrupt requests can be cleared by setting the KWUP interrupt all clear register (KWUPCLR) in an interrupt service routine.

- Each KWUP input pin has a pull-up, which can individually be configured as static or dynamic in the KWUPCRn<DPE> bit.

3.18.2   Block Diagram
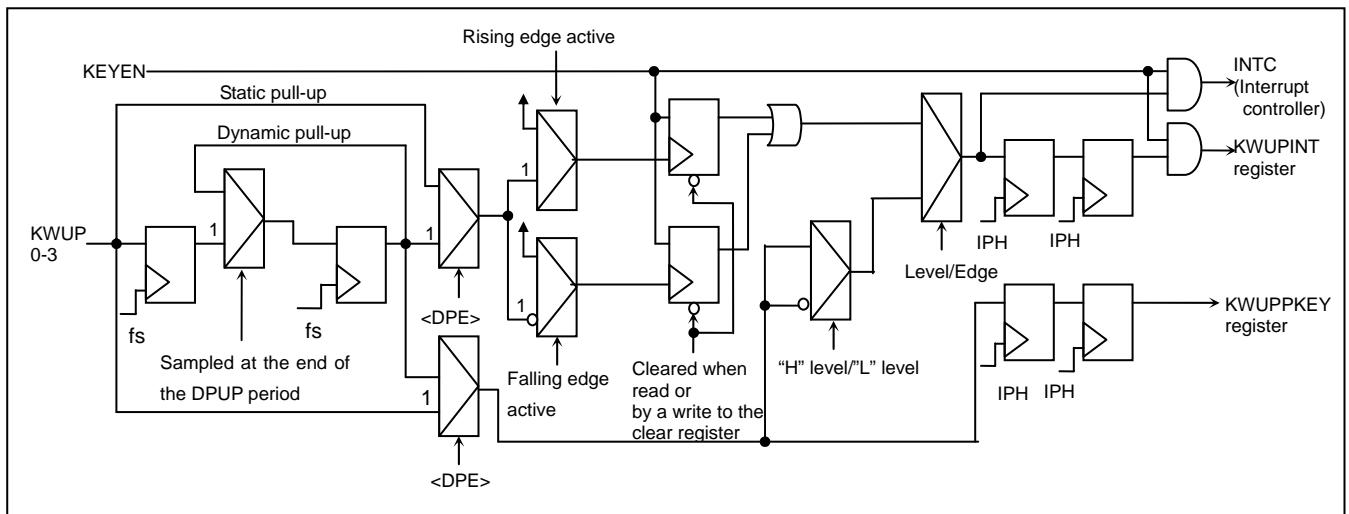
Figure 3.18.1   Block Diagram of the Key-On Wake-Up Circuit

3.18.3 Key-On Wake-Up Operation

The TMPM322F10 has four key input pins (KWUP0 to KWUP3). The <INTLEN> bit of the IMCGF register in the CG block is used to select whether to use KWUP0 to KWUP3 as interrupts for waking up from STOP mode or normal interrupts. Setting <INTLEN>="1" configures KWUP0 to KWUP3 as STOP wake-up interrupt pins. In this case, enable or disable each KWUP input in the KWUPCRn<KEYnEN> bit and also select the active state for each KWUP input in the KWUPCRn<KEY> bits.

When an input is detected on one of the KWUP pins in the key-on wake-up (KWUP) block, it is notified to the IMCGF register in the CG block with a high-level signal. It is therefore necessary to set CGIMCGF<EMCGL>="001" to detect an interrupt with H level.

Clearing CGIMCGF<INTLEN> to "0" (default) configures KWUP0 to KWUP3 as normal interrupt pins. In this case, an H pulse or H level signal needs to be input on these pins for the CPU to detect an interrupt.

The KWUPCRn register is also used to enable or disable each KWUP input and to set the active state for each input when they are used as normal interrupt pins. All key interrupt requests are cleared by writing "1010" in the KWUPCLR<KEYCLR> register in an interrupt service routine.

Note: If multiple KWUP inputs occur, clearing interrupt requests also clears all KWUP input requests.

3.18.4 Pull-Up Function

Each KWUP input pin has a pull-up, which can be configured as static or dynamic by setting the KWUPCRn<DPE> bit.

When static pull-up is selected, the pull-up is enabled irrespective of the KWUPCRn<KEYEN> setting.

3.18.4.1 Settings Required for Using KWUP Inputs with Pull-Up Enabled

(Example: Using Port J7 and double-edged interrupt)

A）When initially configuring a KWUP input after power-on
1) Set the port registers.
   Set PJFR2<PJ7F2>="1" to select the key input function.
   Set PJPUP<PJ7UP>="1" to enable the pull-up.
   Set PJIE<PJ7IE>="1" to enable input.
2) Set KWUPCR3<KEYEN>="0" to disable the KWUP input to be used.
3) Set KWUPCR3<KEY>="100" (both edges) to set the active state of the KWUP input to be used.
4) Set KWUPCR3<KEYEN>="1" to enable the KWUP input to be used.
5) Wait until pull-up is completed.
6) Set KWUPCLR<KEYCLR> ="1010" to clear all interrupt requests.
7) Set the clock generator registers.
   CGIMCGF<EMCGL>="001" (H level)
   CGIMCGF<INTLEN>="1" (enable the KWUP interrupt)
（For how to set these registers, see "3.5 Interrupts". ）

B）When changing the active state of a KWUP input during operation
1) Set the interrupt clear-enable register 2 <1>="1" to disable INTKWUP (interrupt number 33).
2) Set KWUPCR3<KEYEN>="0" to disable the KWUP input to be used.
3) Set KWUPCR3<KEY>="000" to change the active state of the KWUP input to be used. (It is set to L level here.)
4) Set KWUPCR3＜KEYEN＞ ="1" to enable the KWUP input to be used.
5) Set KWUPCLR<KEYCLR> ="1010" to clear all interrupt requests.
6) Set the interrupt priority register PRI_33 (corresponding to INTKWUP) to an appropriate level.
7) Set the interrupt set-enable register 2 <1>="1" to enable INTKWUP (interrupt number 33).

C）When enabling a KWUP input during operation
1) Set the interrupt clear-enable register 2<1>="1" to disable INTKWUP (interrupt number 33).
2) Set KWUPCR3<KEYEN>="0" to disable the KWUP input to be used.
3) Set KWUPCR3<KEY> to select the active state of the KWUP input to be used.
4) Set KWUPCR3<KEYEN>="1" to enable the KWUP input to be used.

5) Set KWUPCLR ="1010" to clear all interrupt requests.

6) Set the interrupt priority register PRI_33 (corresponding to INTKWUP) to an appropriate level.

7) Set the interrupt set-enable register 2 <1>="1" to enable INTKWUP (interrupt number 33).

3.18.4.2   Settings Required for Using KWUP Inputs with Pull-Up Disabled

(Example: Using Port J7 and double-edged interrupt)

A）When initially configuring a KWUP input after power-on
  1) Set PJPUP<PJ7UP> ="0" to disable the pull-up.
  2) Set KWUPCR3<KEYEN>＝"0" to disable the KWUP input to be used.
  3) Set KWUPCR3<KEY> ="000" to set the active state of the KWUP input to be used.
  4) Set KWUPCR3<KEYEN>="1" to enable the KWUP input to be used.
  5) Set KWUPCLR<KEYCLR>="1010" to clear all interrupt requests.
  6) Set KWUPCR3<KEYEN>="1" to enable the KWUP input to be used.
  7) Set the clock generator registers.
        CGIMCGF<EMCGL>="001"(H level)
        CGIMCGF<INTLEN>="1"(enable the KWUP interrupt)
      （For how to set these registers, see "3.5 Interrupts".）

B）When changing the active state of a KWUP input during operation
  1) Set the interrupt clear-enable register 2<1>="1" to disable INTKWUP (interrupt number 33).
  2) Set KWUPCR3<KEYEN>="0" to disable the KWUP input to be used.
  3) Set KWUPCR3<KEY> to change the active state of the KWUP input to be used.
  4) Set KWUPCR3<KEYEN>="1" to enable the KWUP input to be used.
  5) Set KWUPCLR<KEYCLR> ="1010" to clear all interrupt requests.
  6) Set the interrupt priority register PRI_33 (corresponding to INTKWUP) to an appropriate level.
  7) Set the interrupt set-enable register 2<1>="1" to enable INTKWUP (interrupt number 33).

C）When enabling a KWUP input during operation
  1) Set the interrupt clear-enable register 2<1> ="1" to disable INTKWUP (interrupt number 33).
  2) Set KWUPCR3<KEYEN>="0" to disable the KWUP input to be used.
  3) Set KWUPCR3 <KEY> to set the active state of the KWUP input to be used.
  4) Set KWUPCR3<KEYEN>="1" to enable the KWUP input to be used.
  5) Set KWUPCLR ="1010" to clear all interrupt requests.
  6) Set KWUPCR3<KEYEN>="1" to enable the KWUP input to be used.
  7) Set the interrupt priority register PRI_33 (corresponding to INTKWUP) to an appropriate level.
  8) Set the interrupt set-enable register 2<1>="1" to enable INTKWUP (interrupt number 33).

3.18.5   Registers

3.18.5.1   Register List

Base Address = 0x400F_1000

Register Name	Address (base +)	Description
KWUPCR0	0x400F_1000	KWUP Control Register 0
KWUPCR1	0x400F_1004	KWUP Control Register 1
KWUPCR2	0x400F_1008	KWUP Control Register 2
KWUPCR3	0x400F_100C	KWUP Control Register 3
KWUPPKEY	0x400F_1080	KWUP Port Monitor Register
KWUPCNT	0x400F_1084	KWUP Control Register
KWUPCLR	0x400F_1088	KWUP Interrupt All Clear Register
KWUPINT	0x400F_108C	KWUP Interrupt Monitor Register

3.18.5.2 KWUP Control Register n (n=0-3)

KWUPCRn (0x400F_100x)		7	6	5	4	3	2	1	0
	Bit Symbol	DPE	KEY						KEYEN
	Read/Write	R							
	After Reset	0	0	1	0	0	0	0	0
	Function	Pull-up 0: Static 1: Dynamic	KWUPn active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Read as "0".			KWUP0 interrupt input 0: Disable 1: Enable
		15	14	13	12	11	10	9	8
	Bit Symbol								
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Read as "0".							
		23	22	21	20	19	18	17	16
	Bit Symbol								
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Read as "0".							
		31	30	29	28	27	26	25	24
	Bit Symbol								
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Read as "0".							

<KEYEN>   Enables or disables the KWUP interrupt (INTKWUP) input.

<KEY>   Specifies the active state of the KWUPn input.

<DPE>   Selects static or dynamic pull-up.

3.18.5.3　Port Monitor Register

	7	6	5	4	3	2	1	0
Bit Symbol					PKEY3	PKEY2	PKEY1	PKEY0
Read/Write	R							
After Reset	0	0	0	0	0	0	0	0
Function	Read as "0".				Port state 0: "L" 1: "H"	Port state 0: "L" 1: "H"	Port state 0: "L" 1: "H"	Port state 0: "L" 1: "H"
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After Reset	0	0	0	0	0	0	0	0
Function	Read as "0".							
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After Reset	0	0	0	0	0	0	0	0
Function	Read as "0".							
	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After Reset	0	0	0	0	0	0	0	0
Function	Read as "0".							

KWUPPKEY
(0x400F_1080)

<PKEYn>　　The external port state can be monitored by reading KWUPPKEY<PKEYn> even when dynamic pull-up is enabled. Sampling is performed at dynamic pull-up intervals.

3.18.5.4   KWUP Pull-Up Period Register

KWUPCNT (0x400F_1084)		7	6	5	4	3	2	1	0
	Bit Symbol			T2S		T1S			
	Read/Write	R/W	R	R/W				R	
	After Reset	0	0	0	0	0	0	0	0
	Function	Must be written as "0".	Read as "0".	Dynamic pull-up interval 00: 256/fs 01: 512/fs 10: 1024/fs 11: 2048/fs		Dynamic pull-up period 00: 2/fs 01: 4/fs 10: 8/fs 11: 16/fs		Read as "0".	
		15	14	13	12	11	10	9	8
	Bit Symbol								
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Read as "0".							
		23	22	21	20	19	18	17	16
	Bit Symbol								
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Read as "0".							
		31	30	29	28	27	26	25	24
	Bit Symbol								
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Read as "0".							

The dynamic pull-up operation is performed as shown below.
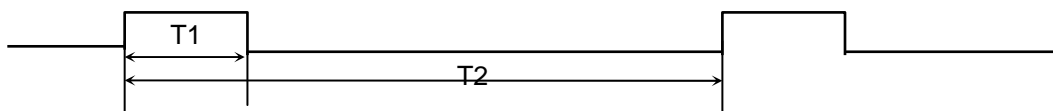
The pull-up is enabled only during the T1 period specified in the <T1S> bits. At other times, the pull-up is disabled.

   00: 2/fs (62.5 μs at fs = 32 kHz)
   01: 4/fs (125 μs at fs = 32 kHz)
   10: 8/fs (250 μs at fs = 32 kHz)
   11: 16/fs (500 μs at fs = 32 kHz)

The dynamic pull-up operation is repeated at intervals specified in the <T2S> bits.

   00: 256/fs (8 ms at fs = 32 kHz)
   01: 512/fs (16 ms at fs = 32 kHz)
   10: 1024/fs (32 ms at fs = 32 kHz)
   11: 2048/fs (64 ms at fs = 32 kHz)

Note 1: Use the fs clock when dynamic pull-up is selected.
Note 2: After switching to dynamic pull-up, wait for one T1 period before making any key input.

3.18.5.5 KWUP Interrupt All Clear Register

	7	6	5	4	3	2	1	0
Bit Symbol					KEYCLR			
Read/Write	R				W			
After Reset	0	0	0	0	0	0	0	0
Function	Read as "0".				Writing "1010" clears all interrupt requests. Each bit reads as "0".			

KWUPCLR (0x400F_1088)

	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After Reset	0	0	0	0	0	0	0	0
Function	Read as "0".							

	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After Reset	0	0	0	0	0	0	0	0
Function	Read as "0".							

	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After Reset	0	0	0	0	0	0	0	0
Function	Read as "0".							

3.18.5.6　KWUP Interrupt Monitor Register

KWUPINT
(0x400F_108C)

		7	6	5	4	3	2	1	0
	Bit Symbol					KEYINT3	KEYINT2	KEYINT1	KEYINT0
	Read/Write					R			
	After Reset	0	0	0	0	0	0	0	0
	Function	Read as "0".				Interrupt 0: No interrupt 1: Interrupt generated	Interrupt 0: No interrupt 1: Interrupt generated	Interrupt 0: No interrupt 1: Interrupt generated	Interrupt 0: No interrupt 1: Interrupt generated
		15	14	13	12	11	10	9	8
	Bit Symbol								
	Read/Write					R			
	After Reset	0	0	0	0	0	0	0	0
	Function	Read as "0".							
		23	22	21	20	19	18	17	16
	Bit Symbol								
	Read/Write					R			
	After Reset	0	0	0	0	0	0	0	0
	Function	Read as "0".							
		31	30	29	28	27	26	25	24
	Bit Symbol								
	Read/Write					R			
	After Reset	0	0	0	0	0	0	0	0
	Function	Read as "0".							

When KWUPCRn<KEYEN>=1, an active signal on KWUPn sets the corresponding KWUPINT<KEYINTn> bit to "1" to indicate that an interrupt is generated. The KWUPINT register is read-only. When a bit in this register is set to "1", it can be cleared by reading, which also clears the corresponding interrupt request. The KWUPCLR register is also provided to clear all the <KEYINT> bits in one operation.

When the active state of a KWUP input is set to the H or L level, the corresponding bit in the KWUPINT register, once set, will not be cleared by reading the bit unless the interrupt request is deasserted at its source.

3.18.6   KWUP Input and Detection Timing

1) When PJPUP<PJnUP> ="1" and KWUPSTn<DPE> ="0" (static pull-up)

   The active state of each KWUP input can be specified as H level, L level, rising edge, falling edge or both edges in KWUPSTn<KEY>. An active state can be detected at any time.

2) When PJPUP<PJnUP> ="1" and KWUPSTn<DPE> ="1" (dynamic pull-up)

   The key input signal is sampled at the end of the T1 period. This means that the active state of each key input (interrupt) can only be detected on a rising or falling edge one fs clock cycle before the end of the T1 period. Therefore, a key input must be maintained for a minimum of the T2 period and it is detected with a maximum delay of the T2 period. The following shows an example where the active state is falling edge.

Pull-up (T1 period)

Interval (T2 period)

T2 period or longer required (L level)

Key input   H or High-Z   H or High-Z or L

Interrupt detection timing

Key input detection

Internal sampling result
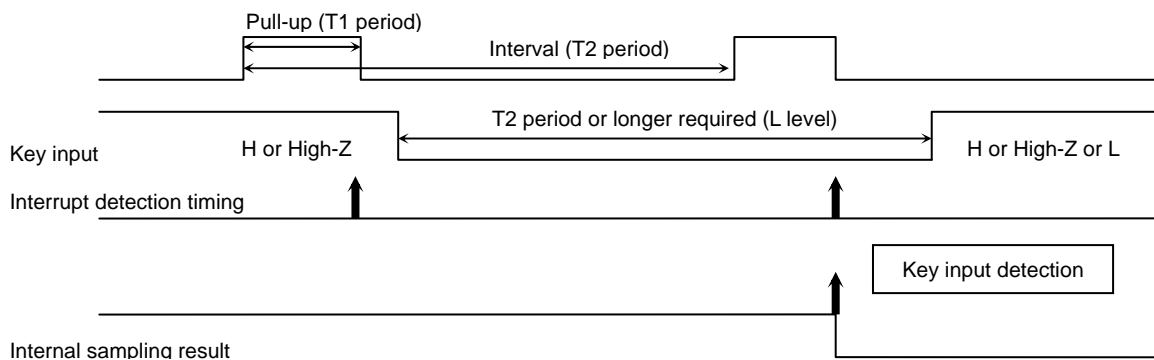
3.19 **Backup module**

3.19.1 Features

 The backup mode, one of the system operation modes, enables the MCU to operate in the low power consumption. By cutting electricity to the entire block, such as CPU or other peripheral IPs, other than the backup module, this mode significantly reduces power consumption.

The backup mode contains two modes:

   -Backup sleep mode (enabling low frequency oscillator)

   -Backup stop mode (disabling low frequency oscillator)

3.19.2 Block Diagram

   Fig. 3.19.1 shows power shutdown blocks during the backup mode operation.
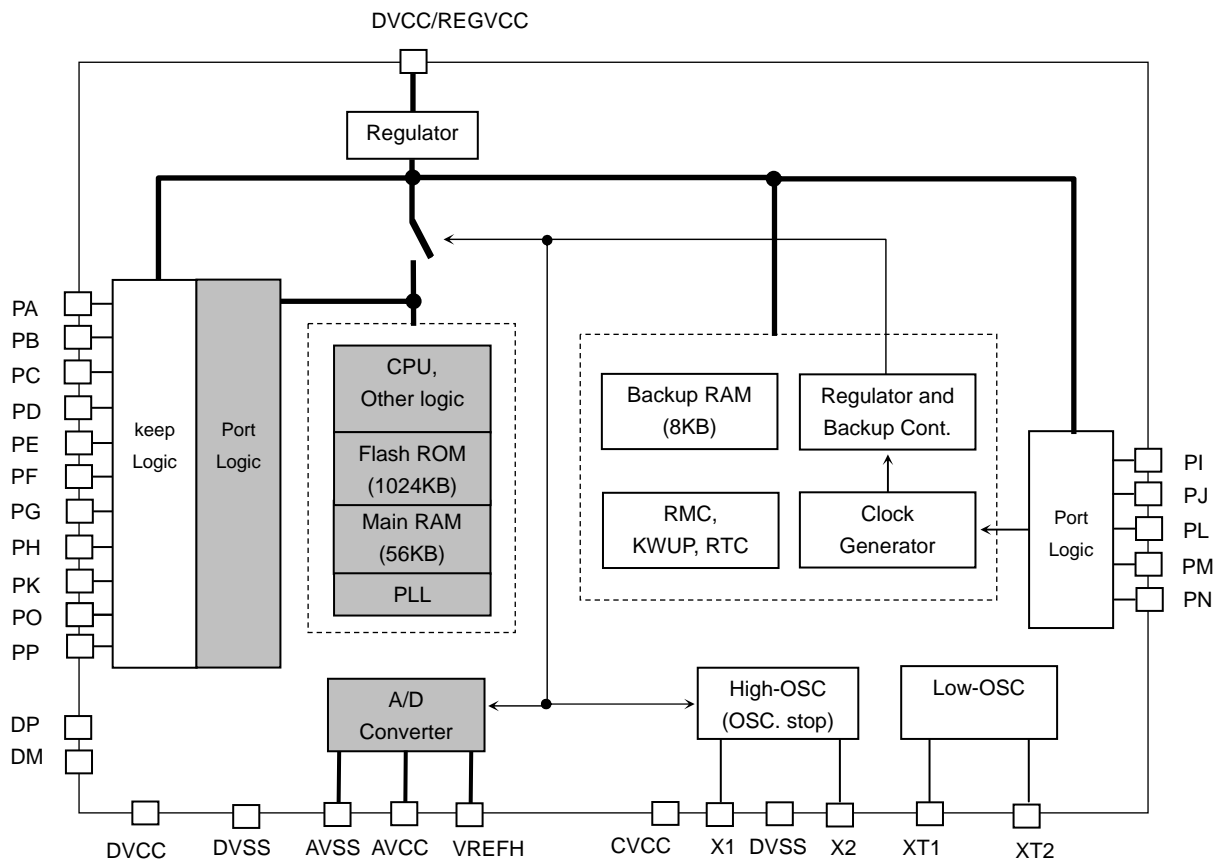
Fig. 3.19.1 Power shutdown blocks in the backup mode

    : Power shutdown blocks in the backup mode

3.19.3 Backup Mode Operation

The backup mode only corresponds to the single chip mode (MCU starts from the built-in flash memory after reset).

3.19.3.1 Operable peripherals in the backup mode

- In the backup sleep mode
  Port output, Key-on-wakeup, remote controller circuit, real-time clock, low speed oscillator, data in back up RAM (8KB).

- In the backup stop mode
  Port output, Key-on-wakeup, data in backup RAM (8KB).

3.19.3.2 Transition to the backup mode

Fig. 3.19.2 shows the state transition between normal mode and backup mode (backup sleep and backup stop). The backup mode (backup sleep and backup stop) will return the preceding mode of transition by the interrupt.
In addition, each mode changes to the reset processing routine if the reset operation occurs.
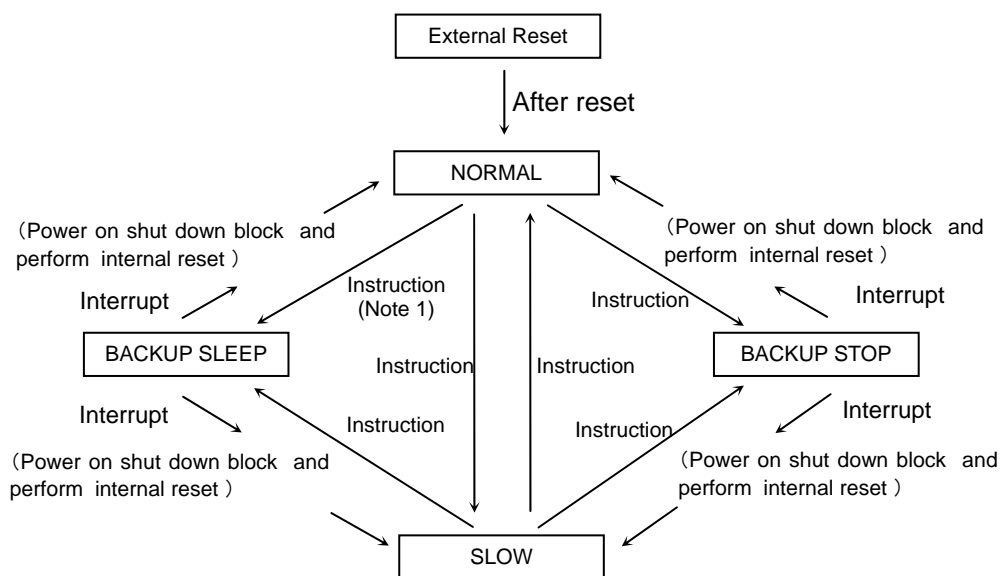
Fig. 3.19.2 Backup Mode Transition Diagram

(Note1)   In case that low-speed oscillator is stopped in the normal mode, make sure to start the low-speed oscillator and confirm the stable oscillation before changing to backup sleep mode.

(Note2) The program for changing to the backup mode must be executed in the built-in flash ROM or built-in RAM.

3.19.3.3 Backup Transition Flow

3.19.3.3.1 Preparing for backup mode

The preparation program for changing to the backup mode must be executed in the built-in flash ROM and built-in RAM.

(1) Stopping peripherals and saving data

In the normal mode stop peripheral functions including DMAC, SMC and WDT. In case of transition to backup sleep mode, no need to stop peripheral functions (RMC, RTC, and KWUP) which are used in backup sleep mode. It is necessary to save data to be preserved in backup RAM. Backup RAM is used only 8KB data from 0x2000_E000 to 0x2000_FFFF.

(2) Prohibit the interrupt

To prevent from obstructing a transition to backup mode, interrupt request set to disable if needed. It is note that NMIn interrupt and INTRTC interrupt request cannot be disabled so that these interrupt requests must be avoided in advance.

(3) Setting of port keep function （CGSTBYCR<PTKEEP>）

Port keep function retains the port status of the momentary when CGSTBYCR<PTKEEP> is set to "1." Object ports are A to H, K, O, P, SWDIO and NMI. Port keep function is capable of retaining input enable/disable, port 0/1 output status and on/off status of pull-up/pull-down resistor.

By these settings made before the transition to the backup mode, port keep function can hold the port status. When using the port keep function, port register of each port must be set properly.

The input/output status of port I, J, L, M and N are depend on the port register and setting of CGSTBYCR<DRVE> regardless the port keep function. The interrupt of backup mode is set by using these ports. All unnecessary ports must be set to disable by the input enable control register.

Table 3.19.1 Pin Status in BACKUP SLEEP mode and BACKUP STOP mode (1/2)

Keep the port status by setting the CGSTBYCR<PTKEEP> to "1" before sifting the backup mode.

Pin Name	Input/Output	BACKUP SLEEP <PTKEEP>=1	BACKUP STOP <DRVE>=0 <PTKEEP>=1	BACKUP STOP <DRVE>=1 <PTKEEP>=1
PA0 to PA7	Input mode	◎	◎	
	Output mode	◎	◎	
	D0 to D7	−(Note1)	−(Note1)	
PB0 to PB7	Input mode	◎	◎	
	Output mode	◎	◎	
	D8 to D15	−(Note1)	−(Note1)	
PC0 to PC7	Input mode, RXD8 to 9, SCLK8 to 9, CTS8n to 9n	◎	◎	
	Output mode, TXD8 to 9, SCLK8 to 9	◎	◎	
	A1 to A8	◎	◎	
PD0 to PD7	Input mode, RXD10 to 11, SCLK10 to 11, CTS10n to 11n	◎	◎	
	INTB	◎	◎	
	Output mode, TXD10 to 11, SCLK10 to 11	◎	◎	
	A9 to A16	◎	◎	
PE0 to PE7	Input mode, TB5IN0 to 1, TB6IN0 to 1, RXD0, SCLK0, CTS0n	◎	◎	
	INT5	◎	◎	
	Output mode, TXD0, SCLK0, SCOUT	◎	◎	
	A17 to A23	◎	◎	
PF0 to PF4	Input mode	◎	◎	
	Output mode	◎	◎	
	TRACECLK, TRACEDATA0 to 3	◎	◎	
PG0 to PG7	Input mode, SDA1 to 2, SCL1 to 2, SCK1 to 2, TB7IN0 to 1, TB9IN0 to 1, USBOCn	◎	◎	
	INT6 to 7	◎	◎	
	Output mode, SDA1 to 2, SCL1 to 2, SCK1 to 2, USBPON, WDTOUTn	◎	◎	
	CS0n, CS1n, CS3n	◎	◎	
PH0 to PH7	Input mode, SDA3 to 4, SCL3 to 4, SCK3 to 4, TBAIN0 to 1, TBBIN0 to 1, TBDIN0 to 1, TBEIN0 to 1	◎	◎	
	INTC to D	◎	◎	
	Output mode, SDA3 to 4, SCL3 to 4, SCK3 to 4	◎	◎	
PI0 to PI1	Input mode	○	−	○
	Output mode	○	−	○
PJ0 to PJ7	Input mode, ADTRGn	○	−	○
	AN0 to 7	−	−	−
	KWUP0 to 3	○	○	○
PK0 to PK7	Input mode	◎	◎	
	AN8 to 15	◎	◎	
PL0 to PL7	Input mode, SDA0, SCL0, SCK0, RXD1, SCLK1, CTS1n	○	−	○
	INT0 to 1	○	○	○
	Output mode, SDA0, SCL0, SCK0, TXD1, SCLK1, TB0 to 7OUT	○	−	○
PM0 to PM7	Input mode, RXD2 to 3, SCLK2 to 3, CTS2n to 3n, TB1IN0 to 1	○	−	○
	INT2 to 3	○	○	○
	Output mode, TXD2 to 3, SCLK2 to 3, ALARMn, TB3OUT	○	−	○
PN0 to PN7	Input mode, RXD4 to 5, SCLK4 to 5, CTS4n to 5n, TB2IN0 to 1, TBFIN0 to 1	○	−	○
	RMIN0 to 1	○	−	○
	INT4, INT8	○	○	○
	Output mode, TXD4 to 5, SCLK4 to 5	○	−	○

Table 3.19.1 Pin Status in BACKUP SLEEP mode and BACKUP STOP mode (2/2)

Pin Name	Input/Output	BACKUP SLEEP \<PTKEEP\>=1	BACKUP STOP \<DRVE\>=0 \<PTKEEP\>=1	BACKUP STOP \<DRVE\>=1 \<PTKEEP\>=1
P00 to P07	Input mode, RXD6 to 7, SCLK6 to 7	◎	◎	
	INT9 to A	◎	◎	
	Output mode, TXD6 to 7, SCLK6 to 7, TB8 to FOUT	◎	◎	
PP0 to PP6	Input mode, SPDI, SPCLK	◎	◎	
	Output mode, SPDO, SPCLK, SPFSS	◎	◎	
	CS2n, BLS0n to 1n, WEn, OEn	◎	◎	
SWDIO		Input (PU) or Output	Input (PU) or Output	
SWCLK		Input (PD)	Input (PD)	
D+, D−		Suspend (Note2)	Suspend (Note2)	
NMIn		Input (PU)	Input (PU)	
TEST1 to 2		−	−	−
RESETn		Input	Input	Input
MODE		Input	Input	Input
XT1		Input	−	−
XT2		Output	H level	H level
X1		−	−	−
X2		H level	H level	H level

○   : At the time of setting of \<DRVE\>=1, Input mode/function input follows the setting of the port input control register (PxIE\<n\>). And ,Output mode/function output follows the setting of the port control register (PxCR\<n\>).

◎   : At the time of setting of \<DRVE\>=1, Input mode/function input follows the setting of the port input control register (PxIE\<n\>). And ,Output mode/function output follows the setting of the port control register (PxCR\<n\>).

-          : Input or output disabled.
Input   : The input gate is active. To prevent the input pin from floating, fix the input voltage to the level
Output   : The pin is in the output state.
Suspend   : Suspend status (Input ON/ Output OFF)
H level   : The pin is "H" level output.

(Note1)	: Complete the access to the external memory before change to the BACKUP mode.
(Note2)	: Suspend the USB port before change to the BACKUP mode.

(4)Clock related setting and warm up time

Stop PLL circuit by setting CGOSCCR\<PLLON\>="0." Set high-speed clock to fc(1/1) by CGSYSCR\<GEAR\>="000."Using backup sleep is needed for starting low-speed oscillator by CGOSCCR\<XTEN\>.

In addition, it is necessary to setting warm up time returning from backup mode by CGOSCCR\<WUPT\>\<WUPTL\>. The Warm up time is referred to the section"3.4.6.8 warm up."

3.19.3.3.2 Transition to backup mode

(1)Setting modes and clearing event of backup mode
   By the CGSTBYCR<STBY> register, set to the backup stop mode or backup sleep mode.

(2)Transition to the backup mode
   Clear the interrupt then execute WFI instruction.

3.19.3.3.3 Precautions for the use of the backup mode

(1) About ICE
   The communication with ICE is disconnected, if MCU changes to the backup mode. In this case, it is necessary to reconnect to ICE.

3.19.3.3.4 Returning from backup mode (Clearing)

(1)Interrupt event of backup
   Table 3.19.2 shows the interrupt events of backup stop mode/backup sleep mode.

Table 3.19.2 Interrupt events of backup mode

Backup modes	Interrupts
Backup stop	External REST input, INT0-4,8, INTKWUP(Static)
Backup sleep	External RESET input, INT0-4,8, INTKWUP(Dynamic/Static), INTRTC, INTRMCRX0, INTRMCRX1

(2)Clearing operation by external REST input
   Backup clearing operation by external reset is referred to "3.1.5.1 Cold reset operation" and "3.1.5.2 Warm reset operation."
   Note that it is not guaranteed the contents of back up RAM and clock/calendar related register of RTC if backup mode is cleared by external reset.

(3)Clearing operation by interrupts of backup mode
   If the events of interrupts are received, regulator starts to supply power to the shut down blocks and high-speed oscillator will start operation.
   The warm up timer will starts when the oscillation becomes stable. During warm up time, internal reset signal of power shut down block which returned from backup mode is continuing low level. Internal reset is cleared after the warm up time has elapsed, and then MCU returns to the preceding mode of backup mode.
   To release the low power consumption mode by using the level mode interrupt, keep the level until the warming-up time end. When the release factor canceled before the warming-up time ends, it returns to the backup mode again.

(4)Precaution after backup mode cleared
   ・   By reading CGRSTFLG register, it can be found which interrupts are occurred.

   ・   Make sure to perform the port A to H, K, O and P port settings before clearing port keep function by CGSTBYCR<PTKEEP>="0."
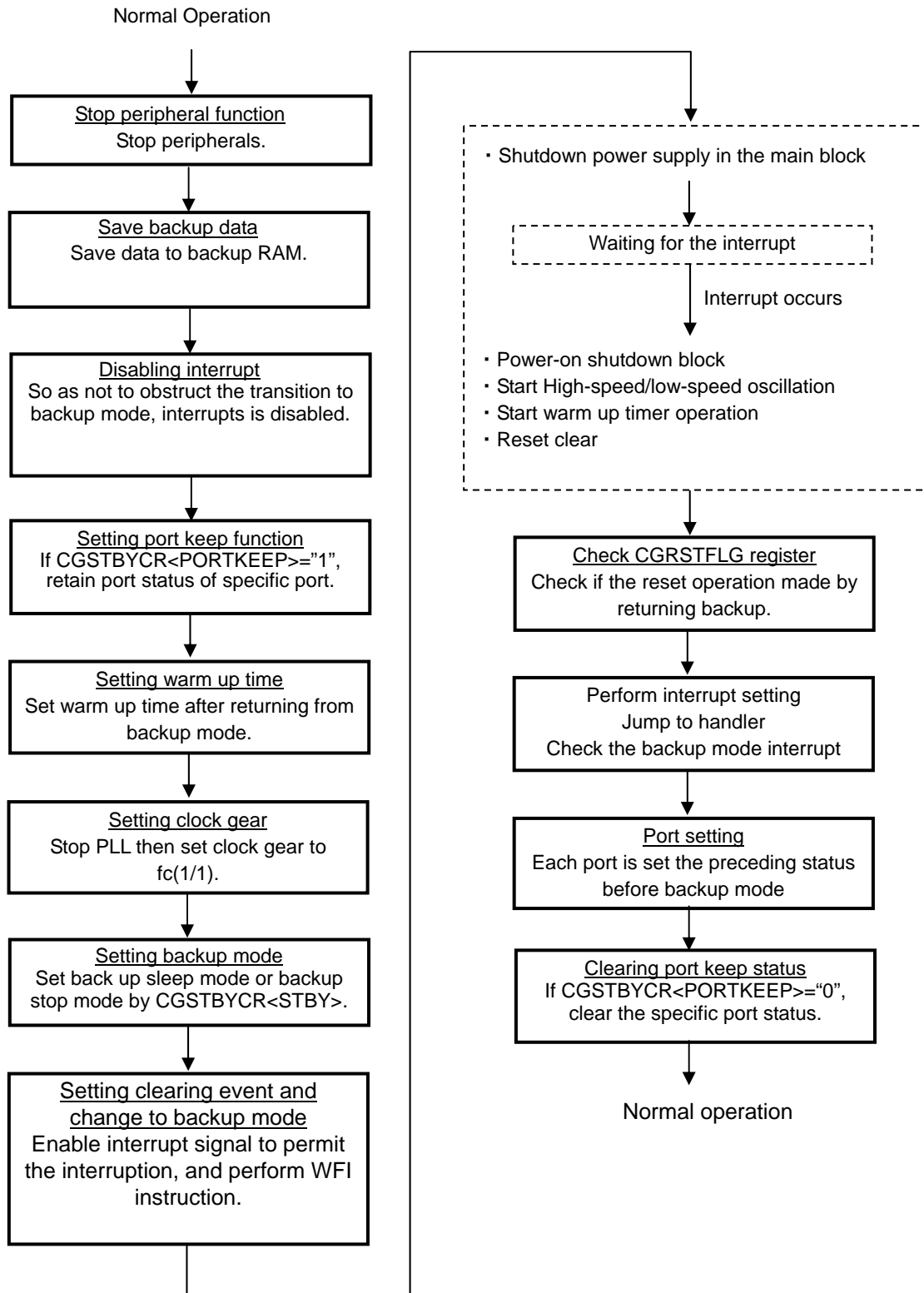
3.19.3.4 Transition Flowchart

Normal Operation

Stop peripheral function
Stop peripherals.

Save backup data
Save data to backup RAM.

Disabling interrupt
So as not to obstruct the transition to backup mode, interrupts is disabled.

Setting port keep function
If CGSTBYCR<PORTKEEP>="1", retain port status of specific port.

Setting warm up time
Set warm up time after returning from backup mode.

Setting clock gear
Stop PLL then set clock gear to fc(1/1).

Setting backup mode
Set back up sleep mode or backup stop mode by CGSTBYCR<STBY>.

Setting clearing event and change to backup mode
Enable interrupt signal to permit the interruption, and perform WFI instruction.

・Shutdown power supply in the main block

Waiting for the interrupt

Interrupt occurs

・Power-on shutdown block
・Start High-speed/low-speed oscillation
・Start warm up timer operation
・Reset clear

Check CGRSTFLG register
Check if the reset operation made by returning backup.

Perform interrupt setting
Jump to handler
Check the backup mode interrupt

Port setting
Each port is set the preceding status before backup mode

Clearing port keep status
If CGSTBYCR<PORTKEEP>="0", clear the specific port status.

Normal operation

Fig. 3.19.3 State transition flowchart

3.19.3.5 Backup mode Timing Chart
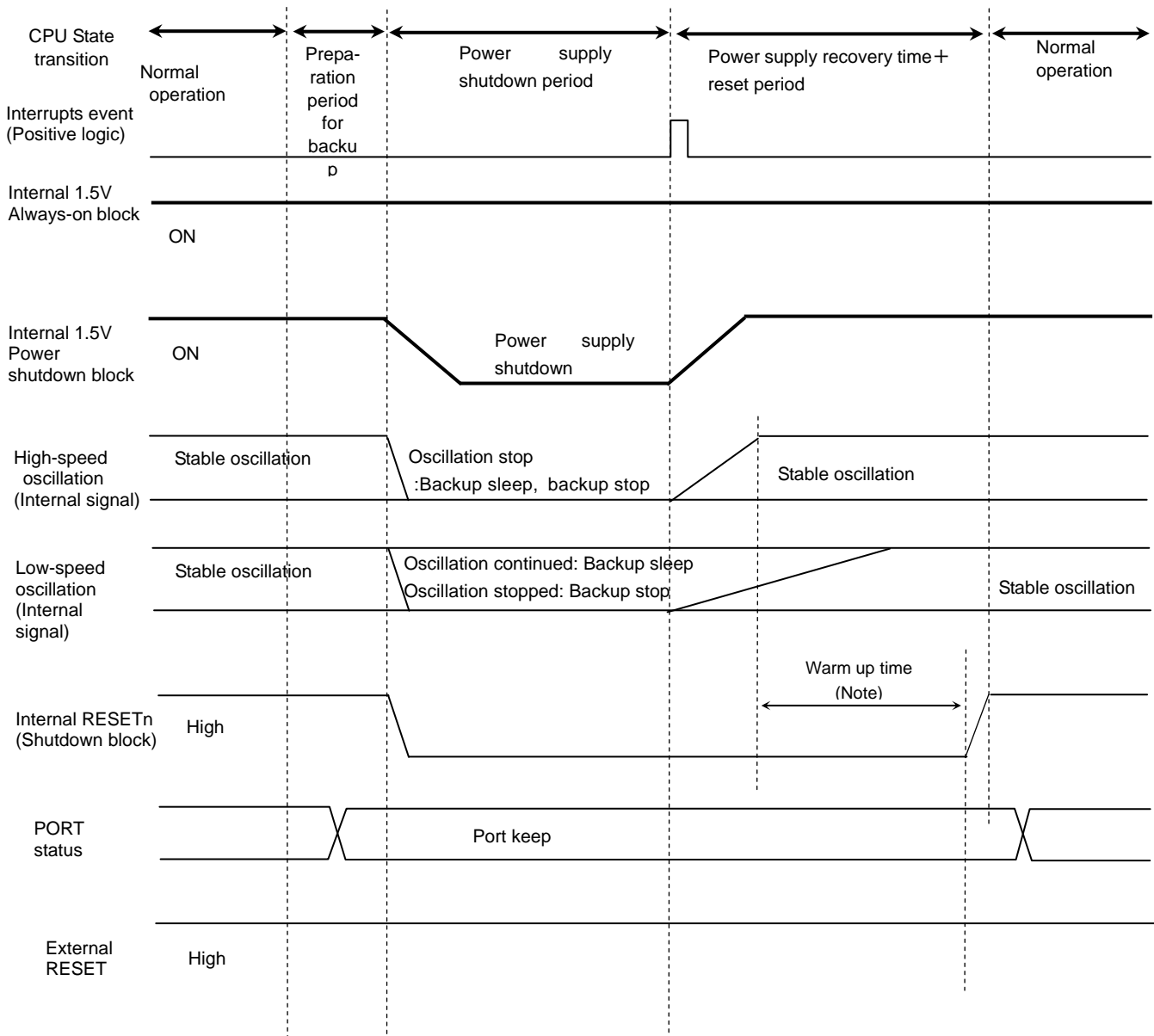
Fig. 3.19.4 Backup mode sequence

(Note)   The warm up time must be set in consideration of the stability time for the oscillator. But, the stability time for regulator (4096 cycles) and Flash (400us) need to warm up.

Example: Setting the minimum warm-up time when using a high-speed 12-MHz oscillator
(in case of the stability time for the oscillator $\leqq$ the stability time for internal cicuit)

the stability time for Flash $=$400us/(1/12MHz) $=$4800 cycles

Total $=$4096$+$4800 $=$8896 cycles

$=$22C0H (HEX)

Drop the lower 4 bits, and set 22CH to CGOSCCR<WUPT>.

Bit	31 30 29 28 27 26 25 24 23 22 21 20
22C H	0 0 1 0 0 0 0 1 0 1 1 0 0 0 b

3.20 Flash Memory Operation

This section describes the hardware configuration and operation of the flash memory.

3.20.1 Flash Memory

3.20.1.1 Features

1) Memory Capacity

The TMPM322F10 devices contain flash memory. The memory sizes and configurations of each device are shown in the table below. Independent write access to each block is available. When the CPU is to access the internal flash memory, 32-bit data bus width is used.

2) Write/Erase time

Writing is executed per page. The TMPM322F10FG contain 128 words in a page.
Page writing requires 1.25ms (typical) regardless of number of words.
A block erase requires 0.1 sec. (typical).
The following table shows write and erase time per chip.

Product Name	Memory Size	Block Configuration			# of Words	Write Time	Erase Time
		128KB	64KB	32KB			
TMPM322F10FG	1024KB	7	1	2	128	2.56sec	1.0sec

(Note) The above values are theoretical values not including data transfer time.
   The write time per chip depends on the write method to be used by the user.

3) Programing method

The onboard programming mode is available for the user to program (rewrite) the device while it is mounted on the user's board.

・The onboard programming mode

3-1) User boot mode
 The user's original rewriting method can be supported.

Rewriting method
The flash memory included in this device is generally compliant with the applicable JEDEC standards except for some specific functions. Therefore, if the user is currently using an external flash memory device, it is easy to implement the functions into this device. Furthermore, the user is not required to build his/her own programs to realize complicated write and erase functions because such functions are automatically performed using the circuits already built-in the flash memory chip.
This device is also implemented with a read-protect function to inhibit reading flash memory data from any external writer device. On the other hand, rewrite protection is available only through command-based software programming; any hardware setting method to apply +12VDC is not supported. See chapter 3.21 for details of ROM protection and security function.

JEDEC compliant functions	Modified, added, or deleted functions
• Automatic programming • Automatic chip erase • Automatic block erase • Data polling/toggle bit	<Modified> Block protect (only software protection is supported) <Deleted> Erase resume - suspend function

3.20.1.2 Block Diagram of the Flash Memory Section
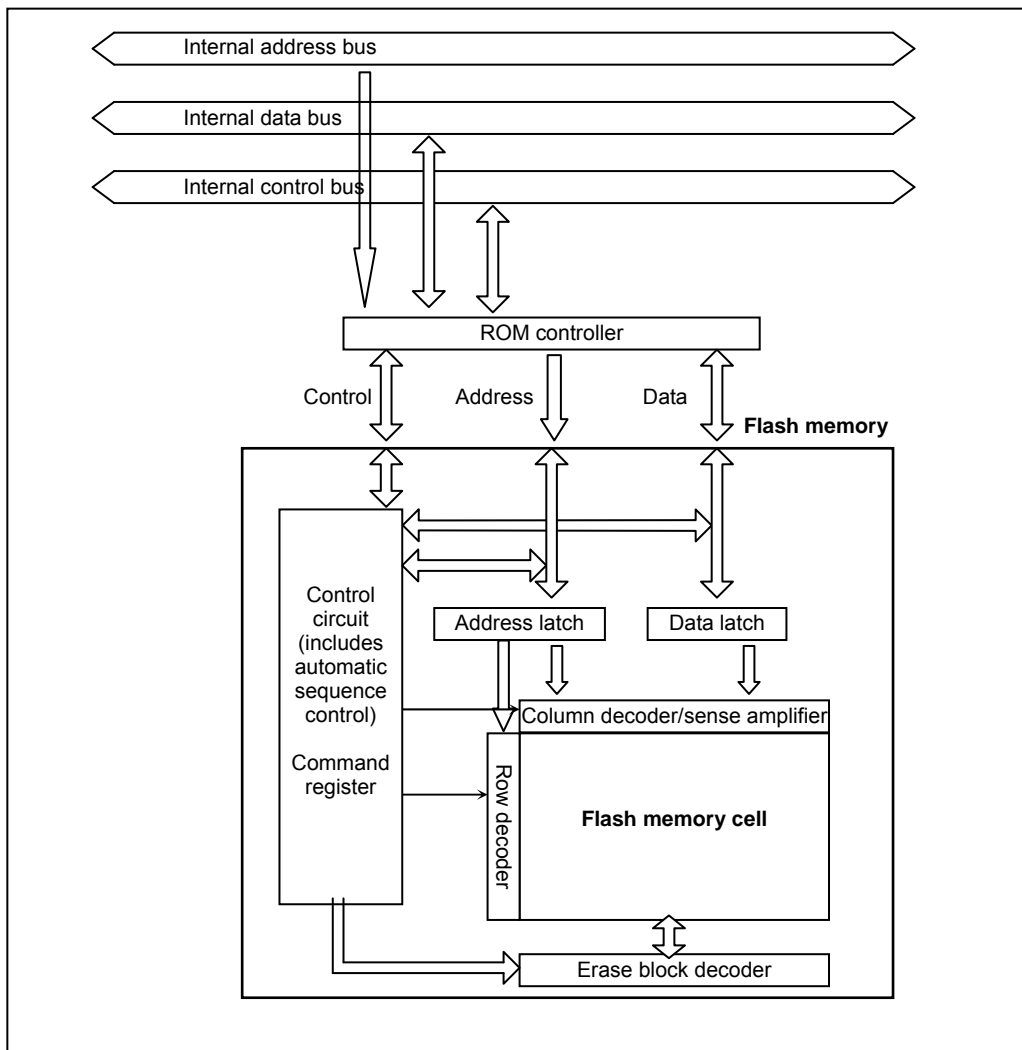
Fig. 3.20.1 Block Diagram of the Flash Memory Section

3.20.2 Operation Mode

This device has twe operation modes including the mode not to use the internal flash memory.

Table 3.20.1 Operation Mode

Operation mode		Operation details
Single chip mode		After reset is cleared, it starts up from the internal flash memory.
	Normal mode	In this operation mode, two different modes, i.e., the mode to execute user application programs and the mode to rewrite the flash memory onboard the user's card, are defined. The former is referred to as "normal mode" and the latter "user boot mode.
	User boot mode	The user can uniquely configure the system to switch between these two modes. For example, the user can freely design the system such that the normal mode is selected when the port "A0" is set to "1" and the user boot mode is selected when it is set to "0." The user should prepare a routine as part of the application program to make the decision on the selection of the modes.

Among the flash memory operation modes listed in the above table, the User Boot mode is the programmable modes. This mode, the User Boot mode , is referred to as "Onboard Programming" modes where onboard rewriting of internal flash memory can be made on the user's card.

3.20.2.1 Reset Operation

To reset the device, ensure that the power supply voltage is within the operating voltage range, that the internal oscillator has been stabilized, and that the RESETn input is held at "0" for a minimum duration of 12 system clocks (0.25 μs with 48MHz operation; the "1/1" clock gear mode is applied after reset).

(Note 1) Regarding power-on reset of devices with internal flash memory;
   for devices with internal flash memory, it is necessary to apply "0" to the RESETn inputs upon power on for a minimum duration of 200 microseconds regardless of the operating frequency.

(Note 2) While flash auto programming or deletion is in progress, at least 0.5 microseconds of reset period is required regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.

3.20.2.2 User Boot Mode (Single chip mode)

User Boot mode is to use flash memory programming routine defined by users. It is used when the data transfer buses for flash memory program code on the old application and for serial I/O are different. It operates at the single chip mode; therefore, a switch from normal mode in which user application is activated at the single chip mode to User Boot Mode for programming flash is required. Specifically, add a mode judgment routine to a reset program in the old application.

The condition to switch the modes needs to be set by using the I/O of M322 in conformity with the user's system setup condition. Also, flash memory programming routine that the user uniquely makes up needs to be set in the new application. This routine is used for programming after being switched to User Boot Mode. The execution of the programming routine must take place while it is stored in the area other than the flash memory since the data in the internal flash memory cannot be read out during delete/ writing mode. Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations. All the interruption including a non-maskable are inhibited at User Boot Mode.

(1-A) and (1-B) are the examples of programming with routines in the internal flash memory and in the external memory. For a detailed description of the erase and program sequence, refer to 3.20.3 On-board Programming of Flash Memory (Rewrite/Erase).
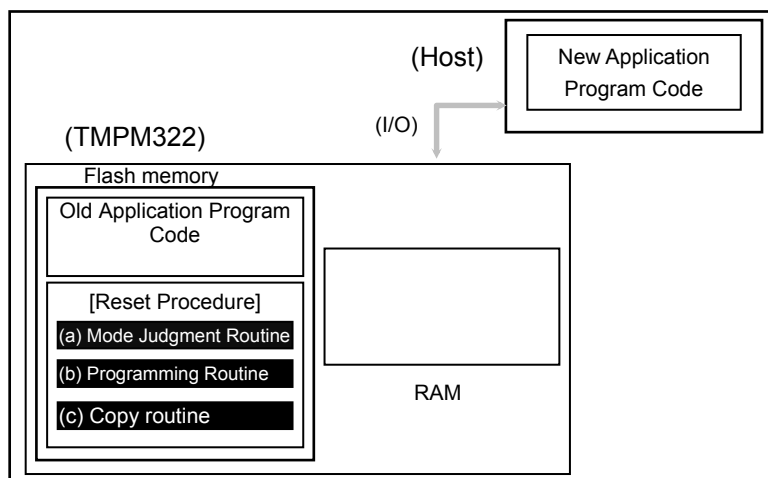
User Boot Mode

(1-A)   Method 1: Storing a Programming Routine in the Flash Memory

(Step-1)

Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM322 on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine: Code to determine whether or not to switch to User Boot mode

(b) Programming routine: Code to download new program code from a host controller and re-program the flash memory

(c) Copy routine: Code to copy the data described in (b) from the TMPM322 flash memory to either the TMPM322 on-chip RAM or external memory device.

(Step-2)

After RESETn is released, the reset procedure determines whether to put the TMPM322 flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be disabled while in User Boot mode.)
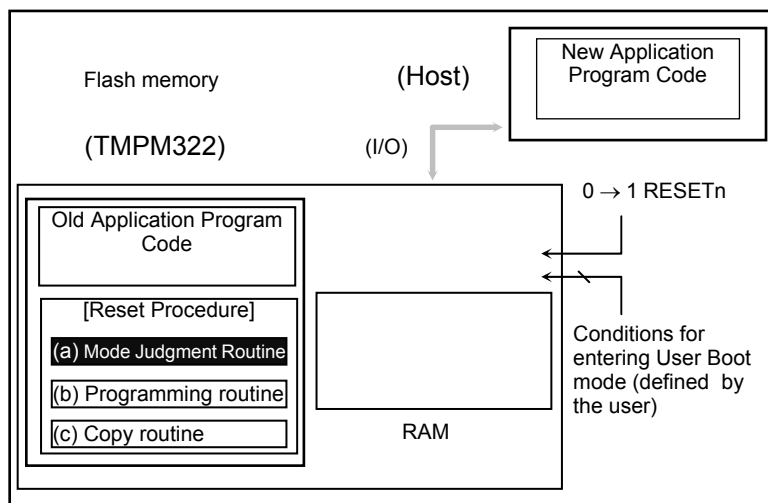
(Step-3)

Once transition to User Boot mode is occurred, execute the copy routine (c) to copy the flash programming routine (b) to the TMPM322 on-chip RAM.
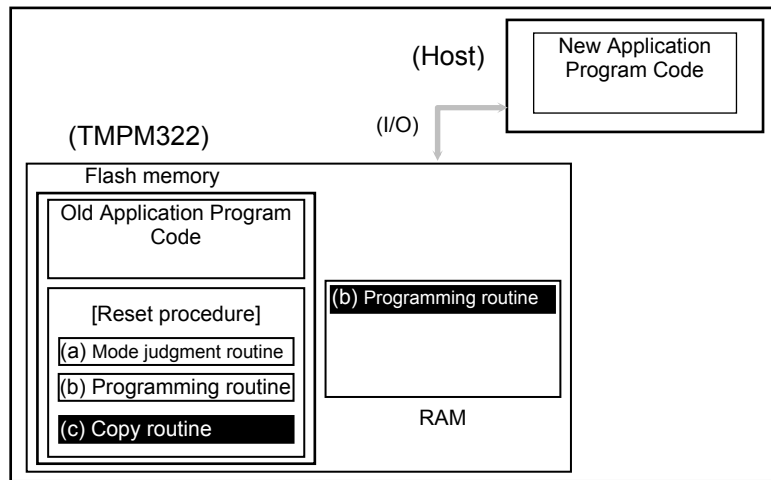
(Step-4)

Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.
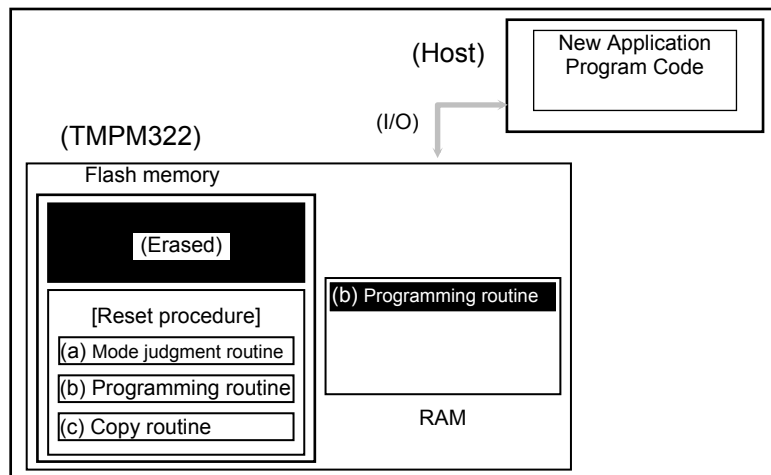
(Step-5)

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. Once programming is complete, turn on the protection of that flash block.
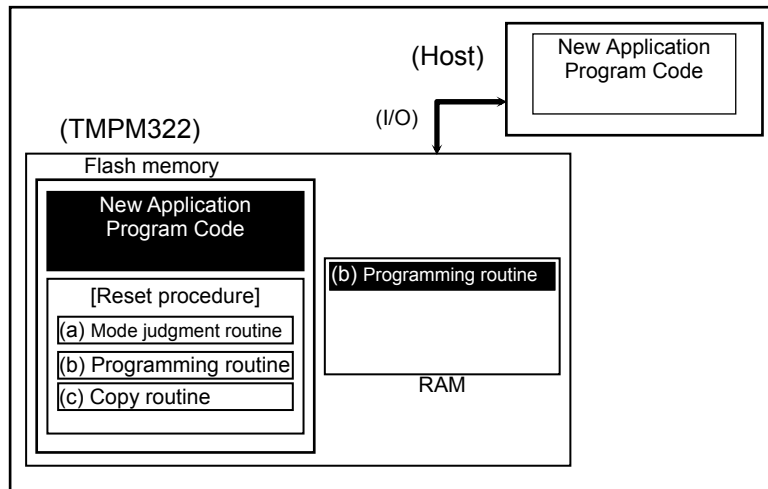
(Step-6)

Set RESETn to "0" to reset the TMPM322. Upon reset, the on-chip flash memory is put in Normal mode. After RESETn is released, the CPU will start executing the new application program code.
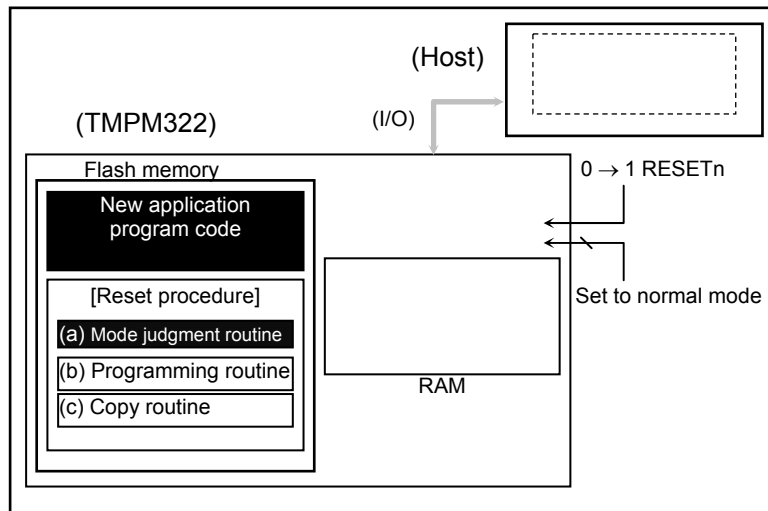
(1-B) Method 2: Transferring a Programming Routine from an External Host

(Step-1)

Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM322 on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine: Code to determine whether or not to switch to User Boot mode

(b) Transfer routine: Code to download new program code from a host controller

Also, prepare a programming routine shown below on the host controller:

(c) Programming routine: Code to download new program code from an external host controller and re-program the flash memory
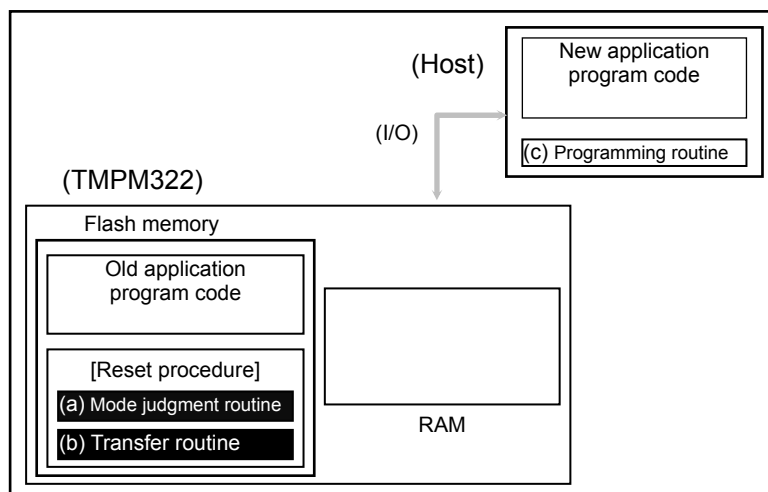

(Step-2)

After RESETn is released, the reset procedure determines whether to put the TMPM322 flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be disabled while in User Boot mode).
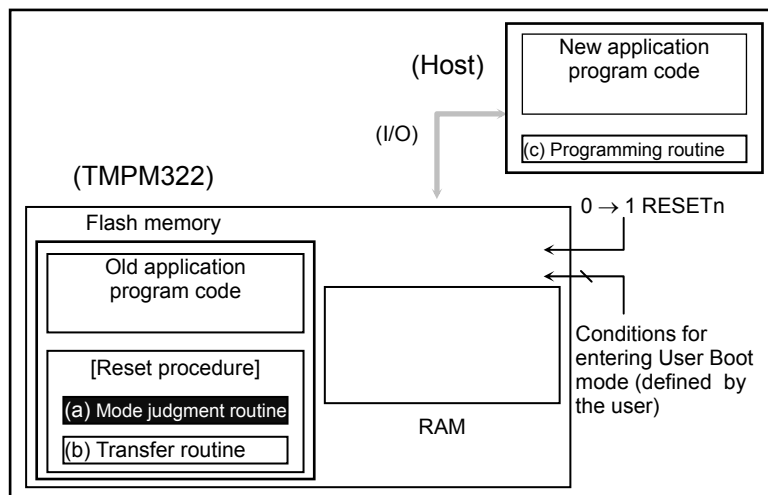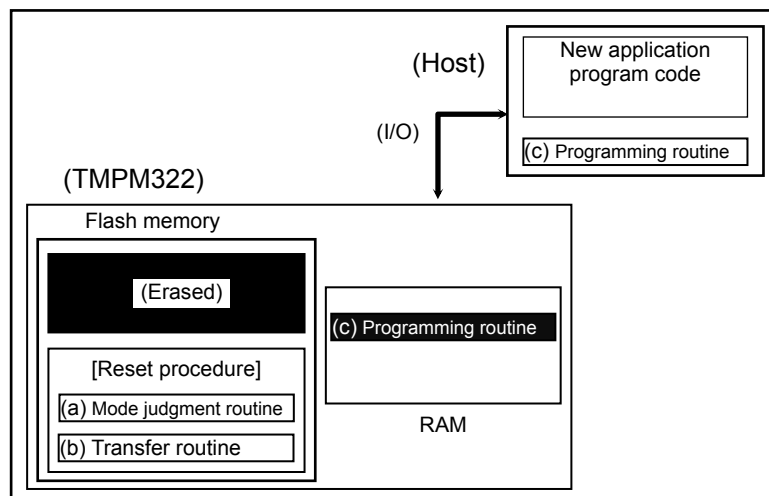
(Step-3)

Once User Boot mode is entered, execute the transfer routine (b) to download the flash programming routine (c) from the host controller to the TMPM322 on-chip RAM.
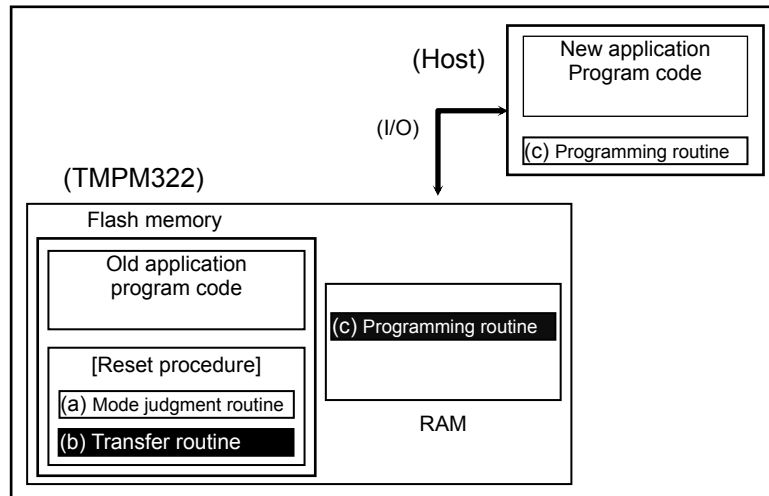

(Step-4)

Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.

(Step-5)

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. Once programming is complete, turn on the protection of that flash block.
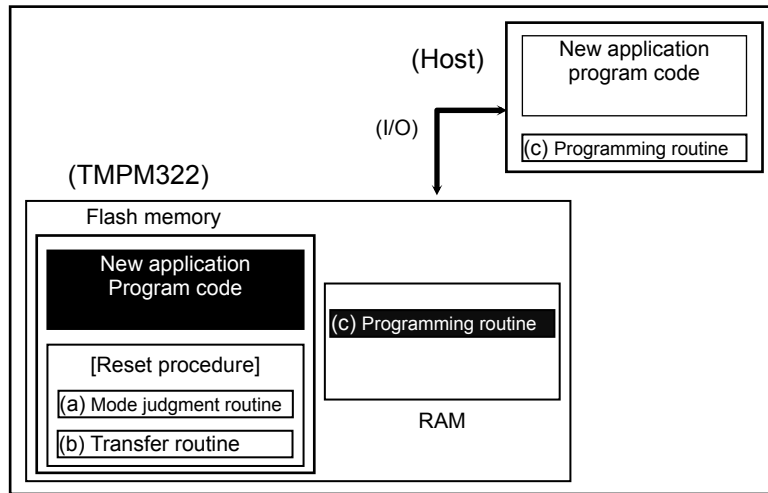
(Step-6)

Set RESETn to "0" low to reset the TMPM322. Upon reset, the on-chip flash memory is put in Normal mode. After RESETn is released, the CPU will start executing the new application program code.
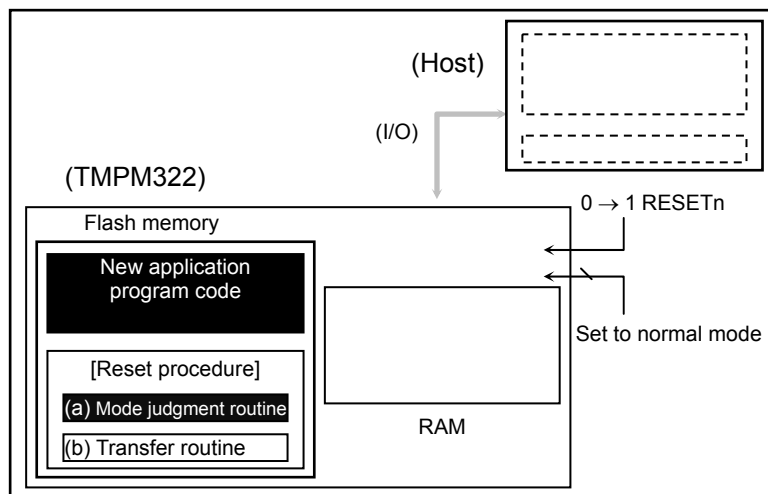
3.20.3 On-board Programming of Flash Memory (Rewrite/Erase)

In on-board programming, the CPU is to execute software commands for rewriting or erasing the flash memory. The rewrite/erase control program should be prepared by the user beforehand. Because the flash memory content cannot be read while it is being written or erased, it is necessary to run the rewrite/erase program from the internal RAM or from an external memory device after shifting to the user boot mode.

3.20.3.1 Flash Memory

Except for some functions, writing and erasing flash memory data are in accordance with the standard JEDEC commands. In writing or erasing, use 32-bit data transfer command of the CPU to enter commands to the flash memory. Once the command is entered, the actual write or erase operation is automatically performed internally.

Table 3.20.2 Flash Memory Functions

Major functions	Description
Automatic page program	Writes data automatically per page.
Automatic chip erase	Erases the entire area of the flash memory automatically.
Automatic block erase	Erases a selected block automatically.
Protect function	By writing a 4-bit protection code, the write or erase function can be individually inhibited for each block.

Note that addressing of operation commands is different from the case of standard commands due to the specific interface arrangements with the CPU. Also note that the flash memory is written in 32-bit blocks. So, 32-bit (word) data transfer commands must be used in writing the flash memory.

(1) Block configuration

User Boot Mode

Page Configuration

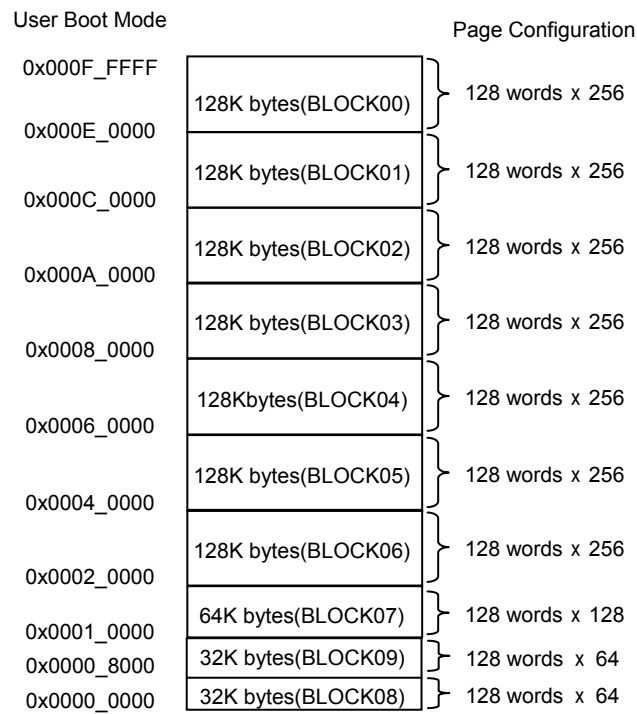| 0x000F_FFFF		
	128K bytes(BLOCK00)	128 words x 256
0x000E_0000		
	128K bytes(BLOCK01)	128 words x 256
0x000C_0000		
	128K bytes(BLOCK02)	128 words x 256
0x000A_0000		
	128K bytes(BLOCK03)	128 words x 256
0x0008_0000		
	128Kbytes(BLOCK04)	128 words x 256
0x0006_0000		
	128K bytes(BLOCK05)	128 words x 256
0x0004_0000		
	128K bytes(BLOCK06)	128 words x 256
0x0002_0000		
0x0001_0000	64K bytes(BLOCK07)	128 words x 128
0x0000_8000	32K bytes(BLOCK09)	128 words x 64
0x0000_0000	32K bytes(BLOCK08)	128 words x 64

Fig. 3.20.2 Block Configuration of Flash Memory (TMPM322F10FG)

(2) Basic operation

Generally speaking, this flash memory device has the following two operation modes:

- The mode to read memory data (Read mode)
- The mode to automatically erase or rewrite memory data (Automatic operation)

Transition to the automatic mode is made by executing a command sequence while it is in the memory read mode. In the automatic operation mode, flash memory data cannot be read and any commands stored in the flash memory cannot be executed. In the automatic operation mode, any interrupt or exception generation cannot set the device to the read mode except when a hardware reset is generated. During automatic operation, be sure not to cause any exceptions other than debug exceptions and reset while a debug port is connected. Any exception generation cannot set the device to the read mode except when a hardware reset is generated.

1) Read

When data is to be read, the flash memory must be set to the read mode. The flash memory will be set to the read mode immediately after power is applied, when CPU reset is removed, or when an automatic operation is normally terminated. In order to return to the read mode from other modes or after an automatic operation has been abnormally terminated, either the Read/reset command (a software command to be described later) or a hardware reset is used. The device must also be in the read mode when any command written on the flash memory is to be executed.

- **Read/reset command and Read command (software reset)**
  When ID-Read command is used, the reading operation is terminated instead of automatically returning to the read mode. In this case, the Read/reset command can be used to return the flash memory to the read mode. Also, when a command that has not been completely written has to be canceled, the Read/reset command must be used. The Read command is used to return to the read mode after executing 32-bit data transfer command to write the data "0x0000_00F0" to an arbitrary address of the flash memory.

- **With the Read/reset command, the device is returned to the read mode after completing the third bus write cycle.**

2) Command write
   This flash memory uses the command control method. Commands are executed by executing a command sequence to the flash memory. The flash memory executes automatic operation commands according to the address and data combinations applied (refer to Command Sequence).

   If it is desired to cancel a command write operation already in progress or when any incorrect command sequence has been entered, the Read/reset command is to be executed. Then, the flash memory will terminate the command execution and return to the read

   While commands are generally comprised of several bus cycles, the operation to apply 32-bit data transmit command to the flash memory is called "bus write cycle." The bus write cycles are to be in a specific sequential order and the flash memory will perform an automatic operation when the sequence of the bus write cycle data and address of a

command write operation is in accordance with a predefined specific sequence. If any bus write cycle does not follow a predefined command write sequence, the flash memory will terminate the command execution and return to the read mode.

(Note 1) Command sequences are executed from outside the flash memory area.

(Note 2) Each bus write cycle must be sequentially executed by 32-bit data transmit command. While a command sequence is being executed, access to the flash memory is prohibited. Also, don't generate any interrupt (except debug exceptions when a DSU probe is connected).If such an operation is made, it can result in an unexpected read access to the flash memory and the command sequencer may not be able to correctly recognize the command. While it could cause an abnormal termination of the command sequence, it is also possible that the written command is incorrectly recognized.

(Note 3) For the command sequencer to recognize a command, the device must be in the read mode prior to executing the command. Be sure to check before the first bus write cycle that the FCFLCS <RDY/BSY> bit is set to "1." It is recommended to subsequently execute a Read command.

(Note 4) Upon issuing a command, if any address or data is incorrectly written, be sure to perform a software reset to return to the read mode again.

---

(3) Reset

    Hardware reset

A hardware reset is used to cancel the operational mode set by the command write operation when forcibly termination during auto programming/ erasing or abnormal termination during auto operations occurs. The flash memory has a reset input as the memory block and it is connected to the CPU reset signal. Therefore, when the RESET input pin of this device is set to V_{IL} or when the CPU is reset due to any overflow of the watch dog timer, the flash memory will return to the read mode terminating any automatic operation that may be in progress. It should also be noted that applying a hardware reset during an automatic operation can result in incorrect rewriting of data. In such a case, be sure to perform the rewriting again.

Refer to Section 3.20.2.1 "Reset Operation" for CPU reset operations. After a given reset input, the CPU will read the reset vector data from the flash memory and starts operation after the reset is removed.

(4) Commands

1) Automatic Page Programming

Writing to a flash memory device is to make "1" data cells to "0" data cells. Any "0" data cell cannot be changed to a "1" data cell. For making "0" data cells to "1" data cells, it is necessary to perform an erase operation.

The automatic page programming function of this device writes data of each page. The TMPM322F10 contain 128 words in a page. A 128 word block is defined by a same [31:9] address and it starts from the address [8:0] = 0 and ends at the address [8:0] = 0x1FF. A 64 word block is defined by a same [31:8] address and it starts from the address [7:0] = 0 and ends at the address [7:0] = 0xFF. This programming unit is hereafter referred to as a "page."

Writing to data cells is automatically performed by an internal sequencer and no external control by the CPU is required. The state of automatic page programming (whether it is in writing operation or not) can be checked by the FCFLCS <RDY/BSY> register.

Also, any new command sequence is not accepted while it is in the automatic page programming mode. If it is desired to interrupt the automatic page programming, use the hardware reset function. If the operation is stopped by a hardware reset operation, it is necessary to once erase the page and then perform the automatic page programming again because writing to the page has not been normally terminated.

The automatic page programming operation is allowed only once for a page already erased. No programming can be performed twice or more times irrespective of the data cell value whether it is "1" or "0." Note that rewriting to a page that has been once written requires execution of the automatic block erase or automatic chip erase command before executing the automatic page programming command again. Note that an attempt to rewrite a page two or more times without erasing the content can cause damages to the device.

No automatic verify operation is performed internally to the device. So, be sure to read the data programmed to confirm that it has been correctly written.

The automatic page programming operation starts when the third bus write cycle of the command cycle is completed. On and after the fifth bus write cycle, data will be written sequentially starting from the next address of the address specified in the fourth bus write cycle (in the fourth bus write cycle, the page top address will be command written) (32 bits of data is input at a time). Be sure to use the 32-bit data transfer command in writing commands on and after the fourth bus cycle. In this, any 32-bit data transfer commands shall not be placed across word boundary. On and after the fifth bus write cycle, data is command written to the same page area. Even if it is desired to write the page only partially, it is required to perform the automatic page programming for the entire page. In this case, the address input for the fourth bus write cycle shall be set to the top address of the page. Be sure to perform command write operation with the input data set to "1" for the data cells not to be set to "0." For example, if the top address of a page is not to be written, set the input data of the fourth bus write cycle to 0xFFFFFFFF to command write the data.

Once the fourth bus cycle is executed, it is in the automatic programming operation. This condition can be checked by monitoring the register bit FCFLCS <RDY/BSY> (See Table 3.20.3). Any new command sequence is not accepted while it is in automatic page programming mode. If it is desired to stop operation, use the hardware reset function. Be careful in doing so because data cannot be written normally if the operation is interrupted.

When a single page has been command written normally terminating the automatic page writing process, the FCFLCS <RDY/BSY> bit is set to "1" and it returns to the read mode. When multiple pages are to be written, it is necessary to execute the page programming command for each page because the number of pages to be written by a single execution of the automatic page program command is limited to only one page. It is not allowed for automatic page programming to process input data across pages.

Data cannot be written to a protected block. When automatic programming is finished, it automatically returns to the read mode. This condition can be checked by monitoring FCFLCS <RDY/BSY> (Table 3.20.3). If automatic programming has failed, the flash memory is locked in the mode and will not return to the read mode. For returning to the read mode, it is necessary to execute hardware reset to reset the flash memory or the device. In this case, while writing to the address has failed, it is recommended not to use the device or not to use the block that includes the failed address.

(Note) **Software reset becomes ineffective in bus write cycles on and after the fourth bus write cycle of the automatic page programming command.**

---

2) Automatic chip erase

The automatic chip erase operation starts when the sixth bus write cycle of the command cycle is completed.

This condition can be checked by monitoring FCFLCS <RDY/BSY> (See Table 3.20.3). While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic chip erase operation. If it is desired to stop operation, use the hardware reset function. If the operation is forced to stop, it is necessary to perform the automatic chip erase operation again because the data erasing operation has not been normally terminated.

Also, any protected blocks cannot be erased. If all the blocks are protected, the automatic chip erase operation will not be performed and it returns to the read mode after completing the sixth bus read cycle of the command sequence. When an automatic chip erase operation is normally terminated, it automatically returns to the read mode. If an automatic chip erase operation has failed, the flash memory is locked in the mode and will not return to the read mode.

For returning to the read mode, it is necessary to execute hardware reset to reset the device. In this case, the failed block cannot be detected. It is recommended not to use the device anymore or to identify the failed block by using the block erase function for not to use the identified block anymore.

3) Automatic block erase (for each block)

The automatic block erase operation starts when the sixth bus write cycle of the command cycle is completed.

This status of the automatic block erase operation can be checked by monitoring FCFLCS <RDY/BSY> (See Table 3.20.3). While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic block erase operation. If it is desired to stop operation, use the hardware reset function. In this case, it is necessary to perform the automatic block erase operation again because the data erasing operation has not been normally terminated.

Also, any protected blocks cannot be erased. If an automatic block erase operation has failed, the flash memory is locked in the mode and will not return to the read mode. In this case, execute hardware reset to reset the device.

4) Automatic programming of protection bits (for each block)

This device is implemented with protection bits. This protection can be set for each block. See Table 3.20.8 for table of protection bit addresses. This device assigns 1 bit to 1 block as a protection bit. The applicable protection bit is specified by PBA in the seventh bus write cycle. By automatically programming the protection bits, write and/or erase functions can be inhibited (for protection) individually for each block. The protection status of each block can be checked by the FCFLCS <BLPRO> register to be described later. This status of the automatic programming operation to set protection bits can be checked by monitoring FCFLCS <RDY/BSY> (See Table 3.20.3). Any new command sequence is not accepted while automatic programming is in progress to program the protection bits. If it is desired to stop the programming operation, use the hardware reset function. In this case, it is necessary to perform the programming operation again because the protection bits may not have been correctly programmed. If all the protection bits have been programmed, all the FCFLCS <BLPRO> bits are set to "1" indicating that it is in the protected state (See Table 3.20.3). This disables subsequent writing and erasing of all blocks.

(Note) Software reset is ineffective in the seventh bus write cycle of the automatic protection bit programming command. The FCFLCS <RDY/BSY> bit turns to "0" after entering the seventh bus write cycle.

5) Automatic erasing of protection bits

Different results will be obtained when the automatic protection bit erase command is executed depending on the status of the protection bits and the security bits. It depends on the status of FCFLCS <BLPRO> whether all the <BLPRO> bits are set to "1" or not if FCSECBIT<SECBIT> is 0x1. Be sure to check the value of FCFLCS <BLPRO> before executing the automatic protection bit erase command. See chapter 3.20 for details.

・When all the FCFLCS <BLPRO> bits are set to "1" (all the protection bits are programmed):

When the automatic protection bit erase command is command written, the flash memory is automatically initialized within the device. When the seventh bus write cycle is completed, the entire area of the flash memory data cells is erased and then the protection bits are erased. This operation can be checked by monitoring FCFLCS <RDY/BSY>. If the automatic operation to erase protection bits is normally terminated, FLCS will be set to "0x00000001." While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that it has been correctly erased. For returning to the read mode while the automatic operation after the seventh bus cycle is in progress, it is necessary to use the hardware reset to reset the device. If this is done, it is necessary to check the status of protection bits by FCFLCS <BLPRO> after retuning to the read mode and perform either the automatic protection bit erase, automatic chip erase, or automatic block erase operation, as appropriate.

・When the FCFLCS <BLPRO> bits include "0" (not all the protection bits are programmed):

The protection condition can be canceled by the automatic protection bit erase operation. With this device, protection bits set by an individual block can be erased handling all the blocks at a time as shown in Table 3.20.8. The target bits are specified in the seventh bus write cycle and when the command is completed, the device is in a condition all the blocks are erased. The protection status of each block can be checked by FCFLCS <BLPRO> to be described later. This status of the programming operation for automatic protection bits can be checked by monitoring FCFLCS <RDY/BSY>. When the automatic operation to erase protection bits is normally terminated, the protection bits of FCFLCS <BLPRO> selected for erasure are set to "0."

In any case, any new command sequence is not accepted while it is in an automatic operation to erase protection bits. If it is desired to stop the operation, use the hardware reset function. When the automatic operation to erase protection bits is normally terminated, it returns to the read mode.

> **(Note) The FCFLCS <RDY/BSY> bit is "0" while in automatic operation and it turns to "1" when the automatic operation is terminated.**

6) ID-Read

Using the ID-Read command, you can obtain the type and other information on the flash memory contained in the device. The data to be loaded will be different depending on the address [15:14] of the fourth and subsequent bus write cycles (recommended input data is 0x00). On and after the fourth bus write cycle, when an arbitrary flash memory area is read, the ID value will be loaded. Once the fourth bus write cycle of an ID-Read command has passed, the device will not automatically return to the read mode. In this condition, the set of the fourth bus write cycle and ID-Read commands can be repetitively executed. For returning to the read mode, use the Read/reset command or hardware reset command.

(5) Flash control/ status register

This resister is used to monitor the status of the flash memory and to indicate the protection status of each block.

Table 3.20.3 Flash Control Register

FCFLCS
0x41FF_F020

	31	30	29	28	27	26	25	24
bit Symbol	–	–	–	–	–	–	BLPRO09	BLPRO08
Read/Write	R						R	R
After reset	0						(Note2)	(Note2)
Function	"0" is read.						Protection for Block 9 0: disabled 1:enabled	Protection for Block 8 0: disabled 1:enabled
	23	22	21	20	19	18	17	16
bit Symbol	BLPRO07	BLPRO06	BLPR05	BLPRO04	BLPRO03	BLPRO02	BLPRO01	BLPRO00
Read/Write	R	R	R	R	R	R	R	R
After reset	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)
Function	Protection for Block 7 0: disabled 1:enabled	Protection for Block 6 0: disabled 1:enabled	Protection for Block 5 0: disabled 1:enabled	Protection for Block 4 0: disabled 1:enabled	Protection for Block 3 0: disabled 1:enabled	Protection for Block 2 0: disabled 1:enabled	Protection for Block 0 0: disabled 1:enabled	Protection for Block 1 0: disabled 1:enabled
	15	14	13	12	11	10	9	8
bit Symbol	–	–	–	–	–	–	–	–
Read/Write	R							
After reset	0							
Function	"0" is read.							
	7	6	5	4	3	2	1	0
bit Symbol	–	–	–	–	–	–	–	RDY/BSY
Read/Write	R							R
After reset	0							1
Function	"0" is read.							Ready/Busy (Note 1) 0:Auto operating 1:Auto operation terminated

Bit 0: Ready/Busy flag bit

The RDY/BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1."

Bit [25:16]: Protection status bits

Each of the protection bits represents the protection status of the corresponding block. When a bit is set to "1," it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.

(Note 1)   **This command must be issued in the ready state. Issuing the command in the busy state may disable both correct command transmission and further command input. To exit from the condition, execute system reset. System reset requires at least 0.5 microseconds regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.**

(Note 2)   **The value varies depending on protection applied.**

Table 3.20.4 Security bit register

FCSECBIT
0x41FF_F010

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0							
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0							
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0							
Function	"0" is read							
	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	-	-	SECBIT
Read/Write	R							R/W
After reset	0							1
Function	"0" is read							Security bits 0:disabled 1:enabled

(Note) This register is initialized only by power-on reset.

(6) List of Command Sequences

Table 3.20.5 Flash Memory Access from the Internal CPU

Command sequence	First bus cycle Addr. Data	Second bus cycle Addr. Data	Third bus cycle Addr. Data	Fourth bus cycle Addr. Data	Fifth bus cycle Addr. Data	Sixth bus cycle Addr. Data	Seventh bus cycle Addr. Data
Read	0xXX	−	−	−	−	−	−
	0xF0	−	−	−	−	−	−
Read/Reset	0x54XX	0xAAXX	0x54XX	RA	−	−	−
	0xAA	0x55	0xF0	RD	−	−	−
ID-Read	0x54XX	0xAAXX	0x54XX	IA	0xXX	−	−
	0xAA	0x55	0x90	0x00	ID	−	−
Automatic page programming	0x54XX	0xAAXX	0x54XX	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic chip erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	−
	0xAA	0x55	0x80	0xAA	0x55	0x10	−
Auto Block erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	BA	−
	0xAA	0x55	0x80	0xAA	0x55	0x30	−
Protection bit programming	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
Protection bit erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

Supplementary explanation

- RA: Read address
- RD: Read data
- IA: ID address
- ID: ID data
- PA: Program page address
  PD: Program data (32 bit data)
  After the fourth bus cycle, enter data in the order of the address for a page.
- BA: Block address
- PBA: Protection bit address

(Note 1) Always set "0" to the address bits [1:0] in the entire bus cycle. (Recommendable setting values to bits [7:2] are "0".)   **(Note 2)** Bus cycles are "bus write cycles" except for the second bus cycle of the Read command, the fourth bus cycle of the Read/reset command, and the fifth bus cycle of the ID-Read command. Bus write cycles are executed by 32-bit data transfer commands. The address [31:16] in each bus write cycle should be the target flash memory address [31:16] of the command sequence. Use "Addr." in the table for the address [15:0].

(7) Address bit configuration for bus write cycles

Table 3.20.6 Address Bit Configuration for Bus Write Cycles

Address	Addr [31:20]	Addr [19]	Addr [18:17]	Addr [16]	Addr [15]	Addr [14]	Addr [13:11]	Addr [10:9]	Addr [8]	Addr [7:0]
Normal commands	**Normal bus write cycle address configuration**									
	Flash area	"0" is recommended.			Command[15:8]					Addr[1:0]="0" (fixed) Others:0 (recommended)

| Block erase | **BA: Block address (Set the sixth bus write cycle address for block erase operation)** | | | | | | | | | |
| | Flash area | Block selection [19:15] | | | Addr[1:0]="0" (fixed) , Others:0 (recommended) | | | | | |

| Auto page programming | **PA: Program page address (Set the fourth bus write cycle address for page programming operation)** | | | | | | | | | |
| | Flash area | Page selection [19:9] | | | | | | | | Addr[1:0]="0" (fixed) Others:0 (recommended) |

| ID-READ | **IA: ID address (Set the fourth bus write cycle address for ID-Read operation)** | | | | | | | | | |
| | Flash area | "0" is recommended. | | | ID address [15:14] | | Addr[1:0]="0" (fixed) , Others:0 (recommended) | | | |

| Protection bit programming | **PBA: Protection bit address (Set the seventh bus erase cycle address for protection bit erasure)** | | | | | | | | | |
| | Flash area | Fixed to "0". | Protection bit selection [18:17] | Fixed to "0". | | | | Protection bit selection [10:9] | | Addr[1:0]="0" (fixed) Others:0 (recommended) |

| Protection bit erase | **PBA: Protection bit address (Set the seventh bus erase cycle address for protection bit erasure)** | | | | | | | | | |
| | Flash area | Fixed to "0". | Protection bit selection [18:17] | Addr[1:0]="0" (fixed) Others:0 (recommended) | | | | | | |

(Note 1)	**Table 3.20.5 "Flash Memory Access from the Internal CPU" can also be used.**
(Note 2)	**Address setting can be performed according to the "Normal bus write cycle address configuration" from the first bus cycle.**
(Note 3)	**"0" is recommended" can be changed as necessary.**

Table 3.20.7 Block Address Table

Block	Address (User boot mode)	Size (Kbyte)
0	0x000E_0000-0x000F_FFFF	128
1	0x000C_0000-0x000D_FFFF	128
2	0x000A_0000-0x000B_FFFF	128
3	0x0008_0000-0x0009_FFFF	128
4	0x0006_0000-0x0007_FFFF	128
5	0x0004_0000-0x0005_FFFF	128
6	0x0002_0000-0x0003_FFFF	128
7	0x0001_0000-0x0001_FFFF	64
9	0x0000_8000-0x0000_FFFF	32
8	0x0000_0000-0x0000_7FFF	32

As block address, specify any address in the block to be erased.

(Note) As for the addresses from the first to the fifth bus cycles, specify the upper 4 bit with the corresponding flash memory addresses of the blocks to be erased.

Table 3.20.8 Protection Bit Programming Address Table

Block	Protection bit	Size (Kbyte)	The seventh bus write cycle address				
			Address [18]	Address [17]	Address [16:11]	Address [10]	Address [9]
Block08	BLPRO08	32KB	1	0		0	0
Block09	BLPRO09	32KB	1	0		0	1
Block07	BLPRO07	64KB	0	1		1	1
Block06	BLPRO06	128KB	0	1		1	0
Block05	BLPRO05	128KB	0	1	Fixed to "0".	0	1
Block04	BLPRO04	128KB	0	1		0	0
Block03	BLPRO03	128KB	0	0		1	1
Block02	BLPRO02	128KB	0	0		1	0
Block01	BLPRO01	128KB	0	0		0	1
Block00	BLPRO00	128KB	0	0		0	0

Table 3.20.9 Protection Bit Erase Address Table

Block	Protection bit	The seventh bus write cycle address [18:17]	
		Address [18]	Address [17]
Block08～09	BLPRO08～09	1	0
Block07～04	BLPRO07～04	0	1
Block03～00	BLPRO03～00	0	0

(Note) The protection bit erase command cannot erase by individual block.

**Table 3.20.10 The ID-Read command's fourth bus write cycle ID address (IA) and
the data to be read by the following 32-bit data transfer command (ID)**

IA [15:14]	ID [7: 0]	Code
00b	0x98	Manufacturer code
01b	0x5A	Device code
10b	Reserved	---
11b	0x10	Macro code

(8) Flowchart
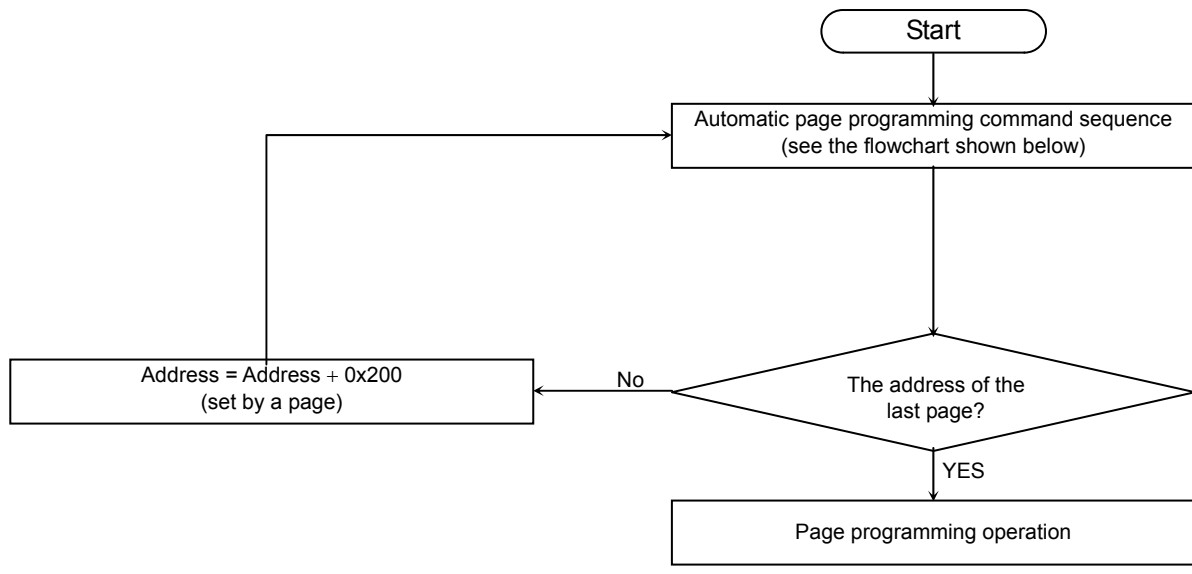
```
                                          ┌─────────────┐
                                          │    Start    │
                                          └─────────────┘
                                                 │
                                                 ▼
                    ┌──────────────────────────────────────────────────┐
          ┌────────▶│    Automatic page programming command sequence    │
          │         │        (see the flowchart shown below)            │
          │         └──────────────────────────────────────────────────┘
          │                                      │
          │                                      ▼
 ┌──────────────────────┐              ╱────────────────────╲
 │ Address = Address + 0x200 │   No    ╱   The address of the  ╲
 │    (set by a page)   │◀────────────   last page?          
 └──────────────────────┘              ╲                    ╱
                                        ╲────────────────────╱
                                                 │ YES
                                                 ▼
                                      ┌──────────────────────────┐
                                      │ Page programming operation│
                                      └──────────────────────────┘
```
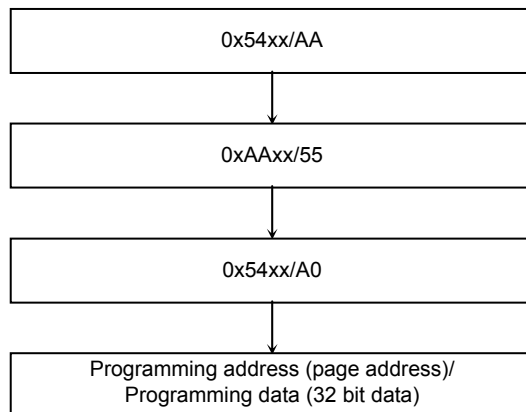
Automatic Page Programming Command Sequence (Address/ Command)

```
              ┌──────────────────────────────────────────┐
              │                0x54xx/AA                  │
              └──────────────────────────────────────────┘
                                   │
                                   ▼
              ┌──────────────────────────────────────────┐
              │                0xAAxx/55                  │
              └──────────────────────────────────────────┘
                                   │
                                   ▼
              ┌──────────────────────────────────────────┐
              │                0x54xx/A0                  │
              └──────────────────────────────────────────┘
                                   │
                                   ▼
              ┌──────────────────────────────────────────┐
              │   Programming address (page address)/     │
              │    Programming data (32 bit data)         │
              └──────────────────────────────────────────┘
```

┌──┐
│ (Note) Command sequence is executed by   │
│ 0x54xx or 0x55xx.   │
└──┘

Fig. 3.20.10 Automatic Programming

```
                    ┌──────────────────────┐
                    │        Start         │
                    └──────────┬───────────┘
                               │
              ┌────────────────▼─────────────────┐
              │ Automatic chip erase command     │
              │ sequence                         │
              │ (see the flowchart shown below)  │
              └────────────────┬─────────────────┘
                               │
              ┌────────────────▼─────────────────┐
              │  Automatic chip erase completed  │
              └──────────────────────────────────┘
```

Automatic chip erase command sequence (address/ command)

```
┌──────────────────────────────┐
│          0x54xx/AAH          │
└───────────────┬──────────────┘
                │
┌───────────────▼──────────────┐
│          0xAAxx/55H          │
└───────────────┬──────────────┘
                │
┌───────────────▼──────────────┐
│          0x54xx/80H          │
└───────────────┬──────────────┘
                │
┌───────────────▼──────────────┐
│          0x54xx/AAH          │
└───────────────┬──────────────┘
                │
┌───────────────▼──────────────┐
│          0xAAxx/55H          │
└───────────────┬──────────────┘
                │
┌───────────────▼──────────────┐
│          0x54xx/10H          │
└──────────────────────────────┘
```

Automatic block/ multi-block erase command sequence (address/ command)

```
┌──────────────────────────────┐
│          0x54xx/AAH          │
└───────────────┬──────────────┘
                │
┌───────────────▼──────────────┐
│          0xAAxx/55H          │
└───────────────┬──────────────┘
                │
┌───────────────▼──────────────┐
│          0x54xx/80H          │
└───────────────┬──────────────┘
                │
┌───────────────▼──────────────┐
│          0x54xx/AAH          │
└───────────────┬──────────────┘
                │
┌───────────────▼──────────────┐
│          0xAAxx/55H          │
└───────────────┬──────────────┘
                │
┌───────────────▼──────────────┐
│       Block address/30H      │
└──────────────────────────────┘
```
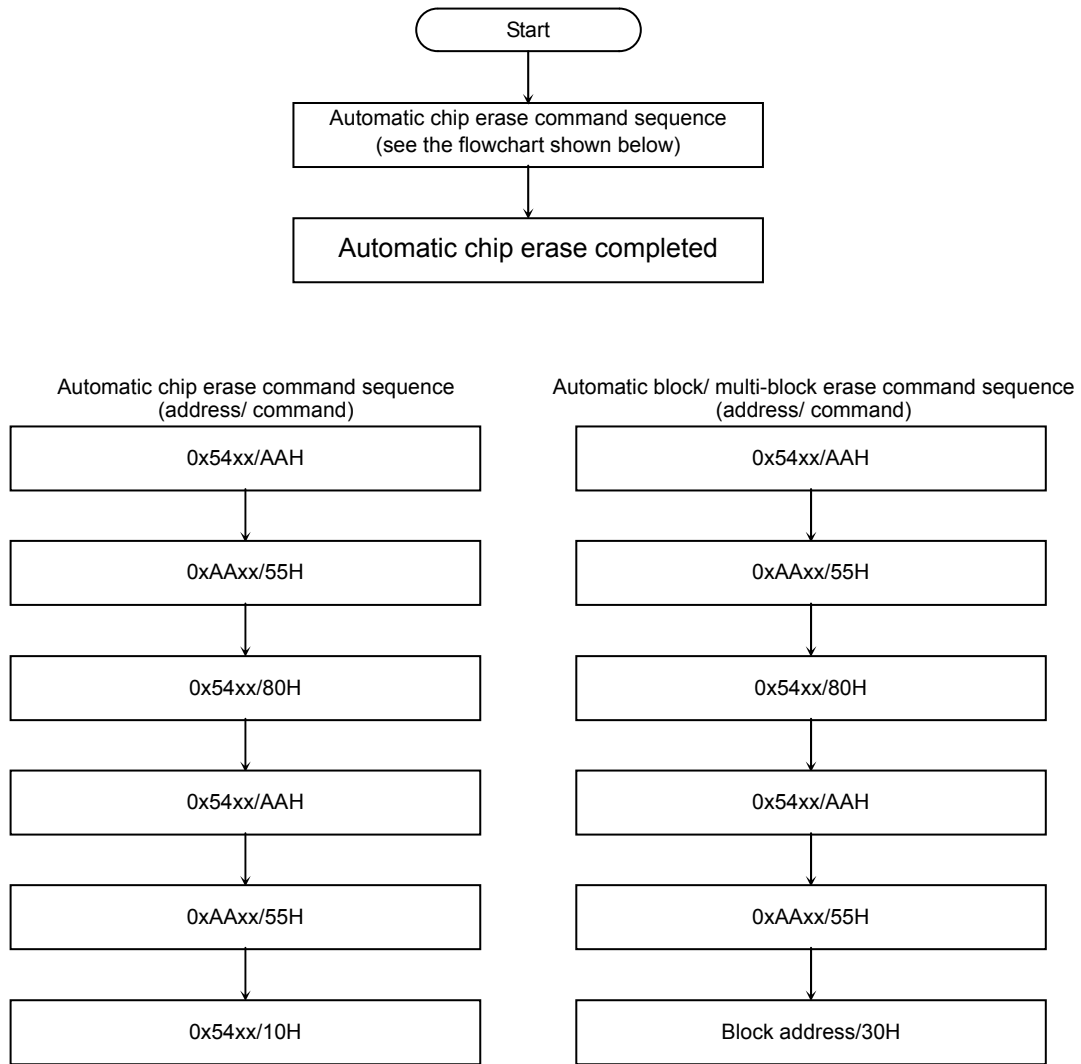
(Note) Command sequence is executed by 0x54xx or 0x55xx.

Fig. 3.20.11 Automatic Erase

3.21 ROM protection

3.21.1 Outline

The TMPM322 offers two kinds of ROM protection/ security functions. One is a write/ erase-protection function for the internal flash ROM data. The other is a security function that restricts internal flash ROM data readout and debugging.

3.21.2 Features

3.21.2.1 Write/ erase-protection function

The write/ erase-protection function enables the internal flash to prohibit the writing and erasing operation for each block.

This function is available with a single chip mode, single boot mode and writer mode. To activate the function, write "1" to the corresponding bits to a block to protect. Writing "0" to the bits cancels the protection. The protection settings of the bits can be monitored by the FCFLCS <BLPROxx> bit. See chapter 3.20 for programming details.

3.21.2.2   Security Function

The security function restricts flash ROM data readout and debugging.
This function is available under the conditions shown below.

   1) The FCSECBIT <SECBIT> bit is set to"1".
   2) All the protection bits (the FCFLCS<BLPROxx> bits) used for the write/erase-protection
       function are set to "1".

Note) The FCSECBIT <SECBIT> bit is set to "1" at a power-on reset right after power-on.

Table 3.21.1 shows details of the restrictions by the security function

Table 3.21.1 Restrictions by the security function

Item	Details
1) ROM data readout	Data in the ROM area cannot be read out when writer mode is set. By executing readout, the company code 0x0098 is read. The ROM reading operation is available with a single chip mode and single boot mode.
2) Debug port	Communication of SWD and trace are prohibited.
3) Command for flash memory	Writing a command to the flash memory is prohibited. An attempt to erase the contents in the bits used for the write/erase-protection erases all the protection bits.

3.21.3 Register

The flash control register shows the status of the flash memory operation and the protection of each block.

Table 3.21.2 Flash Control Register

FCFLCS
(0x41FF_F020)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	BLPRO09	BLPRO08
Read/Write	R						R	R
After reset	0						(Note2)	(Note2)
Function	"0" is read.						Protection for Block 9 0: disabled 1:enabled	Protection for Block 8 0: disabled 1:enabled

	23	22	21	20	19	18	17	16
bit Symbol	BLPRO07	BLPRO06	BLPRO05	BLPRO04	BLPRO03	BLPRO02	BLPRO01	BLPRO00
Read/Write	R	R	R	R	R	R	R	R
After reset	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)
Function	Protection for Block 7 0: disabled 1:enabled	Protection for Block 6 0: disabled 1:enabled	Protection for Block 5 0: disabled 1:enabled	Protection for Block 4 0: disabled 1:enabled	Protection for Block 3 0: disabled 1:enabled	Protection for Block 2 0: disabled 1:enabled	Protection for Block 1 0: disabled 1:enabled	Protection for Block 0 0: disabled 1:enabled

	15	14	13	12	11	10	9	8
bit Symbol	–	–	–	–	–	–	–	–
Read/Write	R							
After reset	0							
Function	"0" is read.							

	7	6	5	4	3	2	1	0
bit Symbol	–	–	–	–	–	–	–	RDY/BSY
Read/Write	R							R
After reset	0							1
Function	"0" is read.							Ready/Busy (Note 1) 0:Auto operating 1:Auto operation terminated

Bit 0: Ready/Busy flag bit

The RDY/BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1."

Bit [25:16]: Protection status bits

Each of the protection bits represents the protection status of the corresponding block. When a bit is set to "1," it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.

(Note 1) This command must be issued in the ready state. Issuing the command in the busy state may disable both correct command transmission and further command input. To exit from the condition, execute system reset. System reset requires at least 0.5 microseconds regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.

(Note 2) The value varies depending on protection applied.

Table 3.21.3 Security bit register

FCSECBIT
0x41FF_F010

	31	30	29	28	27	26	25	24
bit Symbol	–	–	–	–	–	–	–	–
Read/Write	R							
After reset	0							
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	–	–	–	–	–	–	–	–
Read/Write	R							
After reset	0							
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	–	–	–	–	–	–	–	–
Read/Write	R							
After reset	0							
Function	"0" is read							
	7	6	5	4	3	2	1	0
bit Symbol	–	–	–	–	–	–	–	SECBIT
Read/Write	R							R/W
After reset	0							1
Function	"0" is read							Security bits 0:disabled 1:enabled

(Note) This register is initialized only by power-on reset.

3.21.4 Writing and erasing

3.21.4.1   Protection bits

Writing and erasing protection bits are available with a single chip mode, single boot mode and writer mode.
Writing to the protection bits is done on block-by-block basis.
Erasing of the protection bits is done by two groups of the blocks: block 00 through 03, block 04 through 07 and block 08 through 09. When the settings for all the blocks are "1", erasing must be done after setting the FCSECBIT <SECBIT> bit to "0". Setting "1" at that situation erases all the protection bits. To write and erase the protection bits, command sequence is used.
See chapter 20 for details.

3.21.4.2   Security bit

Rewriting of the security bits is available with a single chip mode and single boot mode. The FCSECBIT <SECBIT> bit that activates security function is set to "1" at a power-on reset right after power-on. The bit is rewritten by the following procedure.

1) Write the code 0xa74a9d23 to FCSECBIT register.
2) Write data within 16 clocks from the above.

Note) The above procedure is enabled only when using 32-bit data transfer command.

3.22 Special Function Registers

3.22.1 Outline

Table 3.22.1 shows the base address list of peripherals built in TMPM322.
The address of each peripherals are calculated by these base address plus offset address of peripherals.
Access to the address do not assigned register is prohibited.

Table 3.22.1 Base address of Peripheral

Address	Internal Peripheral	
0x4200_0000 ~ 0x5FFF_FFFF	Fault	
0x41FF_F000	IO-4 (IO BUS)	FLASH I/F
0x400F_5000 ~ 0x41FF_EFFF	Fault	
0x400F_4000	IO-3 (IO BUS)	CG
0x400F_3000		RTC (1ch)
0x400F_2000		WDT (1ch)
0x400F_1000		KWUP (4ch)
0x400F_0000		10bit ADC (16ch)
0x400E_3000		RMC (2ch)
0x400E_2000		Reserved
0x400E_1000		SIO/UART (12ch)
0x400E_0000		SIO/I2C (5ch)
0x400D_0000		16bit Timer (16ch)
0x400C_0000		PORT (A~P)
0x4004_2000 ~ 0x400B_FFFF	Fault	
0x4004_1000	IO-2 (APB)	Reserved
0x4004_0000		SSP (1ch)
0x4000_6000 ~ 0x4003_FFFF	Fault	
0x4000_5000	IO-1 (AHB)	Reserved
0x4000_4000		Reserved
0x4000_3000		USBHC
0x4000_2000		Reserved
0x4000_1000		SMC
0x4000_0000		DMAC (2ch)

3.22.2   Resisters Description

3.22.2.1   Single Master DMAC (DMAC)

Base Address = 0x4000_0000

RegisterName	Address (Base +)	Description
DMACIntStaus	0x0000	DMAC Interrupt Status Register
DMACIntTCStatus	0x0004	DMAC Interrupt Terminal Count Status Register
DMACIntTCClear	0x0008	DMAC Interrupt Terminal Count Clear Register
DMACIntErrorStatus	0x000C	DMAC Interrupt Error Status Register
DMACIntErrClr	0x0010	DMAC Interrupt Error Clear Register
DMACRawIntTCStatus	0x0014	DMAC Raw Interrupt Terminal Count Status Register
DMACRawIntErrorStatus	0x0018	DMAC Raw Error Interrupt Status Register
DMACEnbldChns	0x001C	DMAC Enabled Channel Register
DMACSoftBReq	0x0020	DMAC Software Burst Request Register
DMACSoftSReq	0x0024	DMAC Software Single Request Register
–	0x0028	Reserved
–	0x002C	Reserved
DMACConfiguration	0x0030	DMAC Configuration Register
–	0x0034	Reserved
DMACC0SrcAddr	0x0100	DMAC Channel0 Source Address Register
DMACC0DestAddr	0x0104	DMAC Channel0 Destination Address Register
DMACC0LLI	0x0108	DMAC Channel0 Linked List Item Register
DMACC0Control	0x010C	DMAC Channel0 Control Register
DMACC0Configuration	0x0110	DMAC Channel0 Configuration Register
DMACC1SrcAddr	0x0120	DMAC Channel1 Source Address Register
DMACC1DestAddr	0x0124	DMAC Channel1 Destination Address Register
DMACC1LLI	0x0128	DMAC Channel1 Linked List Item Register
DMACC1Control	0x012C	DMAC Channel1 Control Register
DMACC1Configuration	0x0130	DMAC Channel1 Configuration Register

3.22.2.2 Static Memory Controller（SMC）

Base Address = 0x4000_1000

RegisterName	Address (Base +)	Description
–	0x0000	Reserved
smc_memif_cfg	0x0004	SMC Memory Interface Configuration Register
–	0x0008	Reserved
–	0x000C	Reserved
smc_direct_cmd	0x0010	SMC Direct Command Register
smc_set_cycles	0x0014	SMC Set Cycles Register
smc_set_opmode	0x0018	SMC Set Opmode Register
–	0x0020	Reserved
smc_sram_cycles0_0	0x0100	SMC SRAM Cycles Registers <0>
smc_sram_cycles0_1	0x0120	SMC SRAM Cycles Registers <1>
smc_sram_cycles0_2	0x0140	SMC SRAM Cycles Registers <2>
smc_sram_cycles0_3	0x0160	SMC SRAM Cycles Registers <3>
smc_opmode0_0	0x0104	SMC Opmode Registers <0>
smc_opmode0_1	0x0124	SMC Opmode Registers <1>
smc_opmode0_2	0x0144	SMC Opmode Registers <2>
smc_opmode0_3	0x0164	SMC Opmode Registers <3>
–	0x0200	Reserved
–	0x0204	Reserved
–	0x0E00	Reserved
–	0x0E04	Reserved
–	0x0E08	Reserved
–	0x0FE0 - 0x0FEC	Reserved
–	0x0FF0 - 0x0FFC	Reserved

3.22.2.3 Synchronous Serial Port（SSP）

Base Address = 0x4004_0000

RegisterName	Address (Base +)	Description
SSPCR0	0x0000	SSP Control Register 0
SSPCR1	0x0004	SSP Control Register 1
SSPDR	0x0008	SSP Data Register
SSPSR	0x000C	SSP Status Register
SSPCPSR	0x0010	SSP Clock Prescaler Register
SSPIMSC	0x0014	SSP Interrupt Mask Set and Clear Register
SSPRIS	0x0018	SSP Raw Interrupt Status Register
SSPMIS	0x001C	SSP Masked Interrupt Status Register
SSPICR	0x0020	SSP Interrupt Clear Register
SSPDMACR	0x0024	SSP DMA Control Register
–	0x0028 ~ 0xFFC	Reserved

3.22.2.4   PORT

Port Name	Base Address	0x00	0x04	0x08	0x0C	0x10	0x28	0x2C	0x30	0x38
Port A	0x400C_0000	PADATA	PACR	PAFR1	-	-	PAOD	PAPUP	-	PAIE
Port B	0x400C_0100	PBDATA	PBCR	PBFR1	-	-	PBOD	PBPUP	-	PBIE
Port C	0x400C_0200	PCDATA	PCCR	PCFR1	PCFR2	PCFR3	PCOD	PCPUP	-	PCIE
Port D	0x400C_0300	PDDATA	PDCR	PDFR1	PDFR2	PDFR3	PDOD	PDPUP	-	PDIE
Port E	0x400C_0400	PEDATA	PECR	PEFR1	PEFR2	PEFR3	PEOD	PEPUP	-	PEIE
Port F	0x400C_0500	PFDATA	PFCR	PFFR1	-	-	PFOD	PFPUP	-	PFIE
Port G	0x400C_0600	PGDATA	PGCR	PGFR1	PGFR2	PGFR3	PGOD	PGPUP	-	PGIE
Port H	0x400C_0700	PHDATA	PHCR	PHFR1	PHFR2	-	PHOD	PHPUP	-	PHIE
Port I	0x400C_0800	PIDATA	PICR	-	-	-	PIOD	PIPUP	-	PIIE
Port J	0x400C_0900	PJDATA	-	-	PJFR2	-	-	PJPUP	-	PJIE
Port K	0x400C_0A00	PKDATA	-	-	-	-	-	PKPUP	-	PKIE
Port L	0x400C_0B00	PLDATA	PLCR	PLFR1	PLFR2	PLFR3	PLOD	PLPUP	-	PLIE
Port M	0x400C_0C00	PMDATA	PMCR	PMFR1	PMFR2	PMFR3	PMOD	PMPUP	-	PMIE
Port N	0x400C_0D00	PNDATA	PNCR	PNFR1	PNFR2	PNFR3	PNOD	PNPUP	-	PNIE
Port O	0x400C_0E00	PODATA	POCR	POFR1	POFR2	POFR3	POOD	POPUP	-	POIE
Port P	0x400C_0F00	PPDATA	PPCR	PPFR1	PPFR2	-	PPOD	PPPUP	-	PPIE

Note "-" : Reserved

3.22.2.5   16-bit Timer（TMRB）

Channel	Base Address
Channel 0	0x400D_0000
Channel 1	0x400D_0100
Channel 2	0x400D_0200
Channel 3	0x400D_0300
Channel 4	0x400D_0400
Channel 5	0x400D_0500
Channel 6	0x400D_0600
Channel 7	0x400D_0700
Channel 8	0x400D_0800
Channel 9	0x400D_0900
Channel A	0x400D_0A00
Channel B	0x400D_0B00
Channel C	0x400D_0C00
Channel D	0x400D_0D00
Channel E	0x400D_0E00
Channel F	0x400D_0F00

Register Name (x= 0 ~ F)	Address (base +)	Description
TBxEN	0x0000	Timer Enable Register
TBxRUN	0x0004	Timer RUN Register
TBxCR	0x0008	Timer Control Register
TBxMOD	0x000C	Timer Mode Register
TBxFFCR	0x0010	Timer Flip-Flop Control Register
TBxST	0x0014	Timer Status Register
TBxIM	0x0018	Interrupt Mask Register
TBxUC0	0x001C	Timer Up Counter Register
TBxRG0	0x0020	Timer Register 0
TBxRG1	0x0024	Timer Register 1
TBxCP0	0x0028	Capture register 0
TBxCP1	0x002C	Capture register 1

3.22.2.6   Serial Bus Interface (SBI/SIO)

Channel	Base Address
Channel 0	0x400E_0000
Channel 1	0x400E_0100
Channel 2	0x400E_0200
Channel 3	0x400E_0300
Channel 4	0x400E_0400

Register Name (x= 0 ~ 4)	Address (base +)	Description
SBIxCR0	0x0000	SBI Control Register 0
SBIxCR1	0x0004	SBI Control Register 1
SBIxDBR	0x0008	SBI Data Buffer Register
SBIxI2CAR	0x000C	SBI I^{2}C bus Address Register
SBIxCR2/SR	0x0010	SBI Control Register 2/ SBI Status Register
SBIxBR0	0x0014	SBI Baud Rate Register 0

3.22.2.7   General-purpose Serial Interface (SIO/UART)

Channel	Base Address
Channel 0	0x400E_1000
Channel 1	0x400E_1100
Channel 2	0x400E_1200
Channel 3	0x400E_1300
Channel 4	0x400E_1400
Channel 5	0x400E_1500
Channel 6	0x400E_1600
Channel 7	0x400E_1700
Channel 8	0x400E_1800
Channel 9	0x400E_1900
Channel 10	0x400E_1A00
Channel 11	0x400E_1B00

Register Name (x= 0 ~ 11)	Address (base +)	Description
SCxEN	0x0000	SIO Enable Register
SCxBUF	0x0004	SIO Transmit/ Receive Buffer Register
SCxCR	0x0008	SIO Control Register
SCxMOD0	0x000C	SIO Mode Control Register 0
SCxBRCR	0x0010	SIO Baud Rate Generator Control
SCxBRADD	0x0014	SIO Baud rate Generator Control 2
SCxMOD1	0x0018	SIO Mode Control Register 1
SCxMOD2	0x001C	SIO Mode Control Register 2
SCxRFC	0x0020	SIO Receive FIFO Configuration Register
SCxTFC	0x0024	SIO Transmit FIFO Configuration Register
SCxRST	0x0028	SIO Receive FIFO Status Register
SCxTST	0x002C	SIO Transmit FIFO Status Register
SCxFCNF	0x0030	SIO FIFO Configuration Register

3.22.2.8 Remote Control signal preprocessor (RMC)

Channel	Base Address
Channel 0	0x400E_3000
Channel 1	0x400E_3100

Register Name (x= 0 ~ 1)	Address (base +)	Description
RMCxEN	0x0000	RMC Enable Register
RMCxREN	0x0004	RMC Receive Enable Register
RMCxRBUF1	0x0008	RMC Receive Data Buffer Register 1
RMCxRBUF2	0x000C	RMC Receive Data Buffer Register 2
RMCxRBUF3	0x0010	RMC Receive Data Buffer Register 3
RMCxRCR1	0x0014	RMC Control Register 1
RMCxRCR2	0x0018	RMC Control Register 2
RMCxRCR3	0x001C	RMC Control Register 3
RMCxRCR4	0x0020	RMC Control Register 4
RMCxRSTAT	0x0024	RMC Status Register
RMCxEND1	0x0028	RMC Receive End bit Number Register 1
RMCxEND2	0x002C	RMC Receive End bit Number Register 2
RMCxEND3	0x0030	RMC Receive End bit Number Register 3
RMCxFSSEL	0x0034	RMC Frequency Selection Register

3.22.2.9 10-bit A/D Converter (A/DC)

Base Address = 0x400F_0000

Register Name	Address (base +)	Description
ADCLK	0x0000	A/D Conversion Clock Setting Register
ADMOD0	0x0004	A/D Mode Control Register 0
ADMOD1	0x0008	A/D Mode Control Register 1
ADMOD2	0x000C	A/D Mode Control Register 2
ADMOD3	0x0010	A/D Mode Control Register 3
ADMOD4	0x0014	A/D Mode Control Register 4
ADMOD5	0x0018	A/D Mode Control Register 5
–	0x001C	Reserved
ADCBAS	0x0020	A/D Conversion Accuracy Setting Register
–	0x0024	Reserved
–	0x0028	Reserved
–	0x002C	Reserved
ADREG08	0x0030	A/D Conversion Result Register 08L
ADREG19	0x0034	A/D Conversion Result Register 19L
ADREG2A	0x0038	A/D Conversion Result Register 2AL
ADREG3B	0x003C	A/D Conversion Result Register 3BL
ADREG4C	0x0040	A/D Conversion Result Register 4CL
ADREG5D	0x0044	A/D Conversion Result Register 5DL
ADREG6E	0x0048	A/D Conversion Result Register 6EL
ADREG7F	0x004C	A/D Conversion Result Register 7FL
ADREGSP	0x0050	A/D Conversion Result Register SP
ADCMPREG0	0x0054	A/D Conversion Result Comparison Register 0
ADCMPREG1	0x0058	A/D Conversion Result Comparison Register 1

3.22.2.10 Key on Wake Up (KWUP)

Base Address = 0x400F_1000

Register Name	Address (base +)	Description
KWUPCR0	0x0000	KWUP Control Register 0
KWUPCR1	0x0004	KWUP Control Register 1
KWUPCR2	0x0008	KWUP Control Register 2
KWUPCR3	0x000C	KWUP Control Register 3
KWUPPKEY	0x0080	KWUP Port Monitor Register
KWUPCNT	0x0084	KWUP Control Register
KWUPCLR	0x0088	KWUP Interrupt All Clear Register
KWUPINT	0x008C	KWUP Interrupt Monitor Register

3.22.2.11 Watch Dog Timer (WDT)

Base Address = 0x400F_2000

Register Name	Address (base +)	Description
WDMOD	0x0000	WDT Mode Register
WDCR	0x0004	WDT Control Register

3.22.2.12 Real Time Clock (RTC)

Base Address = 0x400F_3000

Register Name	Address (base +)	Description
RTCSECR	0x0000	Second Column Register
RTCMINR	0x0001	Minute Column Register
RTCHOURR	0x0002	Hour Column Register
–	0x0003	Reserved
RTCDAYR	0x0004	Day of the Week Column Register
RTCDATER	0x0005	Day Column Register
RTCMONTHR	0x0006	Month Column Register
RTCYEARR	0x0007	Year Column Register
RTCPAGER	0x0008	PAGE Register
RTCRESTR	0x000C	Reset Register

3.22.2.13 Clock Generator (CG)

Base Address = 0x400F_4000

Register Name	Address (base +)	Description
CGSYSCR	0x0000	System Control Register
CGOSCCR	0x0004	Oscillation Control Register
CGSTBYCR	0x0008	Standby Control Register
CGPLLSEL	0x000C	PLL Selection Register
CGCKSEL	0x0010	System Clock Selection Register
CGICRCG	0x0014	CG Interrupt Request Clear Register
CGNMIFLG	0x0018	NMI Flag Register
CGRSTFLG	0x001C	Reset Flag Register
CGIMCGA	0x0020	CG Interrupt Mode Control Register A
CGIMCGB	0x0024	CG Interrupt Mode Control Register B
CGIMCGC	0x0028	CG Interrupt Mode Control Register C
CGIMCGD	0x002C	CG Interrupt Mode Control Register D
CGIMCGE	0x0030	CG Interrupt Mode Control Register E
CGIMCGF	0x0034	CG Interrupt Mode Control Register F

3.22.2.14 FLASH Interface

Base Address = 0x41FF_F000

Register Name	Address (base+)	Description
FCSECBIT	0x0010	Security Bit Register
FCFLCS	0x0020	FLASH Control Register

3.22.2.15 RAM Interface

Base Address = 0x41FF_F000

Register Name	Address (base+)	Description
RAMWAIT	0x0058	RAMWAIT

3.22.2.16   USB HOST Controller

Base Address = 0x4000_3000

Register Name	Address (base +)	Description
HcRevision	0x0000	Hc Revision Register
HcControl	0x0004	Hc Control Register
HcCommandStatus	0x0008	Hc Command Status Register
HcInterruptStatus	0x000C	Hc Interrupt Status Register
HcInterruptEnable	0x0010	Hc Interrupt Enable Register
HcInterruptDisable	0x0014	Hc Interrupt Disable Register
HcHCCA	0x0018	Hc Host Controller Communication Area Register
HcPeriodCurrentED	0x001C	Hc Period Current Endpoint Descriptor Register
HcControlHeadED	0x0020	Hc Control Head Endpoint Descriptor Register
HcControlCurrentED	0x0024	Hc Control Current ndpoint Descriptor Register
HcBulkHeadED	0x0028	Hc Bulk Head Endpoint Descriptor Register
HcBulkCurrentED	0x002C	Hc Bulk Current Endpoint Descriptor Register
HcDoneHead	0x0030	Hc Done Head Register
HcFmInterval	0x0034	Hc Frame Interval Register
HcFmRemaining	0x0038	Hc Frame Remaining Register
HcFmNumber	0x003C	Hc Frame Number Register
HcPeriodStart	0x0040	Hc Period Start Register
HcLSThreshold	0x0044	Hc Low Speed Threshold Register
HcRhDescriptorA	0x0048	Hc Root hub Descriptor A Register
HcRhDescripterB	0x004C	Hc Root hub Descriptor B Register
HcRhStatus	0x0050	Hc Root hub Status Register
HcRhPortStatus1	0x0054	Hc Root hub Port Status Register
HcBCR0	0x0080	Hc BCR0 Resister

4. **Electrical Characteristics**

4.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		DVCC (I/O)	$-0.3 \sim 3.9$	V
		AVCC (A/D)	$-0.3 \sim 3.9$	
		CVCC (CLK)	$-0.3 \sim 3.9$	
		REGVCC	$-0.3 \sim 3.9$	
Input voltage		V_{IN}	$-0.3 \sim V_{CC} + 0.3$	V
Low-level output current	Per pin	I_{OL}	5	mA
	Total	ΣI_{OL}	50	
High-level output current	Per pin	I_{OH}	-5	
	Total	ΣI_{OH}	-50	
Power consumption (Ta = 85°C)		PD	600	mW
Soldering temperature (10s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	$-40 \sim 125$	°C
Operating Temperature	Except during Flash W/E	T_{OPR}	$-40 \sim 85$	°C
	During Flash W/E		$0 \sim 70$	

(Note) Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no Absolute maximum rating value is exceeded with respect to current, voltage, power consumption, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning.

4.2 DC Electrical Characteristics

4.2.1 DC Electrical Characteristics (1/4)
Ta＝−40～85℃

Parameter		Symbol	Rating		Min.	Typ. (Note 1)	Max.	Unit
Supply voltage	DVCC = AVCC = CVCC =REGVCC (Note3) DVSS = AVSS = 0V	DVCC AVCC CVCC REGVCC	fosc = 12MHz fsys = 48MHz fs=30 ~ 34KHz	When operating USBHC	3.0	-	3.6	V
				When stopped USBHC	2.7		3.6	
Low-level input voltage	PJ, PK (Note2)	V$_{IL1}$	2.7V ≦ AVCC ≦ 3.6V				0.25 AVCC	V
	Normal port PA0-PA7, PB0-PB7, PP1/3/4/5	V$_{IL2}$	2.7V ≦ DVCC ≦ 3.6V		−0.3	-	0.3 DVCC	
	Schmitt-Triggered port PC0-PC7, PD0-PD7, PE0-PE7, PF0-PF4, PG0-PG7, PH0-PH7, PI0-PI1, PL0-PL7, PM0-PM7 PN0-PN4, PO0-PO7, PP0/2/6, RESETn, NMIn, MODE, SWDIO, SWCLK	V$_{IL3}$					0.25 DVCC	
	X1, XT1	V$_{IL4}$	2.7V ≦ CVCC ≦ 3.6V				0.1 CVCC	
High-level input voltage	PJ, PK (Note2)	V$_{IH1}$	2.7V ≦ AVCC ≦ 3.6V		0.75AVCC		AVCC+0.3	V
	Normal port PA0-PA7, PB0-PB7, PP1/3/4/5	V$_{IH2}$	2.7V ≦ DVCC ≦ 3.6V		0.7 DVCC	-	DVCC+0.3	
	Schmitt-Triggered port PC0-PC7, PD0-PD7, PE0-PE7, PF0-PF4, PG0-PG7, PH0-PH7, PI0-PI1, PL0-PL7, PM0-PM7 PN0-PN4, PO0-PO7, PP0/2/6, RESETn, NMIn, MODE, SWDIO, SWCLK	V$_{IH3}$			0.75 DVCC			
	X1, XT1	V$_{IH4}$	2.7V ≦ CVCC ≦ 3.6V		0.9 CVCC		CVCC+0.3	
Low-level output voltage		V$_{OL1}$	I$_{OL}$ = 2mA DVCC≧2.7V		-	-	0.4	V
Low-level output voltage PL0/1/4/5, PG0/1/4/5		V$_{OL2}$	I$_{OL}$ = 3mA DVCC≧2.7V		-	-	0.4	V
High-level output voltage		V$_{OH}$	I$_{OH}$ = −2mA DVCC≧2.7V		2.4	-	-	V
Input leakage current		I$_{LI}$	0.0 ≦ V$_{IN}$ ≦DVCC 0.0 ≦ V$_{IN}$ ≦AVCC		-	0.02	± 5	μA
Output leakage current		I$_{LO}$	0.2 ≦ V$_{IN}$ ≦ DVCC − 0.2 0.2 ≦ V$_{IN}$ ≦ AVCC − 0.2		-	0.05	± 10	
BACKUP STOP Mode Voltage		V$_{BSTOP}$	DVCC = AVCC = CVCC =REGVCC (Note 3)		2.7	-	3.6	V
Pull-up resister at Reset		RRST	DVCC = 2.7V～3.6V		30	50	150	kΩ
Schmitt-Triggered port		VTH	2.7V ≦ DVCC ≦ 3.6V		0.3	0.6	-	V
Programmable pull-up/ pull-down resistor		PKH	DVCC = 2.7V～3.6V		30	50	150	kΩ
Pin capacitance (Except power supply pins)		C$_{IO}$	fc = 1.5MHz		-	-	10	pF

(Note 1) Ta＝25℃, DVCC = REGVCC = AVCC = CVCC = 3.3V, unless otherwise noted.
(Note 2) In this case, PJ and PK used as general input port.
(Note 3) The same voltage must be supplied to DVCC, AVCC, CVCC and REGVCC.

4.2.2 DC Electrical Characteristics (2/4)

Ta＝－40～85℃

Parameter	Symbol	Raing	Min.	Typ. (Note1)	Max.	Unit
Low level output current (DVCC source)	IOL All ports	2.7V ≦ DVCC ≦ 3.6V Normal port (per pin)	－	－	2	mA
	IOL PL/PG	2.7V ≦ DVCC ≦ 3.6V PL0/1/4/5,PG0/1/4/5 (per pin)	－	－	3	
	∑IOL PL,PI	Total (PORT L/ I)	－	－	20	
	∑IOL PM,PN,PO ,PP	Total (PORT M / N / O /P)	－	－	27	
	∑IOL PA,PB,PC, PD,PE	Total (PORT A/B/C/D/E)	－	－	27	
	∑IOL PF,PG,PH	Total (PORT F/G/H)	－	－	27	
	∑IOL	Total (all ports)	－	－	35	

（Note1）Ta=25℃, DVCC = REGVCC = AVCC = CVCC =3.3V, unless otherwise noted.
（Note2）The same voltage must be supplied to DVCC, AVCC, CVCC and REGVCC.

4.2.3 DC Electrical Characteristics (3/4)

Ta＝－40～85℃

Parameter	Symbol	Raing	Min.	Typ. (Note1)	Max.	Unit
High level output current (DVCCsource)	IOH All ports	2.7V ≦ DVCC ≦ 3.6V Normal port (per pin)	－	－	-2	mA
	∑IOH PI/PL/PM PN/PO/PP	Total (PORT I/L/ M/ N/ O/ P)	－	－	-13	
	∑IOH PA/PB/PC/ PD/PE	Total (PORT A/ B/ C/ D/ E)	－	－	-13	
	∑IOH PF,PG,PH	Total (PORT F/G/H)	－	－	-13	
	∑IOH	Total（All ports）	－	－	-35	

（Note1）Ta=25℃, DVCC = REGVCC = AVCC = CVCC =3.3V, unless otherwise noted.
（Note2）The same voltage must be supplied to DVCC, AVCC, CVCC and REGVCC.
(Note3) High-level output current (∑IOH) is each voltage source total value

4.2.4   DC Electrical Characteristics (4/4)

DVCC = AVCC = = CVCC = REGVCC = 2.7V～3.6V, Ta＝-40～85°C

Parameter	Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
NORMAL (Note 2) Gear 1/1	I_{CC}	f_{sys}= 48MHz (fosc= 12MHz)	-	85	95	mA
IDLE 2 (Note 3)			-	40	48	
IDLE1 (Note 4)		f_{sys}= 1.5 MHz (fosc=12MHz,PLL=OFF, CG=1/8)		2	5	
SLOW (Note 5)		fs = 32.768kHz	-	1	6	
SLEEP (Note 6)			-	260	2450	µA
STOP		-	-	250	2400	
BACKUP SLEEP (Note 7)		fs = 32.768kHz		35	210	
BACKUP STOP (Note 8)		DVCC = AVCC = REGVCC = 2.7V		25	200	

(Note 1) Ta=25°C, DVCC = AVCC = CVCC = REGVCC = 3.3V, unless otherwise noted.

(Note 2) ICC NORMAL measuring conditions:

      Measured with the dhrystone ver. 2.1 operated in FLASH.

      All functions operates except A/D converter.

(Note 3) ICC IDLE2 measuring conditions:

      CPU is stopped and all periferals operates.

(Note 4) ICC IDLE1 measuring conditions:

      CPU is stopped and some periferals operates.

(Note 5) ICC SLOW measuring conditions:

      CPU, Timer, RMC and RTC operate.

(Note 6) ICC SLEEP measuring conditions:

      RMC and RTC operate.

(Note 7) ICC BACKUPSLEEP measuring conditions:

      RMC and RTC operate, keeping the BACKUP RAM, others are power off.

(Note 8) ICC BACKUPSTOP measuring conditions:

      Keeping the BACKUP RAM, others are power off.

4.2.5   10-bit ADC Electrical Characteristics

DVCC = AVCC = CVCC = REGVCC = VREFH = 2.7V~3.6V,
AVSS = DVSS, Ta＝－40~85℃
AVCC load capacitance ⩾3.3μF, VREFH load capacitance ⩾3.3μF

Parameter		Symbol	Rating	Min	Typ.	Max	Unit
Analog reference voltage (+)		VREFH	-	2.7	3.3	3.6	V
Analog input voltage		VAIN	-	AVSS	-	VREFH	V
Analog supply current	A/D conversion	IREF	DVSS = AVSS	-	2.5	5.5	mA
	Non-A/D conversion			-	± 0.02	± 5	μA
Supply current	A/D conversion	-	Non-IREF	-	-	3	mA
INL error		-	AIN resistance ≤600Ω AIN load capacitance ≤30pF Conversion time ⩾1.15μs	-	± 2	± 3	LSB
DNL error				-	± 1	± 2	
Offset error				-	± 2	± 4	
Full-scale error				-	± 2	± 4	
INL error		-	AIN resistance ≤1.3kΩ AIN load capacitance ⩾0.1μF Conversion time ⩾1.15μs	-	± 2	± 3	
DNL error				-	± 1	± 2	
Offset error				-	± 2	± 4	
Full-scale error				-	± 2	± 4	
INL error		-	AIN resistance ≤10kΩ AIN load capacitance ⩾0.1μF Conversion time ⩾2.30μs	-	± 2	± 3	
DNL error				-	± 1	± 2	
Offset error				-	± 2	± 4	
Full-scale error				-	± 2	± 4	

(Note) 1LSB = (VREFH - AVSS) / 1024[V]

(Note) This specification is operationg A/D converter solely.

4.3   AC Electrical Characteristics

All the AC characteristics shown in this section are measured under the following conditions, unless otherwise noted.

AC measuring conditions:

- f_{SYS} in the heading row denotes the system clock frequency and the symbol T denotes the period of half system clock.

- Output level : High $= 0.7 \times$ DVCC, Low $= 0.3 \times$ DVCC

- Input level: High $= 0.7 \times$ DVCC, Low $= 0.3 \times$ DVCC

- Load capacitace CL = 40pF

(Note)The "equations" in the table show the specifications in the range of DVCC = 2.7 to 3.6 V.

4.3.1   Basic Bus cycle

No.	Parameter	Symbol	Equation		$f_{sys} =48\,MHz$ N = 4 M = 1 K = 5 L = 2 P = 2 Q = 2	Unit
			Min	Max		
Read cycle						
1	SMCCLK	t_{CYC}	31.25	2000	31.3	
2	A1 ~ A23 Valid → D0 ~ D15. input	t_{ADH}		(N)T − 35.0	131.7	
3	A1 ~ A23 Valid → D0 ~ D15. input (Page mode）	t_{AD1}		(Q)T − 35.0	48.3	
4	OEn folling edge → D0 ~ D15. input	t_{OED}		(N-M)T − 25.0	100	
5	OEn low-level puls width	t_{OEW}	(N-M)T − 13.0		80.8	ns
6	A1 ~ A23 Valid → OEn falling edge	t_{AOEH}	MT − 15.0		26.7	
7	OEn raising edge→ D0 ~ D15.Hold	t_{HR}	0		0.0	
8	A1 ~ A23 Valid → D0 ~ D15.Hold	t_{HA}	0		0.0	
9	OEn High-level pulse width	t_{OEHW}	MT − 13.0		28.7	
Write cycle						
10	D0 ~ D15.Valid → WEn Rasing Edge	t_{DW}	(L+1)T − 23.0		102	
11	WEn Low Level puls Width	t_{WW}	LT − 13.0		70.3	
12	A1 ~ A23 Vaild → WEn falling Edge	t_{AW}	T − 15.0		26.7	ns
13	WEn Rasing edge → A1 ~ A23 Hold	t_{WA}	(K-L)T − 13.0		112	
14	WEn Rasing edge → D0 ~ D15. Hold	t_{WD}	(K-L-1)T − 10.0		73.3	

- Equation Measurment condition:

  N = t_{RC} Cycle $\geqq$ 3   M = t_{CEOE} Cycle $\geqq$ 1

  K = t_{WC} Cycle $\geqq$ 3   L = t_{WP} Cycle $\geqq$ 1

  P = t_{TR} Cycle $\geqq$ 1   Q = t_{PC} Cycle $\geqq$ 1

(1) Memory Read cycle timing (Separate Bus) (Example when set to t_{RC}=4, t_{CEOE}=1)
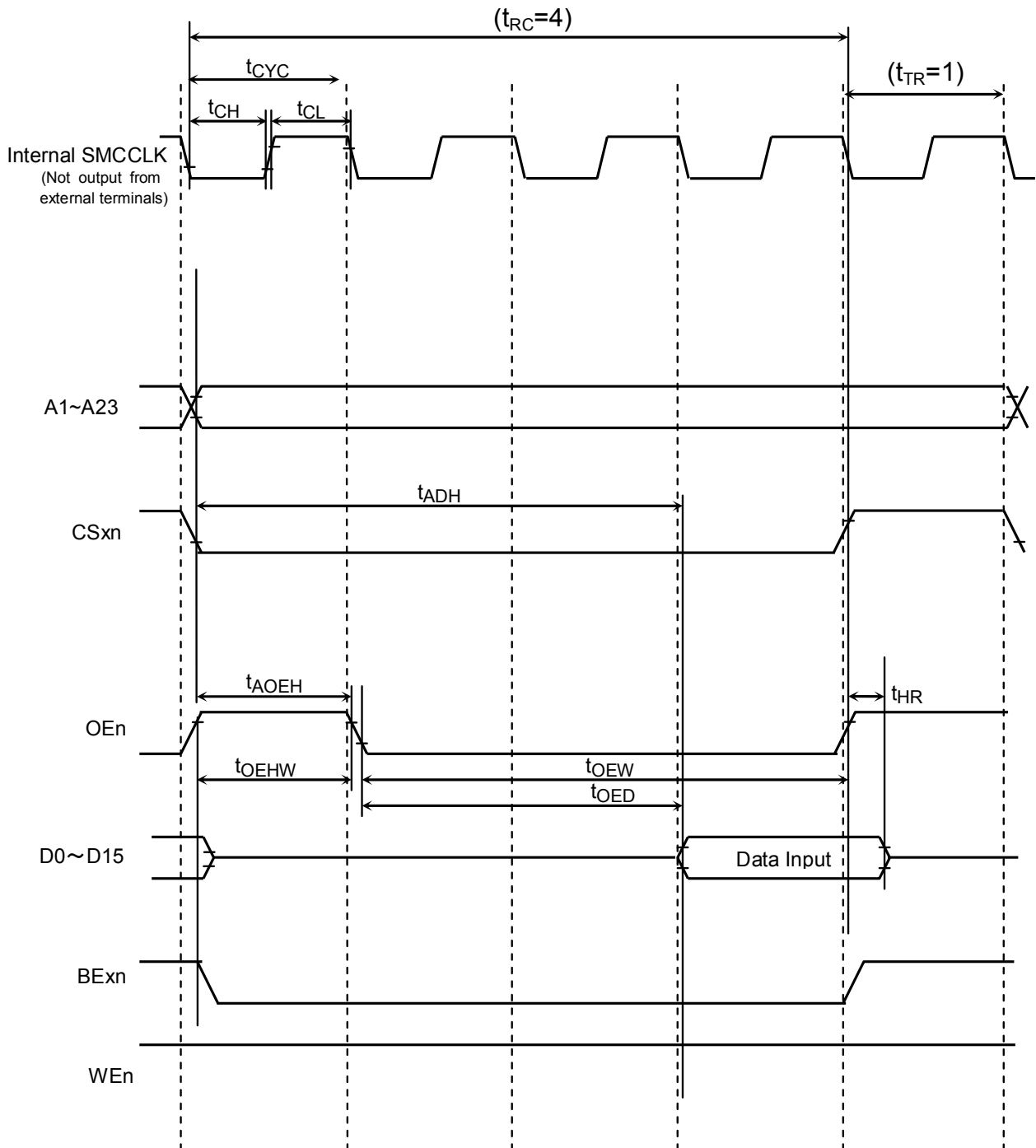
(2) Memory Write cycle timing (Separate Bus) (Example when set to t_{WC}=4, t_{WP}=2)
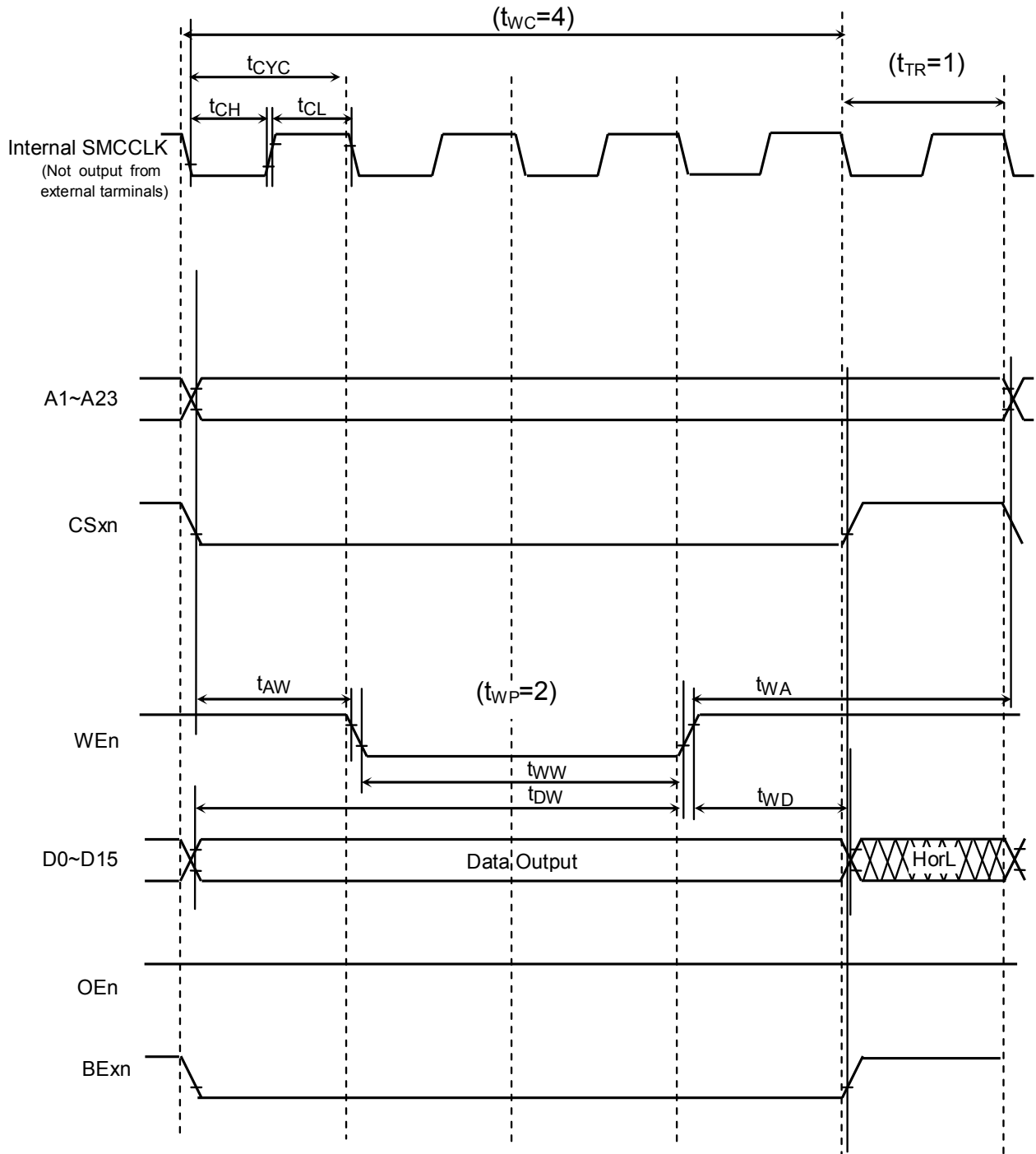
(3) Memory Read cycle timing (Separate Bus) Page access

(Example when set to t_{RC}=4, t_{CEOE}=1, t_{PC}=2 4burst)
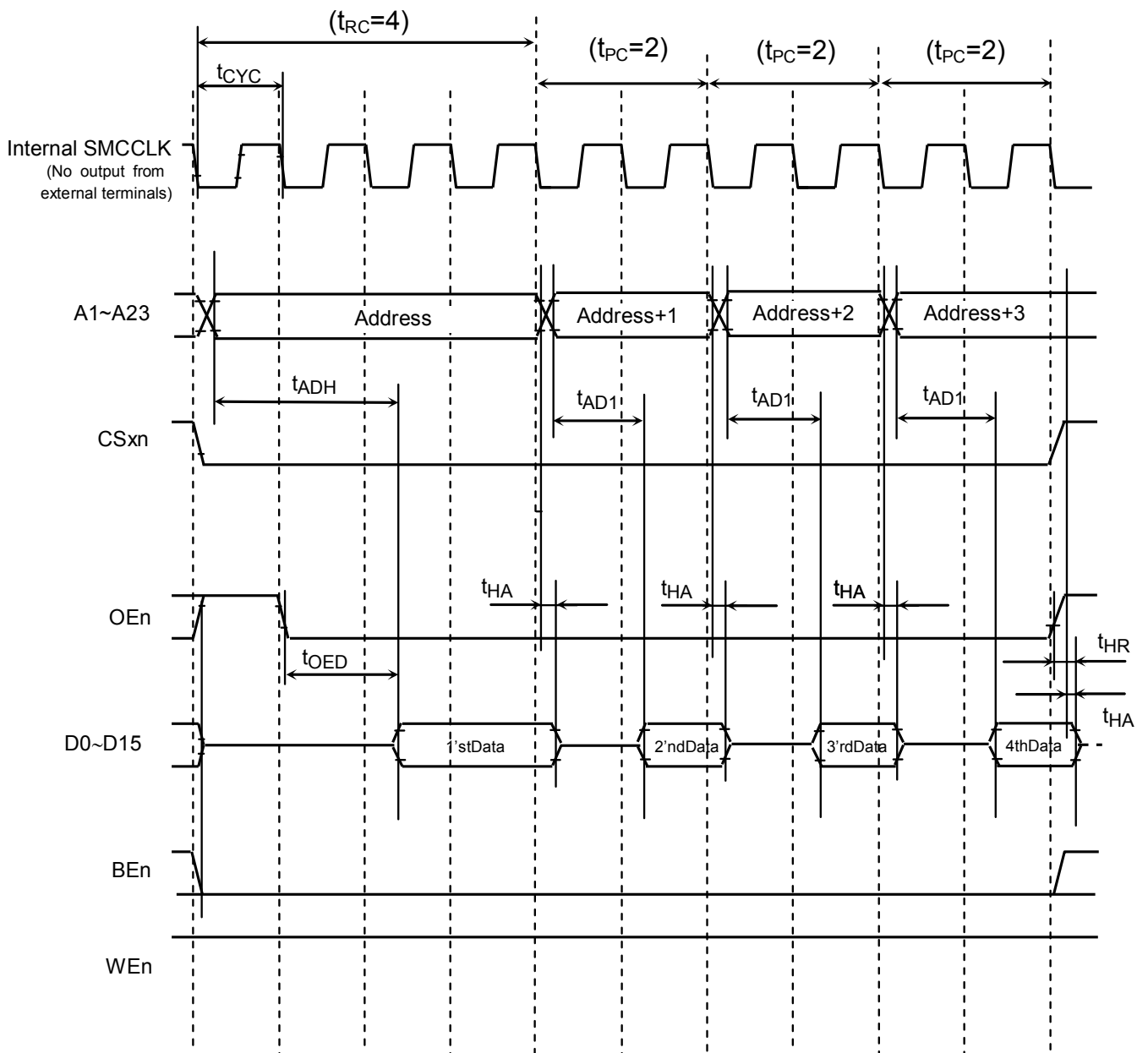
4.3.2 Serial Channel Timing(SIO)

4.3.2.1 I/O Interface mode

In the table below, the letter x represents the SIO operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

AC measurement condition
/Output levels: High 0.8DVCC / Low 0.2DVCC , CL=30 pF
/Input levels: Refer to low-level input voltage and high-level input voltage in 4.2 DC Electrical Characteristics.

1) SCLK input mode（SIO0～SIO11）

Parameter	Symbol	Equation		fsys = 48MHz		Unit
		Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	$4x$		83.3		ns
SCLK Clock Low width (input)	t_{SCL}	$4x$		83.3		
SCLK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$		166.7		
TxD to SCLK rise or fall (Note 1)	t_{OSS}	$t_{SCY}/2 - 3x - 45$		−24.2 (Note2)		
TxD hold or fall after SCLK rising (Note 1)	t_{OHS}	$t_{SCY}/2$		83.3		
RxD valid to SCLK rise or fall (Note 1)	t_{SRD}	30		30		
RxD hold or fall after SCLK rising (Note 1)	t_{HSR}	$x + 30$		50.8		

(Note 1) SCLK rise or fall: Measured relative to the programmed active edge of SCLK.

(Note 2) Keep this value positive by adjusting SCLK cycle.

2）SCLK output mode（SIO0～SIO11）

Parameter	Symbol	Equation		fsys = 48MHz		Unit
		Min	Max	Min	Max	
SCLK cycle (programmable)	t_{SCY}	$4x$		83.3		ns
TxD to SCLK rise	t_{OSS}	$t_{SCY}/2 - 20$		21.7		
TxD hold after SCLK rising	t_{OHS}	$t_{SCY}/2 - 20$		21.7		
RxD valid to SCLK rise	t_{SRD}	45		45		
RxD hold after SCLK rising	t_{HSR}	0		0		
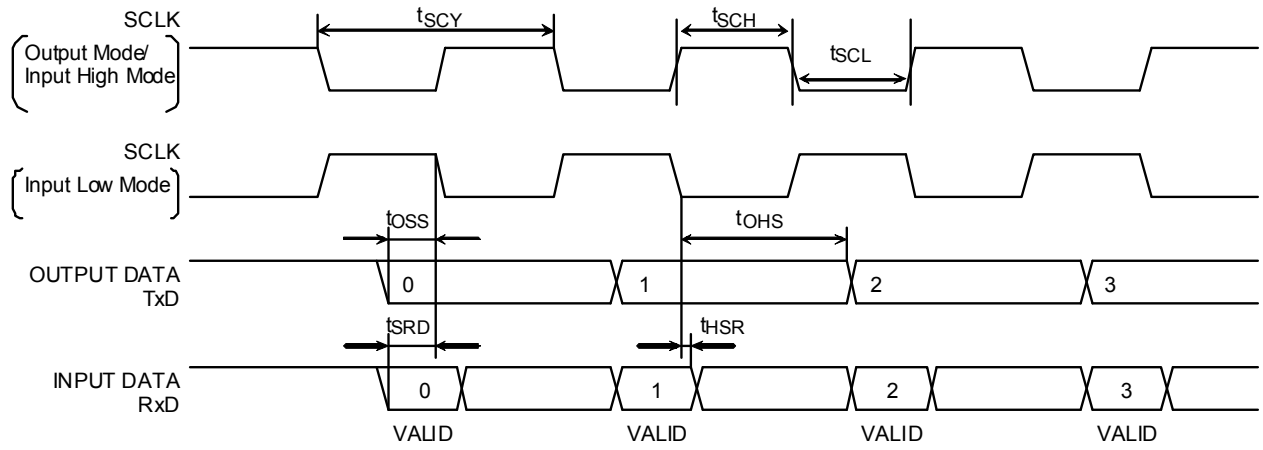
4.3.3   Serial Bus Inter face(SBI/SIO)

4.3.3.1   I2C mode

In the table below, the letter x denotes the fsys cycle, the letter T denotes ϕT1.

n denotes the value of n programmed into the SCK (SCL output frequency select) field in the SBInCR.

AC measurement condition

Output levels: High 0.8DVCC / Low 0.2DVCC , CL=30 pF

Input levels: Refer to low-level input voltage and high-level input voltage in 4.2 DC Electrical Characteristics.

Parameter	Symbol	Equation		Standard Mode		Fast Mode		Unit
		Min	Max	Min	Max	Min	Max	
SCL clock frequency	t_{SCL}	0		0	100	0	400	kHz
Hold time for START condition	$t_{HD:STA}$			4.0		0.6		
SCL low width (Input) (Note 1)	t_{LOW}			4.7		1.3		
SCL high width (Input) (Note 2)	t_{HIGH}			4.0		0.6		μs
Setup time for a repeated START condition	$t_{SU;STA}$	(Note 5)		4.7		0.6		
Data hold time (Input) (Note 3, 4)	$t_{HD;DAT}$			0.0		0.0		
Data setup time	$t_{SU;DAT}$			250		100		ns
Setup time for a stop condition	$t_{SU;STO}$			4.0		0.6		μs
Bus free time between stop condition and start condition	t_{BUF}	(Note 5)		4.7		1.3		

(Note 1) SCL clock low width (output) is calculated with: $(2^{n-1}+58)/x$

(Note 2) SCL clock high width (output) is calculated with: $(2^{n-1}+12)/x$

> Notice: On I2C-bus specification, Maximum Speed of Standard Mode is 100KHz, Fast mode is 400Khz. Internal SCL Frequency setting should comply with Note1 & Note2 shown above.

(Note 3) The output data hold time is equal to 12x of internal SCL.

(Note 4) The Philips I2C-bus specification states that a device must internally provide a hold time of at least 300 ns for the SDA signal to bridge the undefined region of the falling edge of SCL. However, this SBI does not satisfy this requirement. Also, the output buffer for SCL does not incorporate slope control of the falling edges; therefore, the equipment manufacturer should design so that the input data hold time shown in the table is satisfied, including tr/tf of the SCL and SDA lines.

(Note 5) Software-dependent.

(Note 6) The Philips I2C-bus specification instructs that if the power supply to a Fast-mode device is switched off, the SDA and SCL I/O pins must be floating so that they don't obstruct the bus lines. However, this SBI does not satisfy this requirement.
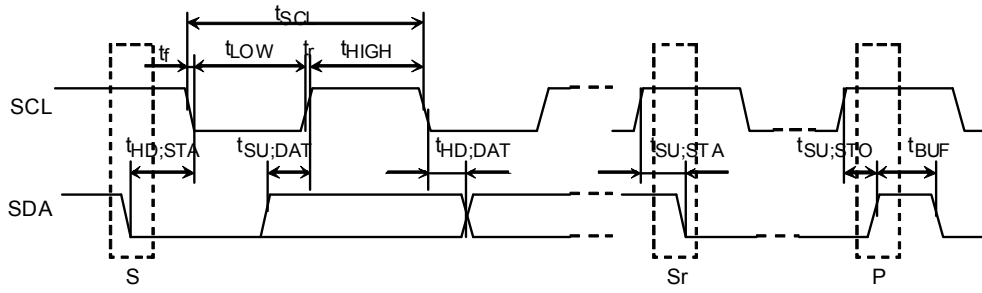
S: Start condition
Sr: Repeated start condition
P: Stop condition

4.3.3.2　Clock-Synchronous 8-Bit SIO mode

In the table below, the letter x denotes the fsys cycle, the letter T denotes ϕT1.

n denotes the value of n programmed into the SCK (SCL output frequency select) field in the SBInCR.

The electrical specifications below are for an SCK signal with a 50% duty cycle.

AC measurement condition

　Output levels: High 0.8DVCC / Low 0.2DVCC , CL=30 pF

　Input levels: Refer to low-level input voltage and high-level input voltage in 4.2 DC Electrical Characteristics.

1) SCK Input Mode

Parameter	Symbol	Equation		fsys = 48MHz		Unit
		Min	Max	Min	Max	
SCK Clock High width (input)	t_{SCH}	4x		83.3		ns
SCK Clock Low width (input)	t_{SCL}	4x		83.3		
SCK cycle	t_{SCY}	8x		166.7		
TxD to SCK rise	t_{OSS}	$t_{SCY}/2 - 3x - 45$		−24.2 (Note)		
TxD hold after SCK rising	t_{OHS}	$t_{SCY}/2 + x$		104.2		
RxD valid to SCK rise	t_{SRD}	$30 - x$		9.2		
RxD hold after SCK rising	t_{HSR}	30		30		

(Note) Keep this value positive by adjusting SCK cycle.

2) SCK Output Mode

Parameter	Symbol	Equation		fsys = 48MHz		Unit
		Min	Max	Min	Max	
SCK cycle (programmable)	t_{SCY}	16x		333.3		ns
TxD to SCK rise	t_{OSS}	$t_{SCY}/2 - 20$		146.7		
TxD hold after SCK rising	t_{OHS}	$t_{SCY}/2 - 20$		146.7		
RxD valid to SCK rise	t_{SRD}	45		45		
RxD data hold after SCK rising	t_{HSR}	0		0		
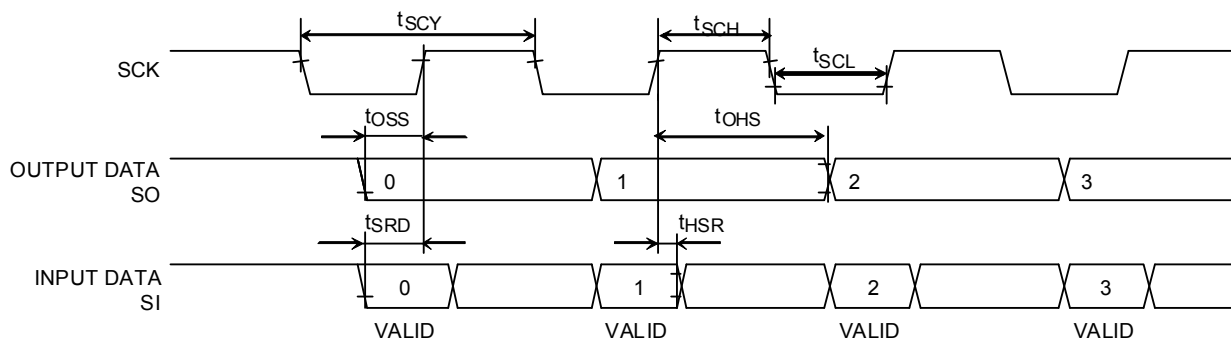
4.3.4 SSP controller

AC measuring conditions:

- In the table below, the letter T denotes the period of internal priscalar clock $f_{SPCLK.}$

- Output level : High = 0.7 × DVCC, Low = 0.3 × DVCC
- Input level: High = 0.9 × DVCC, Low = 0.1 × DVCC

- Load capacitace CL = 30pF

(Note)The "equations" in the table show the specifications in the range of DVCC = 2.7 to 3.6 V.

Parameter	Symbol	Equation Min	Max	fsys 48MHz (m=4 n=12)	Unit
SPCLK period (Master)	T_m	(m)T where 50 nS or more		83.3 (12MHz)	
SPCLK period (Slave)	T_s	(n)T		250.0 (4MHz)	
SPCLK rising time	t_r		10.0	10.0	
SPCLK falling time	t_f		10.0	10.0	
SPCLK low-level pulse width for master mode	t_{WLM}	(m)T / 2 - 10.0		31.7	
SPCLK high-level pulse width for master mode	t_{WHM}	(m)T / 2 −10.0		31.7	
SPCLK low-level pulse width for slave mode	t_{WLS}	(n)T / 2 −10.0		115.0	
SPCLK high-level pulse width for slave mode	t_{WHS}	(n)T / 2 −10.0		115.0	
Master mode: SPCLK rising/falling -> Output data enable	t_{ODSM}		15.0	15.0	
Master mode: SPCLK rising/falling -> Output data hold	t_{ODHM}	(m)T/2 -10		31.7	
Master mode: SPCLK rising/falling -> Input data enable, Delay time	t_{IDSM}	15.0		15.0	nS
Master mode: SPCLK rising/falling -> Input data hold	t_{IDHM}	5.0		5.0	
Master mode: SPFSS enable -> SPCLK rising/falling	t_{OFSM}	(m)T -10	(m)T+10	31.7 – 51.7	
Slave mode: SPCLK rising/falling (Output data enable, Delay time	t_{ODSS}		(3T) + 22	84.5	
Slave mode: SPCLK rising/falling -> Output data hold	t_{ODHS}	(n)T /2 + (2T)		166.7	
Slave mode: SPCLK rising/falling -> Input data enable, Delay time	t_{IDSS}	0		0	
Slave mode: SPCLK rising/falling -> Input data hold	t_{IDHS}	(3T) +10		72.5	
Slave mode: SPFSS enable -> SPCLK rising/falling	t_{OFSS}	(n)T - 15		235.0	

Note 1) The communication baud rate clock needs to be set in the following condition ranges:

Master mode:
$m = (<CPSDVSR> \times (1+<SCR>)) = f_{sys} / f_{SPCLK}$

Only an even number can be set in <CPSDVSR>. The range of "m" is as follows: 65024 ≥ m ≥ 4.

Slave mode:
$n = f_{sys} / f_{SPCLK}$ (65024 ≥ n ≥ 12)

SSP SPI mode (Master)

$f_{sys} \geq 4 \times f_{SPCLK}$ (maximum)

$f_{sys} \geq 65024 \times f_{SPCLK}$ (minimum)

(1) Master SSPCR0<SPH>= "0" (Data is latched at the 1st edge.)
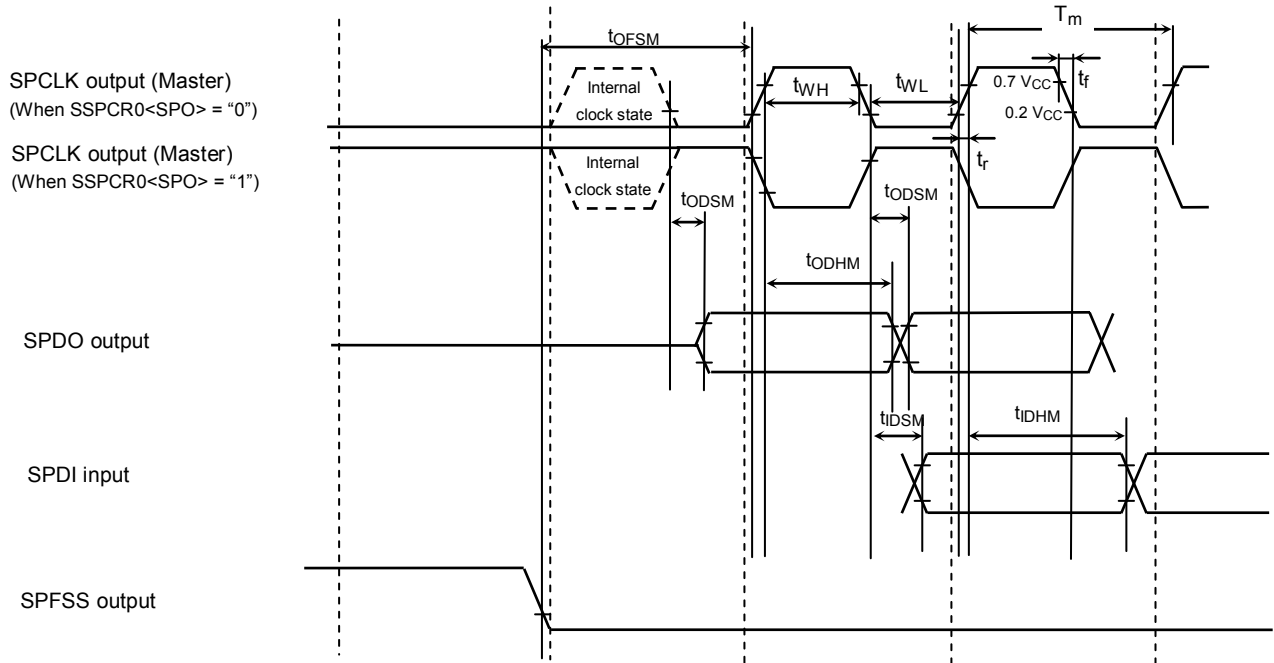
SSP SPI mode (Master)

(2) Master SSPCR0<SPH>= "1" (Data is latched at the 2nd edge.)
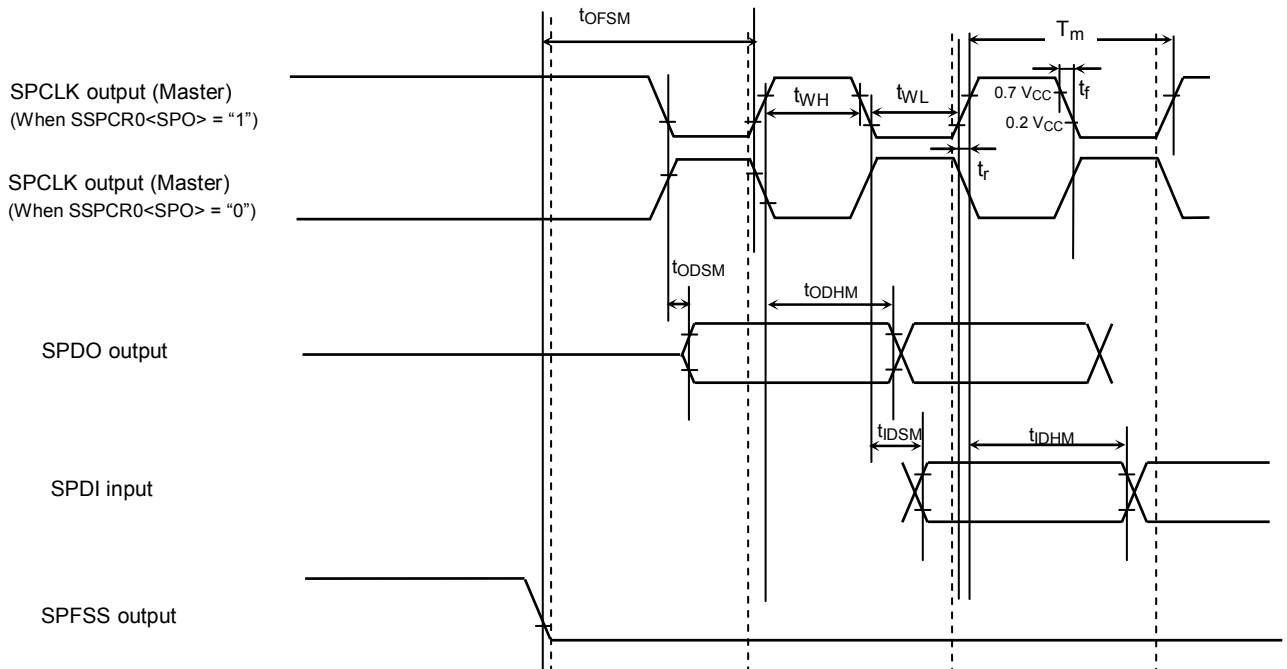
SSP SPI mode (Slave)
  $f_{sys} \geq 12 \times f_{SPCLK}$ (maximum)
  $f_{sys} \geq 65024 \times f_{SPCLK}$ (minimum)

(3) Slave SSPCR0<SPH>= "0" (Data is latched at the 1st edge.)
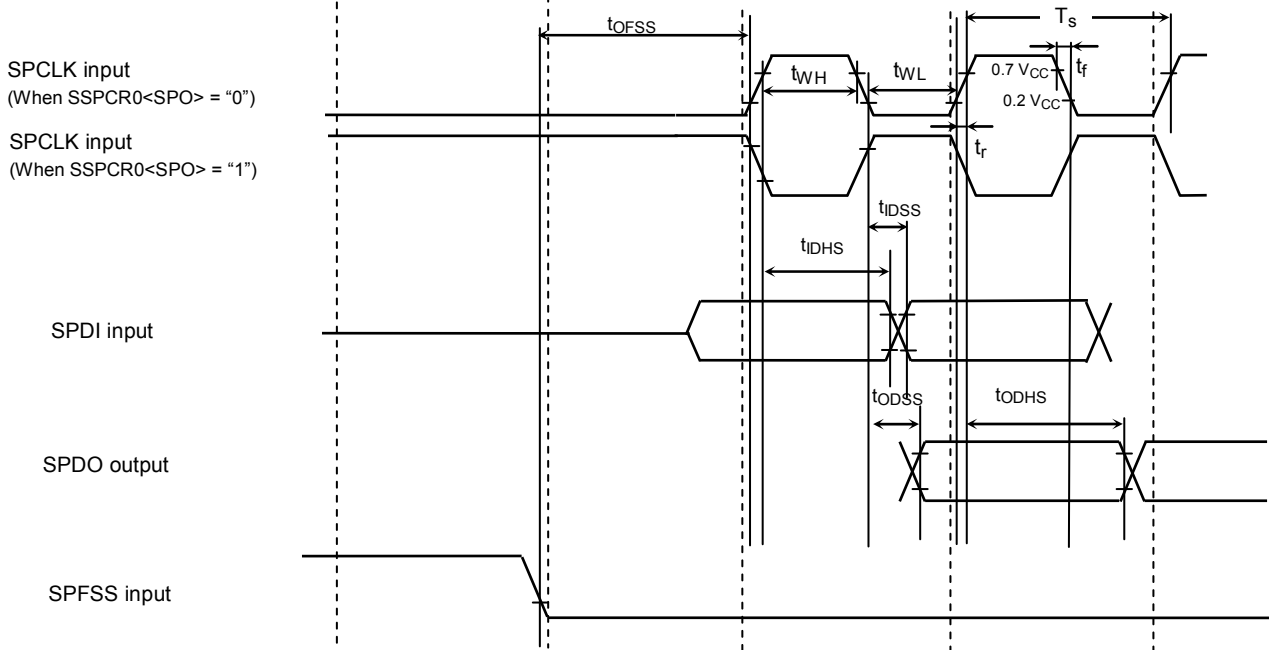
SSP SPI mode (Slave)
(4) Slave SSPCR0<SPH>= "1" (Data is latched at the 2nd edge.)
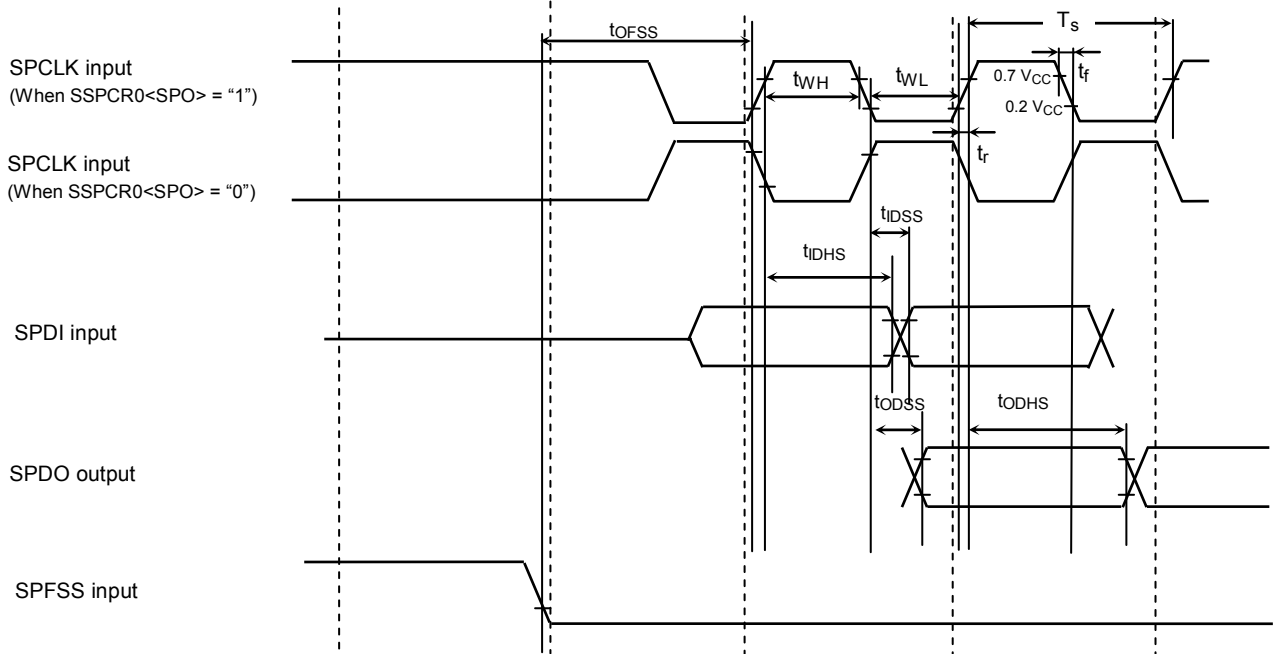
4.3.5 Event Counter (TMRB)

In the table below, the letter x represents the TMRB operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		48MHz		Unit
		Min	Max	Min	Max	
Clock low pulse width	t_{VCKL}	2x + 100		141.7		ns
Clock high pulse width	t_{VCKH}	2x + 100		141.7		ns

AC measurement condition
  /Output levels: High 0.8DVCC / Low 0.2DVCC, CL=30 pF
  /Input levels: Refer to low-level input voltage and high-level input voltage in 4.2 DC Electrical Characteristics.

4.3.6 Capture (TMRB)

In the table below, the letter x represents the TMRB operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		48MHz		Unit
		Min	Max	Min	Max	
Low pulse width	t_{CPL}	2x + 100	–	141.7	–	ns
High pulse width	t_{CPH}	2x + 100	–	141.7	–	ns

AC measurement condition
  /Output levels: High 0.8DVCC / Low 0.2DVCC , CL=30 pF
  /Input levels: Refer to low-level input voltage and high-level input voltage in 4.2 DC Electrical Characteristics.

4.3.7 General Interrupts (except STOP Release Interrupt)

In the table below, the letter x represents the fsys cycle time.

Parameter	Symbol	Equation		48MHz		Unit
		Min	Max	Min	Max	
Low pulse width for INT0~7	t_{CPL}	x + 100	-	120.8	-	ns
High pulse width for INT0~7	t_{CPH}	x + 100	-	120.8	-	ns

AC measurement condition
  /Output levels: High 0.8DVCC / Low 0.2DVCC , CL=30 pF
  /Input levels: Refer to low-level input voltage and high-level input voltage in 4.2 DC Electrical Characteristics.

4.3.8 Interrupts (STOP and NMIn Release Interrupts)

Parameter	Symbol	Min	Max	Unit
Low pulse width for INT0~D and NMIn	t_{INTBL}	100	-	ns
High pulse width for INT0~D	t_{INTBH}	100	-	ns

AC measurement condition

/Output levels: High 0.8DVCC / Low 0.2DVCC , CL=30 pF

/Input levels: Refer to low-level input voltage and high-level input voltage in 4.2 DC Electrical Characteristics.

4.3.9 SCOUT Pin AC Characteristic

Parameter	Symbol	Equation		48MHz		Unit
		Min	Max	Min	Max	
High pulse width	t_{SCH}	0.5T − 5		5.4		ns
Low pulse width	t_{SCL}	0.5T − 5		5.4		ns

AC measurement condition

/Output levels: High 0.8DVCC / Low 0.2DVCC , CL=30 pF

/Input levels: Refer to low-level input voltage and high-level input voltage in 4.2 DC Electrical Characteristics.

(Note)In the above table, the letter T represents the cycle time of the SCOUT output clock.
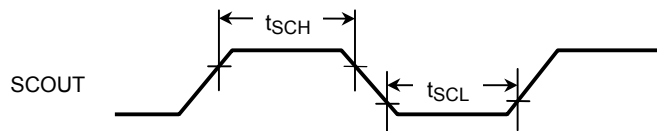
4.3.10 USB Host Controller

DVCC = AVCC = REGVCC = 3.0V～3.6V / f_{USB} = 48 MHz

Parameter	Symbol	Min.	Max.	Unit	Note
D+, D− Rising Time	t_R	4	20	ns	
D+, D− Rising Time	t_F	4	20		
Differential Common mode Voltage range	VDI	0.2		V	∣(D+) −(D-)∣
Output Signal Cross Voltage	V_{CRS}	1.3	2.0	V	DVCC=3.0V~3.45V
			2.0(Note)	V	DVCC=3.0V~3.6V

(Note)Average value

AC measuring conditions:

<Full Speed>

TMPM322F10FG
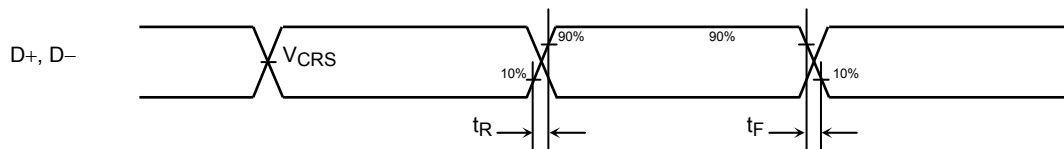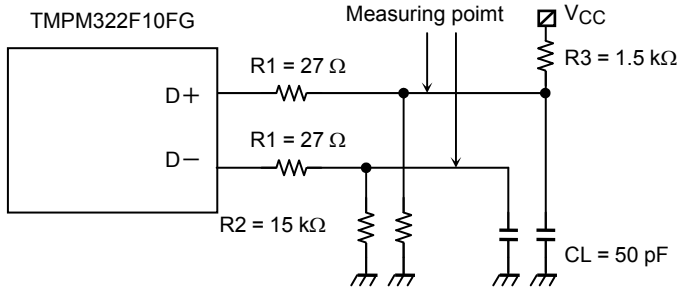
4.4 Flash Characteristics

Parameter	Rating	Min	Typ.	Max	Unit
Flash memory erase / write times	DVCC=AVCC= CVCC=REGVCC =2.7V~3.6V Ta=0～70℃	-	-	100	times

4.5   Recommened Oscillation Circuit
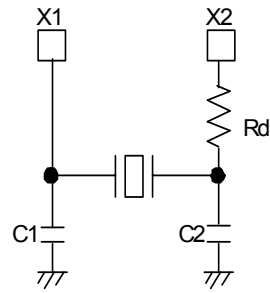
Fig 4.5.1 High-frequency oscillation connection
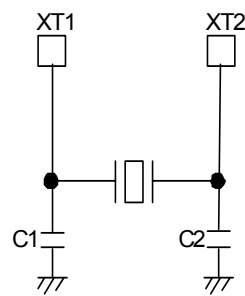
Fig 4.5.2 Low-frequency oscillation connection

Note : The load value of the oscillator is the sum of loads (C1 and C2) and the floating load of the
actual assembled board. There is a possibility of operating error when using C1 and C2
values in the table below. When designing the board, design the minimum length pattern
around the oscillator evaluation be carried out using the actual board.

4.5.1   Crystal oscillator

The TX03 recommends the high-frequency oscillator by KYOCERA KINSEKI Corporation.
Please refer to the following URL for details.

http://www.kinseki.co.jp

4.6   Handling Precaution

4.6.1   Solderability of lead free products

Test parameter	Test condition	Note
Solderability	Use of Sn-37Pb solder Bath Solder bath temperature = 230°C, Dipping time = 5 seconds, The number of times = one, Use of R-type flux	Pass : Solderability rate until forming ≥ 95%
	Use of Sn-3.0Ag-0.5Cu solder bath Solder bath temperature = 245°C, Dipping time = 5 seconds, The number of times = one, Use of R-type flux	

5. **Port Section Equivalent Circuit Schematic**

- Reading the schematics

    Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

    The input protection resistance ranges from several tens of ohms to several hundreds of ohms. Damping resistors X2 and XT2 are shown with a typical value.

■ PA0-PA7, PB0-PB7, PP1, PP3-PP5
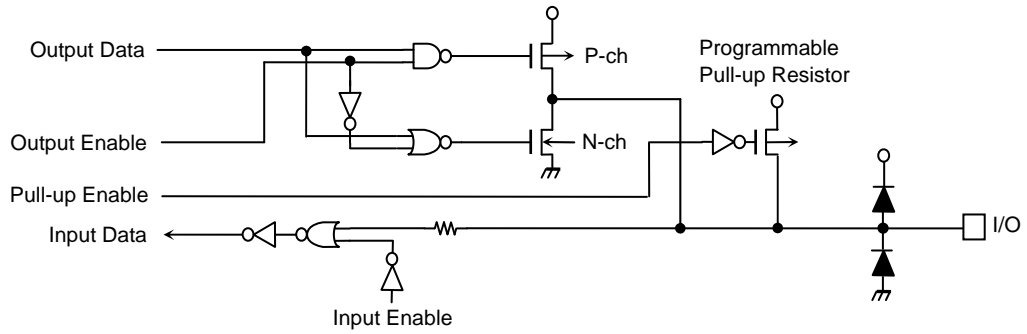
■ PC0-PC7, PD0-PD7, PE0-PE7, PF0-PF4, PG0-PG7, PH0-PH7, PI0, PL0-PL7, PM0-PM7,
   PN0-PN7, PO0-PO7, PP0, PP2, PP6
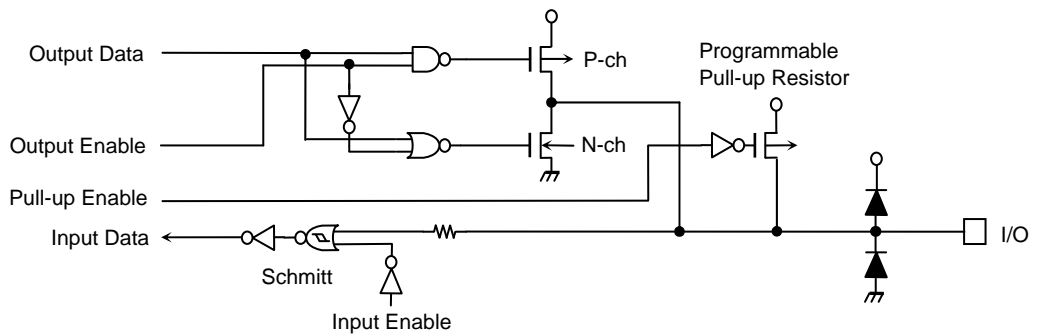
■ PI1
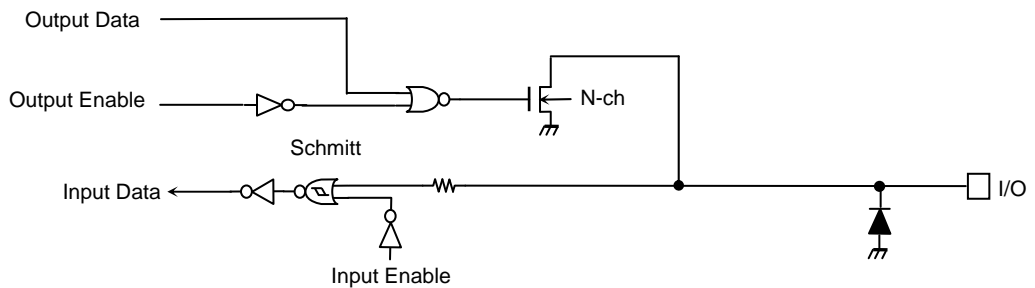
■ PJ0-PJ7, PK0-PK7
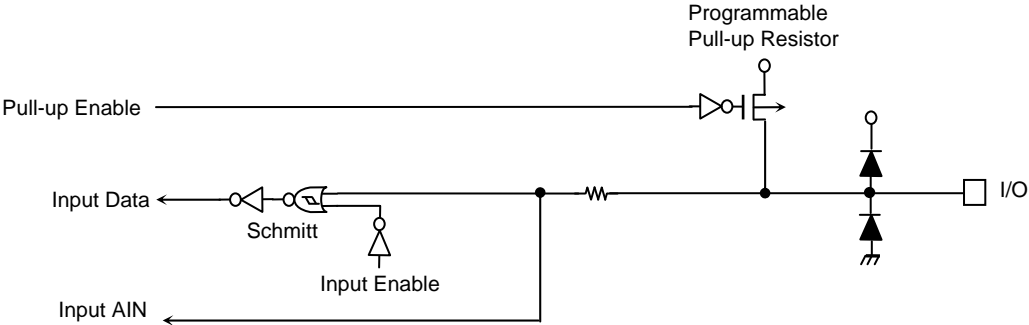
■ RESETn, NMIn, SWDIO
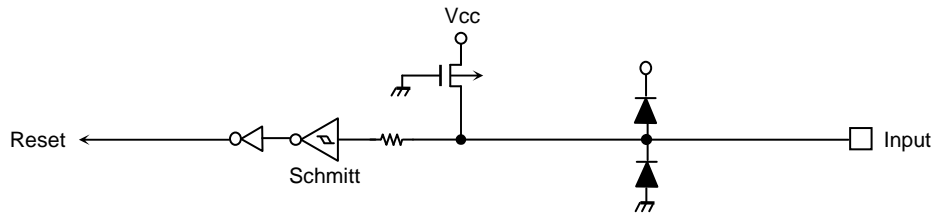
■ MODE, SWCLK
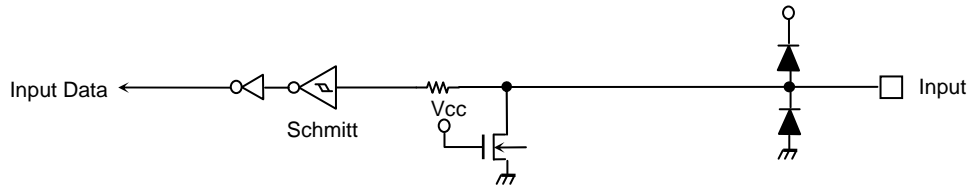
■ D+, D -
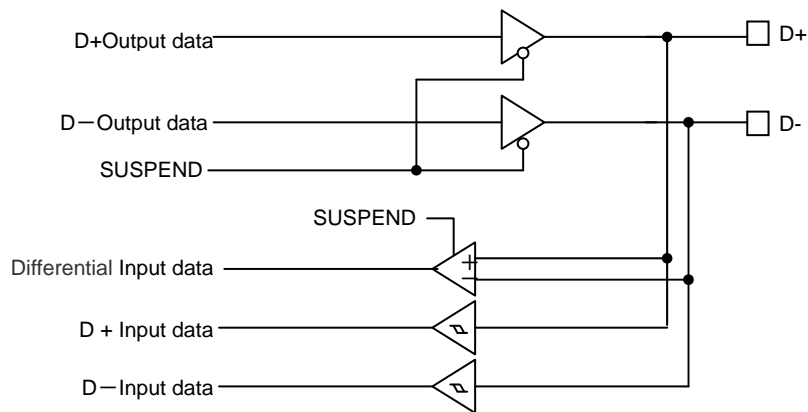
■ X1, X2
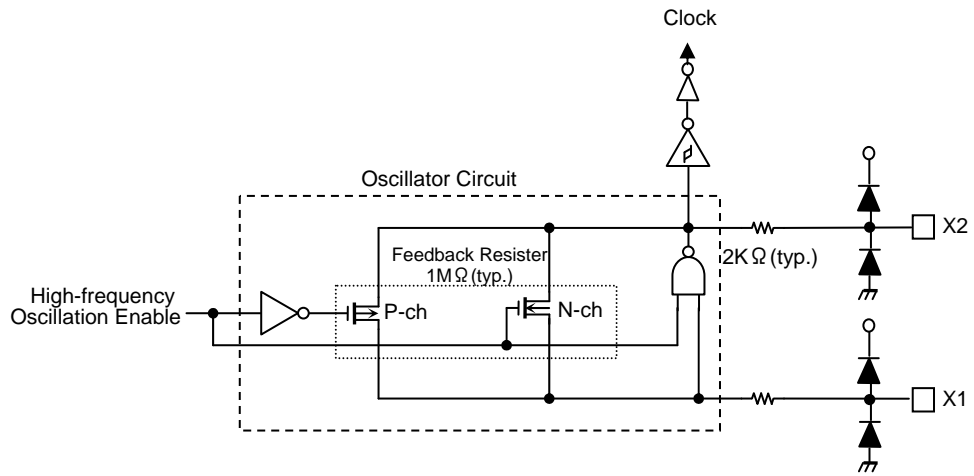
■ XT1, XT2
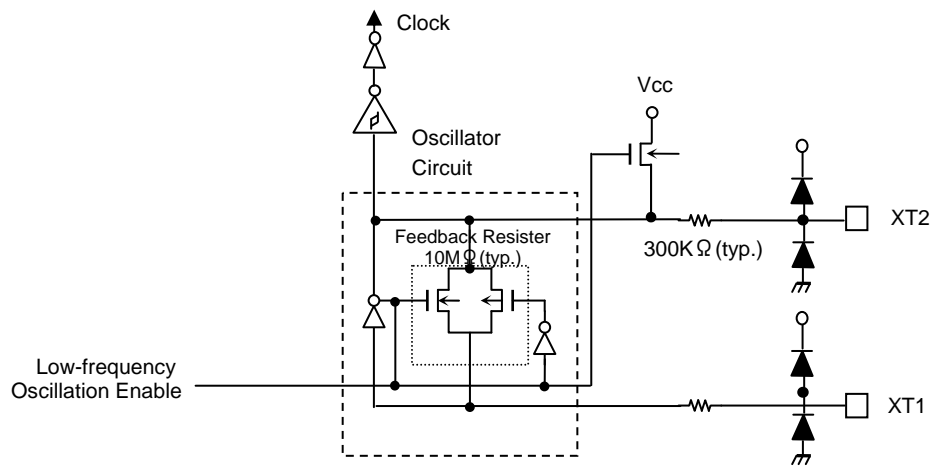
■   VREFH, AVSS
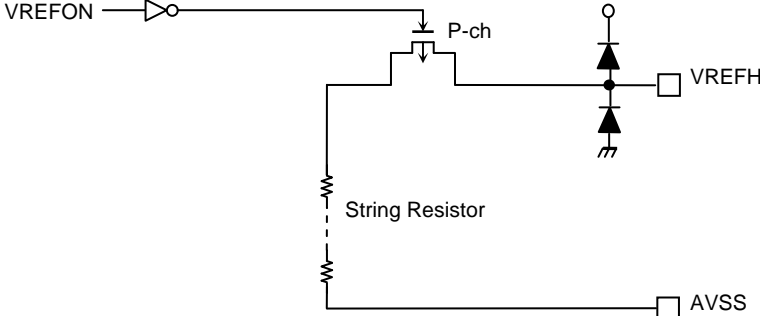
6. Package

LQFP144-2020-0.50E

Unit:mm
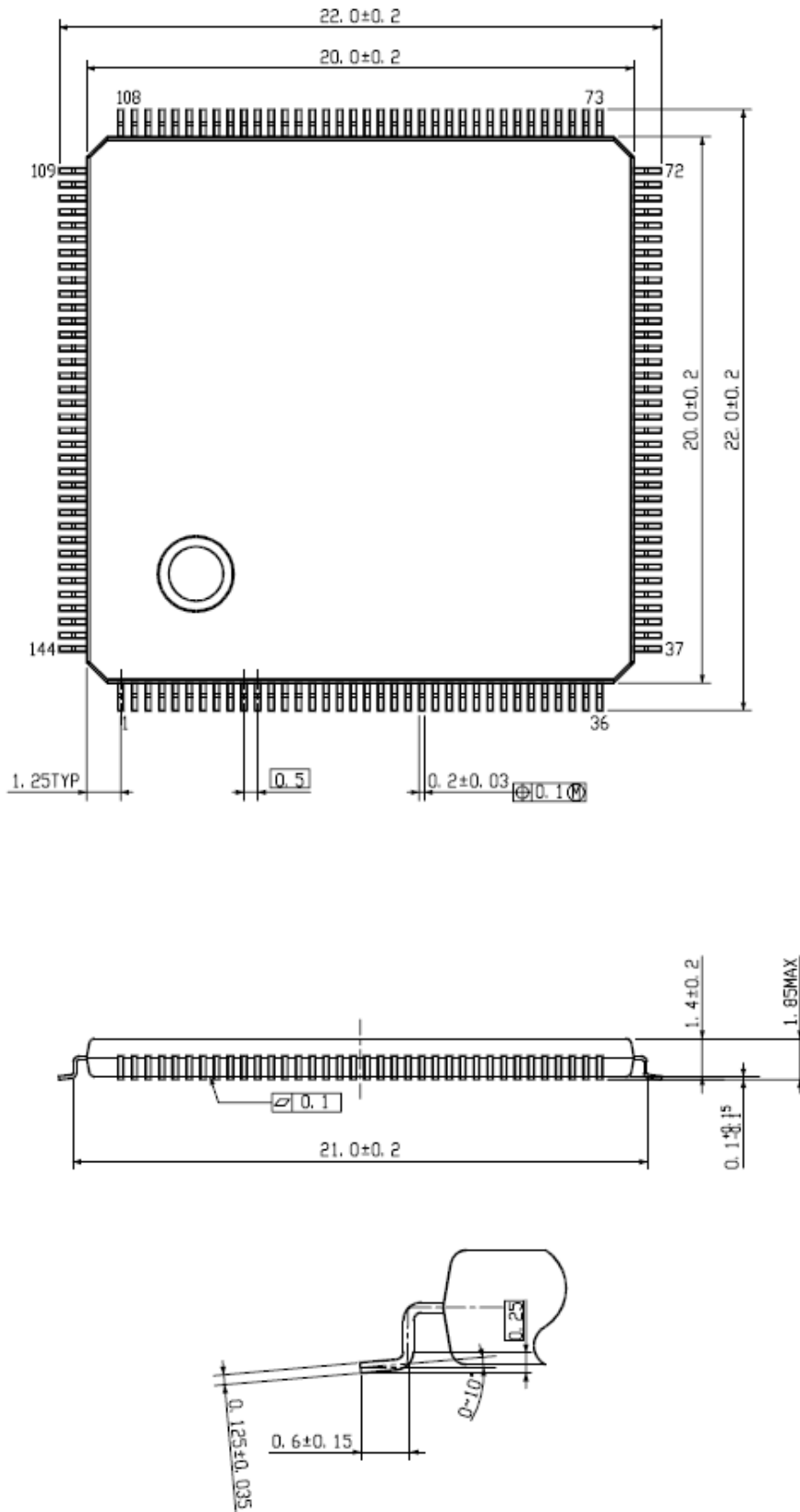
RESTRICTIONS ON PRODUCT USE

- Toshiba Corporation, and its subsidiaries and affiliates (collectively "TOSHIBA"), reserve the right to make changes to the information in this document, and related hardware, software and systems (collectively "Product") without notice.

- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.

- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before creating and producing designs and using, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application that Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**

- Product is intended for use in general electronics applications (e.g., computers, personal equipment, office equipment, measuring equipment, industrial robots and home electronics appliances) or for specific applications as expressly stated in this document. Product is neither intended nor warranted for use in equipment or systems that require extraordinarily high levels of quality and/or reliability and/or a malfunction or failure of which may cause loss of human life, bodily injury, serious property damage or serious public impact ("Unintended Use"). Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. Do not use Product for Unintended Use unless specifically permitted in this document.

- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.

- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.

- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.

- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**

- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.

- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. TOSHIBA assumes no liability for damages or losses occurring as a result of noncompliance with applicable laws and regulations.