TOSHIBA

32 Bit RISC Microcontroller
TX03 Series

TMPM330FDFG/FYFG/FWFG

TOSHIBA CORPORATION

Introduction: Notes on the description of SFR (Special Function Register) under this specification

An SFR (Special Function Register) is a control register for periperal circuits (IP).

The SFR addressses of IPs are described in the chapter on memory map, and the details of SFR are given in the chapter of each IP.

Definition of SFR used in this specification is in accordance with the following rules.

a. SFR table of each IP as an example

- SFR tables in each chapter of IP provides register names, addresses and brief descriptions.
- All registers have a 32-bit unique address and the addresses of the registers are defined as follows, with some exceptions: "Base address + (Unique) address"

Base Address = 0x0000_0000

Register name		Address(Base+)
Control register	SAMCR	0x0004
		0x000C

Note: **SAMCR register address is 32 bits wide from the address 0x0000_0004 (Base Address(0x00000000) + unique address (0x0004)).**

Note: **The register shown above is an example for explanation purpose and not for demonstration purpose. This register does not exist in this microcontroller.**

b. SFR(register)

- Each register basically consists of a 32-bit register (some exceptions).
- The description of each register provides bits, bit symbols, types, initial values after reset and functions.

1.2.2 SAMCR(Control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	MODE	
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MODE	TDATA						
After reset	0	0	0	1	0	0	0	0

Bit	Bit Symbol	Type	Function
31-10	−	R	"0" can be read.
9-7	MODE[2:0]	R/W	Operation mode settings 000 : Sample mode 0 001 : Sample mode 1 010 : Sample mode 2 011 : Sample mode 3 The settings other than those above: Reserved
6-0	TDATA[6:0]	W	Transmitted data

Note: **The Type is divided into three as shown below.**

R / W	READ WRITE
R	READ
W	WRITE

c. Data descriptopn

Meanings of symbols used in the SFR description are as shown below.

- x:channel numbers/ports
- n,m:bit numbers

d. Register descriptoption

Registers are described as shown below.

- Register name <Bit Symbol>

  Exmaple: SAMCR<MODE>="000" or SAMCR<MODE[2:0]>="000"

  <MODE[2:0]> indicates bit 2 to bit 0 in bit symbol mode (3bit width).

- Register name [Bit]

  Example: SAMCR[9:7]="000"

  It indicates bit 9 to bit 7 of the register SAMCR (32 bit width).

Revision History

Date	Revision	Comment
2010/4/26	1	First Release
2010/10/6	2	Contents Revised

Table of Contents

10. Serial Channel (SIO/UART)

11. Serial Bus Interface (I2C/SIO)

12. Consumer Electronics Control (CEC)

13. Remote control signal preprocessor (RMC)

14. Analog/Digital Converter (ADC)

15. Watchdog Timer(WDT)

16. Real Time Clock (RTC)

17. Flash Memory Operation

18. ROM protection

19. Electrical Characteristics

20. Port Section Equivalent Circuit Schematic

21. Package Dimensions

TMPM330FDFG/FYFG/FWFG

The TMPM330FDFG/FYFG/FWFG is a 32-bit RISC microprocessor series with an ARM Cortex™-M3 microprocessor core.

Product Name	ROM (FLASH)	RAM	Package
TMPM330FDFG	512 Kbyte	32 Kbyte	LQFP100-P-1414-0.50H
TMPM330FYFG	256 Kbyte	16 Kbyte	
TMPM330FWFG	128 Kbyte	8 Kbyte	

Features of the TMPM330FDFG/FYFG/FWFG are as follows:

1.1 Features

1. ARM Cortex-M3 microprocessor core
   a. Improved code efficiency has been realized through the use of Thumb® -2 instruction.
      - New 16-bit Thumb instructions for improved program flow
      - New 32-bit Thumb instructions for improved performance
      - New Thumb mixed 16-/32-bit instruction set can produce faster, more efficient code.
   b. Both high performance and low power consumption have been achieved.

      [High performance]
      - A 32-bit multiplication (32×32=32 bit) can be executed with one clock.
      - Division takes between 2 and 12 cycles depending on dividend and devisor

      [Low power consumption]
      - Optimized design using a low power consumption library
      - Standby function that stops the operation of the micro controller core
   c. High-speed interrupt response suitable for real-time control
      - An interruptible long instruction.
      - Stack push automatically handled by hardware.
2. On Chip program memory and data memory

Product name	On chip Flash ROM	On chip RAM
TMPM330FDFG	512 Kbyte	32 Kbyte
TMPM330FYFG	256 Kbyte	16 Kbyte
TMPM330FWFG	128 Kbyte	8 Kbyte

3. 16-bit timer (TMRB): 10 channels
   - 16-bit interval timer mode
   - 16-bit event counter mode
   - 16-bit PPG output
   - Input capture function
4. Real time clock (RTC): 1 channel
   - Clock (hour, minute and second)
   - Calendar (month, week, date and leap year)

- Time correction + or − 30seconds (by software)
- Alarm (Alarm output)
- Alarm interrupt

5. Watchdog timer (WDT): 1 channel

   Watchdog timer (WDT) generates a reset or a non-maskable interrupt (NMI).

6. General-purpose serial interface (SIO/UART): 3 channels

   Either UART mode or synchronous mode can be selected (4byte FIFO equipped)

7. Serial bus interface (I2C/SIO): 3channels

   Either I2C bus mode or synchronous mode can be selected.

8. CEC function (CEC): 1 channel

   Transmission and reception per 1 byte.

9. Remote control signal preprocessor (RMC): 2 channels

   Can receive up to 72bit data at a time

10. 10-bit AD converter (ADC): 12 channels
    - Start by an internal timer trigger
    - Fixed channel/scan mode
    - Single/repeat mode
    - AD monitoring 2ch
    - Conversion speed 1.15μsec. (@fsys = 40MHz)

11. Interrupt source
    - Internal: 42 factors...The order of precedence can be set over 7 levels (except the watchdog timer interrupt).
    - External: 8 factors...The order of precedence can be set over 7 levels.

12. Non-maskable interrupt (NMI)

    Non-maskable interrupt (NMI) is generated by a watchdog timer or a $\overline{\text{NMI}}$ pin.

13. Input/ output ports (PORT): 78 pins

14. Standby mode
    - Standby modes: IDLE, SLOW, SLEEP, STOP
    - Sub clock operation(32.768kHz):SLOW, SLEEP

15. Clock generator (CG)
    - On-chip PLL (quadrupled)
    - Clock gear function: The high-speed clock can be divided into 1/1, 1/2, 1/4 or 1/8.

16. Endian

    Little endian

17. Maximum operating frequency: 40 MHz

18. Operating voltage range

    2.7 V to 3.6 V (with on-chip regulator)

19. Temperature range
    - -20 to 85 degrees (except during Flash writing/ erasing)
    - 0 to 70 degrees (during Flash writing/ erasing)

20. Package

LQFP100-P-1414-0.50H (14mm × 14mm, 0.5mm pitch)

1.2 Block Diagram

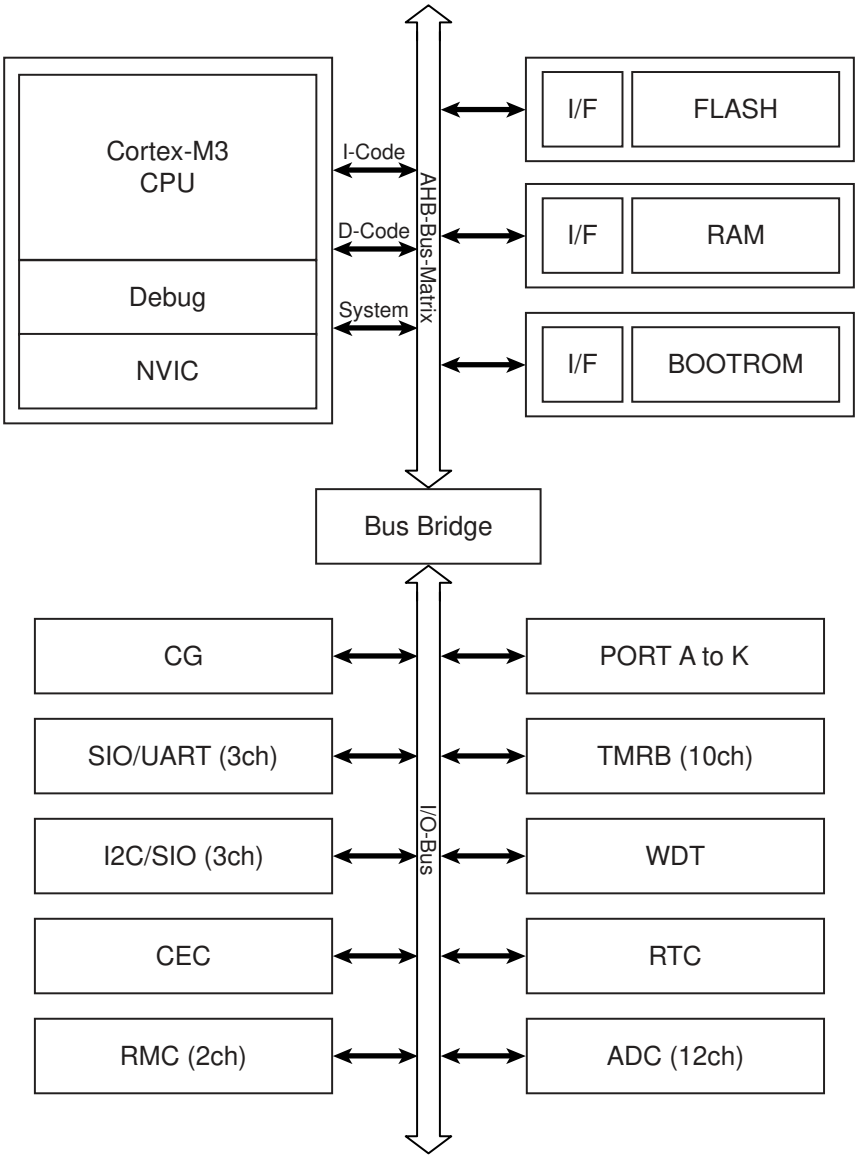


Figure 1-1 TMPM330FDFG/FYFG/FWFGBlock Diagram

1.3 Pin Layout (Top view)

Figure 1-2 shows the pin layout of TMPM330FDFG/FYFG/FWFG.

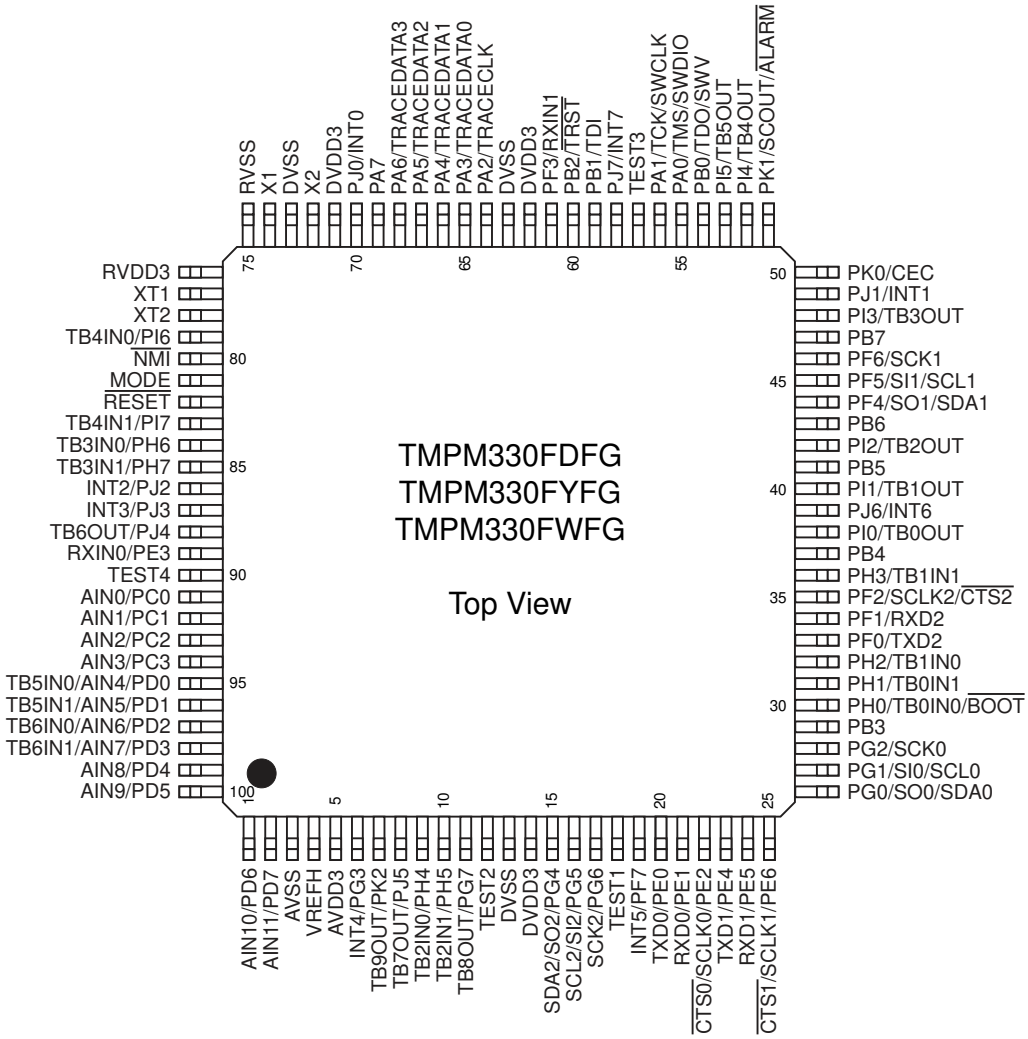


Figure 1-2 Pin Layout (LQFP100)

1.4 Pin names and Functions

Table 1-1 and Table 1-2 sort the input and output pins of the TMPM330FDFG/FYFG/FWFG by pin or port. Each table includes alternate pin names and functions for multi-function pins.

1.4.1 Sorted by Pin

Table 1-1 Pin Names and Functions Sorted by Pin (1/6)

Type	Pin No.	Pin Name	Input/ Out-put	Function
Function	1	PD6 AIN10	I I	Input port Analog input
Function	2	PD7 AIN11	I I	Input port Analog input
PS	3	AVSS	I	AD converter: GND pin (0V) (note) AVSS must be connected to GND even if the A/D converter is not used.
PS	4	VREFH	I	Supplying the AD converter with a reference power supply. (note) VREFH must be connected to power supply even if A/D converter is not used.
PS	5	AVDD3	I	Supplying the AD converter with a power supply. (note) AVDD must be connected to power supply even if A/D converter is not used.
Function	6	PG3 INT4	I/O I	I/O port External interrupt pin
Function	7	PK2 TB9OUT	I/O O	I/O port Timer B output
Function	8	PJ5 TB7OUT	I/O O	I/O port Timer B output
Function	9	PH4 TB2IN0	I/O I	I/O port Inputting the timer B capture trigger
Function	10	PH5 TB2IN1	I/O I	I/O port Inputting the timer B capture trigger
Function	11	PG7 TB8OUT	I/O O	I/O port Timer B output
Test	12	TEST2	−	TEST pin: (note) TEST pin must be left OPEN.
PS	13	DVSS	−	GND pin
PS	14	DVDD3	−	Power supply pin
Function	15	PG4 SDA2/SO2	I/O I/O	I/O port If the serial bus interface operates -in the I2C mode: data pin -in the SIO mode: data pin
Function	16	PG5 SCL2/SI2	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
Function	17	PG6 SCK2	I/O I/O	I/O port Inputting and outputting a clock if the serial bus interface operates in the SIO mode.
Test	18	TEST1	−	TEST pin: (note) TEST pin must be left OPEN.

Table 1-1 Pin Names and Functions Sorted by Pin (2/6)

Type	Pin No.	Pin Name	Input/ Out-put	Function
Function	19	PF7 INT5	I/O I	I/O port External interrupt pin
Function	20	PE0 TXD0	I/O O	I/O port Sending serial data
Function	21	PE1 RXD0	I/O I	I/O port Receiving serial data
Function	22	PE2 SCLK0 $\overline{\text{CTS0}}$	I/O I/O I	I/O port Serial clock input/ output Handshake input pin
Function	23	PE4 TXD1	I/O O	I/O port Sending serial data
Function	24	PE5 RXD1	I/O I	I/O port Receiving serial data
Function	25	PE6 SCLK1 $\overline{\text{CTS1}}$	I/O I/O I	I/O port Serial clock input/ output Handshake input pin
Function	26	PG0 SDA0/SO0	I/O I/O	I/O port -in the I2C mode: data pin -in the SIO mode: data pin
Function	27	PG1 SCL0/SI0	I/O I/O	I/O port -in the I2C mode: clock pin -in the SIO mode: data pin
Function	28	PG2 SCK0	I/O I/O	I/O port Inputting and outputting a clock if the serial bus interface operates in the SIO mode.
Function	29	PB3	I/O	I/O port
Function/ Control	30	PH0 TB0IN0 $\overline{\text{BOOT}}$	I/O I I	I/O port Inputting the timer B capture trigger Setting a single boot mode: (note) This pin goes into single boot mode by sampling "Low" at the rise of a RESET signal.
Function	31	PH1 TB0IN1	I/O I	I/O port Inputting the timer B capture trigger
Function	32	PH2 TB1IN0	I/O I	I/O port Inputting the timer B capture trigger
Function	33	PF0 TXD2	I/O O	I/O port Sending serial data
Function	34	PF1 RXD2	I/O I	I/O port Receiving serial data
Function	35	PF2 SCLK2 $\overline{\text{CTS2}}$	I/O I/O I	I/O port Serial clock input/ output Handshake input pin

Table 1-1 Pin Names and Functions Sorted by Pin (3/6)

Type	Pin No.	Pin Name	Input/ Out-put	Function
Function	36	PH3 TB1IN1	I/O I	I/O port Inputting the timer B capture trigger
Function	37	PB4	I/O	I/O port
Function	38	PI0 TB0OUT	I/O O	I/O port Timer B output
Function	39	PJ6 INT6	I/O I	I/O port External interrupt pin
Function	40	PI1 TB1OUT	I/O O	I/O port Timer B output
Function	41	PB5	I/O	I/O port
Function	42	PI2 TB2OUT	I/O O	I/O port Timer B output
Function	43	PB6	I/O	I/O port
Function	44	PF4 SDA1/SO1	I/O I/O	I/O port -in the I2C mode: data pin -in the SIO mode: data pin
Function	45	PF5 SCL1/SI1	I/O I/O	I/O port -in the I2C mode: clock pin -in the SIO mode: data pin
Function	46	PF6 SCK1	I/O I/O	I/O port Inputting and outputting a clock if the serial bus interface operates in the SIO mode.
Function	47	PB7	I/O	I/O port
Function	48	PI3 TB3OUT	I/O O	I/O port Timer B output
Function	49	PJ1 INT1	I/O I	I/O port External interrupt pin
Function	50	PK0 CEC	I/O I/O	I/O port CEC Pin (note) Nch open drain port.
Function	51	PK1 SCOUT $\overline{\text{ALARM}}$	I/O O O	I/O port System clock output Alarm output
Function	52	PI4 TB4OUT	I/O O	I/O port Timer B output
Function	53	PI5 TB5OUT	I/O O	I/O port Timer B output
Function/ Debug	54	PB0 TDO/SWV	I/O O	I/O port Debug pin
Function/ Debug	55	PA0 TMS/SWDIO	I/O I/O	I/O port Debug pin

Table 1-1 Pin Names and Functions Sorted by Pin (4/6)

Type	Pin No.	Pin Name	Input/ Out-put	Function
Function/ Debug	56	PA1 TCK/SWCLK	I/O I	I/O port Debug pin
Test	57	TEST3	−	TEST pin: (note) TEST pin must be left OPEN.
Function	58	PJ7 INT7	I/O I	I/O port External interrupt pin
Function/ Debug	59	PB1 TDI	I/O I	I/O port Debug pin
Function/ Debug	60	PB2 $\overline{\text{TRST}}$	I/O I	I/O port Debug pin
Function	61	PF3 RXIN1	I/O I	I/O port Inputting signal to remote controller
PS	62	DVDD3	−	Power supply pin
PS	63	DVSS	−	GND pin
Function/ Debug	64	PA2 TRACECLK	I/O O	I/O port Debug pin
Function/ Debug	65	PA3 TRACEDA-TA0	I/O O	I/O port Debug pin
Function/ Debug	66	PA4 TRACEDA-TA1	I/O O	I/O port Debug pin
Function/ Debug	67	PA5 TRACEDA-TA2	I/O O	I/O port Debug pin
Function/ Debug	68	PA6 TRACEDA-TA3	I/O O	I/O port Debug pin
Function	69	PA7	I/O	I/O port
Function	70	PJ0 INT0	I/O I	I/O port External interrupt pin
PS	71	DVDD3	−	Power supply pin
Clock	72	X2	O	Connected to a high-speed oscillator.
PS	73	DVSS	−	GND pin
Clock	74	X1	I	Connected to a high-speed oscillator.
PS	75	RVSS	−	GND pin
PS	76	RVDD3	−	Power supply pin
Clock	77	XT1	I	Connected to a low-speed oscillator.
Clock	78	XT2	O	Connected to a low-speed oscillator.

Table 1-1 Pin Names and Functions Sorted by Pin (5/6)

Type	Pin No.	Pin Name	Input/ Out-put	Function
Function	79	PI6 TB4IN0	I/O I	I/O port Inputting the timer B capture trigger
Function	80	$\overline{\text{NMI}}$	I	Non-maskable interrupt (note) With a noise filter (about 30ns (typical value))
Control	81	MODE	I	Mode pin: (note) MODE pin must be connected to GND.
Function	82	$\overline{\text{RESET}}$	I	Reset input pin (note) With a pull-up and a noise filter (about 30ns (typical value))
Function	83	PI7 TB4IN1	I/O I	I/O port Inputting the timer B capture trigger
Function	84	PH6 TB3IN0	I/O I	I/O port Inputting the timer B capture trigger
Function	85	PH7 TB3IN1	I/O I	I/O port Inputting the timer B capture trigger
Function	86	PJ2 INT2	I/O I	I/O port External interrupt pin
Function	87	PJ3 INT3	I/O I	I/O port External interrupt pin
Function	88	PJ4 TB6OUT	I/O O	I/O port Timer B output
Function	89	PE3 RXIN0	I/O I	I/O port Inputting signal to remote controller
Test	90	TEST4	−	TEST pin: (note) TEST pin must be left OPEN.
Function	91	PC0 AIN0	I I	Input port Analog input
Function	92	PC1 AIN1	I I	Input port Analog input
Function	93	PC2 AIN2	I I	Input port Analog input
Function	94	PC3 AIN3	I I	Input port Analog input
Function	95	PD0 AIN4 TB5IN0	I I I	Input port Analog input Inputting the timer B capture trigger
Function	96	PD1 AIN5 TB5IN1	I I I	Input port Analog input Inputting the timer B capture trigger

Table 1-1 Pin Names and Functions Sorted by Pin (6/6)

Type	Pin No.	Pin Name	Input/ Out-put	Function
Function	97	PD2 AIN6 TB6IN0	I I I	Input port Analog input Inputting the timer B capture trigger
Function	98	PD3 AIN7 TB6IN1	I I I	Input port Analog input Inputting the timer B capture trigger
Function	99	PD4 AIN8	I I	Input port Analog input
Function	100	PD5 AIN9	I I	Input port Analog input

1.4.2 Sorted by Port

Table 1-2 Pin Names and Functions Sorted by Port (1/6)

PORT	Type	Pin No.	Pin Name	Input/ Out-put	Function
PORT A	Function/ Debug	55	PA0 TMS/SWDIO	I/O I/O	I/O port Debug pin
PORT A	Function/ Debug	56	PA1 TCK/SWCLK	I/O I	I/O port Debug pin
PORT A	Function/ Debug	64	PA2 TRACECLK	I/O O	I/O port Debug pin
PORT A	Function/ Debug	65	PA3 TRACEDA-TA0	I/O O	I/O port Debug pin
PORT A	Function/ Debug	66	PA4 TRACEDA-TA1	I/O O	I/O port Debug pin
PORT A	Function/ Debug	67	PA5 TRACEDA-TA2	I/O O	I/O port Debug pin
PORT A	Function/ Debug	68	PA6 TRACEDA-TA3	I/O O	I/O port Debug pin
PORT A	Function	69	PA7	I/O	I/O port
PORT B	Function/ Debug	54	PB0 TDO/SWV	I/O O	I/O port Debug pin
PORT B	Function/ Debug	59	PB1 TDI	I/O I	I/O port Debug pin
PORT B	Function/ Debug	60	PB2 $\overline{\text{TRST}}$	I/O I	I/O port Debug pin
PORT B	Function	29	PB3	I/O	I/O port
PORT B	Function	37	PB4	I/O	I/O port
PORT B	Function	41	PB5	I/O	I/O port
PORT B	Function	43	PB6	I/O	I/O port
PORT B	Function	47	PB7	I/O	I/O port
PORT C	Function	91	PC0 AIN0	I I	Input port Analog input
PORT C	Function	92	PC1 AIN1	I I	Input port Analog input
PORT C	Function	93	PC2 AIN2	I I	Input port Analog input
PORT C	Function	94	PC3 AIN3	I I	Input port Analog input
PORT D	Function	95	PD0 AIN4 TB5IN0	I I I	Input port Analog input Inputting the timer B capture trigger

Table 1-2 Pin Names and Functions Sorted by Port (2/6)

PORT	Type	Pin No.	Pin Name	Input/ Out-put	Function
PORT D	Function	96	PD1 AIN5 TB5IN1	I I I	Input port Analog input Inputting the timer B capture trigger
PORT D	Function	97	PD2 AIN6 TB6IN0	I I I	Input port Analog input Inputting the timer B capture trigger
PORT D	Function	98	PD3 AIN7 TB6IN1	I I I	Input port Analog input Inputting the timer B capture trigger
PORT D	Function	99	PD4 AIN8	I I	Input port Analog input
PORT D	Function	100	PD5 AIN9	I I	Input port Analog input
PORT D	Function	1	PD6 AIN10	I I	Input port Analog input
PORT D	Function	2	PD7 AIN11	I I	Input port Analog input
PORT E	Function	20	PE0 TXD0	I/O O	I/O port Sending serial data
PORT E	Function	21	PE1 RXD0	I/O I	I/O port Receiving serial data
PORT E	Function	22	PE2 SCLK0 $\overline{\text{CTS0}}$	I/O I/O I	I/O port Serial clock input/ output Handshake input pin
PORT E	Function	89	PE3 RXIN0	I/O I	I/O port Inputting signal to remote controller
PORT E	Function	23	PE4 TXD1	I/O O	I/O port Sending serial data
PORT E	Function	24	PE5 RXD1	I/O I	I/O port Receiving serial data
PORT E	Function	25	PE6 SCLK1 $\overline{\text{CTS1}}$	I/O I/O I	I/O port Serial clock input/ output Handshake input pin
PORT F	Function	33	PF0 TXD2	I/O O	I/O port Sending serial data
PORT F	Function	34	PF1 RXD2	I/O I	I/O port Receiving serial data
PORT F	Function	35	PF2 SCLK2 $\overline{\text{CTS2}}$	I/O I/O I	I/O port Serial clock input/ output Handshake input pin
PORT F	Function	61	PF3 RXIN1	I/O I	I/O port Inputting signal to remote controller

Table 1-2 Pin Names and Functions Sorted by Port (3/6)

PORT	Type	Pin No.	Pin Name	Input/ Out-put	Function
PORT F	Function	44	PF4 SDA1/SO1	I/O I/O	I/O port -in the I2C mode: data pin -in the SIO mode: data pin
PORT F	Function	45	PF5 SCL1/SI1	I/O I/O	I/O port -in the I2C mode: clock pin -in the SIO mode: data pin
PORT F	Function	46	PF6 SCK1	I/O I/O	I/O port Inputting and outputting a clock if the serial bus interface operates in the SIO mode.
PORT F	Function	19	PF7 INT5	I/O I	I/O port External interrupt pin
PORT G	Function	26	PG0 SDA0/SO0	I/O I/O	I/O port -in the I2C mode: data pin -in the SIO mode: data pin
PORT G	Function	27	PG1 SCL0/SI0	I/O I/O	I/O port -in the I2C mode: clock pin -in the SIO mode: data pin
PORT G	Function	28	PG2 SCK0	I/O I/O	I/O port Inputting and outputting a clock if the serial bus interface operates in the SIO mode.
PORT G	Function	6	PG3 INT4	I/O I	I/O port External interrupt pin
PORT G	Function	15	PG4 SDA2/SO2	I/O I/O	I/O port If the serial bus interface operates -in the I2C mode: data pin -in the SIO mode: data pin
PORT G	Function	16	PG5 SCL2/SI2	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
PORT G	Function	17	PG6 SCK2	I/O I/O	I/O port Inputting and outputting a clock if the serial bus interface operates in the SIO mode.
PORT G	Function	11	PG7 TB8OUT	I/O O	I/O port Timer B output
PORT H	Function/ Control	30	PH0 TB0IN0 $\overline{\text{BOOT}}$	I/O I I	I/O port Inputting the timer B capture trigger Setting a single boot mode: This pin goes into single boot mode by sampling "Low" at the rise of a RESET signal.
PORT H	Function	31	PH1 TB0IN1	I/O I	I/O port Inputting the timer B capture trigger
PORT H	Function	32	PH2 TB1IN0	I/O I	I/O port Inputting the timer B capture trigger
PORT H	Function	36	PH3 TB1IN1	I/O I	I/O port Inputting the timer B capture trigger

Table 1-2 Pin Names and Functions Sorted by Port (4/6)

PORT	Type	Pin No.	Pin Name	Input/ Out-put	Function
PORT H	Function	9	PH4 TB2IN0	I/O I	I/O port Inputting the timer B capture trigger
PORT H	Function	10	PH5 TB2IN1	I/O I	I/O port Inputting the timer B capture trigger
PORT H	Function	84	PH6 TB3IN0	I/O I	I/O port Inputting the timer B capture trigger
PORT H	Function	85	PH7 TB3IN1	I/O I	I/O port Inputting the timer B capture trigger
PORT I	Function	38	PI0 TB0OUT	I/O O	I/O port Timer B output
PORT I	Function	40	PI1 TB1OUT	I/O O	I/O port Timer B output
PORT I	Function	42	PI2 TB2OUT	I/O O	I/O port Timer B output
PORT I	Function	48	PI3 TB3OUT	I/O O	I/O port Timer B output
PORT I	Function	52	PI4 TB4OUT	I/O O	I/O port Timer B output
PORT I	Function	53	PI5 TB5OUT	I/O O	I/O port Timer B output
PORT I	Function	79	PI6 TB4IN0	I/O I	I/O port Inputting the timer B capture trigger
PORT I	Function	83	PI7 TB4IN1	I/O I	I/O port Inputting the timer B capture trigger
PORT J	Function	70	PJ0 INT0	I/O I	I/O port External interrupt pin
PORT J	Function	49	PJ1 INT1	I/O I	I/O port External interrupt pin
PORT J	Function	86	PJ2 INT2	I/O I	I/O port External interrupt pin
PORT J	Function	87	PJ3 INT3	I/O I	I/O port External interrupt pin
PORT J	Function	88	PJ4 TB6OUT	I/O O	I/O port Timer B output
PORT J	Function	8	PJ5 TB7OUT	I/O O	I/O port Timer B output
PORT J	Function	39	PJ6 INT6	I/O I	I/O port External interrupt pin
PORT J	Function	58	PJ7 INT7	I/O I	I/O port External interrupt pin

Table 1-2 Pin Names and Functions Sorted by Port (5/6)

PORT	Type	Pin No.	Pin Name	Input/ Out-put	Function
PORT K	Function	50	PK0 CEC	I/O I/O	I/O port CEC Pin (note) Nch open drain port.
PORT K	Function	51	PK1 SCOUT $\overline{\text{ALARM}}$	I/O O O	I/O port System clock output Alarm output
PORT K	Function	7	PK2 TB9OUT	I/O O	I/O port Timer B output
-	Function	82	$\overline{\text{RESET}}$	I	Reset input pin (note) With a pull-up and a noise filter (about 30ns (typical value))
-	Function	80	$\overline{\text{NMI}}$	I	Non-maskable interrupt (note) With a noise filter (about 30ns (typical value))
-	Control	81	MODE	I	Mode pin: (note) MODE pin must be connected to GND.
-	Clock	72	X2	O	Connected to a high-speed oscillator.
-	Clock	74	X1	I	Connected to a high-speed oscillator.
-	Clock	77	XT1	I	Connected to a low-speed oscillator.
-	Clock	78	XT2	O	Connected to a low-speed oscillator.
-	Test	12	TEST2	−	TEST pin: (note) TEST pin must be left OPEN.
-	Test	18	TEST1	−	TEST pin: (note) TEST pin must be left OPEN.
-	Test	57	TEST3	−	TEST pin: (note) TEST pin must be left OPEN.
-	Test	90	TEST4	−	TEST pin: (note) TEST pin must be left OPEN.
-	PS	3	AVSS	I	AD converter: GND pin (0V) (note) AVSS must be connected to GND even if the A/D converter is not used.
-	PS	4	VREFH	I	Supplying the AD converter with a reference power supply. (note) VREFH must be connected to power supply even if A/D converter is not used.
-	PS	5	AVDD3	I	Supplying the AD converter with a power supply. (note) AVDD must be connected to power supply even if A/D converter is not used.
-	PS	13	DVSS	−	GND pin
-	PS	14	DVDD3	−	Power supply pin
-	PS	62	DVDD3	−	Power supply pin
-	PS	63	DVSS	−	GND pin

Table 1-2 Pin Names and Functions Sorted by Port (6/6)

PORT	Type	Pin No.	Pin Name	Input/ Out- put	Function
-	PS	71	DVDD3	−	Power supply pin
-	PS	73	DVSS	−	GND pin
-	PS	75	RVSS	−	GND pin
-	PS	76	RVDD3	−	Power supply pin

1.5 Pin Numbers and Power Supply Pins

Table 1-3 Pin Numbers and Power Supplies

Power supply	Voltage range	Pin No.	Pin name
DVDD3	2.7 to 3.6V	14, 62,71	PA,PB,PE,PF,PG,PH,PI,PJ,PK,X1,X2,XT1, XT2,$\overline{\text{RESET}}$,$\overline{\text{NMI}}$,MODE
AVDD3		5	PC,PD
RVDD3		76	−

2. Processor Core

The TX03 series 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 TX03 series that are not explained in that document.

2.1 Information on the processor core

The following table shows the revision of the processor core in the TMPM330FDFG/FYFG/FWFG.

Refer to the detailed information about the CPU core and architecture, refer to the ARM manual "Cortex-M series processors" in the following URL:

http://infocenter.arm.com/help/index.jsp

Product Name	Core Revision
TMPM330FDFG TMPM330FYFG	r1p1-00rel0
TMPM330FWFG	r1p1-01rel0

2.2 Configurable Options

The Cortex-M3 core has optional blocks. The optional blocks of the revision r1p1 are ETM™ and MPU. The following tables shows the configurable options in the TMPM330FDFG/FYFG/FWFG.

Configurable Options	Implementation
MPU	Not implementable
ETM	Implementable

2.3 Exceptions/ Interruptions

Exceptions and interruptions are described in the following section.

2.3.1 Number of Interrupt Inputs

The number of interrupt inputs can optionally be defined from 1 to 240 in the Cortex-M3 core.

TMPM330FDFG/FYFG/FWFG has 50 interrupt inputs. The number of interrupt inputs is reflected in <INTLINESNUM[4:0]> bit of NVIC register. In this product, if read <INTLINESNUM[4:0]> bit, "0y00001" is read out.

2.3.2 Number of Priority Level Interrupt Bits

The Cortex-M3 core can optionally configure the number of priority level interrupt bits from 3 bits to 8 bits.

TMPM330FDFG/FYFG/FWFG has three priority level interrupt bits. The number of priority level interrupt bits is used for assigning a priority level in the interrupt priority registers and system handler priority registers.

2.3.3 SysTick

The Cortex-M3 core has a SysTick timer which can generate SysTick exception.

In the TMPM330FDFG/FYFG/FWFG, the clock that is input from X1 pin dividing by 32 is used as a count clock for the Systic timer. SysTick calibration register can set a calibration value to measure 10ms. In this product, when 8MHz is input to X1 pin, calibration value is set to 0x9C4 which can measure 10ms. Additionally, if this value is read as "0" both of <NOREF> bit and <SKEW> bit, it indicates that external reference clock are available and the calibration value is accurate as 10ms.

2.3.4 SYSRESETREQ

The Cortex-M3 core outputs SYSRESETREQ signal when <SYSRESETREQ> bit of Application Interrupt and Reset Control Register are set.

TMPM330FDFG/FYFG/FWFG provides the same operation when SYSRESETREQ signal are output.

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

2.3.5 LOCKUP

When irreparable exception generates, the Cortex-M3 core outputs LOCKUP signal to show a serious error included in software.

TMPM330FDFG/FYFG/FWFG does not use this signal. To return from LOCKUP status, it is necessary to use non-maskable interruput (NMI) or reset.

2.3.6 Auxiliary Fault Status register

The Cortex-M3 core provides auxiliary fault status registers to supply additional system fault information to software.

However, TMPM330FDFG/FYFG/FWFG is not defined this function. If auxiliary fault status register is read, always "0x0000_0000" is read out.

2.4 Events

The Cortex-M3 core has event output signals and event input signals. An event output signal is output by SEV instruction execution. If an event is input, the core returns from low-power consumption mode caused by WFE instruction.

TMPM330FDFG/FYFG/FWFG does not use event output signals and event input signals. Please do not use SEV instruction and WFE instruction.

2.5 Power Management

The Cortex-M3 core provides power management system which uses SLEEPING signals and SLEEPDEEP signals. SLEEPDEEP signals are output when <SLEEPDEEP> bit of System Control Register is set.

These signals are output in the following circumstances:

-Wait-For-Interrupt (WFI) instruction execution

-Wait-For-Event (WFE) instruction execution

-the timing when interrupt-service-routine (ISR) exit in case that <SLEEPONEXIT> bit of System Control Register is set.

TMPM330FDFG/FYFG/FWFG does not use SLEEPDEEP signals so that <SLEEPDEEP> bit must not be set. And also event signals are not used so that please do not use WFE instruction.

For detail of power management, refer to the Chapter "Clock/Mode control."

2.6 Exclusive access

In Cortex-M3 core, the DCode bus system supports exclusive access. However TMPM330FDFG/FYFG/FWFG does not use this function.

3. Debug Interface

3.1 Specification Overview

TMPM330FDFG/FYFG/FWFG contains the Serial Wire JTAG Debug Port (SWJ-DP) unit for interfacing with the debugging tools and the Embedded Trace Macrocell™(ETM) unit for instruction trace output.Trace data is output to the dedicated pins(TRACEDATA[3:0], SWV) for the debugging via the on-chip Trace Port Interface Unit (TPIU).

For details about SWJ-DP, ETM and TPIU, refer to "Cortex-M3 Technical Reference Manual" .

3.2 SWJ-DP

SWJ-DP supports the Serial Wire Debug Port (SWDCK, SWDIO) and the JTAG Debug Port (TDI, TDO, TMS, TCK, $\overline{\text{TRST}}$).

3.3 ETM

ETM supports four data signal pins (TRACEDATA[3:0]), one clock signal pin (TRACECLK) and trace output from SWV.

3.4 Pin Functions

The debug interface pins can also be used as general-purpose ports.

The PA0 and PA1 pins are shared between the JTAG debug port function and the Serial Wire Debug Port function. The PB0 pin is shared between the JTAG debug port function and the SWV trace output function.

Table 3-1 SWJ-DP,ETM Debug Functions

SWJ-DP Pin Name	General-purpose Port Name	JTAG Debug Function		SW Debug Function	
		I / O	Explanation	I / O	Explanation
TMS / SWDIO	PA0	Input	JTAG Test Mode Selection	I / O	Serial Wire Data Input/Output
TCK / SWCLK	PA1	Input	JTAG Test Check	Input	Serial Wire Clock
TDO / SWV	PB0	Output	JTAG Test Data Output	(Output)(Note)	(Serial Wire Viewer Output)
TDI	PB1	Input	JTAG Test Data Input	-	-
$\overline{\text{TRST}}$	PB2	Input	JTAG Test $\overline{\text{RESET}}$	-	-
TRACECLK	PA2	Output	TRACE Clock Output		
TRACEDATA0	PA3	Output	TRACE DATA Output0		
TRACEDATA1	PA4	Output	TRACE DATA Output1		
TRACEDATA2	PA5	Output	TRACE DATA Output2		
TRACEDATA3	PA6	Output	TRACE DATA Output3		

Note: **When SWV function is enabled.**

After reset, PA0, PA1, PB0, PB1 and PB2 pins are configured as debug port function pins. The functions of other debug interface pins need to be programmed as required.

When using a low power consumption mode, take note of the following points.

Note 1: If PA0 and PB0 are configured as TMS/SWDIO and TDO/SWV, output continues to be enabled even in STOP mode regardless of the setting of the CGSTBYCR<DRVE> bit.

Note 2: If PA1 is configured as a debug function pin, it prevents a low power consumption mode from being fully effective. Configure PA1 to function as a general-purpose port if the debug function is not used.

Table 3-2 summarizes the debug interface pin and related port settings after reset.

Table 3-2 Debug Interface Pins and Related Port Settings after Reset

Port Name (Bit Name)	Debug Function	Value of Related port settings after reset				
		Function (PxFR)	Input (PxIE)	Output (PxCR)	Pull-up (PxPUP)	Pull-down (PxPDN)
PA0	TMS/SWDIO	1	1	1	1	−
PA1	TCK/SWCLK	1	1	0	−	1
PB0	TDO/SWV	1	0	1	0	−
PB1	TDI	1	1	0	1	−
PB2	$\overline{\text{TRST}}$	1	1	0	1	−
PA2	TRACECLK	0	0	0	0	−
PA3	TRACEDATA0	0	0	0	0	−
PA4	TRACEDATA1	0	0	0	0	−
PA5	TRACEDATA2	0	0	0	0	−
PA6	TRACEDATA3	0	0	0	0	−

− : Don't care

3.5 Peripheral Functions in Halt Mode

When the Cortex-M3 core enters in the halt mode, the watchdog-timer (WDT) automatically stops. Other peripheral functions continue to operate.

3.6 Reset Vector Break

TMPM330FDFG/FYFG/FWFG is prohibited from transmission with debug tools while reset caused by $\overline{\text{RESET}}$ pin is effective.When setting a stop by using reset vector, set the following procedure after reset; set break points from the debug tools, then set the application interrupt and the <SYSRESETREQ> bit of the reset control register to reset again.

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

3.7 Connection with a Debug Tool

3.7.1 About connection with debug tool

Concerning a connection with debug tools, refer to manufactures recommendations.

Debug interface pins contain a pull-up resistor and a pull-down resistor.When debug interface pins are connected with external pull-up or pull-down, please pay attention to input level.

3.7.2 Important points of using debug interface pins used as general-purpose ports

TMPM330FDFG/FYFG/FWFG is prohibited from transmission with debug tools while reset caused by $\overline{\text{RESET}}$ pin is effective. Therefor it cannot change to the debug mode.

The PA0, PA1, PB0, PB1 and PB2 ports are the debug interface pins after reset however if these pins are changed to the general-purpose port immediately after reset, the control from the debug tools are not accepted under some circumstances.When changing the settings, please pay attention to the status of debug interface pins.

Table 3-3 Table of using debug interface pins

	Debug interface pins						
	$\overline{\text{TRST}}$	TDI	TDO / SWV	TCK / SWCLK	TMS / SWDIO	TRACE DATA[3:0]	TRACE CLK
JTAG+SW (After reset)	o	o	o	o	o	×	×
JTAG+SW (without TRST)	×	o	o	o	o	×	×
JTAG+TRACE	o	o	o	o	o	o	o
SW	×	×	×	o	o	×	×
SW+SWV	×	×	o	o	o	×	×
Debugging function disabled	×	×	×	×	×	×	×

o : Enabled × : Disabled (Usable as general-purpose port)

4. Memory Map

4.1 Memory map

The memory maps for theTMPM330FDFG/FYFG/FWFG are based on the ARM Cortex-M3 processor core memory map.

The internal ROM is mapped to the code of the Cortex-M3 core memory, the internal RAM is mapped to the SRAM region and the special function register (SFR) is mapped to the peripheral region respectively.

The special function register (SFR) indicates I/O ports and control registers for the peripheral function. The SRAM and SFR 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.

4.1.1 Memory map of the TMPM330FDFG

Figure 4-1shows the memory map of the TMPM330FDFG.

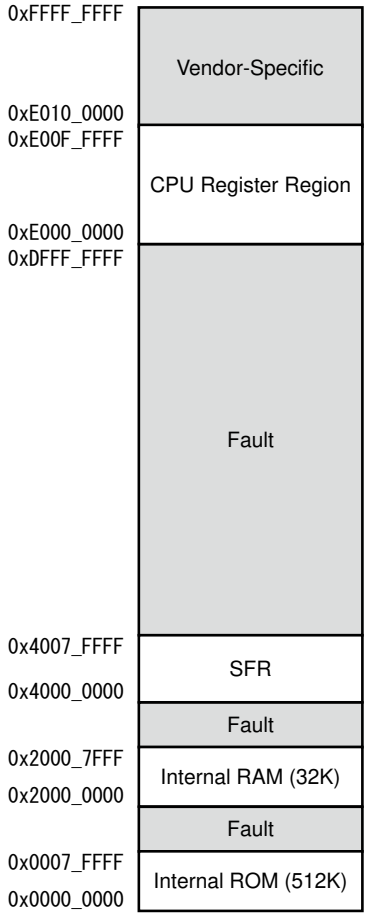


Figure 4-1 Memory Map (TMPM330FDFG)

4.1.2 Memory Map of TMPM330FYFG

Figure 4-2 shows the memory map of the TMPM330FYFG.

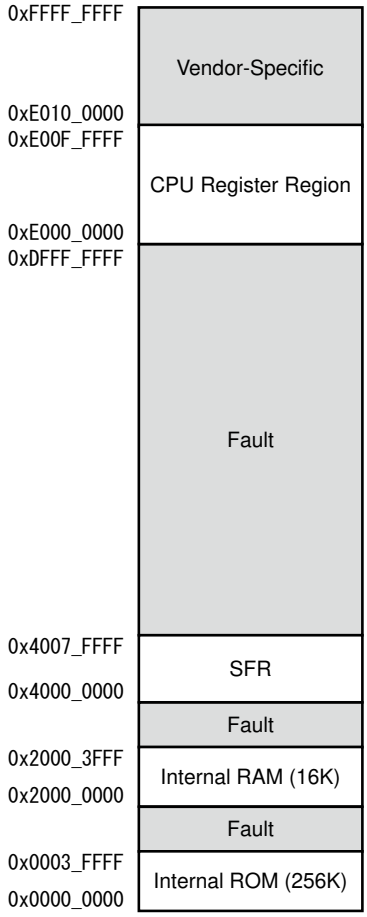


Figure 4-2 Memory Map (TMPM330FYFG)

Note: In addition to 256KB flash area, the TMPM330FYFG provides 128-word data/ password area (1 page) for Show Product Information command in the address range 0x0007_FE00 - 0x0007_FFFF. See the Chapter "Flash Memory Operation" for details on the single boot mode. Do not Access to the range from 0x0004_0000 through the password area.

4.1.3 Memory Map of TMPM330FWFG

Figure 4-3 shows the memory map of the TMPM330FWFG.

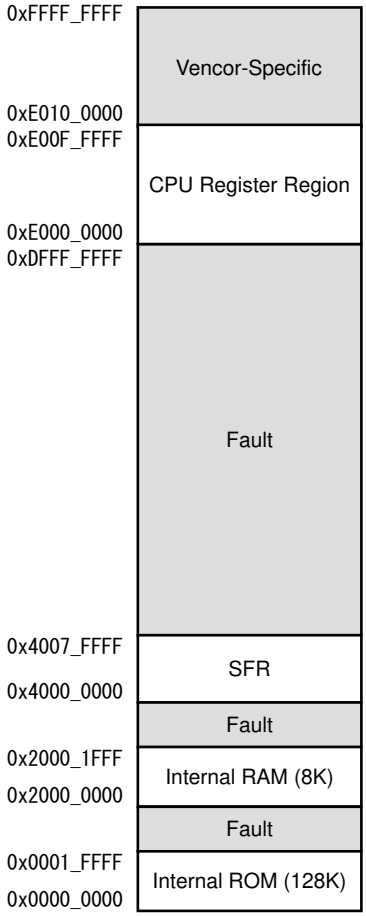


Figure 4-3 Memory Map (TMPM330FWFG)

4.2 SFR area detail

This section contains the list of addresses in the SFR area (0x4000_0000 through 0x4007_FFFF) assigned to peripheral function.

Access to the Reserved areas in the Table 4-1 is prohibited. As for the SFR area, reading the areas not described in the Table 4-1 yields undefined value. Writing these area is ignored.

Table 4-1 SFR area detail

Start Address	End Address	Peripheral	Reserved
0x4000_0000	0x4000_02BF	PORT(A to K)	0x4000_0190 to 0x4000_0193 0x4000_01D0 to 0x4000_01D3 0x4000_0210 to 0x4000_0213 0x4000_0250 to 0x4000_0253
0x4001_0000	0x4001_027F	TMRB(10ch)	
0x4002_0000	0x4002_007F	I2C/SIO(3ch)	
0x4002_0080	0x4002_013F	SIO/UART(3ch)	
0x4003_0000	0x4003_007F	ADC(12ch)	0x4003_0024 - 0x4003_002F
0x4004_0000	0x4004_003F	WDT	
0x4004_0100	0x4004_013F	RTC	0x4004_010D
0x4004_0200	0x4004_023F	CG	0x4004_0230 to 0x4004_023F
0x4004_0300	0x4004_033F	CEC	
0x4004_0400	0x4004_047F	RMC(2ch)	0x4004_0428 to 0x4004_0433 0x4004_0468 to 0x4004_0473
0x4004_0500	0x4004_053F	FLASH	0x4004_0504 to 0x4004_0507 0x4004_0524 to 0x4004_052B
0x4004_0540	0x4004_05BF	Reserved	0x4004_0540 to 0x4004_0547 0x4004_0550 to 0x4004_0553 0x4004_0560 to 0x4004_0593
0x4004_0700	0x4004_073F	Reserved	0x4004_0700 to 0x4004_0707

5. Reset

The TMPM330FDFG/FYFG/FWFG has three reset sources: an external reset pin ($\overline{\text{RESET}}$), a watchdog timer (WDT) and the setting <SYSRESETREQ> in the Application Interrupt and Reset Control Register.

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.**

5.1 Cold reset

The power-on sequence must consider the time for the internal regulator and oscillator to be stable.In the TMPM330FDFG/FYFG/FWFG, the internal regulator requires at least 700 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.

Figure 5-1 shows the power-on sequence.

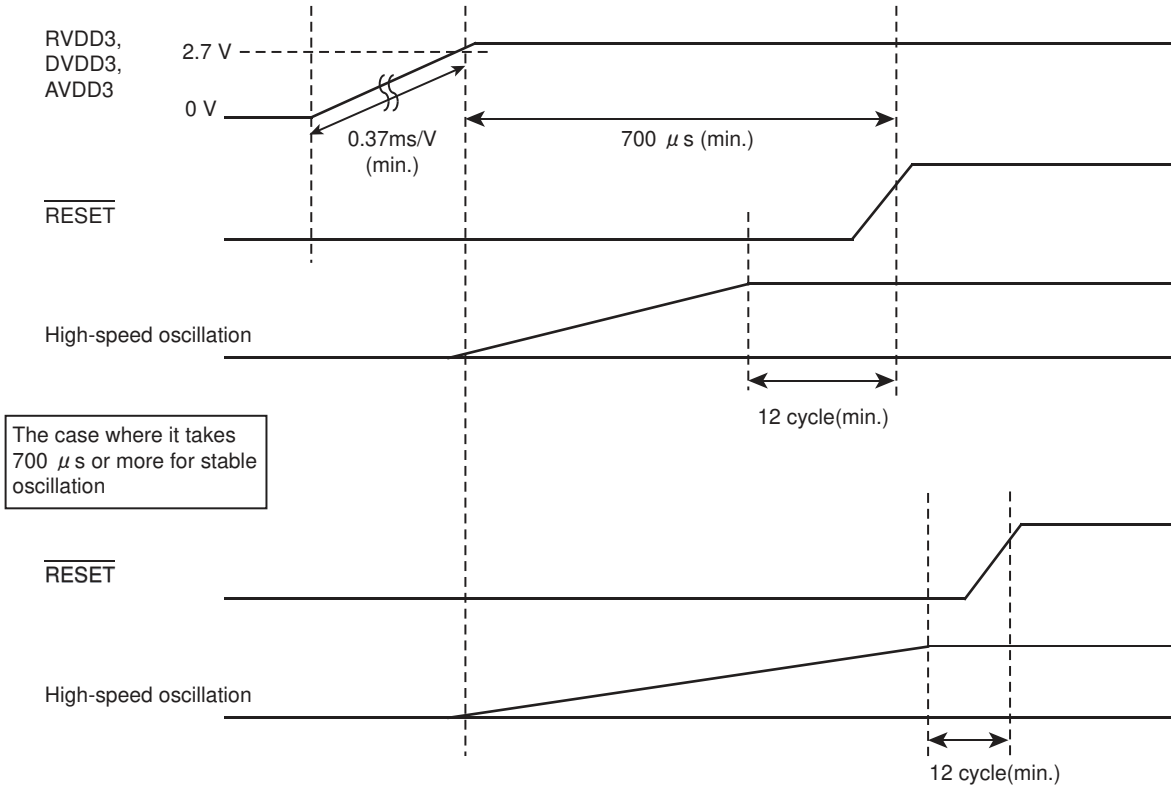


Figure 5-1 Cold Reset Sequence

Note 1: **The power supply must be raised (from 0V to 2.7V) at a speed of 0.37ms/V or slower.**

Note 2: **Turn on the power while the $\overline{\text{RESET}}$ pin is fixed to "Low". Release the reset while all the power supplies are stabilized within operating voltage.**

5.2 Warm reset

5.2.1 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 TMPM330FDFG/FYFG/FWFG, assert the $\overline{\text{RESET}}$ signal (active low) for a minimum duration of 12 system clocks.

5.2.2 After reset

A warm reset initializes the majority of the Cortex-M3 processor core's system control registers and internal function registers.

The processor core's system debug components (FPB, DWT, ITM) register, the clock generator's CGRSTFLG register and the FCSECBIT register are initialized by a only cold reset.

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: **The reset operation may alter the internal RAM state.**

6. Clock/Mode control

6.1 Features

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

There is also the low power consumption mode which can reduce power consumption by mode transitions.

This chapter describes how to control 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 TMPM330FDFG/FYFG/FWFG can operate in three types of low power mode to reduce power consumption according to its usage conditions.

6.2 Registers

6.2.1 Register List

The following table shows the CG-related registers and addresses.

Base Address = 0x4004_0200

Register name		Address (Base+)
System control register	CGSYSCR	0x0000
Oscillation control register	CGOSCCR	0x0004
Standby control register	CGSTBYCR	0x0008
PLL selection register	CGPLLSEL	0x000C
System clock selection register	CGCKSEL	0x0010

6.2.2 CGSYSCR (System control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	SCOSEL	
After reset	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	FPSEL	-	PRCK		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	GEAR		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-18	−	R	Read as 0.
17-16	SCOSEL[1:0]	R/W	SCOUT out 00: fs 01: fsys/2 10: fsys 11: φT0 Enables to output the specified clock from SCOUT pin.
15-13	−	R	Read as 0.
12	FPSEL	R/W	fperiph 0: fgear 1: fc Specifies the source clock to fperiph.
11	−	R	Read as 0.
10-8	PRCK[2:0]	R/W	Prescaler clock 000: fperiph 001: fperiph/2 010: fperiph/4 011: fperiph/8 100: fperiph/16 101: fperiph/32 110: Reserved 111: Reserved Specifies the prescaler clock to peripheral I/O.
7-3	−	R	Read as 0.
2-0	GEAR[2:0]	R/W	High-speed clock gear (fc) gear 000: fc 001: Reserved 010: Reserved 011: Reserved 100: fc/2 101: fc/4 110: fc/8 111: Reserved

6.2.3 CGOSCCR (Oscillation control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	XTEN	XEN
After reset	0	0	0	0	0	0	1	1
	7	6	5	4	3	2	1	0
bit symbol	-	WUPT			WUPSEL	PLLON	WUEF	WUEON
After reset	0	0	0	1	0	0	0	0

Bit	Bit Symbol	Type	Function
31-14	−	R	Read as 0.
13-12	−	R/W	Write "0".
11-10	−	R	Read as 0
9	XTEN	R/W	Low-speed oscillator 0: Stop 1:Oscillation
8	XEN	R/W	High-speed oscillator 0: stop 1:Oscillation
7	−	R	Read as 0
6-4	WUPT[2:0]	R/W	Warm-up time X1 / XT1 000: No warm-up / 000:No warm-up 001: 2^{10}/ input freq. / 001: 2^{6}/ input freq. 010: 2^{11}/ input freq. / 010: 2^{7}/ input freq. 011: 2^{12}/ input freq. / 011: 2^{8}/ input freq. 100: 2^{13}/input freq. / 100: 2^{15}/ input freq. 101: 2^{14}/ input freq. / 101: 2^{16}/ input freq. 110: 2^{15}/ input freq. / 110: 2^{17}/ input freq. 111: 2^{16}/ input freq. / 111: 2^{18}/ input freq.
3	WUPSEL	R/W	Warm-up counter 0: X1 1: XT1 Specifies the oscillator to warm-up. A clock generated by the specified oscillator is used for the warm-up timer count.
2	PLLON	R/W	PLL operation 0: Stop 1: Oscillation Specifies operation of the PLL. It stops after reset.Setting the bit is required.
1	WUEF	R	Status of Warm-up timer (WUP) 0:Warm-up completed 1: Warm-up operation Enables to monitor the status of the warm-up timer.
0	WUEON	W	Operation of warm-up timer 0: don't care 1: starting warm-up Enables to start the warm-up timer.

6.2.4 CGSTBYCR (Standby control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	DRVE
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	RXTEN	RXEN
After reset	0	0	0	0	0	0	0	1
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	STBY		
After reset	0	0	0	0	0	0	1	1

Bit	Bit Symbol	Type	Function
31-18	−	R	Read as 0.
17	−	R/W	Write "0".
16	DRVE	R/W	Pin status in STOP mode. 0: Inactive 1:Active
15-10	−	R	Read as 0
9	RXTEN	R/W	Low-speed oscillator operation after releasing the STOP mode. 0: Stop 1:Oscillation
8	RXEN	R/W	High-speed oscillator operation after releasing the STOP mode. 0: Stop 1:Oscillation
7-3	−	R	Read as 0.
2-0	STBY[2:0]	R/W	Low power consumption mode 000: Reserved 001: STOP 010: SLEEP 011: IDLE 100: Reserved 101: Reserved 110: Reserved 111: Reserved

6.2.5 CGPLLSEL (PLL Selection Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	PLLSEL
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	PLLSEL	R/W	Use of PLL 0: Disuse. X1 selected 1: Use Specifies use or disuse of the clock multiplied by the PLL. "X1" is automatically set after reset. Resetting is required when using the PLL.

6.2.6 CGCKSEL (System clock selection register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	SYSCK	SYSCKFLG
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	SYSCK	R/W	System clock 0: High-speed (fc) 1: Low-speed (fs) Enables to specify the system clock. Setting CGOSCCR<XEN> and <XTEN> to "1" in advance is required.
0	SYSCKFLG	R	System clock status 0: High-speed (fc) 1: Low-speed (fs) 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.

6.3 Clock control

6.3.1 Clock System Block 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[2:0]>
fsys	: Clock specified by CGCKSEL<SYSCK> (system clock)
fperiph	: Clock specified by CGSYSCR<FPSEL>
φT0	: Clock specified by CGSYSCR<PRCK[2:0]> (prescaler clock)

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

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

CPU uses the following clocks. HCLK and FCLK stop in the low power consumption mode (IDLE,SLEEP,STOP.)

HCLK,FCLK	: fsys
STCLK (Systick timer)	: fosc/32

6.3.2 Initial Values after Reset

Reset operation initializes the clock configuration as follows.

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

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

fc = fosc
fsys = fosc
φT0 = fosc

For example, reset operation configures fsys as 10MHz when a 10MHz oscillator is connected to the X1 or X2 pin.

6.3.3 Clock system Diagram

Figure 6-1 shows the clock system diagram.

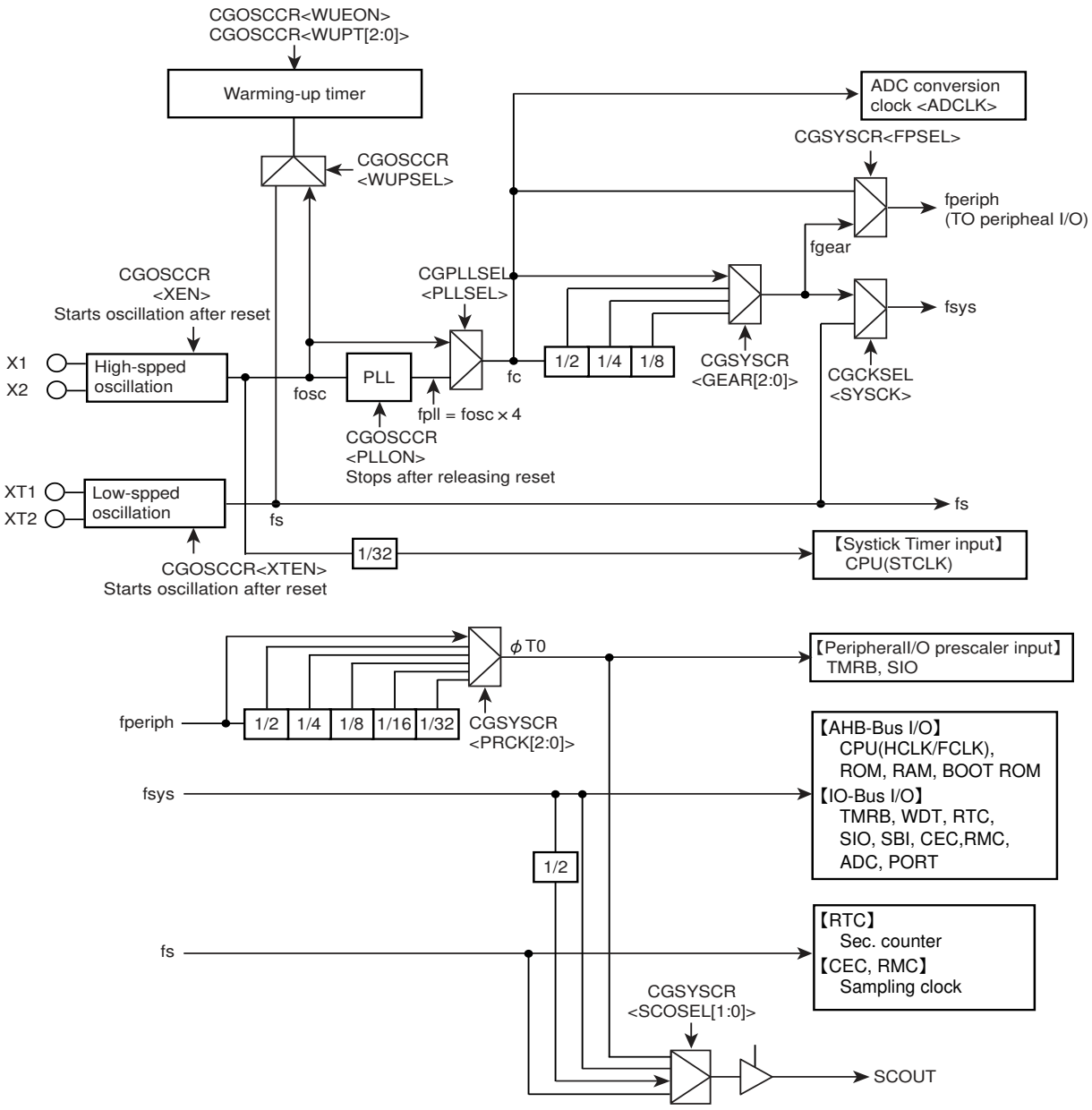


Figure 6-1 Clock Block Diagram

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

6.3.4 Clock Multiplication Circuit (PLL)

This circuit outputs the fpll clock that is quadruple of the high-speed oscillator output clock (fosc.) As a result, the input frequency to oscillator can be low, and the internal clock be made high-speed.

The PLL is disabled after reset. 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: **It takes approximately 200μs for the PLL to be stabilized.**

6.3.5 Warm-up function

The warm-up function secures the stability time for the oscillator and the PLL with the warm-up timer.

The warm-up function is used when returning from STOP/SLEEP mode.

In this case, an interrupt for returning from the low power consumption mode triggers the automatic timer count. After the specified time is reached, the system clock is output and the CPU starts operation.

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.

How to configure the warm-up function.

Specify the count up clock for the warm-up counter in the CGOSCCR<WUPSEL> bit.

The warm-up time can be selected by setting the CGOSCCR<WUPT[2:0]>.

The CGOSCCR<WUEON><WUEF> is used to confirm the start and completion of warm-up through software (instruction). After the completion 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>

Table 6-1 shows the warm-up time.

Table 6-1 Warm-up Time (fosc = 10MHz, fs = 32.768kHz)

Warm-up time options CGOSCCR<WUPT[2:0]>	High-speed clock (fosc) CGOSCCR<WUPSEL> = "0"		Low-speed clock (fs) CGOSCCR<WUPSEL> = "1"	
000	−	Without WUP	−	With WUP
001	2^{10}/input frequency	102.4 (μs)	2^{6}/input frequency	1.953 (ms)
010	2^{11}/input frequency	204.8 (μs)	2^{7}/input frequency	3.906 (ms)
011	2^{12}/input frequency	409.6 (μs)	2^{8}/input frequency	7.813 (ms)
100	2^{13}/input frequency	819.2 (μs)	2^{15}/input frequency	1.0 (s)
101	2^{14}/input frequency	1.638 (ms)	2^{16}/input frequency	2.0 (s)
110	2^{15}/input frequency	3.277 (ms)	2^{17}/input frequency	4.0 (s)
111	2^{16}/input frequency	6.554 (ms)	2^{18}/input frequency	8.0 (s)

Note: **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.**

The following are the examples of the warm-up function configuration.

<Example1> Securing the stability time for the PLL

CGOSCCR<WUPSEL> = "0"	: Specify the warm-up counter
CGOSCCR<WUPT[2:0]> = "010"	: Specify the warm-up time (204.8μs)
CGOSCCR<WUEON> = "1"	: Start the warm-up timer (WUP)
CGOSCCR<WUEF> Read	: Wait until the state becomes "0" (warm-up is finished)

<Example2> Transition from the NORMAL mode to the SLOW mode

CGOSCCR<WUPSEL> = "1"	: Specify the warm-up counter
CGOSCCR<WUPT[2:0]> = "xxx"	: Specify the warm-up time
CGOSCCR<XTEN> = "1"	: Enable the low-speed oscillation (fs)
CGOSCCR<WUEON> = "1"	: Start the warm-up timer
CGOSCCR<WUEF> Read	: Wait until the state becomes "0" (warm-up is finished)
CGCKSEL<SYSCK> = "1"	: Switch the system clock to low speed (fs)
CGCKSEL<SYSCKFLG> Read	: Confirm that the current state is "1" (the current system clock is fs)
CGOSCCR<XEN> = "0"	: Disable the high-speed oscillation (fosc)

<Example3> Transition from the SLOW mode to the NORMAL mode

CGOSCCR<WUPSEL> = "0"	: Specify the warm-up counter
CGOSCCR<WUPT[2:0]> = "xxx"	: Specify the warm-up time
CGOSCCR<XEN> = "1"	: Enable the high-speed oscillation (fosc)
CGOSCCR<WUEON> = "1"	:Start the warm-up timer
CGOSCCR<WUEF> Read	: Wait until the state becomes "0" (warm-up is finished)
CGCKSEL<SYSCK> = "0"	: Switch the system clock to high speed (fgear)
CGCKSEL<SYSCKFLG> Read	: Confirm that the current state is "0" (the current system clock is fgear)
CGOSCCR<XTEN> = "0"	: Disable the low-speed oscillation (fs)

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

6.3.6 System Clock

The TMPM330FDFG/FYFG/FWFG offers two selectable system clocks: low-speed or high-speed. The high-speed clock is dividable.

Note 1: **Switching of clock gear is executed when a value is written to the CGSYSCR<GEAR[2:0]> register. The actual switching takes place after a slight delay.**

Note 2: The CEC function uses the low-speed clock as a sampling clock. The allowable margin of error when the CEC function is used is approximately ±4% at 32.768 kHz.

6.3.6.1 High speed clock

- Input frequency from X1 and X2: 8MHz to 10MHz
- Allows for oscillator connection or external clock input
- Clock gear:1/1, 1/2, 1/4, 1/8 (after reset: 1/1)

Table 6-2 Range of High Speed frequency

Input freq. X1, 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
8MHz	1 MHz	40 MHz	8	32	16	8	4	8	4	2	1
10MHz			10	40	20	10	5	10	5	2.5	1.25

Note: **PLL=ON/OFF setting: available in CGOSCCR<PLLON> Clock gear setting: available in CGSYSCR<GEAR[2:0]>.**

6.3.6.2 Low speed clock

Input frequency from XT1 and XT2

Table 6-3 Range of Low Speed Frequency

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

6.3.7 Prescaler Clock Control

Each peripheral function 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[2:0]>. 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.**

6.3.8 System Clock Pin Output Function

The TX03 enables to output the system clock from a pin. The PK1/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 PKCR<PK1C> and PKFR1<PK1F1> to "1", the PK1/SCOUT pin (pin number 51) becomes the SCOUT output pin. The output clock is selected by setting the CGSYSCR<SCOSEL[1:0]>.

Table 6-4 shows the pin status in each mode when the SCOUT pin is set to the SCOUT output.

Table 6-4 SCOUT Output Status in Each Mode

Mode / SCOUT selection CGSYSCR	NORMAL	SLOW	Low power consumption mode		
			IDLE	SLEEP	STOP
<SCOSEL[1:0]> = "00"	Output the fs clock				
<SCOSEL[1:0]> = "01"	Output the fsys/2 clock				
<SCOSEL[1:0]> = "10"	Output the fsys clock				
<SCOSEL[1:0]> = "11"	Output the φT0 clock	Fixed to "0" or "1".			

Note: **The phase difference (AC timing) between the system clock output by the SCOUT and the internal clock is not guaranteed.**

6.4 Modes and Mode Transitions

6.4.1 Mode Transitions

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

The IDLE, 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.

Figure 6-2 shows a mode transition diagram.

For a detail of sleep-on-exit, refer to "Cortex-M3 Technical Reference Manual."

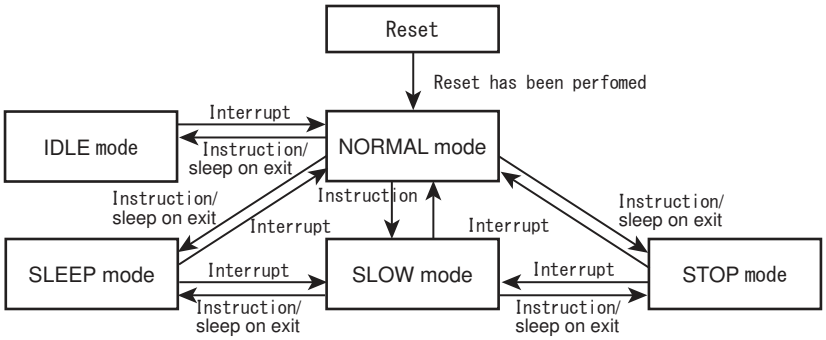


Figure 6-2 Mode Transition Diagram

6.5 Operation mode

Two operation modes, NORMAL and SLOW, are available. The features of each mode are described in the following section.

6.5.1 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.

6.5.2 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, real-time clock (RTC), CEC and remote control signal preprocessor (RMC).

Note 1: **Be sure to stop peripheral functions except for the CPU, RTC, I/O ports, CEC and RMC before switching to the SLOW mode.**

Note 2: **In the slow mode, be sure not to perform reset using the Application Interrupt and Reset Control Register <SYSRESETREQ> of the Cortex-M3 NVIC register.**

6.6 Low Power Consumption Modes

The TX03 has three low power consumption modes: IDLE, SLEEP and STOP. To shift to the low power consumption mode, specify the mode in the system control register CGSTBYCR<STBY[2:0]> 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 the chapter "Exceptions" for details.

Note 1: **The TX03 does not offer any event for releasing the low power consumption mode. Transition to the low power consumption mode by executing the WFE (Wait For Event) instruction is prohibited.**

Note 2: **The TX03 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.

6.6.1 IDLE mode

Only 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/UART)
- Serial bus interface (I2C/SIO)
- CEC
- Remote control signal preprocessor (RMC)
- Analog Digital converter (ADC)
- Watch dog timer (WDT)

6.6.2 SLEEP mode

In the SLEEP mode, the internal low-speed oscillator, real time clock, CEC and RMC can be operated.

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

Note: **When PA1 (pin number 56) is configured as a debug function pin, it prevents the low power consumption mode from being fully effective. Configure PA1 to function as a general-purpose port if the debug function is not used.**

6.6.3 STOP mode

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

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 6-5 shows the pin status in the STOP mode.

Note: **When PA1 (pin number 56) is configured as a debug function pin, it prevents the low power consumption mode from being fully effective. Configure PA1 to function as a general-purpose port when the debug function is not used.**

Table 6-5 Pin States in the STOP Mode

	Pin Name	I/O	<DRVE> = 0	<DRVE> = 1
Not Ports	X1, XT1	Input only	×	×
	X2, XT2	Output only	"High" level output.	"High" level output.
	$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, MODE	Input only	o	o
Ports	PA0, PB0 [When used as a debug pin (PxFRn<PxmFn>=1) and output is enabled (PxCR<PxmC>=1)] (Note)	Input	×	Depends on (PxIE[m])
		Output	Enabled when data is valid. Disabled when data is invalid.	
	PF7, PG3, PJ0, PJ1, PJ2, PJ3, PJ6, PJ7 [When used as an interrupt pin (PxFRn<PxmFn>=1) and input is enabled (PxIE<PxmIE>=1)] (Note)	Input	o	o
		Output	×	Depends on (PxCR[m])
	other port pins	Input	×	Depends on (PxIE[m])
		Output	×	Depends on (PxCR[m])

o : Input or output enabled.

× : Input or output disabled.

Note: **x: port number / m: corresponding bit / n: function register number**

6.6.4 Low power Consumption Mode Setting

The low power consumption mode is specified by the setting of the standby control register CGSTBYCR<STBY[2:0]>.

Table 6-6 shows the mode setting in the <STBY[2:0]>.

Table 6-6 Low power consumption mode setting

Mode	CGSTBYCR <STBY[2:0]>
STOP	001
SLEEP	010
IDLE	011

Note: **Do not set any value other than those shown above in <STBY[2:0]>.**

6.6.5 Operational Status in Each Mode

Table 6-7 show the operational status in each mode.

For I/O port, "o" and "×" indicate that input/output is enabled and disabled respectively.

For other functions, "o" and "×" indicate that clock is supplied and is not supplied respectively.

Table 6-7 Operational Status in Each Mode

Block	NORMAL	SLOW	IDLE	SLEEP	STOP
Processor core	o	o	×	×	×
I/O port	o	o	o	o	* (Note 3)
ADC	o	× (Note 1)	ON/OFF selectable for each module	×	×
SIO	o	× (Note 1)		×	×
SBI	o	× (Note 1)		×	×
TMRB	o	× (Note 1)		×	×
WDT	o	× (Note 1)		×	×
CEC	o	o	o	o	×
RMC	o	o	o	o	×
RTC	o	o	o	o	×
CG	o	o	o	o	×
PLL	o	×	o	×	×
High-speed oscillator (fc)	o	* (Note 2)	o	×	×
Low-speed oscillator (fs)	o	o	o	o	×

Note 1: **In the SLOW mode, the ADC, SIO, SBI, TMRB and WDT cannot be used and must be stopped.**

Note 2: **The high-speed oscillator does not stop automatically and must be stopped by setting the CGOSCCR<XEN> bit.**

Note 3: **The status depends on the CGSTBYCR<DRVE> bit.**

6.6.6 Releasing the Low Power Consumption Mode

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

Details are shown in Table 6-8.

Table 6-8 Release Source in Each Mode

Low power consumption mode			IDLE	SLEEP	STOP
Release source	Interrupt	INT0 to 7 (Note1)	o	o	o
		INTRTC	o	o	×
		INTTB0 to 9	o	×	×
		INTCAP00 to 60, 01 to 61	o	×	×
		INTRX0 to 2, INTTX0 to 2	o	×	×
		INTSBI0 to 2	o	×	×
		INTCECRX, INTCECTX	o	o	×
		INTRMCRX0, 1	o	o	×
		INTAD/INTADHP/INTADM0, 1	o	×	×
	NMI (INTWDT)		o	×	×
	NMI ($\overline{\text{NMI}}$ pin)		o	o	o
	RESET ($\overline{\text{RESET}}$ pin)		o	o	o

o : Starts the interrupt handling after the mode is released. (The reset initializes the LSI)

× : Unavailable

Note 1: **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 2: **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.**

- 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 Non-Maskable Interrupt (NMI)

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

- Release by reset

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

  Note that returning to the STOP 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.

6.6.7 Warm-up

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

In the mode transition from STOP to the NORMAL/ SLOW or from 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 a oscillator to be used for warm-up in the CGOSCCR<WUPSEL> and to set a warm-up time in the CGOSCCR<WUPT[2:0]> before executing the instruction to enter the STOP/ 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 approximately 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 6-9 shows whether the warm-up setting of each mode transition is required or not.

Table 6-9 Warm-up setting in mode transition

Mode transition	Warm-up setting
NORMAL → IDLE	Not required
NORMAL → SLEEP	(Note1)
NORMAL → SLOW	(Note 1)
NORMAL → STOP	Not required
IDLE → NORMAL	Not required
SLEEP → NORMAL	Auto-warm-up
SLEEP → SLOW	Not required
SLOW → NORMAL	(Note 2)
SLOW → SLEEP	Not required
SLOW → STOP	Not required
STOP → NORMAL	Auto-warm-up (Note 3)
STOP → SLOW	Auto-warm-up

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: **Returning to NORMAL mode by reset does not induce the automatic warm-up. Keep the reset signal valid until the oscillator operation becomes stable.**

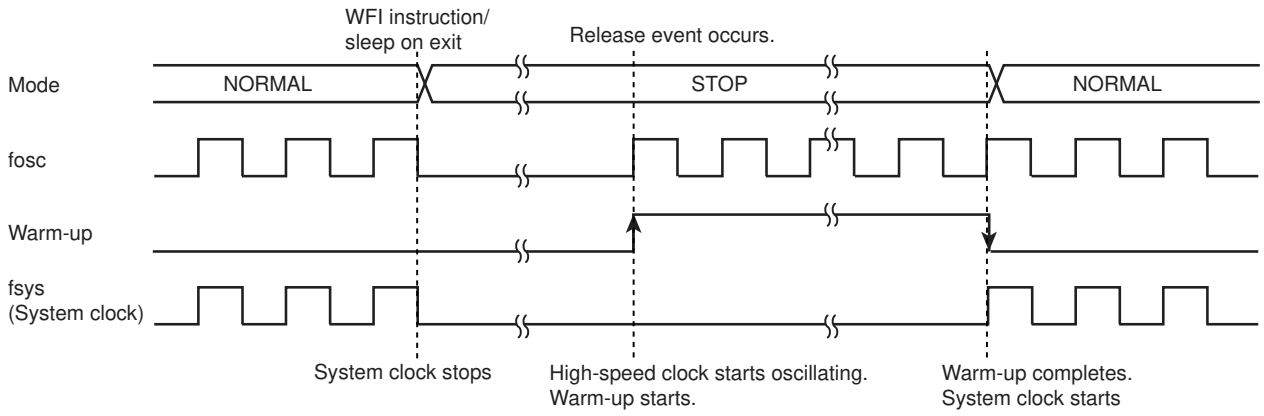
6.6.8 Clock Operations in Mode Transition

The clock operations in mode transition are described in Chapter 6.6.8.1 to 6.6.8.4.

6.6.8.1 Transition of operation modes: NORMAL → STOP → NORMAL

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

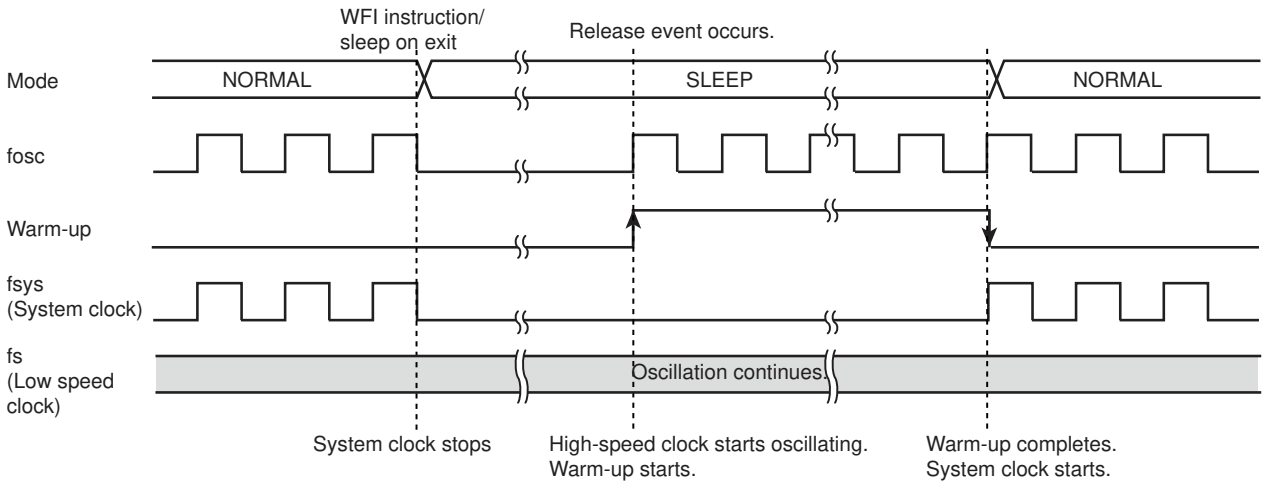
Returning to the NORMAL mode by reset does not induce the automatic warm-up. Keep the reset signal asserted until the oscillator operation becomes stable.



6.6.8.2 Transition of operation modes: NORMAL → SLEEP → NORMAL

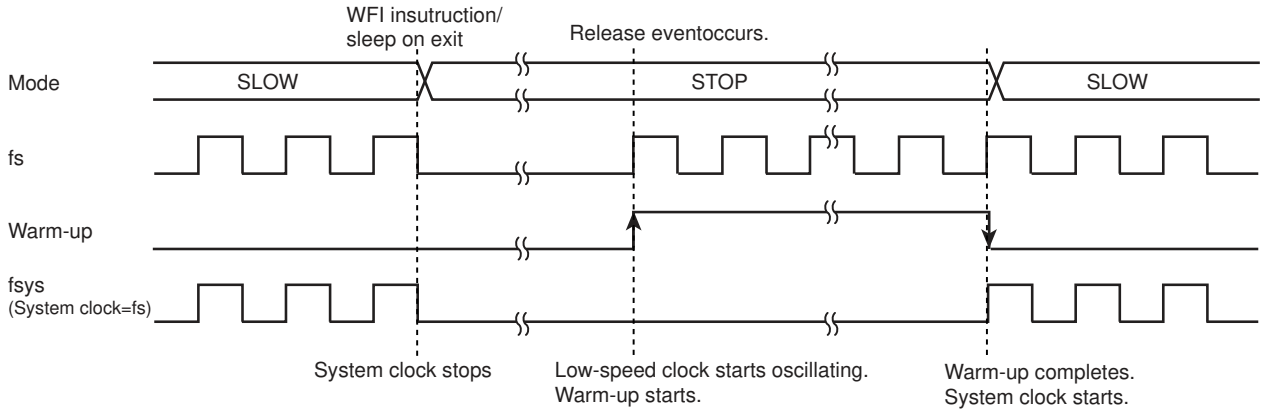
When returning to the NORMAL mode from the SLEEP mode, the warm-up is activated automatically. It is necessary to set the warm-up time before entering the SLEEP mode.

Returning to the NORMAL mode by reset does not induce the automatic warm-up. Keep the reset signal asserted until the oscillator operation becomes stable.



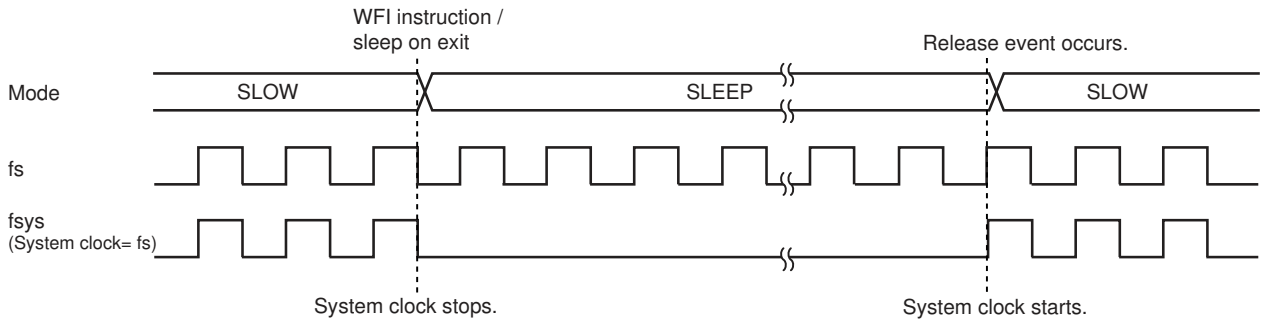
6.6.8.3 Transition of operation modes: SLOW → STOP → SLOW

The warm-up is activated automatically. It is necessary to set the warm-up time before entering the STOP mode.



6.6.8.4 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.



7. 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.

7.1 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.

7.1.1 Exception Types

The following types of exceptions exist in the Cortex-M3.

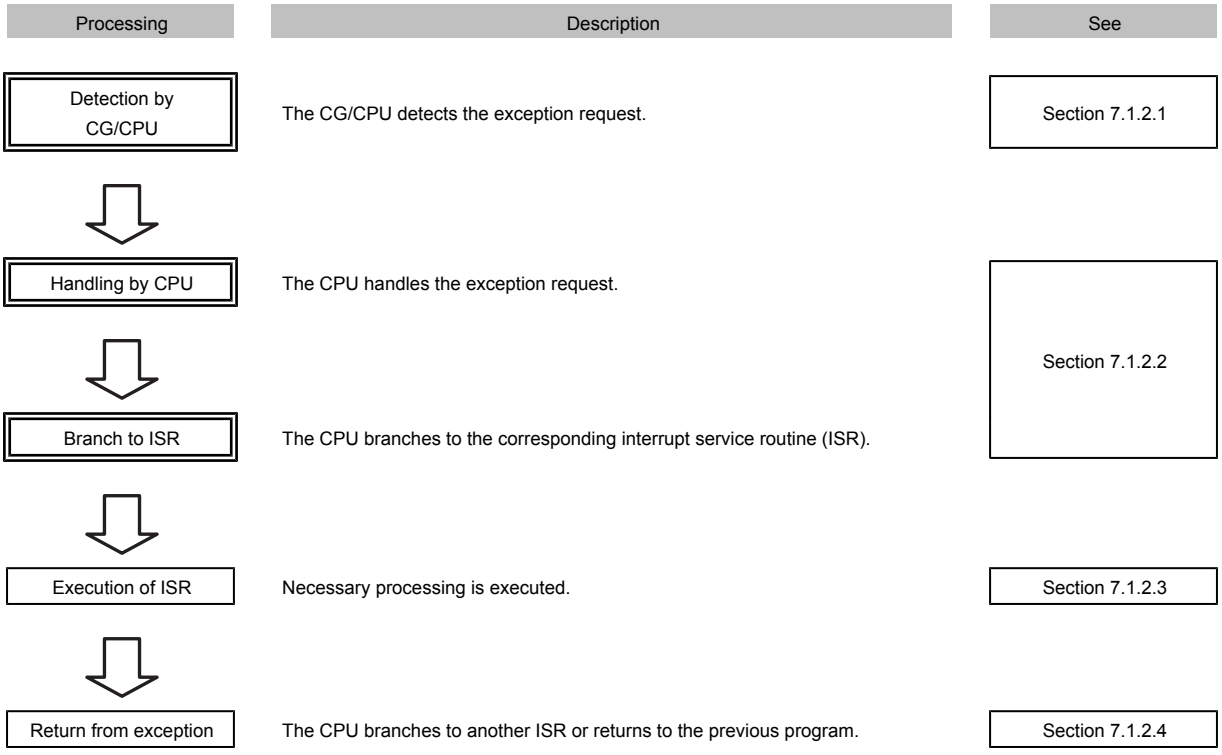
For detailed descriptions on each exception, refer to "Cortex-M3 Technical Reference Manual".

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

7.1.2 Handling Flowchart

The following shows how an exception/interrupt is handled. In the following descriptions, indicates hardware handling. Indicates software handling.

Each step is described later in this chapter.



7.1.2.1 Exception 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 (XN) region or an access violation to the Fault region.

An interrupt request 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 "7.5 Interrupts".

(2) Exception detection

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

Table 7-1 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 7-1 Exception Types and Priority

No.	Exception type	Priority	Description
1	Reset	−3 (highest)	Reset pin, WDT or SYSRETREQ
2	Non-Maskable Interrupt	−2	$\overline{\text{NMI}}$ 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 CPU is not faulting
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 "7.5.1.5 List of Interrupt Sources".**

(3) Priority setting

- Priority levels

  The external interrupt priority is set to the interrupt priority register and other exceptions are set to <PRI_n> bit in the system handler priority register.

  The configuration <PRI_n> can be changed, and the number of bits required for setting the priority varies from 3 bits to 8 bits depending on products. Thus, the range of priority values you can specify is different depending on products.

  In the case of 8-bit configuration, the priority can be configured in the range from 0 to 255. The highest priority is "0". If multiple elements with the same priority exist, the smaller the number, the higher the priority becomes.

Note: **<PRI_n> bit is defined as a 3-bit configuration with this product.**

- Priority grouping

  The priority group can be split into groups. By setting the <PRIGROUP> of the application interrupt and reset control register, <PRI_n> can be divided into the pre-emption priority and the sub priority.

  A priority is compared with the pre-emption priority. If the priority is the same as the pre-emption priority, then it is compared with the sub priority. If the sub priority is the same as the priority, the smaller the exception number, the higher the priority.

The Table 7-2 shows the priority group setting. The pre-emption priority and the sub priority in the table are the number in the case that <PRI_n> is defined as an 8-bit configuration.

Table 7-2 Priority grouping setting

<PRIGROUP[2:0]> setting	<PRI_n[7:0]>		Number of pre-emption priorities	Number of subpriorities
	Pre-emption field	Subpriority field		
000	[7:1]	[0]	128	2
001	[7:2]	[1:0]	64	4
010	[7:3]	[2:0]	32	8
011	[7:4]	[3:0]	16	16
100	[7:5]	[4:0]	8	32
101	[7:6]	[5:0]	4	64
110	[7]	[6:0]	2	128
111	None	[7:0]	1	256

Note: **If the configuration of <PRI_n> is less than 8 bits, the lower bit is "0". For the example, in the case of 3-bit configuration, the priority is set as <PRI_n[7:5]> and <PRI_n[4:0]> is "00000".**

7.1.2.2 Exception Handling and Branch to the Interrupt Service Routine (Pre-emption)

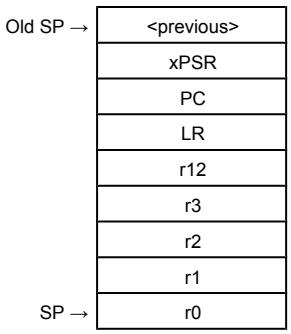
When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called "pre-emption".

(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.



(2) Fetching an ISR

The CPU enables instruction to fetch the interrupt processing with data store to the register.

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 area.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 contents 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 Management	ISR address	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

7.1.2.3 Executing 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 "7.5 Interrupts".

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.

7.1.2.4 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

  If there are no pending or stacked exceptions, the CPU returns to the previous program.

(2) Exception 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.

7.2 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 "Low" to "High".

- 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.

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

7.3 Non-Maskable Interrupts (NMI)

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 $\overline{\text{NMI}}$ pin

  A non-maskable interrupt is generated when an external $\overline{\text{NMI}}$ pin changes from "High" to "Low".

- Non-maskable interrupt by WDT

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

7.4 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 be pending 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.**

7.5 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.

7.5.1 Interrupt Sources

7.5.1.1 Interrupt Route

Figure 7-1 shows an interrupt request route.

The interrupts issued by the peripheral function that is not used to release standby are directly input to the CPU (route1).

The peripheral function interrupts used to release standby (route 2) and interrupts from the external interrupt pin (route 3) are input to the clock generator and are input to the CPU through the logic for releasing standby (route 4 and 5).

If interrupts from the external interrupt pins are not used to release standby, they are directly input to the CPU, not through the logic for standby release (route 6).

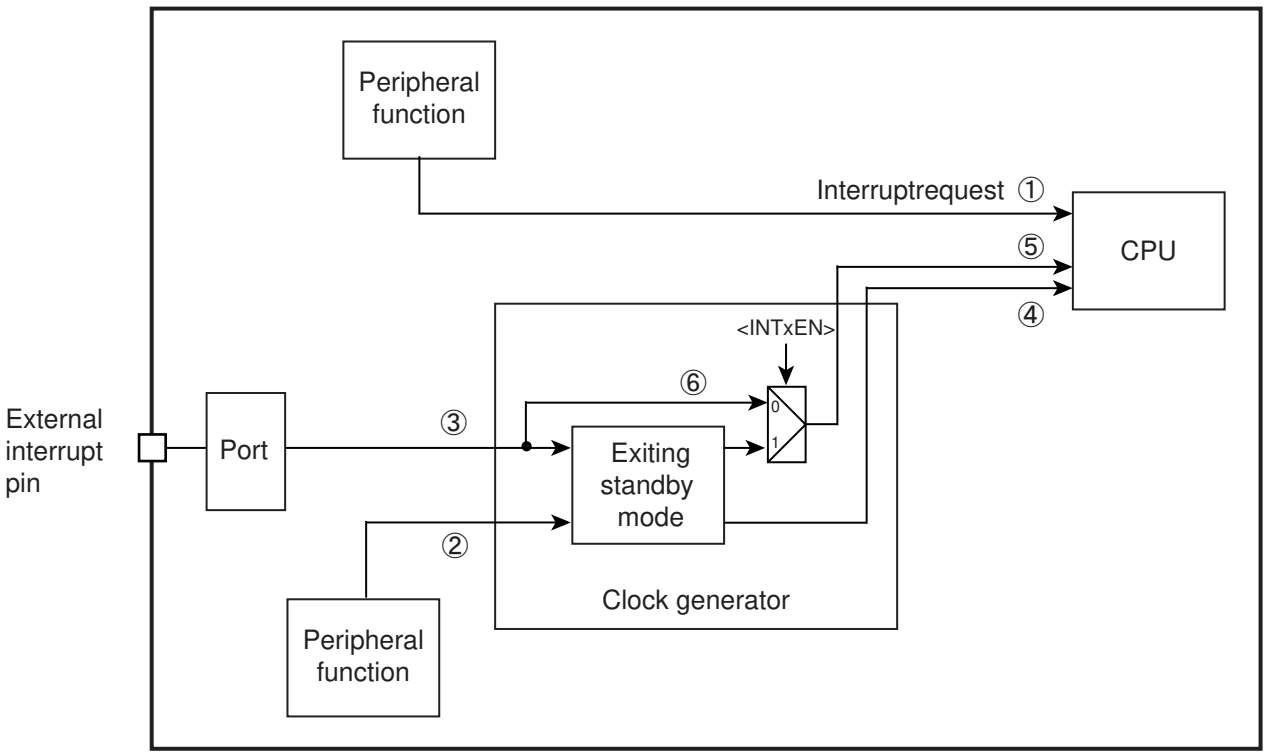


Figure 7-1 Interrupt Route

7.5.1.2 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.

7.5.1.3 Transmission

An interrupt signal from an external pin or peripheral function is directly sent to the CPU unless it is used to exit 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 exiting a standby mode can be used without setting the clock generator.

7.5.1.4 Precautions when using external interrupt pins

If you use external interrupts, be aware the followings not to generate unexpected interrupts.

If input disabled (PxIE<PxmIE>="0"), inputs from external interrupt pins are "High". Also, if external interrupts are not used as a trigger to release standby (route 6 of "Figure 7-1 Interrupt Route"), input signals from the external interrupt pins are directly sent to the CPU. Since the CPU recognizes "High" input as an interrupt, interrupts occur if corresponding interrupts are enabled by the CPU as inputs are being disabled.

To use the external interrupt without setting it as a standby trigger, set the interrupt pin input as "Low" and enable it. Then, enable interrupts on the CPU.

7.5.1.5 List of Interrupt Sources

Table 7-3 shows the list of interrupt sources.

Table 7-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
0	INT0	Interrupt pin (PJ0/70pin)	Selectable	CGIMCGA
1	INT1	Interrupt pin (PJ1/49pin)		
2	INT2	Interrupt pin (PJ2/86pin)		
3	INT3	Interrupt pin (PJ3/87pin)		
4	INT4	Interrupt pin (PG3/6pin)		CGIMCGB
5	INT5	Interrupt pin (PF7/19pin)		
6	INTRX0	Serial reception (channel.0)		
7	INTTX0	Serial transmission (channel.0)		
8	INTRX1	Serial reception (channel.1)		
9	INTTX1	Serial transmission (channel.1)		
10	INTSBI0	Serial bus interface 0		
11	INTSBI1	Serial bus interface 1		
12	INTCECRX	CEC reception	Rising edge	CGIMCGB
13	INTCECTX	CEC transmission		CGIMCGD
14	INTRMCRX0	Remote control signal reception (channel.0)		CGIMCGB
15	INTADHP	Highest priority AD conversion complete interrupt		
16	INTADM0	AD conversion monitoring function interrupt 0		
17	INTADM1	AD conversion monitoring function interrupt 1		
18	INTTB0	16-bit TMRB match detection 0		
19	INTTB1	16-bit TMRB match detection 1		
20	INTTB2	16-bit TMRB match detection 2		
21	INTTB3	16-bit TMRB match detection 3		
22	INTTB4	16-bit TMRB match detection 4		
23	INTTB5	16-bit TMRB match detection 5		
24	INTTB6	16-bit TMRB match detection 6		
25	INTRTC	Real time clock	Falling edge	CGIMCGC
26	INTCAP00	16-bit TMRB input capture 00		
27	INTCAP01	16-bit TMRB input capture 01		
28	INTCAP10	16-bit TMRB input capture 10		
29	INTCAP11	16-bit TMRB input capture 11		
30	INTCAP50	16-bit TMRB input capture 50		
31	INTCAP51	16-bit TMRB input capture 51		
32	INTCAP60	16-bit TMRB input capture 60		
33	INTCAP61	16-bit TMRB input capture 61		
34	INT6	Interrupt pin (PJ6/39pin)	Selectable	CGIMCGC
35	INT7	Interrupt pin (PJ7/58pin)		
36	INTRX2	Serial reception (channel.2)		
37	INTTX2	Serial transmission (channel.2)		
38	INTSBI2	Serial bus interface 2		
39	INTRMCRX1	Remote control signal reception (channel.1)	Rising edge	CGIMCGC

Table 7-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
40	INTTB7	16-bit TMRB match detection 7		
41	INTTB8	16-bit TMRB match detection 8		
42	INTTB9	16-bit TMRB match detection 9		
43	INTCAP20	16-bit TMRB input capture 20		
44	INTCAP21	16-bit TMRB input capture 21		
45	INTCAP30	16-bit TMRB input capture 30		
46	INTCAP31	16-bit TMRB input capture 31		
47	INTCAP40	16-bit TMRB input capture 40		
48	INTCAP41	16-bit TMRB input capture 41		
49	INTAD	A/D conversion completion		

7.5.1.6 Active level

The active level indicates which change in signal of an interrupt source triggers an interrupt. The CPU recognizes interrupt signals in "High" level as interrupt. Interrupt signals directly sent from peripheral functions to the CPU are configured to output "High" to indicate an interrupt request.

Active level is set to the clock generator for interrupts which can be a trigger to release standby. 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 ("High" or "Low") 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 level in the CGIMCGx<EMCGx> bits. You must set the active level for interrupt requests from each peripheral function as shown in Table 7-3.

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

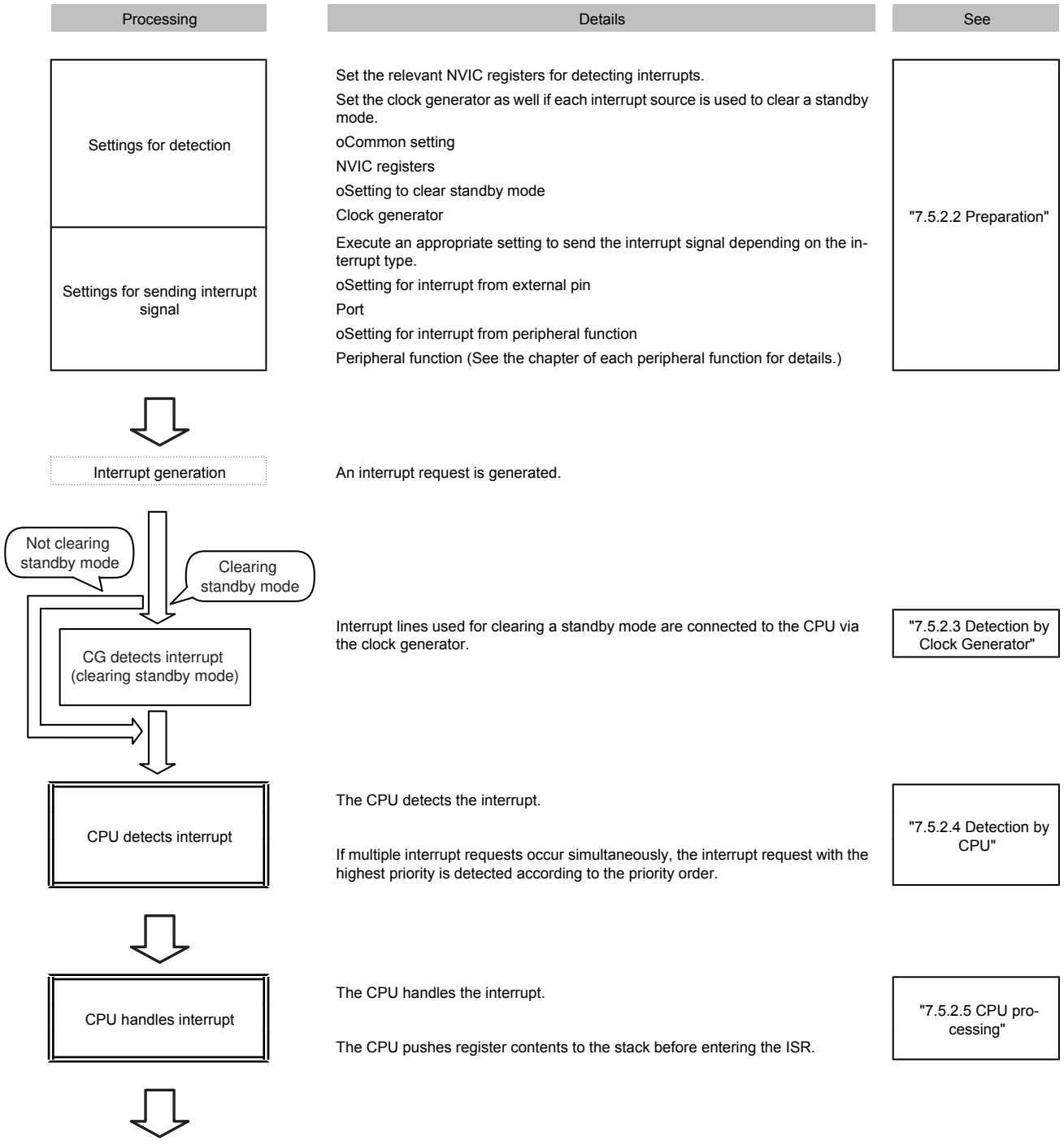
Note: **For the CEC reception/transmission, remote control signal reception and real time clock interrupts, set the <INTxEN> bit to "1" and specify the active level, even when they are not used for clearing a standby mode.**

7.5.2 Interrupt Handling

7.5.2.1 Flowchart

The following shows how an interrupt is handled.

In the following descriptions, ▭ indicates hardware handling. ▭ indicates software handling.



Processing	Details	See
ISR execution	Program for the ISR. Clear the interrupt source if needed.	"7.5.2.6 Interrupt Service Routine (ISR)"
Return to preceding program	Configure to return to the preceding program of the ISR.	

7.5.2.2 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 PRIMASK Register. All interrupts and exceptions other than non-maskable interrupts and hard faults can be masked.

Use "MSR" instruction to set this register.

Interrupt mask register		
PRIMASK	←	"1" (interrupt disabled)

Note 1: PRIMASK register cannot be modified by the user access level.
Note 2: **If a fault causes when "1" is set to the PRIMASK register, it is treated as a hard fault.**

(2) CPU registers setting

You can assign a priority level by writing to <PRI_n> field in an Interrupt Priority Register of the NVIC register.

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.

You can assign grouping priority by using the PRIGROUP field in the Application Interrupt and Reset Control Register.

NVIC register		
<PRI_n>	←	"priority"
<PRIGROUP>	←	"group priority"(This is configurable if required.)

Note: **"n" indicates the corresponding exceptions/interrupts.**
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 PxFRn[m] allows the pin to be used as the function pin. Setting PxIE[m] allows the pin to be used as the input port.

Port register		
PxFRn<PxmFn>	←	"1"
PxIE<PxmIE>	←	"1"

Note: **x: port number / m: corresponding bit / n: function register number**
In modes other than STOP mode, setting PxIE to enable input enables the corresponding interrupt input regardless of the PxFR setting. Be careful not to enable interrupts that are not used. Also, be aware of the description of "7.5.1.4 Precautions when using external interrupt pins".

(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 exiting a standby mode, you need to set the active level 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 "7.6.3.5 CGICRCG(CG Interrupt Request Clear Register)" for each value.

Interrupt requests from external pins can be used without setting the clock generator if they are not used for exiting a standby mode. However, an "High" pulse or "High"-level signal must be input so that the CPU can detect it as an interrupt request. Also, be aware of the description of"7.5.1.4 Precautions when using external interrupt pins".

Clock generator register		
CGIMCGn<EMCGm>	←	active level
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 Interrupt Clear-Pending Register clears the suspended interrupt. Writing "1" to the corresponding bit of the Interrupt Set-Enable Register enables the intended interrupt.

To generate interrupts in the Interrupt Set-Pending Register setting, factors to trigger interrupts are lost if pending interrupts are cleared. Thus, this operation is not necessary.

At the end, PRIMASK register is zero cleared.

NVIC register		
Interrupt Clear-Pending [m]	←	"1"
Interrupt Set-Enable [m]	←	"1"
Interrupt mask register		
PRIMASK	←	"0"

Note 1: **m : corresponding bit**
Note 2: **PRIMASK register cannot be modified by the user access level.**

7.5.2.3 Detection by Clock Generator

If an interrupt source is used for exiting 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 "High" 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.

7.5.2.4 Detection by CPU

The CPU detects an interrupt request with the highest priority.

7.5.2.5 CPU processing

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

7.5.2.6 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.

7.6 Exception/Interrupt-Related Registers

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

7.6.1 Register List

NVIC registers — Base Address = 0xE000_E000

Register name	Address
SysTick Control and Status Register	0x0010
SysTick Reload Value Register	0x0014
SysTick Current Value Register	0x0018
SysTick Calibration Value Register	0x001C
Interrupt Set-Enable Register 1	0x0100
Interrupt Set-Enable Register 2	0x0104
Interrupt Clear-Enable Register 1	0x0180
Interrupt Clear-Enable Register 2	0x0184
Interrupt Set-Pending Register 1	0x0200
Interrupt Set-Pending Register 2	0x0204
Interrupt Clear-Pending Register 1	0x0280
Interrupt Clear-Pending Register 2	0x0284
Interrupt Priority Register	0x0400 ~ 0x0430
Vector Table Offset Register	0x0D08
Application Interrupt and Reset Control Register	0x0D0C
System Handler Priority Register	0x0D18, 0x0D1C, 0x0D20
System Handler Control and State Register	0x0D24

Clock generator registers — Base Address = 0x4004_0200

Register name		Address
CG Interrupt Request Clear Register	CGICRCG	0x0014
NMI Flag Register	CGNMIFLG	0x0018
Reset Flag Register	CGRSTFLG	0x001C
CG Interrupt Mode Control Register A	CGIMCGA	0x0020
CG Interrupt Mode Control Register B	CGIMCGB	0x0024
CG Interrupt Mode Control Register C	CGIMCGC	0x0028
CG Interrupt Mode Control Register D	CGIMCGD	0x002C
Reserved	-	0x0030
Reserved	-	0x0034
Reserved	-	0x0038
Reserved	-	0x003C

Note: **Access to the "Reserved" areas is prohibited.**

7.6.2 NVIC Registers

7.6.2.1 SysTick Control and Status Register

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	COUNTFLAG
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	CLKSOURCE	TICKINT	ENABLE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-17	−	R	Read as 0.
16	COUNTFLAG	R/W	0: Timer not counted to 0 1: Timer counted to 0 Returns "1" if timer counted to "0" since last time this was read. Clears on read of any part of the SysTick Control and Status Register.
15-3	−	R	Read as 0.
2	CLKSOURCE	R/W	0: External reference clock 1: CPU clock
1	TICKINT	R/W	0: Do not pend SysTick 1: Pend SysTick
0	ENABLE	R/W	0: Disable 1: Enable If "1" is set, it reloads with the value of the Reload Value Register and starts operation.

7.6.2.2 SysTick Reload Value Register

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	RELOAD							
After reset	Undefined							
	15	14	13	12	11	10	9	8
bit symbol	RELOAD							
After reset	Undefined							
	7	6	5	4	3	2	1	0
bit symbol	RELOAD							
After reset	Undefined							

Bit	Bit Symbol	Type	Function
31-24	−	R	Read as 0.
23-0	RELOAD	R/W	Reload value 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.**

7.6.2.3 SysTick Current Value Register

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	CURRENT							
After reset	Undefined							
	15	14	13	12	11	10	9	8
bit symbol	CURRENT							
After reset	Undefined							
	7	6	5	4	3	2	1	0
bit symbol	CURRENT							
After reset	Undefined							

Bit	Bit Symbol	Type	Function
31-24	−	R	Read as 0.
23-0	CURRENT	R/W	[Read] Current SysTick timer value [Write] Clear 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.

7.6.2.4 SysTick Calibration Value Register

	31	30	29	28	27	26	25	24
bit symbol	NOREF	SKEW	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TENMS							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TENMS							
After reset	0	0	0	0	1	0	0	1
	7	6	5	4	3	2	1	0
bit symbol	TENMS							
After reset	1	1	0	0	0	1	0	0

Bit	Bit Symbol	Type	Function
31	NOREF	R	0: Reference clock provided 1: No reference clock
30	SKEW	R	0: Calibration value is 10 ms. 1: Calibration value is not 10 ms.
29-24	−	R	Read as 0.
23-0	TENMS	R	Calibration value Reload value to use for 10 ms timing (0x9C4). (Note)

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.**

7.6.2.5 Interrupt Set-Enable Register 1

	31	30	29	28	27	26	25	24
bit symbol	SETENA (Interrupt 31)	SETENA (Interrupt 30)	SETENA (Interrupt 29)	SETENA (Interrupt 28)	SETENA (Interrupt 27)	SETENA (Interrupt 26)	SETENA (Interrupt 25)	SETENA (Interrupt 24)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	SETENA (Interrupt 23	SETENA (Interrupt 22)	SETENA (Interrupt 21)	SETENA (Interrupt 20)	SETENA (Interrupt 19)	SETENA (Interrupt 18)	SETENA (Interrupt 17)	SETENA (Interrupt 16)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SETENA (Interrupt 15)	SETENA (Interrupt 14)	SETENA (Interrupt 13)	SETENA (Interrupt 12)	SETENA (Interrupt 11)	SETENA (Interrupt 10)	SETENA (Interrupt 9)	SETENA (Interrupt 8)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SETENA (Interrupt 7)	SETENA (Interrupt 6)	SETENA (Interrupt 5)	SETENA (Interrupt 4)	SETENA (Interrupt 3)	SETENA (Interrupt 2)	SETENA (Interrupt 1)	SETENA (Interrupt 0)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	SETENA	R/W	Interrupt number [31:0] [Write] 1: Enable [Read] 0: Disabled 1: Enabled Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

Note: **For descriptions of interrupts and interrupt numbers, see Section "7.5.1.5 List of Interrupt Sources".**

7.6.2.6 Interrupt Set-Enable Register 2

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	SETENA (Interrupt 49)	SETENA (Interrupt 48)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SETENA (Interrupt 47)	SETENA (Interrupt 46)	SETENA (Interrupt 45)	SETENA (Interrupt 44)	SETENA (Interrupt 43)	SETENA (Interrupt 42)	SETENA (Interrupt 41)	SETENA (Interrupt 40)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SETENA (Interrupt 39)	SETENA (Interrupt 38)	SETENA (Interrupt 37)	SETENA (Interrupt 36)	SETENA (Interrupt 35)	SETENA (Interrupt 34)	SETENA (Interrupt 33)	SETENA (Interrupt 32)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-18	−	R	Read as 0.
17-0	SETENA	R/W	Interrupt number [49:32] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

Note: **For descriptions of interrupts and interrupt numbers, see Section "7.5.1.5 List of Interrupt Sources".**

7.6.2.7 Interrupt Clear-Enable Register 1

	31	30	29	28	27	26	25	24
bit symbol	CLRENA (Interrupt 31)	CLRENA (Interrupt 30)	CLRENA (Interrupt 29)	CLRENA (Interrupt 28)	CLRENA (Interrupt 27)	CLRENA (Interrupt 26)	CLRENA (Interrupt 25)	CLRENA (Interrupt 24)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	CLRENA (Interrupt 23)	CLRENA (Interrupt 22)	CLRENA (Interrupt 21)	CLRENA (Interrupt 20)	CLRENA (Interrupt 19)	CLRENA (Interrupt 18)	CLRENA (Interrupt 17)	CLRENA (Interrupt 16)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CLRENA (Interrupt 15)	CLRENA (Interrupt 14)	CLRENA (Interrupt 13)	CLRENA (Interrupt 12)	CLRENA (Interrupt 11)	CLRENA (Interrupt 10)	CLRENA (Interrupt 9)	CLRENA (Interrupt 8)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CLRENA (Interrupt 7)	CLRENA (Interrupt 6)	CLRENA (Interrupt 5)	CLRENA (Interrupt 4)	CLRENA (Interrupt 3)	CLRENA (Interrupt 2)	CLRENA (Interrupt 1)	CLRENA (Interrupt 0)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	CLRENA	R/W	Interrupt number [31:0] [Write] 1: Disabled [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. It can be performed to enable interrupts and to check if interrupts are disabled. Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

Note: **For descriptions of interrupts and interrupt numbers, see Section "7.5.1.5 List of Interrupt Sources".**

7.6.2.8 Interrupt Clear-Enable Register 2

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	CLRENA (Interrupt 49)	CLRENA (Interrupt 48)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CLRENA (Interrupt 47)	CLRENA (Interrupt 46)	CLRENA (Interrupt 45)	CLRENA (Interrupt 44)	CLRENA (Interrupt 43)	CLRENA (Interrupt 42)	CLRENA (Interrupt 41)	CLRENA (Interrupt 40)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CLRENA (Interrupt 39)	CLRENA (Interrupt 38)	CLRENA (Interrupt 37)	CLRENA (Interrupt 36)	CLRENA (Interrupt 35)	CLRENA (Interrupt 34)	CLRENA (Interrupt 33)	CLRENA (Interrupt 32)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-18	−	R	Read as 0.
17-0	CLRENA	R/W	Interrupt number [49:32] [Write] 1: Disabled [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. It can be performed to enable interrupts and to check if interrupts are disabled. Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

Note: **For descriptions of interrupts and interrupt numbers, see Section "7.5.1.5 List of Interrupt Sources".**

7.6.2.9 Interrupt Set-Pending Register 1

	31	30	29	28	27	26	25	24
bit symbol	SETPEND (Interrupt 31)	SETPEND (Interrupt 30)	SETPEND (Interrupt 29)	SETPEND (Interrupt 28)	SETPEND (Interrupt 27)	SETPEND (Interrupt 26)	SETPEND (Interrupt 25)	SETPEND (Interrupt 24)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	SETPEND (Interrupt 23)	SETPEND (Interrupt 22)	SETPEND (Interrupt 21)	SETPEND (Interrupt 20)	SETPEND (Interrupt 19)	SETPEND (Interrupt 18)	SETPEND (Interrupt 17)	SETPEND (Interrupt 16)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	SETPEND (Interrupt 15)	SETPEND (Interrupt 14)	SETPEND (Interrupt 13)	SETPEND (Interrupt 12)	SETPEND (Interrupt 11)	SETPEND (Interrupt 10)	SETPEND (Interrupt 9)	SETPEND (Interrupt 8)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	SETPEND (Interrupt 7)	SETPEND (Interrupt 6)	SETPEND (Interrupt 5)	SETPEND (Interrupt 4)	SETPEND (Interrupt 3)	SETPEND (Interrupt 2)	SETPEND (Interrupt 1)	SETPEND (Interrupt 0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-0	SETPEND	R/W	Interrupt number [31:0] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines 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 the bit returns the current state of the corresponding interrupts. Writing "1" to a corresponding bit in the Interrupt Clear-Pending Register clears the bit in this register.

Note: **For descriptions of interrupts and interrupt numbers, see Section "7.5.1.5 List of Interrupt Sources".**

7.6.2.10 Interrupt Set-Pending Register 2

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	SETPEND (Interrupt 49)	SETPEND (Interrupt 48)
After reset	0	0	0	0	0	0	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	SETPEND (Interrupt 47)	SETPEND (Interrupt 46)	SETPEND (Interrupt 45)	SETPEND (Interrupt 44)	SETPEND (Interrupt 43)	SETPEND (Interrupt 42)	SETPEND (Interrupt 41)	SETPEND (Interrupt 40)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	SETPEND (Interrupt 39)	SETPEND (Interrupt 38)	SETPEND (Interrupt 37)	SETPEND (Interrupt 36)	SETPEND (Interrupt 35)	SETPEND (Interrupt 34)	SETPEND (Interrupt 33)	SETPEND (Interrupt 32)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-18	−	R	Read as 0.
17-0	SETPEND	R/W	Interrupt number [49:32] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines 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 the bit returns the current state of the corresponding interrupts. Clear and Interrupt Set-Pending Register bit by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.

Note: **For descriptions of interrupts and interrupt numbers, see Section "7.5.1.5 List of Interrupt Sources".**

7.6.2.11 Interrupt Clear-Pending Register 1

	31	30	29	28	27	26	25	24
bit symbol	CLRPEND (Interrupt 31)	CLRPEND (Interrupt 30)	CLRPEND (Interrupt 29)	CLRPEND (Interrupt 28)	CLRPEND (Interrupt 27)	CLRPEND (Interrupt 26)	CLRPEND (Interrupt 25)	CLRPEND (Interrupt 24)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	CLRPEND (Interrupt 23)	CLRPEND (Interrupt 22)	CLRPEND (Interrupt 21)	CLRPEND (Interrupt 20)	CLRPEND (Interrupt 19)	CLRPEND (Interrupt 18)	CLRPEND (Interrupt 17)	CLRPEND (Interrupt 16)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	CLRPEND (Interrupt 15)	CLRPEND (Interrupt 14)	CLRPEND (Interrupt 13)	CLRPEND (Interrupt 12)	CLRPEND (Interrupt 11)	CLRPEND (Interrupt 10)	CLRPEND (Interrupt 9)	CLRPEND (Interrupt 8)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	CLRPEND (Interrupt 7)	CLRPEND (Interrupt 6)	CLRPEND (Interrupt 5)	CLRPEND (Interrupt 4)	CLRPEND (Interrupt 3)	CLRPEND (Interrupt 2)	CLRPEND (Interrupt 1)	CLRPEND (Interrupt 0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-0	CLRPEND	R/W	Interrupt number [31:0] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines 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 the bit returns the current state of the corresponding interrupts.

Note: **For descriptions of interrupts and interrupt numbers, see Section "7.5.1.5 List of Interrupt Sources".**

7.6.2.12 Interrupt Clear-Pending Register 2

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	CLRPEND (Interrupt 49)	CLRPEND (Interrupt 48)
After reset	0	0	0	0	0	0	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	CLRPEND (Interrupt 47)	CLRPEND (Interrupt 46)	CLRPEND (Interrupt 45)	CLRPEND (Interrupt 44)	CLRPEND (Interrupt 43)	CLRPEND (Interrupt 42)	CLRPEND (Interrupt 41)	CLRPEND (Interrupt 40)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	CLRPEND (Interrupt 39)	CLRPEND (Interrupt 38)	CLRPEND (Interrupt 37)	CLRPEND (Interrupt 36)	CLRPEND (Interrupt 35)	CLRPEND (Interrupt 34)	CLRPEND (Interrupt 33)	CLRPEND (Interrupt 32)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-18	−	R	Read as 0.
17-0	CLRPEND	R/W	Interrupt number [49:32] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines 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 the bit returns the current state of the corresponding interrupts.

Note: **For descriptions of interrupts and interrupt numbers, see Section "7.5.1.5 List of Interrupt Sources".**

7.6.2.13 Interrupt Priority Register

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	PRI_15	PRI_14	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	PRI_29	PRI_28
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_49	PRI_48

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.

	31	30	29	28	27	26	25	24
bit symbol	PRI_3			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	PRI_2			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	PRI_1			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PRI_0			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-29	PRI_3	R/W	Priority of interrupt number 3
28-24	−	R	Read as 0.
23-21	PRI_2	R/W	Priority of interrupt number 2
20-16	−	R	Read as 0.
15-13	PRI_1	R/W	Priority of interrupt number 1
12-8	−	R	Read as 0.
7-5	PRI_0	R/W	Priority of interrupt number 0
4-0	−	R	Read as 0.

7.6.2.14 Vector Table Offset Register

	31	30	29	28	27	26	25	24
bit symbol	-	-	TBLBASE	TBLOFF				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TBLOFF							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TBLOFF							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBLOFF	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-30	−	R	Read as 0.
29	TBLBASE	R/W	Table base The vector table is in: 0: Code space 1: SRAM space
28-7	TBLOFF	R/W	Offset value 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.
6-0	−	R	Read as 0.

7.6.2.15 Application Interrupt and Reset Control Register

	31	30	29	28	27	26	25	24
bit symbol	VECTKEY/VECTKEYSTAT							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	VECTKEY/VECTKEYSTAT							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENDIANESS	-	-	-	-	PRIGROUP		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	SYSRESET REQ	VECTCLR ACTIVE	VECTRESET
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	VECTKEY (Write)/ VECTKEY-STAT(Read)	R/W	Register key [Write] Writing to this register requires 0x5FA in the <VECTKEY> field. [Read] Read as 0xFA05.
15	ENDIANESS	R/W	Endianness bit:(Note1) 1: big endian 0: little endian
14-11	−	R	Read as 0.
10-8	PRIGROUP	R/W	Interrupt priority grouping 000: seven bits of pre-emption priority, one bit of subpriority 001: six bits of pre-emption priority, two bits of subpriority 010: five bits of pre-emption priority, three bits of subpriority 011: four bits of pre-emption priority, four bits of subpriority 100: three bits of pre-emption priority, five bits of subpriority 101: two bits of pre-emption priority, six bits of subpriority 110: one bit of pre-emption priority, seven bits of subpriority 111: no pre-emption priority, eight bits of subpriority The bit configuration to split the interrupt priority register <PRI_n> into pre-emption priority and sub priority.
7-3	−	R	Read as 0.
2	SYSRESET REQ	R/W	System Reset Request. 1=CPU outputs a SYSRESETREQ signal. (note2)
1	VECTCLR ACTIVE	R/W	Clear active vector bit 1: clear all state information for active NMI, fault, and interrupts 0: do not clear. This bit self-clears. It is the responsibility of the application to reinitialize the stack.
0	VECTRESET	R/W	System Reset bit 1: reset system 0: do not reset system Resets the system, with the exception of debug components (FPB, DWT and ITM) by setting "1" and this bit is also zero cleared.

Note 1: **Little-endian is the default memory format for this product.**

Note 2: **When SYSRESETREQ is output, warm reset is performed on this product. <SYSRESETREQ> is cleared by warm reset.**

7.6.2.16 System Handler Priority Register

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. Unused bits return "0" when read, and writing to unused bits has no effect.

	31	30	29	28	27	26	25	24
bit symbol	PRI_7			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	PRI_6			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	PRI_5			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PRI_4			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-29	PRI_7	R/W	Reserved
28-24	−	R	Read as 0.
23-21	PRI_6	R/W	Priority of Usage Fault
20-16	−	R	Read as 0.
15-13	PRI_5	R/W	Priority of Bus Fault
12-8	−	R	Read as 0.
7-5	PRI_4	R/W	Priority of Memory Management
4-0	−	R	Read as 0.

7.6.2.17 System Handler Control and State Register

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	USGFAULT ENA	BUSFAULT ENA	MEMFAULT ENA
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SVCALL PENDED	BUSFAULT PENDED	MEMFAULT PENDED	USGFAULT PENDED	SYSTICKACT	PENDSVACT	-	MONITOR ACT
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SVCALLACT	-	-	-	USGFAULT ACT	-	BUSFAULT ACT	MEMFAULT ACT
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-19	−	R	Read as 0.
18	USGFAULT ENA	R/W	Usage Fault 0: Disabled 1: Enable
17	BUSFAUL TENA	R/W	Bus Fault 0: Disabled 1: Enable
16	MEMFAULT ENA	R/W	Memory Management 0: Disabled 1: Enable
15	SVCALL PENDED	R/W	SVCall 0: Not pended 1: Pended
14	BUSFAULT PENDED	R/W	Bus Fault 0: Not pended 1: Pended
13	MEMFAULT PENDED	R/W	Memory Management 0: Not pended 1: Pended
12	USGFAULT PENDED	R/W	Usage Fault 0: Not pended 1: Pended
11	SYSTICKACT	R/W	SysTick 0: Inactive 1: Active
10	PENDSVACT	R/W	PendSV 0: Inactive 1: Active
9	−	R	Read as 0.
8	MONITORACT	R/W	Debug Monitor 0: Inactive 1: Active
7	SVCALLACT	R/W	SVCall 0: Inactive 1: Active
6-4	−	R	Read as 0.

Bit	Bit Symbol	Type	Function
3	USGFAULT ACT	R/W	Usage Fault 0: Inactive 1: Active
2	−	R	Read as 0.
1	BUSFAULT ACT	R/W	Bus Fault 0: Inactive 1: Active
0	MEMFAULT ACT	R/W	Memory Management 0: Inactive 1: Active

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

7.6.3 Clock generator registers

7.6.3.1 CGIMCGA(CG Interrupt Mode Control Register A)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCG3			EMST3		-	INT3EN
After reset	0	0	1	0	0	0	Undefined	0
	23	22	21	20	19	18	17	16
bit symbol	-	EMCG2			EMST2		-	INT2EN
After reset	0	0	1	0	0	0	Undefined	0
	15	14	13	12	11	10	9	8
bit symbol	-	EMCG1			EMST1		-	INT1EN
After reset	0	0	1	0	0	0	Undefined	0
	7	6	5	4	3	2	1	0
bit symbol	-	EMCG0			EMST0		-	INT0EN
After reset	0	0	1	0	0	0	Undefined	0

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-28	EMCG3[2:0]	R/W	active level setting of INT3 standby clear request. (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
27-26	EMST3[1:0]	R	active level of INT3 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
25	−	R	Reads as undefined.
24	INT3EN	R/W	INT3 clear input 0:Disable 1: Enable
23	−	R	Read as 0.
22-20	EMCG2[2:0]	R/W	active level setting of INT2 standby clear request. (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
19-18	EMST2[1:0]	R	active level of INT2 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
17	−	R	Reads as undefined.
16	INT2EN	R/W	INT2 clear input 0:Disable 1: Enable
15	−	R	Read as 0.

Bit	Bit Symbol	Type	Function
14-12	EMCG1[2:0]	R/W	active level setting of INT1 standby clear request. (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
11-10	EMST1[1:0]	R	active level of INT1 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
9	−	R	Reads as undefined.
8	INT1EN	R/W	INT1 clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCG0[2:0]	R/W	active level setting of INT0 standby clear request. (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
3-2	EMST0[1:0]	R	active level of INT0 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
1	−	R	Reads as undefined.
0	INT0EN	R/W	INT0 clear input 0:Disable 1: Enable

Note 1: **<EMSTx> is effective only when <EMCGx[2:0]> is set to "100" for both rising and falling edge. The active level used for the reset of standby can be checked by referring <EMSTx>. If interrupts are cleared with the CGICRCG register, <EMSTx> is also cleared.**

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

7.6.3.2 CGIMCGB(CG Interrupt Mode Control Register B)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCG7			EMST7		-	INT7EN
After reset	0	0	1	0	0	0	Undefined	0
	23	22	21	20	19	18	17	16
bit symbol	-	EMCG6			EMST6		-	INT6EN
After reset	0	0	1	0	0	0	Undefined	0
	15	14	13	12	11	10	9	8
bit symbol	-	EMCG5			EMST5		-	INT5EN
After reset	0	0	1	0	0	0	Undefined	0
	7	6	5	4	3	2	1	0
bit symbol	-	EMCG4			EMST4		-	INT4EN
After reset	0	0	1	0	0	0	Undefined	0

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-28	EMCG7[2:0]	R/W	active level setting of INTRMCRX0 standby clear request **Set it as shown below.** 011: Rising edge
27-26	EMST7[1:0]	R	active level of INTRMCRX0 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
25	−	R	Reads as undefined.
24	INT7EN	R/W	INTRMCRX0 clear input 0:Disable 1: Enable
23	−	R	Read as 0.
22-20	EMCG6[2:0]	R/W	active level setting of INTCECRX standby clear request **Set it as shown below.** 011: Rising edge
19-18	EMST6[1:0]	R	active level of INTCECRX standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
17	−	R	Reads as undefined.
16	INT6EN	R/W	INTCECRX clear input 0:Disable 1: Enable
15	−	R	Read as 0.
14-12	EMCG5[2:0]	R/W	active level setting of INT5 standby clear request (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
11-10	EMST5[1:0]	R	active level of INT5 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
9	−	R	Reads as undefined.

Bit	Bit Symbol	Type	Function
8	INT5EN	R/W	INT5 clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCG4[2:0]	R/W	active level setting of INT4 standby clear request (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
3-2	EMST4[1:0]	R	active level of INT4 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
1	−	R	Reads as undefined.
0	INT4EN	R/W	INT4 clear input 0:Disable 1: Enable

Note 1: **<EMSTx> is effective only when <EMCGx[2:0]> is set to "100" for both rising and falling edge. The active level used for the reset of standby can be checked by referring <EMSTx>. If interrupts are cleared with the CGICRCG register, <EMSTx> is also cleared.**

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

7.6.3.3 CGIMCGC(CG Interrupt Mode Control Register C)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCGB			EMSTB		-	INTBEN
After reset	0	0	1	0	0	0	Undefined	0
	23	22	21	20	19	18	17	16
bit symbol	-	EMCGA			EMSTA		-	INTAEN
After reset	0	0	1	0	0	0	Undefined	0
	15	14	13	12	11	10	9	8
bit symbol	-	EMCG9			EMST9		-	INT9EN
After reset	0	0	1	0	0	0	Undefined	0
	7	6	5	4	3	2	1	0
bit symbol	-	EMCG8			EMST8		-	INT8EN
After reset	0	0	1	0	0	0	Undefined	0

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-28	EMCGB[2:0]	R/W	active level setting of INTRMCRX1 standby clear request. **Set it as shown below.** 011: Rising edge
27-26	EMSTB[1:0]	R	active level of INTRMCRX1 standby clear request. 00: − 01: Rising edge 10: Falling edge 11: Both edges
25	−	R	Reads as undefined.
24	INTBEN	R/W	INTRMCRX1 clear input 0:Disable 1: Enable
23	−	R	Read as 0.
22-20	EMCGA[2:0]	R/W	active level setting of INT7 standby clear request. (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
19-18	EMSTA[1:0]	R	active level of INT7 standby clear request. 00: − 01: Rising edge 10: Falling edge 11: Both edges
17	−	R	Reads as undefined.
16	INTAEN	R/W	INT7clear input 0:Disable 1: Enable
15	−	R	Read as 0.
14-12	EMCG9[2:0]	R/W	active level setting of INT6 standby clear request. (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges

Bit	Bit Symbol	Type	Function
11-10	EMST9[1:0]	R	active level of INT6 standby clear request. 00: − 01: Rising edge 10: Falling edge 11: Both edges
9	−	R	Reads as undefined.
8	INT9EN	R/W	INT6 clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCG8[2:0]	R/W	active level setting of INTRTC standby clear request. **Set it as shown below.** 010: Falling edge
3-2	EMST8[1:0]	R	active level of INTRTC standby clear request. 00: − 01: Rising edge 10: Falling edge 11: Both edges
1	−	R	Reads as undefined.
0	INT8EN	R/W	INTRTC clear input 0:Disable 1: Enable

Note 1: <EMSTx> is effective only when <EMCGx[2:0]> is set to "100" for both rising and falling edge. The active level used for the reset of standby can be checked by referring <EMSTx>. If interrupts are cleared with the CGICRCG register, <EMSTx> is also cleared.

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

7.6.3.4 CGIMCGD(CG Interrupt Mode Control Register D)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	1	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	1	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	1	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	EMCGC			EMSTC		-	INTCEN
After reset	0	0	1	0	0	0	Undefined	0

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-28	−	R/W	Write any value.
27-25	−	R	Read as 0.
24	−	R/W	Write as 0.
23	−	R	Read as 0.
22-20	−	R/W	Write any value.
19-17	−	R	Read as 0.
16	−	R/W	Write as 0.
15	−	R	Read as 0.
14-12	−	R/W	Write any value.
11-9	−	R	Read as 0.
8	−	R/W	Write as 0.
7	−	R	Read as 0.
6-4	EMCGC[2:0]	R/W	active level setting of INTCECTX standby clear request. **Set it as shown below.** 011: Rising edge
3-2	EMSTC[1:0]	R	active level of INTCECTX standby clear request. 00: − 01: Rising edge 10: Falling edge 11: Both edges
1	−	R	Reads as undefined.
0	INTCEN	R/W	INTCECTX Clear input 0:Disable 1: Enable

Note 1: **<EMSTx> is effective only when <EMCGx[2:0]> is set to "100" for both rising and falling edge. The active level used for the reset of standby can be checked by referring <EMSTx>. If interrupts are cleared with the CGICRCG register, <EMSTx> is also cleared.**

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

7.6.3.5 CGICRCG(CG Interrupt Request Clear Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	ICRCG				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4-0	ICRCG[4:0]	W	Clear interrupt requests. 0_0000: INT0 0_0001: INT1 0_0010: INT2 0_0011: INT3 0_0100: INT4 0_0101: INT5 0_0110: INTCECRX 0_0111: INTRMCRX0 0_1000: INTRTC 0_1001: INT6 0_1010: INT7 0_1011: INTRMCRX1 0_1100: INTCECTX 0_1101 to 1_1111: setting prohibited. Read as 0.

7.6.3.6 CGNMIFLG(NMI Flag Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	NMIFLG1	NMIFLG0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	NMIFLG1	R	NMI source generation flag 0: not applicable 1: generated from $\overline{\text{NMI}}$ pin
0	NMIFLG0	R	NMI source generation flag 0: not applicable 1: generated from WDT

Note: **<NMIFLG> are cleared to "0" when they are read.**

7.6.3.7 CGRSTFLG (Reset Flag Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After pin reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After pin reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After pin reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	SYSRSTF	-	WDTRSTF	PINRSTF	PONRSTF
After pin reset	0	0	0	0	0	0	1	1 / 0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4	SYSRSTF	R/W	Debug reset flag(Note1) 0: "0" is written 1: Reset from SYSRESETREQ
3	−	R/W	Write as 0.
2	WDTRSTF	R/W	WDT reset flag 0: "0" is written 1: Reset from WDT
1	PINRSTF	R/W	$\overline{\text{RESET}}$ pin flag 0: "0" is written 1: Reset from RESET pin
0	PONRSTF	R/W	Power-on flag 0: "0" is written 1: "1" is set to this bit in initial reset state right after power-on.

Note 1: **This flag indicates a reset generated by the SYSRESETREQ bit of the Application Interrupt and Reset Control Register of the CPU's NVIC.**

Note 2: **This register is not cleared automatically. Write "0" to clear the register.**

8. Input/Output Ports

8.1 Port Functions

8.1.1 Function Lists

TMPM330FDFG/FYFG/FWFG has 78 ports. Besides the ports function, these ports can be used as I/O pins for peripheral functions.

Table 8-1, Table 8-2 and Table 8-3 show the port function table.

Table 8-1 Port Function List (Port A-Port C)

Port	Pin	Input/ Output	Pull-up Pull-down	Schmitt input	Noise Filter	Program-mable Open-drain	Function pin
Port A	PA0	I/O	Pull-up	o	−	−	TMS/ SWDIO
	PA1	I/O	Pull-down	−	−	−	TCK/ SWCLK
	PA2	I/O	Pull-up	−	−	−	TRACECLK
	PA3	I/O	Pull-up	−	−	−	TRACEDATA0
	PA4	I/O	Pull-up	−	−	−	TRACEDATA1
	PA5	I/O	Pull-up	−	−	−	TRACEDATA2
	PA6	I/O	Pull-up	−	−	−	TRACEDATA3
	PA7	I/O	Pull-up	−	−	−	−
Port B	PB0	I/O	Pull-up	−	−	−	TDO/ SWV
	PB1	I/O	Pull-up	−	−	−	TDI
	PB2	I/O	Pull-up	o	−	−	$\overline{\text{TRST}}$
	PB3	I/O	Pull-up	−	−	−	−
	PB4	I/O	Pull-up	−	−	−	−
	PB5	I/O	Pull-up	−	−	−	−
	PB6	I/O	Pull-up	−	−	−	−
	PB7	I/O	Pull-up	−	−	−	−
Port C	PC0	Input	Pull-up	−	−	−	AIN0
	PC1	Input	Pull-up	−	−	−	AIN1
	PC2	Input	Pull-up	−	−	−	AIN2
	PC3	Input	Pull-up	−	−	−	AIN3

o : Exist

- : Not exist

Note: The noise elimination width of the noise filter is approximately 30 ns under typical conditions.

Table 8-2 Port Function List (Port D-Port G)

Port	Pin	Input/Output	Pull-up Pull-down	Schmitt Input	Noise Filter	Programmable Open-drain	Function pin
Port D	PD0	Input	Pull-up	−	−	−	AIN4, TB5IN0
	PD1	Input	Pull-up	−	−	−	AIN5, TB5IN1
	PD2	Input	Pull-up	−	−	−	AIN6, TB6IN0
	PD3	Input	Pull-up	−	−	−	AIN7, TB6IN1
	PD4	Input	Pull-up	−	−	−	AIN8
	PD5	Input	Pull-up	−	−	−	AIN9
	PD6	Input	Pull-up	−	−	−	AIN10
	PD7	Input	Pull-up	−	−	−	AIN11
Port E	PE0	I/O	Pull-up	−	−	o	TXD0
	PE1	I/O	Pull-up	o	−	o	RXD0
	PE2	I/O	Pull-up	o	−	o	SCLK0, $\overline{\text{CTS0}}$
	PE3	I/O	Pull-up	o	−	o	RXIN0
	PE4	I/O	Pull-up	−	−	o	TXD1
	PE5	I/O	Pull-up	o	−	o	RXD1
	PE6	I/O	Pull-up	o	−	o	SCLK1, $\overline{\text{CTS1}}$
Port F	PF0	I/O	Pull-up	−	−	o	TXD2
	PF1	I/O	Pull-up	o	−	o	RXD2
	PF2	I/O	Pull-up	o	−	o	SCLK2, $\overline{\text{CTS2}}$
	PF3	I/O	Pull-up	o	−	o	RXIN1
	PF4	I/O	Pull-up	o	−	o	SDA1/ SO1
	PF5	I/O	Pull-up	o	−	o	SCL1/ SI1
	PF6	I/O	Pull-up	o	−	o	SCK1
	PF7	I/O	Pull-up	o	o	o	INT5
Port G	PG0	I/O	Pull-up	o	−	o	SDA0/ SO0
	PG1	I/O	Pull-up	o	−	o	SCL0/ SI0
	PG2	I/O	Pull-up	o	−	o	SCK0
	PG3	I/O	Pull-up	o	o	o	INT4
	PG4	I/O	Pull-up	o	−	o	SDA2/ SO2
	PG5	I/O	Pull-up	o	−	o	SCL2/ SI2
	PG6	I/O	Pull-up	o	−	o	SCK2
	PG7	I/O	Pull-up	−	−	o	TB8OUT

o : Exist
- : Not exist

Note: The noise elimination width of the noise filter is approximately 30 ns under typical conditions.

Table 8-3 Port Function List (Port H-Port K)

Port	Pin	Input/Out-put	Pull-up Pull-down	Schmitt input	Noise Filter	Program-mable Open-drain	Function pin
Port H	PH0	I/O	Pull-up	o	−	−	TB0IN0, $\overline{\text{BOOT}}$
	PH1	I/O	Pull-up	o	−	−	TB0IN1
	PH2	I/O	Pull-up	o	−	−	TB1IN0
	PH3	I/O	Pull-up	o	−	−	TB1IN1
	PH4	I/O	Pull-up	o	−	−	TB2IN0
	PH5	I/O	Pull-up	o	−	−	TB2IN1
	PH6	I/O	Pull-up	o	−	−	TB3IN0
	PH7	I/O	Pull-up	o	−	−	TB3IN1
Port I	PI0	I/O	Pull-up	−	−	−	TB0OUT
	PI1	I/O	Pull-up	−	−	−	TB1OUT
	PI2	I/O	Pull-up	−	−	−	TB2OUT
	PI3	I/O	Pull-up	−	−	−	TB3OUT
	PI4	I/O	Pull-up	−	−	−	TB4OUT
	PI5	I/O	Pull-up	−	−	−	TB5OUT
	PI6	I/O	Pull-up	o	−	−	TB4IN0
	PI7	I/O	Pull-up	o	−	−	TB4IN1
Port J	PJ0	I/O	Pull-up	o	o	−	INT0
	PJ1	I/O	Pull-up	o	o	−	INT1
	PJ2	I/O	Pull-up	o	o	−	INT2
	PJ3	I/O	Pull-up	o	o	−	INT3
	PJ4	I/O	Pull-up	−	−	−	TB6OUT
	PJ5	I/O	Pull-up	−	−	−	TB7OUT
	PJ6	I/O	Pull-up	o	o	−	INT6
	PJ7	I/O	Pull-up	o	o	−	INT7
Port K	PK0	I/O	−	o	−	o (Note1)	CEC
	PK1	I/O	Pull-up	−	−	−	SCOUT, $\overline{\text{ALARM}}$
	PK2	I/O	Pull-up	−	−	−	TB9OUT

o : Exist
- : Not exist

Note 1: N-ch open drain port.
Note 2: The noise elimination width of the noise filter is approximately 30 ns under typical conditions.

8.1.2 Port Registers Outline

The following registers need to be configured to use ports.

- PxDATA: Port x data register

  To read/ write port data.

- PxCR: Port x output control register

  To control output.

  PxIE needs to be configured to control input.

- PxFRn: Port x function register n

  To set functions.

  An assigned function can be activated by setting "1".

- PxOD: Port x open drain control register

  To control the programmable open drain.

  Programmable open drain is function to be materialized pseudo-open-drain by setting the PxOD.

  When PxOD is set "1",output buffer is disabled and pseudo-open-drain is materialized.

- PxPUP: Port x pull-up control register

  To control program pull ups.

- PxPDN: Port x pull-down control register

  To control programmable pull downs.

- PxIE: Port x input control register

  To control inputs.

  For avoided through current, default setting prohibits inputs.

8.1.3 Port States in STOP Mode

Input and output in STOP mode are enabled/disabled by the CGSTBYCR<DRVE> bit.

If PxIE or PxCR is enabled with <DRVE>=1, input or output is enabled respectively in STOP mode.If <DRVE>=0, both input and output are disabled in STOP mode except for some ports even if PxIE or PxCR are enabled.

Table 8-4 shows the pin conditions in STOP mode.

Table 8-4 Port conditions in STOP mode

	Pin Name	I/O	<DRVE> = 0	<DRVE> = 1
Exclud-ing port	X1, XT1	Input only	×	×
	X2, XT2	Output only	"High" Level Output	"High" Level Output
	$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, MODE	Input only	o	o
Port	PA0, PB0 [When used for debug (PxFRn<PxmFn>=1) and output is enabled (PxCR<PxmC>=1)] (note)	Input	×	Depends on PxIE[m]
		Output	Enabled when data is valid. Disabled when data is invalid.	
	PF7, PG3, PJ0 to PJ3, PJ6, PJ7 [When used for interrupt (PxFRn<PxmFn>=1) and input is enabled (PxIE<PxmIE>=1)] (note)	Input	o	o
		Output	×	Depends on PxCR[m].
	Other ports	Input	×	Depends on PxIE[m].
		Output	×	Depends on PxCR[m].

o :Input or output enabled

× :Input or output disabled

Note:"x" indicates a port number, "m" a corresponding bit and "n" a function register number.

8.1.4 Precautions for Mode Transition between STOP and SLEEP

If PA1 is configured as a debug function pin of TCK/SWCLK, it prevents the low power consumption mode from being fully effective.

Configure PA1 to function as a general-purpose port if the debug function is not used.

8.2 Port functions

This chapter describes the port registers detail.

This chapter describes only "circuit type" reading circuit configuration.For detailed circuit diagram, refer to "8.3 Block Diagrams of Ports".

8.2.1 Port A (PA0 to PA7)

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 debug interface and the debug trace output.

PA0 and PA1 are assigned as the debug interface after reset. PA0 is initialized as the TMS/SWDIO pin with input, output and pull-up enabled. PA1 is initialized as the TCK/SWCLK pin with input and pull-down enabled. Pins from PA2 to PA7 operate as general-purpose-ports, and input, output and pull-up are disabled.

Note 1: If PA0 is configured as the TMS/SWDIO pin, output is enabled even in STOP mode regardless of the CGSTBYCR<DRVE> bit setting

Note 2: If PA1 is configured as the TCK/SWCLK pin, it prevents the low power consumption mode from being fully effective. Configure PA1 to function as a general-purpose port if the TCK/SWCLK is not used.

8.2.1.1 Port A Circuit Type

	7	6	5	4	3	2	1	0
Type	T1	T9	T9	T9	T9	T9	T6	T12

8.2.1.2 Port A register

Base Address = 0x4000_0000

Register name		Address (Base+)
Port A data register	PADATA	0x0000
Port A output control register	PACR	0x0004
Port A function register 1	PAFR1	0x0008
Port A pull-up control register	PAPUP	0x002C
Port A pull-down control register	PAPDN	0x0030
Port A input control register	PAIE	0x0038

8.2.1.3 PADATA (Port A data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PA7-PA0	R/W	Port A data register.

8.2.1.4 PACR (Port A output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PA7C-PA0C	R/W	Output 0: disable 1: enable

8.2.1.5 PAFR1 (Port A function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PA6F1	PA5F1	PA4F1	PA3F1	PA2F1	PA1F1	PA0F1
After reset	0	0	0	0	0	0	1	1

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	PA6F1	R/W	0: PORT 1: TRACEDATA3
5	PA5F1	R/W	0: PORT 1: TRACEDATA2
4	PA4F1	R/W	0: PORT 1: TRACEDATA1
3	PA3F1	R/W	0: PORT 1: TRACEDATA0
2	PA2F1	R/W	0: PORT 1: TRACECLK
1	PA1F1	R/W	0: PORT 1: TCK/SWCLK
0	PA0F1	R/W	0: PORT 1: TMS/SWDIO

8.2.1.6 PAPUP (Port A pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PA7UP	PA6UP	PA5UP	PA4UP	PA3UP	PA2UP	-	PA0UP
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-2	PA7UP-PA2UP	R/W	Pull-up 0: Disable 1: Enable
1	−	R	Read as 0.
0	PA0UP	R/W	Pull-up 0:Disable 1:Enable

8.2.1.7 PAPDN (Port A pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PA1DN	-
After reset	0	0	0	0	0	0	1	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	PA1DN	R/W	Pull-down 0: Disable 1: Enable
0	−	R	Read as 0.

8.2.1.8 PAIE (Port A input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PA7IE	PA6IE	PA5IE	PA4IE	PA3IE	PA2IE	PA1IE	PA0IE
After reset	0	0	0	0	0	0	1	1

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PA7IE-PA0IE	R/W	Input 0: DIsable 1: Enable

8.2.2 Port B (PB0 to PB7)

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 B performs the debug interface.

Reset configures PB0, PB1 and PB2 as debug interface.PB0 is initialized as the TDO/SWV pin with output enabled. PB1 is initialized as the TDI pin with input and pull-up enabled. PB2 is initialized as the TRST pin with input and pull-up enabled.PB3 to PB7 are initialized as general-purpose ports with input, output and pull-up disabled.

Note: If PB0 is configured as the TDO/SWV pin, output is enabled even in STOP mode regardless of the CGSTBYCR<DRVE> bit setting.

8.2.2.1 Port B Circuit Type

	7	6	5	4	3	2	1	0
Type	T1	T1	T1	T1	T1	T2	T2	T11

8.2.2.2 Port B Register

Base Address = 0x4000_0040

Register name		Address (Base+)
Port B data register	PBDATA	0x0000
Port B output control register	PBCR	0x0004
Port B function register 1	PBFR1	0x0008
Port B pull-up control register	PBPUP	0x002C
Port B input control register	PBIE	0x0038

8.2.2.3 PBDATA (Port B data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PB7-PB0	R/W	Port B data register.

8.2.2.4 PBCR (Port B output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PB7C	PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PB7C-PB0C	R/W	Output 0: Disable 1: Enable

8.2.2.5 PBFR1 (Port B function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PB2F1	PB1F1	PB0F1
After reset	0	0	0	0	0	1	1	1

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PB2F1	R/W	0: PORT 1: $\overline{\text{TRST}}$
1	PB1F1	R/W	0: PORT 1: TDI
0	PB0F1	R/W	0: PORT 1: TDO/SWV

8.2.2.6 PBPUP (Port B pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PB7UP	PB6UP	PB5UP	PB4UP	PB3UP	PB2UP	PB1UP	PB0UP
After reset	0	0	0	0	0	1	1	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PB7UP-PB0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.2.7 PBIE (Port B input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PB7IE	PB6IE	PB5IE	PB4IE	PB3IE	PB2IE	PB1IE	PB0IE
After reset	0	0	0	0	0	1	1	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PB7IE-PB0IE	R/W	Input 0: Disable 1: Enable

8.2.3 Port C (PC0 to PC3)

The port C is a 4-bit input port. Besides the general-purpose input function, the port C functions as analog input pins of the AD converter.

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

To use the Port C as an analog input of the AD converter, disable input on PCIE and disable pull-up on PCPUP.

Note: Unless you use all the bits of port C and port D as analog input pins, conversion accuracy may be reduced. Be sure to verify that this causes no problem on your system.

8.2.3.1 Port C Circuit Type

	7	6	5	4	3	2	1	0
Type	–	–	–	–	T17	T17	T17	T17

8.2.3.2 Port C Register

Base Address = 0x4000_0080

Register name		Address (Base+)
Port C data register	PCDATA	0x0000
Port C pull-up control register	PCPUP	0x002C
Port C input control register	PCIE	0x0038

8.2.3.3 PCDATA (Port C data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PC3	PC2	PC1	PC0
After reset	0	0	0	0	1	1	1	1

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as 0.
3-0	PC3-PC0	R	Port C data register.

8.2.3.4 PCPUP (Port C pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PC3UP	PC2UP	PC1UP	PC0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as 0.
3-0	PC3UP-PC0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.3.5 PCIE (Port C input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PC3IE	PC2IE	PC1IE	PC0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as 0.
3-0	PC3IE-PC0IE	R/W	input 0: Disable 1: Enable

8.2.4 Port D (PD0 to PD7)

The port D is an 8-bit input port. Besides the general-purpose input function, the port D receives an analog input of the AD converter and a 16-bit timer input.

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

Set the PDFR1 and PDIE when you use the port D as input pins of the 16-bit timer.

To use the Port D as an analog input of the AD converter, disable input on PDIE and disable pull-up on PDPUP.

Note: Unless you use all the bits of port C and port D as analog input pins, conversion accuracy may be reduced. Be sure to verify that this causes no problem on your system.

8.2.4.1 Port D Circuit Type

	7	6	5	4	3	2	1	0
Type	T17	T17	T17	T17	T18	T18	T18	T18

8.2.4.2 Port D Register

Base Address = 0x4000_00C0

Register name		Address (Base+)
Port D data register	PDDATA	0x0000
Port D function register 1	PDFR1	0x0008
Port D pull-up control register	PDPUP	0x002C
Port D input control register	PDIE	0x0038

8.2.4.3 PDDATA (Port D data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
After reset	1	1	1	1	1	1	1	1

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PD7-PD0	R	Port D data register.

8.2.4.4 PDFR1 (Port D function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PD3F1	PD2F1	PD1F1	PD0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as 0.
3	PD3F1	R/W	0: PORT 1: TB6IN1
2	PD2F1	R/W	0: PORT 1: TB6IN0
1	PD1F1	R/W	0: PORT 1: TB5IN1
0	PD0F1	R/W	0: PORT 1: TB5IN0

8.2.4.5 PDPUP (Port D pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PD7UP	PD6UP	PD5UP	PD4UP	PD3UP	PD2UP	PD1UP	PD0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PD7UP-PD0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.4.6 PDIE (Port D input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PD7IE	PD6IE	PD5IE	PD4IE	PD3IE	PD2IE	PD1IE	PD0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PD7IE-PD0IE	R/W	Input 0: Disable 1: Enable

8.2.5 Port E (PE0 to PE6)

The port E 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 port function, the port E performs the serial interface function and the remote control signal preprocessor input function.

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

The port E has two types of function register. If you use the port E as a general-purpose port, set "0" to the corresponding bit of the two registers. If you use the port E as other than a general-purpose port, set "1" to the corresponding bit of the function register. Do not set "1" to the both function registers at the same time.

8.2.5.1 Port E Circuit Type

	7	6	5	4	3	2	1	0
Type	−	T16	T4	T10	T4	T16	T4	T10

8.2.5.2 Port E Register

Base Address = 0x4000_0100

Register name		Address (Base+)
Port E data register	PEDATA	0x0000
Port E output control register	PECR	0x0004
Port E function register 1	PEFR1	0x0008
Port E function register 2	PEFR2	0x000C
Port E open drain control register	PEOD	0x0028
Port E pull-up control register	PEPUP	0x002C
Port E input control register	PEIE	0x0038

8.2.5.3 PEDATA (Port E data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PE6	PE5	PE4	PE3	PE2	PE1	PE0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PE6-PE0	R/W	Port E data register

8.2.5.4 PECR (Port E output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PE6C	PE5C	PE4C	PE3C	PE2C	PE1C	PE0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PE6C-PE0C	R/W	Output 0: Disable 1: Enable

8.2.5.5 PEFR1(Port E function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PE6F1	PE5F1	PE4F1	PE3F1	PE2F1	PE1F1	PE0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	PE6F1	R/W	0: PORT 1: SCLK1
5	PE5F1	R/W	0: PORT 1: RXD1
4	PE4F1	R/W	0: PORT 1: TXD1
3	PE3F1	R/W	0: PORT 1: RXIN0
2	PE2F1	R/W	0: PORT 1: SCLK0
1	PE1F1	R/W	0: PORT 1: RXD0
0	PE0F1	R/W	0: PORT 1: TXD0

8.2.5.6 PEFR2(Port E function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PE6F2	-	-	-	PE2F2	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	PE6F2	R/W	0: PORT 1: $\overline{\text{CTS1}}$
5-3	−	R	Read as 0.
2	PE2F2	R/W	0: PORT 1: $\overline{\text{CTS0}}$
1-0	−	R	Read as 0.

8.2.5.7 PEOD (Port E open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PE6OD	PE5OD	PE4OD	PE3OD	PE2OD	PE1OD	PE0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PE6OD- PE0OD	R/W	0: CMOS 1: Open-drain

8.2.5.8 PEPUP (Port E pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PE6UP	PE5UP	PE4UP	PE3UP	PE2UP	PE1UP	PE0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PE6UP-PE0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.5.9 PEIE (Port E input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PE6IE	PE5IE	PE4IE	PE3IE	PE2IE	PE1IE	PE0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PE6IE-PE0IE	R/W	Input 0: Disable 1: Enable

8.2.6 Port F (PF0 to PF7)

The port F 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 F performs the functions of the serial interface, the remote control signal preprocessor, the serial bus interface and the external interrupt input.

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

The port F has two types of function register. If you use the port F as a general-purpose port, set "0" to the corresponding bit of the two registers. If you use the port F as other than a general-purpose port, set "1" to the corresponding bit of the function register. Do not set "1" to the both function registers at the same time.

To use the external interrupt input for releasing STOP mode, select this function in the PFFR1 and enable input in the PFIE register.

These settings enable the interrupt input even if the CGSTBYCR<DRVE> bit in the clock/mode control block is set to stop driving of pins during STOP mode.

Note: In modes other than STOP mode, interrupt input is enabled regardless of the PFFR register setting if input is enabled in PxIE. Make sure to disable unused interrupts when programming the device.

8.2.6.1 Port F Circuit Type

	7	6	5	4	3	2	1	0
Type	T8	T13	T13	T13	T4	T16	T4	T10

8.2.6.2 Port F Register

Base Address = 0x4000_0140

Register name		Address (Base+)
Port F data register	PFDATA	0x0000
Port F output control register	PFCR	0x0004
Port F function register 1	PFFR1	0x0008
Port F function register 2	PFFR2	0x000C
Port F open drain control register	PFOD	0x0028
Port F pull-up control register	PFPUP	0x002C
Port F input control register	PFIE	0x0038

8.2.6.3 PFDATA (Port F data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PF7-PF0	R/W	Port F data register

8.2.6.4 PFCR (Port F output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PF7C	PF6C	PF5C	PF4C	PF3C	PF2C	PF1C	PF0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PF7C-PF0C	R/W	Output 0: Disable 1: Enable

8.2.6.5 PFFR1(Port F function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PF7F1	PF6F1	PF5F1	PF4F1	PF3F1	PF2F1	PF1F1	PF0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PF7F1	R/W	0: PORT 1: INT5
6	PF6F1	R/W	0: PORT 1: SCK1
5	PF5F1	R/W	0: PORT 1: SI1/SCL1
4	PF4F1	R/W	0: PORT 1: SO1/SDA1
3	PF3F1	R/W	0: PORT 1: RXIN1
2	PF2F1	R/W	0: PORT 1: SCLK2
1	PF1F1	R/W	0: PORT 1: RXD2
0	PF0F1	R/W	0: PORT 1: TXD2

8.2.6.6 PFFR2(Port F function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PF2F2	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PF2F2	R/W	0: PORT 1: $\overline{CTS2}$
1-0	−	R	Read as 0.

8.2.6.7 PFOD (Port F open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PF7OD	PF6OD	PF5OD	PF4OD	PF3OD	PF2OD	PF1OD	PF0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PF7OD-PF0OD	R/W	0: CMOS 1: Open-drain

8.2.6.8 PFPUP (Port F pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PF7UP	PF6UP	PF5UP	PF4UP	PF3UP	PF2UP	PF1UP	PF0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PF7UP-PF0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.6.9 PFIE (Port F input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PF7IE	PF6IE	PF5IE	PF4IE	PF3IE	PF2IE	PF1IE	PF0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PF7IE-PF0IE	R/W	Input 0: Disable 1: Enable

8.2.7 Port G (PG0 to PG7)

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, the external interrupt input, and the 16-bit timer output.

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

To use the external interrupt input for releasing STOP mode, select function in the PGFR register and enable input in the PGIE register.

These settings enable the interrupt input even if the CGSTBYCR<DRVE> bit in the clock/mode control block is set to stop driving of pins during STOP mode.

Note: In modes other than STOP mode, interrupt input is enabled regardless of the PGFR register setting if input is enabled in PGIE. Make sure to disable unused interrupts when programming the device.

8.2.7.1 Port G Circuit Type

	7	6	5	4	3	2	1	0
Type	T10	T13	T13	T13	T8	T13	T13	T13

8.2.7.2 Port G Register

Base Address = 0x4000_0180

Register name		Address (Base+)
Port G data register	PGDATA	0x0000
Port G output control register	PGCR	0x0004
Port G function register 1	PGFR1	0x0008
Reserved	-	0x0010
Port G open drain control register	PGOD	0x0028
Port G pull-up control register	PGPUP	0x002C
Port G input control register	PGIE	0x0038

Note: Access to the "reserved" areas is prohibited.

8.2.7.3 PGDATA (Port G data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PG7-PG0	R/W	Port G data register.

8.2.7.4 PGCR (Port G output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7C	PG6C	PG5C	PG4C	PG3C	PG2C	PG1C	PG0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PG7C-PG0C	R/W	Output 0: Disable 1: Enable

8.2.7.5 PGFR1(Port G function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7F1	PG6F1	PG5F1	PG4F1	PG3F1	PG2F1	PG1F1	PG0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PG7F1	R/W	0: PORT 1: TB8OUT
6	PG6F1	R/W	0: PORT 1: SCK2
5	PG5F1	R/W	0: PORT 1: SI2/SCL2
4	PG4F1	R/W	0: PORT 1: SO2/SDA2
3	PG3F1	R/W	0: PORT 1: INT4
2	PG2F1	R/W	0: PORT 1: SCK0
1	PG1F1	R/W	0: PORT 1: SI0/SCL0
0	PG0F1	R/W	0: PORT 1: SO0/SDA0

8.2.7.6 PGOD (Port G open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7OD	PG6OD	PG5OD	PG4OD	PG3OD	PG2OD	PG1OD	PG0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PG7OD- PG0OD	R/W	0: CMOS 1: Open-drain

8.2.7.7 PGPUP (Port G pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7UP	PG6UP	PG5UP	PG4UP	PG3UP	PG2UP	PG1UP	PG0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PG7UP- PG0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.7.8 PGIE (Port G input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7IE	PG6IE	PG5IE	PG4IE	PG3IE	PG2IE	PG1IE	PG0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PG7IE-PG0IE	R/W	Input 0: Disable 1: Enable

8.2.8 Port H (PH0 to PH7)

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 port function, the port H performs the functions of the 16-bit timer input and the operation mode setting.

While a reset signal is in "Low" state, the 「PH0/$\overline{\text{BOOT}}$」 input and pull-up are enabled. At the rising edge of the reset signal, if 「PH0」 is "High", the device enters single mode and boots from the on-chip flash memory. If 「PH0」 is "Low", the device enters single BOOT mode and boots from the internal BOOT program.For details of single boot mode, refer to "Flash Memory Operation".

Reset initializes PH0 to PH7 bits of the port H as general-purpose ports with input and output disabled.Pull-up is enabled for PH0 and disabled for PH1 to PH7.

8.2.8.1 Port H Circuit Type

	7	6	5	4	3	2	1	0
Type	T3	T3	T3	T3	T3	T3	T3	T5

8.2.8.2 Port H Register

Base Address = 0x4000_01C0

Register name		Address (Base+)
Port H data register	PHDATA	0x0000
Port H output control register	PHCR	0x0004
Port H function register 1	PHFR1	0x0008
Reserved	-	0x0010
Port H pull-up control register	PHPUP	0x002C
Port H input control register	PHIE	0x0038

Note: Access to the "reserved" areas is prohibited.

8.2.8.3 PHDATA (Port H data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PH7-PH0	R/W	Port H data register.

8.2.8.4 PHCR (Port H output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PH7C	PH6C	PH5C	PH4C	PH3C	PH2C	PH1C	PH0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PH7C-PH0C	R/W	Output 0: Disable 1: Enable

8.2.8.5 PHFR1(Port H function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PH7F1	PH6F1	PH5F1	PH4F1	PH3F1	PH2F1	PH1F1	PH0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PH7F1	R/W	0: PORT 1: TB3IN1
6	PH6F1	R/W	0: PORT 1: TB3IN0
5	PH5F1	R/W	0: PORT 1: TB2IN1
4	PH4F1	R/W	0: PORT 1: TB2IN0
3	PH3F1	R/W	0: PORT 1: TB1IN1
2	PH2F1	R/W	0: PORT 1: TB1IN0
1	PH1F1	R/W	0: PORT 1: TB0IN1
0	PH0F1	R/W	0: PORT 1: TB0IN0

8.2.8.6 PHPUP (Port H pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PH7UP	PH6UP	PH5UP	PH4UP	PH3UP	PH2UP	PH1UP	PH0UP
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PH7UP-PH0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.8.7 PHIE (Port H input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PH7IE	PH6IE	PH5IE	PH4IE	PH3IE	PH2IE	PH1IE	PH0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PH7IE-PH0IE	R/W	Input 0: Disable 1: Enable

8.2.9 Port I (PI0 to PI7)

The port I 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 I performs the 16-bit timer input/output function.

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

8.2.9.1 Port I Circuit Type

	7	6	5	4	3	2	1	0
Type	T3	T3	T9	T9	T9	T9	T9	T9

8.2.9.2 Port I Register

Base Address = 0x4000_0200

Register name		Address (Base+)
Port I data register	PIDATA	0x0000
Port I output control register	PICR	0x0004
Port I function register 1	PIFR1	0x0008
Reserve	-	0x0010
Port I pull-up control register	PIPUP	0x002C
Port I input control register	PIIE	0x0038

Note: Access to the "reserved" areas is prohibited.

8.2.9.3 PIDATA(Port I data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PI7	PI6	PI5	PI4	PI3	PI2	PI1	PI0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PI7-PI0	R/W	Port I data register.

8.2.9.4 PICR (Port I output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PI7C	PI6C	PI5C	PI4C	PI3C	PI2C	PI1C	PI0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PI7C-PI0C	R/W	Output 0: Disable 1: Enable

8.2.9.5 PIFR1(Port I function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PI7F1	PI6F1	PI5F1	PI4F1	PI3F1	PI2F1	PI1F1	PI0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PI7F1	R/W	0: PORT 1: TB4IN1
6	PI6F1	R/W	0: PORT 1: TB4IN0
5	PI5F1	R/W	0: PORT 1: TB5OUT
4	PI4F1	R/W	0: PORT 1: TB4OUT
3	PI3F1	R/W	0: PORT 1: TB3OUT
2	PI2F1	R/W	0: PORT 1: TB2OUT
1	PI1F1	R/W	0: PORT 1: TB1OUT
0	PI0F1	R/W	0: PORT 1: TB0OUT

8.2.9.6 PIPUP (Port I pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PI7UP	PI6UP	PI5UP	PI4UP	PI3UP	PI2UP	PI1UP	PI0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PI7UP-PI0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.9.7 PIIE (Port I input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PI7IE	PI6IE	PI5IE	PI4IE	PI3IE	PI2IE	PI1IE	PI0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PI7IE-PI0IE	R/W	Input 0: Disable 1: Enable

8.2.10 Port J (PJ0 to PJ7)

The port J 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 J performs the functions of the 16-bit timer output and the external interrupt input.

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

To use the external interrupt input for releasing STOP mode, select this function in the PJFR1 register and enable input in the PJIE register.

These settings enable the interrupt input even if the CGSTBYCR<DRVE> bit in the clock/mode control block is set to stop driving of pins during STOP mode.

Note: In modes other than STOP mode, interrupt input is enabled regardless of the PJFR register setting if input is enabled in PJIE. Make sure to disable unused interrupts when programming the device.

8.2.10.1 Port J Circuit Type

	7	6	5	4	3	2	1	0
Type	T7	T7	T9	T9	T7	T7	T7	T7

8.2.10.2 Port J Register

Base Address = 0x4000_0240

Register name		Address (Base+)
Port J data register	PJDATA	0x0000
Port J output control register	PJCR	0x0004
Port J function register 1	PJFR1	0x0008
Reserved	-	0x0010
Port J pull-up control register	PJPUP	0x002C
Port J input control register	PJIE	0x0038

Note: Access to the "reserved" areas is prohibited.

8.2.10.3 PJDATA (Port J data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PJ7-PJ0	R/W	Port J data register.

8.2.10.4 PJCR (Port J output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7C	PJ6C	PJ5C	PJ4C	PJ3C	PJ2C	PJ1C	PJ0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PJ7C-PJ0C	R/W	Output 0: Disable 1: Enable

8.2.10.5 PJFR1(Port J function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7F1	PJ6F1	PJ5F1	PJ4F1	PJ3F1	PJ2F1	PJ1F1	PJ0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PJ7F1	R/W	0: PORT 1: INT7
6	PJ6F1	R/W	0: PORT 1: INT6
5	PJ5F1	R/W	0: PORT 1: TB7OUT
4	PJ4F1	R/W	0: PORT 1: TB6OUT
3	PJ3F1	R/W	0: PORT 1: INT3
2	PJ2F1	R/W	0: PORT 1: INT2
1	PJ1F1	R/W	0: PORT 1: INT1
0	PJ0F1	R/W	0: PORT 1: INT0

8.2.10.6 PJPUP (Port J pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7UP	PJ6UP	PJ5UP	PJ4UP	PJ3UP	PJ2UP	PJ1UP	PJ0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PJ7UP-PJ0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.10.7 PJIE (Port J input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7IE	PJ6IE	PJ5IE	PJ4IE	PJ3IE	PJ2IE	PJ1IE	PJ0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PJ7IE-PJ0IE	R/W	Input 0: Disable 1: Enable

8.2.11 Port K (PK0 to PK2)

The port K is a general-purpose, 3-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 K performs the functions of the 16-bit timer output, the CEC input, the clock output and the alarm output.

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

Note: PK0 is an N-ch open drain port.

8.2.11.1 Port K Circuit Type

	7	6	5	4	3	2	1	0
Type	–	–	–	–	–	T9	T15	T14

8.2.11.2 Port K Register

Base Address = 0x4000_0280

Register name		Address (Base+)
Port K data register	PKDATA	0x0000
Port K output control register	PKCR	0x0004
Port K function register 1	PKFR1	0x0008
Port K function register 2	PKFR2	0x000C
Port K pull-up control register	PKPUP	0x002C
Port K input control register	PKIE	0x0038

8.2.11.3 PKDATA(Port K data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PK2	PK1	PK0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2-0	PK2-PK0	R/W	Port K data register.

8.2.11.4 PKCR (Port K output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PK2C	PK1C	PK0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2-0	PK2C-PK0C	R/W	Output 0: Disable 1: Enable

8.2.11.5 PKFR1(Port K function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PK2F1	PK1F1	PK0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PK2F1	R/W	0: PORT 1: TB9OUT
1	PK1F1	R/W	0: PORT 1: SCOUT
0	PK0F1	R/W	0: PORT 1: CEC

8.2.11.6 PKFR2(Port K function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PK1F2	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	PK1F2	R/W	0: PORT 1: $\overline{\text{ALARM}}$
0	−	R	Read as 0.

8.2.11.7 PKPUP (Port K pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PK2UP	PK1UP	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2-1	PK2UP-PK1UP	R/W	Pull-up 0: Disable 1: Enable
0	−	R	Read as 0.

8.2.11.8 PKIE (Port K input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PK2IE	PK1IE	PK0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2-0	PK2IE-PK0IE	R/W	Input 0: Disable 1: Enable

8.3 Block Diagrams of Ports

8.3.1 Port Types

The ports are classified as shown below. Please refer to the following pages for the block diagrams of each port type.

Dot lines in the figure indicate the part of the equivalent circuit described in the "Block diagrams of ports".

Table 8-5 Function Lists

Type	GP Port	Function 1	Function 2	Analog	Pull-up	Pull-down	Programmable open-drain	Note
T1	I/O	−	−	−	R	−	−	
T2	I/O	Input	−	−	NoR	−	−	
T3	I/O	Input	−	−	R	−	−	
T4	I/O	Input	−	−	R	−	o	
T5	I/O	Input	−	−	NoR	−	−	$\overline{\text{BOOT}}$ input enabled during reset
T6	I/O	Input	−	−	−	NoR	−	
T7	I/O	Input (int)	−	−	R	−	−	
T8	I/O	Input (int)	−	−	R	−	o	
T9	I/O	Output	−	−	R	−	−	
T10	I/O	Output	−	−	R	−	o	
T11	I/O	Output	−	−	R	−	−	Function output triggered by enable signal
T12	I/O	I/O	−	−	NoR	−	−	Function output triggered by enable signal
T13	I/O	I/O	−	−	R	−	o	
T14	I/O	I/O	−	−	−	−	−	Nch open drain port
T15	I/O	Output	Output	−	R	−	−	
T16	I/O	I/O	Input	−	R	−	o	
T17	Input	−	−	o	R	−	−	
T18	Input	Input	−	o	R	−	−	

int: Interrupt input

−: Not exist

o: Exist

R: Forced disable during reset.

NoR: Unaffected by reset.

8.3.2 Type T1

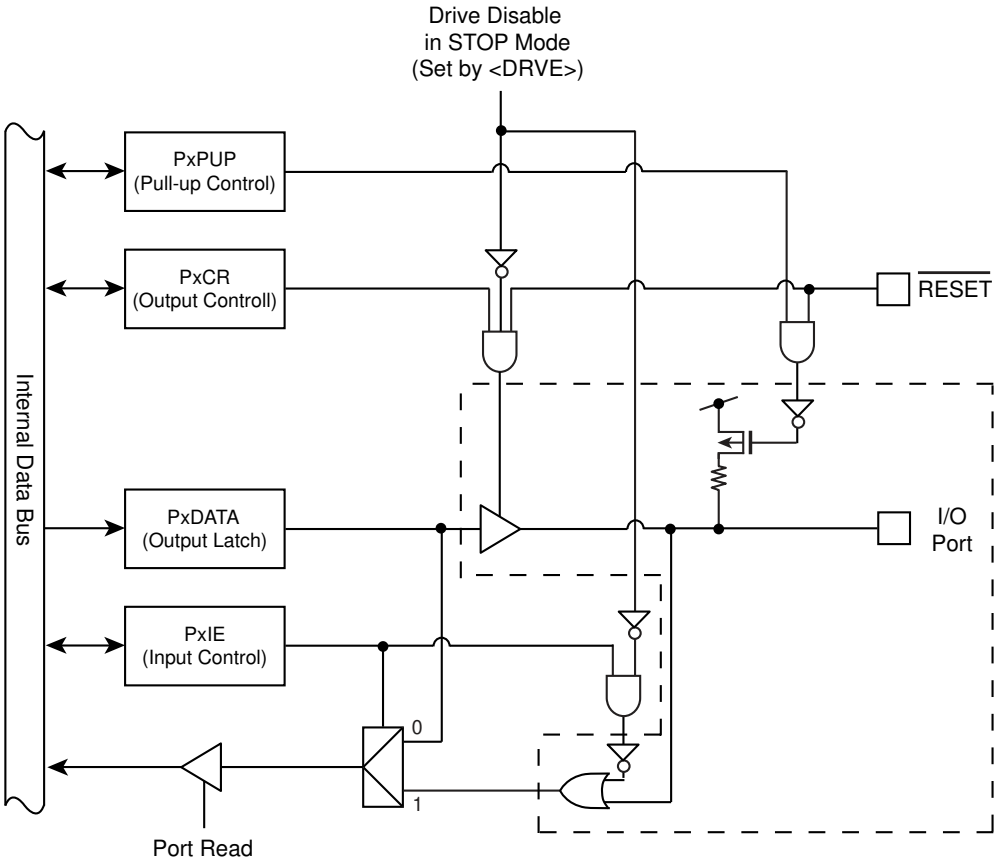


Figure 8-1 Port Type T1

8.3.3 Type T2

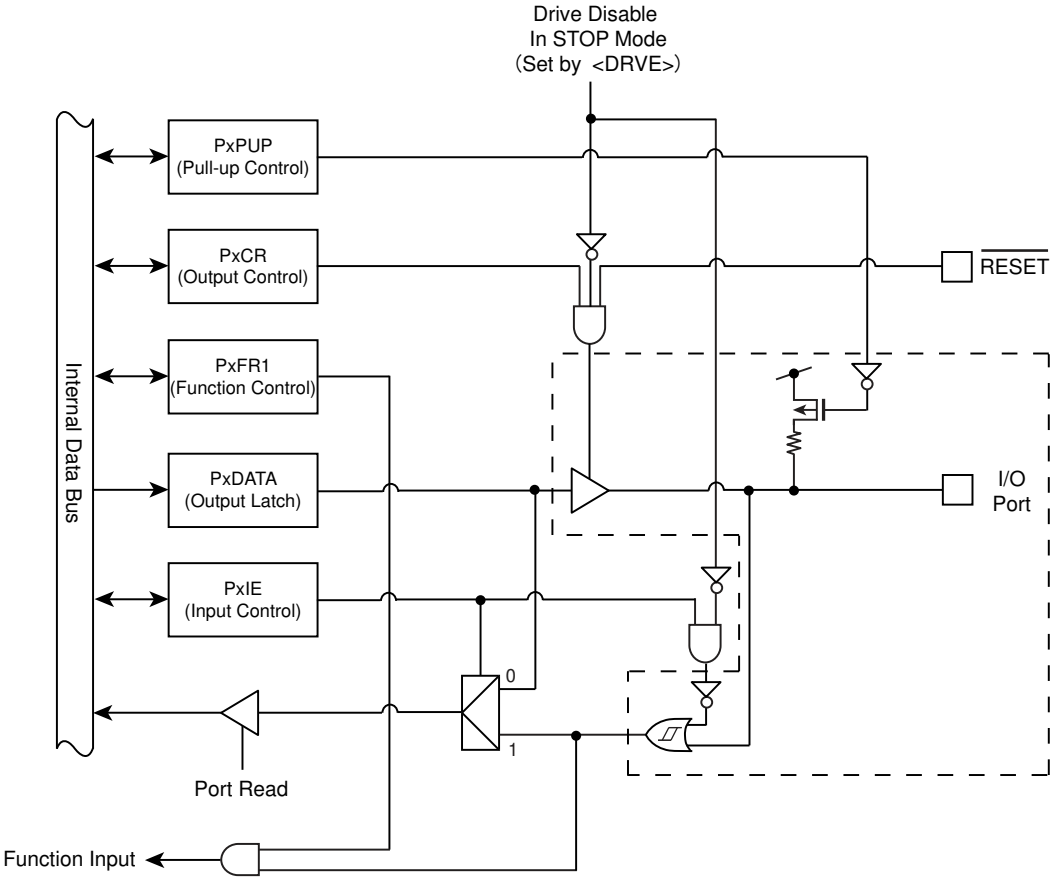


Figure 8-2 Port type T2

8.3.4 Type T3

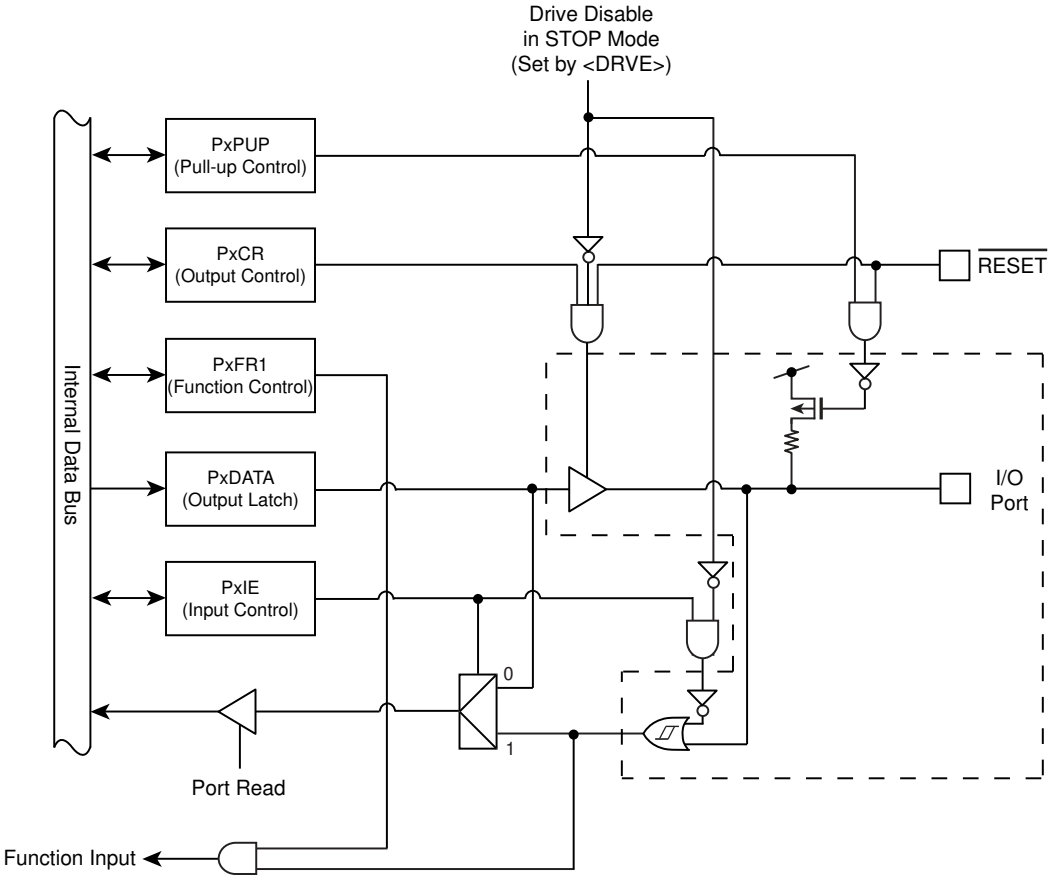


Figure 8-3 Port Type T3

8.3.5 Type T4

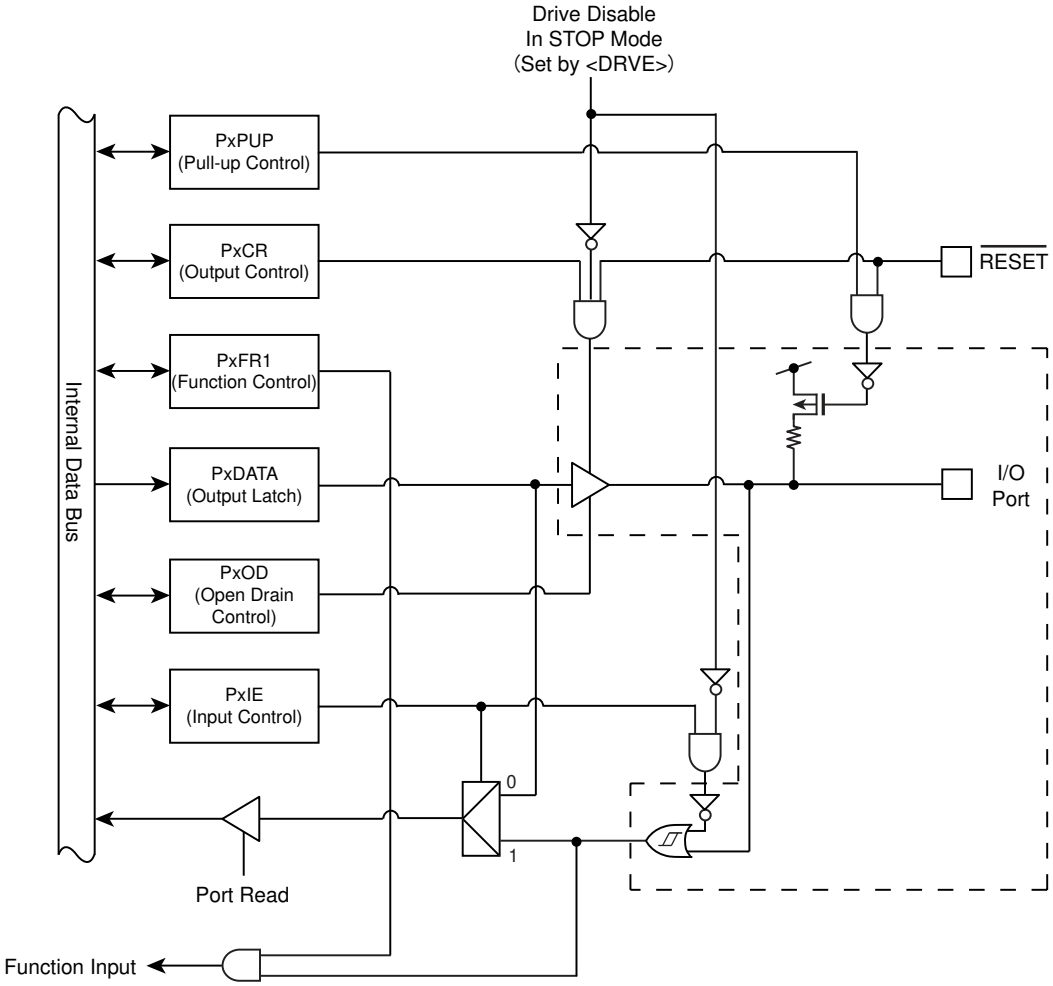


Figure 8-4 Port Type T4

8.3.6 Type5 T5

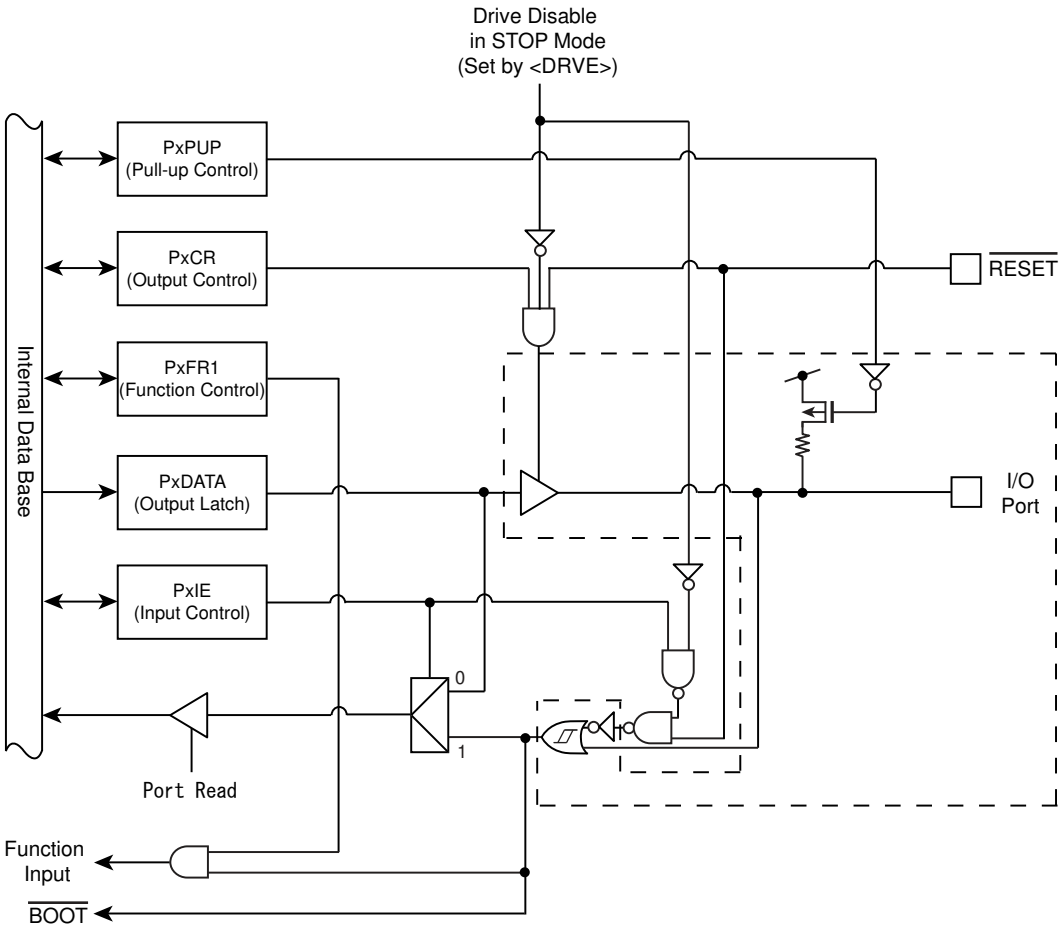


Figure 8-5 Port Type T5

8.3.7 Type T6

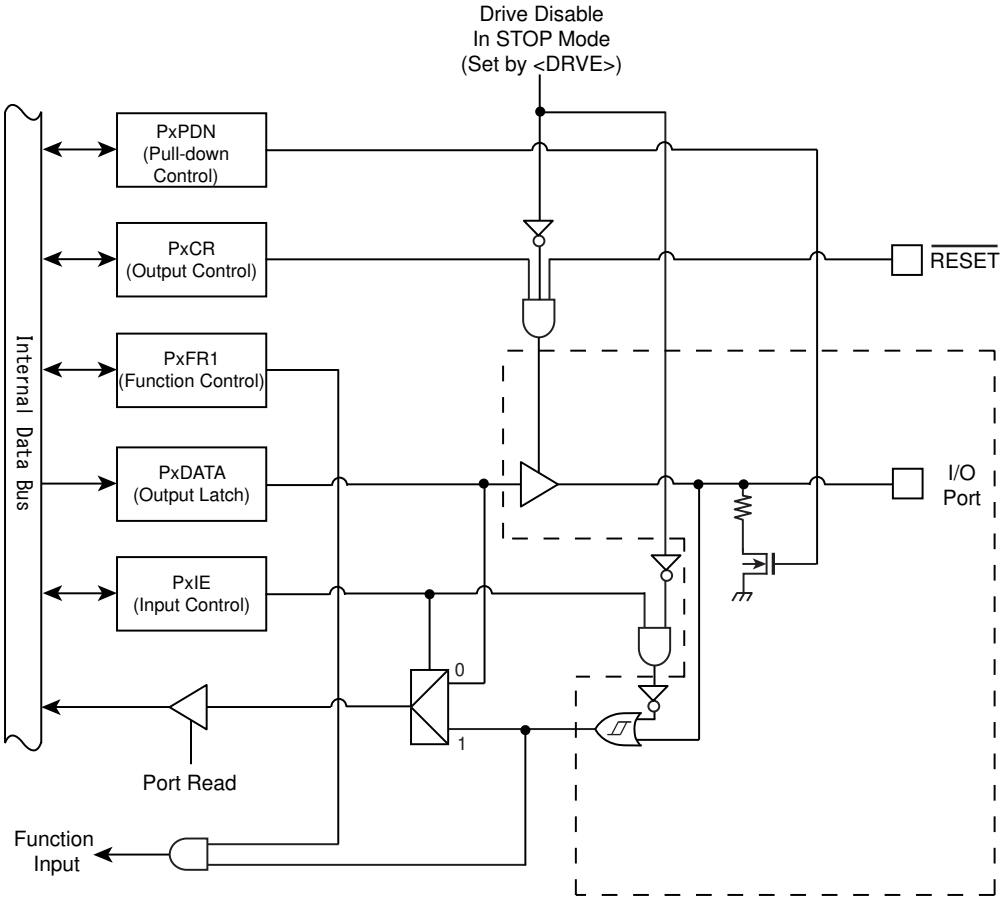


Figure 8-6 Port Type T6

8.3.8 Type T7

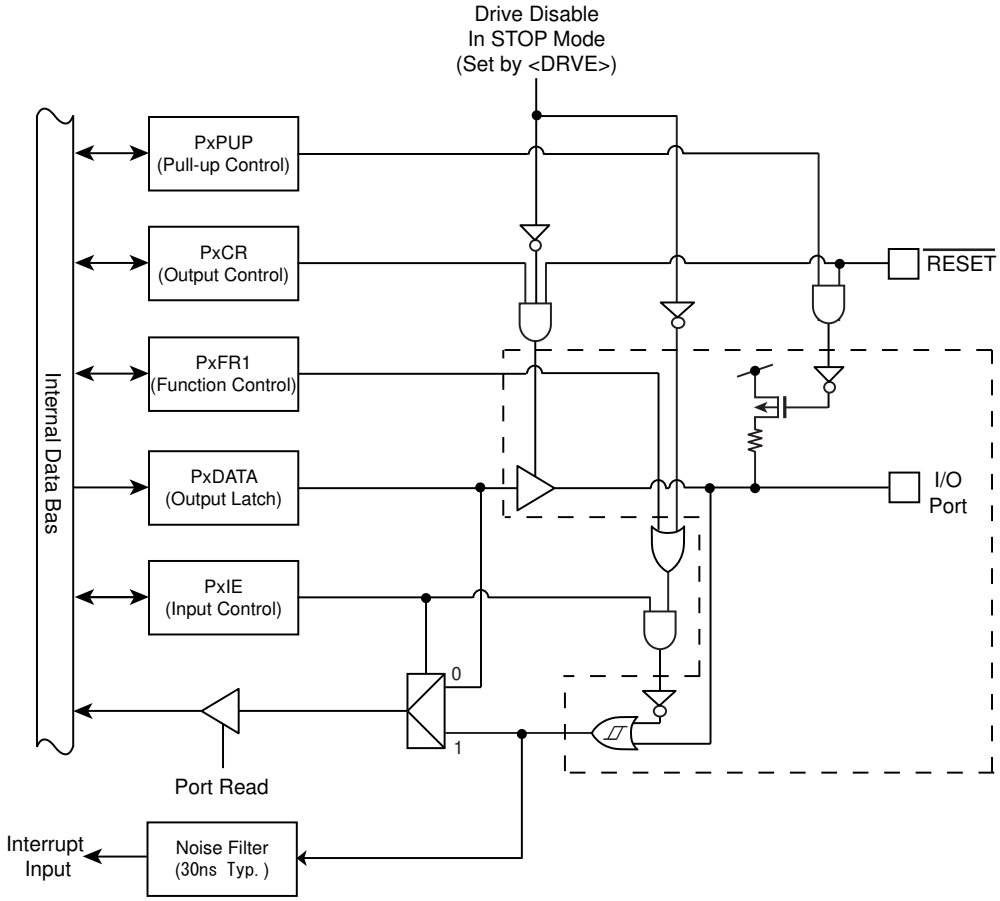


Figure 8-7 Port Type T7

8.3.9 Type T8

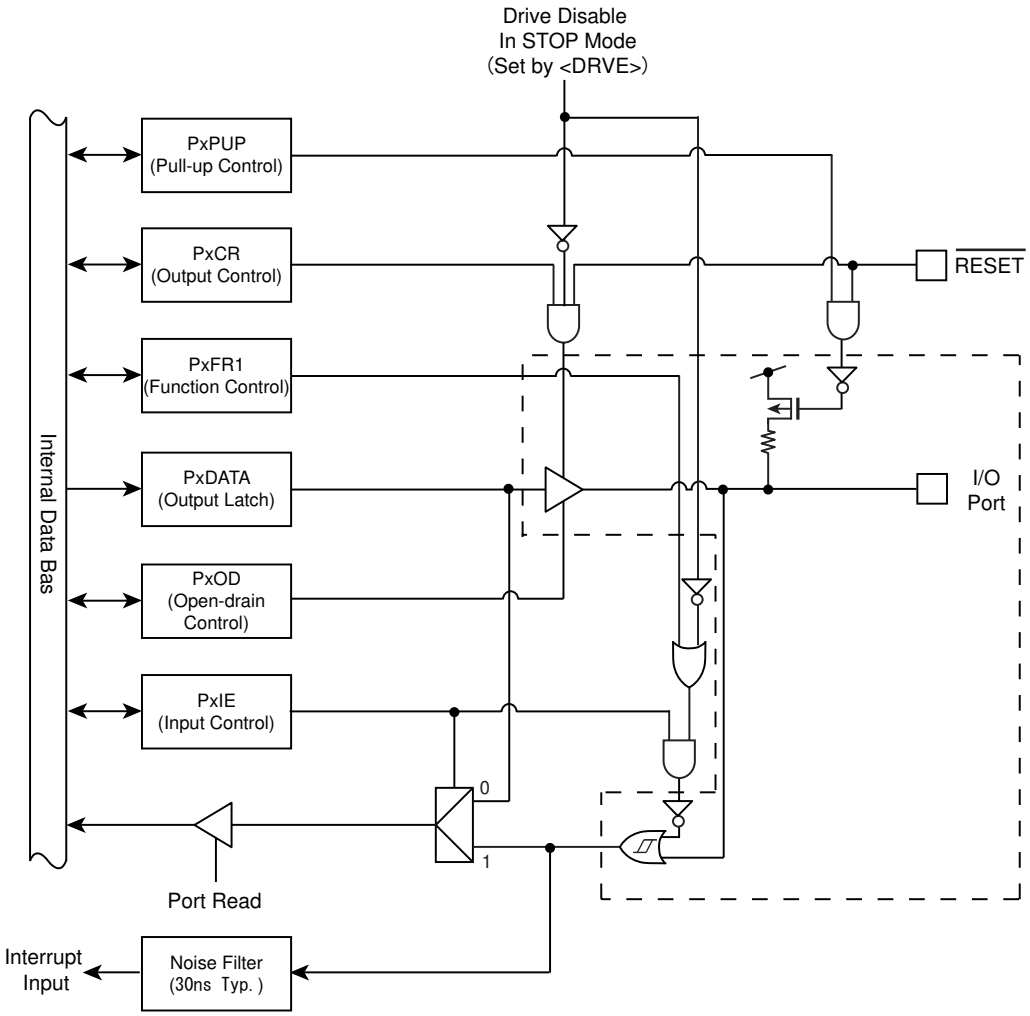


Figure 8-8 Port Type T8

8.3.10 Type T9

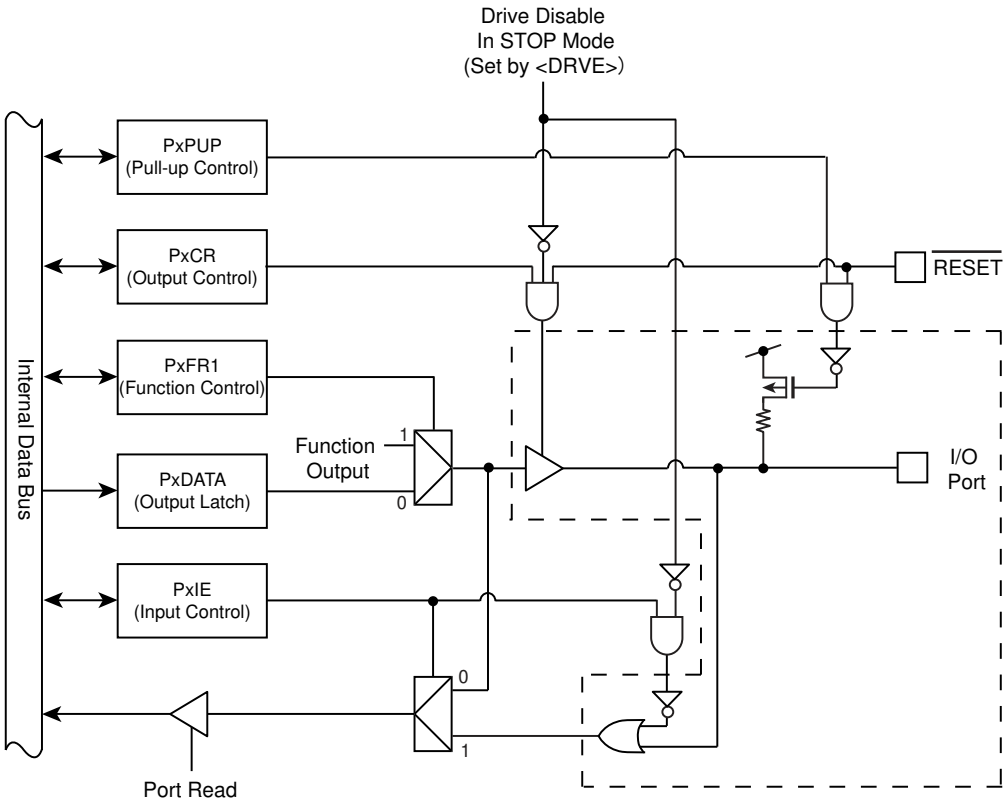


Figure 8-9 Port Type T9

8.3.11 Type T10

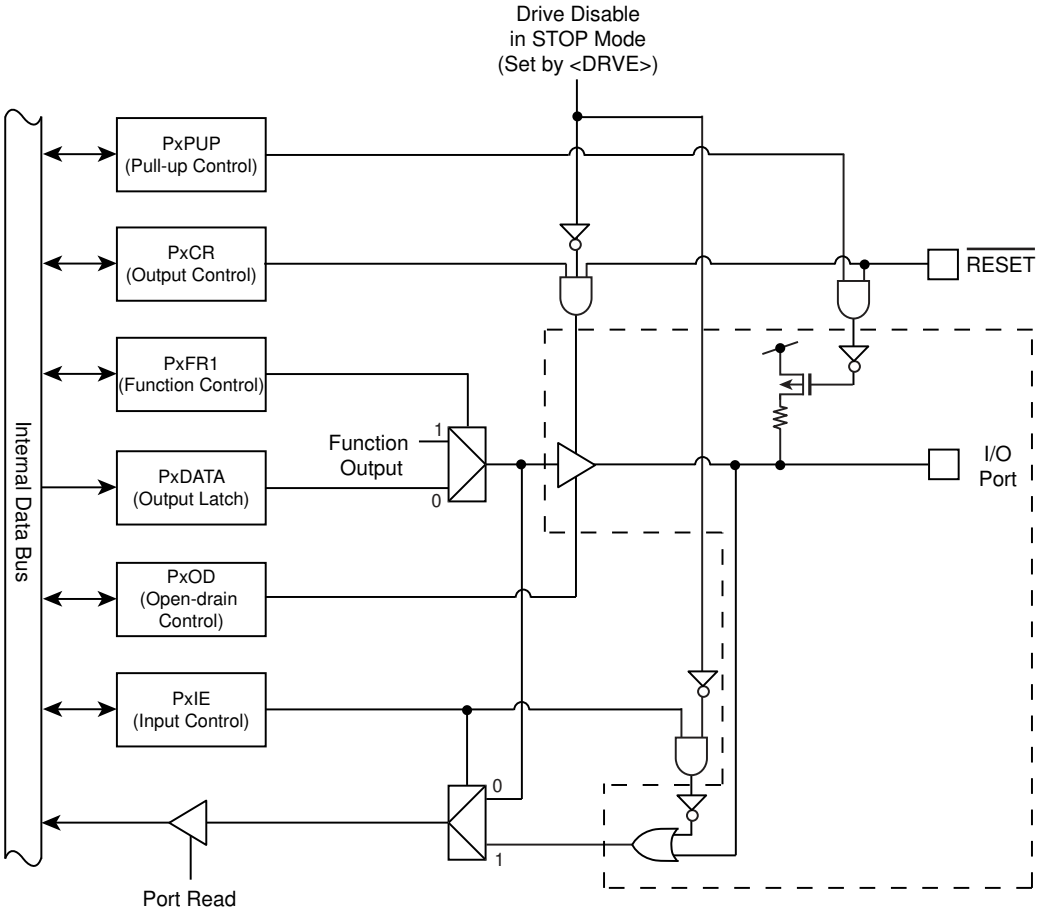


Figure 8-10 Port Type T10

8.3.12 Type T11

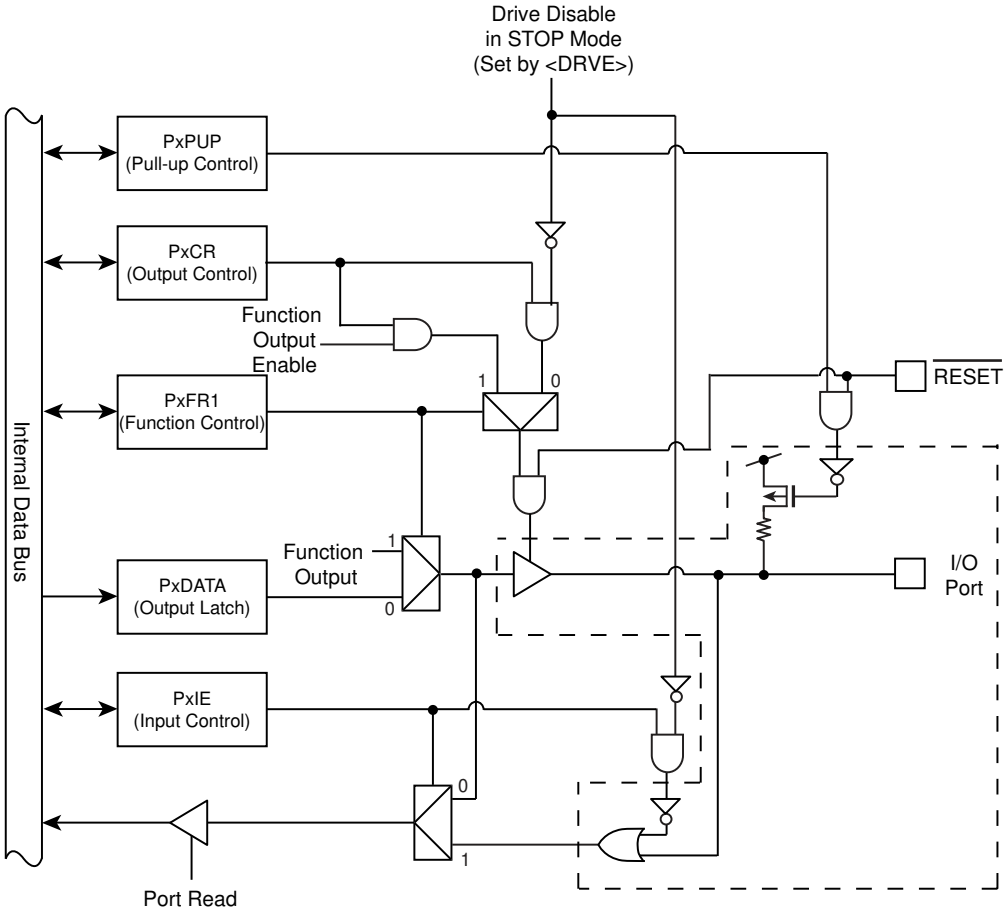


Figure 8-11 Port Type T11

8.3.13 Type T12

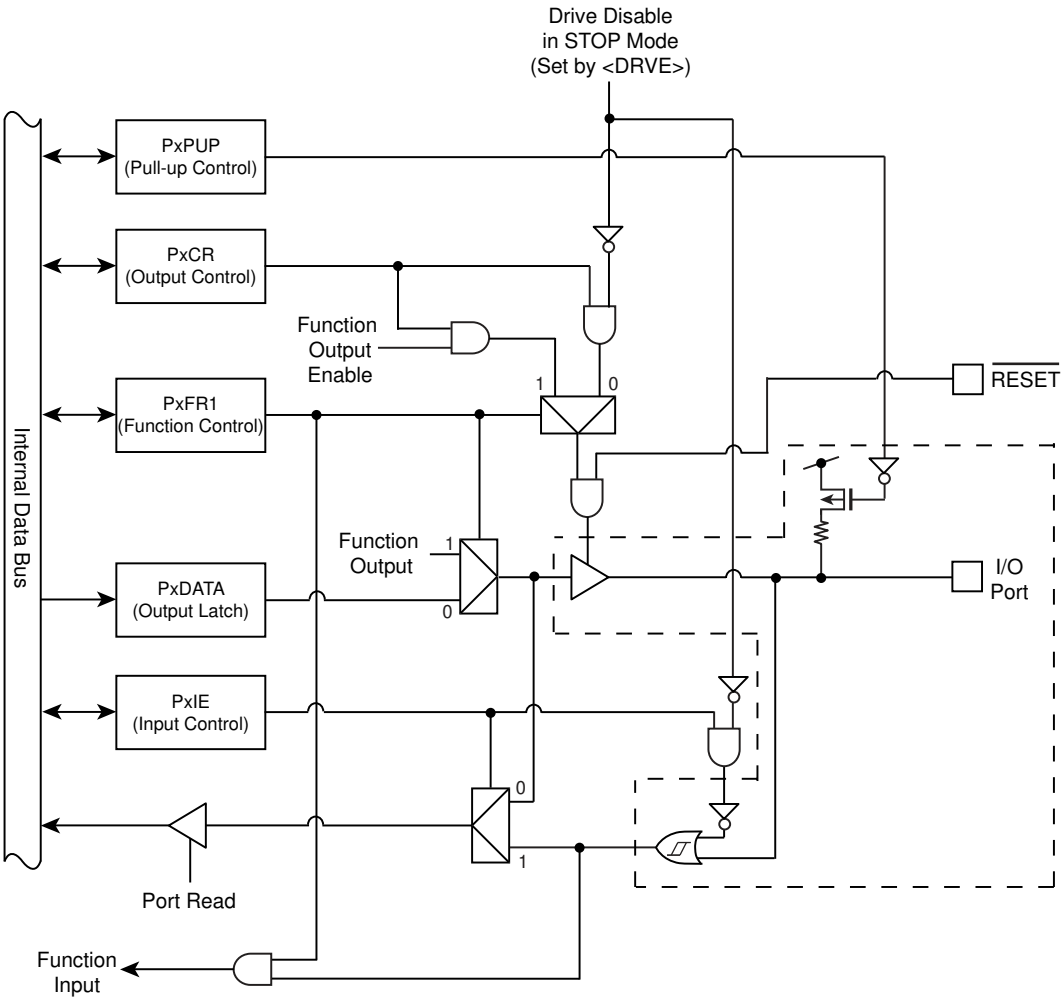


Figure 8-12 Port Type T12

8.3.14 Type T13

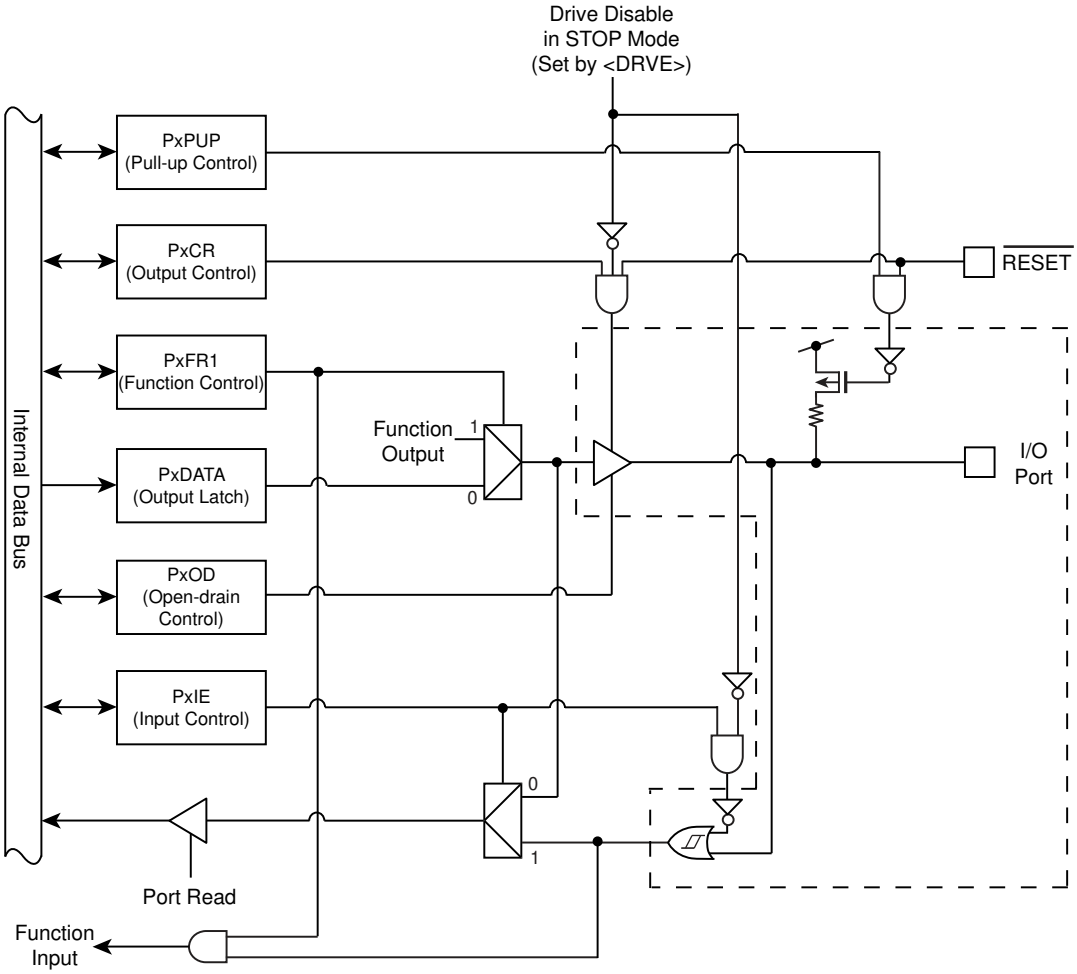


Figure 8-13 Port Type T13

8.3.15 Type T14

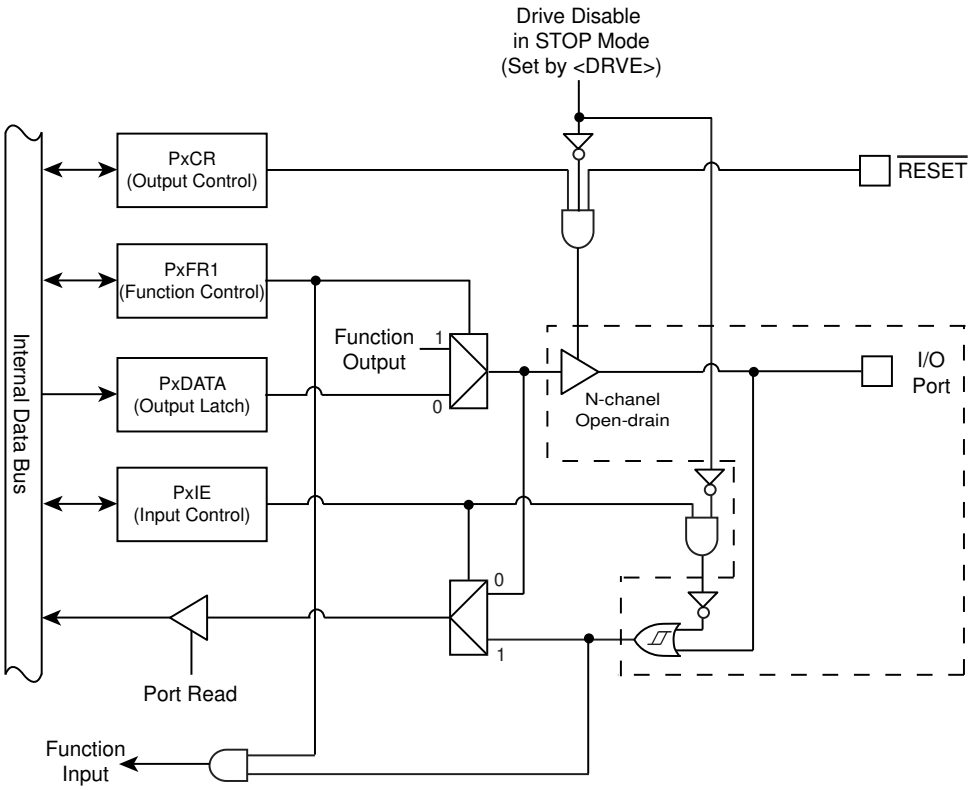


Figure 8-14 Port Type T14

8.3.16 Type T15

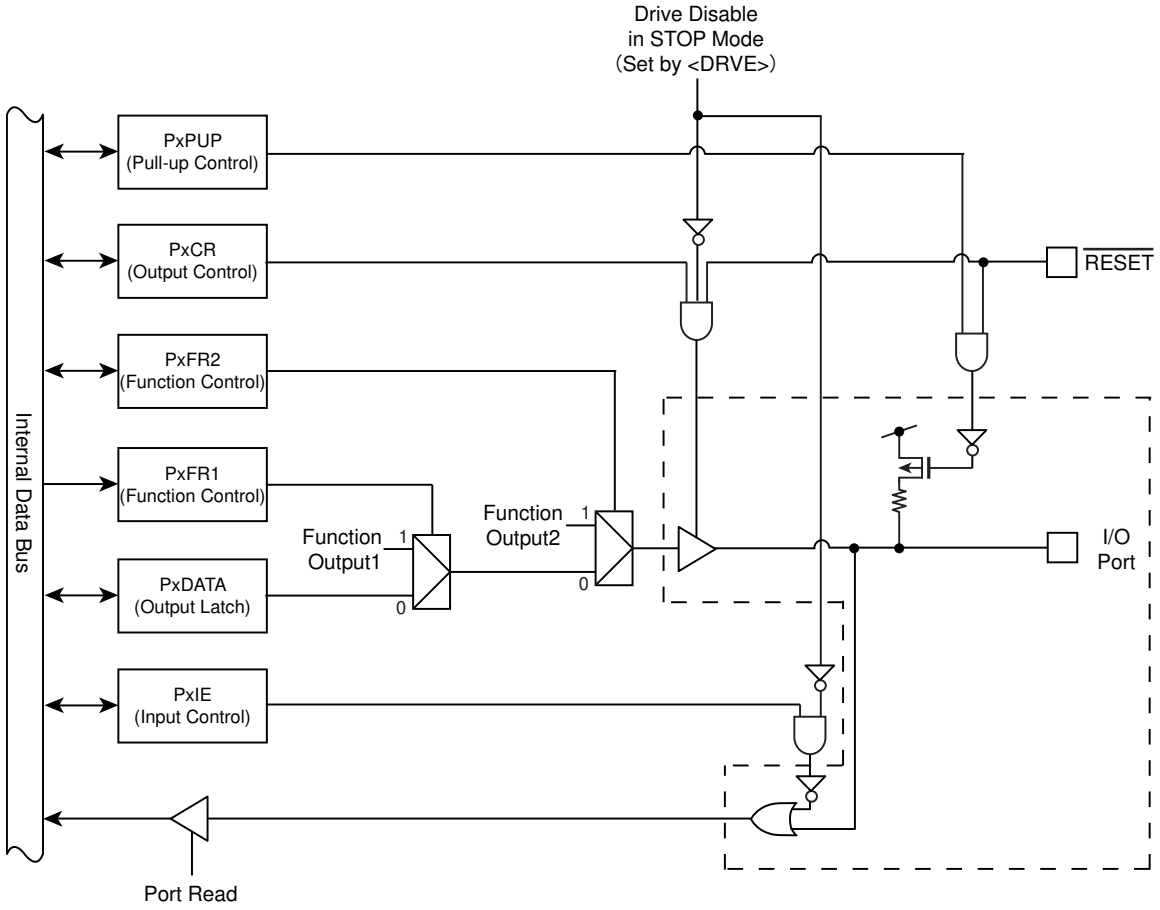


Figure 8-15 Port Type T15

8.3.17 Type T16

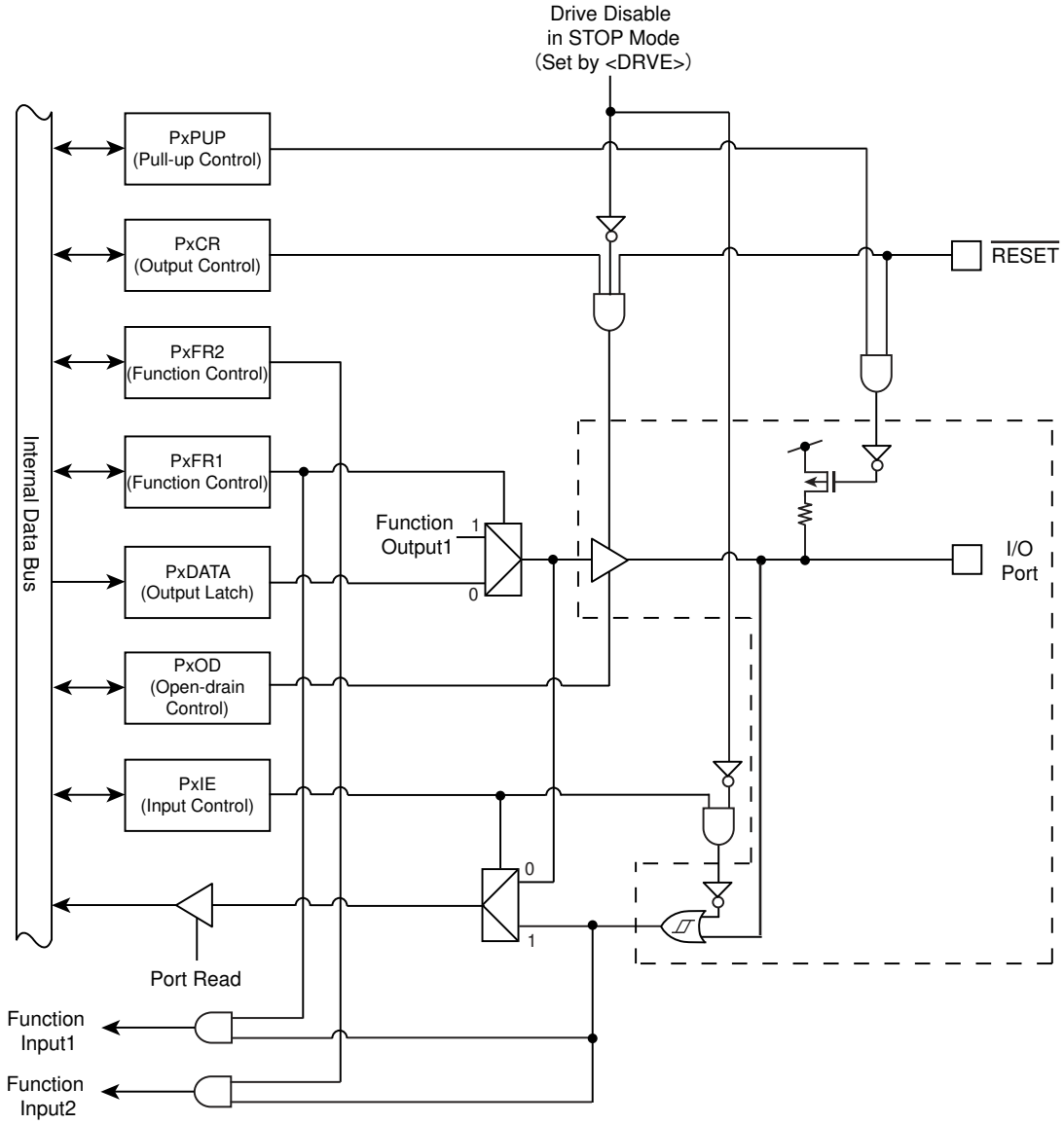


Figure 8-16 Port Type T16

8.3.18 Type T17

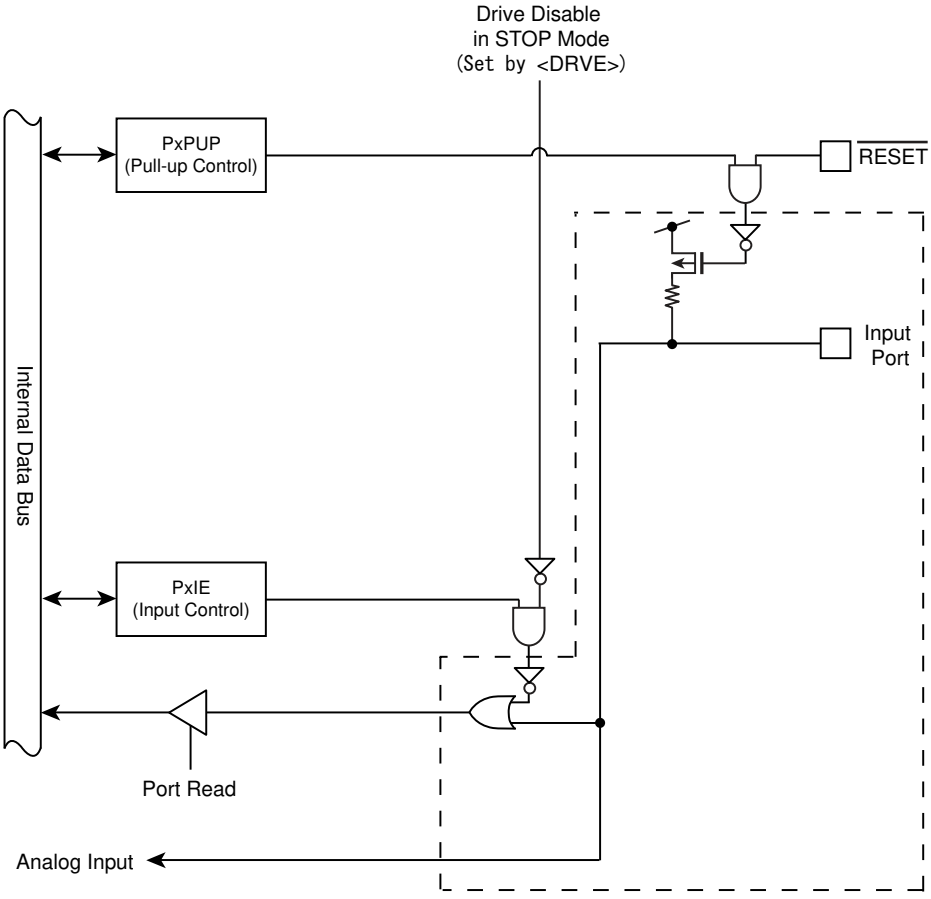


Figure 8-17 Port Type T17

8.3.19 Type T18

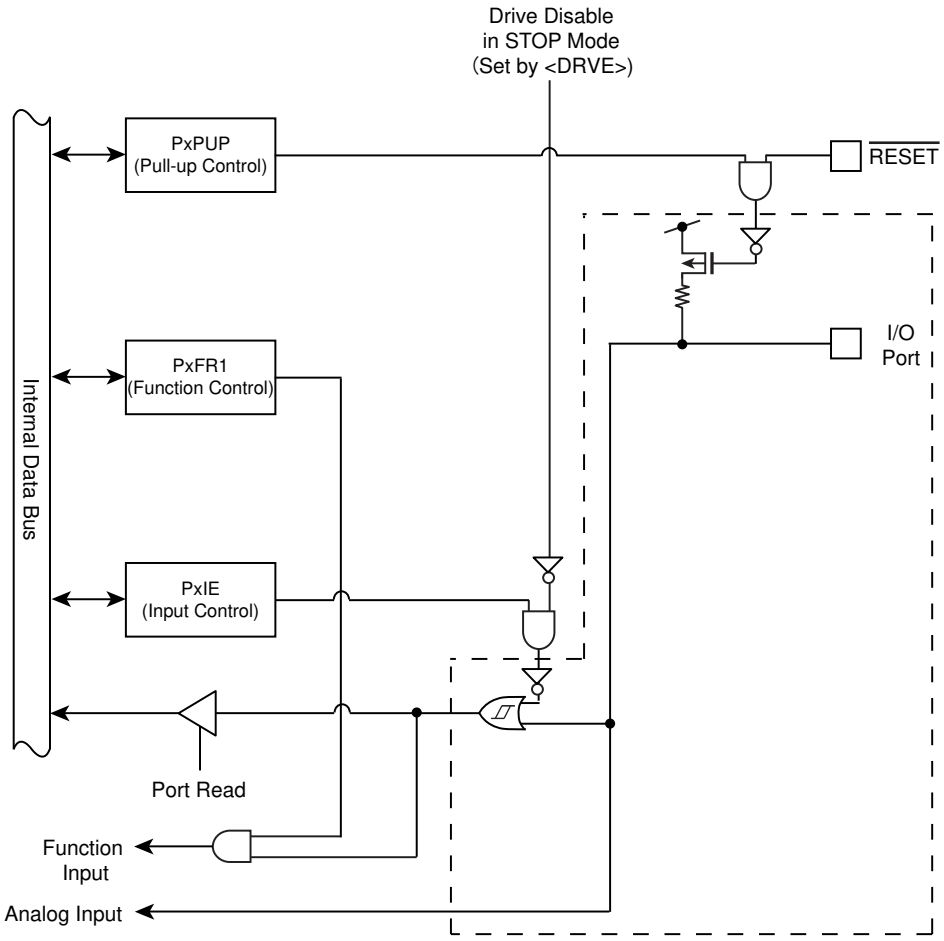


Figure 8-18 Port TypeT18

8.4 Appendix (Port setting List)

The following table shows the register setting for each function.

Initialization of the ports where the [・] does not exist in the "After reset" field is set to "0" for all register settings.

Setting for the bit "x" can be arbitrarily-specified.

8.4.1 Port A Setting

Table 8-6 Port Setting List (Port A)

Pin	Port Type	Function	After reset	PACR	PAFR1	PAPUP	PAPDN	PAIE
PA0	T12	Input Port		0	0	x	0	1
		Output Port		1	0	x	0	0
		TMS(Input)/ SWDIO(I/O)	・	1	1	1	0	1
PA1	T6	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
		TCK(Input)/ SWCLK(Input)	・	0	1	0	1	1
PA2	T9	Input Port		0	0	x	0	1
		Output Port		1	0	x	0	0
		TRACECLK(Output)		1	1	x	0	0
PA3	T9	Input Port		0	0	x	0	1
		Output Port		1	0	x	0	0
		TRACEDATA0(Output)		1	1	x	0	0
PA4	T9	Input Port		0	0	x	0	1
		Output Port		1	0	x	0	0
		TRACEDATA1(Output)		1	1	x	0	0
PA5	T9	Input Port		0	0	x	0	1
		Output Port		1	0	x	0	0
		TRACEDATA2(Output)		1	1	x	0	0
PA6	T9	Input Port		0	0	x	0	1
		Output Port		1	0	x	0	0
		TRACEDATA3(Output)		1	1	x	0	0
PA7	T1	Input Port		0	0	x	0	1
		Output Port		1	0	x	0	0

8.4.2 Port B Setting

Table 8-7 Port Setting List (Port B)

Pin	Port Type	Function	After re-set	PBCR	PBFR1	PBPUP	PBIE
PB0	T11	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TDO(Output)/ SWV(Output)	-	1	1	0	0
PB1	T2	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TDI(Input)	-	0	1	1	1
PB2	T2	Input Port		0	0	x	1
		Output Port		1	0	x	0
		$\overline{\text{TRST}}$(Input)	-	0	1	1	1
PB3	T1	Input Port		0	0	x	1
		Output Port		1	0	x	0
PB4	T1	Input Port		0	0	x	1
		Output Port		1	0	x	0
PB5	T1	Input Port		0	0	x	1
		Output Port		1	0	x	0
PB6	T1	Input Port		0	0	x	1
		Output Port		1	0	x	0
PB7	T1	Input Port		0	0	x	1
		Output Port		1	0	x	0

8.4.3 Port C Setting

Table 8-8 Port Setting List (Port C)

Pin	Port Type	Function	After re-set	PCPUP	PCIE
PC0	T17	Input Port		x	1
		Analog Input	-	0	0
PC1	T17	Input Port		x	1
		Analog Input	-	0	0
PC2	T17	Input Port		x	1
		Analog Input	-	0	0
PC3	T17	Input Port		x	1
		Analog Input	-	0	0

8.4.4 Port D Setting

Table 8-9 Port Setting List (Port D)

Pin	Port Type	Function	After re-set	PDFR1	PDPUP	PDIE
PD0	T18	Input Port		0	x	1
		TB5IN0(Input)		1	x	1
		Analog Input	-	x	0	0
PD1	T18	Input Port		0	x	1
		TB5IN1(Input)		1	x	1
		Analog Input	-	x	0	0
PD2	T18	Input Port		0	x	1
		TB6IN0(Input)		1	x	1
		Analog Input	-	x	0	0
PD3	T18	Input Port		0	x	1
		TB6IN1(Input)		1	x	1
		Analog Input	-	x	0	0
PD4	T17	Input Port		0	x	1
		Analog Input	-	x	0	0
PD5	T17	Input Port		0	x	1
		Analog Input	-	x	0	0
PD6	T17	Input Port		0	x	1
		Analog Input	-	x	0	0
PD7	T17	Input Port		0	x	1
		Analog Input	-	x	0	0

8.4.5 Port E Setting

Table 8-10 Port Setting List (Port E)

Pin	Port Type	Function	After re-set	PECR	PEFR1	PEFR2	PEOD	PEPUP	PEIE
PE0	T10	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		TXD0(Output)		1	1	0	x	x	0
PE1	T4	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		RXD0(Input)		0	1	0	x	x	1
PE2	T16	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		SCLK0(Input)		0	1	0	x	x	1
		SCLK0(Output)		1	1	0	x	x	0
		$\overline{\text{CTS0}}$(Input)		0	0	1	x	x	1
PE3	T4	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		RXIN0(Input)		0	1	0	x	x	1
PE4	T10	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		TXD1(Output)		1	1	0	x	x	0
PE5	T4	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		RXD1(Input)		0	1	0	x	x	1
PE6	T16	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		SCLK1(Input)		0	1	0	x	x	1
		SCLK1(Output)		1	1	0	x	x	0
		$\overline{\text{CTS1}}$(Input)		0	0	1	x	x	1

8.4.6 Port F Setting

Table 8-11 Port Setting List (Port F)

Pin	Port Type	Function	After re-set	PFCR	PFFR1	PFFR2	PFOD	PFPUP	PFIE
PF0	T10	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		TXD2(Output)		1	1	0	x	x	0
PF1	T4	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		RXD2(Input)		0	1	0	x	x	1
PF2	T16	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		SCLK2(Input)		0	1	1	x	x	1
		SCLK2(Output)		1	1	0	x	x	0
		$\overline{\text{CTS2}}$(Input)		0	0	0	x	x	1
PF3	T4	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		RXIN1(Input)		0	1	0	x	x	1
PF4	T13	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		SO1(Output)		1	1	0	x	x	0
		SDA1(Input/Output)		1	1	0	1	x	1
PF5	T13	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		SI1(Input)		0	1	0	x	x	1
		SCL1(Input/Output)		1	1	0	1	x	1
PF6	T13	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		SCK1(Input)		0	1	0	x	x	1
		SCK1(Output)		1	1	0	x	x	0
PF7	T8	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		INT5(Input)		0	1	0	x	x	1

8.4.7 Port G Setting

Table 8-12 Port Setting List (Port G)

Pin	Port Type	Function	After re-set	PGCR	PGFR1	PGOD	PGPUP	PGIE
PG0	T13	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		SO0(Output)		1	1	x	x	0
		SDA0(Input/Output)		1	1	1	x	1
PG1	T13	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		SI0(Input)		0	1	x	x	1
		SCL0(Input/Output)		1	1	1	x	1
PG2	T13	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		SCK0(Input)		0	1	x	x	1
		SCK0(Output)		1	1	x	x	0
PG3	T8	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		INT4(Input)		0	1	x	x	1
PG4	T13	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		SO2(Output)		1	1	x	x	0
		SDA2(Input/Output)		1	1	1	x	1
PG5	T13	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		SI2(Input)		0	1	x	x	1
		SCL2(Input/Output)		1	1	1	x	1
PG6	T13	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		SCK2(Input)		0	1	x	x	1
		SCK2(Output)		1	1	x	x	0
PG7	T10	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		TB8OUT(Output)		1	1	x	x	0

8.4.8 Port H Setting

Table 8-13 Port Setting List (Port H)

Pin	Port Type	Function	After re-set	PHCR	PHFR1	PHPUP	PHIE
PH0	T5	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB0IN0(Input)		0	1	x	1
PH1	T3	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB0IN1(Input)		0	1	x	1
PH2	T3	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB1IN0(Input)		0	1	x	1
PH3	T3	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB1IN1(Input)		0	1	x	1
PH4	T3	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB2IN0(Input)		0	1	x	1
PH5	T3	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB2IN1(Input)		0	1	x	1
PH6	T3	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB3IN0(Input)		0	1	x	1
PH7	T3	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB3IN1(Input)		0	1	x	1

Note: The PH0 input and pull-up are enabled and act as $\overline{\text{BOOT}}$ input pin while a $\overline{\text{RESET}}$ is in "Low" state.

8.4.9 Port I Setting

Table 8-14 Port Setting List (Port I)

Pin	Port Type	Function	After re-set	PICR	PIFR1	PIPUP	PIIE
PI0	T9	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB0OUT(Output)		1	1	x	0
PI1	T9	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB1OUT(Output)		1	1	x	0
PI2	T9	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB2OUT(Output)		1	1	x	0
PI3	T9	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB3OUT(Output)		1	1	x	0
PI4	T9	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB4OUT(Output)		1	1	x	0
PI5	T9	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB5OUT(Output)		1	1	x	0
PI6	T3	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB4IN0(Input)		0	1	x	1
PI7	T3	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB4IN1(Input)		0	1	x	1

8.4.10 Port J Setting

Table 8-15 Port Setting List (Port J)

pin	Port Type	Function	After re-set	PJCR	PJFR1	PJPUP	PJIE
PJ0	T7	Input Port		0	0	x	1
		Output Port		1	0	x	0
		INT0(Input)		0	1	x	1
PJ1	T7	Input Port		0	0	x	1
		Output Port		1	0	x	0
		INT1(Input)		0	1	x	1
PJ2	T7	Input Port		0	0	x	1
		Output Port		1	0	x	0
		INT2(Input)		0	1	x	1
PJ3	T7	Input Port		0	0	x	1
		Output Port		1	0	x	0
		INT3(Input)		0	1	x	1
PJ4	T9	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB6OUT(Output)		1	1	x	0
PJ5	T9	Input Port		0	0	x	1
		Output Port		1	0	x	0
		TB7OUT(Output)		1	1	x	0
PJ6	T7	Input Port		0	0	x	1
		Output Port		1	0	x	0
		INT6(Input)		0	1	x	1
PJ7	T7	Input Port		0	0	x	1
		Output Port		1	0	x	0
		INT7(Input)		0	1	x	1

8.4.11 Port K Setting

Table 8-16 Port Setting List (Port K)

Pin	Port Type	Function	After re-set	PKCR	PKFR1	PKFR2	PKPUP	PKIE
PK0	T14	Input Port		0	0	0	0	1
		Output Port		1	0	0	0	0
		CEC(Input/Output)		1	1	0	0	1
PK1	T15	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
		SCOUT(Output)		1	1	0	x	0
		$\overline{\text{ALARM}}$(Output)		1	0	1	x	0
PK2	T9	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
		TB9OUT(Output)		1	1	0	x	0

Note: PK0 is an N-ch open drain port.

9. 16-bit Timer/Event Counters(TMRB)

9.1 Outline

TMRB operate in the following four operation modes:

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable pulse generation mode (PPG)
- Timer synchronous mode

The use of the capture function allows TMRB to perform the following three measurements.

- Frequency measurement
- Pulse width measurement
- Time difference measurement

In the following explanation of this section, "x" indicates a channel number.

9.2 Differences in the Specifications

TMPM330FDFG/FYFG/FWFG contains 10-channel of TMRB.

Each channel functions independently and the channels operate in the same way except for the differences in their specification as shown in Table 9-1.

Some of the channels can put the capture trigger and the synchronous start trigger on other channels.

1. The flip-flop output of TMRB 7 through TMRB 9 can be used as the capture trigger of other channels.
   - TB7OUT → available for TMRB0 through TMRB1
   - TB8OUT → available for TMRB2 through TMRB4
   - TB9OUT → available for TMRB5 through TMRB6
2. The start trigger of the timer synchronous mode (with TBxRUN)
   - TMRB0 → can start TMRB0 through TMRB3 synchronously
   - TMRB4 → can start TMRB4 through TMRB7 synchronously

Table 9-1 Differences in the Specifications of TMRB Modules

Specification / Channel	External pins				Trigger function between timers		Interrupt	
	External clock/ capture trigger input pins		Timer flip-flop output pin		Capture trigger	Synchronous start trigger channel	Capture interrupt	TMRB interrupt
	Signal	Port (Pin number)	Signal	Port (Pin number)				
TMRB0	TB0IN0 TB0IN1	PH0 (30) PH1 (31)	TB0OUT	PI0 (38)	TB7OUT	−	INTCAP00 INTCAP01	INTTB0
TMRB1	TB1IN0 TB1IN1	PH2 (32) PH3 (36)	TB1OUT	PI1 (40)	TB7OUT	TMRB0	INTCAP10 INTCAP11	INTTB1
TMRB2	TB2IN0 TB2IN1	PH4 (9) PH5 (10)	TB2OUT	PI2 (42)	TB8OUT	TMRB0	INTCAP20 INTCAP21	INTTB2
TMRB3	TB3IN0 TB3IN1	PH6 (84) PH7 (85)	TB3OUT	PI3 (48)	TB8OUT	TMRB0	INTCAP30 INTCAP31	INTTB3
TMRB4	TB4IN0 TB4IN1	PI6 (79) PI7 (83)	TB4OUT	PI4 (52)	TB8OUT	−	INTCAP40 INTCAP41	INTTB4
TMRB5	TB5IN0 TB5IN1	PD0 (95) PD1 (96)	TB5OUT	PI5 (53)	TB9OUT	TMRB4	INTCAP50 INTCAP51	INTTB5
TMRB6	TB6IN0 TB6IN1	PD2 (97) PD3 (98)	TB6OUT	PJ4 (88)	TB9OUT	TMRB4	INTCAP60 INTCAP61	INTTB6
TMRB7	−	−	TB7OUT	PJ5 (8)	−	TMRB4	−	INTTB7
TMRB8	−	−	TB8OUT	PG7 (11)	−	−	−	INTTB8
TMRB9	−	−	TB9OUT	PK2 (7)	−	−	−	INTTB9

9.3 Configuration

Each channel consists of a 16-bit up-counter, two 16-bit timer registers (double-buffered), 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.

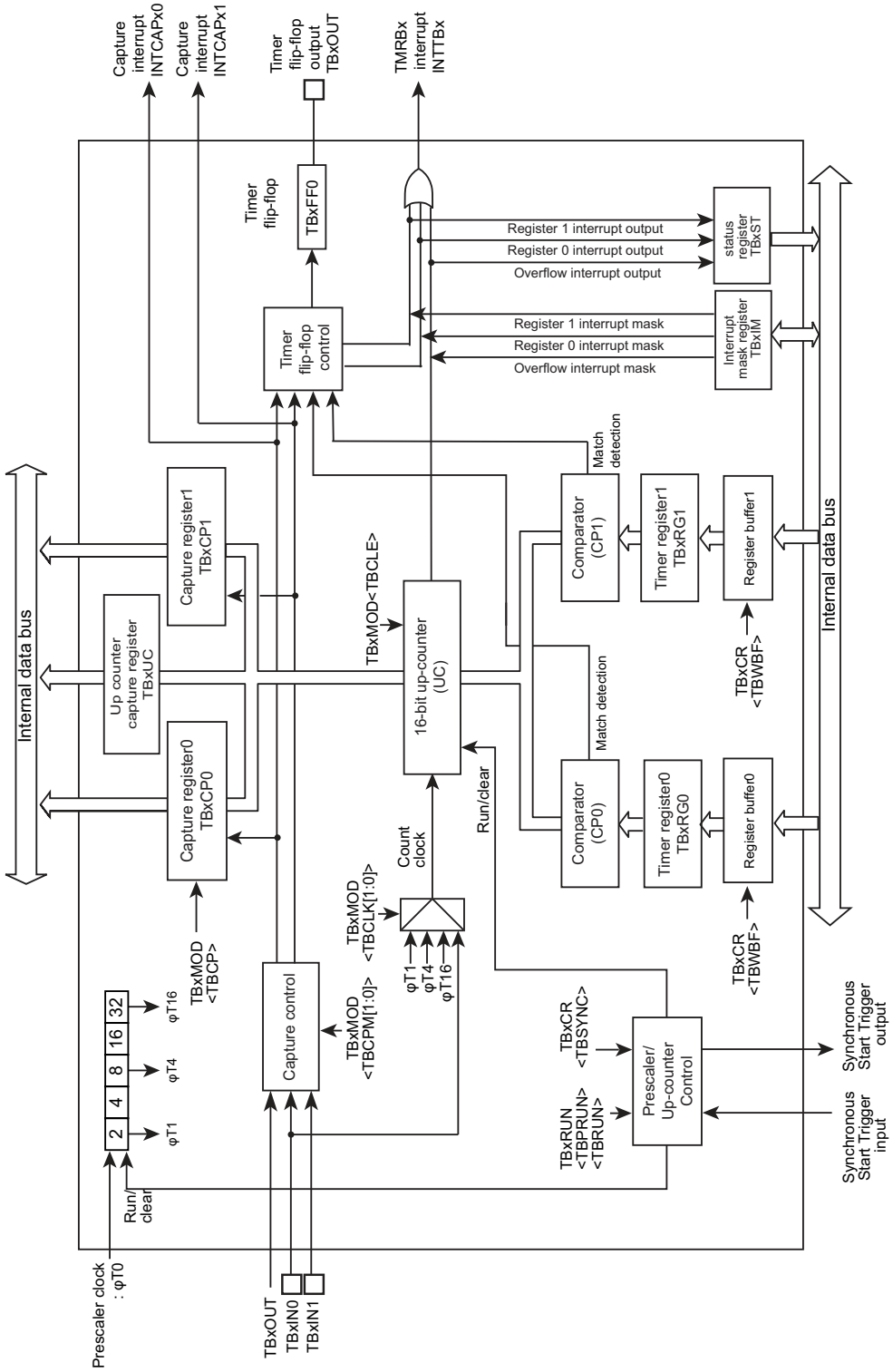


Figure 9-1 TMRBx Block Diagram(x= 0 to 9)

9.4 Registers

9.4.1 Register list according to channel

The following table shows the register names and addresses of each channel.

Channel x	Base Address
Channel0	0x4001_0000
Channel1	0x4001_0040
Channel2	0x4001_0080
Channel3	0x4001_00C0
Channel4	0x4001_0100
Channel5	0x4001_0140
Channel6	0x4001_0180
Channel7	0x4001_01C0
Channel8	0x4001_0200
Channel9	0x4001_0240

Register name(x=0 to 9)		Address(Base+)
Enable register	TBxEN	0x0000
RUN register	TBxRUN	0x0004
Control register	TBxCR	0x0008
Mode register	TBxMOD	0x000C
Flip-flop control register	TBxFFCR	0x0010
Status register	TBxST	0x0014
Interrupt mask register	TBxIM	0x0018
Up counter capture register	TBxUC	0x001C
Timer register 0	TBxRG0	0x0020
Timer register 1	TBxRG1	0x0024
Capture register 0	TBxCP0	0x0028
Capture register 1	TBxCP1	0x002C

9.4.2 TBxEN (Enable register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBEN	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	TBEN	R/W	TMRBx operation 0: Disabled 1: Enabled 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 except TBxEN register.) 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.
6-0	−	R	Read as 0.

9.4.3 TBxRUN(RUN register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	TBPRUN	-	TBRUN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	TBPRUN	R/W	Prescaler operation 0: Stop & clear 1: Count
1	−	R	Read as 0.
0	TBRUN	R/W	Count operation 0: Stop & clear 1: Count

9.4.4 TBxCR(Control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBWBF	-	TBSYNC	-	I2TB	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	TBWBF	R/W	Double Buffer 0: Disabled 1: Enabled
6	−	R/W	Write 0.
5	TBSYNC	R/W	Synchronous mode switching 0: individual (unit of channel) 1: synchronous
4	−	R	Read as 0.
3	I2TB	R/W	Operation at IDLE mode 0: Stop 1:Operation
2-0	−	R	Read as 0.

9.4.5 TBxMOD(Mode register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	TBCP	TBCPM		TBCLE	TBCLK	
After reset	0	0	1	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	−	R/W	Write 0.
5	TBCP	W	Capture control by software 0: Capture by software 1: Don't care When "0" is written, the capture register 0 (TBxCP0) takes count value. Read as 1.
4-3	TBCPM[1:0]	R/W	Capture timing 00: Disable Capture timing 01: TBxIN0↑ TBxIN1↑ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN0 pin input. Takes count values into capture register 1 (TBxCP1) upon rising of TBxIN1 pin input. 10: TBxIN0↑ TBxIN0↓ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN0 pin input. Takes count values into capture register 1 (TBxCP1) upon falling of TBxIN0 pin input. 11: TBxOUT↑ TBxOUT↓ Takes count values into capture register 0 (TBxCP0) upon rising of 16-bit timer match output (TBxOUT) and into capture register 1 (TBxCP1) upon falling of TBxOUT. (TMRB0 and TMRB1:TB7OUT, TMRB2 through TMRB4:TB8OUT, TMRB5 and TMRB6:TB9OUT).
2	TBCLE	R/W	Up-counter control 0: Disables clearing of the up-counter. 1: Enables clearing of the up-counter. Clears and controls the up-counter. When "0" is written, it disables clearing of the up-counter. When "1" is written, it clears up counter when there is a match with Timer Regsiter1 (TBxRG1).
1-0	TBCLK[1:0]	R/W	Selects the TMRBx source clock. 00: TBxIN0 pin input 01: φT1 10: φT4 11: φT16

9.4.6 TBxFFCR(Flip-flop control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	TBC1T1	TBC0T1	TBE1T1	TBE0T1	TBFF0C	
After reset	1	1	0	0	0	0	1	1

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-6	−	R	Read as 1.
5	TBC1T1	R/W	TBxFF0 reverse trigger when the up-counter value is taken into the TBxCP1. 0: Disable trigger 1: Enable trigger By setting "1", the timer-flip-flop reverses when the up-counter value is taken into the Capture register 1 (TBxCP1).
4	TBC0T1	R/W	TBxFF0 reverse trigger when the up-counter value is taken into the TBxCP0. 0: Disable trigger 1: Enable trigger By setting "1", the timer-flip-flop reverses when the up-counter value is taken into the Capture register 0 (TBxCP0).
3	TBE1T1	R/W	TBxFF0 reverse trigger when the up-counter value is matched with TBxRG1. 0: Disable trigger 1: Enable trigger By setting "1", the timer-flip-flop reverses when the up-counter value is matched with the Timer register 1 (TBxRG1).
2	TBE0T1	R/W	TBxFF0 reverse trigger when the up-counter value is matched with TBxRG0. 0: Disable trigger 1: Enable trigger By setting "1", the timer-flip-flop reverses when an up-counter value is matched with the Timer register 0 (TBxRG0).
1-0	TBFF0C[1:0]	R/W	TBxFF0 control 00: Invert Reverses the value of TBxFF0 (reverse by using software). 01: Set Sets TBxFF0 to "1". 10: Clear Clears TBxFF0 to "0". 11: Don't care * This is always read as "11".

9.4.7 TBxST(Status register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	INTTBOF	INTTB1	INTTB0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	INTTBOF	R	Overflow flag 0:No overflow occurs 1:Overflow occurs When an up-counter is overflow, "1" is set.
1	INTTB1	R	Match flag (TBxRG1) 0:No detection of a mach 1:Detects a match with TBxRG1 When a match with the timer register 1 (TBxRG1) is detected,"1" is set.
0	INTTB0	R	Match flag (TBxRG0) 0:No match is detected 1:Detects a match with TBxRG0 When a match with the timer register 0 (TBxRG0) is detected, "1" is set.

Note 1: The factors only which is not masked by TBxIM output interrupt request to the CPU.Even if the mask setting is done, the flag is set.

Note 2: The flag is cleared by reading the TBxST register.To clear the flag, TBxST register should be read.

9.4.8 TBxIM(Interrupt mask register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	TBIMOF	TBIM1	TBIM0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	TBIMOF	R/W	Overflow interrupt mask 0:Disable 1:Enable Sets the up-counter overflow interrupt to disable or enable.
1	TBIM1	R/W	Match interrupt mask (TBxRG1) 0:Disable 1:Enable Sets the match interrupt mask with the Timer register 1 (TBxRG1) to enable or disable.
0	TBIM0	R/W	Match interrupt mask (TBxRG0) 0:Disable 1:Enable Sets the match interrupt mask with the Timer register 0 (TBxRG0) to enable or disable.

9.4.9 TBxUC(Up counter capture register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TBUC							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBUC							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-0	TBUC[15:0]	R	Captures a value by reading up-counter out. If TBxUC is read, current up-counter value can be captured.

9.4.10 TBxRG0(Timer register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TBRG0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBRG0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-0	TBRG0[15:0]	R/W	Sets a value comparing to the up-counter.

9.4.11 TBxRG1(Timer register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TBRG1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBRG1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-0	TBRG1[15:0]	R/W	Sets a value comparing to the up-counter.

9.4.12 TBxCP0(Capture register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TBCP0							
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	TBCP0							
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-0	TBCP0[15:0]	R	A value captured from the up-counter is read.

9.4.13 TBxCP1(Capture register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TBCP1							
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	TBCP1							
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-0	TBCP1[15:0]	R	A value captured from the up-counter is read.

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 9-1 .

9.5.1 Prescaler

There is a 4-bit prescaler to generate the source clock for up-counter UC.

The prescaler input clock φT0 is fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16 or fperiph/32 selected by CGSYSCR<PRCK[2:0]> 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 TBxRUN<TBPRUN> where writing "1" starts counting and writing "0" clears and stops counting. Table 9-2 and Table 9-3 show prescaler output clock resolutions.

Table 9-2 Prescaler Output Clock Resolutions(fc = 40MHz)

Select peripheral clock CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
0 (fgear)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.05 μs)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)
		001 (fperiph/2)	$fc/2^2$ (0.1 μs)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)
		010 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	100 (fc/2)	000 (fperiph/1)	$fc/2^2$ (0.1 μs)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)
		001 (fperiph/2)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		010 (fperiph/4)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		011 (fperiph/8)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		100 (fperiph/16)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		101 (fperiph/32)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
	101 (fc/4)	000 (fperiph/1)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		001 (fperiph/2)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		010 (fperiph/4)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		011 (fperiph/8)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		100 (fperiph/16)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		101 (fperiph/32)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)
	110 (fc/8)	000 (fperiph/1)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		001 (fperiph/2)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		010 (fperiph/4)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		011 (fperiph/8)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		100 (fperiph/16)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)
		101 (fperiph/32)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{13}$ (204.8 μs)

Table 9-2 Prescaler Output Clock Resolutions(fc = 40MHz)

Select peripheral clock CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
1 (fc)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.05 μs)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)
		001 (fperiph/2)	$fc/2^2$ (0.1 μs)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)
		010 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	100 (fc/2)	000 (fperiph/1)	−	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)
		001 (fperiph/2)	$fc/2^2$ (0.1 μs)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)
		010 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	101 (fc/4)	000 (fperiph/1)	−	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)
		001 (fperiph/2)	−	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)
		010 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	110 (fc/8)	000 (fperiph/1)	−	−	$fc/2^5$ (0.8 μs)
		001 (fperiph/2)	−	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)
		010 (fperiph/4)	−	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)

Note 1: The prescaler output clock φTn must be selected so that φTn < fsys is satisfied (so that φTn is slower than fsys).

Note 2: Do not change the clock gear while the timer is operating.

Note 3: "−" denotes a setting prohibited.

Table 9-3 Prescaler Output Clock Resolutions(fc = 32MHz)

Select peripheral clock CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
0 (fgear)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.0625 μs)	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)
		001 (fperiph/2)	$fc/2^2$ (0.125 μs)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)
		010 (fperiph/4)	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
		011 (fperiph/8)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)
		100 (fperiph/16)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16.0 μs)
		101 (fperiph/32)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)	$fc/2^{10}$ (32.0 μs)
	100 (fc/2)	000 (fperiph/1)	$fc/2^2$ (0.125 μs)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)
		001 (fperiph/2)	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
		010 (fperiph/4)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)
		011 (fperiph/8)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16.0 μs)
		100 (fperiph/16)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)	$fc/2^{10}$ (32.0 μs)
		101 (fperiph/32)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16.0 μs)	$fc/2^{11}$ (64.0 μs)
	101 (fc/4)	000 (fperiph/1)	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
		001 (fperiph/2)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)
		010 (fperiph/4)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16.0 μs)
		011 (fperiph/8)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)	$fc/2^{10}$ (32.0 μs)
		100 (fperiph/16)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16.0 μs)	$fc/2^{11}$ (64.0 μs)
		101 (fperiph/32)	$fc/2^8$ (8.0 μs)	$fc/2^{10}$ (32.0 μs)	$fc/2^{12}$ (128.0 μs)
	110 (fc/8)	000 (fperiph/1)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)
		001 (fperiph/2)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16.0 μs)
		010 (fperiph/4)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)	$fc/2^{10}$ (32.0 μs)
		011 (fperiph/8)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16.0 μs)	$fc/2^{11}$ (64.0 μs)
		100 (fperiph/16)	$fc/2^8$ (8.0 μs)	$fc/2^{10}$ (32.0 μs)	$fc/2^{12}$ (128.0 μs)
		101 (fperiph/32)	$fc/2^9$ (16.0 μs)	$fc/2^{11}$ (64.0 μs)	$fc/2^{13}$ (256.0 μs)

Table 9-3 Prescaler Output Clock Resolutions(fc = 32MHz)

Select peripheral clock CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
1 (fc)	000 (fc)	000 (fperiph/1)	fc/2^1 (0.0625 μs)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)
		001 (fperiph/2)	fc/2^2 (0.125 μs)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)
		010 (fperiph/4)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)
		011 (fperiph/8)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)
		100 (fperiph/16)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		101 (fperiph/32)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)
	100 (fc/2)	000 (fperiph/1)	−	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)
		001 (fperiph/2)	fc/2^2 (0.125 μs)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)
		010 (fperiph/4)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)
		011 (fperiph/8)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)
		100 (fperiph/16)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		101 (fperiph/32)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)
	101 (fc/4)	000 (fperiph/1)	−	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)
		001 (fperiph/2)	−	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)
		010 (fperiph/4)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)
		011 (fperiph/8)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)
		100 (fperiph/16)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		101 (fperiph/32)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)
	110 (fc/8)	000 (fperiph/1)	−	−	fc/2^5 (1.0 μs)
		001 (fperiph/2)	−	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)
		010 (fperiph/4)	−	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)
		011 (fperiph/8)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)
		100 (fperiph/16)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		101 (fperiph/32)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)

Note 1: The prescaler output clock φTn must be selected so that φTn < fsys is satisfied (so that φTn is slower than fsys).

Note 2: Do not change the clock gear while the timer is operating.

Note 3: "−" denotes a setting prohibited.

9.5.2 Up-counter (UC)

UC is a 16-bit binary counter.

- Source clock

  UC source clock, specified by TBxMOD<TBCLK[1:0]>, can be selected from either three types - φT1, φT4 and φT16 - of prescaler output clock or the external clock of the TBxIN0 pin.

- Count start/ stop

  Counter operation is specified by TBxRUN<TBRUN>. UC starts counting if <TBRUN> = "1", and stops counting and clears counter value if <TBRUN> = "0".

- Timing to clear UC

  1. When a match is detected

     By setting TBxMOD<TBCLE> = "1", UC is cleared if when the comparator detects a match between counter value and the value set in TBxRG1. UC operates as a free-running counter if TBxMOD<TBCLE> = "0".

  2. When UC stops

     UC stops counting and clears counter value if TBxRUN<TBRUN> = "0".

- UC overflow

  If UC overflow occurs, the INTTBx overflow interrupt is generated.

9.5.3 Timer registers (TBxRG0, TBxRG1)

TBxRG0 and TBxRG1 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 UC up-counter, it outputs the match detection signal.

TBxRG0 and TBxRG1 are consisted of the double-buffered configuration which are paired with register buffers. The double buffering is disabled in the initial state.

Controlling double buffering disable or enable is specified by TBxCR<TBWBF> bit. If <TBWBF> = "0", the double buffering becomes disable. If <TBWBF> = "1", it becomes enable. When the double buffering is enabled, a data transfer from the register buffer to the timer register (TBxRG0/1) is done in the case that UC is matched with TBxRG1.When the counter is stopped even if double buffering is enabled, the double buffering operates as a single buffer, and an immediate data can be written to the TBxRG0 and TBxRG1.

9.5.4 Capture

This is a circuit that controls the timing of latching values from the UC up-counter into the TBxCP0 and TBxCP1 capture registers. The timing with which to latch data is specified by TBxMOD<TBCPM[1:0]>.

Software can also be used to import values from the UC up-counter into the capture register; specifically, UC values are taken into the TBxCP0 capture register each time "0" is written to TBxMOD<TBCP>.

9.5.5 Capture registers (TBxCP0, TBxCP1)

This register captures an up-counter (UC) value.

9.5.6 Up-counter capture register (TBxUC)

Other than the capturing functions shown above, the current count value of the UC can be captured by reading the TBxUC registers.

9.5.7 Comparators (CP0, CP1)

This register compares with the up-counter (UC) and the value setting of the Timer Register (TBxRG0 and TBxRG1) to detect whether there is a match or not. If a match is detected, INTTBx is generated.

9.5.8 Timer Flip-flop (TBxFF0)

The timer flip-flop (TBxFF0) 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 TBxFFCR<TBC1T1, TBC0T1, TBE1T1, TBE0T1>.

The value of TBxFF0 becomes undefined after a reset. The flip-flop can be reversed by writing "00" to TBxFFCR<TBFF0C[1:0]>. It can be set to "1" by writing "01," and can be cleared to "0" by writing "10."

The value of TBxFF0 can be output to the Timer output pin (TBxOUT). If the timer output is performed, the corresponding port settings must be programmed beforehand.

9.5.9 Capture interrupt (INTCAPx0, INTCAPx1)

Interrupts INTCAPx0 and INTCAPx1 can be generated at the timing of latching values from the UC up-counter into the TBxCP0 and TBxCP1 capture registers. The interrupt timing is specified by the CPU.

9.6 Description of Operations for Each Mode

9.6.1 16-bit Interval Timer Mode

In the case of generating constant period interrupt, set the interval time to the Timer register (TBxRG1) to generate the INTTBx interrupt.

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits INTTBx interrupt by setting corresponding bit to "1".
TBxFFCR	←	X	X	0	0	0	0	1	1	Disable to TBxFF0 reverse trigger
TBxMOD	←	X	0	1	0	0	1	*	*	Changes to prescaler output clock as input clock. Specifies Capture function to disable.
							(** = 01, 10, 11)			
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a time interval. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note: X; Don't care
−; No change

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 (TBxIN0 pin input).

The up-counter counts up on the rising edge of TBxIN0 pin input. It is possible to read the count value by capturing value using software and reading the captured value.

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
PxIE[m]	←								1	Allocates corresponding port toTBxIN0.
PxFR1[m]	←								1	
TBxFFCR	←	X	X	0	0	0	0	1	1	Disables to TBxFF0 reverse trigger
TBxMOD	←	X	0	1	0	0	0	0	0	Changes toTBxIN0 as an input clock
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.
TBxMOD	←	X	0	0	0	0	0	0	0	Software capture is done.

Note 1: m: corresponding bit of port
Note 2: X; Don't care
−; No change

9.6.3 16-bit PPG (Programmable Pulse Generation) Output Mode

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 TBxOUT pin by triggering the timer flip-flop (TBxFF) to reverse when the set value of the up-counter (UC) matches the set values of the timer registers (TBxRG0 and TBxRG1). Note that the set values of TBxRG0 and TBxRG1 must satisfy the following requirement:

(Set value of TBxRG0) < (Set value of TBxRG1)

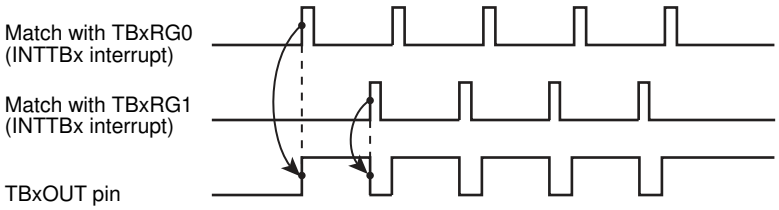


Figure 9-2 Example of Output of Programmable Pulse Generation (PPG)

In this mode, by enabling the double buffering of TBxRG0, the value of register buffer 0 is shifted into TBxRG0 when the set value of the up-counter matches the set value of TBxRG1. This facilitates handling of small duties.

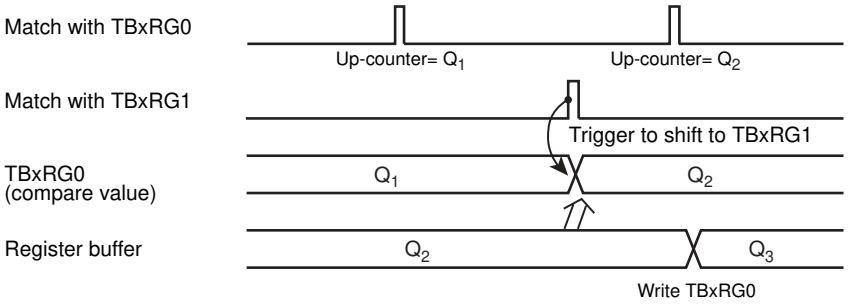


Figure 9-3 Register Buffer Operation

The block diagram of this mode is shown below.

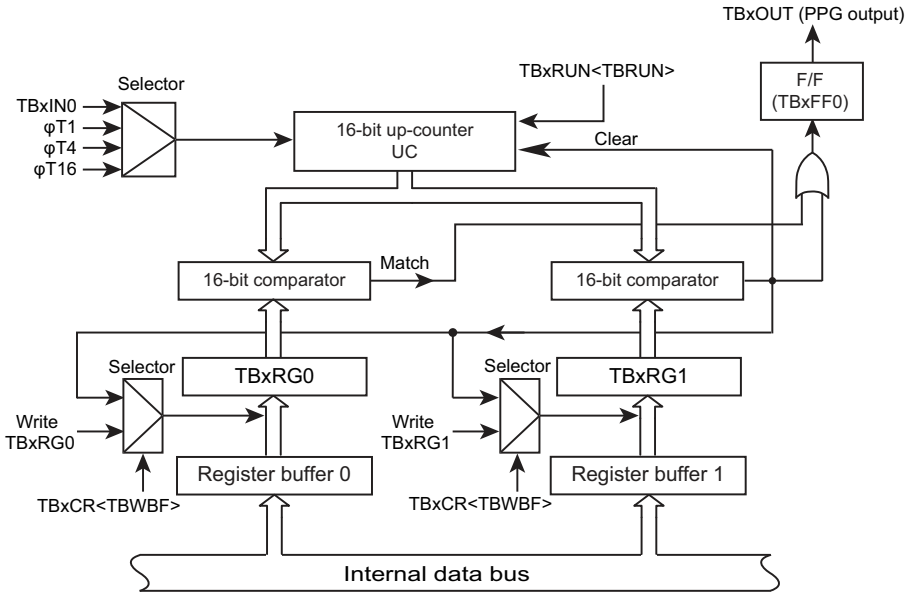


Figure 9-4 Block Diagram of 16-bit PPG Mode

Each register in the 16-bit PPG output mode must be programmed as listed below.

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
TBxCR	←	0	0	−	X	−	X	X	X	Disables double buffering.
TBxRG0	←	*	*	*	*	*	*	*	*	Specifies a duty. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a cycle. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxCR	←	1	0	X	0	0	0	0	0	Enables the TBxRG0 double buffering.
										(Changes the duty/cycle when the INTTBx interrupt is generated)
TBxFFCR	←	X	X	0	0	1	1	1	0	Specifies to trigger TBxFF0 to reverse when a match with TBxRG0 or TBxRG1 is detected,and sets the initial value of TBxFF0 to "0."
TBxMOD	←	X	0	1	0	0	1	*	*	Designates the prescaler output clock as the input clock,and disables the capture function.
							(** = 01, 10, 11)			
PxCR[m]	←								1	Allocates corresponding port to TBxOUT.
PxFR1[m]	←								1	
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note 1: m: corresponding bit of port

Note 2: X; Don't care

−; No change

9.6.4 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.

TMRB is consisted of two pairs of 4-channel TMRB. If one channel starts, remaining 3 channels can be start synchronously. In the TMPM330FDFG/FYFG/FWFG, the following combinations allow to use.

Start trigger channel (Master channel)	Synchronous operation channel (Slave channel)
TMRB0	TMRB1, TMRB2, TMRB3
TMRB4	TMRB5, TMRB6, TMRB7

Use of the timer synchronous mode is specified in TBxCR<TBSYNC> bit.

- <TBSYNC> = "0" : Timer operates individually.
- <TBSYNC> = "1" : Timers operates synchronously.

Set "0" to the <TBSYNC> bit in the master channel.

If <TBSYNC>= "1" is set in the slave channel, the start timing is synchronized with master channel start timing. Setting of start timing for TBxRUN<TBPRUN, TBRUN> bit in the slave channel is not required.

9.7 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

9.7.1 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 TBxIN0 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 (TBxCP0).

The CPU must be programmed so that an interrupt INTCAPx0 is generated at the rising of an external trigger pulse. This interrupt is used to set the timer registers (TBxRG0) to the sum of the TBxCP0 value (c) and the delay time (d), (c + d), and set the timer registers (TBxRG1) to the sum of the TBxRG0 values and the pulse width (p) of one-shot pulse, (c + d + p).[TBxRG1 change must be completed before the next match.]

In addition, the timer flip-flop control registers(TBxFFCR<TBE1T1, TBE0T1>) must be set to "11". This enables triggering the timer flip-flop (TBxFF0) to reverse when TBxUC matches TBxRG0 and TBxRG1. This trigger is disabled by the INTTBx 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 "Figure 9-5 One-shot Pulse Output (With Delay)".

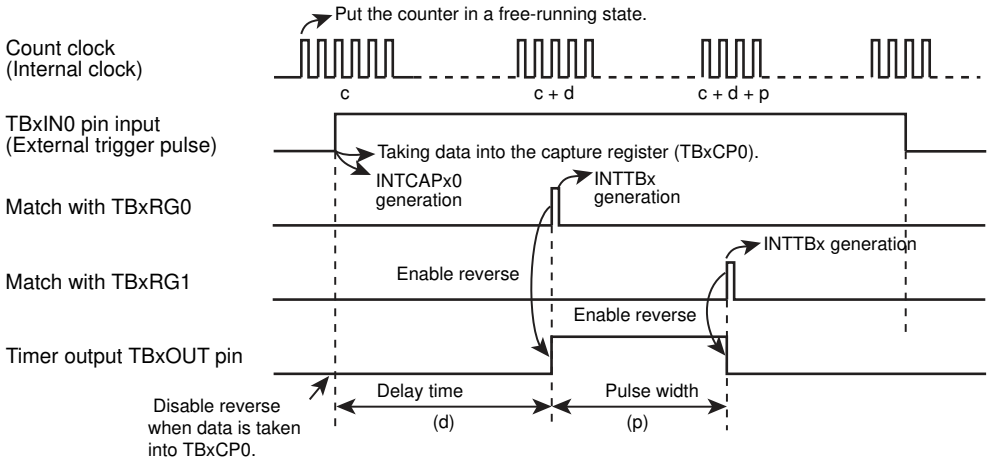


Figure 9-5 One-shot Pulse Output (With Delay)

The followings show the settings in the case that 2 ms width one-shot pulse is output after 3ms by triggering TBxIN0 input at the rising edge. (ΦT1 is selected for counting.)

Changes source clock to ΦT1. Fetches a count value into the TBxCP0 at the rising edge of TBxIN0.

		7	6	5	4	3	2	1	0	
[Main processing] Capture setting by TBxIN0										
PxIE[m]	←								1	Allocates corresponding port to TBxIN0.
PxFR1[m]	←								1	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
TBxMOD	←	X	0	1	0	1	0	0	1	Changes source clock to ΦT1. Fetches a count value into the TBxCP0 at the rising edge of TBxIN0.
TBxFFCR	←	X	X	0	0	0	0	1	0	Clears TBxFF0 reverse trigger and disables.
PxCR[m]	←								1	Allocates corresponding port toTBxOUT.
PxFR1[m]	←								1	
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits to generate interrupts specified by INTCAPx0 interrupt corresponding bit by setting to "1".
TBxRUN	←	*	*	*	*	*	1	X	1	Starts the TMRBx module.
[Processing of INTCAPx0 interrupt service routine] Pulse output setting										
TBxRG0	←	*	*	*	*	*	*	*	*	Sets count value. (TBxCAP0 + 3ms/ΦT1)
TBxRG1	←	*	*	*	*	*	*	*	*	Sets count value.(TBxCAP0 + (3+2)ms/ΦT1)
TBxFFCR	←	X	X	−	−	1	1	−	−	Reverses TBxFF0 if TBxRG0 consistent with TBxRG1.
TBxIM	←	X	X	X	X	X	1	0	1	Masks except TBxRG1 correspondence interrupt.
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits to generate interrupt specified by INTTBx interrupt corresponding bit setting to "1".
[Processing of INTTBx interrupt service routine] Output disable										
TBxFFCR	←	X	X	−	−	0	0	−	−	Clears TBxFF0 reverse trigger setting.
Interrupt enable clear register	←	*	*	*	*	*	*	*	*	Prohibits interrupts specified by INTTBx interrupt corresponding bit by setting to "1".

Note 1: m: corresponding bit of port

Note 2: X; Don't care −; No change

If a delay is not required, TBxFF0 is reversed when data is taken into TBxCP0, and TBxRG1 is set to the sum of the TBxCP0 value (c) and the one-shot pulse width (p), (c + p), by generating the INTCAPx0 interrupt. (TBxRG1 change must be completed before the next match.)

TBxFF0 is enabled to reverse when UC matches with TBxRG1, and is disabled by generating the INTTBx interrupt.

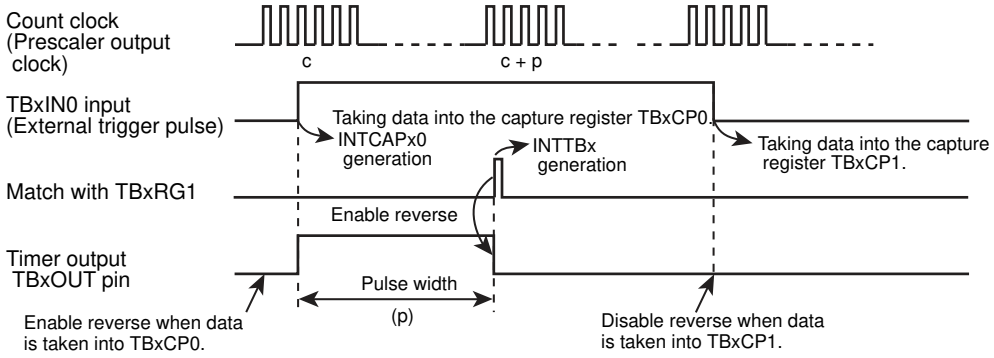


Figure 9-6 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

9.7.2 Frequency measurement

The frequency of an external clock can be measured by using the capture function.

To measure frequency, another 16-bit timer is used in combination with the 16-bit event counter mode. As an example, we explain with TMRB3 and TMRB8. TB8OUT of the 16-bit timer TMRB8 is used to specify the measurement time.

TMRB3 count clock selects TB3IN0 input and performs count operation by using external clock input. If TB3MOD<TBCPM[1:0]> is set "11", TMRB3 count clock takes the counter value into the TB3CP0 at the rising edge of TB8OUT and takes the counter value into TB3CP1 at the falling edge of TB8OUT.

This setting allows a count value of the 16-bit up-counter UC to be taken into the capture register (TB3CP0) upon rising of a timer flip-flop output (TB8OUT) of the 16-bit timer (TMRB8), and an UC counter value to be taken into the capture register (TB3CP1) upon falling of TB8OUT of the 16-bit timer (TMRB8).

A frequency is then obtained from the difference between TB3CP0 and TB3CP1 based on the measurement, by generating the INTTB8 16-bit timer interrupt.

For example, if the difference between TB3CP0 and TB3CP1 is 100 and the level width setting value of TB8OUT is 0.5 s, the frequency is 200 Hz (100 ÷ 0.5 s = 200 Hz).

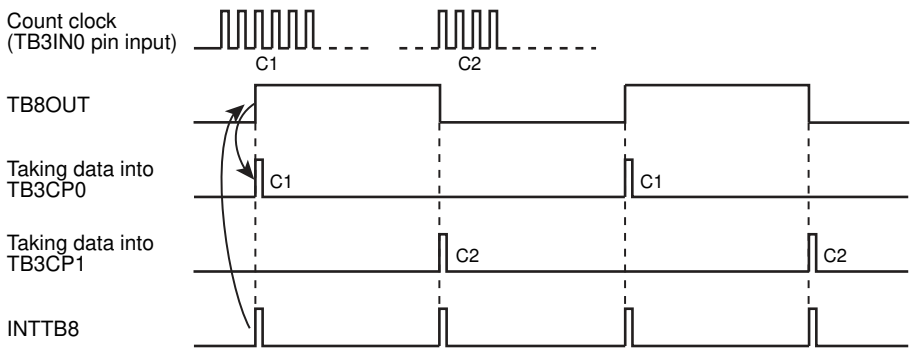


Figure 9-7 Frequency Measurement

9.7.3 Pulse width measurement

By using the capture function, the "High" 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 TBxIN0 pin and the up-counter (UC) 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 (TBxCP0, TBxCP1). The CPU must be programmed so that INTCAPx1 is generated at the falling edge of an external pulse input through the TBxIN0 pin.

The "High" level pulse width can be calculated by multiplying the difference between TBxCP0 and TBxCP1 by the clock cycle of an internal clock.

For example, if the difference between TBxCP0 and TBxCP1 is 100 and the cycle of the prescaler output clock is 0.5 μs , the pulse width is 100 × 0.5 μs = 50 μs.

Caution must be exercised when measuring pulse widths exceeding the UC maximum count time which is dependant upon the source clock used. The measurement of such pulse widths must be made using software.

The "Low" 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 INTCAPx0 interrupt processing as shown in "Figure 9-8 Pulse Width Measurement" and this difference is multiplied by the cycle of the prescaler output clock to obtain the "Low" level width.

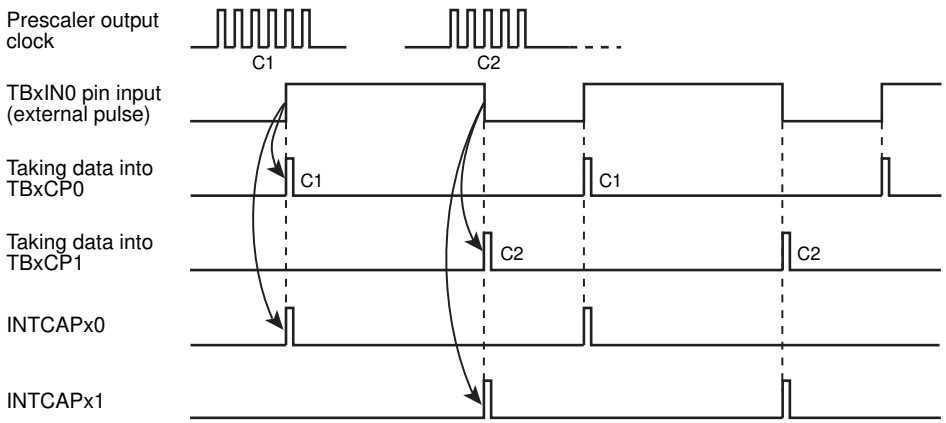


Figure 9-8 Pulse Width Measurement

9.7.4 Time Difference Measurement

The time difference of two events can be measured by the capture function. The up-counter (UC) is made to count up by putting it in a free-running state using the prescaler output clock.

The value of UC is taken into the capture register (TBxCP0) at the rising edge of the TBxIN0 pin input pulse. The CPU must be programmed to generate INTCAPx0 interrupt at this time.

The value of UC is taken into the capture register (TBxCP1) at the rising edge of the TBxIN1 pin input pulse. The CPU must be programmed to generate INTCAPx1 interrupt at this time.

The time difference can be calculated by multiplying the difference between TBxCP1 and TBxCP0 by the clock cycle of an internal clock.

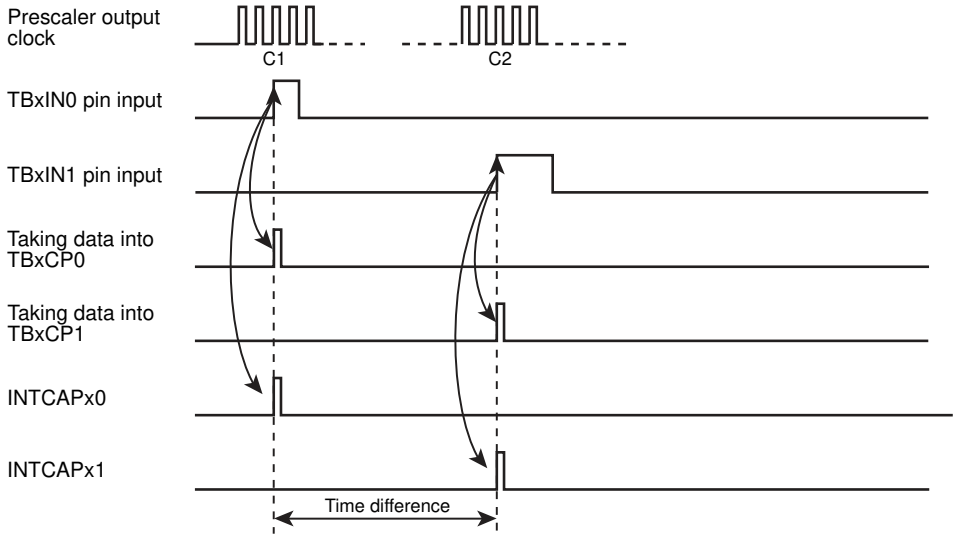


Figure 9-9 Time Difference Measurement

10. Serial Channel (SIO/UART)

10.1 Overview

This device has two mode for the serial channel, one is the synchronous communication mode (I/O interface mode), and the other is the asynchronous communication mode (UART mode).

Their features are given in the following.

- Transfer Clock
  - Dividing by the prescaler, from the peripheral clock (ΦT0) frequency into 1/2, 1/8, 1/32, 1/128.
  - Make it possible to divide from the prescaler output clock frequency into 1-16.
  - Make it possible to divide from the prescaler output clock frequency into 1, N+m/16 (N=2-15, m=1-15), 16. (only UART mode)
  - The usable system clock (only UART mode).
- Double Buffer /FIFO

  The usable double buffer function, and the usable FIFO buffers of transmit and receive in all for maximum 4-byte.
- I/O Interface Mode
  - Transfer Mode: the half duplex (transmit/receive), the full duplex
  - Clock: Output (fixed rising edge) /Input (selectable rising/falling edge)
  - Make it possible to specify the interval time of continuous transmission.
- UART Mode
  - Data length: 7 bits, 8bits, 9bits
  - Add parity bit (to be against 9bits data length)
  - Serial links to use wake-up function
  - Handshaking function with $\overline{\text{CTS}}$ pin

In the following explanation, "x" represents channel number.

10.2 Difference in the Specifications of SIO Modules

TMPM330FDFG/FYFG/FWFG has three SIO channels.

Each channel functions independently. The used pins and interrupt in each channel are collected in the following.

Table 10-1 Difference in the Specifications of SIO Modules

		Channel 0	Channel 1	Channel 2
Pin name	TXD	PE0(20pin)	PE4(23pin)	PF0(33pin)
	RXD	PE1(21pin)	PE5(24pin)	PF1(34pin)
	$\overline{\text{CTS}}$/SLCK	PE2(22pin)	PE6(25pin)	PF2(35pin)
Interrupt	Receive Interrupt	INTRX0	INTRX1	INTRX2
	Transmit Interrupt	INTTX0	INTTX1	INTTX2

10.3 Configuration

Figure 10-1 shows SIO block diagram.

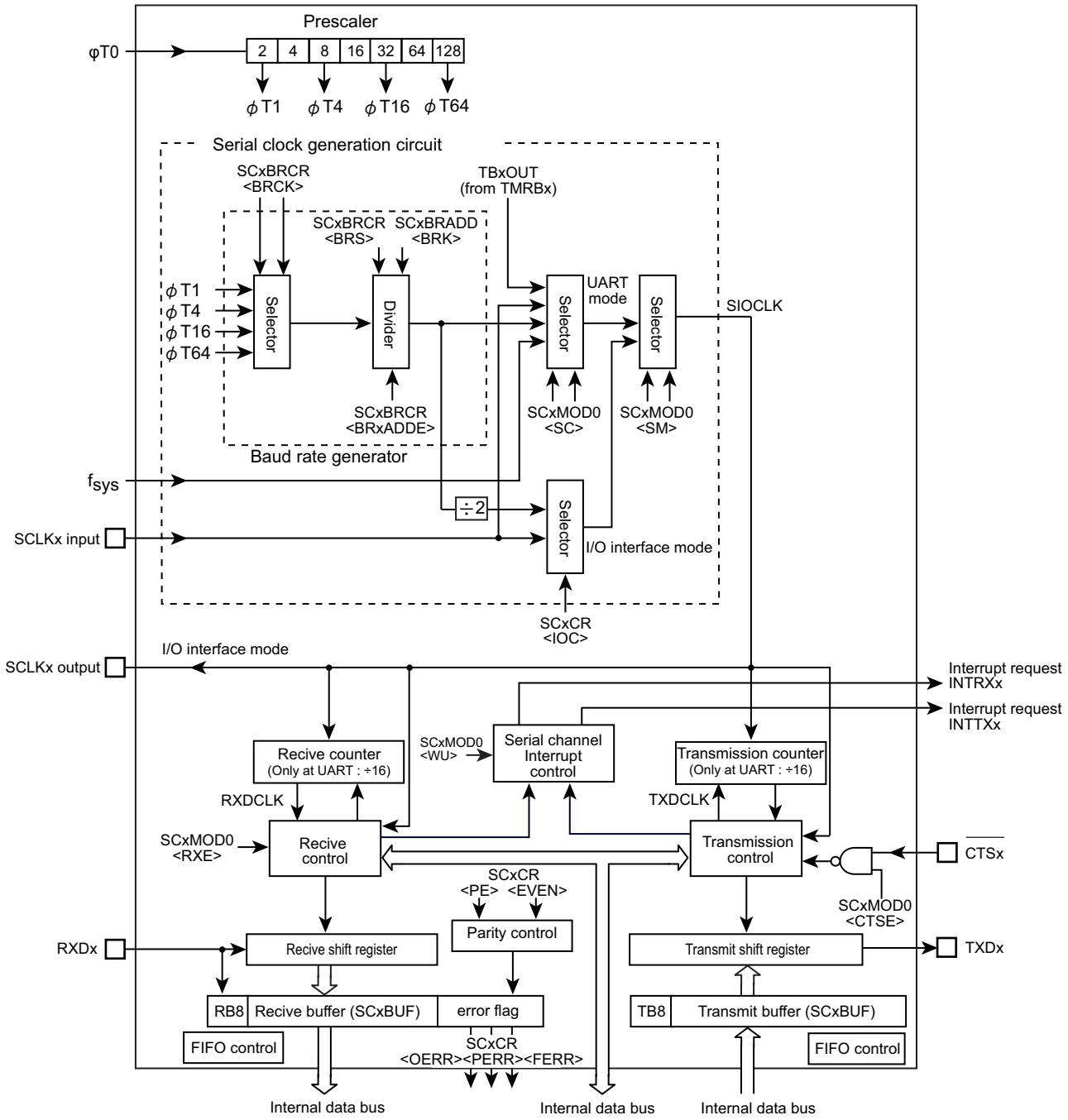


Figure 10-1 SIO Block Diagram

10.4 Registers Description

10.4.1 Registers List in Each Channel

The each channel registers and addresses are shown here.

Channel x	Base Address
Channel0	0x4002_0080
Channel1	0x4002_00C0
Channel2	0x4002_0100

Register name (x=0 to 2)		Address (Base+)
Enable register	SCxEN	0x0000
Buffer register	SCxBUF	0x0004
Control register	SCxCR	0x0008
Mode control register 0	SCxMOD0	0x000C
Baud rate generator control register	SCxBRCR	0x0010
Baud rate generator control register 2	SCxBRADD	0x0014
Mode control register 1	SCxMOD1	0x0018
Mode control register 2	SCxMOD2	0x001C
RX FIFO configuration register	SCxRFC	0x0020
TX FIFO configuration register	SCxTFC	0x0024
RX FIFO status register	SCxRST	0x0028
TX FIFO status register	SCxTST	0x002C
FIFO configuration register	SCxFCNF	0x0030

Note: Do not modify any control register when data is being transmitted or received.

10.4.2 SCxEN (Enable Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	SIOE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	SIOE	R/W	SIO operation 0: Disabled 1: Enabled Specified the SIO operation. To use the SIO, set <SIOE> = "1". 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 for SCxTFC<TIL>.

10.4.3 SCxBUF (Buffer Register)

SCxBUF works as a transmit buffer or FIFO for write operation and as a receive buffer or FIFO for read operation.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TB / RB							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	TB[7:0] / RB [7:0]	R/W	[write] TB : Transmit buffer / FIFO [read] RB : Receive buffer / FIFO

10.4.4 SCxCR (Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	RB8	R	Receive data bit 8 (For UART) 9th bit of the received data in the 9 bits UART mode.
6	EVEN	R/W	Parity (For UART) 0: Odd 1: 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.
5	PE	R/W	Add parity (For UART) 0: Disabled 1: Enabled Controls enabling/ disabling parity. The parity bit may be used only in the 7- or 8-bit UART mode.
4	OERR	R	Overrun error flag (Note) 0: Normal operation 1: Error
3	PERR	R	Parity / Underrun error flag (Note) 0: Normal operation 1: Error
2	FERR	R	Framing error flag (Note) 0: Normal operation 1: Error
1	SCLKS	R/W	Selecting input clock edge (For I/O Interface) 0: Rising edges 1: Falling edges Selects input clock edge for data transmission and reception. Set to "0" in the clock output mode.
0	IOC	R/W	Selecting clock (For I/O Interface) 0: Baud rate generator 1: SCLK pin input

Note: **Any error flag (OERR, PERR, FERR) is cleared to "0" when read.**

10.4.5 SCxMOD0 (Mode Control Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TB8	CTSE	RXE	WU	SM		SC	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	TB8	R/W	Transmit data bit 8 (For UART) Writes the 9th bit of transmit data in the 9 bits UART mode.
6	CTSE	R/W	Handshake function control (For UART) 0: CTS disabled 1: CTS enabled Controls handshake function. Setting "1" enables handshake function using $\overline{CTS}$ pin.
5	RXE	R/W	Receive control (Note) 0: Disabled 1: Enabled
4	WU	R/W	Wake-up function (For UART) 0: Disabled 1: Enabled This function is available only at 9-bit UART mode. In other mode, this function has no meaning. In it is Enabled, Interrupt only when RB9 = "1" at 9-bit UART mode.
3-2	SM[1:0]	R/W	Specifies 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
1-0	SC[1:0]	R/W	Serial transfer clock (For UART) 00: Timer TB9OUT 01: Baud rate generator 10: Internal clock fsys 11: External clock (SCLK input) (As for the I/O interface mode, the serial transfer clock can be set in the control register (SCxCR).

Note 1: **With <RXE> set to "0", set each mode register (SCxMOD0, SCxMOD1 and SCxMOD2). Then set <RXE> to "1".**

Note 2: Do not stop the receive operation (by setting SCxMOD0<RXE> = "0") when data is being received.

10.4.6 SCxMOD1 (Mode Control Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	I2SC	FDPX		TXE	SINT			-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	I2SC	R/W	IDLE 0: Stop 1: Operate Specifies the IDLE mode operation.
6-5	FDPX[1:0]	R/W	Transfer mode setting 00: Transfer prohibited 01: Half duplex (Recieve) 10: Half duplex (Trasmitt) 11: Full duplex 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.
4	TXE	R/W	Transmit control (Note) 0 :Disabled 1: Enabled This bit enables transmission and is valid for all the transfer modes.
3-1	SINT[2:0]	R/W	Interval time of continuous transmission (For I/O interface) 000: None 001: 1SCLK 010: 2SCLK 011: 4SCLK 100: 8SCLK 101: 16SCLK 110: 32SCLK 111: 64SCLK This parameter is valid only for the I/O interface mode when SCLK pin output is selected. In other modes, this function has no meaning. Specifies the interval time of continuous transmission when double buffering or FIFO is enabled in the I/O interface mode.
0	−	R/W	Write a "0".

Note 1: **Specify the all mode first and then enable the <TXE> bit.**

Note 2: **Do not stop the transmit operation (by setting <TXE> = "0")when data is being transmitted.**

10.4.7 SCxMOD2 (Mode Control Register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBEMP	RBFLL	TXRUN	SBLEN	DRCHG	WBUF	SWRST	
After reset	1	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	TBEMP	R	Transmit buffer empty flag. 0: Full 1: Empty If double buffering is disabled, this flag is insignificant. 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".
6	RBFLL	R	Receive buffer full flag. 0: Empty 1: Full 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.
5	TXRUN	R	In transmission flag 0: Stop 1: Operate This is a status flag to show that data transmission is in progress. <TXRUN> and <TBEMP> bits indicate the following status.
4	SBLEN	R/W	STOP bit (for UART) 0 : 1-bit 1 : 2-bit This specifies the length of transmission stop bit in the UART mode. On the receive side, the decision is made using only a single bit regardless of the <SBLEN> setting.
3	DRCHG	R/W	Setting transfer direction 0: LSB first 1: MSB first Specifies the direction of data transfer in the I/O interface mode. In the UART mode, set this bit to LSB first.

<TXRUN>	<TBEMP>	Status
1	−	Transmission in progress
0	1	Transmission completed
	0	Wait state with data in Transmitt buffer

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>2</td><td>WBUF</td><td>R/W</td><td>Double-buffer
0: Disabled
1 : Enabled
This parameter enables or disables the transmit/receive double 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 mode.
When receiving data in the I/O interface mode (SCLK input) and UART mode, double buffering is enabled in both cases that 0 or 1 is set to <WBUF> bit.</td></tr>
<tr><td>1-0</td><td>SWRST[1:0]</td><td>R/W</td><td>Software reset
Overwriting "01" in place of "10" generates a software reset. When this software reset is executed, the following bits are initialized and the transmit/receive circuit, the transmit circuit and the FIFO become initial state (see Note1 and Note2).
<table>
<tr><th>Register</th><th>Bit</th></tr>
<tr><td>SCxMOD0</td><td>RXE</td></tr>
<tr><td>SCxMOD1</td><td>TXE</td></tr>
<tr><td>SCxMOD2</td><td>TBEMP, RBFLL, TXRUN</td></tr>
<tr><td>SCxCR</td><td>OERR, PERR, FERR</td></tr>
</table></td></tr>
</table>

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.**

10.4.8 SCxBRCR (Baud Rate Generator Control Register), SCxBRADD (Baud Rate Generator Control Register 2)

The division ratio of the baud rate generator can be specified in the registers shown below.

SCxBRCR

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	BRADDE	BRCK		BRS			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	−	R/W	Write "0".
6	BRADDE	R/W	N + (16 − K)/16 divider function (For UART) 0: disabled 1: enabled This division function can only be used in the UART mode.
5-4	BRCK[1:0]	R/W	Select input clock to the baud rate generator. 00: φT1 01: φT4 10: φT16 11: φT64
3-0	BRS[3:0]	R/W	Division ratio "N" 0000: 16 0001: 1 0010: 2 ... 1111: 15

SCxBRADD

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	BRK			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as 0.
3-0	BRK[3:0]	R/W	Specify K for the "N + (16 − K)/16" division (For UART) 0000: Prohibited 0001: K = 1 0010: K = 2 ... 1111: K = 15

Table 10-2 lists the settings of baud rate generator division ratio.

Table 10-2 Setting division ratio

	<BRADDE> = "0"	<BRADDE> = "1" (Note1) (Only UART mode)
<BRS>	Specify "N" (Note2) (Note3)	
<BRK>	No setting required	Specify "K" (Note4)
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 <BRADDE> to "1" after setting the K value to <BRK>. The "N + (16 - K)/16" division function can only be used in the UART mode.**

Note 2: **As a division ratio, 1 ("0001") or 16 ("0000") can not be applied to N when using the "N + (16 - K)/16" division function in the UART mode.**

Note 3: **The division ratio "1" of the baud rate generator can be specified only when the double buffering is used in the I/O interface mode.**

Note 4: **Specifying "K = 0" is prohibited.**

10.4.9 SCxFCNF (FIFO Configuration Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	RFST	TFIE	RFIE	RXTXCNT	CNFG
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0
7-5	-	R/W	Be sure to write "000"
4	RFST	R/W	Bytes used in RX FIFO 0:Maximum 1:Same as FILL level of RX FIFO When RX FIFO is enabled, the number of RX FIFO bytes to be used is selected (Note1) 0: The maximun number of bytes of the FIFO configured (see also <CNFG>). 1: Same as the fill level for receive interrupt generation specified by SCxRFC <RIL[1:0]>
3	TFIE	R/W	TX interrupt for TX FIFO 0: Disabled 1:Enabled When TX FIFO is enabled, transmit interrupts are enabled or disabled by this parameter.
2	RFIE	R/W	RX interrupt for RX FIFO 0: Disabled 1:Enabled When RX FIFO is enabled, receive interrupts are enabled or disabled by this parameter.
1	RXTXCNT	R/W	Automatic disable of RXE/TXE 0: None 1: Auto diabled Controls automatic disabling of transmission and reception. Setting "1" enables to operate as follows Half duplex RX: When receive shift register, the receive buffer and the RX FIFO are filled, SCxMOD0<RXE> is automatically set to "0" to inhibit further reception. Half duplex TX: When the TX FIFO, the transmit buffer and the transmit shift register is empty, SCxMOD1<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.
0	CNFG	R/W	Enables FIFO. 0: Disabled 1: Enabled If enabled, the SCxMOD1 <FDPX[1:0]> setting automatically configures FIFO as follows: (The type of TX/RX can be specified in the mode control register 1 SCxMOD1<FDPX[1:0]>). Half duplex RX: RX FIFO 4byte Half duplex TX: TX FIFO 4byte Full duplex: RX FIFO 2byte + TX FIFO 2byte

Note 1: **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.**

Note 2: **The FIFO can not use in 9bit UART mode.**

10.4.10 SCxRFC (RX FIFO Configuration Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RFCS	RFIS	-	-	-	-	RIL	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	RFCS	W	RX FIFO clear 1: Clear Setting "1" clears RX FIFO and "0" is always read.
6	RFIS	R/W	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 exceeds the specified fill level at the time data is read.
5-2	−	R	Read as 0.
1-0	RIL[1:0]	R/W	FIFO fill level to generate RX interrupts

	Half duplex	Full duplex
00	4byte	2byte
01	1byte	1byte
10	2byte	2byte
11	3byte	1byte

Note: To use TX/RX FIFO buffer, TX/RX FIFO must be cleared after setting the SIO transfer mode (half duplex/ full duplex) and enabling FIFO (SCxFCNF<CNFG> = "1").

10.4.11 SCxTFC (TX FIFO Configuration Register) (Note2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TFCS	TFIS	-	-	-	-	TIL	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	TFCS	W	TX FIFO clear (Note 1) 1: Clears TX FIFO. Setting "1" clears TX FIFO and "0" is always read.
6	TFIS	R/W	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 can not reach the specified fill level at the time new data is read.
5-2	−	R	Read as 0.
1-0	TIL[1:0]	R/W	Selects FIFO fill level.

	Other than full duplex	Full duplex
00	Empty	Empty
01	1 byte	1 byte
10	2 byte	Empty
11	3 byte	1 byte

Note 1: **To use TX/RX FIFO buffer, TX/RX FIFO must be cleared after setting the SIO transfer mode (half duplex/full duplex) and enabling FIFO (SCxFCNF<CNFG> = "1").**

Note 2: **After you perform the following operations, configure the SCxTFC register again.**

SCxEN<SIOE> = "0" (SIO operation stop)

Conditions are as follows:SCxMOD1<I2SC> = "0" (operation is prohibited in IDLE mode) and releasing the low power consumption mode which started by the WFI (Wait For Interrupt) instruction.

10.4.12 SCxRST (RX FIFO Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ROR	-	-	-	-	RLVL		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	ROR	R	RX FIFO Overrun (Note) 0: Not generated 1: Generated
6-3	−	R	Read as 0.
2-0	RLVL[2:0]	R	Status of RX FIFO fill level. 000: Empty 001: 1 byte 010: 2 byte 011: 3 byte 100: 4 byte

Note: **The <ROR> bit is cleared to "0" when receive data is read from the SCxBUF register.**

10.4.13 SCxTST (TX FIFO Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TUR	-	-	-	-	TLVL		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	TUR	R	TX FIFO under run (Note) 0: Not generated 1: Generated.
6-3	−	R	Read as 0.
2-0	TLVL[2:0]	R	Status of TX FIFO fill level. 000: Empty 001: 1 byte 010: 2 byte 011: 3 byte 100: 4 byte

Note: **The <TUR> bit is cleared to "0" when transmit data is written to the SCxBUF register.**

10.5 Operation in Each Mode

Table 10-3 shows the modes and data formats.

Table 10-3 Mode and Data format

Mode	Mode type	Data length	Transfer direction	Specifies whether to use parity bits.	STOP bit length (transmit)
Mode 0	Synchronous communication mode (IO interface mode)	8 bit	LSB first/MSB first	-	-
Mode 1	Asynchronous communication mode (UART mode)	7 bit	LSB first	o	1 bit or 2 bit
Mode 2		8 bit		o	
Mode 3		9 bit		×	

Mode 0 is a synchronous communication and can be used to extend I/O. This mode transmits and receives data in synchronization with SCLK. SCLK can be used for both input and output.

The direction of data transfer can be selected from LSB first and MSB first. This mode is not allowed either to use parity bits or STOP bits.

The mode 1, mode 2 and mode 3 are asynchronous modes and the transfer direction is fixed to the LSB first.

Parity bits can be added in the mode 1 and mode 2. The mode 3 has a wakeup function in which the master controller can start up slave controllers via the serial link (multi-controller system).

STOP bit in transmission can be selected from 1 bit and 2 bits. The STOP bit length in reception is fixed to a one bit.

10.6 Data Format

10.6.1 Data Format List

Figure 10-2 shows data format.

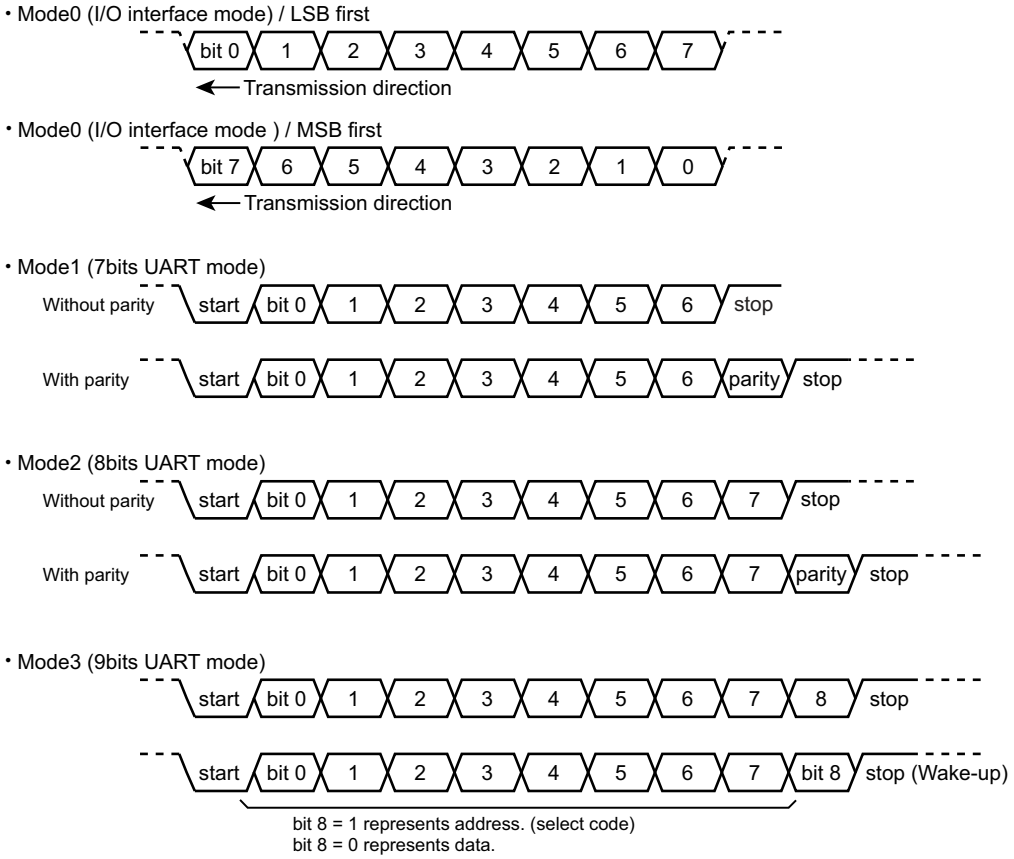


Figure 10-2 Data Format

10.6.2 Parity Control

The parity bit can be added only in the 7- or 8-bit UART mode.

Setting "1" to SCxCR<PE> enables the parity.

The <EVEN> bit of SCxCR selects either even or odd parity.

10.6.2.1 Transmission

Upon data transmission, the parity control circuit automatically generates the parity with the data in the transmit buffer.

After data transmission is complete, the parity bit will be stored in SCxBUF<TB7> in the 7-bit UART mode and SCxMOD<TB8> in the 8-bit UART mode.

The <PE> and <EVEN> settings must be completed before data is written to the transmit buffer.

10.6.2.2 Receiving Data

If the received data is moved from the receive shift register to the receive buffer, a parity is generated.

In the 7-bit UART mode, the generated parity is compared with the parity stored in SCxBUF<RB7>, while in the 8-bit UART mode, it is compared with the one in SCxCR<RB8>.

If there is any difference, a parity error occurs and the <PERR> of the SCxCR register is set to "1".

In use of the FIFO, <RERR> indicates that a parity error was generated in one of the recieved data.

10.6.3 STOP Bit Length

The length of the STOP bit in the UART transmission mode can be selected from one bit or two bits by setting the SCxMOD2<SBLEN>. The length of the STOP bit data is determined as one-bit when it is received regardless of the setting of this bit.

10.7 Clock Control

10.7.1 Prescaler

There is a 7-bit prescaler to divide a prescaler input clock ΦT0 by 2, 8, 32 and 128.

Use the CGSYSCR register in the clock/mode control block to select the input clock ΦT0 of the prescaler.

The prescaler becomes active only when the baud rate generator is selected as a transfer clock by SCxMOD0<SC[1:0]> = "11".

Table 10-4 (operation frequency 40MHz), Table 10-5 (operation frequency 32MHz) show the resolution of the input clock to the baud rate generator.

Table 10-4 Clock Resolution to the Baud Rate Generator fc = 40 MHz

peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
0 (fgear)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.05 μs)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		001 (fperiph/2)	$fc/2^2$ (0.1 μs)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		010 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)
	100 (fc/2)	000 (fperiph/1)	$fc/2^2$ (0.1 μs)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		001 (fperiph/2)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		010 (fperiph/4)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		011 (fperiph/8)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		100 (fperiph/16)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)
		101 (fperiph/32)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{13}$ (204.8 μs)
	101 (fc/4)	000 (fperiph/1)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		001 (fperiph/2)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		010 (fperiph/4)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		011 (fperiph/8)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)
		100 (fperiph/16)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{13}$ (204.8 μs)
		101 (fperiph/32)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)	$fc/2^{14}$ (409.6 μs)
	110 (fc/8)	000 (fperiph/1)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		001 (fperiph/2)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		010 (fperiph/4)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)
		011 (fperiph/8)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{13}$ (204.8 μs)
		100 (fperiph/16)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)	$fc/2^{14}$ (409.6 μs)
		101 (fperiph/32)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{13}$ (204.8 μs)	$fc/2^{15}$ (819.2 μs)

Table 10-4 Clock Resolution to the Baud Rate Generator fc = 40 MHz

peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
1 (fc)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.05 μs)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		001 (fperiph/2)	$fc/2^2$ (0.1 μs)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		010 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)
	100 (fc/2)	000 (fperiph/1)	–	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		001 (fperiph/2)	$fc/2^2$ (0.1 μs)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		010 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)
	101 (fc/4)	000 (fperiph/1)	–	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		001 (fperiph/2)	–	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		010 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)
	110 (fc/8)	000 (fperiph/1)	–	–	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		001 (fperiph/2)	–	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		010 (fperiph/4)	–	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)

Note 1: **The prescaler output clock φTn must be selected so that the relationship "φTn ≤ fsys / 2" is satisfied (so that φTn is slower than fsys).**

Note 2: **Do not change the clock gear while SIO is operating.**

Note 3: **The dashes in the above table indicate that the setting is prohibited.**

Table 10-5 Clock Resolution to the Baud Rate Generator fc = 32 MHz

peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
0 (fgear)	000 (fc)	000 (fperiph/1)	fc/2^1 (0.0625 μs)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)
		001 (fperiph/2)	fc/2^2 (0.125 μs)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)
		010 (fperiph/4)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		011 (fperiph/8)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)
		100 (fperiph/16)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)	fc/2^{11} (64.0 μs)
		101 (fperiph/32)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)	fc/2^{12} (128.0 μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^1 (0.0625 μs)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)
		001 (fperiph/2)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		010 (fperiph/4)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)
		011 (fperiph/8)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)	fc/2^{11} (64.0 μs)
		100 (fperiph/16)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)	fc/2^{12} (128.0 μs)
		101 (fperiph/32)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)	fc/2^{11} (64.0 μs)	fc/2^{13} (256.0 μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^1 (0.0625 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		001 (fperiph/2)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)
		010 (fperiph/4)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)	fc/2^{11} (64.0 μs)
		011 (fperiph/8)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)	fc/2^{12} (128.0 μs)
		100 (fperiph/16)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)	fc/2^{11} (64.0 μs)	fc/2^{13} (256.0 μs)
		101 (fperiph/32)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)	fc/2^{12} (128.0 μs)	fc/2^{14} (512.0 μs)
	110 (fc/8)	000 (fperiph/1)	fc/2^1 (0.0625 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)
		001 (fperiph/2)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)	fc/2^{11} (64.0 μs)
		010 (fperiph/4)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)	fc/2^{12} (128.0 μs)
		011 (fperiph/8)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)	fc/2^{11} (64.0 μs)	fc/2^{13} (256.0 μs)
		100 (fperiph/16)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)	fc/2^{12} (128.0 μs)	fc/2^{14} (512.0 μs)
		101 (fperiph/32)	fc/2^9 (16.0 μs)	fc/2^{11} (64.0 μs)	fc/2^{13} (256.0 μs)	fc/2^{15} (1024 μs)

Table 10-5 Clock Resolution to the Baud Rate Generator fc = 32 MHz

peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
1 (fc)	000 (fc)	000 (fperiph/1)	fc/2^1 (0.0625 μs)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)
		001 (fperiph/2)	fc/2^2 (0.125 μs)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)
		010 (fperiph/4)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		011 (fperiph/8)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)
		100 (fperiph/16)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)	fc/2^{11} (64.0 μs)
		101 (fperiph/32)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)	fc/2^{12} (128.0 μs)
	100 (fc/2)	000 (fperiph/1)	−	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)
		001 (fperiph/2)	fc/2^2 (0.125 μs)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)
		010 (fperiph/4)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		011 (fperiph/8)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)
		100 (fperiph/16)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)	fc/2^{11} (64.0 μs)
		101 (fperiph/32)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)	fc/2^{12} (128.0 μs)
	101 (fc/4)	000 (fperiph/1)	−	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)
		001 (fperiph/2)	−	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)
		010 (fperiph/4)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		011 (fperiph/8)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)
		100 (fperiph/16)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)	fc/2^{11} (64.0 μs)
		101 (fperiph/32)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)	fc/2^{12} (128.0 μs)
	110 (fc/8)	000 (fperiph/1)	−	−	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)
		001 (fperiph/2)	−	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)
		010 (fperiph/4)	−	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		011 (fperiph/8)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)
		100 (fperiph/16)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)	fc/2^{11} (64.0 μs)
		101 (fperiph/32)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)	fc/2^{12} (128.0 μs)

Note 1: **The prescaler output clock φTn must be selected so that the relationship "φTn ≤ fsys / 2" is satisfied (so that φTn is slower than fsys / 2).**

Note 2: **Do not change the clock gear while SIO is operating.**

Note 3: **The dashes in the above table indicate that the setting is prohibited.**

10.7.2 Serial Clock Generation Circuit

The serial clock circuit is a block to generate transmit and receive clocks (SIOCLK) and consists of the circuits in which clocks can be selected by the settings of the baud rates generator and modes.

10.7.2.1 Baud Rate Generator

The baud rate generator generates transmit and receive clocks to determine the serial channel transfer rate.

(1) Baud Rate Generator input clock

The input clock of the baud rate generator is selected from the prescaler outputs divided by 2, 8, 32 and 128.

This input clock is selected by setting the SCxBRCR<BRCK>.

(2) Baud Rate Generator output clock

The frequency division ratio of the output clock in the baud rate generator is set by SCxBRCR and SCxBRADD.

The following frequency divide ratios can be used; 1/N frequency division in the I/O interface mode ,either 1/N or N + (16-K)/16 in the UART mode.

The table below shows the frequency division ratio which can be selected.

Mode	Divide Function Setting SCxBRCR<BRADDE>	Divide by N SCxBRCR<BRS>	Divide by K SCxBRADD<BRK>
I/O interface	Divide by N	1 to 16 (Note)	-
UART	Divide by N	1 to 16	-
	N + (16-K)/16 division	2 to 15	1 to 15

Note: 1/N (N=1)frequency division ratio can be used only when a double buffer is enabled.

10.7.2.2 Clock Selection Circuit

A clock can be selected by setting the modes and the register.

Modes can be specified by setting the SCxMOD0<SM>.

The input clock in I/O interface mode is selected by setting SCxCR. The clock in UART mode is selected by setting SCxMOD0<SC>.

(1) Transfer Clock in I/O interface mode

Table 10-6 shows clock selection in I/O interface mode.

Table 10-6 Clock Selection in I/O Interface Mode

Mode SCxMOD0<SM>	Input/Output selection SCxCR<IOC>	Clock edge selection SCxCR<SCLKS>	Clock of use
I/O interface mode	SCLK output	Set to "0". (Fixed to the rising edge)	Divided by 2 of the baud rate generator output.
	SCLK input	Rising edge	SCLK input rising edge
		Falling edge	SCLK input falling edge

To get the highest baud rate, the baud rate generator must be set as below.

Note: **When deciding clock settings, make sure that AC electrical character is satisfied.**

- Clock/mode control block settings
  - fc = 40MHz
  - fgear = 40MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
  - ΦT0 = 40MHz (CGSYSCR<PRCK[2:0]> = "000" : 1 division ratio)
- SIO settings (if double buffer is used)
  - Clock (SCxBRCR<BRCK[1:0]> = "00" : ΦT1 selected) = 20MHz
  - Divided clock frequency (SCxBRCR<BRS[3:0]> = "0001" : 1 division ratio) = 20MHz

  1 division ratio can be selected if double buffer is used. In this case, baud rate is 10Mbps because 20MHz is divided by 2.
- SIO settings (if double buffer is not used)
  - Clock (SCxBRCR<BRCK[1:0]> = "00" : ΦT1 selected) = 20MHz
  - Divided clock frequency (SCxBRCR<BRS[3:0]> = "0010" : 2 division ratio) = 10MHz

  2 division ratio is the highest if double buffer is not used. In this case, baud rate is 5Mbps because 10MHz is divided by 2.

To use SCLK input, the following conditions must be satisfied.

- If double buffer is used

  - SCLK cycle > 6/fsys

  The highest baud rate is less than 40 ÷ 6 = 6.66 Mbps.
- If double buffer is not used

  - SCLK cycle > 8/fsys

The highest baud rate is less than 40 ÷ 8 = 5.0 Mbps.

(2) Transfer clock in the UART mode

Table 10-7 shows the clock selection in the UART mode. In the UART mode, selected clock is divided by 16 in the receive counter or the transmit counter before use.

Table 10-7 Clock Selection in UART Mode

Mode SCxMOD0<SM>	Clock selection SCxMOD0<SC>
UART Mode	Timer output
	Baud rate generator
	fsys
	SCLK input

The examples of baud rate in each clock settings.

- If the baud rate generator is used
  - fc = 40MHz
  - fgear = 40MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
  - ΦT0 = 40MHz (CGSYSCR<PRCK[2:0]> = "000" : 1 division ratio)
  - Clock = ΦT1 = 20MHz (SCxBRCR<BRCK[1:0]> = "00" : ΦT1 selected)

  The highest baud rate is 1.25Mbps because 20MHz is divided by 16.

  Table 10-8 shows examples of baud rate when the baud rate generator is used with the following clock settings.

  - fc = 9.8304MHz
  - fgear = 9.8304MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
  - ΦT0 = 4.9152MHz (CGSYSCR<PRCK[2:0]> = "001" : 2 division ratio)

Table 10-8 Example of UART Mode Baud Rate (Using the Baud Rate Generator)

fc [MHz]	Division ratio N (SCxBRCR<BRS>)	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
	16	9.600	2.400	0.600	0.150

Unit : kbps

- If the SCLK input is used

  To use SCLK input, the following conditions must be satisfied.

  - SCLK cycle > 2/fsys

  The highest baud rate must be less than 40 ÷ 2 ÷ 16 = 1.25 Mbps.

- If fsys is used

  Since the highest value of fsys is 40MHz, the highest baud rate is 40 ÷ 16 = 2.5Mbps.

- If timer output is used

  To enable the timer output, the following condition must be set: a timer flip-flop output inverts when the value of the counter and that of TBxRG0 match. The SIOCLK clock frequency is "Setting value of TBxRG0 × 2".

  Baud rates can be obtained by using the following formula.

Baud rate calculation

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by CGSYSCR<PRCK[1:0]>}}{(\text{TBxRG0} \times 2) \times 2 \times 16}$$

In the case the timer prescaler clock ΦT1 (2divition ratio) is selected.

One clock cycle is a period that the timer flip-flop is inverted twice.

Table 10-9 shows the examples of baud rates when the timer output is used with the following clock settings.

- fc = 32MHz / 9.8304MHz / 8MHz
- fgear = 32MHz / 9.8304MHz / 8MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
- ΦT0 = 16MHz / 4.9152MHz / 4MHz (CGSYSCR<PRCK[2:0]> = "001" : 2 division ratio)
- Timer count clock = 4MHz / 1.2287MHz / 1MHz (TBxMOD<TBCLK[1:0]> = "01" : ΦT1 selected)

Table 10-9 Example of UART Mode Baud Rate (Using the Timer Output)

TBxRG0 setting	fc		
	32MHz	9.8304MHz	8MHz
0x0001	250	76.8	62.5
0x0002	125	38.4	31.25
0x0003	-	25.6	-
0x0004	62.5	19.2	15.625
0x0005	50	15.36	12.5
0x0006	-	12.8	-
0x0008	31.25	9.6	-
0x000A	25	7.68	6.25
0x0010	15.625	4.8	-
0x0014	12.5	3.84	3.125

Unit : kbps

10.8 Transmit/Receive Buffer and FIFO

10.8.1 Configuration

Figure 10-3 shows the configuration of transmit buffer, receive buffer and FIFO.

Appropriate settings are required for using buffer and FIFO. The configuration may be predefined depending on the mode.

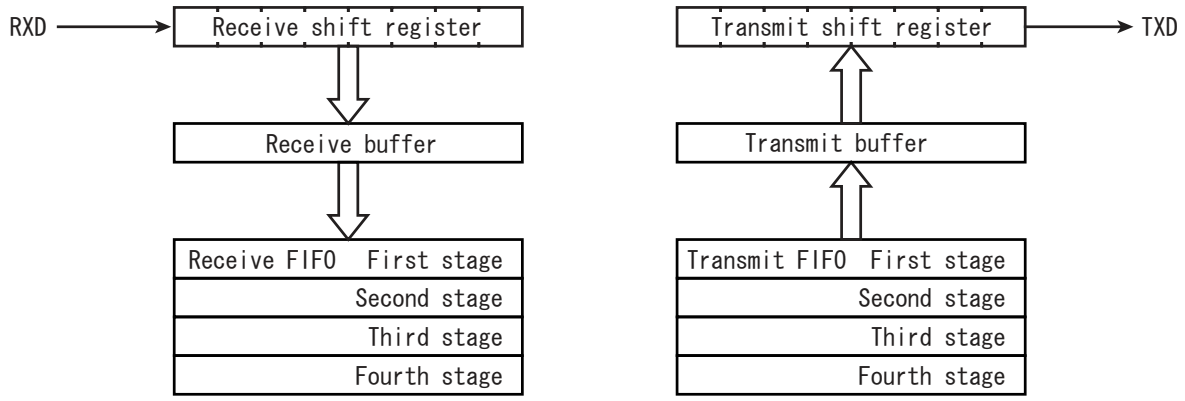


Figure 10-3 The Configuration of Buffer and FIFO

10.8.2 Transmit/Receive Buffer

Transmit buffer and receive buffer are double-buffered. The buffer configuration is specified by SCxMOD2<WBUF>.

In the case of using a receive buffer, if SCLK input is set to generate clock output in the I/O interface mode or the UART mode is selected, it's double buffered despite the <WBUF> settings. In other modes, it's according to the <WBUF> settings.

Table 10-10 shows correlation between modes and buffers.

Table 10-10 Mode and buffer Composition

Mode		SCxMOD2<WBUF>	
		"0"	"1"
UART	Transmit	Single	Double
	Receive	Double	Double
I/O interface (SCLK input)	Transmit	Single	Double
	Receive	Double	Double
I/O interface (SCLK output)	Transmit	Single	Double
	Receive	Single	Double

10.8.3 FIFO

In addition to the double buffer function above described, 4-byte FIFO can be used.

To enable FIFO, enable the double buffer by setting SCxMOD2<WBUF> to "1" and SCxFCNF<CNFG> to "1". The FIFO buffer configuration is specified by SCxMOD1<FDPX[1:0]>.

Note: To use TX/RX FIFO buffer, TX/RX FIFO must be cleared after setting the SIO transfer mode (half duplex/ full duplex) and enabling FIFO (SCxFCNF<CNFG> = "1").

Table 10-11 shows correlation between modes and FIFO.

Table 10-11 Mode and FIFO Composition

	SCxMOD1<FDPX[1:0]>	RX FIFO	TX FIFO
Half duplex RX	"01"	4byte	-
Half duplex TX	"10"	-	4byte
Full duplex	"11"	2byte	2byte

10.9 Status Flag

The SCxMOD2 register has two types of flag. This bit is significant only when the double buffer is enabled.

<RBFLL> is a flag to show that the receive buffer is full. When one frame of data is received and the data is moved from the receive shift register to the receive buffers, this bit changes to "1" while reading this bit changes it to "0".

<TBEMP> shows that the transmit buffers are empty. When data in the transmit buffers is moved to the transmit shift register, this bit is set to "1" When data is set to the transmit buffers, the bit is cleared to "0".

10.10 Error Flag

Three error flags are provided in the SCxCR register. The meaning of the flags is changed depending on the modes. The table below shows the meanings in each mode.

These flags are cleared to "0" after reading the SCxCR register.

Mode	Flag		
	<OERR>	<PERR>	<FERR>
UART	Overrun error	Parity error	Framing error
I/O Interface (SCLK input)	Overrun error	Underrun error (When using double buffer or FIFO)	Fixed to 0
		Fixed to 0 (When a double buffer and FIFO unused)	
I/O Interface (SCLK output)	Undefined	Undefined	Fixed to 0

10.10.1 OERR Flag

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 of 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).

In the I/O interface with SCLK output mode, the SCLK output stops upon setting the flag.

Note: To switch the I/O interface SCLK output mode to other modes, read the SCxCR register and clear the overrun flag.

10.10.2 PERR Flag

This flag indicates a parity error in the UART mode and an under-run error in the I/O interface mode.

In the UART mode, <PERR> is set to "1" when the parity generated from the received data is different from the parity received.

In the I/O interface mode, <PERR> is set to "1" under the following conditions when a double buffer is enabled.

In the SCLK input mode, <PERR> is set to "1" when the SCLK is input after completing data output of the transmit shift register with no data in the tarnsmit buffer.

In the SCLK output mode, <PERR> is set to "1" after completing output of all data and the SCLK output stops.

Note: To switch the I/O interface SCLK output mode to other modes, read the SCxCR register and clear the under-run flag.

10.10.3 FERR Flag

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 stop bit length settings in the SCxMOD2<SBLEN>register, the stop bit status is determined by only 1.

This bit is fixed to "0" in the I/O interface mode.

10.11 Receive

10.11.1 Receive Counter

The receive counter is a 4-bit binary counter and is up-counted by SIOCLK. In the UART mode, sixteen SIOCLK clock pulses are used in receiving a single data bit and 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.

10.11.2 Receive Control Unit

10.11.2.1 I/O interface mode

In the SCLK output mode with SCxCR <IOC> set to "0", the RXD pin is sampled on the rising edge of the shift clock outputted to the SCLK pin.

In the SCLK input mode with SCxCR <IOC> set to "1", the serial receive data RXD pin is sampled on the rising or falling edge of SCLK input signal depending on the SCxCR <SCLKS> setting.

10.11.2.2 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.

10.11.3 Receive Operation

10.11.3.1 Receive Buffer

The received data is stored by 1 bit in the receive shift register. When a complete set of bits has been stored, the interrupt INTRXx is generated.

When the double buffer is enabled, the data is moved to the receive buffer (SCxBUF) and the receive buffer full flag (SCxMOD2<RBFLL>) is set to "1". The receive buffer full flag is "0" cleared by reading the receive buffer.

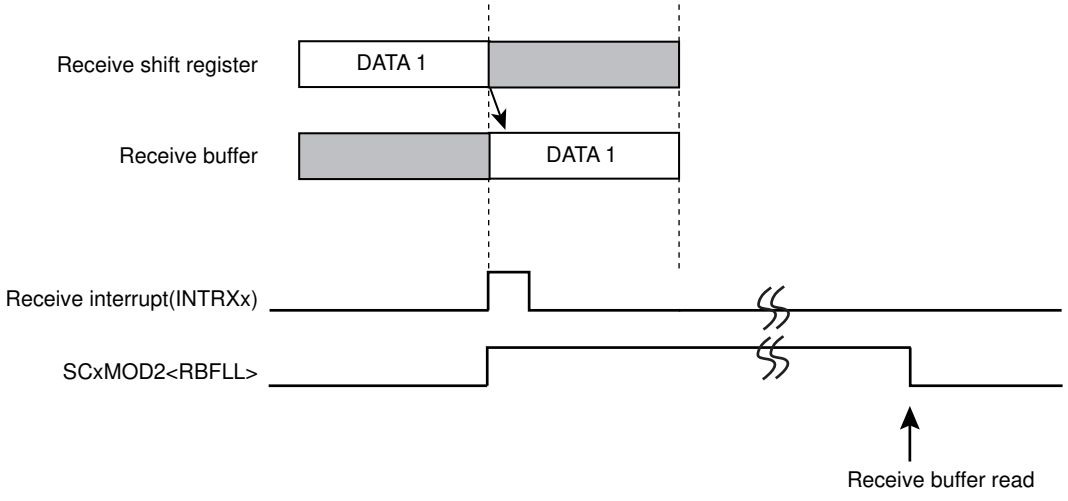


Figure 10-4 Receive Buffer Operation

10.11.3.2 Receive FIFO Operation

When FIFO is enabled, the received data is moved from receive buffer to receive FIFO and the receive buffer full flag is cleared immediately. An interrupt will be generated according to the SCxRFC<RIL> setting.

Note: When the data with parity bit are received in UART mode by using the FIFO, the parity error flag is shown the occurring the parity error in the received data.

The following describes configurations and operations in the half duplex RX mode.

SCxMOD1[6:5] =01	: Transfer mode is set to half duplex mode
SCxFCNF[4:0] = 10111	: Automatically inhibits continuous 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.
SCxRFC[1:0] = 00	: The fill level of FIFO in which generated receive interrupt is set to 4-byte.
SCxRFC[7:6] = 11	: Clears receive FIFO and sets the condition of interrupt generation.

After setting of the above FIFO configuration, the data reception is started by writing "1" to the SCxMOD0 <RXE>. When the data is stored all in the receive shift register, receive buffer and receive FIFO, SCxMOD0<RXE> is automatically cleared and the receive operation is finished.

In this above condition, if the cotinuous reception after reaching the fill level is enabed, and it is possible to receive a data continuouslywith and reading the data in the FIFO.

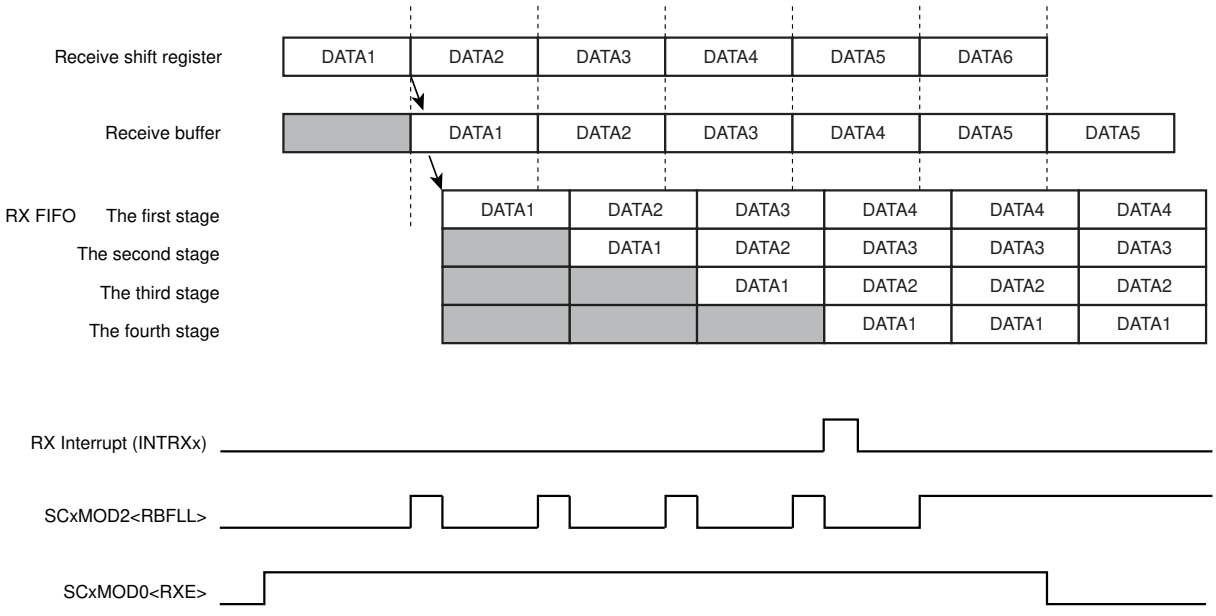


Figure 10-5 Receive FIFO Operation

10.11.3.3 I/O interface mode with SCLK output

In the I/O interface mode and SCLK output setting, SCLK output stops when all received data is stored in the receive buffer and FIFO. So, in this mode, the overrun error flag has no meaning.

The timing of SCLK output stop and re-output depends on receive buffer and FIFO.

(1) Case of single buffer

Stop SCLK output after receiving a data. In this mode, I/O interface can transfer each data with the transfer device by hand-shake.

When the data in a buffer is read, SCLK output is restarted.

(2) Case of double buffer

Stop SCLK output after receiving the data into a receive shift register and a receive buffer.

When the data is read, SCLK output is restarted.

(3) Case of FIFO

Stop SCLK output after receiving the data into a shift register, received buffer and FIFO.

When one byte data is read, the data in the received buffer is transferred into FIFO and the data in the receive shift register is transferred into received buffer and SCLK output is restarted.

And if SCxFCNF<RXTXCNT>is set to "1", SCLK stops and receive operation stops with clearing SCxMOD0<RXE> bit too.

10.11.3.4 Read Received Data

In spite of enabling or disabling FIFO, read the received data from the receive buffer (SCxBUF).

When receive FIFO is disabled, the buffer full flag SCxMOD2<RBFLL> is cleared to "0" by this reading. In the case of the next data can be received in the receive shift register before reading a data from the receive buffer. 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 SCxCR<RB8>.

When the receive FIFO is available, the 9-bit UART mode is prohibited because up to 8-bit data can be stored in FIFO. In the 8-bit UART mode, the parity bit is lost but parity error is determined and the result is stored in SCxCR<PERR>.

10.11.3.5 Wake-up Function

In the 9-bit UART mode, the slave controller can be operated in the wake-up mode by setting the wake-up function SCxMOD0 <WU> to "1." In this case, the interrupt INTRXx will be generated only when SCxCR <RB8> is set to "1."

10.11.3.6 Overrun Error

When FIFO is disabled, the overrun errorr is occurred and set overrun flag without completing data read before receiveing the next data. When overrun error is ocurred, a content of receive buffer and SCxCR<RB8> is not lost, but a content of receive shift register is lost.

When FIFO is enabled, overrun error is ocurred and set overrun flag by no reading the data before moving the next data into received buffer when FIFO is full . In this case, the contens of FIFO are not lost.

In the I/O interface mode with SCLK output setting, the clock output automatically stops, so this flag has no meaning.

Note: When the mode is changed from I/O interface SCLK outout mode to the other mode, read SCxCR and clear overrun flag.

10.12 Transmission

10.12.1 Transmission Counter

The transmit counter is a 4-bit binary counter and is counted by SIOCLK as in the case of the receive counter. In UART mode, it generates a transmit clock (TXDCLK) on every 16th clock pulse.

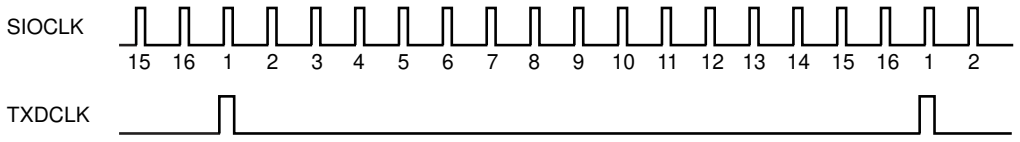


Figure 10-6 Generation of Transmission Clock

10.12.2 Transmission Control

10.12.2.1 I/O Interface Mode

In the SCLK output mode with SCxCR<IOC> set to "0", each bit of data in the transmit buffer is outputted to the TXD pin on the falling edge of the shift clock outputted from the SCLK pin.

In the SCLK input mode with SCxCR<IOC> set to "1", each bit of data in the transmit buffer is outputted to the TXD pin on the rising or falling edge of the SCLK input signal according to the SCxCR<SCLKS> setting.

10.12.2.2 UART Mode

When the transmit data is written in the transmit buffer, data transmission is initiated on the rising edge of the next TXDCLK and the transmit shift clock signal is also generated.

10.12.3 Transmit Operation

10.12.3.1 Operation of Transmission Buffer

If double buffering is disabled, the CPU writes data only to Transmit shift Buffer and the transmit interrupt INTTXx is generated upon completion of data transmission.

If double buffering is enabled (including the case the transmit FIFO is enabled), data written to the transmit buffer is moved to the transmit shift register. The INTTXx interrupt is generated at the same time and the transmit buffer empty flag (SCxMOD2<TBEMP>) is set to "1". This flag indicates that the next transmit data can be written. When the next data is written to the transmit buffer, the <TBEMP> flag is cleared to "0".

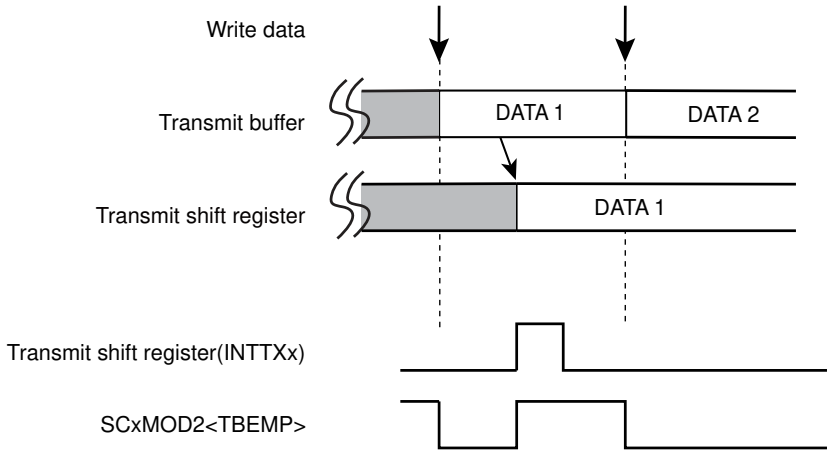


Figure 10-7 Operation of Transmission Buffer (Double-buffer is enabled)

10.12.3.2 Transmit FIFO Operation

When FIFO is enabled, the maximum 5-byte data can be stored using the transmit buffer and FIFO. Once transmission is enabled, data is transferred to the transmit shift register from the transmit buffer and start transmission. If data exists in the FIFO, the data is moved to the transmit buffer immediately, and the <TBEMP> flag is cleared to "0".

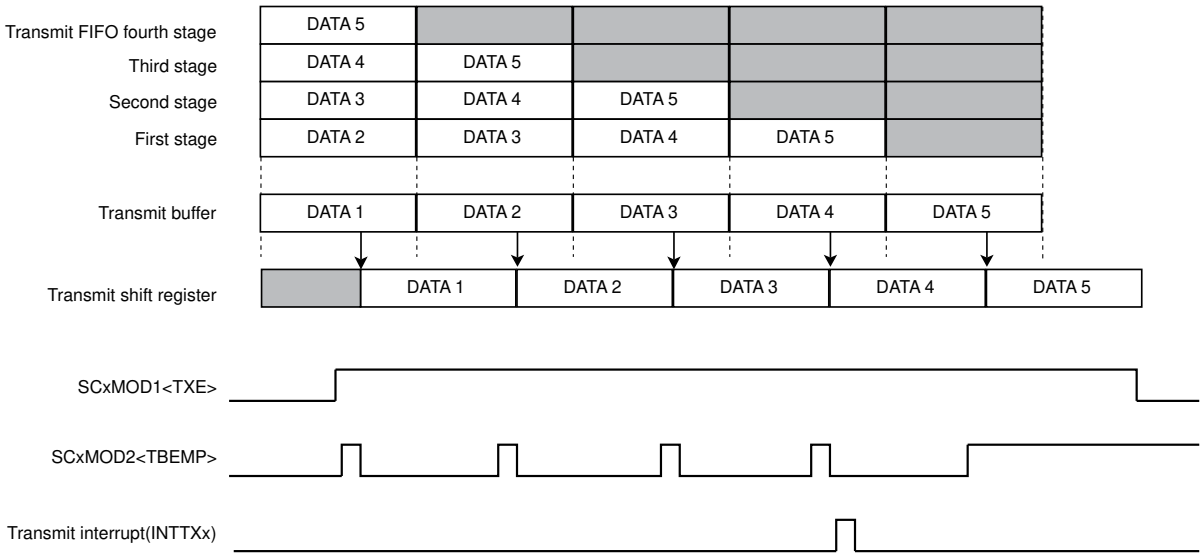
Note: **To use TX FIFO buffer, TX FIFO must be cleared after setting the SIO transfer mode (half duplex/ full duplex) and enabling FIFO (SCxFCNF<CNFG> = "1").**

Settings and operations to transmit 4-byte data stream by setting the transfer mode to half duplex are shown as below.

SCxMOD1[6:5] =10	: Transfer mode is set to half duplex.
SCxFCNF[4:0] = 11011	: Transmission is automatically disabled if FIFO becomes empty. The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.
SCxTFC[1:0] = 00	: Sets the interrupt generation fill level to "0".
SCxTFC[7:6] = 11	: Clears receive FIFO and sets the condition of interrupt generation.

After above settings are configured, data transmission can be initiated by writing 5 bytes of data to the transmit buffer or FIFO, and setting the SCxMOD1<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.

Once above settings are configured, if the transmission is not set as auto disabled, the transmission should lasts by writing transmit data.



10.12.3.3 I/O interface Mode/Transmission by SCLK Output

If SCLK is set to generate clock the I/O interface mode, the SCLK output automatically stops when all data transmission is completed and underrun error will not occur.

The timing of suspension and resume of SCLK output is different depending on the buffer and FIFO usage.

(1) Single Buffer

The SCLK output stops each time one frame of data is transferred. Handshaking for each data with the other side of communication can be enabled. The SCLK output resumes when the next data is written in the buffer.

(2) Double Buffer

The SCLK output stops upon completion of data transmission of the transmit shift register and the transmit buffer. The SCLK output resumes when the next data is written in the buffer.

(3) FIFO

The transmission of all data stored in the transmit shift register, transmit buffer and FIFO is completed, the SCLK output stops. The next data is written, SCLK output resumes.

If SCxFCNF<RXTXCNT> is configured, SCxMOD0<TXE> bit is cleared at the same time as SCLK stop and the transmission stops.

10.12.3.4 Under-run error

If the transmit FIFO is disabled in the I/O interface SCLK input mode and if no data is set in transmit buffer before the next frame clock input, which occurs upon completion of data transmission from transmit shift register, an under-run error occurs and SCxCR<PERR> is set to "1".

In the I/O interface mode with SCLK output setting, the clock output automatically stops, so this flag has no meaning.

Note: **Before switching the I/O interface SCLK output mode to other modes, read the SCxCR register and clear the underrun flag.**

10.13 Handshake function

The function of the handshake is to enable frame-by-frame data transmission by using the CTS (Clear to send) pin and to prevent overrun errors. This function can be enabled or disabled by SCxMOD0<CTSE>.

When the $\overline{\text{CTS}}$ pin is set to "High" level, the current data transmission can be completed but the next data transmission is suspended until the $\overline{\text{CTS}}$ pin returns to the "Low" level. However in this case, the INTTXx interrupt is generated in the normal timing, the next transmit data is written in the transmit buffer, and it waits until it is ready to transmit data.

Note: **(1) If the $\overline{\text{CTS}}$ signal is set to "H" during transmission, the next data transmission is suspended after the current transmission is completed.**
(2) Data transmission starts on the first falling edge of the TXDCLK clock after $\overline{\text{CTS}}$ is set to "L".

Although no $\overline{\text{RTS}}$ pin is provided, a handshake control function can easily implemented by assigning one bit of the port for the $\overline{\text{RTS}}$ function. By setting the port to "High" level upon completion of data reception (in the receive interrupt routine), the transmit side can be requested to suspend data transmission.

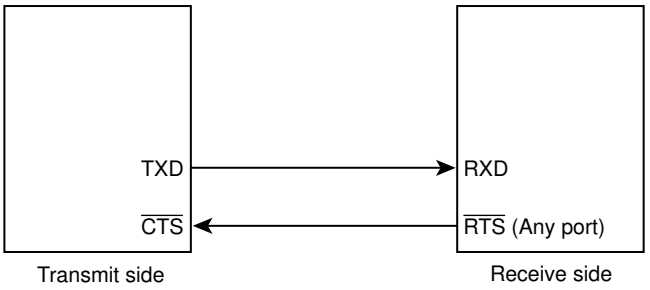


Figure 10-8 Handshake Function

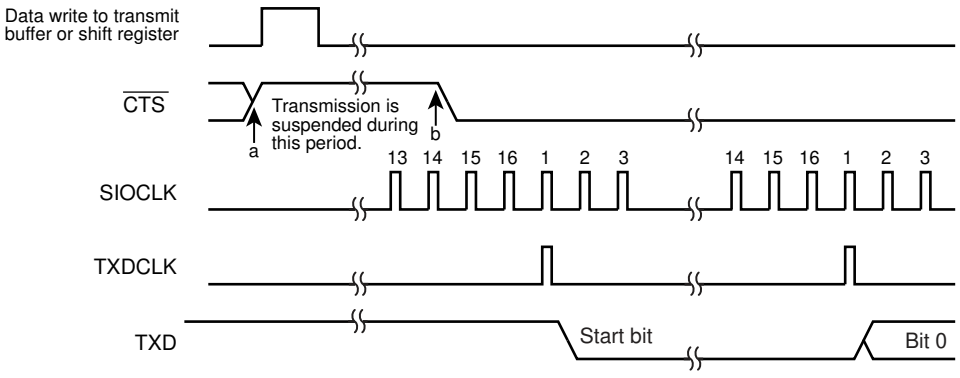


Figure 10-9 $\overline{\text{CTS}}$ Signal timing

10.14 Interrupt/Error Generation Timing

10.14.1 RX Interrupts

Figure 10-10 shows the data flow of receive operation and the route of read.

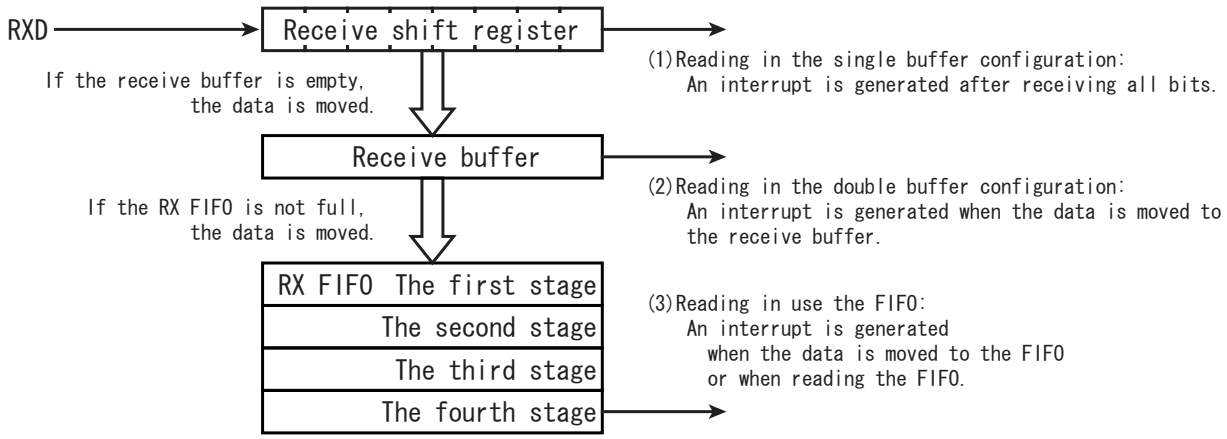


Figure 10-10 Receive Buffer/FIFO Configuration Diagram

10.14.1.1 Single Buffer / Double Buffer

RX interrupts are generated at the time depends on the transfer mode and the buffer configurations, which are given as follows.

Buffer Configurations	UART modes	IO interface modes
Single Buffer	−	・Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	・Around the center of the first stop bit	・Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.) ・On data transfer from the shift register to the buffer by reading buffer.

Note: Interrupts are not generated when an overrun error is occurred.

10.14.1.2 FIFO

In use of FIFO, receive interrupt is generated on the condition that the following either operation and SCxRFC<RFIS > setting are established.

- Reception completion of all bits of one frame.
- Reading FIFO

Interrupt conditions are decided by the SCxRFC<RFIS> settings as described in Table 10-12.

Table 10-12 Receive Interrupt conditions in use of FIFO

SCxRFC<RFIS>	Interrupt conditions
"0"	"The fill level of FIFO" is equal to "the fill level of FIFO interruption generation."
"1"	"The fill level of FIFO" is greater than or equal to "the fill level of FIFO intrruption generation."

10.14.2 TX interrupts

Figure 10-11 shows the data flow of transmit operation and the route of read.

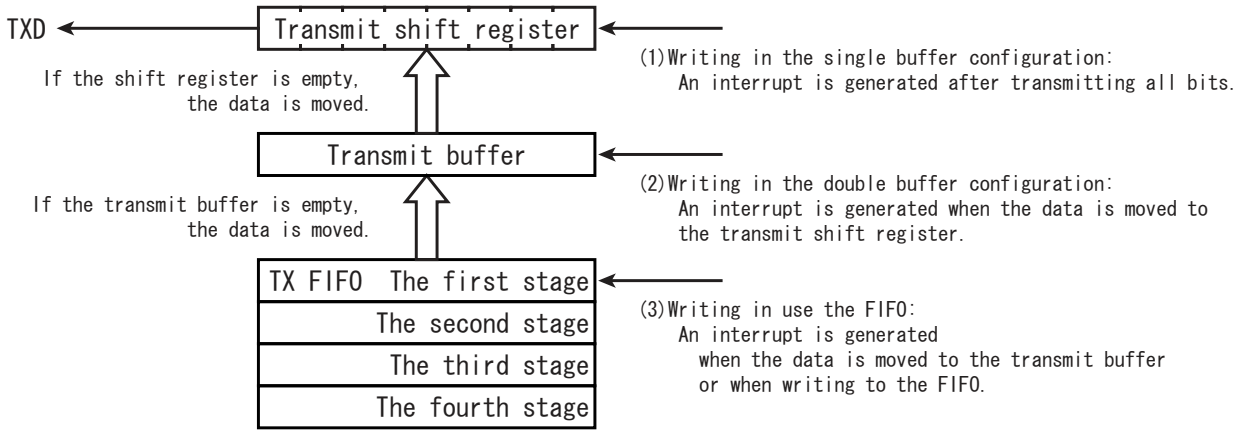


Figure 10-11 Transmit Buffer/FIFO Configuration Diagram

10.14.2.1 Single Buffer / Double Buffer

TX interrupts are generated at the time depends on the transfer mode and the buffer configurations, which are given as follows.

Buffer Configurations	UART modes	IO interface modes
Single Buffer	Just before the stop bit is sent	Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	When a data is moved from the transmit buffet to the transmit shift register.	

Note: If double buffer is enabled, a interrupt is also generated when the data is moved from the buffer to the shift register by writing to the buffer.

10.14.2.2 FIFO

In use of FIFO, transmit interrupt is generated on the condition that the fllowing either operation and SCxTFC<TFIS> setting are established.

- Transmittion completion of all bits of one frame.
- Writing FIFO

Interrupt conditions are decided by the SCxTFC<TFIS> settings as described in Table 10-13.

Table 10-13 Transmit Interrupt conditions in use of FIFO

SCxTFC<TFIS>	Interrupt conditions
"0"	"The fill level of FIFO" is equal to "the fill level of FIFO interruption generation."
"1"	"The fill level of FIFO" is smaller than or equal to "the fill level of FIFO intrruption generation."

10.14.3 Error Generation

10.14.3.1 UART Mode

modes	9 bits	7 bits 8 bits 7 bits+ Parity 8 bits + Parity
Framing Error Overrun Error	Around the center of stop bit	
Parity Error	−	Around the center of parity bit

10.14.3.2 IO Interface Mode

Overrun Error	Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Underrun Error	Immediately after the rising or falling edge of the next SCLK. (Rising or falling is determined according to SCxCR<SCLKS> setting.)

Note: Over-run error and Under-run error have no meaning in SCLK output mode.

10.15 Software Reset

Software reset is generated by writing SCxMOD2<SWRST[1:0]> as "10" followed by "01".

As a result, SCxMOD0<RXE>, SCxMOD1<TXE>, SCxMOD2<TBEMP><RBFLL><TXRUN>, SCxCR

<OERR><PERR><FERR> are initialized. And the receive circuit, the transmit circuit and the FIFO become initial stete. Other states are maintained.

10.16 Operation in Each Mode

10.16.1 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.

10.16.1.1 Transmitting Data

(1) SCLK Output Mode

- If the transmit double buffer is disabled (SCxMOD2<WBUF> = "0")

  Data is output from the TXD pin and the clock is output from the SCLK pin each time the CPU writes data to the transmit buffer. When all data is output, an interrupt (INTTXx) is generated.

- If the transmit double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data is moved from the transmit buffer to the transmit shift register when the CPU writes data to the transmit buffer while data transmission is halted or when data transmission from the transmit buffer (shift register) is completed. Simultaneously, the transmit buffer empty flag SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

  When data is moved from the transmit buffer to the transmit shift register, if the transmit buffer has no data to be moved to the transmit shift register, INTTXx interrupt is not generated and the SCLK output stops.

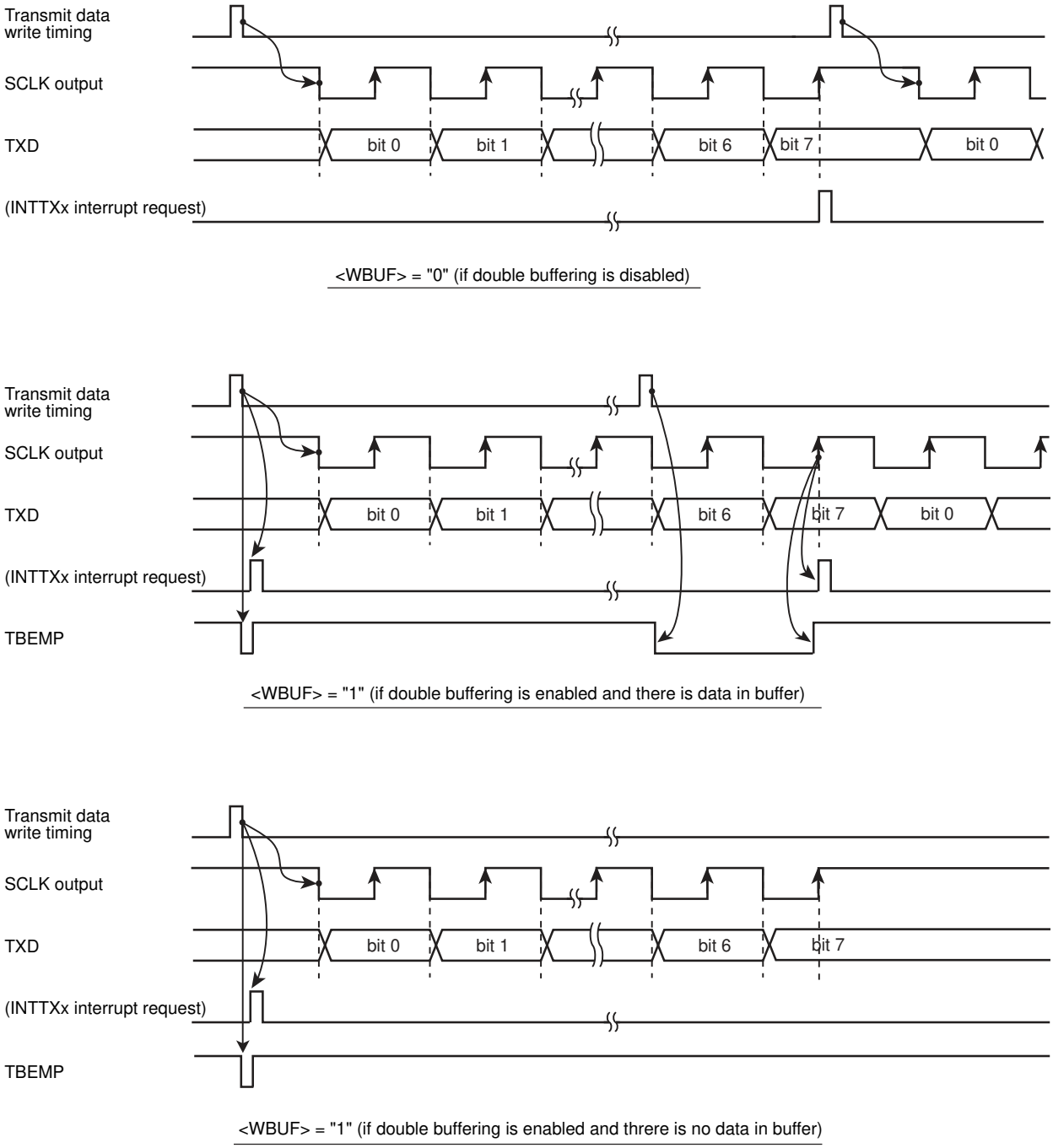


Figure 10-12 Transmit Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

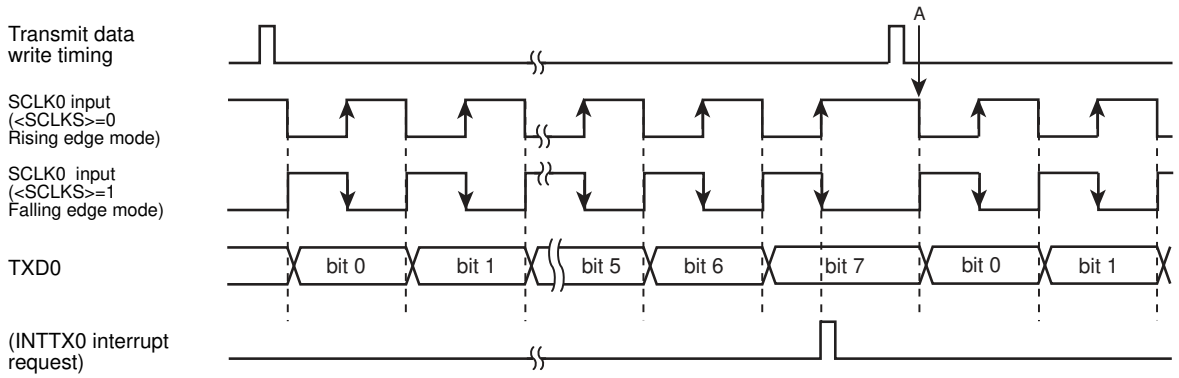
- If double buffering is disabled (SCxMOD2<WBUF> = "0")

  If the SCLK is input in the condition where data is written in the transmit buffer, 8-bit data is outputted from the TXD pin. When all data is output, an interrupt INTTXx is generated. The next transmit data must be written before the timing point "A" as shown in Figure 10-13.

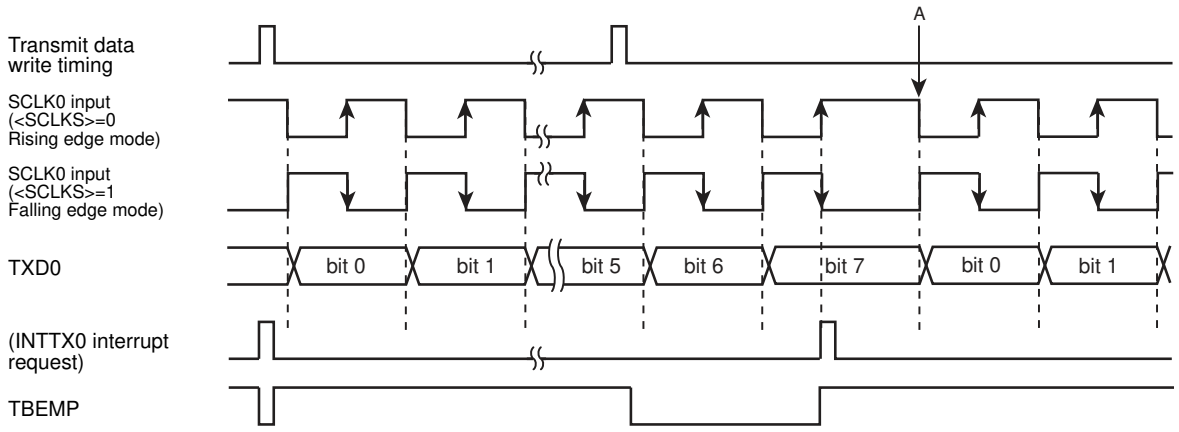
- If double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data is moved from the transmit buffer to the transmit shift register when the CPU writes data to the transmit buffer before the SCLK input becomes active or when data transmission from the transmit shift register is completed. Simultaneously, the transmit buffer empty flag SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

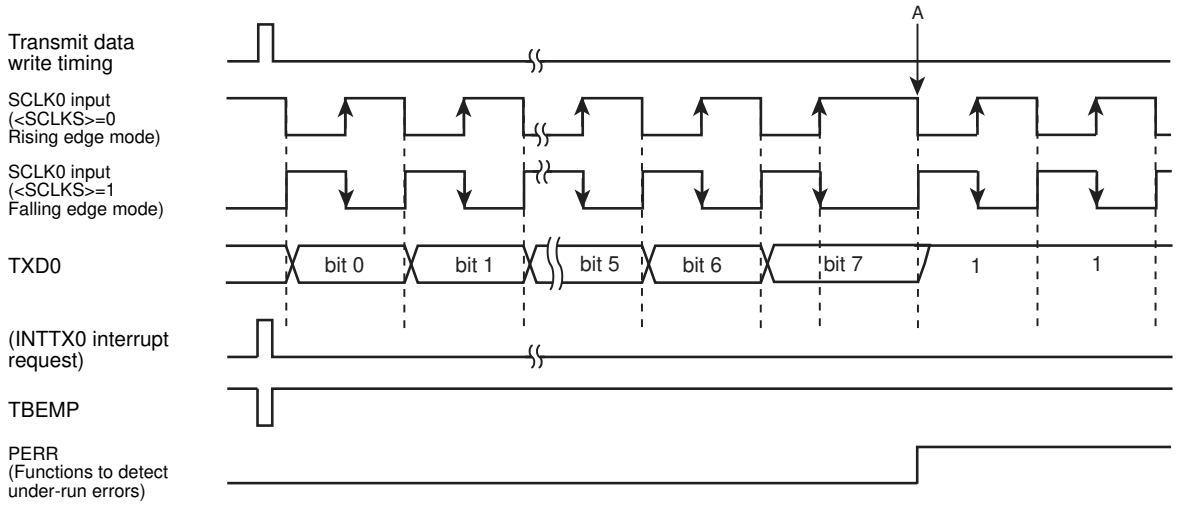
  If the SCLK input becomes active while no data is in the transmit buffer, although the internal bit counter is started, an under-run error occurs and 8-bit dummy data (0xFF) is sent.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled and there is data in buffer2)



<WBUF> = "1" (if double buffering is enabled and there is no data in buffer2)

Figure 10-13 Transmit Operation in the I/O Interface Mode (SCLK Input Mode)

10.16.1.2 Receive

(1) SCLK Output Mode

The SCLK output can be started by setting the receive enable bit SCxMOD0<RXE> to "1".

- If double buffer is disabled (SCxMOD2<WBUF> = "0")

  A clock pulse is outputted from the SCLK pin and the next data is stored into the shift register each time the CPU reads received data. When all the 8 bits are received, the INTRXx interrupt is generated.

- If double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data stored in the shift register is moved to the receive buffer and the receive buffer can receive the next frame. A data is moved from the shift register to the receive buffer, the receive buffer full flag SCxMOD2<RBFLL> is set to "1" and the INTRXx is generated.

  While data is in the receive buffer, if the data cannot be read from the receive buffer before completing reception of the next 8 bits, the INTRXx interrupt is not generated and the SCLK output stops. In this state, reading data from the receive buffer allows data in the shift register to move to the receive buffer and thus the INTRXx interrupt is generated and data reception resumes.

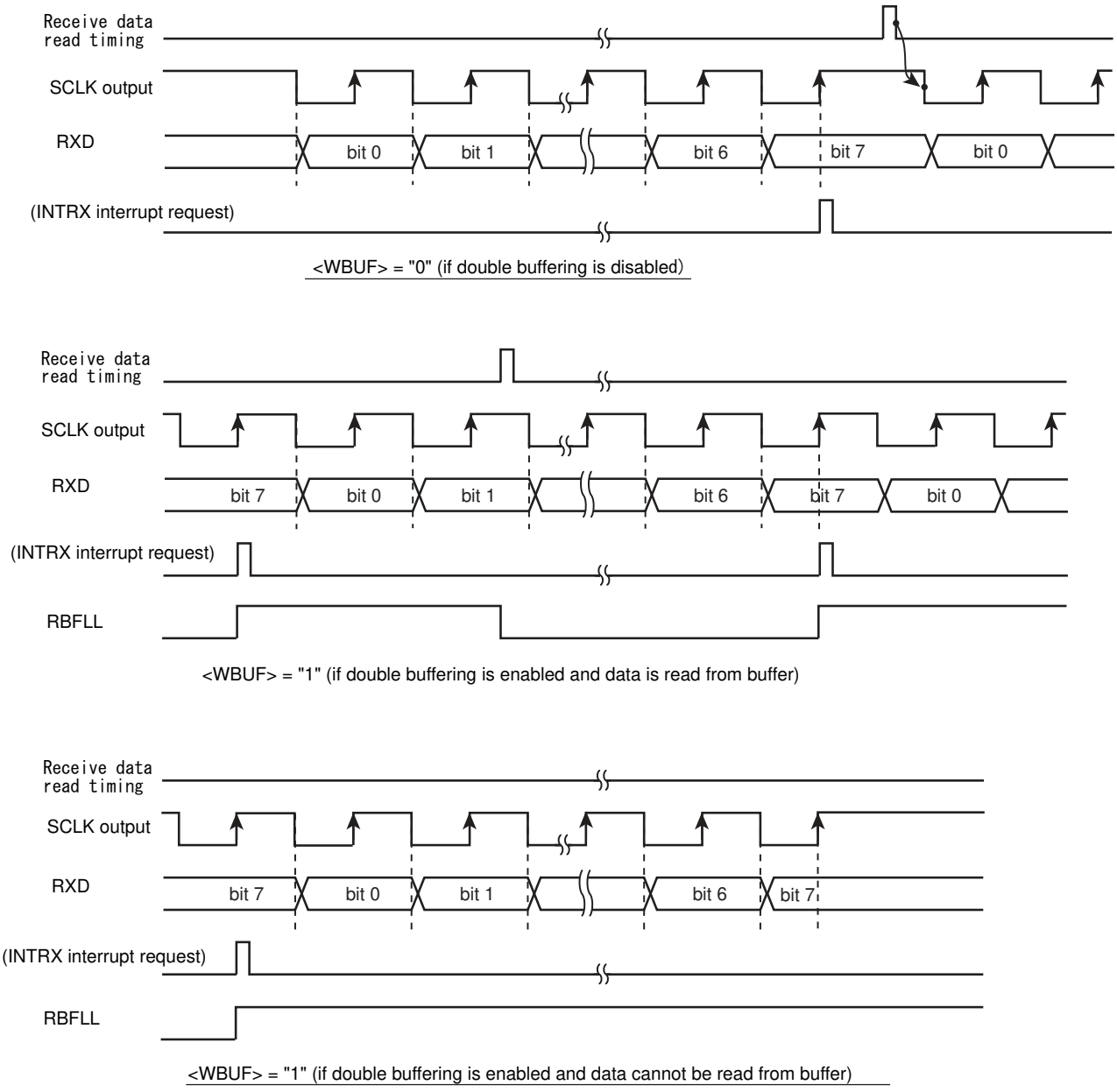


Figure 10-14 Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

In the SCLK input mode, receiving double buffering is always enabled, the received frame can be moved to the receive buffer from the shift register, and the receive buffer can receive the next frame successively.

The INTRx receive interrupt is generated each time received data is moved to the receive buffer.

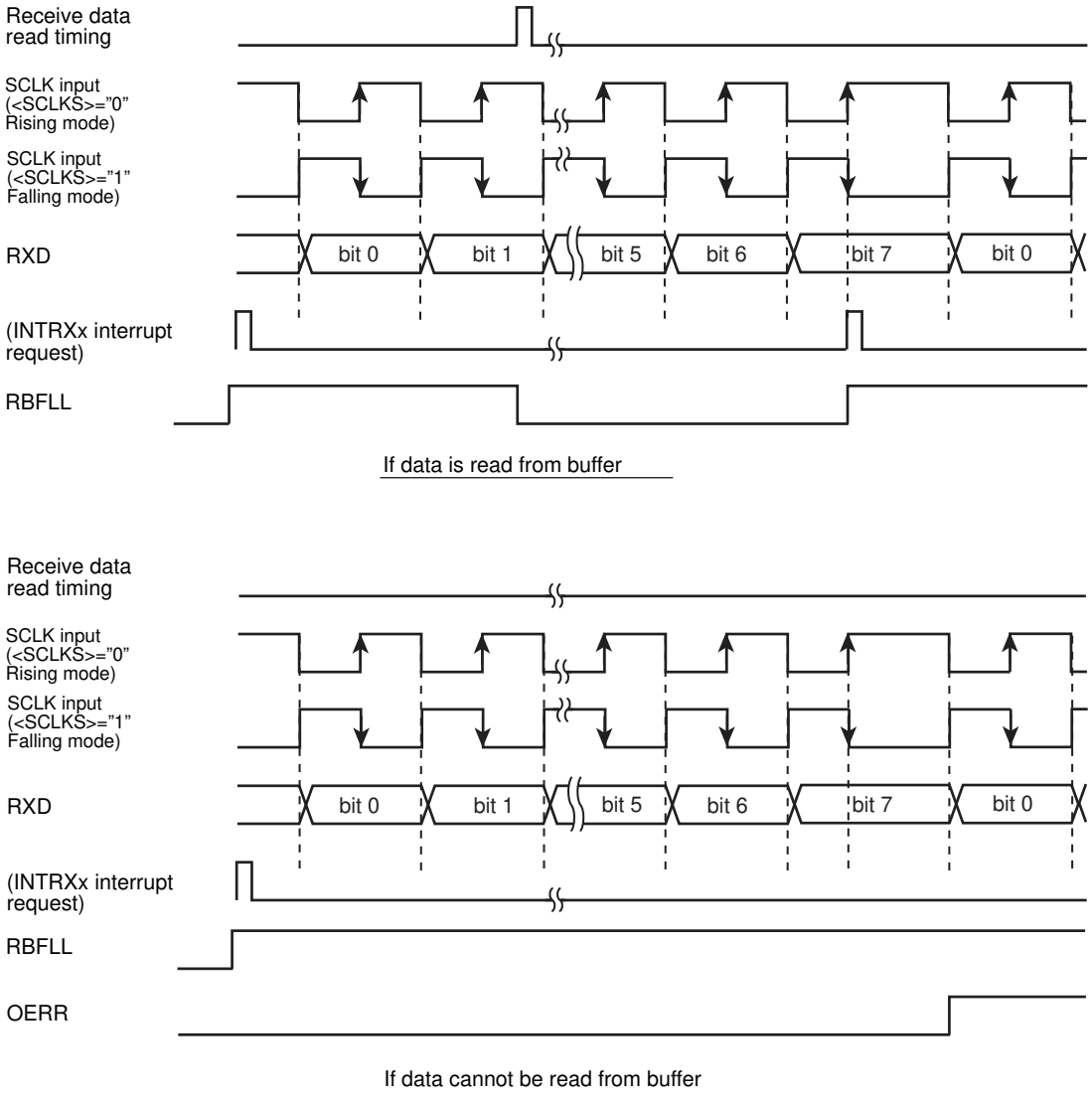


Figure 10-15 Receive Operation in the I/O Interface Mode (SCLK Input Mode)

10.16.1.3 Transmit and Receive (Full-duplex)

(1) SCLK Output Mode

- If SCxMOD2<WBUF> is set to "0" and the double buffers are disabled

  SCLK is outputted when the CPU writes data to the transmit buffer.

  Subsequently, 8 bits of data are shifted into receive buffer and the INTRXx receive interrupt is generated. Concurrently, 8 bits of data written to the transmit buffer are outputted from the TXD pin, the INTTXx transmit interrupt is generated when transmission of all data bits has been completed. Then, the SCLK output stops.

  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 SCxMOD2<WBUF> is set to "1" and the double buffers are enabled

  SCLK is outputted when the CPU writes data to the transmit buffer.

  8 bits of data are shifted into the receive shift register, moved to the receive buffer, and the INTRXx interrupt is generated. While 8 bits of data is received, 8 bits of transmit data is outputted from the TXD pin. When all data bits are sent out, the INTTXx interrupt is generated and the next data is moved from the transmit buffer to the transmit shift register.

  If the transmit buffer has no data to be moved to the transmit buffer (SCxMOD2<TBEMP> = 1) or when the receive buffer is full (SCxMOD2<RBFULL> = 1), the SCLK output 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 and reception is started.

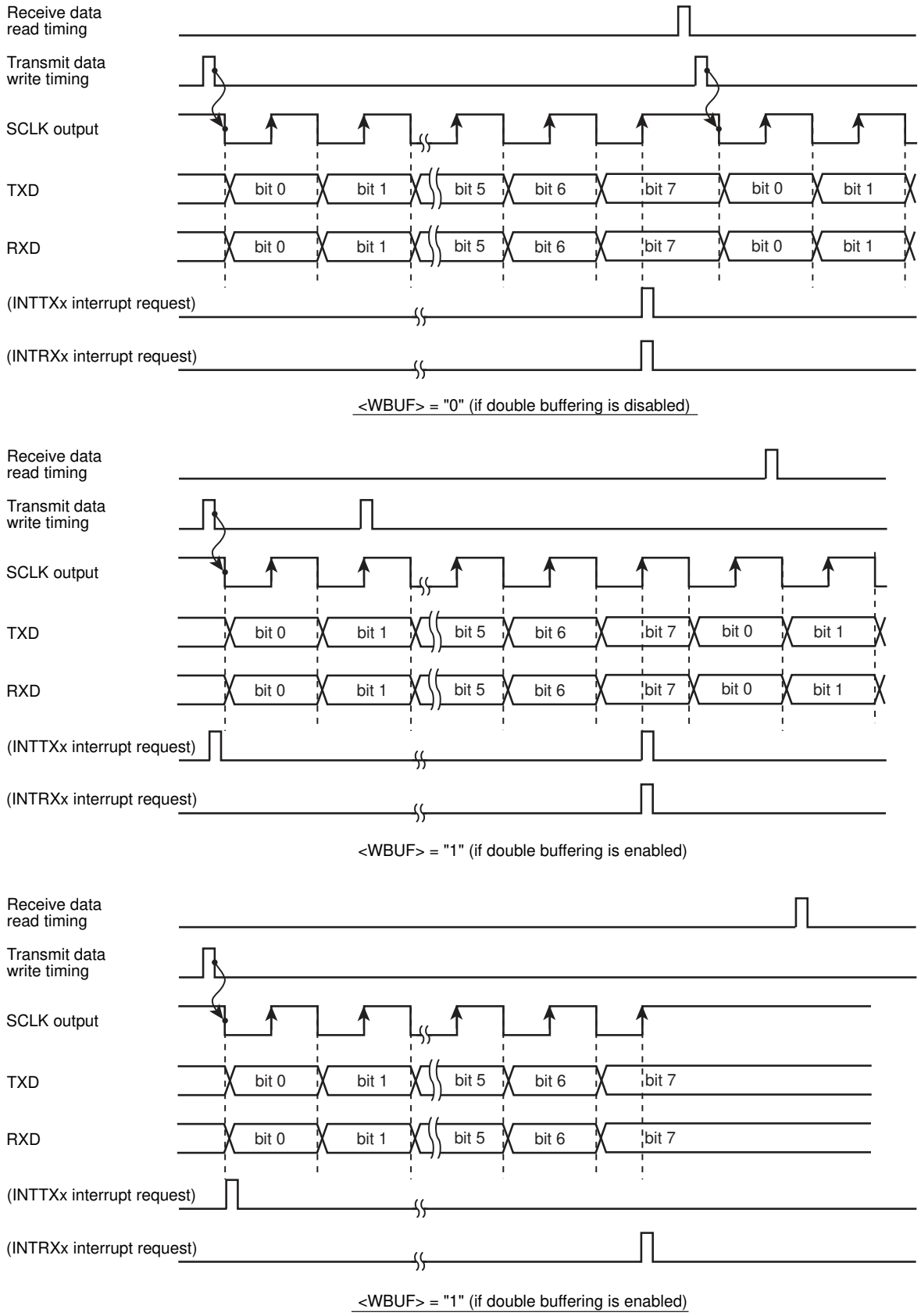


Figure 10-16 Transmit/Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

- If SCxMOD2<WBUF> is set to "0" and the transmit double buffer is disabled

  When receiving data, double buffer is always enabled regardless of the SCxMOD2 <WBUF> settings.

  8-bit data written in the transmit buffer is outputted from the TXD pin and 8 bit of data is shifted into the receive buffer when the SCLK input becomes active.The INTTXx interrupt is generated upon completion of data transmission. The INTTRXx interrupt is generated when the data is moved from shift register to receive buffer after completion of data reception.

  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 Figure 10-17). Data must be read before completing reception of the next frame data.

- If SCxMOD2<WBUF> is set to "1" and the double buffer is enabled.

  The interrupt INTRXx is generated at the timing the transmit buffer data is moved to the transmit shift register after completing data transmission from the transmit shift register. At the same time, data received is shifted to the shift register, it is moved to the receive buffer, and the INTRXx interrupt is generated.

  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 Figure 10-17). Data must be read before completing reception of the next frame data.

  Upon the SCLK input for the next frame, transmission from transmit shift register (in which data has been moved from transmit buffer) is started while receive data is shifted into receive shift register simultaneously.

  If data in receive buffer 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 when SCLK for the next frame is input, an under-run error occurs.

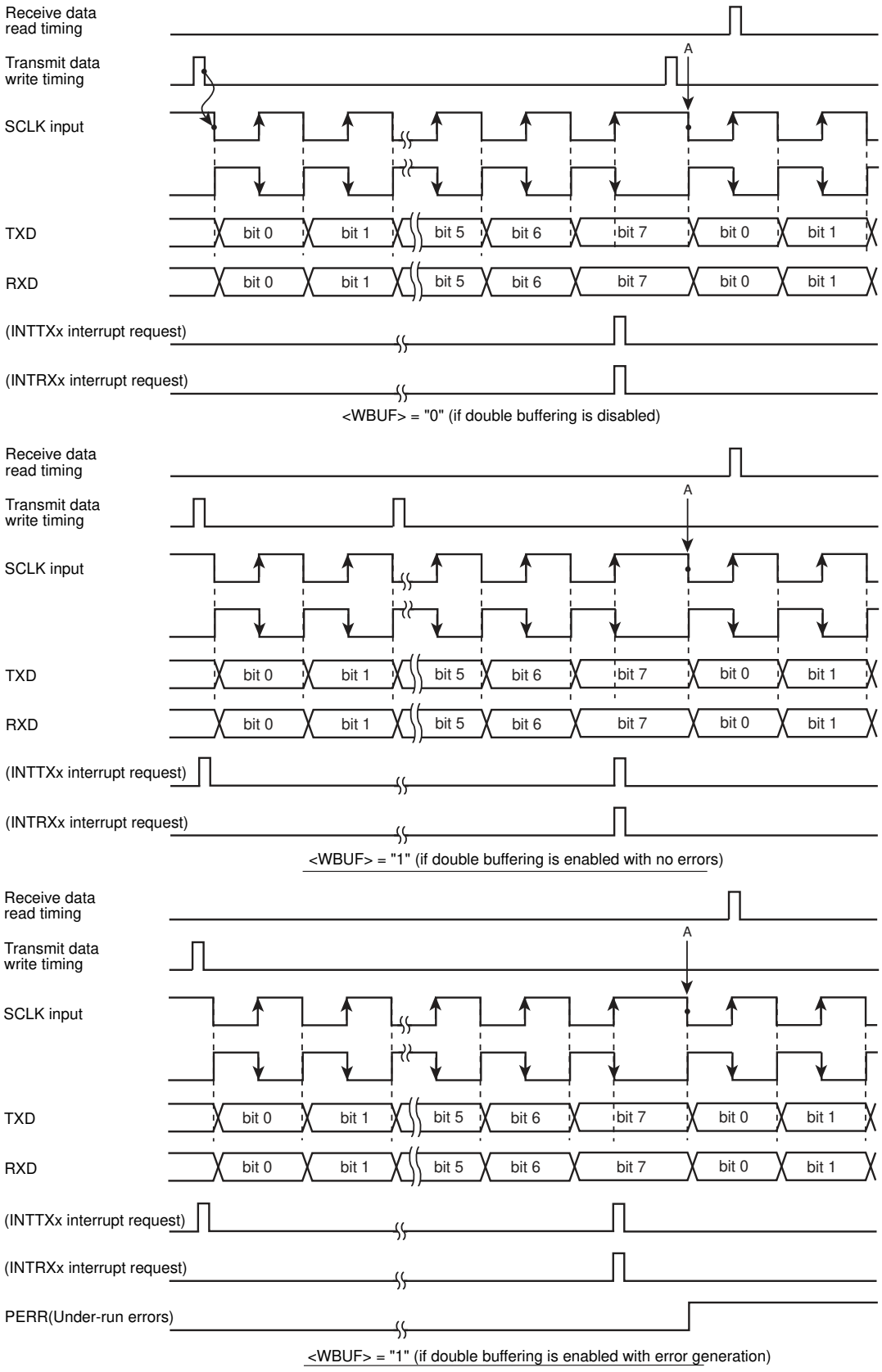


Figure 10-17 Transmit/Receive Operation in the I/O Interface Mode (SCLK Input Mode)

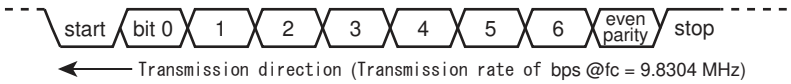
10.16.2 Mode 1 (7-bit UART mode)

The 7-bit UART mode can be selected by setting the serial mode control register (SCxMOD<SM[1:0]>) to "01".

In this mode, parity bits can be added to the transmit data stream; the serial mode control register (SCxCR<PE>) controls the parity enable/disable setting.

When <PE> is set to "1" (enable), either even or odd parity may be selected using the SCxCR<EVEN> bit. The length of the stop bit can be specified using SCxMOD2<SBLEN>.

The following table shows the control register settings for transmitting in the following data format.



Clocking conditions	system clock :	High-speed (fc)
	High-speed clock gear:	X1 (fc)
	Prescaler clock:	fperiph/2 (fperiph = fsys)

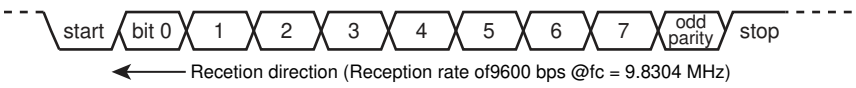
		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	-	0	0	1	0	1	Set 7-bit UART mode
SCxCR	←	x	1	1	x	x	x	0	0	Even parity enabled
SCxBRCR	←	0	0	1	0	0	1	0	0	Set 2400bps
SCxBUF	←	*	*	*	*	*	*	*	*	Set transmit data

x : don't care - : no change

10.16.3 Mode 2 (8-bit UART mode)

The 8-bit UART mode can be selected by setting SCxMOD0<SM[1:0]> to "10." In this mode, parity bits can be added and parity enable/disable is controlled using SCxCR<PE>. If <PE> = "1" (enabled), either even or odd parity can be selected using SCxCR<EVEN>.

The control register settings for receiving data in the following format are as follows:



Clocking conditions	System clock:	High-speed (fc)
	High-speed clock gear:	X1
	Prescaler clock:	fperiph/2 (fperiph = fsys)

		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	0	0	1	0	0	1	SEt 8-bit UART mode
SCxCR	←	x	0	1	x	x	x	0	0	Odd parity enabled
SCxBRCR	←	0	0	0	1	0	1	0	0	Set 9600bps
SCxMOD0	←	-	-	1	-	-	-	-	-	Reception enabled

x : don't care - : no change

10.16.4 Mode 3 (9-bit UART mode)

The 9-bit UART mode can be selected by setting SCxMOD0<SM[1:0]> to "11." In this mode, parity bits must be disabled (SCxCR<PE> = "0").

The most significant bit (9th bit) is written to bit 7 <TB8> of the serial mode control register 0 (SCxMOD0) for transmitting data. The data is stored in bit 7 <RB8> of the serial control register SCxCR.

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 SCxBUF.

The stop bit length can be specified using SCxMOD2<SBLEN>.

10.16.4.1 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 SCxMOD0<WU> to "1."

In this case, the interrupt INTRXx will be generated only when SCxCR<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.

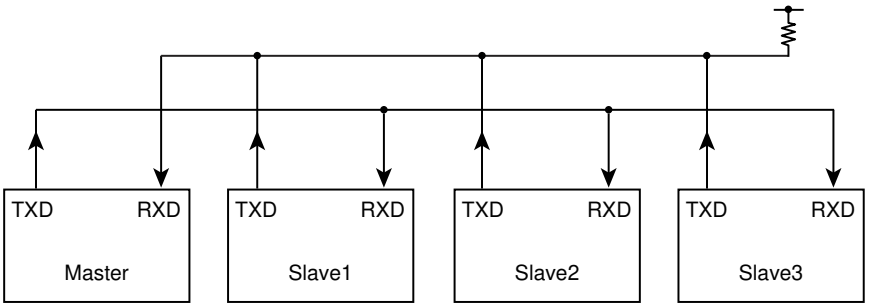
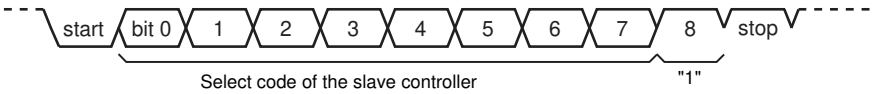


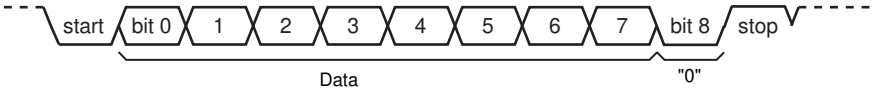
Figure 10-18 Serial Links to Use Wake-up Function

10.16.4.2 Protocol

1. Select the 9-bit UART mode for the master and slave controllers.
2. Set SCxMOD<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 SCxMOD<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 (INTRXx) 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.

11. Serial Bus Interface (I2C/SIO)

The TMPM330FDFG/FYFG/FWFG contains three Serial Bus Interface (I2C/SIO) channels, in which the following two operating modes are included:

- I2C bus mode (with multi-master capability)
- Clock-synchronous 8-bit SIO mode

In the I2C bus mode, the I2C/SIO is connected to external devices via SCL and SDA.

In the clock-synchronous 8-bit SIO mode, the I2C/SIO is connected to external devices via SCK, SI and SO.

The following table shows the programming required to put the I2C/SIO in each operating mode.

Table 11-1 Port settings for using serial bus interface

channel	Operating mode	pin	Port Function Register	Port Output Control Register	Port Input Control Register	Port Open Drain Output Control Register
SBI0	I2C bus mode	SCL0 :PG1 SDA0 :PG0	PGFR1[1:0] = "11"	PGCR[1:0] = "11"	PGIE[1:0] = "11"	PGOD[1:0] = "11"
	SIO mode	SCK0 :PG2 SI0 :PG1 SO0 :PG0	PGFR1[2:0] = "111"	PGCR[2:0] = "101" (SCK0 output) PGCR[2:0] = "001"(SCK0 input)	PGIE[2:0] = "010" (SCK0 output) PGIE[2:0] = "110" (SCK0 input)	PGOD[2:0] = "xxx"
SBI1	I2C bus mode	SCL1 :PF5 SDA1 :PF4	PFFR1[5:4] = "11"	PFCR[5:4] = "11"	PFIE[5:4] = "11"	PFOD[5:4] = "11"
	SIO mode	SCK1 :PF6 SI1 :PF5 SO1 :PF4	PFFR1[6:4] = "111"	PFCR[6:4] = "101" (SCK1output) PFCR[6:4] = "001" (SCK1 input)	PFIE[6:4] = "010" (SCK1 output) PFIE[6:4] = "110" (SCK1 input)	PFOD[6:4] = "xxx"
SBI2	I2C bus mode	SCL2 :PG5 SDA2 :PG4	PGFR1[5:4] = "11"	PGCR[5:4] = "11"	PGIE[5:4] = "11"	PGOD[5:4] = "11"
	SIO mode	SCK2 :PG6 SI2 :PG5 SO2 :PG4	PGFR1[6:4] = "111"	PGCR[6:4] = "101" (SCK2 output) PGCR[6:4] = "001" (SCK2 input)	PGIE[6:4] = "010" (SCK2 output) PGIE[6:4] = "110" (SCK2 input)	PGOD[6:4] = "xxx"

Note: x: Don't care

11.1 Configuration

The configuration is shown in Figure 11-1.

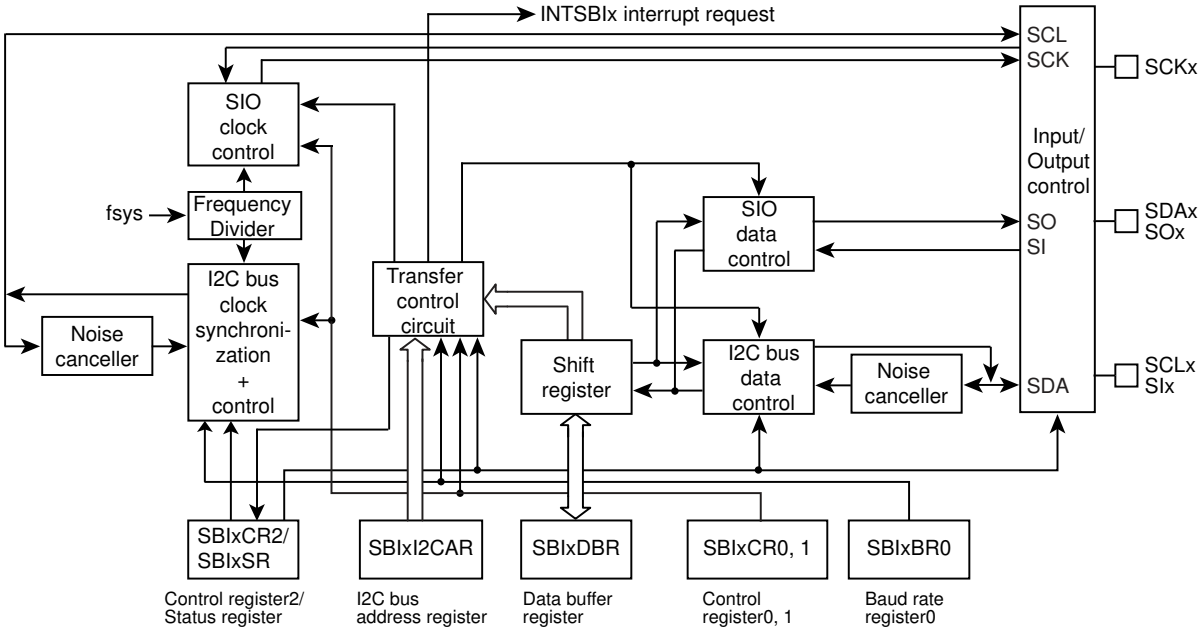


Figure 11-1 (I2C/SIO) Block Interface

11.2 Register

The following registers control the serial bus interface and provide its status information for monitoring.

The register below performs different functions depending on the mode. For details, refer to "11.4 Control Registers in the I2C Bus Mode" and "11.7 Control register of SIO mode".

11.2.1 Registers for each channel

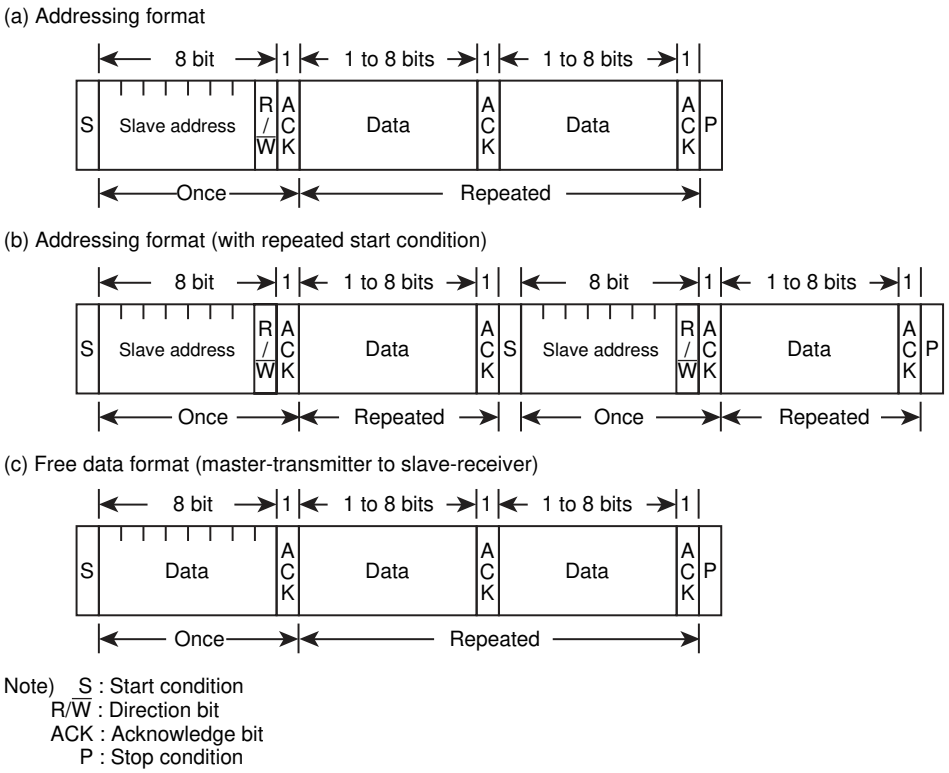
The tables below show the registers and register addresses for each channel.

Channel x	Base Address
Channel0	0x4002_0000
Channel1	0x4002_0020
Channel2	0x4002_0040

Register name (x=0 to 2)		Address(Base+)
Control register 0	SBIxCR0	0x0000
Control register 1	SBIxCR1	0x0004
Data buffer register	SBIxDBR	0x0008
I2C bus address register	SBIxI2CAR	0x000C
Control register 2	SBIxCR2 (writing)	0x0010
Status register	SBIxSR (reading)	
Baud rate register 0	SBIxBR0	0x0014

11.3 I2C Bus Mode Data Format

Figure 11-2 shows the data formats used in the I2C bus mode.



Note) S : Start condition
R/$\overline{W}$: Direction bit
ACK : Acknowledge bit
P : Stop condition

Figure 11-2 I2C Bus Mode Data Formats

11.4 Control Registers in the I2C Bus Mode

The following registers control the serial bus interface in the I2C bus mode and provide its status information for monitoring.

11.4.1 SBIxCR0(Control register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SBIEN	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SBIEN	R/W	Serial bus interface operation 0:Disable 1:Enable To use the serial bus interface, enable this bit first. Since all clocks except SBIxCR0 stop if this bit is disabled, power consumption can be reduced by disabling this bit. If this bit is disabled after it's been enabled once, the settings of each register are retained.
6-0	-	R	Read as 0.

11.4.2 SBIxCR1(Control register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	BC			ACK	-	SCK2	SCK1	SCK0 / SWRMON
After reset	0	0	0	0	1	0	0	1(Note3)

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-5	BC[2:0]	R/W	Select the number of bits per transfer (Note 1)
4	ACK	R/W	Master mode 0: Acknowledgement clock pulse is not generated. 1: Acknowledgement clock pulse is generated. Slave mode 0: Acknowledgement clock pulse is not counted. 1: Acknowledgement clock pulse is counted.
3	-	R	Read as 1.
2-1	SCK[2:1]	R/W	Select internal SCL output clock frequency (Note 2).
0	SCK[0]	W	
	SWRMON	R	On reading <SWRMON>: Software reset status monitor 0:Software reset operation is in progress. 1:Software reset operation is not in progress.

<BC>	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

000	n = 5	385 kHz
001	n = 6	294 kHz
010	n = 7	200 kHz
011	n = 8	122 kHz
100	n = 9	68 kHz
101	n = 10	36 kHz
110	n = 11	19 kHz
111		reserved

System Clock: fsys (= 40 MHz)
Clock gear : fc/1
Frequency = $\frac{fsys}{2^n + 72}$ [Hz]

Note 1: **Clear <BC[2:0]> to "000" before switching the operation mode to the SIO mode.**

Note 2: **For details on the SCL line clock frequency, refer to "11.5.1 Serial Clock".**

Note 3: **After a reset, the <SCK[0]/SWRMON> bit is read as "1". However, if the SIO mode is selected at the SBIxCR2 register, the initial value of the <SCK[0]> bit is "0".**

Note 4: **The initial value for selecting a frequency is <SCK[2:0]>=000 and is independent of the read initial value.**

11.4.3 SBIxCR2(Control register 2)

This register serves as SBIxSR register by reading it.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MST	TRX	BB	PIN	SBIM		SWRST	
After reset	0	0	0	1	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	W	Select master/slave 0: Slave mode 1: Master mode
6	TRX	W	Select transmit/ receive 0: Receive 1: Transmit
5	BB	W	Start/stop condition generation 0: Stop condition generated 1: Start condition generated
4	PIN	W	Clear INTSBIx interrupt request 0: − 1: Clear interrupt request
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note) 00: Port mode (Disables a serial bus interface output) 01: SIO mode 10: I2C bus mode 11: Reserved
1-0	SWRST[1:0]	W	Software reset generation Write "10" followed by "01" to generate a reset.

Note: **Make sure that modes are not changed during a communication session.Ensure that the bus is free before switching the operating mode to the port mode. Ensure that the port is at the "High" level before switching the operating mode from the port mode to the I2C bus or clock-synchronous 8-bit SIO mode.**

11.4.4 SBIxSR (Status Register)

This register serves as SBIxCR2 by writing to it.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MST	TRX	BB	PIN	AL	AAS	ADO	LRB
After reset	0	0	0	1	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	R	Master/slave selection monitor 0: Slave mode 1: Master mode
6	TRX	R	Transmit/receive selection monitor 0: Receive 1: Transmit
5	BB	R	I2C bus state monitor 0: Free 1: Busy
4	PIN	R	INTSBIx interrupt request monitor 0:Interrupt request generated 1: Interrupt request cleared
3	AL	R	Arbitration lost detection 0: − 1:Detected
2	AAS	R	Slave address match detection 0: − 1: Detected (This bit is set when the general call is detected as well.)
1	ADO	R	General call detection 0: − 1:Detected
0	LRB	R	Last received bit monitor 0:Last received bit "0" 1:Last received bit "1"

11.4.5 SBIxBR0(Serial bus interface baud rate register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
After reset	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	-	R	Read as 1.
6	I2SBI	R/W	Operation at the IDLE mode 0: Stop 1: Operate
5-1	-	R	Read as 1.
0	-	R/W	Be sure to write "0".

11.4.6 SBIxDBR (Serial bus interface data buffer register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	DB							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DB[7:0]	R (Receive)/ W (Transmit)	Receive data / Transmit data

Note 1: **The transmission data must be written in to the register from the MSB (bit 7). The received data is stored in the LSB.**

Note 2: **Since SBIxI2CAR has independent buffers for writing and reading, a written data cannot be read. Thus, read-modify-write instructions, such as bit manipulation, cannot be used.**

11.4.7 SBIxI2CAR (I2Cbus address register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SA							ALS
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-1	SA[6:0]	R/W	Set the slave address when the SBI acts as a slave device.
0	ALS	R/W	Specify address recognition mode. 0: Recognize its slave address. 1: Do not recognize its slave address (free-data format).

Note 1: **Please set the bit 0 <ALS> of I2C bus address register SBIxI2CAR 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.**

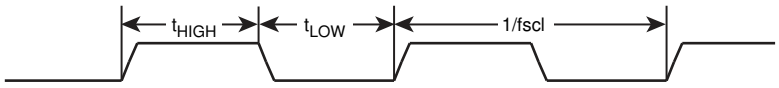
Note 2: **Do not set SBIxI2CAR to "0x00" in slave mode. (If SBIxI2CAR is set to "0x00", it's recognized that the slave address matches the START byte ("0x01") of the I2C standard received in slave mode.)**

11.5 Control in the I2C Bus Mode

11.5.1 Serial Clock

11.5.1.1 Clock source

SBIxCR1<SCK[2:0]> specifies the maximum frequency of the serial clock to be output from the SCL pin in the master mode.



$t_{LOW} = 2^{n-1}/fsys + 58/fsys$
$t_{HIGH} = 2^{n-1}/fsys + 14/fsys$
$fscl = 1/(t_{LOW} + t_{HIGH})$
$= \frac{fsys}{2^n + 72}$

SBIxCR1<SCK[2:0]>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Figure 11-3 Clock source

Note: **The maximum speeds in the standard and high-speed modes are specified to 100kHz and 400kHz respectively following the communications standards. Notice that the internal SCL clock frequency is determined by the fsys used and the calculation formula shown above.**

11.5.1.2 Clock Synchronization

The I2C bus is driven by using the wired-AND connection due to its pin structure. The first master that pulls its clock line to the "Low" level overrides other masters producing the "High" level on their clock lines. This must be detected and responded by the masters producing the "High" 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.

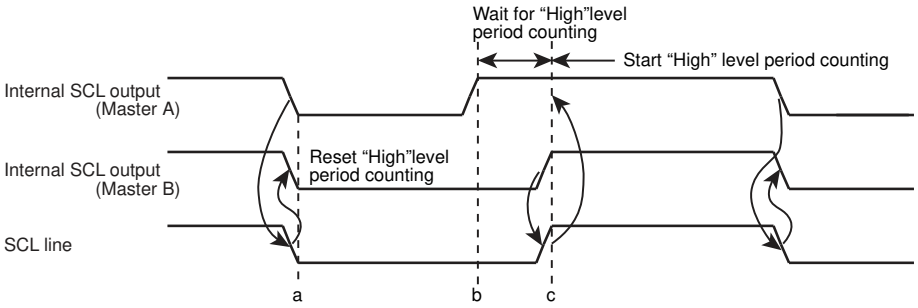


Figure 11-4 Example of Clock Synchronization

At the point a, Master A pulls its internal SCL output to the "Low" level, bringing the SCL bus line to the "Low" level. Master B detects this transition, resets its "High" level period counter, and pulls its internal SCL output level to the "Low" level.

Master A completes counting of its "Low" level period at the point b, and brings its internal SCL output to the "High" level. However, Master B still keeps the SCL bus line at the "Low" level, and Master A stops counting of its "High" level period counting.After Master A detects that Master B brings its internal SCL output to the "High" level and brings the SCL bus line to the "High" level at the point c, it starts counting of its "High" level period.

After that Master finishes counting the "High" level period, the Master pulls the SCL pin to "Low" and the SCL bus line becomes "Low".

This way, the clock on the bus is determined by the master with the shortest "High" level period and the master with the longest "Low" level period among those connected to the bus.

11.5.2 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 signal. In slave mode, the clock for acknowledgement signals is counted. In transmitter mode, the SBI releases the SDAx pin during clock cycle to receive acknowledgement signals from the receiver. In receiver mode, the SBI pulls the SDAx pin to the "Low" level during the clock cycle and generates acknowledgement signals. Also in slave mode, if a general-call address is received, the SBI pulls the SDAx pin to the "Low" level during the clock cycle and generates acknowledgement 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. In slave mode, the clock for acknowledgement signals is counted.

11.5.3 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.

11.5.4 Slave Addressing and Address Recognition Mode

Setting "0" to SBIxI2CAR<ALS> and a slave address in SBIxI2CAR<SA[6:0]> sets addressing format, and then 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. In the case of free data format, a slave address and a direction bit are not recognized; they are recognized as data immediately after generation of the start condition.

11.5.5 Operating mode

The setting of SBIxCR2<SBIM[1:0]> controls the operating mode. To operate in I2C mode, ensure that the serial bus interface pins are at "High" level before setting <SBIM[1:0]> to "10". Also, ensure that the bus is free before switching the operating mode to the port mode.

11.5.6 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:

- when data is transmitted in the addressing format.
- when the received slave address matches the value specified at SBIxI2CAR.
- 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.

If SBI is used in free data format, <TRX> is not changed by the hardware.

11.5.7 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.

11.5.8 Generating Start and Stop Conditions

When SBIxSR<BB> is "0", writing "1" to SBIxCR2<MST, TRX, BB, PIN> causes the SBI to start a sequence for generating the start condition and to output the slave address and the direction bit prospectively written in the data buffer register. <ACK> must be set to "1" in advance.

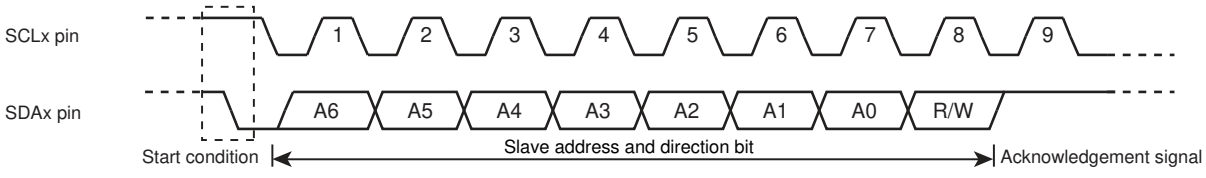


Figure 11-5 Generating the Start Condition and a Slave Address

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.

If SCL bus line is pulled "Low" by other devices when the stop condition is generated, the stop condition is generated after the SCL line is released.

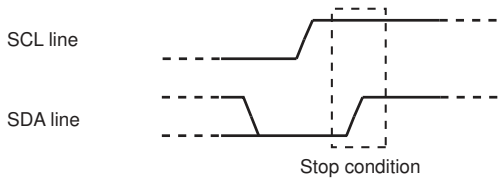


Figure 11-6 Generating the Stop Condition

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 cleared to "0" when the stop condition is detected (the bus is free).

11.5.9 Interrupt Service Request and Release

In master mode, a serial bus interface request (INTSBIx) is generated when the transfer of the number of clock cycles set by <BC> and <ACK> is completed.

In slave mode, INTSBIx is generated under the following conditions.

- After output of the acknowledge signal which is generated when the received slave address matches the slave address set to SBIxI2CAR<SA[6:0]>.
- After the acknowledge signal is generated when a general-call address is received.
- When the slave address matches or a data transfer is completed after receiving a general-call address.

In the address recognition mode (<ALS> = "0"), INTSBIx is generated when the received slave address matches the values specified at SBIxI2CAR or when a general-call (eight bits data following the start condition is all "0") is received.

When an interrupt request (INTSBIx) is generated, SBIxCR2<PIN> is cleared to "0". While <PIN> is cleared to "0", the SBI pulls the SCL line to the "Low" level.

<PIN> 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". When the program writes "1" to <PIN>, it is set to "1". However, writing "0" does not clear this bit to "0".

Note: When arbitration is lost in master mode, <PIN> is not cleared to "0" if the slave address does not match (INTSBIx is generated).

11.5.10 Arbitration Lost Detection Monitor

The I2C 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 I2C-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 "Low" level and Master B outputs the "High" level.

Then Master A pulls the SDA bus line to the "Low" 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 "Arbitration Lost". 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.

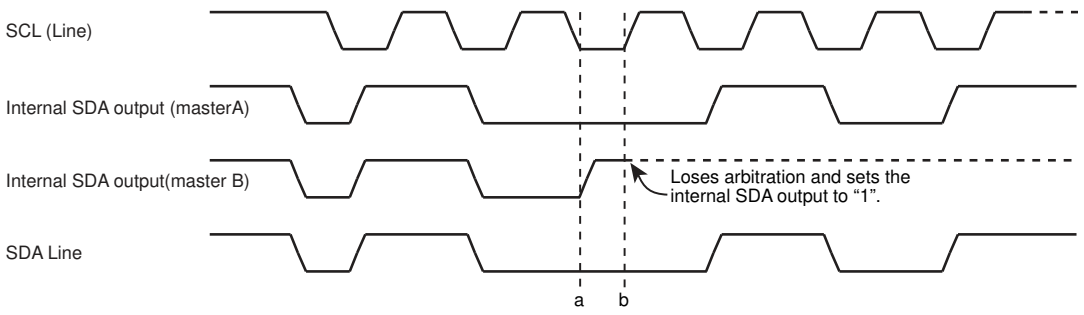


Figure 11-7 Lost Arbitration

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.Therefore, the serial bus interface circuit stops the clock output during data transfer after <AL> is set to "1".

<AL> is cleared to "0" when data is written to or read from SBIxDBR or data is written to SBIxCR2.

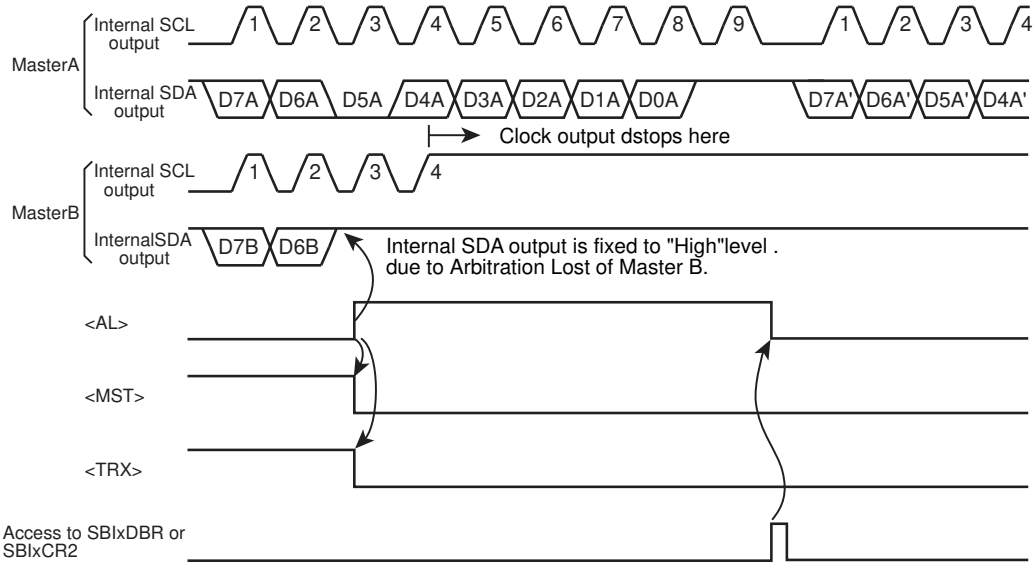


Figure 11-8 Example of Master B Lost Arbitration (D7A = D7B, D6A = D6B)

11.5.11 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.

11.5.12 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.

11.5.13 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.

11.5.14 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.

11.5.15 Baud Rate 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.

11.5.16 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 I2C mode to the port mode.

11.6 Data Transfer Procedure in the I2C Bus ModeI2C

11.6.1 Device Initialization

First, program SBIxCR1<ACK, SCK[2:0]>. Writing "000" to SBIxCR1<BC[2:0]> at the time.

Next, program SBIxI2CAR by specifying a slave address at <SA[6:0]> and an address recognition mode at <ALS>. (<ALS> must be cleared to "0" when using the addressing format).

To configure the Serial Bus Interface as a slave receiver, ensure that the serial bus interface pin is at "High" first. Then write "0" to SBIxCR2<MST, TRX, BB>, "1" to <PIN>, "10" to <SBIM[1:0]> and "0" to the bit 1 and 0.

Note: Initialization of the serial bus interface circuit must be completed within a period that any device does not generate start condition after all devices connected to the bus were initialized. If this rule is not followed, data may not be received correctly because other devices may start transfer before the initialization of the serial bus interface circuit is completed.

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	0	0	X	0	X	X	X	Specifies ACK and SCL clock.
SBIxI2CAR	←	X	X	X	X	X	X	X	X	Specifies a slave address and an address recognition mode.
SBIxCR2	←	0	0	0	1	1	0	0	0	Configures the SBI as a slave receiver.

Note: X; Don't care

11.6.2 Generating the Start Condition and a Slave Address

11.6.2.1 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 "Low" 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.

Note: To output salve address, check with software that the bus is free before writing to SBIxDBR. If this rule is not followed, data being output on the bus may get ruined.

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
```

11.6.2.2 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 "Low" 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 "Low" level while <PIN> is "0".

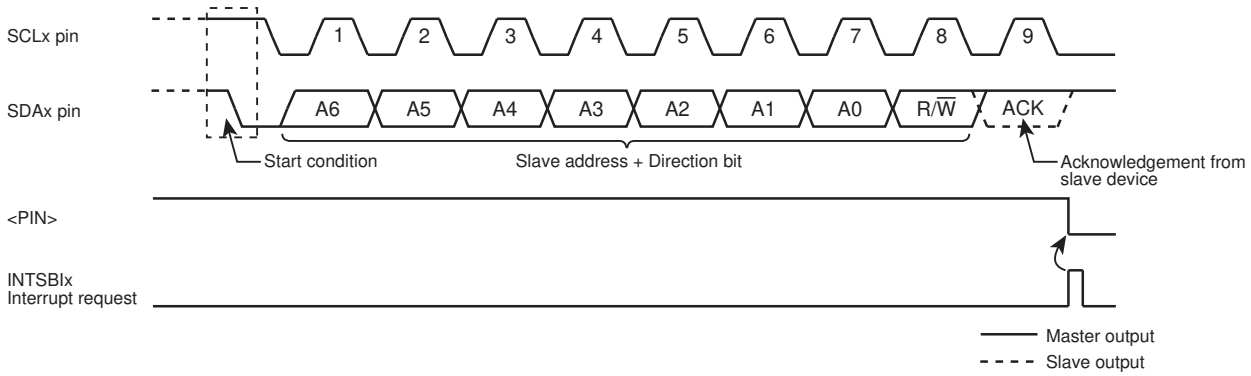


Figure 11-9 Generation of the Start Condition and a Slave Address

11.6.3 Transferring 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.

11.6.3.1 Master mode (<MST> = "1")

Test <TRX> to determine whether the SBI is configured as a transmitter or a receiver.

(1) 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 cleared to "0", and the SCL pin is pulled to the "Low" 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**

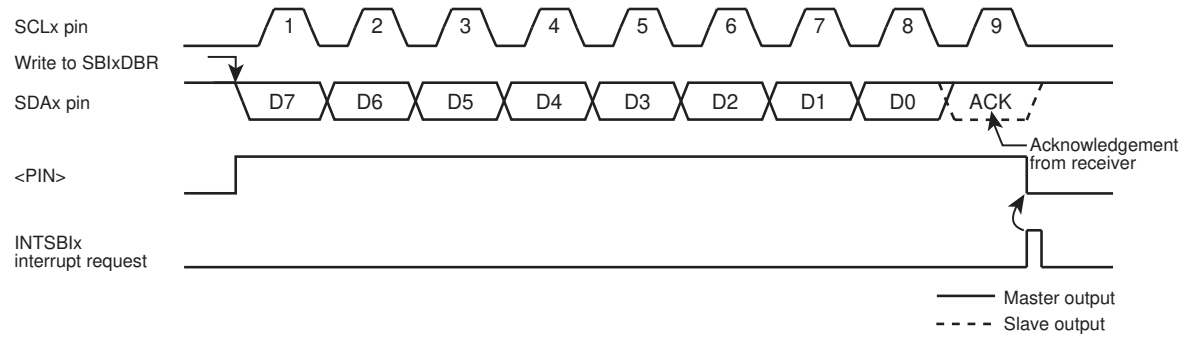


Figure 11-10 <BC[2:0]>= "000",<ACK>= "1" (Transmitter Mode)

(2) 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 "Low" 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 "Low" level.Each time the received data is read from SBIxDBR, one-word transfer clock and an acknowledgement signal are output.

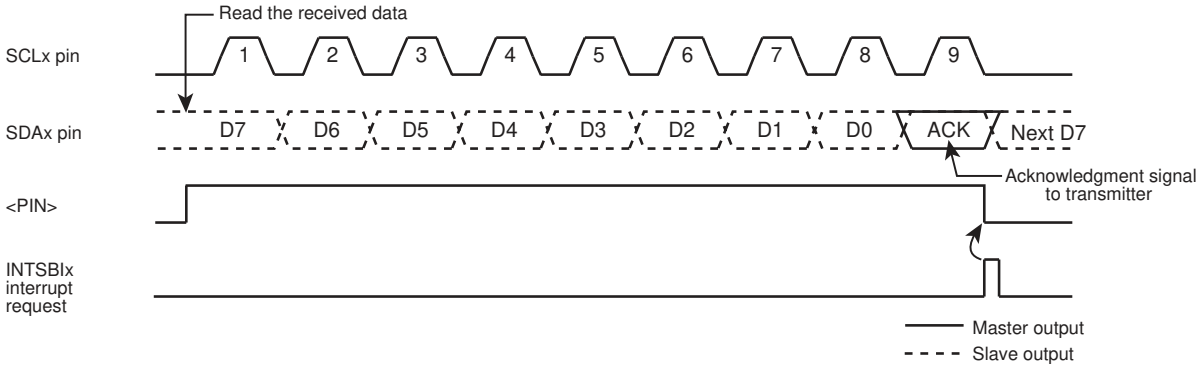


Figure 11-11 <BC[2:0]>= "000",<ACK>= "1" (Receiver Mode)

To terminate the data transmission from the transmitter, <ACK> must be cleared 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 "High" 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.

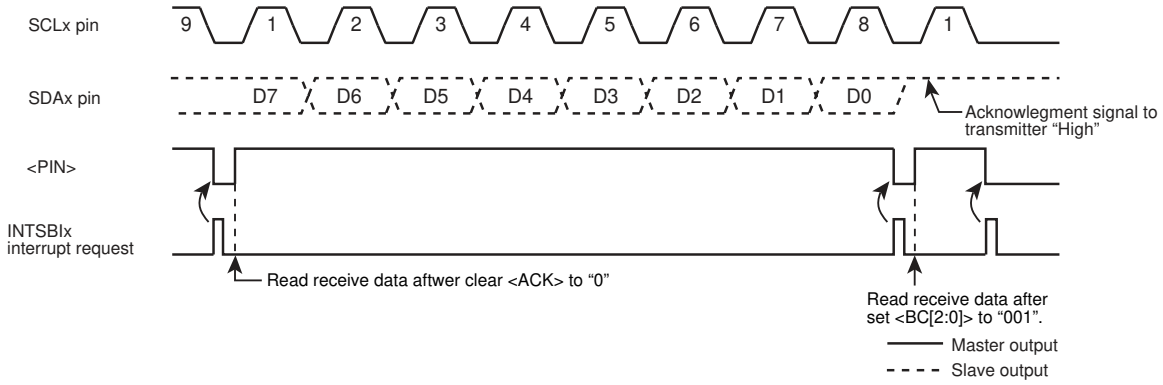


Figure 11-12 Terminating Data Transmission in the Master Receiver Mode

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

11.6.3.2 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 address.

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 "Low" level.

When data is written to or read from SBIxDBR or when <PIN> is set to "1", the SCLx 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 11-2 Processing in Slave Mode"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 = 0
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 11-2 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	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.

11.6.4 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.

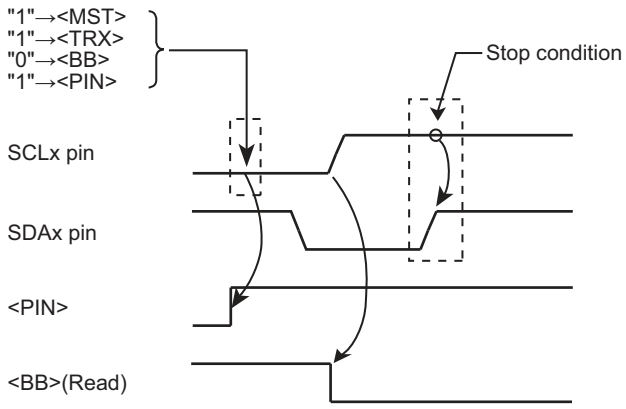


Figure 11-13 Generating the Stop Condition

11.6.5 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, write SBIxCR2<MST, TRX, BB> to "0" and write "1" to <PIN> to release the bus. At this time, the SDAx pin is held at the "High" level and the SCLx 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 SCLx pin is released.

Next, test <LRB> and wait until it becomes "1" to ensure that no other device is pulling the SCLx bus line to the "Low" level.

Once the bus is determined to be free by following the above procedures, follow the procedures described in "11.6.2 Generating the Start Condition and a Slave Address"to generate the start condition.

To satisfy the setup time of restart, at least 4.7μs wait period (in the standard mode) must be created by the software after the bus is determined to be free.

Note 1: **Do not write <MST> to "0" when it is "0". (Restart cannot be initiated.)**

Note 2: **When the master device is acting as a receiver, data transmission from the slave device which serves as a transmitter must be completed before generating a restart. To complete data transfer, slave device must receive a "High" level acknowledge signal. For this reason, <LBR> before generating a**

restart becomes "1", the rising edge of the SCL line is not detected even <LBR>= "1" is confirmed by following the restart procedure. To check the status of the SCL line, read the port.

		7	6	5	4	3	2	1	0	
SBIxCR2	←	0	0	0	1	1	0	0	0	Releases the bus.
if SBIxSR<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 "Low".
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

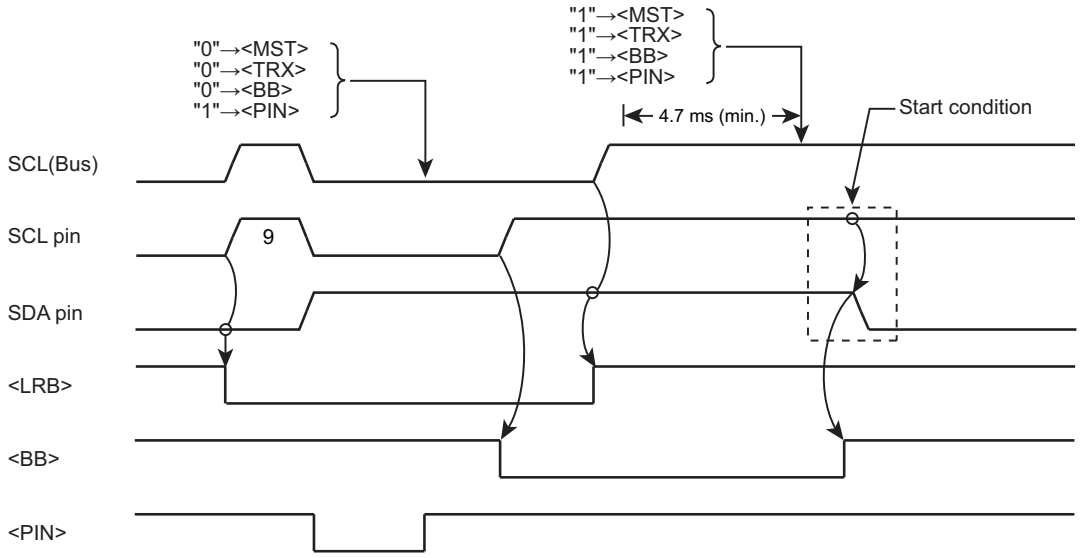


Figure 11-14 Timing Chart of Generating a Restart

11.7 Control register of 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.

11.7.1 SBIxCR0(control register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SBIEN	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SBIEN	R/W	Serial bus interface operation. 0:Disable 1: Enable Enable this bit before using the serial bus interface. If this bit is disabled, power consumption can be reduced because all clocks except SBIxCR0 stop. If the serial bus interface operation is enabled and then disabled, the settings will be maintained in each register.
6-0	-	R	Read as 0.

11.7.2 SBIxCR1(Control register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SIOS	SIOINH	SIOM		-	SCK		
After reset	0	0	0	0	1	0	0	0(Note 1)

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SIOS	R/W	Transfer Start/Stop 0: Stop 1: Start
6	SIOINH	R/W	Transfer 0: Continue 1: Forced termination
5-4	SIOM[1:0]	R/W	Select transfer mode 00: Transmit mode 01: Reserved 10:Transmit/receive mode 11:Receive mode
3	-	R	Read as 1.
2-0	SCK[2:0]	R/W	On writing <SCK[2:0]>: Select serial clock frequency. (Note 1)

000	n = 3	2.5 MHz
001	n = 4	1.25 MHz
010	n = 5	625 kHz
011	n = 6	313 kHz
100	n = 7	156 kHz
101	n = 8	78 kHz
110	n = 9	39 kHz
111	−	External clock

System clock: fsys (= 40 MHz)
Clock gear: fc/1
Frequency = $\frac{fsys/2}{2^n}$ [Hz]

Note 1: **After a reset, the <SCK[0]> bit is read as "1". However, if the SIO mode is selected at the SBIxCR2 register, the initial value is read as "0". In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state. The descriptions of the SBIxCR2 register and the SBIxSR register are the same.**

Note 2: **Set <SIOS> to "0" and <SIOINH> to "1" before programming the transfer mode and the serial clock.**

11.7.3 SBIxDBR (Data buffer register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	DB							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DB[7:0]	R	Receive data
		W	Transmit data

Note 1: **The transmission data must be written in to the register from the MSB (bit 7). The received data is stored in the LSB.**

Note 2: **Since SBIxI2CAR has independent buffers for writing and reading, a written data cannot be read. Thus, read-modify-write instructions, such as bit manipulation, cannot be used.**

11.7.4 SBIxCR2(Control register 2)

This register serves as SBIxSR register by writing to it.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	SBIM		-	-
After reset	1(Note 1)	1(Note 1)	1(Note 1)	1(Note 1)	0	0	1(Note 1)	1(Note 1)

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-4	-	R	Read as 1. (Note 1)
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note 2) 00: Port mode 01: SIO mode 10: I2Cbus mode 11: Reserved
1-0	-	R	Read as 1. (Note 1)

Note 1: **In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state.**

Note 2: **Make sure that modes are not changed during a communication session.**

11.7.5 SBIxSR (Status Register)

This register serves as SBIxCR2 by writing to it.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	SIOF	SEF	-	-
After reset	1(Note 1)	1(Note 1)	1(Note 1)	1(Note 1)	0	0	1(Note 1)	1(Note 1)

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-4	-	R	Read as 1.(Note 1)
3	SIOF	R	Serial transfer status monitor. 0: Completed 1: In progress
2	SEF	R	Shift operation status monitor 0: Completed. 1: In progress
1-0	-	R	Read as 1. (Note 1)

Note: **In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state.**

11.7.6 SBIxBR0 (Baud rate register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
After reset	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	-	R	Read as 1.
6	I2SBI	R/W	Operation in IDLE mode. 0: Stop 1: Operate
5-1	-	R	Read as 1.
0	-	R/W	Make sure to write "0".

11.8 Control in SIO mode

11.8.1 Serial Clock

11.8.1.1 Clock source

Internal or external clocks can be selected by programming SBIxCR1<SCK[2:0]>.

(1) 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 SCKx pin.

At the beginning of a transfer, the SCKx pin output becomes the "High" 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.

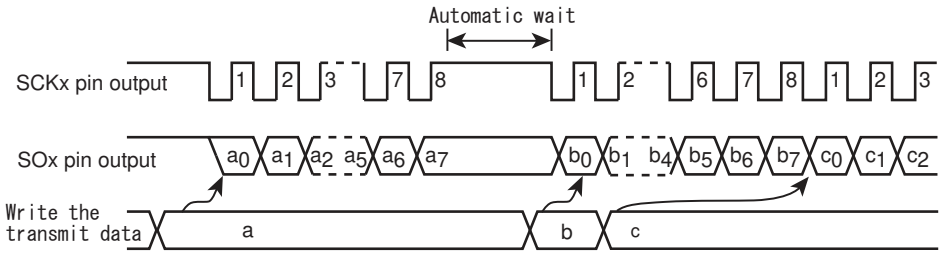
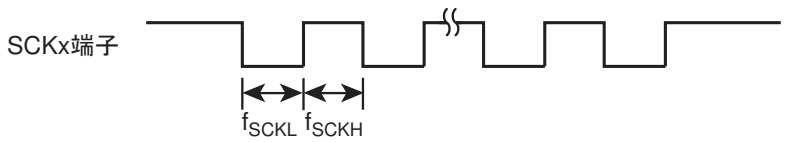


Figure 11-15 Automatic Wait

(2) External clock (<SCK[2:0]> = "111")

The SBI uses an external clock supplied from the outside to the SCKx pin as a serial clock.

For proper shift operations, the serial clock at the "High" and "Low" levels must have the pulse widths as shown below.



f_{SCKL}, f_{SCKH} > 4/fsys

Figure 11-16 Maximum Transfer Frequency of External Clock Input

11.8.1.2 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 SCKx pin input/output).

- Trailing-edge shift

  Data is shifted at the trailing edge of the serial clock (or the rising edge of the SCKx pin input/output).

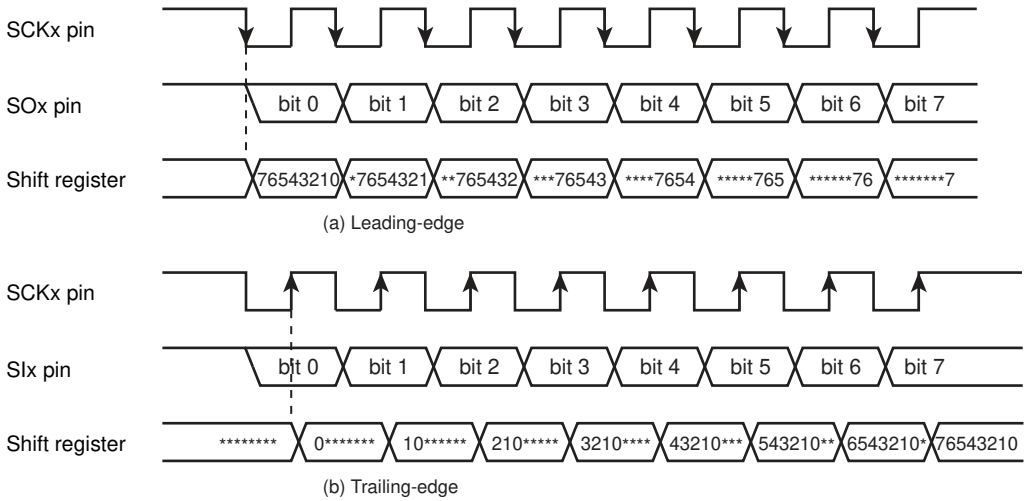


Figure 11-17 Shift Edge

11.8.2 Transfer Modes

The transmit mode, the receive mode or the transmit/receive mode can be selected by programming SBIxCR1<SIOM[1:0]>.

11.8.2.1 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".

When in the external clock mode, <SIOS> must be cleared to "0" before next data shifting. If <SIOS> does not be cleared to "0" before next data shifting, SBI output dummy data and stopped.

		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.

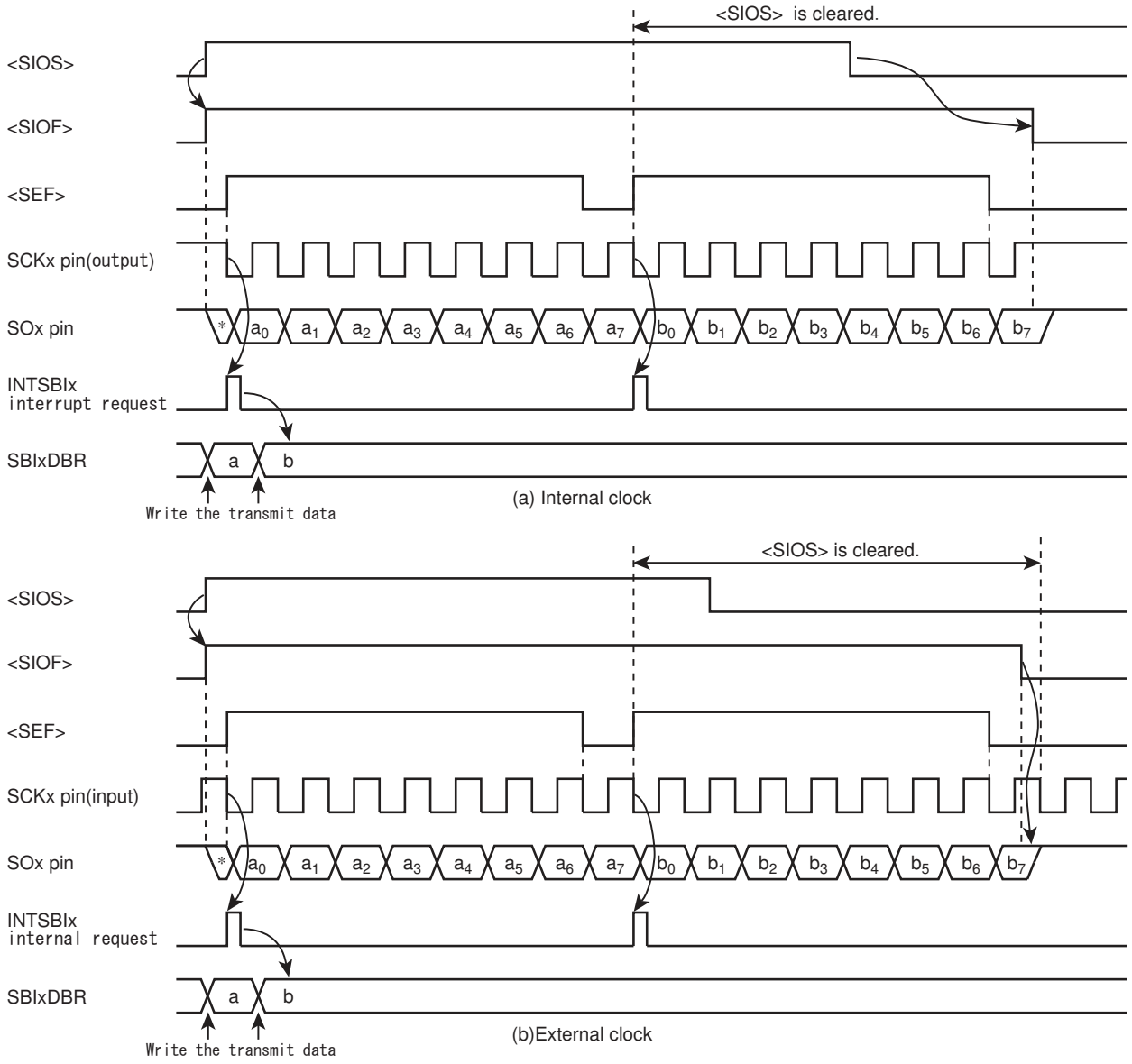


Figure 11-18 Transmit Mode

Example: Example of programming (external clock) to terminate transmission by <SIO>

```
                                7  6  5  4  3  2  1  0
if SBIxSR<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.
```

11.8.2.2 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	X	X	X	Starts reception.

INTSBIx interrupt

Reg.	←	SBIxDBR	Reads the received data.

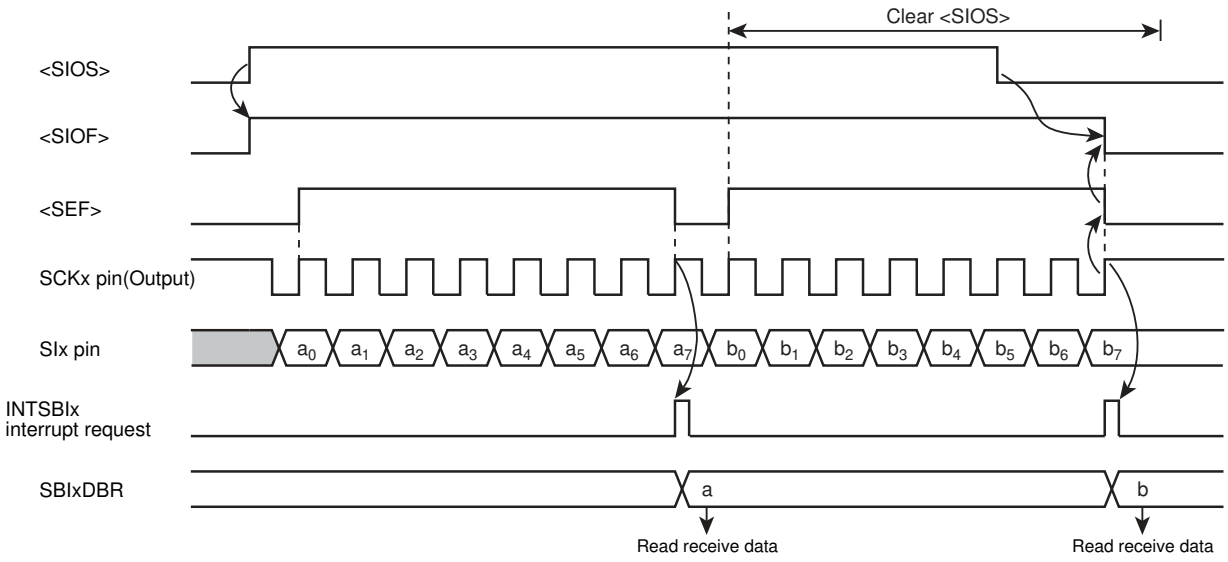


Figure 11-19 Receive Mode (Example: Internal Clock)

11.8.2.3 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 SOx 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 to "1", the transmission and reception is 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.**

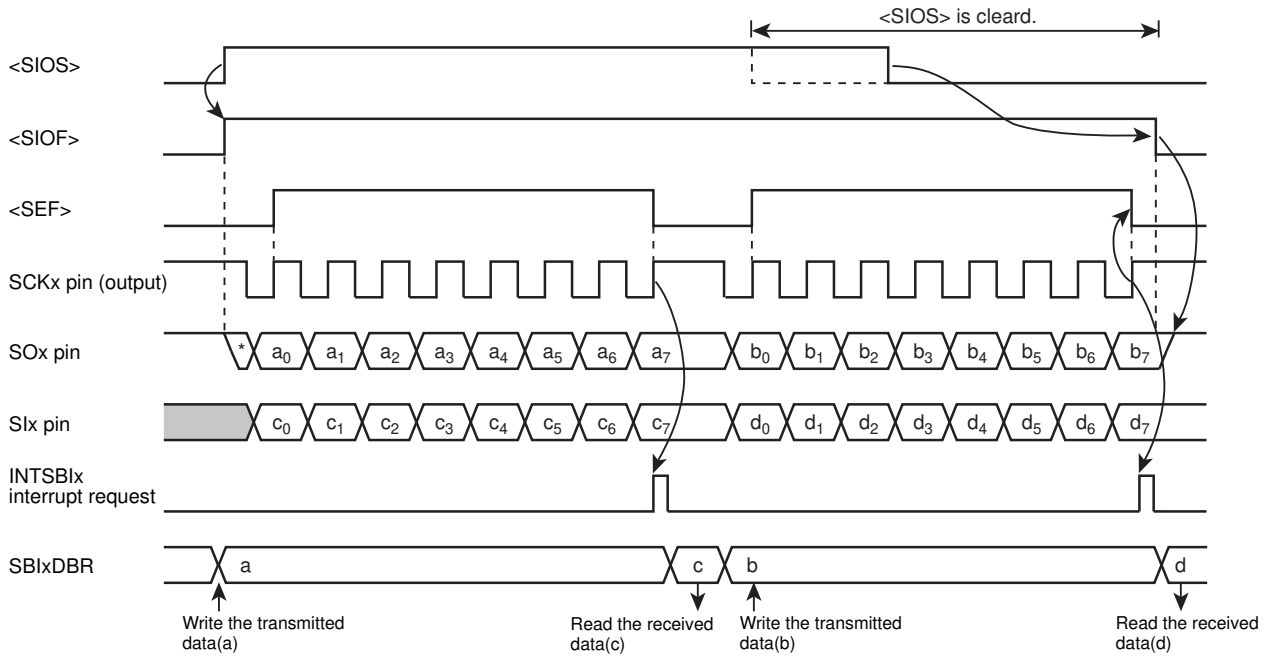


Figure 11-20 Transmit/Receive Mode (Example: Internal Clock)

		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.

11.8.2.4 Data retention time of the last bit at the end of transmission

Under the condition SBIxCR1<SIOS>= "0", the last bit of the transmitted data retains the data of SCK rising edge as shown below. Transmit mode and transmit/receive mode are the same.

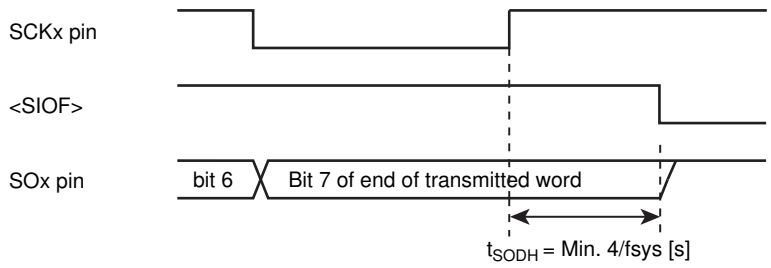


Figure 11-21 Data retention time of the last bit at the end of transmission

12. Consumer Electronics Control (CEC)

12.1 Outline

This IP enables to transmit or receive data that conforms to Consumer Electronics Control (hereafter referred to as CEC) protocol.

This IP can operate conformably to HDMI 1.3a specifications.

12.1.1 Reception

- Clock sampling at 32.768kHz
  - -Adjustable noise canceling time
- Data reception per 1byte
  - -Flexible data sampling point
  - -Data reception is available even when an address discrepancy is detected.
- Error detection
  - -Cycle error (min./max.)
  - -ACK collision
  - -Waveform error

12.1.2 Transmission

- Data transmission per 1byte
  - -Triggered by auto-detection of bus free state
- Flexible waveform
  - -Adjustable rising edge and cycle
- Error detection
  - -Arbitration lost
  - -ACK response error

12.1.3 Precautions

Be careful about the following in the receive operation.

Address condition	Setting of CECRCR1<CECOTH>	Precautions
Logical address match	−	If the initiator sends a new message beginning with the start bit without having sent the last block with EOM="1", a maximum cycle error is determined for the ACK bit and an interrupt is generated. Then, the receive operation is performed in the usual way.
Logical address disacre-pancy	When data reception at logical address discrepancy is enabled. (CECRCR1<CECOTH> = "1")	
	When data reception at logical address discrepancy is disabled. (CECRCR1<CECOTH> = "0")	The initiator must send the last block of data with the EOM bit set to "1". If the last block is sent with EOM="0", the subsequent operation cannot be guaranteed.

12.2 Block Diagram

Figure 12-1 shows the Block Diagram of CEC

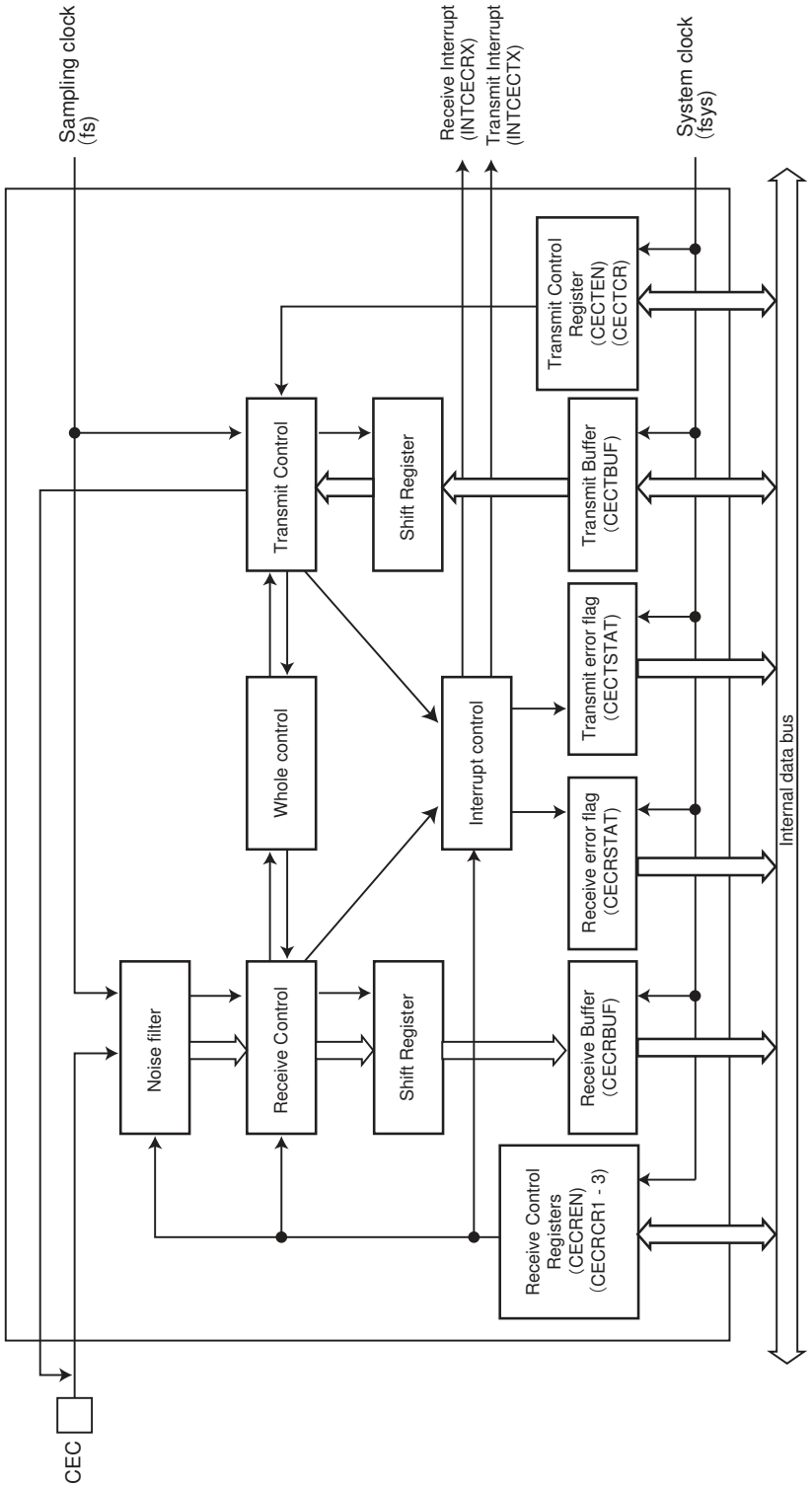


Figure 12-1 Block Diagram of CEC

12.3 Registers

12.3.1 Register List

The control registers and address for CEC are as follows.

Base Address = 0x4004_0300

Registers		Address (Base+)
CEC Enable Register	CECEN	0x0000
Logical Address Register	CECADD	0x0004
Software Reset Register	CECRESET	0x0008
Receive Enable Register	CECREN	0x000C
Receive Buffer Register	CECRBUF	0x0010
Receive Control Register 1	CECRCR1	0x0014
Receive Control Register 2	CECRCR2	0x0018
Receive Control Register 3	CECRCR3	0x001C
Transmit Enable Register	CECTEN	0x0020
Transmit Buffer Register	CECTBUF	0x0024
Transmit Control Register	CECTCR	0x0028
Receive Interrupt Status Register	CECRSTAT	0x002C
Transmit Interrupt Status Register	CECTSTAT	0x0030

12.3.2 CECEN (CEC Enable Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	I2CEC	CECEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	I2CEC	R/W	CEC operation at IDLE. 0 : Disabled 1 : Enabled Controls the CEC operation at the IDLE mode. Set this bit to "1" when using CEC at the IDLE mode. The <I2CEC> and <CECEN> bits can be set simultaneously.
0	CECEN	R/W	CEC operation 0 : Disabled 1 : Enabled Specifies the CEC operation. Enable CEC before using. When the CEC operation is disabled, no clocks are supplied to the CEC module except for the CECEN register. Thus power consumption can be reduced. When CEC is disabled after it was enabled, each register setting is maintained.

12.3.3 CECADD (Logical Address Register)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-0	CECADD[15:0]	R/W	Logical address 15 to 0 Specifies the logical address assigned to CEC. Multiple addresses can be set simultaneously since each bit corresponds with each address.

Note: A broadcast message is received regardless of the register setting. By allocating a logical address of a device to 15, logical "0" is sent as an ACK response to the broadcast message.

12.3.4 CECRESET (Software Reset Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	CECRESET
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	FUnction
31-1	−	R	Read as 0
0	CECRESET	W	Software reset 0: Disabled 1: Enabled Stops all the CEC operation and initializes the register. Setting this bit to "1" affects as follows: Reception: Stops immediately. The received data is discarded. Transmission (including the CEC line): Stops immediately. Register: All the registers other than CECEN are initialized. Read as 0.

12.3.5 CECREN (Receive Enable Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	CECREN
After reset	0	0	0	0	0	0	0	Undefined

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0
0	CECREN	R/W	Reception control [Write] 0 : Disabled 1 : Enabled [Read] 0 : Stopped 1 : In operation Controls the reception operation of CEC. Writing "0" or "1" to this bit enables or disables data reception. This bit becomes ready for data reception by writing "1". The state of the reception circuit is monitored by reading this bit. It enables you to check if what you set has properly been reflected.

Note 1: Enable the <CECREN> bit after setting the CECRCR1, CECRCR2 and CECRCR3.

Note 2: It takes a little time to reflect the setting of the <CECREN> bit to the circuit. Stop transmission and reception before changing the settings or enabling the transmission and reception.

12.3.6 CECRBUF (Receive Buffer Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	CECACK	CECEOM
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CECRBUF							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-10	−	R	Read as 0.
9	CECACK	R	ACK bit Reads the received ACK bit.
8	CECEOM	R	EOM bit Reads the received EOM bit.
7-0	CECRBUF[7:0]	R	Received data Reads one byte of data received. The bit 7 is the MSB.

Note 1: Writing to this register is ignored.

Note 2: Read this register as soon as a receive interrupt is generated. The subsequent reading data may not be ensured.

12.3.7 CECRCR1 (Receive Control Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	CECACKDIS
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	CECHNC		-	CECLNC		
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	CECMIN			-	CECMAX		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	CECDAT			CECTOUT		CECRIHLD	CECOTH
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-25	−	R	Read as 0.
24	CECACKDIS	R/W	Logical "0" as ACK response 0: send 1: not send Specifies if logical "0" is sent or not as an ACK response to the data block when destination address corresponds with the address set in the logical address register. (The header block sends logical "0" as an ACK response regardless of the bit setting when detecting the addresses corresponding)
23-22	−	R	Read as 0.
21-20	CECHNC[1:0]	R/W	The number of "High" samplings for noise cancellation. 00: None (one time of fs clock observed.) 01: 1/fs (two consecutive fs clocks observed.) 10: 2/fs (three consecutive fs clocks observed.) 11: 3/fs (four consecutive fs clocks observed.) Specifies the time of the noise cancellation for each 1/fs when detecting "High". It is considered as noise if "High"s of the same number as the specified cycles are not sampled.
19	−	R	Read as 0.
18-16	CECLNC[2:0]	R/W	The number of "Low" samplings for noise cancellation. 000: None (one time of fs clock observed.) 100: − (Reserved) 001: 1/fs (two consecutive fs clocks observed) 101: − (Reserved) 010: 2/fs (three consecutive fs clocks observed) 110: − (Reserved) 011: 3/fs (four consecutive fs clocks observed.) 111: − (Reserved) Specifies the time of the noise cancellation for each 1/fs when detecting "Low". It is considered as noise if "Low"s of the same number as the specified cycles are not sampled.
15	−	R	Read as 0.
14-12	CECMIN[2:0]	R/W	Time to identify as minimum cycle error 000: 67/fs (approx.2.045ms) 100: 67/fs − 1/fs 001: 67/fs + 1/fs 101: 67/fs − 2/fs 010: 67/fs + 2/fs 110: 67/fs − 3/fs 011: 67/fs + 3/fs 111: 67/fs − 4/fs Specifies the minimum time to identify a valid bit. Base time is 67/fs (approx.2.045) ms. Enables to specify it between the ranges −4/fs to +3/fs by the unit of 1/fs. An interrupt is generated and "Low" is output to CEC for approx. 3.63 ms when one bit cycle is shorter than the specified time.
11	−	R	Read as 0.
10-8	CECMAX[2:0]	R/W	Time to identify as maximum cycle error 000: 90/fs (approx. 2.747ms) 100: 90/fs − 1/fs 001: 90/fs + 1/fs 101: 90/fs − 2/fs 010: 90/fs + 2/fs 110: 90/fs − 3/fs 011: 90/fs + 3/fs 111: 90/fs − 4/fs

Bit	Bit Symbol	Type	Function
			Specifies the maximum time to identify as a valid bit. Base time is 90/fs (approx.2.747 ms). Enables to specify it between the ranges −4/fs to +3/fs by the unit of 1/fs. An interrupt is generated when one bit cycle is longer than the specified time.
7	−	R	Read as 0.
6-4	CECDAT[2:0]	R/W	Point of determining the data as 0 or 1. 000: 34/fs (approx. 1.038ms) 100: 34/fs − 2/fs 001: 34/fs + 2/fs 101: 34/fs − 4/fs 010: 34/fs + 4/fs 110: 34/fs − 6/fs 011: 34/fs + 6/fs 111: Reserved Specifies the point of determining the data as logical "0" or logical "1". Base time is 34/fs (approx.1.038 ms). Enables to specify it within ±6/fs by the unit of 2/fs.
3-2	CECTOUT[1:0]	R/W	Cycle to identify timeout 00: 1 bit cycle 01: 2 bit cycle 10: 3 bit cycle 11: Reserved Specifies the time to determine a timeout. Enables to specify it between 1 bit and 3 bits for each bit cycle. This setting is used to detect a timeout occurs when the <CECRIHLD> bit is valid.
1	CECRIHLD	R/W	Error interrupt suspend 0: Not suspended 1: Suspended Specifies if a receive error interrupt (maximum cycle error, buffer overrun and waveform error) is suspended or not. Setting "1" generates no interrupt at the error detection. If data continues to an ACK bit, an ACK response is executed by a reversed logic. If the subsequent bits are interrupted, it is determined as a timeout, based on the setting in <CECTOUT>. After the ACK response or the timeout determination, an interrupt is generated.
0	CECOTH	R/W	Data reception at logical address discrepancy 0: Not received 1: Received Specifies if data is received or not when destination address does not correspond with the address set in the CECADD register.

Note 1: The settings in <CECHNC>, <CECLNC> and <CECDAT> are also used in receiving an ACK response at transmission.

Note 2: Changing the configurations during transmission or reception may harm its proper operation. Before the change, set the CECREN <CECREN> bit to disable the reception and read the <CECREN> bit and the CECTEN <CECTRANS> bit to ensure that the operation is stopped.

Note 3: A broadcast message is received regardless of the <CECOTH> register setting.

Note 4: <CECLNC> must be used under the same setting as CECTCR<CECDTRS>.

12.3.8 CECRCR2 (Receive Control Register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	CECSWAV3			-	CECSWAV2		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	CECSWAV1			-	CECSWAV0		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-15	−	R	Read as 0.
14-12	CECSWAV3[2:0]	R/W	Max. cycle to detect start bit. 000: 154/fs (approx. 4.700 ms) 001: 154/fs + 1/fs 001: 154/fs + 2/fs 011: 154/fs + 3/fs 100: 154/fs + 4/fs 101: 154/fs + 5/fs 110: 154/fs + 6/fs 111: 154/fs + 7/fs
11	−	R	Read as 0.
10-8	CECSWAV2[2:0]	R/W	Min. cycle to detect start bit. 000: 141/fs (approx. 4.303 ms) 001: 141/fs − 1/fs 001: 141/fs − 2/fs 011: 141/fs − 3/fs 100: 141/fs − 4/fs 101: 141/fs − 5/fs 110: 141/fs − 6/fs 111: 141/fs − 7/fs
7	−	R	Read as 0.
6-4	CECSWAV1[2:0]	R/W	Max. time of start bit rising timing. 000: 128/fs (approx. 3.906 ms) 001: 128/fs + 1/fs 001: 128/fs + 2/fs 011: 128/fs + 3/fs 100: 128/fs + 4/fs 101: 128/fs + 5/fs 110: 128/fs + 6/fs 111: 128/fs + 7/fs
3	−	R	Read as 0.
2-0	CECSWAV0[2:0]	R/W	Min. time of start bit rising timing. 000: 115/fs (approx. 3.510 ms) 001: 115/fs − 1/fs 001: 115/fs − 2/fs 011: 115/fs − 3/fs 100: 115/fs − 4/fs 101: 115/fs − 5/fs 110: 115/fs − 6/fs 111: 115/fs − 7/fs

<CECSWAV3>:/ Specifies the cycles to detect a start bit.

<CECSWAV2>: <CECSWAV3> is for the maximum cycles. Base time is 154/fs (approx.4.700 ms). Enables to specify it between the ranges 0 to +7/fs by the unit of 1/fs.

<CECSWAV2> is for the minimum cycles. Base time is 141/fs (approx.4.303 ms). Enables to specify it between the ranges 0 to -7/fs by the unit of 1/fs.

<CECSWAV1>:/ Specifies the rising timing of a start bit in its detection.

<CECSWAV0>: <CECSWAV1> is for the maximum time of the rising timing.Base time is 128/fs (approx.3.906 ms). Enables to specify it between the ranges 0 to +7/fs by the unit of 1/fs.

<CECSWAV0> is for the minimum time of the rising timing. Base time is 115/fs (approx.3.510 ms). Enables to specify it between the ranges 0 to -7/fs by the unit of 1/fs.

Note: Changing the configurations during reception may harm its proper operation. Before the change, set CECREN <CECREN> to disable the reception and read the <CECREN> bit to ensure that the operation is stopped.

12.3.9 CECRCR3 (Receive Control Register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	CECWAV3			-	CECWAV2		
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	CECWAV1			-	CECWAV0		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	CECWAVEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-23	−	R	Read as 0.
22-20	CECWAV3[2:0]	R/W	The latest rising timing of logical "0" determined as proper waveform. 000: 56/fs (approx. 1.709 ms) 100: 56/fs + 4/fs 001: 56/fs + 1/fs 101: 56/fs + 5/fs 001: 56/fs + 2/fs 110: 56/fs + 6/fs 011: 56/fs + 3/fs 111: 56/fs + 7/fs
19	−	R	Read as 0.
18-16	CECWAV2[2:0]	R/W	The fastest rising timing of logical "0" deterrmined as proper. 000: 43/fs (approx.1.312 ms) 100: 43/fs − 4/fs 001: 43/fs − 1/fs 101: 43/fs − 5/fs 010: 43/fs − 2/fs 110: 43/fs − 6/fs 011: 43/fs − 3/fs 111: 43/fs − 7/fs
15	−	R	Read as 0.
14-12	CECWAV1[2:0]	R/W	The latest rising timing of logical "1" determined as proper waveform. 000: 26/fs (approx. 0.793 ms) 100: 26/fs + 4/fs 001: 26/fs + 1/fs 101: 26/fs + 5/fs 001: 26/fs + 2/fs 110: 26/fs + 6/fs 011: 26/fs + 3/fs 111: 26/fs + 7/fs
11	−	R	Read as 0
10-8	CECWAV0[2:0]	R/W	The fastest rising timing of logical "1" determined as proper. 000: 13/fs (approx. 0.396 ms) 100: 13/fs − 4/fs 001: 13/fs − 1/fs 101: 13/fs − 5/fs 010: 13/fs − 2/fs 110: 13/fs − 6/fs 011: 13/fs − 3/fs 111: 13/fs − 7/fs
7-1	−	R	Read as 0.
0	CECWAVEN	R/W	Waveform error detection 1: Enable 0: Disable Detects a received waveform does not identical to the one defined and generates waveform error interrupt. If enabled, an error is detected according to the setting of <CECWAV0> <CECWAV1> <CECWAV2> <CEC WAV3>.

<CECWAV3>: This setting is enabled when the <CECWAVEN> bit is set to "1".
By setting these bits, an error is detected if rising edge of the received waveform comes later than that of proper logical "0".
Base time is 56/fs (approx. 1.709ms). Enables to specify it between the ranges 0 to +7/fs by the unit of 1/fs.
The received waveform is considered to be an error if a rising edge is not detected from the start point of the bit to the value specified in <CECWAV3>.

<CECWAV2>/ This setting is enabled when the <CECWAVEN> bit is set to "1".

<CECWAV1>: By setting these bits, an error is detected if rising edge of the received waveform comes faster than logical "0" and later than that of proper logical "1".
Base time for <CECWAV1> bit is 26/fs (approx. 0.793ms). Enables to specify it between the ranges 0 to +7/fs by the unit of 1/fs.
Base time for <CECWAV2> bit is 43/fs (approx.1.312ms). Enables to specify it between the ranges 0 to −7/fs by the unit of 1/fs.
If a rising edge is detected during <CECWAV2> bit and <CECWAV1> bit setting, an error occurs.

<CECWAV0>: This setting is enabled when the <CECWAVEN> bit is set to "1".
By setting these bits, an error is detected if rising edge of the received waveform comes faster than that of proper logical "1".
Base time is 13/fs (approx. 0.396ms). Enables to specify it between the ranges 0 to −7/fs by the unit of 1/fs.
The received waveform is considered to be an error if a rising edge is not detected from a start point of the bit to the value specified in <CECWAV0>.

Note: Changing the configurations during reception may harm its proper operation. Before the change, set CECREN <CECREN> to disable the reception and read the <CECREN> bit to ensure that the operation is stopped.

12.3.10 CECTEN (Transmit Enable Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	CECTRANS	CECTEN
After reset	0	0	0	0	0	0	0	Undefined

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	CECTRANS	R	Transmission state 0: not in progress 1:in progress Indicates whether the transmission is in progress or not. It indicates "1" upon starting the transmission of the start bit. It indicates "0" if transmission is completed or an interrupt is generated. Writing to this bit is ignored.
0	CECTEN	W	Transmission control 0: Disable 1: Enable Controls the CEC transmission. Writing this bit enables or disables the transmission. Writing "1" to this bit initiates the transmission. This bit is automatically cleared by a transmit completion interrupt or an error interrupt.

Note 1: Set <CECTEN> after setting the CECTBUF and CECTCR register.

Note 2: Stop transmission and reception before changing the settings or enabling the transmission and reception.

12.3.11 CECTBUF (Transmit Buffer Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	CECTEOM
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CECTBUF							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-9	−	R	Read as 0.
8	CECTEOM	R/W	EOM bit Specifies the EOM bit to transmit.
7-0	CECTBUF[7:0]	R/W	Transmitted data Specifies a byte of data to transmit. The bit 7 is the MSB.

12.3.12 CECTCR (Transmit Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	CECSTRS			-	CECSPRD		
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	CECDTRS			CECDPRD			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	CECBRD	CECFREE			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-23	−	R	Read as 0.
22-20	CECSTRS[2:0]	R/W	Rising timing of start bit. 000: Base time 001: Base time− 1/fs 010: Base time− 2/fs 011: Base time− 3/fs 100: Base time− 4/fs 101: Base time− 5/fs 110: Base time− 6/fs 111: Base time− 7/fs Specifies the rising timing of a start bit. Base time is 121/fs (approx. 3.693 ms). Enables to specify it between the ranges 0 to −7/fs by the unit of 1/fs.
19	−	R	Read as 0.
18-16	CECSPRD[2:0]	R/W	Start bit cycle 000: Base time 001: Base time− 1/fs 010: Base time− 2/fs 011: Base time− 3/fs 100: Base time− 4/fs 101: Base time− 5/fs 110: Base time− 6/fs 111: Base time− 7/fs Specifies a cycle of a start bit. Base time is 147/fs (approx. 4.486 ms). Enables to specify it between the ranges 0 to −7/fs by the unit of 1/fs.
15	−	R	Read as 0.
14-12	CECDTRS[2:0]	R/W	Rising timing of data bit. 000: Base time 001: Base time− 1/fs 010: Base time− 2/fs 011: Base time− 3/fs 100: Reserved 101: Reserved 110: Reserved 111: Reserved Specifies the rising timing of a data bit Base time is 20/fs (approx. 0.610 ms, when logical "1") or 49/fs (approx. 1.495 ms, when logical "0"). Enables to specify it between the ranges 0 to −3/fs by the unit of 1/fs.
11-8	CECDPRD[2:0]	R/W	Data bit cycle 0000: Base time 0001: Base time− 1/fs 0010: Base time− 2/fs 0011: Base time− 3/fs 0100: Base time− 4/fs 0101: Base time− 5/fs 0110: Base time− 6/fs 0111: Base time− 7/fs 1000: Base time− 8/fs 1001: Base time− 9/fs 1010: Base time− 10/fs 1011: Base time− 11/fs 1100: Base time− 12/fs 1101: Base time− 13/fs 1110: Base time− 14/fs 1111: Base time− 15/fs Specifies a cycle of a data bit. Base time is 79/fs (approx. 2.411 ms). Enables to specify it between the ranges 0 to −15/fs by the unit of 1/fs.
7-5	−	R	Read as 0.

Bit	Bit Symbol	Type	Function
4	CECBRD	R/W	Broadcast transmission 0: Not broadcast transmission 1: Broadcast transmission Set this bit to "1" when transmitting a broadcast message.
3-0	CECFREE[3:0]	R/W	Time of bus to be free 0000: 1bit cycle 0001: 2bit cycle 0010: 3bit cycle 0011: 4bit cycle 0100: 5bit cycle 0101: 6bit cycle 0110: 7bit cycle 0111: 8bit cycle 1000: 9bit cycle 1001: 10bit cycle 1010: 11bit cycle 1011: 12bit cycle 1100: 13bit cycle 1101: 14bit cycle 1110: 15bit cycle 1111: 16bit cycle Specifies time of a bus to be free that checked before transmission. Start transmission after checking the CEC line kept inactive during the specified cycles.

Note: <CECDTRS> must be used under the same setting as CECRCR1<CECLNC>.

12.3.13 CECRSTAT (Receive Interrupt Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	CECRIWAV	CECRIOR	CECRIACK	CECRIMIN	CECRIMAX	CECRISTA	CECRIEND
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	CECRIWAV	R	Interrupt flag 0: No wave form error 1: Wave form error Indicates that waveform error is detected. The error occurs when waveform error detection is enabled in CECRCR3 <CECWAVEN>.
5	CECRIOR	R	Interrupt flag 0: No receive buffer overrun 1:Receive buffer overrun Indicates the receive buffer receives next data before reading the data that had already been set.
4	CECRIACK	R	Interrupt flag 0: No ACK collision 1: ACK collision Indicates "0" is detected after the specified time to output ACK bit "0".
3	CECRIMIN	R	Interrupt flag 0: No minimum cycle error 1:Minimum cycle error Indicates one bit cycle is shorter than the minimum cycle error detection time specified in CECRCR1<CECMIN>.
2	CECRIMAX	R	Interrupt flag 0: No maximum cycle error 1: Maximum cycle error Indicates one bit cycle is longer than the maximum cycle error detection time specified in CECRCR1<CECMAX>.
1	CECRISTA	R	Interrupt flag 0: No start bit detection 1: Start bit detection Indicates a start bit is detected.
0	CECRIEND	R	Interrupt flag 0: Not one byte data reception completed 1: Completion of 1 byte data reception Indicates 1 byte of data reception is completed.

Note: Writing to this bit is ignored.

12.3.14 CECTSTAT (Transmit Interrupt Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	CECTIUR	CECTIACK	CECTIAL	CECTIEND	CECTISTA
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4	CECTIUR	R	Interrupt flag 0: No transmit buffer underrun 1: Transmit buffer underrun Indicates next data has not set to the transmission buffer within a byte of data transmission.
3	CECTIACK	R	Interrupt flag 0: No ACK error detection 1: ACK error detection Indicates one of the following conditions occurs. ・When logical "0" is not detected in transmission to the specific address. ・When logical "1" is not detected in transmission of a broadcast message.
2	CECTIAL	R	Interrupt flag 0: No arbitration lost 1: Arbitration lost occurs Indicates "Low" is detected while outputting "High".
1	CECTIEND	R	Interrupt flag 0: No data transmission completion 1: data transmission is completed Indicates data transmission including the EOM bit is completed.
0	CECTISTA	R	Interrupt flag 0: No start transmission 1: Start transmission Indicates 1 byte of data transmission is started.

Note: Writing to this bit is ignored.

12.4 Operations

12.4.1 Sampling clock

CEC lines are sampled by a 32.768kHz of low speed clock (fs).

12.4.2 Reception

12.4.2.1 Basic Operation

If a start bit is detected, a start bit interruption generates. By generating start bit interruption, CECRSTAT<CECRISTA> is set.

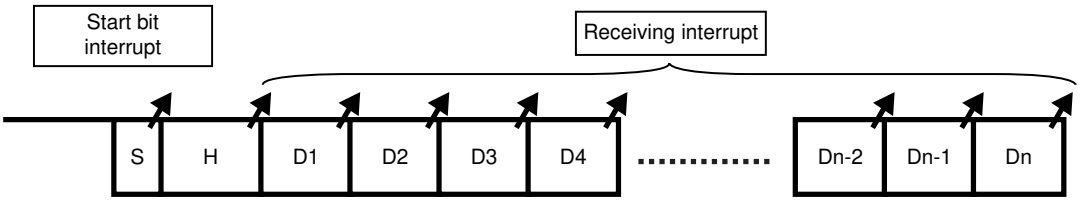
If one byte data, EOM bit and ACK bit are received, the received data is stored in CECRBUF register, and a received interruption generates. By generating the received interruption, CECRSTAT<CECRIEND> is set.

In the CECRBUF register, 8 bit data, EOM bit and ACK bit are stored. The ACK bit is not generated in the CEC circuit internally. This bit is generated from a observation of CEC signal same as other data.

After one data block is received, receiving operation continues until detecting the last block of data with EOM bit set to"1". Detecting the end of last block, CEC becomes the start bit waiting mode.

Detecting an error during data reception causes an error interrupt, and CEC waits for the next start bit. The received data is discarded.

Note: Be careful about the precautions of chapter 12.1.3 in the receive operation.



12.4.2.2 Preconfiguration

Before receiving data, reception settings to the Logical Address Register <CECADD>, the Receive Control Register 1 <CECRCR1>, the Receive Control Register 2 <CECRCR2> and the Receive Control Register 3 <CECRCR3> are required.

(1) Logical Address Configuration

Configure logical address assigned to this product to the CECADD register. Multiple addresses can be set simultaneously since every bit in this register corresponds with each address.

Note: A broadcast message is received regardless of the CECADD register setting. By allocating a logical address of a device to 15, logical "0" is sent as an ACK response to the broadcast message.

(2) Noise Cancellation Time

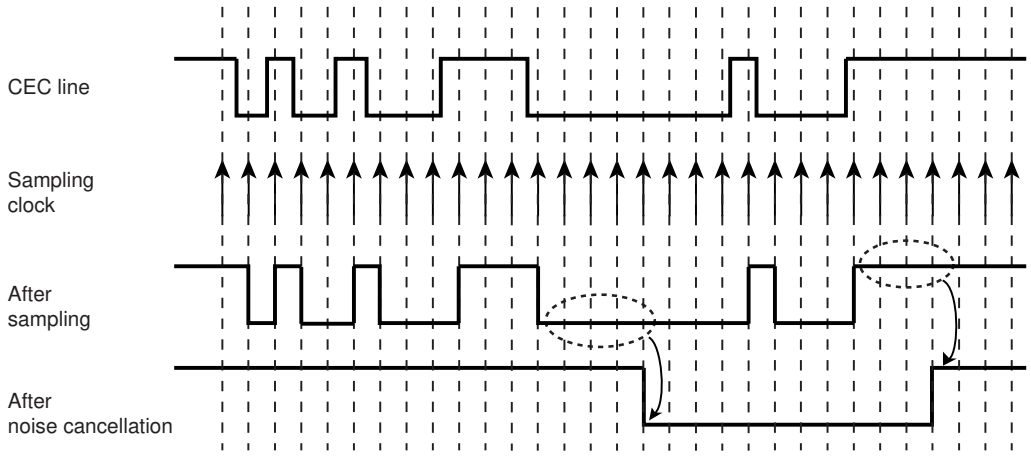
The noise cancellation time is configurable with the <CECHNC> and <CECLNC> bits of the CECRCR1 register. It is considered as noise if "High"or "Low"of the same number as the specified value are not sampled.You can configure the time to detect "High" and "Low" respectively.

A CEC line is monitored at each rising edge of a sampling clock. In the case that the CEC line is changed from "High" to "Low", the change is fully recognized if "Low"s of the same number as specified in the <CECLNC> bit are monitored. In the case that the CEC line is changed from "Low" to "High", the change is fully recognized if "High" of the same number as specified in the <CECHNC> bit are sampled.

Note: <CECLNC> must be used under the same setting as CECTCR<CECDTRS>.

The following illustrates the operation of a case that a noise cancelling is configured as <CECHNC [1:0]> = "10" (3 samplings) and <CECLNC[2:0]> = "011" (4 samplings). By cancelling the noise, a signal "1" shifts to "0" after "0" is sampled four times. The signal "0" shifts to "1" after "1" is sampled three times.

<CECHNC[1:0]> = 10 (3 samplings)
<CECLNC[2:0]> = 011 (4 samplings)



(3) Cycle error

Configure CECRCR1<CECMIN><CECMAX> bits to detect a cycle error.

A cycle error can be detected from each sampling clock cycle between the ranges −4/fs to +3/fs by the unit of 1/fs from the minimum value (67/fs, approx. 2.045ms) or the maximum value (90/fs approx. 2.747ms).

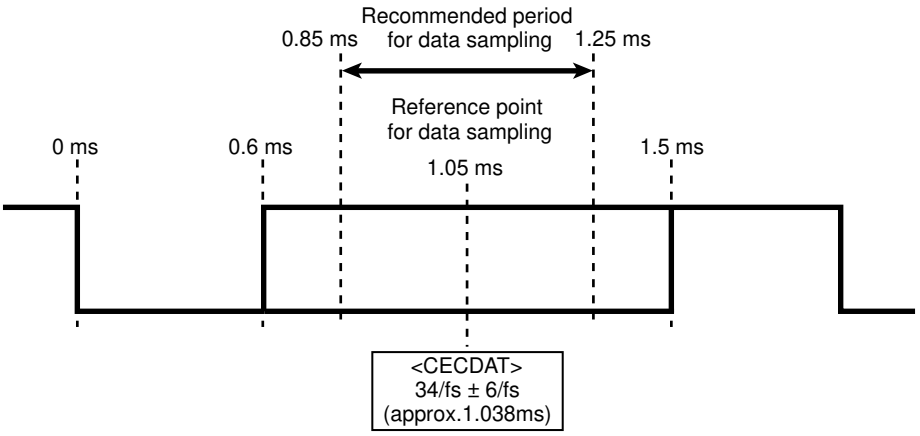
Detecting an error during data reception causes an error interrupt, and CEC waits for the next start bit. The received data is discarded.

(4) Point of Determining Data

Configure the CECRCR1 <CECDAT> bit for the point of determining the data as "0" or "1".

Base time is 34/fs (approx.1.038ms) from the start point and also configurable ±6/fs by the unit of 2/fs.

Data sampring timing that specification recommends



(5) ACK Response

Configuring the CECRCR1 <CECACKDIS> bit enables you to specify if logical "0" is sent or not as an ACK response to the data block when destination address corresponds with the address set in the logical address register.

The header block sends logical "0" as an ACK response regardless of the bit setting when detecting the addresses corresponding.

The following lists the ACK responses.

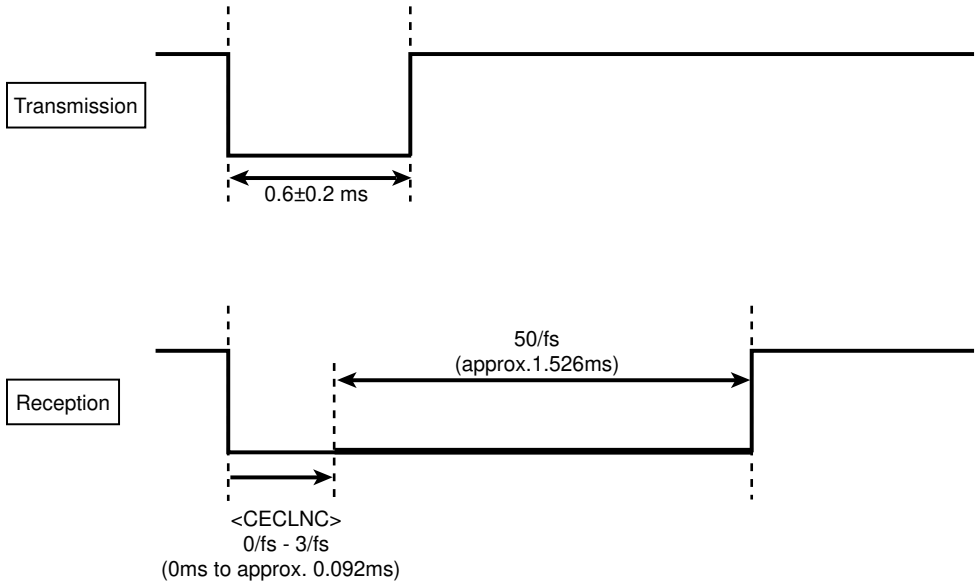
"Yes" indicates that CEC outputs "0" as a response to the ACK signal from a transmission device (ACK bit: logical "0"). "No" indicates that CEC does not output "0" as a response to the ACK signal from a transmission device (ACK bit: logical "1").

Register setting		Header block address		Data block address	
		Conformity	Discrepancy	Conformity	Discrepancy
CECRCR1 <CECACKDIS>	"0" (responding logical "0")	Yes	No	Yes	No
	"1" (not responding logical "0")			No	No

The following describes the ACK response timing.

When the falling edge of the ACK bit from the initiator is detected, this IP outputs "Low" for approximately 1.526 ms. The start time of outputting "Low" is specified with CECRCR1<CECLNC> bit that sets the noise cancelling time.

Note: <CECLNC> must be used under the same setting as CECTCR<CECDTRS>.



(6) Receive Error Interrupt Suspend

Configure the CECRCR1 <CECRIHLD> bit to specify if a receive error interrupt (maximum cycle error, buffer overrun and waveform error) is suspended or not. Setting "1" generates no interrupt at the error detection.

If data continues to the ACK bit, an ACK response is executed by a reversed logic. If the subsequent bits are interrupted, it is determined as a timeout, based on the setting in <CECTOUT> of the CECRCR1 register.

After the ACK response or the timeout determination, an interrupt is generated.

(7) Cycles to Identify Timeout

Configure the CECRCR1<CECTOUT> bit to specify the time to determine a timeout.

This is used when the setting of a receive error interrupt suspension, which is specified in CECRCR1 <CECRIHLD>, is valid.

(8) Data Reception at Logical Address Discrepancy

By setting CECRCR1 <CECOTH>, you can specify if data is received or not when destination address does not correspond with the address set in the CECADD register.

In this case, data is received as usual, and an interrupt is generated by detecting an error. However, an ACK response of neither the header block nor the data block is sent.

Note 1: A broadcast message is received regardless of the <CECOTH> register setting.

Note 2: If the initiator sends a new message beginning with the start bit without having sent the last block with EOM="1", a maximum cycle error is determined for the ACK bit and an interrupt is generated. Then, the receive operation is performed in the usual way.

(9) Start Bit Detection

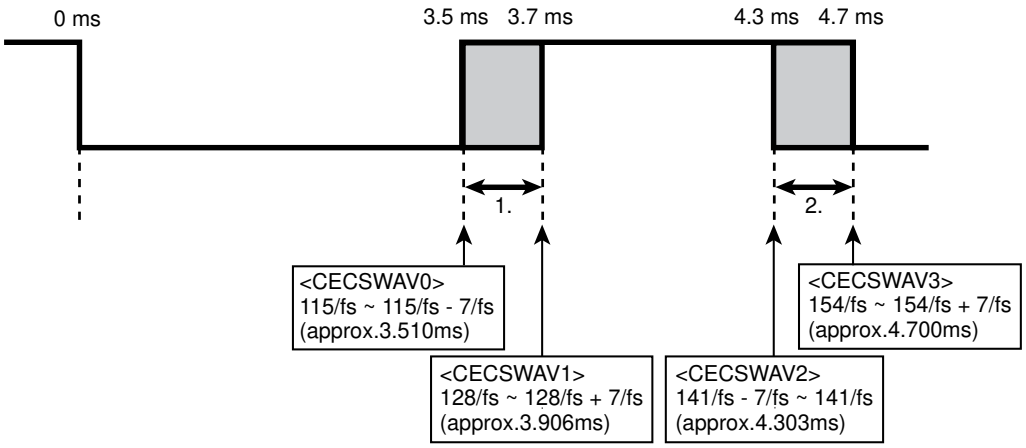
Configuring the CECRCR2 register allows you to specify the rising timing and a cycle of the start bit detection respectively.

<CECSWAV0> is to specify the fastest start bit rising timing. <CECSWAV1> is to specify the latest start bit rising timing (1. in the figure shown below).

<CECSWAV2> is to specify the minimum cycle of a start bit. <CECSWAV3> is to specify the maximum cycle of a start bit (2. in the figure shown below).

If a rising edge during the period 1. and a falling edge during the period 2. are detected, the start bit is considered to be valid.

Permissible value of signal transition timing on specification (Start bit)



(10) Waveform Error Detection

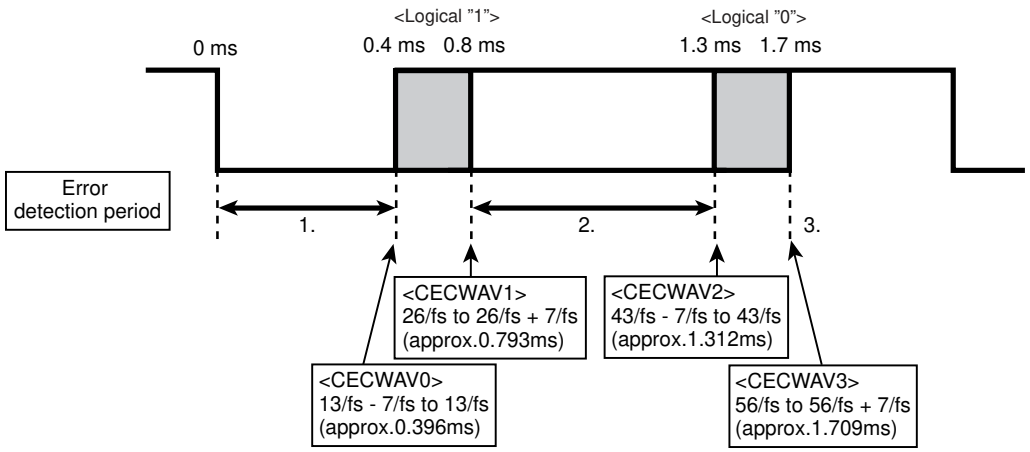
To detect an error when a received waveform is out of the defined tolerance range, configure the CECRCR3 register.

An error is detected when the <CECWAVEN> bit of the CECRCR3 register is enabled. You can specify the detection time in the <CECWAV0> <CECWAV1> <CECWAV2> <CECWAV3> bits.

If the rising edge is detected during the period 1. or 2. shown below, or not detected in the timing described in 3., a waveform error interrupt is generated.

1. A period between the beginning of a bit and the fastest logical "1" rising timing
2. A period between the latest logical "1" rising timing and the fastest logical "0" rising timing.
3. The latest logical "0" rising timing.

Permissible value of signal transition timing on specification (Data bit)



12.4.2.3 Enabling Reception

After configuring the CECADD, CECRCR1, CECRCR2 and CECRCR3 registers, CEC is ready for reception by enabling the CECREN <CECREN> bit. Detecting a start bit initiates the reception.

Note: Changing the configurations of the CECADD, CECRCR1, CECRCR2 and CECRCR3 registers during reception may harm its proper operation. Before the change of the registers shown below, set the CECREN <CECREN> bit to disable the reception and read the <CECREN> bit and the CECTEN <CECTRANS> bit to ensure that the operation is stopped.

Register name	Bit Symbol	Setting item
CECADD	<CECADD[15:0]>	Logical address
CECRCR1	<CECHNC><CECLNC>	Noise cancellation time
	<CECMIN><CECMAX>	Time to identify cycle error
	<CECOTH>	Data reception at logical address discrepancy
CECRCR2	<CECSWAV0><CECSWAV1> <CECSWAV2><CECSWAV3>	Start bit detection
CECRCR3	<CECWAV0><CECWAV1> <CECWAV2><CECWAV3>	Waveform error detection (when enabled)

12.4.2.4 Detecting Error Interrupt

Detecting an error during data reception causes an error interrupt, and CEC waits for the next start bit. The received data is discarded.

It is possible to suspend a receive error interrupt (maximum cycle error, receive buffer overrun and waveform error), continue reception and send the reversed ACK response.

You can check the interrupt factor by monitoring the bit of the CECRSTAT register corresponding to the interrupt.

12.4.2.5 Details of reception error

(1) Cycle error

Period between the falling edges of the two sequential bits is measured during reception. If the period does not comply with the specified minimum or maximum value, a cycle error interrupt is generated.

A setting of maximum cycle and minimum cycle time is specified by CECRCR1<CECMIN> and <CECMAX> bits. Maximum value is 90/fs (approx.2.747ms) and minimum value is 67/fs (approx. 2.045ms). It can be specified between the ranges −4/fs to +3/fs by the unit of 1/fs to detect cycle errors.

The CECRSTAT <CECRIMIN> bit or the <CECRIMAX> bit is set if a cycle error interrupt is generated.

The minimum cycle error causes CEC to output "Low" for approx. 3.63 ms.

Note 1: When minimum cycle error is detected, "Low" is output after "Low" detecting noise cancellation time.

Note 2: If the initiator sends a new message beginning with the start bit without having sent the last block with EOM="1", a maximum cycle error may be determined for the ACK bit. For detail , refer to the Chapter 12.1.3.

(2) ACK Collision

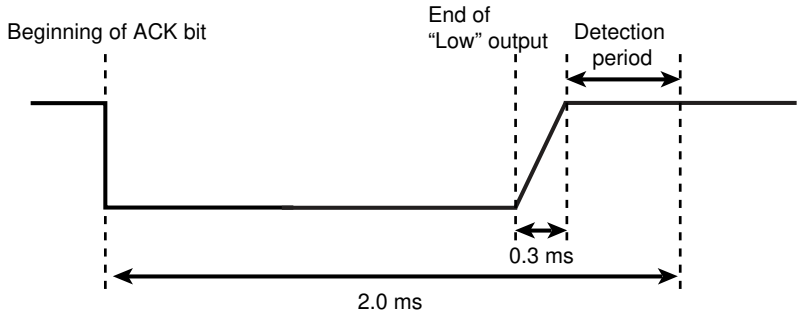
At an ACK response, detecting "Low" after the specified period to output generates an ACK collision interrupt or a minimum cycle error interrupt.

The ACK collision interrupt sets the CECRSTAT <CECRIACK> bit. The minimum cycle error interrupt sets the CECRSTAT <CECRIMIN> bit.

The following describes the period and method of detection.

Detection starts approx. 0.3 ms after the end of the period of outputting "Low" and ends approx 2.0 ms from the starting point (the falling edge) of the ACK bit.

At 0.3 ms from the end of the period of outputting "Low", CEC checks if the CEC line is "0" or not. If it is "Low", an ACK collision interrupt is generated. If it is "High", and "Low" is detected during the detection period, the minimum cycle error interrupt is generated. The minimum cycle error causes CEC to output "Low" for approx. 3.63ms.



(3) Receive Buffer Overrun

A receive buffer overrun interrupt is generated when the next data reception is completed before reading the data stored in the receive buffer.

The interrupt sets the CECRSTAT <CECRIOR> bit.

(4) Waveform Error

A waveform error occurs when waveform error detection is enabled in CECRCR3.

Detecting a waveform, which does not identical to the defined, results in the waveform error. The interrupt is generated.

The interrupt sets the CECRSTAT <CECRIWAV> bit.

(5) Suspending Receive Error Interrupt

You can specify if a maximum cycle error, a buffer overrun and a waveform error are suspended or not without generating an interrupt at error detection. This can be set in the CECRCR1 <CECRIHLD> bit. To enable the setting, a timeout setting with the CECRCR1 <CECTOUT> bit is required.

Under suspend-enable condition, if CEC keeps receiving the next bit and the entire reception including the ACK bit is completed, CEC generates an interrupt after a reversed ACK response is executed. "1" is set to the bits of the CECRSTAT register: the <CECRIEND> bit that indicates the reception completion, and the bits corresponding to the detected errors.

If the reception of the next bit is interrupted, CEC starts to measure the timeout period, and an interrupt is generated after the timeout. "1" is set to the bits of the CECRSTAT register corresponding to the detected error.

The timeout is measured from the end of the last bit received as is the case with wait time of a bus to be free in transmission.

The information that the interrupts are suspended is held until the EOM bit is received or the timeout occurs. Thus, an interrupt is generated in each reception of a byte of data if multiple bytes are received while interrupts are suspended. "1" is set to the bits of the CECRSTAT register: the <CECRIEND> bit that indicates the reception completion, and the bits corresponding to the detected errors. The flags of the suspended interrupts and the reception completion are set to the bits of the CECRSTAT register.

Note 1: A minimum cycle error interrupt is generated upon detecting a minimum cycle error in the next received bit while interrupts are suspended. "Low" is output to CEC for approx. 3.63 ms. The flags of the suspended interrupts and the minimum cycle error are set to the bits of the CECRSTAT register.

Note 2: If an interrupt other than a minimum cycle error interrupt is generated while interrupts are suspended, CEC continues reception until the ACK response or the timeout. All the flags of the detected interrupts are set to the bits of the CECRSTAT register.

12.4.2.6 Stopping Reception

Writing "0" to the CECREN <CECREN> bit disables data reception. The reception is stopped upon disabling the bit during reception. The received data is discarded.

Note: If the reception is disabled while "Low" is sent as a signal of minimum cycle error, the "Low" output is stopped as well.

12.4.3 Transmission

12.4.3.1 Basic Operation

In the transmission setting, CEC firstly confirms bus free wait state to check whether CEC falling edge signals is not existed for specified bit cycles and then sends a start bit. The confirmation of bus free wait is performed all the time. Thus once bus free wait condition is satisfied, a transmission will start soon when transmission setting is done.

After transmitting a start bit, CEC transmits one byte data and EOM data that are stored in the transmit buffer to the shift register. When the transmission of the first bit of the one byte data begins, transmission interrupt is generates, and CECTSTAT<CECTISTA> is set. After transmission interrupt generation, next one byte data is prepared to the transmit data buffer.

One byte data transmission completes in order of transmission of 8 bits data, EOM bit, ACK bit transmission and ACK bit response confirmation.

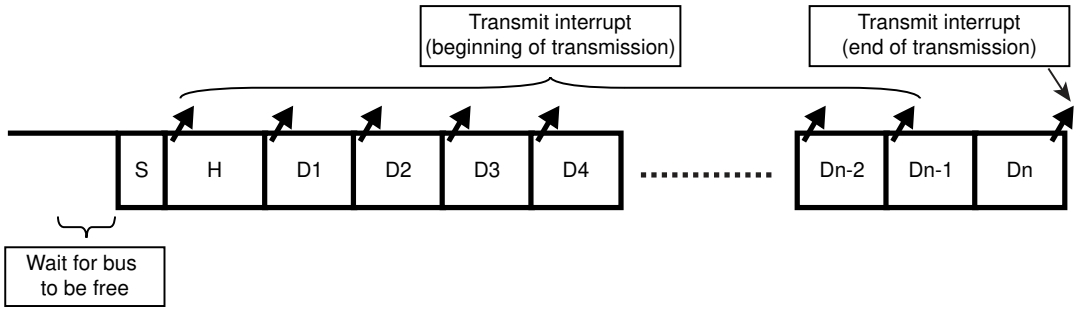
Data transmission continues until EOM is set to "1".

If EOM is set to "1", the end of transmission interrupt generates after confirmation of data, EOM, ACK bit transmission and ACK bit response. By the end of transmission interrupt generates, CECTSTAT<CECTIEND> is set.

Interrupt generation ends a series of transmission process, and CECTEN<CECTEN> is cleared.

If an error is generated during transmission, an error interrupt is generates to stop transmission.

Even if reception is enabled, no reception is executed during transmission.



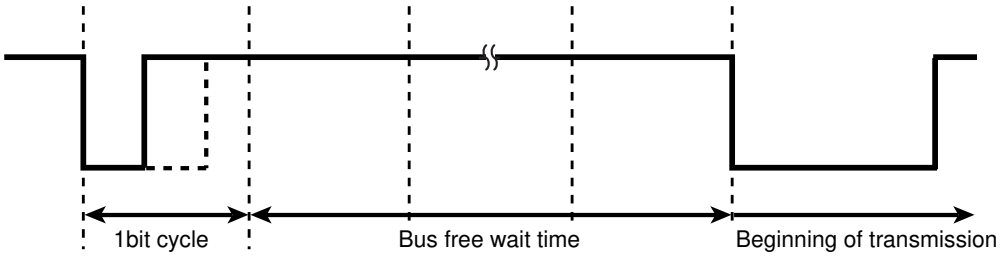
12.4.3.2 Preconfiguration

Before transmitting data, transmission settings to the Transmit Control Register (CECTCR) and the Transmit Buffer Register (CECTBUF) are required.

(1) Bus Free Wait Time

Specify the bus free wait time in the CECTCR<CECFREE> bits. It can be specified in a range of 1 to 16 bit cycles.

Counting of the bus free wait time begins one bit cycle after the falling edge of the final bit. If the signal stays high for the specified number of bit cycles, transmission starts.



(2) Transmitting Broadcast Message

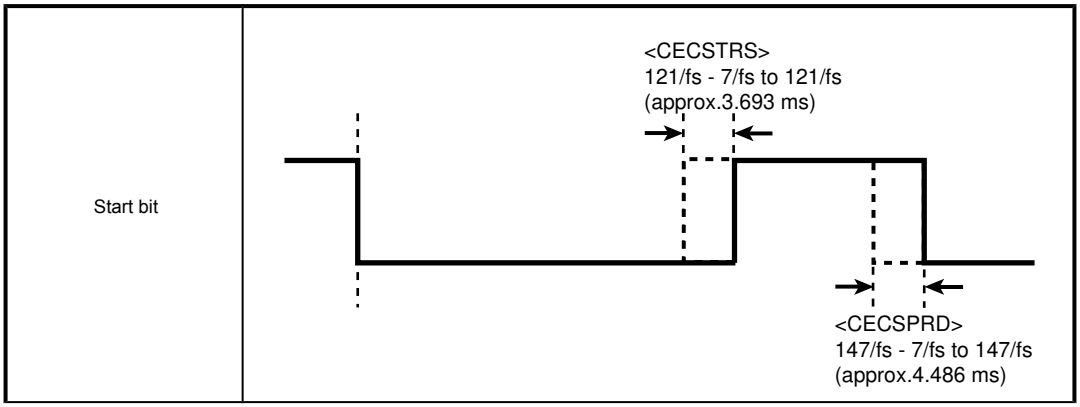
Set the CECTCR <CECBRD> bit when transmitting a broadcast message.If this bit is set, logical "0" response during an ACK cycle results in an error.If not, logical "1" response during an ACK cycle results in an error.

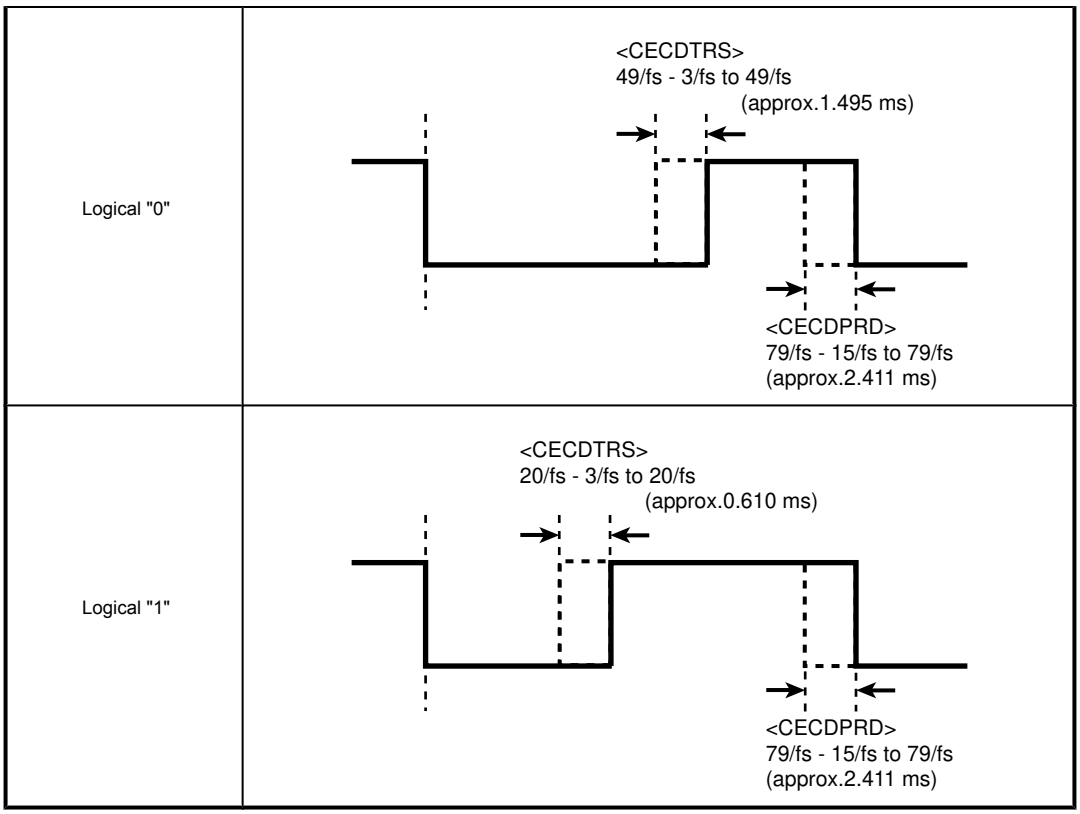
(3) Adjusting Transmission Waveform

Both start bit and data bit are capable of adjusting the rising timing and cycle. With the CECTCR <CECSTRS> <CECSPRD> <CECDTRS> <CECDPRD> bits, the timing can be specified between the defined fastest rising/cycle timing and the reference value.

The following figures show how the waveforms differ according to the configurations of the start bit, logical "0" and logical "1".

Note: <CECDTRS> must be used under the same setting as CECRCR1<CECLNC>.





(4) Preparing Transmission Data

Configure a byte of transmission data and EOM data with the CECTBUF register.

12.4.3.3 Detecting Transmission Error

Error detection during transmission generates an interrupt and stops transmission. It clears the CECTEN <CECTEN> bit.

To identify an error factor, the CECTSTAT register has bits that correspond with each interrupt. You can identify the interrupt factor by checking these bits.

Note: An attempt to stop transmission by an error may cause an improper waveform output to CEC. This is because output is stopped immediately after the error occurs.

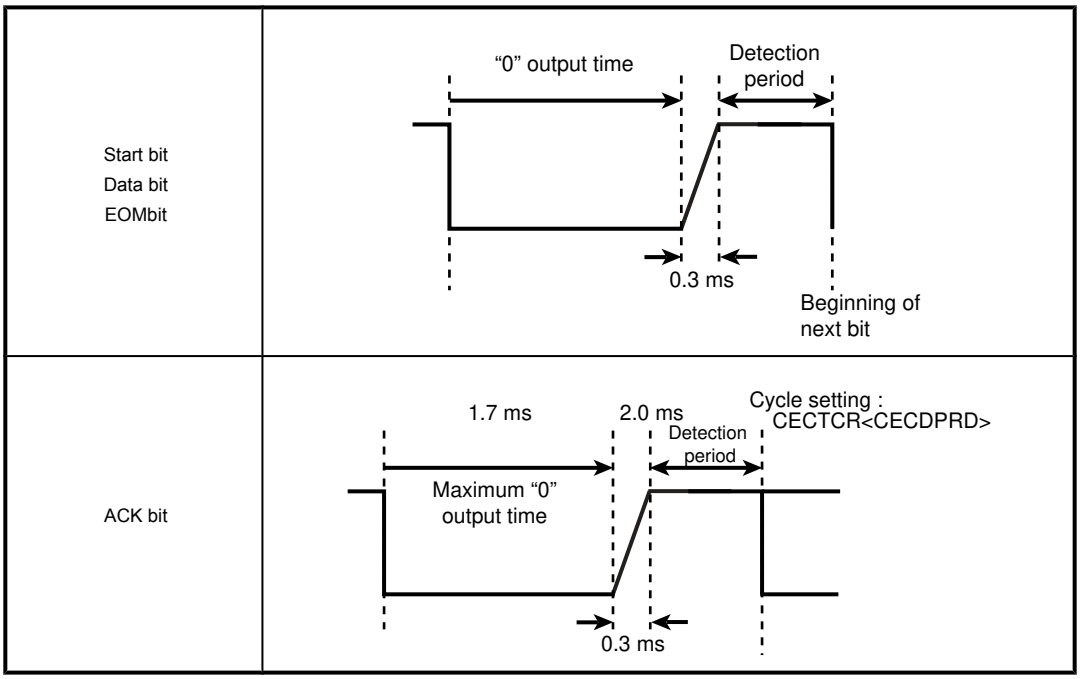
12.4.3.4 Details of Transmission Error

(1) Arbitration Lost

An arbitration lost error occurs when CEC detects "Low" on completion of appropriate low duration.

Detecting an arbitration lost error sets the CECTSTAT <CECTIAL> bit.

Two types of the arbitration lost detection periods are shown below.



(2) ACK error

An ACK error interrupt occurs when an ACK response does not conform to the configuration specified in the CECTCR <CECBRD> bit.

When the ACK error interrupt occurs, the CECTSTAT <CECTIACK> bit is set.

The ACK error is detected in the following cases.

Configuration	Determined as an ACK error when
<CECBRD> = 0 Broadcast transmission?: No	ACK response is logical "1"
<CECBRD> = 1 Broadcast transmission?: Yes	ACK response is logical "0"

(3) Transmit Buffer Underrun

A transmit buffer underrun error is caused by the following sequence.

1. Data in the transmit buffer is transmit to the shift register.

2. An interrupt occurs.

3. A byte of data is transmitted.

4. No data is set to the transmit buffer before starting transmission of a byte of subsequent data.

When an underrun error occurs, the CECTSTAT <CECTIUR> bit is set.

(4) Order of ACK Error and Transmit Buffer Overrun

If interrupt factors of the ACK error and transmit buffer underrun are detected at the end of transmission of a byte of data, the transmit buffer underrun has priority.

The transmit buffer underrun interrupt occurs first and then the ACK error interrupt occurs.

12.4.3.5 Stopping Transmission

To stop transmission, send data including the EOM bit that indicates "1". This generates a transmit completion interrupt.

Please note that proper operation is not ensured if the start bit of transmission is set to "0" during transmission.

12.4.3.6 Retransmission

Transmission is stopped by error detection. To retry the transmission, configure the condition and data of starting the transmission.

12.4.4 Software Reset

The entire CEC function can be initialized by software.

Setting "1" to the CECRESET <CECRESET> bit causes the following operations.

- Reception : Immediately stops. The received data is discarded.
- Transmission : Immediately stops including output to the CEC line.
- Register : All the registers other than CECEN are initialized.

Please note that software reset during transmission may cause the CEC line waveform that does not identical to the defined.

13. Remote control signal preprocessor (RMC)

13.1 Basic operation

Remote control signal preprocessor (hereafter referred to as RMC) receives a remote control signal of which carrier is removed.

13.1.1 Reception of Remote Control Signal

- Sampled 32.768 kHz clock
- Noise canceller
- Leader detection
- Batch reception up to 72bit of data

13.2 Block Diagram

Figure 13-1 shows block diagram of RMC

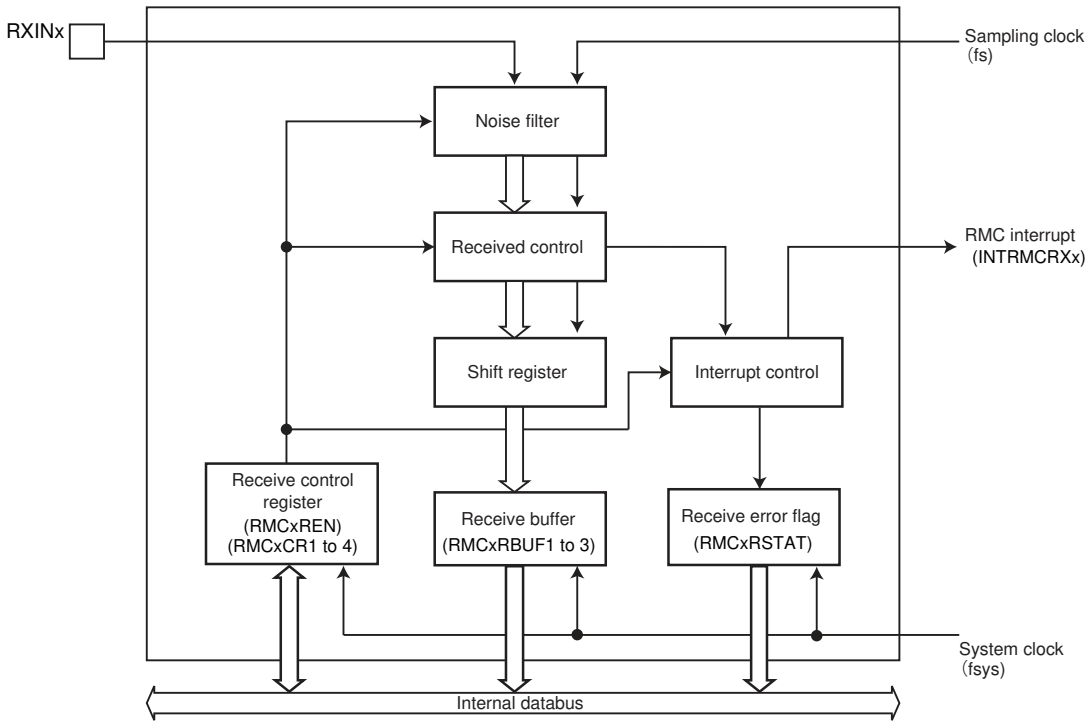


Figure 13-1 Block diagram of RMC

13.3 Registers

13.3.1 Register List

Addresses and names of RMC control registers are shown below.

Channel x	Base Address
Channel0	0x4004_0400
Channel1	0x4004_0440

Register (x=0 to 1)		Address(Base+)
Remote Enable Register	RMCxEN	0x0000
Receive Enable Register	RMCxREN	0x0004
Receive Data Buffer Register 1	RMCxRBUF1	0x0008
Receive Data Buffer Register 2	RMCxRBUF2	0x000C
Receive Data Buffer Register 3	RMCxRBUF3	0x0010
Receive Control Register 1	RMCxRCR1	0x0014
Receive Control Register2	RMCxRCR2	0x0018
Receive Control Register3	RMCxRCR3	0x001C
Receive Control Register 4	RMCxRCR4	0x0020
Receive Status Register	RMCxRSTAT	0x0024
Reserved	-	0x0028
Reserved	-	0x002C
Reserved	-	0x0030

Note: **Access to the <reserved> address is prohibited.**

13.3.2 RMCxEN (Enable Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	I2RMC	RMCEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	I2RMC	R/W	RMC in IDLE mode 0: Disabled 1: Enabled Controls RMC operation in IDLE mode. Set "1" to enable RMC in IDLE Mode. This bit and the <RMCEN> bit can be set simultaneously.
0	RMCEN	R/W	Controls RMC operation. 0: Disabled 1:Enabled 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.

13.3.3 RMCxREN (Receive Enable Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	RMCREN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	RMCREN	R/W	Reception 0: Disabled 1: Enabled Controls reception of RMC. Setting this bit to "1" enables reception.

Note: Enable the <RMCREN> bit after setting the RMCxRCR1, RMCxRCR2 and RMCxRCR3

13.3.4 RMCxRBUF1(Receive Data Buffer Register 1)

	31	30	29	28	27	26	25	24
bit symbol	RMCRBUF (Received data 31 to 24 bit)							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	RMCRBUF (Received data 23 to 16 bit)							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	RMCRBUF (Received data 15 to 8bit)							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RMCRBUF (Received data 7 to 0 bit)							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	RMCRBUF[31:0]	R	Received data (31 to 0 bit) Reads 4 bytes of received data. (31 to 0 bit)

13.3.5 RMCxRBUF2(Receive Data Buffer Register 2)

	31	30	29	28	27	26	25	24
bit symbol	RMCRBUF (Received data 63 to 54 bit)							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	RMCRBUF (Received data 55 to 48 bit)							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	RMCRBUF (Received data 47 to 40 bit)							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RMCRBUF (Received data 39 to 32 bit)							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	RMCRBUF[63:32]	R	Received data (63 to 32 bit) Reads 4 bytes of received data. (63 to 32 bit)

13.3.6 RMCxRBUF3(Receive Data Buffer Register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RMCRBUF (Received data 71 to 64 bit)							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	RMCRBUF[71:64]	R	Received data (71 to 64 bit). Reads 4 bytes of received data. (71 to 64 bit).

Note: The received bit is stored in the data buffer register in MSB-first order, and the last received bit is stored in the LSB (bit 0). If the remote control signal is received in the LSB first algorithm, the received data is stored in reverse sequence.

13.3.7 RMCxRCR1(Receive Control Register 1)

	31	30	29	28	27	26	25	24
bit symbol	RMCLCMAX							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	RMCLCMIN							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	RMCLLMAX							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RMCLLMIN							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-24	RMCLCMAX[7:0]	R/W	Specifies a maximum cycle of leader detection. Calculating formula of the maximum cycle: <RMCLCMAX> × 4/fs [s].
23-16	RMCLCMIN[7:0]	R/W	Specifies a minimum cycle of leader detection. Calculating formula of the minimum cycle: <RMCLCMIN> × 4/fs [s].
15-8	RMCLLMAX[7:0]	R/W	Specifies a maximum low width of leader detection. Calculating formula of the maximum low width: <RMCLLMAX> × 4/fs [s]
7-0	RMCLLMIN[7:0]	R/W	Specifies a minimum low width of leader detection. Calculating formula for the minimum low width: <RMCLLMIN> × 4/fs [s] When RMCxRCR2<RMCLD> = 1, a value of the low-pulse width is less than the specified value, it is defined as data bit.

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 high width	<RMCLCMAX[7:0]> > <RMCLCMIN[7:0]> <RMCLLMAX[7:0]> = 0x00 <RMCLLMIN[7:0]> = don't care
No Leader	<RMCLCMAX[7:0]> = 0x00 <RMCLCMIN[7:0]> = don't care <RMCLLMAX[7:0]> = don't care <RMCLLMIN[[7:0]> = don't care

13.3.8 RMCxRCR2(Receive Control Register 2)

	31	30	29	28	27	26	25	24
bit symbol	RMCLIEN	RMCEDIEN	-	-	-	-	RMCLD	RMCPHM
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	RMCLL							
After reset	1	1	1	1	1	1	1	1
	7	6	5	4	3	2	1	0
bit symbol	RMCDMAX							
After reset	1	1	1	1	1	1	1	1

Bit	Bit Symbol	Type	Function
31	RMCLIEN	R/W	Leader detection interrupt 0: Not generated 1: Generated
30	RMCEDIEN	R/W	Remote control input falling edge interrupt 0: Not generated 1: Generated
29-26	−	R	Read as 0.
25	RMCLD	R/W	Receiving remote control signal with or without leader 0: Disabled 1: Enabled
24	RMCPHM	R/W	Receiving a remote control signal by a phase modulation 0: Not receiving a remote control signal by a phase modulation. (receive by a cycle modulation) 1: Receive remote control signal by a fixed-frequency pulse modulation. To receive a fixed-frequency remote control signal by a pulse modulation, set this bit to "1".
23-16	−	R	Read as 0.
15-8	RMCLL[7:0]	R/W	Excess low width that triggers reception completion and interrupt generation. 0000_0000 to 1111_1110: Reception completion and interrupt generation at <RMCLL> × 1/fs [s]. 1111_1111: not to use as the trigger
7-0	RMCDMAX[7:0]	R/W	Maximum data bit cycle that triggers reception completion and interrupt generation. 0000_0000 to 1111_1110: Reception completion and interrupt generation at <RMCDMAX> × 1/fs [s]. 1111_1111: not to use as the trigger

13.3.9 RMCxRCR3(Receive Control Register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	RMCDATH						
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	RMCDATL						
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-15	-	R	Read as 0.
14-8	RMCDATH[6:0]	R/W	Larger threshold to determine a signal pattern in a phase method Calculating formula of the threshold: <RMCDATH> × 1/fs [s] 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".
7	-	R	Read as 0.
6-0	RMCDATL[6:0]	R/W	Threshold to determine 0 or 1 smaller threshold to determine a signal pattern in a phase method. Calculating formula of the threshold: <RMCDATL> × 1/fs [s] 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".

Note: If the <RMCPHM> bit of the Receive Control Register 2 is "0", <RMCDATH[6:0]> are not enabled. The bits are enabled when <RMCPHM> is "1".

13.3.10 RMCxRCR4(Receive Control Register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RMCPO	-	-	-	RMCNC			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	RMCPO	R/W	Remote control input signal 0: Not reversed 1: Reversed
6-4	−	R	Read as 0.
3-0	RMCNC[3:0]	R/W	Specifies noise cancellation time. 0000: No cancellation 0001 to 1111: cancellation Calculating formula of noise cancellation time: <RMCNC> × 1/fs [s]

13.3.11 RMCxRSTAT (Receive Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	RMCRLIF	RMCLOIF	RMCDMAXIF	RMCEDIF	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RMCRLDR	RMCRNUM						
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15	RMCRLIF	R	Interrupt source flag 0: No leader detection interrupt generated. 1: Leader detection interrupt generated.
14	RMCLOIF	R	Interrupt source flag 0: No low width detection interrupt generated. 1: Low width detection interrupt generated.
13	RMCDMAXIF	R	Interrupt source flag 0: No maximum data bit cycle interrupt generated. 1: Maximum data bit cycle interrupt generated.
12	RMCEDIF	R	Interrupt source flag 0: No falling edge interrupt generated. 1: Falling edge interrupt generated.
11-8	−	R	Read as 0.
7	RMCRLDR	R	Leader detection. 0: Disable leader detection. 1: Enable leader detection.
6-0	RMCRNUM[6:0]	R	The number of received data bit 000_0000:no data bit (only with leader) 000_0001 to 100_1000: 1 to 72bit 100_1001 to 111_1111: 73bit and more 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.

13.4 Operation Description

13.4.1 Reception of Remote Control Signal

13.4.1.1 Sampling clock

A remote control signal is sampled by low-speed 32.768kHz clock (fs).

13.4.1.2 Basic operation

RMC set RMCxRSTAT<RMCRLDR> bit when a leader is detected.

At this time, if RMCxRCR2<RMCLIEN> bit is set, a leader detection interrupt will generate when a leader detecting. When a leader detection interrupt generates, RMCxRSTAT<RMCRLIF> bit is set.

After the leader detecting, each data bit is determined as "0" or "1" in sequence. The results are stored in RMCxRBUF1, RMCxRBUF2 and RMCxRBUF3 registers up to 72 bits. By setting RMCxRCR2< RMCEDIEN> bit, a remote control signal input falling edge interrupt can be generated in each falling edge of data bit. When a remote control signal input falling edge interrupt is generated, RMCxRSTAT< RMCEDIF > bit is set.

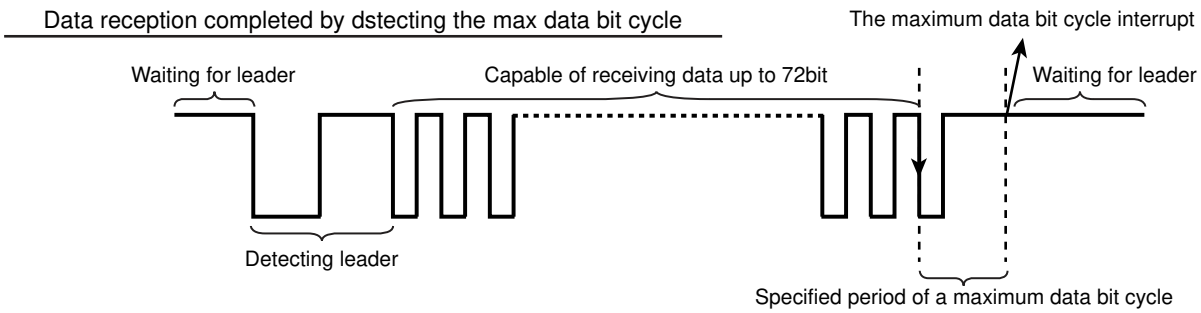
Detecting the maximum data bit cycle or the excess low width completes reception and generates an interrupt.

To check the status of RMC if 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 as 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.



13.4.1.3 Preparation

Configure reception operation of a remote control signal with the Remote Control Signal Receive Control Registers (RMCxRCR1, RMCxRCR2 and RMCxRCR3,RMCxRCR4) before reception.

(1) Settings of Noise Cancelling Time

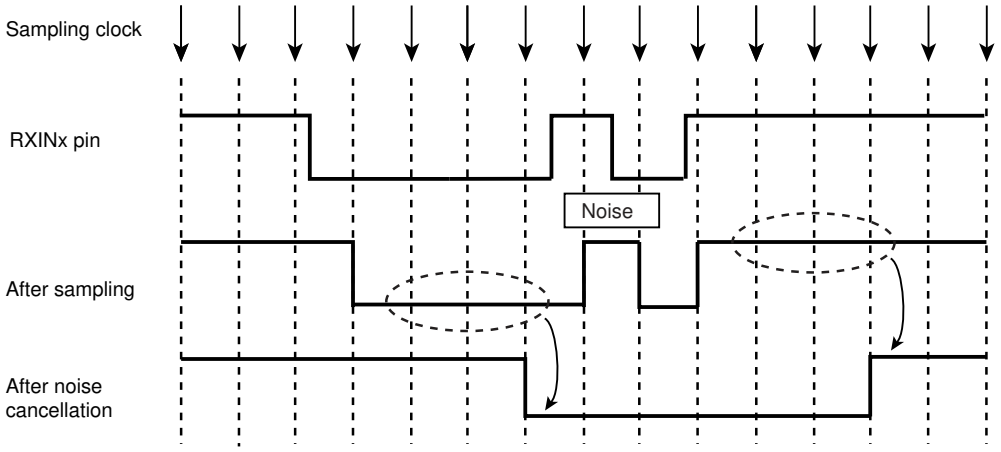
Configure noise cancelling time with the RMCxRCR4 <RMCNC[3:0]> bit.

The signal which is sampled by a sampling clock is cancelled noise.

RMC monitors a sampled remote control signal in each rising edge of a sampling clock. If "High" is monitored, RMC recognizes that the signal was changed to "Low" after monitoring cycles of "Low"s specified in <RMCNC>. If "Low" is monitored, RMC recognizes that the signal was changed to "High" after monitoring cycles of "High" 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 "High" to "Low" upon monitoring 3 cycles of "Low", and the signal is changed from "Low" to "High" upon monitoring 3 cycles of "High".

<RMCNC [3:0]> = 0011 (3 cycle)

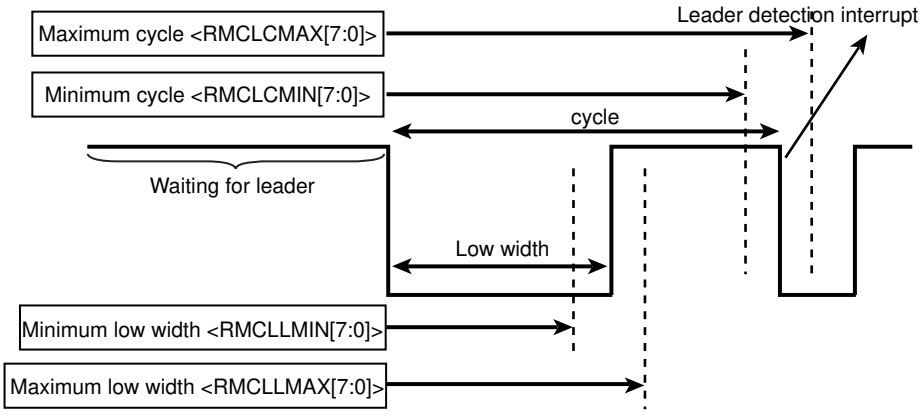


(2) Settings of Detecting Leader

To detect a leader, configure cycle and low width of the leader with the RMCxRCR1 <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]> <RMCLCMIN[7:0]> > <RMCLLMAX[7:0]>
Only high width	<RMCLCMAX[7:0]> > <RMCLCMIN[7:0]> <RMCLLMAX[7:0]> = 0y00000000 <RMCLLMIN[7:0]> = don't care
No leader	<RMCLCMAX[7:0]> = 0y00000000 <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 RMCxRCR1 register settings.



If you want to generate an interrupt when detecting a leader, configure the RMCxRCR2 <RMCLIEN> bit.

A remote control signal without a leader cannot generate a leader detection interrupt.

(3) Setting of 0/1 determination data bit

Based on a falling edge cycle, the data bit is determined as 0 or 1.

There are two kinds of determinations:

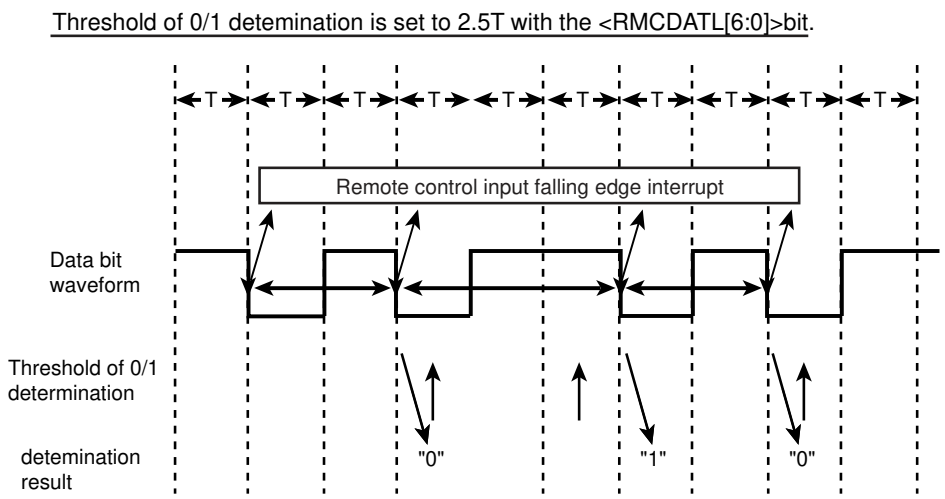
1. Determination by threshold.

   Configure a threshold value to RMCxRCR3<RMCDATL[6:0]> bit which determines data bit as "0" or "1." If the determination value is equal to threshold value or more, it is determined as "1." If the determination value is less than threshold value, it is determined as "0."

2. Determination by falling edge interrupt inputs.

   By setting "1" to the RMCxRCR2<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 timer enables the determination to be done by software.

The followings shows the determination model of data bit.



As for data bit determination of a remote control signal in a phase method, see"13.4.1.8 Receiving a Remote Control Signal in a Phase Method".

(4) Settings 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.

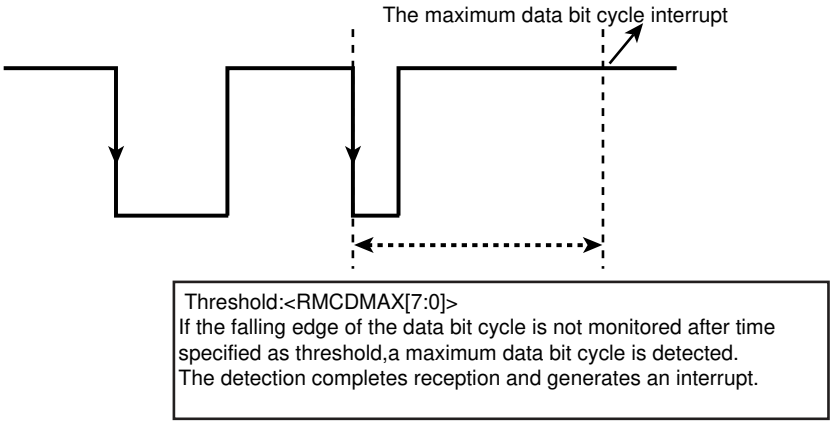
1. Completed by a maximum data bit cycle

   To complete reception by detecting a maximum data bit cycle, you need to configure the RMCxRCR2 <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.

   After interrupt inputs generated, RMCxRSTAT< RMCDMAXIF > bit is set to "1".

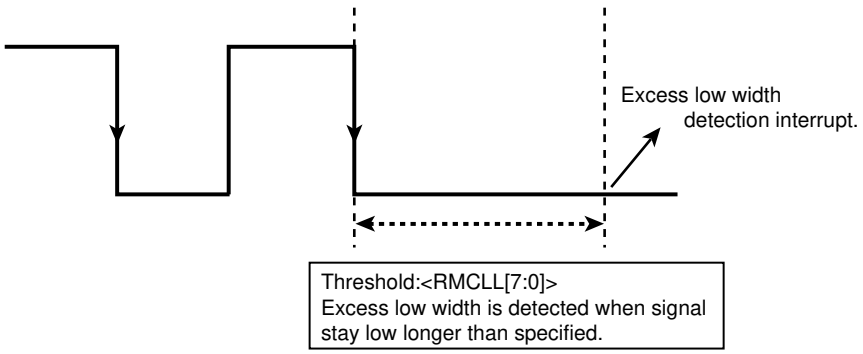


2. Completed by excess low width

   To complete reception by detecting the low width, you need to configure the RMCxRCR2 <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.

   After interrupt inputs generated, RMCxRSTAT<RMCLOIF> bit is set to "1."



13.4.1.4 Enabling Reception

By enabling the RMCxREN <RMCREN> bit after configuring the RMCxRCR1, RMCxRCR2, RMCxRCR3 and RMCxRCR4 registers, RMC is ready for reception. Detecting a leader initiates reception.

Note: Changing the configurations of the RMCxRCR1, RMCxRCR2, RMCxRCR3 and RMCxRCR4 registers during reception may harm their proper operation. Be careful if you change them during reception.

13.4.1.5 Stopping Reception

RMC stops reception by clearing the RMCxREN <RMCREN> bit to "0" (reception disabled).

Clearing this bit during reception stops reception immediately and the received data is discarded.

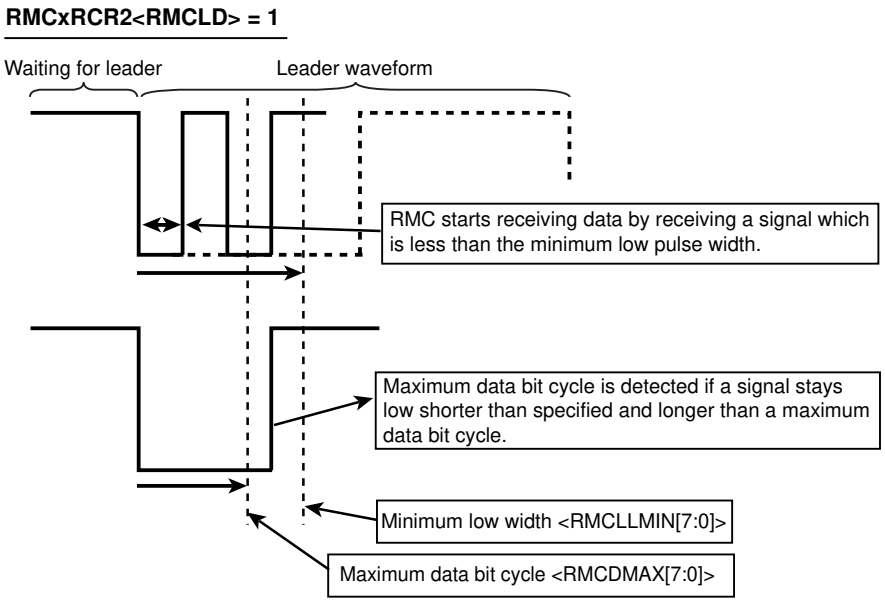
13.4.1.6 Receiving Remote Control Signal without Leader in Waiting Leader

Setting RMCxRCR2 <RMCLD> enables RMC to receive signals with or without a leader.

By setting RMCxRCR2 <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 RMCxRCR1 <RMCLLMAX[7:0] > bits. RMC keeps receiving data until the final data bit is received.

If RMCxRCR2 <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.



13.4.1.7 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 RMCxRCR4 <RMCPO> bit to "1".

This is because RMC is configured to detect a data bit cycle from the falling edge

To detect a leader, configure only low-pulse width of the leader with the <RMCLLMAX[7 : 0]>=0y0000 _ 0000,<RMCLCMAX[7:0]>><RMCLCMIN[7:0]>.

In this case, the value of <RMCLLMIN[7:0]> is set as "don't care".

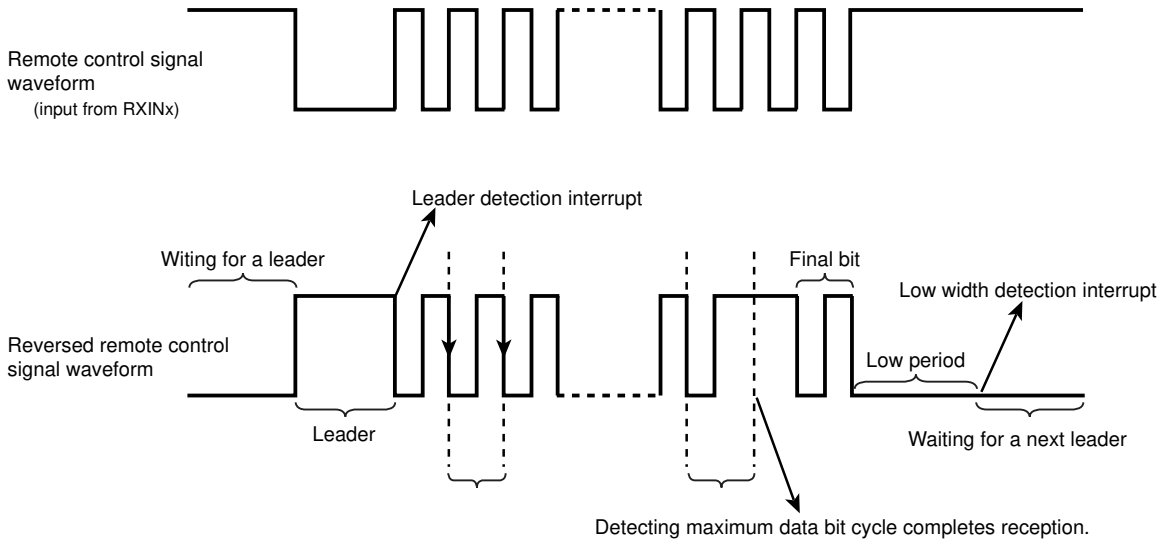
To detect whether data "0" or data "1", configure the threshold of 0/1 detection with the RMCxRCR3 <RMCDATL[6:0]>.

The maximum data bit cycle is configured with the <RMCDMAX[7:0]> of the RMCxRCR2.

To complete data reception, configure the maximum data bit cycle with <RMCDMAX[7:0]> of the RMCxRCR2, and configure the low-pulse width detection with <RMCLL[7:0]>.

After detecting the maximum data bit cycle and confirming the low-pulse with which is specified after receiving the last bit, receiving data is completed.

The RMC generates an interrupt and waits for the next leader.



13.4.1.8 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 RMCxRBUF1, RMCxRBUF 2 and RMCxRBUF3.

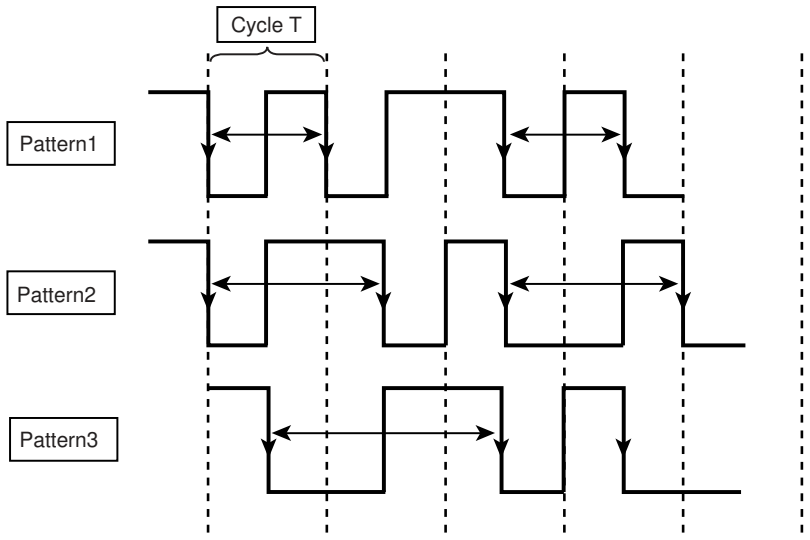
By setting RMCxRCR2<RMCPHM> = "1", RMC enables to receive a remote control signal in the phase method. Each threshold can be configured with the RMCxRCR3 <RMCDATL[6:0]> bits and <RMCDATH [6:0]> bits.

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.

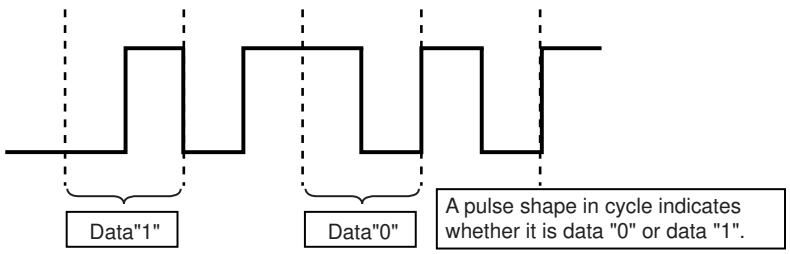
	Determined by	Threshold	Register bits to set
Threshold 1	Pattern 1 & pattern 2	1T to 1.5T	RMCxRCR3<RMCDATL[6:0]>
Threshold 2	Pattern 2 & pattern 3	1.5T to 2T	RMCxRCR3<RMCDATH[6:0]>

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 "11".

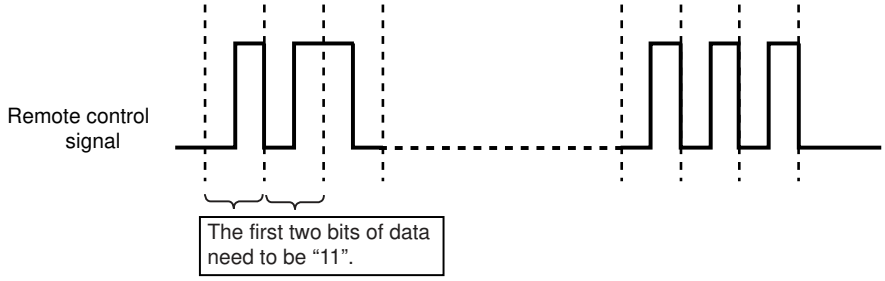
Waveform pattern in phase method



Remote control signal data in phase method



Remote control signal in phase method



14. Analog/Digital Converter (ADC)

14.1 Outline

A 10-bit, sequential-conversion analog/digital converter (AD converter) is built into the TMPM330FDFG/FYFG/FWFG.

This AD converter is equipped with 12 analog input channels.

These 12 analog input channels (pins AIN0 through AIN11) are also used as input/output ports.

Note 1: **To assure conversion accuracy, the specified value must be set to the ADCBAS register.**

Note 2: **If it is necessary to reduce a power current by operating the TMPM330FDFG/FYFG/FWFG in IDLE or STOP mode and if either case shown below is applicable, you must first stop the AD converter and then execute the instruction to put the TMPM330FDFG/FYFG/FWFG into standby mode.**

1. The TMPM330FDFG/FYFG/FWFG must be put into IDLE mode when ADMOD1<I2AD> is "0".

2.The TMPM330FDFG/FYFG/FWFG must be put into STOP mode.

14.2 Configuration

Figure 14-1 shows the block diagram of this AD converter.

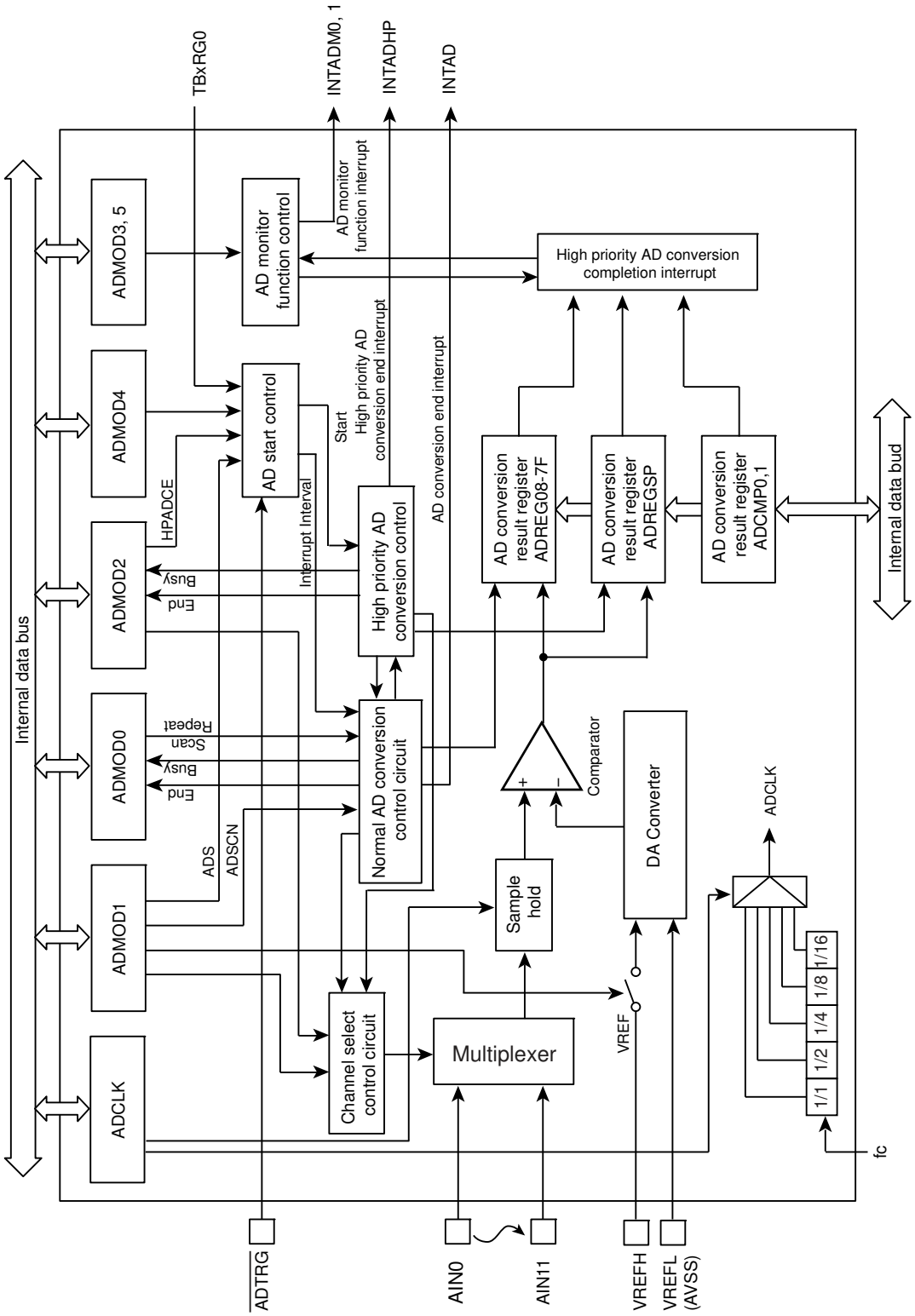


Figure 14-1 AD Converter Block Diagram

14.3 Registers

14.3.1 Register list

The control registers and addresses of the AD converter are as follows.

The AD converter is controlled by the AD mode control registers (ADMOD0 through ADMOD5). The result of AD conversion is stored in the eight AD conversion result registers, ADREG08 through ADREG7F. The highest-priority conversion result is stored in the register ADREGSP.

To assure conversion accuracy, the specified value must be set to the ADCBAS register.

Base Address = 0x4003_0000

Register name		Address (Base+)
Conversion Clock Setting Register	ADCLK	0x0000
Mode Control Register 0	ADMOD0	0x0004
Mode Control Register 1	ADMOD1	0x0008
Mode Control Register 2	ADMOD2	0x000C
Mode Control Register 3	ADMOD3	0x0010
Mode Control Register 4	ADMOD4	0x0014
Mode Control Register 5	ADMOD5	0x0018
Conversion Accuracy Setting Register	ADCBAS	0x0020
Reserved	-	0x0024
Reserved	-	0x0028
Conversion Result Register 08	ADREG08	0x0030
Conversion Result Register 19	ADREG19	0x0034
Conversion Result Register 2A	ADREG2A	0x0038
Conversion Result Register 3B	ADREG3B	0x003C
Conversion Result Register 4C	ADREG4C	0x0040
Conversion Result Register 5D	ADREG5D	0x0044
Conversion Result Register 6E	ADREG6E	0x0048
Conversion Result Register 7F	ADREG7F	0x004C
Conversion Result Register SP	ADREGSP	0x0050
Conversion Result Comparison Register 0	ADCMP0	0x0054
Conversion Result Comparison Register 1	ADCMP1	0x0058

Note: **Access to the "Reserved" address is prohibited.**

14.3.2 ADCBAS (Conversion Accuracy Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADCBAS							
After reset	0	0	1	1	1	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	ADCBAS[7:0]	R/W	Write "0x58".

Note: To assure conversion accuracy, the specified value (0x0000_0058) must be set to the ADCBAS register.

14.3.3 ADCLK (Conversion Clock Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	TSH				-	ADCLK		
After reset	1	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-4	TSH[3:0]	R/W	Select the AD 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
3	−	R	Read as 0.
2-0	ADCLK[2:0]	R/W	Select the AD conversion clock. 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 111: Reserved

A clock count required for conversion is 46 clocks at the minimum.

Examples of sample hold time and conversion time as shown as below.

(Example: If fc = 40MHz)

<TSH[3:0]>	Sample hold time	Conversion time(<ADCLK[2:0]> setting)				
		000 (fc)	001 (fc/2)	010 (fc/4)	011 (fc/8)	100 (fc/16)
1000 (8 conversion clock)	0.2 µs	1.15 µs	2.3 µs	4.6 µs	9.2µs	18.4µs
1001 (16 conversion clock)	0.4 µs	1.35 µs	2.7µs	5.4µs	10.8µs	21.6µs
1010 (24 conversion clock)	0.6 µs	1.55 µs	3.1µs	6.2µs	12.4µs	24.8µs
1011 (32 conversion clock)	0.8 µs	1.75 µs	3.5µs	7.0µs	14.0µs	28.0µs
0011 (64 conversion clock)	1.6 µs	2.55 µs	5.1µs	10.2µs	20.4µs	40.8µs
1100 (128 conversion clock)	3.2 µs	4.15 µs	8.3µs	16.6µs	33.2µs	66.4µs
1101 (512 conversion clock)	12.8 µs	13.75 µs	27.5µs	55.0µs	110.0µs	220.0µs

Note: Do not change the setting of the AD conversion clock during AD conversion.

14.3.4 ADMOD0 (Mode Control Register 0)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	EOCFN	ADBFN	-	ITM		REPEAT	SCAN	ADS
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	EOCFN	R	Normal AD conversion completion flag (note1) 0: Before or during conversion 1: Completion
6	ADBFN	R	Normal AD conversion BUSY flag 0: Conversion stop 1: During conversion
5	−	R	Read as 0.
4-3	ITM[1:0]	R/W	Specify interrupt in fixed channel repeat conversion mode (refer to the table below and note 2)
2	REPEAT	R/W	Specify repeat mode 0: Single conversion mode 1: Repeat conversion mode
1	SCAN	R/W	Specify scan mode 0: Fixed channel mode 1: Channel scan mode
0	ADS	R/W	Start AD conversion start (note 3) 0: Don't care 1: Start conversion "0" is always read.

Specify AD conversion interrupt in fixed channel repeat conversion mode

<ITM[1:0]>	Fixed channel repeat conversion mode <SCAN> = "0", <REPEAT> = "1"
00	Generate in interrupt once every single conversion.
01	Generate interrupt once every 4 conversions.
10	Generate interrupt once every 8 conversions.
11	Setting prohibited.

Note 1: **This flag is "0" cleared by reading the ADMOD0 register.**

Note 2: **It is valid only when it's specified in the fixed channel repeat mode (<REPEAT> ="1", <SCAN> = "0")**

Note 3: **Conversion must be started after setting the mode.**

14.3.5 ADMOD1 (Mode Control Register 1)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	VREFON	I2AD	ADSCN	-	ADCH			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	VREFON	R/W	VREF application control(Note1 and Note2) 0: OFF 1: ON
6	I2AD	R/W	Specify operation mode in IDLE mode 0: Stop 1: Operation
5	ADSCN	R/W	Specify operation mode in channel scan mode 0: 4-channel scan 1: 8-channel scan
4	−	R/W	Write "0".
3-0	ADCH[3:0]	R/W	Select analog input channel (Refer to the below table.)

Select Analog Input Channel

ADMOD0<SCAN> ADMOD1<ADCH[3:0]>	0 Fixed channel	1 Channel scan (<ADSCN> = 0)	1 Channel scan (<ADSCN> = 1)
0000	AIN0	AIN0	AIN0
0001	AIN1	AIN0 to AIN1	AIN0 to AIN1
0010	AIN2	AIN0 to AIN2	AIN0 to AIN2
0011	AIN3	AIN0 to AIN3	AIN0 to AIN3
0100	AIN4	AIN4	AIN0 to AIN4
0101	AIN5	AIN4 to AIN5	AIN0 to AIN5
0110	AIN6	AIN4 to AIN6	AIN0 to AIN6
0111	AIN7	AIN4 to AIN7	AIN0 to AIN7
1000	AIN8	AIN8	AIN8
1001	AIN9	AIN8 to AIN9	AIN8 to AIN9
1010	AIN10	AIN8 to AIN10	AIN8 to AIN10
1011	AIN11	AIN8 to AIN11	AIN8 to AIN11
1100	Setting prohibited		
1101			
1110			
1111			

Note 1: **Before starting AD conversion, write "1" to the <VREFON> bit, wait for 3μs during which time the internal reference voltage should stabilize, and then write "1" to the ADMOD0<ADS>.**

Note 2: **Set <VREFON> to "0" to go into standby mode upon completion of AD conversion.**

14.3.6 ADMOD2 (Mode Control Register 2)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	EOCFHP	ADBFHP	HPADCE	-	HPADCH			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	EOCFHP	R	Top-priority AD conversion completion flag (Note1) 0: Before or during conversion 1: Completion
6	ADBFHP	R	Top-priority AD conversion BUSY flag 0: During conversion halts 1: During conversion
5	HPADCE	R/W	Activate top-priority conversion 0: Don't care 1: Start conversion "0" is always read.
4	−	R/W	Write "0".
3-0	HPADCH[3:0]	R/W	Select analog input channel when activating top-priority conversion. (See the table below)

<HPADCH[3:0]>	Analog input channel whene xecuting top-priority conversion
0000	AIN0
0001	AIN1
0010	AIN2
0011	AIN3
0100	AIN4
0101	AIN5
0110	AIN6
0111	AIN7
1000	AIN8
1001	AIN9
1010	AIN10
1011	AIN11
1100	Setting prohibited
1101	
1110	
1111	

Note 1: **This flag is "0" cleared by reading the ADMOD2 register.**

14.3.7 ADMOD4 (Mode Control Register 4)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	HADHS	HADHTG	ADHS	ADHTG	-	-	ADRST	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	HADHS	R/W	H/W source for activating top-priority AD conversion 0: External trigger 1: Match with timer register 0 (TB5RG0)
6	HADHTG	R/W	H/W for activating top-priority AD conversion 0: Disable 1: Enable
5	ADHS	R/W	H/W source for activating normal AD conversion (note1) 0: External trigger 1: Match with timer register 0 (TB6RG0)
4	ADHTG	R/W	HW for activating normal AD conversion 0: Disable 1: Enable
3-2	−	R	Read as 0.
1-0	ADRST[1:0]	W	Overwriting 10 with 01 allows ADC to be software reset.(note 2)

Note 1: **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 2: **A software reset initializes all the registers except for ADCLK<ADCLK>.**

Note: **The TX03 disables the external trigger used for H/W activation. Therefore "0" cannot be set to <HADHS> and <ADHS>.**

14.3.8 ADMOD3 (Mode Control Register 3)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	-	-	ADOBIC0	ADREGS0				ADOBSV0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	−	R/W	Write "0".
6	−	R	Read as 0.
5	ADOBIC0	R/W	Set the AD monitor function interrupt 0 0: If the value of the conversion result is smaller than the comparison register 0, an interrupt is generated. 1: If the value of the conversion result is bigger than the comparison register 0, an interrupt is generated.
4-1	ADREGS0[3:0]	R/W	Select a target conversion result register when using the AD monitor function 0 (See the below table).
0	ADOBSV0	R/W	AD monitor function 0 0: Disable 1: Enable

<ADREGS0[3:0]>	Conversion result register to be compared	<ADREGS0[3:0]>	Conversion result register to be compared
0000	ADREG08	0100	ADREG4C
0001	ADREG19	0101	ADREG5D
0010	ADREG2A	0110	ADREG6E
0011	ADREG3B	0111	ADREG7F
-	-	1xxx	ADREGSP

14.3.9 ADMOD5 (Mode Control Register 5)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	-	-	ADOBIC1	ADREGS1				ADOBSV1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as 0.
5	ADOBIC1	R/W	Set the AD monitor function interrupt 1. 0: If the value of the conversion result is smaller than the comparison register 1, an interrupt is generated. 1: If the value of the conversion result is bigger than the comparison register 1, an interrupt is generated.
4-1	ADREGS1[3:0]	R/W	Select a target conversion result register when using the AD monitor function 1 (See the below table).
0	ADOBSV1	R/W	AD monitor function 1 0: Disable 1: Enable

<ADREGS1[3:0]>	Conversion result register to be compared	<ADREGS1[3:0]>	Conversion result register to be compared
0000	ADREG08	0100	ADREG4C
0001	ADREG19	0101	ADREG5D
0010	ADREG2A	0110	ADREG6E
0011	ADREG3B	0111	ADREG7F
−	−	1xxx	ADREGSP

14.3.10 ADREG08 (Conversion Result Register 08)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADR0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADR0		-	-	-	-	OVR0	ADR0RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADR0[9:0]	R	AD conversion result Conversion result is stored. For information about the correlation between the conversion channel and the conversion result register, refer to the Table 14-2, chapter 14.4.5.7.
5-2	−	R	Read as 0.
1	OVR0	R	Overrun flag 0: Not generated 1: Generated If the conversion result is overwritten before reading <ADR0>, "1" is set. This bit is "0" cleared when it is read.
0	ADR0RF	R	AD conversion result storage flag 0: Conversion result is not stored 1: Conversion result is stored. If a conversion result is stored, "1" is set. This bit is "0" cleared when the conversion result is read.

Note: **Access to this register must be a half word or a word access.**

14.3.11 ADREG19 (AD Conversion Result Register 19)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADR1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADR1		-	-	-	-	OVR1	ADR1RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADR1[9:0]	R	AD conversion result Conversion result is stored. For information about the correlation between the conversion channel and the conversion result register, refer to the Table 14-2, chapter 14.4.5.7.
5-2	−	R	Read as 0.
1	OVR1	R	Overrun flag 0: Not generated 1: Generated If the conversion result is overwritten before <ADR1>, "1" is set. This bit is "0" cleared when it is read.
0	ADR1RF	R	AD conversion result storage flag 0: Conversion result is not stored. 1: Conversion result is stored. If a conversion result is stored, "1" is set. This bit is "0" cleared when the conversion result is read.

Note: **Access to this register must be a half word or a word access.**

14.3.12 ADREG2A (AD Conversion Result Register 2A)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADR2							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADR2		-	-	-	-	OVR2	ADR2RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADR2[9:0]	R	AD conversion result Conversion result is stored. For information about the correlation between the conversion channel and the conversion result register, refer to the Table 14-2, chapter 14.4.5.7.
5-2	−	R	Read as 0.
1	OVR2	R	Overrun flag 0: Not generated. 1: Generated. If a conversion result is overwritten before reading <ADR2>, "1" is set. This bit is "0" cleared when it is read.
0	ADR2RF	R	AD conversion result storage flag 0: Conversion result is not stored. 1: Conversion result is stored. If a conversion result is stored, "1" is set. This bit is "0" cleared when the conversion result is read.

Note: **Access to this register must be a half word or a word access.**

14.3.13 ADREG3B (AD Conversion Result Register 3B)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADR3							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADR3		-	-	-	-	OVR3	ADR3RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADR3[9:0]	R	AD conversion result. Conversion result is stored. For information about the correlation between the conversion channel and the conversion result register, refer to the Table 14-2, chapter 14.4.5.7.
5-2	−	R	Read as 0.
1	OVR3	R	Overrun flag 0: Not generated. 1: Generated. If a conversion result is overwritten before reading <ADR3>, "1" is set. This bit is "0" cleared when it is read.
0	ADR3RF	R	AD conversion result storage flag 0: Conversion result is not stored. 1: Conversion result is stored. If a conversion result is stored, "1" is set. This bit is "0" cleared when the conversion result is read.

Note: **Access to this register must be a half word or a word access.**

14.3.14 ADREG4C (AD Conversion Result Register 4C)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADR4							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADR4		-	-	-	-	OVR4	ADR4RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADR4[9:0]	R	AD conversion result. Conversion result is stored. For information about the correlation between the conversion channel and the conversion result register, refer to the Table 14-2, chapter 14.4.5.7.
5-2	−	R	Read as 0.
1	OVR4	R	Overrun flag 0: Not generated. 1: Generated If a conversion result is overwritten before reading <ADR4>, "1" is set. This bit is "0" cleared when it is read.
0	ADR4RF	R	AD conversion result storage flag 0: Conversion result is not stored. 1: Conversion result is stored. If a conversion result is stored, "1" is set. This bit is "0" cleared when the conversion result is read.

Note: **Access to this register must be a halfword or a word access.**

14.3.15 ADREG5D (AD Conversion Result Register 5D)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADR5							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADR5		-	-	-	-	OVR5	ADR5RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADR5[9:0]	R	AD conversion result. Conversion result is stored. For information about the correlation between the conversion channel and the conversion result register, refer to the Table 14-2, chapter 14.4.5.7.
5-2	−	R	Read as 0.
1	OVR5	R	Overrun flag 0: Not generated. 1: Generated If a conversion result is overwritten before reading <ADR5>, "1" is set. This bit is "0" cleared when it is read.
0	ADR5RF	R	AD conversion result storage flag 0: Conversion result is not stored. 1: Conversion result is stored. If a conversion result is stored, "1" is set. This bit is "0" cleared when the conversion result is read.

Note: **Access to this register must be a half word or a word access.**

14.3.16 ADREG6E (AD Conversion Result Register 6E)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADR6							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADR6		-	-	-	-	OVR6	ADR6RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADR6[9:0]	R	AD conversion result. Conversion result is stored. For information about the correlation between the conversion channel and the conversion result register, refer to the Table 14-2, chapter14.4.5.7.
5-2	−	R	Read as 0.
1	OVR6	R	Overrun flag 0: Not generated. 1: Generated. If a conversion result is overwritten before reading <ADR6>, "1" is set. This bit is "0" cleared when it is read.
0	ADR6RF	R	AD conversion result storage flag 0: Conversion result is not stored. 1: Conversion result is stored. If a conversion result is stored, "1" is set. This bit is "0" cleared when the conversion result is read.

Note: **Access to this register must be a half word or a word access.**

14.3.17 ADREG7F (AD Conversion Result Register 7F)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADR7							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADR7		-	-	-	-	OVR7	ADR7RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADR7[9:0]	R	AD conversion result. Conversion result is stored. For information about the correlation between the conversion channel and the conversion result register, refer to the Table 14-2, chapter 14.4.5.7.
5-2	−	R	Read as 0.
1	OVR7	R	Overrun flag 0: Not generated 1: Generated If a conversion result is overwritten before reading <ADR7>, "1" is set. This bit is "0" cleared when it is read.
0	ADR7RF	R	AD conversion result storage flag 0: Conversion result is not stored. 1: Conversion result is stored. If a conversion result is stored, "1" is set. This bit is "0" cleared when the conversion result is read.

Note: **Access to this register must be a half word or a word access.**

14.3.18 ADREGSP (AD Conversion Result Register SP)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADRSP							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADRSP		-	-	-	-	OVRSP	ADRSPRF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADRSP[9:0]	R	AD conversion result. Top-priority AD conversion result is stored
5-2	−	R	Read as 0
1	OVRSP	R	Overrun flag 0: Not generated 1: Generated If a conversion result is overwritten before reading <ADRSP>, "1" is set. This bit is "0" cleared when it is read.
0	ADRSPRF	R	AD conversion result storage flag 0: Conversion result is not stored. 1: Conversion result is stored. If a conversion result is stored, "1" is set. This bit is "0" cleared when the conversion result is read.

Note: **Access to this register must be a half word or a word access.**

14.3.19 ADCMP0 (AD Conversion Result Comparison Register 0)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADCOM0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADCOM0		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADCOM0[9:0]	R/W	When AD monitor function 0 is enabled, it sets a value to be compared with the value of the conversion result register specified by ADMOD3<ADREGS0>.
5-0	−	R	Read as 0.

Note: **To write values into this register, the AD monitor function 0 must be disabled (ADMOD3<ADBSV0> ="0").**

14.3.20 ADCMP1 (AD Conversion Result Comparison Register 1)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADCOM1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADCOM1		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADCOM1[9:0]	R/W	When AD monitor function 1 is enabled, it sets a value to be compared with the value of the conversion result register specified by ADMODt<ADREGS1>.
5-0	−	R	Read as 0.

Note: **To write values into this register, the AD monitor function 1 must be disabled (ADMOD5<ADBSV1>="0").**

14.4 Description of Operations

14.4.1 Analog Reference Voltage

The "High" level of the analog reference voltage shall be applied to the VRFEH pin, and the "Low" shall be applied to the VREFL pin.

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.

By writing "0" to the ADMOD1<VREFON> bit, a switched-on state of VREFH − VREFL can be turned into a switched -off state. To switch to the power-consumption mode, set "0" to the <VREFON> bit after conversion.

Note: **VREFL and AVSS are shared by TMPM330FDFG/FYFG/FWFG.**

14.4.2 AD Conversion Mode

Two types of AD conversion are supported: normal AD conversion and top-priority AD conversion.

For normal AD conversion, the following four operation modes are supported.

14.4.2.1 Normal AD conversion

For normal AD conversion, the following four operation modes are supported and the operation mode is selected with the ADMOD0<REPEAT, SCAN>.

- Fixed channel single conversion mode
- Channel scan single conversion mode
- Fixed channel repeat conversion mode
- Channel scan repeat conversion mode

(1) Fixed channel single conversion mode

If ADMOD0<REPEAT, SCAN> is set to "00", "AD conversion is performed in the fixed channel single conversion mode.

In this mode, AD conversion is performed once for one channel selected. After AD conversion is completed, ADMOD0<EOCFN> is set to "1", ADMOD0<ADBFN> is cleared to "0", and the AD conversion completion interrupt request (INTAD) is generated. <EOCFN> is cleared to "0" upon read.

(2) Channel scan single conversion mode

If ADMOD0 <REPEAT, SCAN> is set to "01," AD conversion is performed in the channel scan single conversion mode.

In this mode, AD conversion is performed once for each scan channel selected. After AD scan conversion is completed, ADMOD0<EOCFN> is set to "1", ADMOD0<ADBFN> is cleared to "0", and the conversion completion interrupt request (INTAD) is generated. <EOCFN> is cleared to "0".

(3) Fixed channel repeat conversion mode

If ADMOD0<REPEAT, SCAN> is set to "10", AD conversion is performed in fixed channel repeat conversation mode.

In this mode, AD conversion is performed repeatedly for one channel selected. After AD conversion is completed, ADMOD0<EOCFN> is set to "1". ADMOD0<ADBFN> is not cleared to "0". It remains at "1". The timing with which the conversion completion interrupt request (INTAD) is generated can be selected by setting ADMOD0<ITM> to an appropriate setting. <EOCFN> is set with the same timing as this interrupt INTAD is generated.

By reading <EOCFN>, it is cleared to "0".

(4) Channel scan repeat conversion mode

If ADMOD0<REPEAT, SCAN> is set to "11", AD conversion is performed in the channel scan repeat conversion mode.

In this mode, AD conversion is performed repeatedly for a scan channel selected. Each time one AD scan conversion is completed, ADMOD0<EOCFN> is set to "1", and the conversion completion interrupt request (INTAD) is generated. ADMOD0<ADBFN> is not cleared to "0". It remains at "1". <EOCFN> is cleared to "0" upon read.

14.4.2.2 Top-priority AD conversion

By interrupting ongoing normal AD conversion, top-priority AD conversion can be performed.

The fixed-channel single conversion is automatically selected, irrespective of the ADMOD0<REPEAT, SCAN> setting. When conditions to start operation are met, a conversion is performed just once for a channel designated by ADMOD2<HPADCH>. When conversion is completed, the top-priority AD conversion completion interrupt (INTADHP) is generated, and ADMOD2<EOCFHP> showing the completion of AD conversion is set to "1". <ADBFHP> returns to "0". EOCFHP flag is cleared to "0" upon read.

Top-priority AD conversion activated while top-priority AD conversion is under way is ignored.

14.4.3 AD Monitor Function

There are two channels of AD monitor function.

If ADMOD3<ADOBSV0> and ADMOD5<ADOBSV1> are set to "1", the AD monitor function is enabled. If the value of the conversion result register specified by ADMOD3<ADREGS0> and ADMOD5<ADREGS1> becomes larger or smaller ("Larger" or "Smaller" to be designated by ADMOD3<ADOBIC0> and ADMOD5<ADBIC1>) than the value of a comparison register, the AD monitor function interrupt (INTADM0,INTADM1) is generated. This comparison operation is performed each time a result is stored in a corresponding conversion result register.

If the conversion result register assigned to perform the AD monitor function is continuously used without reading the conversion result, the conversion result is overwritten. The conversion result storage flag <ADRxRF> and the overrun flag <OVRx> remain being set.

14.4.4 Selecting the Input Channel

After reset, ADMOD0<REPEAT,SCAN> is initialized to "00" and ADMOD1<ADCH[3:0]> is initialized to "0000".

The channels to be converted are selected according to the operation mode of the AD converter as shown below.

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 AIN11 by setting ADMOD1<ADCH> to an appropriate setting.

   - If the analog input channel is used in a scan state (ADMOD0<SCAN> = "1")

     One scan mode is selected from the scan modes by setting ADMOD1 <ADCH> and ADSCN to an appropriate setting.

2. Top-priority AD conversion mode

   One channel is selected from analog input pins from AIN0 through AIN11 by setting ADMOD2<HPADCH> to an appropriate setting.

14.4.5 AD Conversion Details

14.4.5.1 Starting AD Conversion

Normal AD conversion is activated by setting ADMOD0<ADS> to "1". Top-priority AD conversion is activated by setting ADMOD2<HPADCE> to "1".

Four 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<REPEAT,SCAN> 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 H/W activation source selected by ADMOD4<ADHS>, and top-priority AD conversion can be activated using the HW activation source selected by ADMOD4<HADHS>. If bits of <ADHS> and <HADHS> are "0", normal and top-priority AD conversions are activated in response to the input of a falling edge through the $\overline{\text{ADTRG}}$ pin. If these bits are "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.

To permit H/W activation, set ADMOD4<ADHTG> to "1" for normal AD conversion and set ADMOD4<HADHTG> to "1" for top-priority AD conversion.

Software activation is still valid even after H/W activation has been permitted.

Note: **When an external trigger is used for the HW activation source of a top-priority AD conversion, an external trigger cannot be set for activating normal AD conversion H/W.**

Note: **The TMPM330FDFG/FYFG/FWFG disables the external trigger used for H/W activation. Therefore "0" cannot be set to <HADHS> and <ADHS>.**

14.4.5.2 AD Conversion

When normal AD conversion starts, the AD conversion Busy flag (ADMOD0<ADBFN>) showing that AD conversion is under way is set to "1".

When top-priority AD conversion starts, the top-priority AD conversion Busy flag (ADMOD2<ADBFHP>) showing that AD conversion is underway is set to "1". At that time, the value of the Busy flag ADMOD0<ADBFN> for normal AD conversion before the start of top-priority AD conversions are retained. The value of the conversion completion flag ADMOD0<EOCFN> for normal AD conversion before the start of top-priority AD conversion is retained.

Note: **Normal AD conversion must not be activated when top-priority AD conversion is under way.**

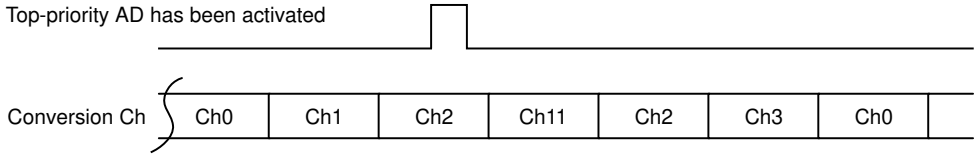
14.4.5.3 Top-priority AD conversion during normal AD conversion

If top-priority AD conversion has been activated during normal AD conversion, ongoing normal AD conversion is suspended, and restarts normal AD conversion after top-priority AD conversion is completed.

If ADMOD2<HPADCE> is set to "1" during normal AD conversion, ongoing normal AD conversion is suspended, and the top-priority AD conversion starts; specifically, AD conversion (fixed-channel single conversion) is executed for a channel designated by ADMOD2<HPADCH>. After the result of this top-priority AD conversion is stored in the storage register ADREGSP, normal AD conversion is resumed.

If H/W activation of top-priority AD conversion is authorized during normal AD conversion, ongoing AD conversion is discontinued when requirements for activation using a H/W activation resource are met, and top-priority AD conversion (fixed-channel single conversion) starts for a channel designated by ADMOD2<HPADCH>. After the result of this top-priority AD conversion is stored in the storage register ADREGSP, normal AD conversion is resumed.

For example, if channel repeat conversion is activated for channels AIN0 through AIN3 and if <HPADCE> is set to "1" during AIN2 conversion, AIN2 conversion is suspended, and conversion is performed for a channel designated by <HPADCH> (AIN11 in the case shown below). After the result of conversion is stored in ADREGSP, channel repeat conversion is resumed, starting from AIN2.



14.4.5.4 Stopping Repeat Conversion Mode

To stop the AD conversion operation in the repeat conversion mode (fixed-channel repeat conversion mode or channel scan conversion mode), write "0" to ADMOD0<REPEAT>. When ongoing AD conversion is completed, the repeat conversion mode terminates, and ADMOD0<ADBFN> is set to "0".

14.4.5.5 Reactivating normal AD conversion

To reactivate normal AD conversion while the conversion is underway, a software reset (ADMOD3<ADRST>) must be performed before starting AD conversion. The H/W activation method must not be used to reactivate normal AD conversion.

14.4.5.6 Conversion completion

(1) Normal AD conversion completion

When normal AD conversion is completed, the AD conversion completion interrupt (INTAD) is generated. The result of AD conversion is stored in the storage register, and two registers change: the register ADMOD0<EOCFN> which indicates the completion of AD conversion and the register ADMOD0<ADBFN>.

Interrupt request, conversion register storage register and <EOCFN><ADBFN> change with a different timing according to a mode selected.

In mode other than fixed-channel repeat conversion mode, conversion results are stored in AD conversion result registers (ADREG08 through ADRG7F) corresponding to a channel.

In fixed-channel repeat conversion mode, the conversion results are sequentially stored in storage registers ADREG08 through ADREG7F. However, if interrupt setting on <ITM> is set to be generated each time one AD conversion is completed, the conversion result is stored only in ADREG08. If interrupt setting on <ITM> is set to be generated each time four AD conversions are completed, the conversion results are sequentially stored in ADREG08H through ADREG3B.

Interrupt requests, flag changes and conversion result registers in each mode are as shown below.

- Fixed-channel single conversion mode

  After AD conversion completed, ADMOD0<EOCFN> is set to "1", ADMOD0<ADBFN> is cleared to "0", and the interrupt request is generated.

  Conversion results are stored a conversion result register correspond to a channel.

- Channel scan single conversion mode

  After the channel scan conversion is completed, ADMOD0<EOCFN> is set to "1", ADMOD0<ADBFN> is set to "0", and the interrupt request INTAD is generated.

  Conversion results are stored a conversion result register correspond to a channel.

- Fixed-channel repeat conversion mode

  ADMOD0<ADBFN> 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> to an appropriate setting. ADMOD0<EOCFN> is set with the same timing as this interrupt INTAD is generated.

  a. One conversion

     With <ITM[1:0]> set to "00", an interrupt request is generated each time one AD conversion is completed. In this case, the conversion results are always stored in the storage register ADREG08. After the conversion result is stored, <EOCFN> changes to "1".

  b. Four conversions

     With <ITM[1:0]> set to "01", an interrupt request is generated each time four AD conversions are completed. In this case, the conversion results are sequentially stored in the storage register ADREG08 through ADREG3B. After the conversion result is stored in ADREG3B, <EOCFN> is set to "1", and the storage of subsequent conversion results starts from ADREG08.

c. 8 conversions

With <ITM[1:0]> set to "10", an interrupt request is generated each time eight AD conversions are completed. In this case, the conversion results are sequentially stored in the storage register ADREG08 through ADREG7F. After the conversion result is stored in ADREG7F, <EOCFN> is set to "1", and the storage of subsequent conversion results starts from ADREG08.

- Channel scan repeat conversion mode

Each time one AD conversion is completed, ADMOD0<EOCF> is set to "1" and interrupt request INTAD is generated. ADMOD0<ADBFN> is not cleared to "0". It remains at "1".

AD conversion results are stored in a AD conversion result register corresponding to a channel.

(2) Top-priority AD conversion completion

After the AD conversion is completed, the top-priority AD conversion completion interrupt (INTADHP) is generated, and ADMOD2<EOCFHP> which indicates the completion of top-priority AD conversion is set to "1".

AD conversion results are stored in the AD conversion result register SP.

(3) Data polling

To confirm the completion of AD conversion without using interrupts, data polling can be used. When AD conversion is completed, ADMOD0<EOCFN> is set to "1". To confirm the completion of AD conversion and to obtain the results, poll this bit.

AD conversion result storage register must be read by half word or word access. If <OVRx> = “0” and <ADRxRF> = “1”, a correct conversion result has been obtained.

14.4.5.7 Interrupt generation timings and AD conversion result storage register

Table 14-1 shows a relation in the following three items: AD conversion modes, interrupt generation timings and flag operations. Table 14-2 shows a relation between analog channel inputs and AD conversion result registers.

Table 14-1 Relations in conversion modes, interrupt generation timings and flag operations

Conversion mode		Scan/repeat mode setting (ADMOD0)			Interrupt generation timing	<EOCFN>/ <EOCFHP> set timing (See note)	ADMOD0	ADMOD2
		<REPEAT>	<SCAIN>	<ITM[1:0]>			<ADBFN> (After the interrupt is generated)	<ADBFHP>
Normal conversion	Fixed-channel single conversion	0	0	–	After generation is completed.	After conversion is completed.	0	–
	Fixed-channel repeat conversion	1	0	00	Each time one conversion is completed.	After one conversion is completed.	1	–
				01	Each time four conversion is completed.	After four conversions are completed.	1	–
				10	Each time eight conversion is completed.	After eight conversions are completed.	1	–
	Channel scan single conversion	0	1	–	After scan conversion is completed.	After scan conversion is completed.	0	–
	Channel scan repeat conversion	1	1	–	After one scan conversion is completed.	After one scan conversion is completed.	1	–
Top-priority conversion		–	–	–	After completion is completed.	Conversion completion	–	0

Note: **ADMOD0<EOCFN> and ADMOD2<EOCFHP> are cleared upon read.**

Table 14-2 Relation between analog channels input and AD conversion result registers

Analog input channels	Normal AD conversion				Top-priority AD conversion
	Other conversion mode than those shown on the right side	Fixed channel repeat conversion mode (every one conversion)	Fixed channel repeat conversion mode (every four conversions)	Fixed channel repeat conversion mode (every eight conversions)	
AIN0	ADREG08	ADREG08 fixed	ADREG08 → ADREG3B	ADREG08 → ADREG7F	ADREGSP
AIN1	ADREG19				
AIN2	ADREG2A				
AIN3	ADREG3B				
AIN4	ADREG4C				
AIN5	ADREG5D				
AIN6	ADREG6E				
AIN7	ADREG7F				
AIN8	ADREG08				
AIN9	ADREG19				
AIN10	ADREG2A				
AIN11	ADREG3B				

Note: **To access the conversion result register, use a half-word or a word access.**

Cautions
The result value of AD conversion may vary depending on the fluctuation of the supply voltage, or may be affected by noise. When using analog input pins and ports alternately, do not read and write ports during conversion because the conversion accuracy may be reduced. Also the conversion accuracy may be reduced if the output ports current fluctuate during AD conversion. Please take counteractive measures with the program such as averaging the AD conversion results.

15. 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 watchdog timer detects a runaway, it generates a INTWDT interrupt or reset.

Note: INTWDT interrupt is a factor of the non-maskable interrupts (NMI).

Also, the watchdog timer notifies of the detecting malfunction to the external peripheral devices from the watchdog timer pin ($\overline{\text{WDTOUT}}$) by outputting "Low".

Note: This product does not have the watchdog timer out pin ($\overline{\text{WDTOUT}}$).

15.1 Configuration

Figure 15-1shows the block diagram of the watchdog timer.

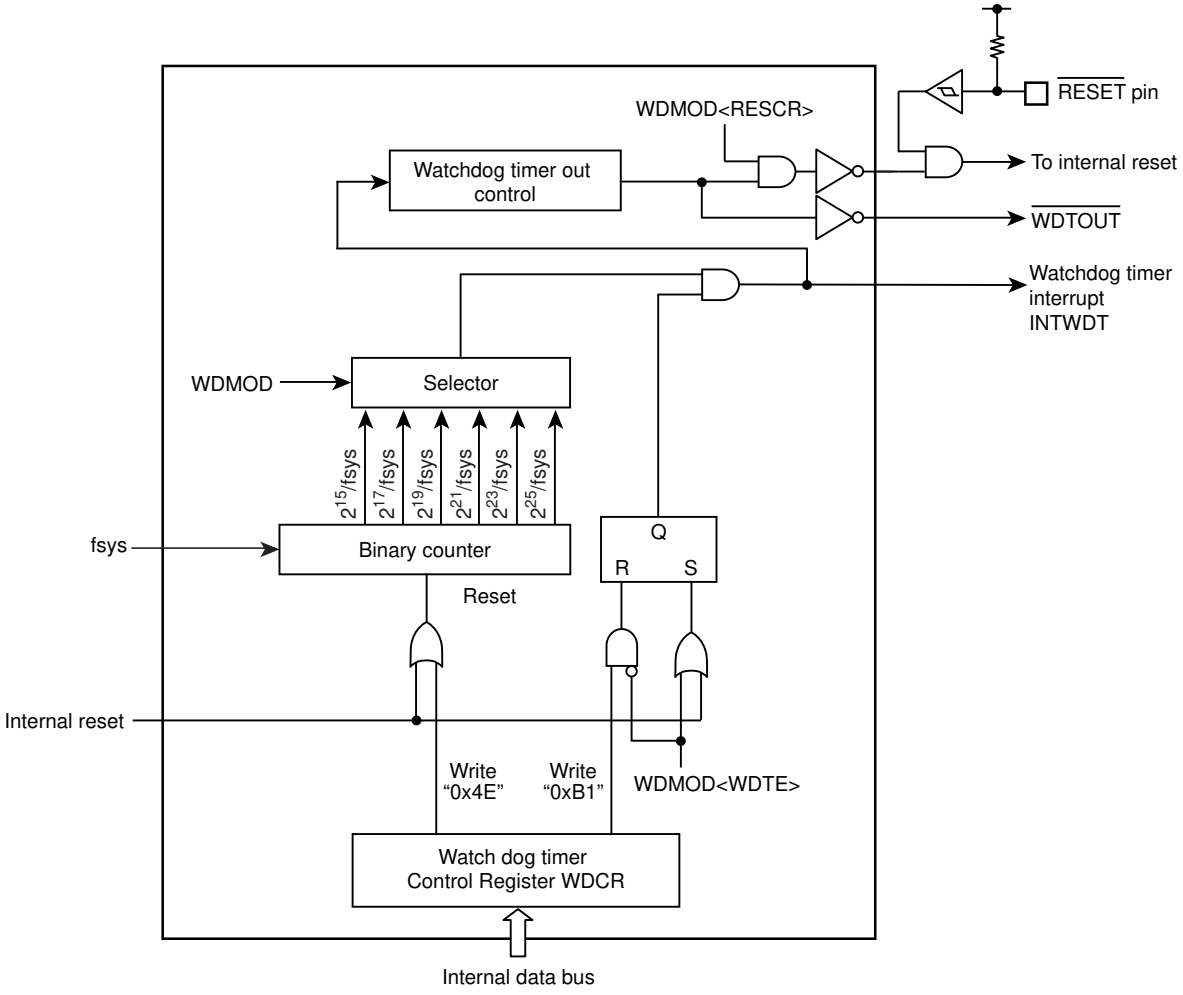


Figure 15-1 Block Diagram of the Watchdog Timer

15.2 Register

The followings are the watchdog timer control registers and addresses.

Base Address = 0x4004_0000

Register name		Address(Base+)
Watchdog Timer Mode Register	WDMOD	0x0000
Watchdog Timer Control Register	WDCR	0x0004

15.2.1 WDMOD(Watchdog Timer Mode Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	WDTE	WDTP			-	I2WDT	RESCR	-
After reset	1	0	0	0	0	0	1	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	WDTE	R/W	Enable/Disable control 0:Disable 1:Enable
6-4	WDTP[2:0]	R/W	Selects WDT detection time(Refer toTable 15-1) 000: 2^{15}/fsys 100: 2^{23}/fsys 001: 2^{17}/fsys 101: 2^{25}/fsys 010: 2^{19}/fsys 110:Setting prohibited. 011: 2^{21}/fsys 111:Setting prohibited.
3	−	R	Read as 0.
2	I2WDT	R/W	Operation when IDLE mode 0: Stop 1:In operation
1	RESCR	R/W	Operation after detecting malfunction 0: INTWDT interrupt request generates. (Note) 1: Reset
0	−	R/W	Write 0.

Note: INTWDT interrupt is a factor of the non-maskable interrupts (NMI).

Table 15-1 Detection time of watchdog timer (fc = 40 MHz)

Clock gear value CGSYSCR<GEAR[2:0]>	WDMOD<WDTP[2:0]>					
	000	001	010	011	100	101
000 (fc)	0.82 ms	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms
100 (fc/2)	1.63 ms	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s
101 (fc/4)	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s
110 (fc/8)	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s	6.71 s

15.2.2 WDCR (Watchdog Timer Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	WDCR							
After reset	-	-	-	-	-	-	-	-

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	WDCR	W	Disable/Clear code 0xB1:Disable code 0x4E: Clear code Others:Reserved

15.3 Operations

15.3.1 Basic Operation

The Watchdog timer is consists of the binary counters that work using the system clock (fsys) as an input. Detecting time can be selected between 2^{15}, 2^{17}, 2^{19} , 2^{21}, 2^{23} and 2^{25} by the WDMOD<WDTP[2:0]>. The detecting time as specified is elapsed, the watchdog timer interrupt (INTWDT) generates, and the watchdog timer out pin ($\overline{\text{WDTOUT}}$) output "Low".

To detect malfunctions (runaways) of the CPU caused by noise or other disturbances, the binary counter of the watchdog timer should be cleared by software instruction before INTWDT interrupt generates. If the binary counter is not cleared, the non-maskable interrupt generates by INTWDT. Thus CPU detects malfunction (runway), malfunction countermeasure program is performed to return to the normal operation.

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.

Note: This product does not include a watchdog timer out pin ($\overline{\text{WDTOUT}}$).

15.3.2 Operation Mode and Status

The watchdog timer begins operation immediately after a reset is cleared.

If not using the watchdog timer, it should be disabled.

The watchdog timer cannot be used in the STOP mode, SLEEP mode and SLOW mode where high-speed frequency clock is stopped. Before transition to these modes, the watchdog timer should be disabled.

In IDLE mode, its operation depends on the WDMOD <I2WDT> setting.

Also, the binary counter is automatically stopped during debug mode.

15.4 Operation when malfunction (runaway) is detected

15.4.1 INTWDT interrupt generation

In the Figure 15-2 shows the case that INTWDT interrupt generates (WDMOD<RESCR>="0").

When an overflow of the binary counter occurs, INTWDT interrupt generates. It is a factor of non-maskable interrupt (NMI). Thus CPU detects non-maskable interrupt and performs the countermeasure program.

The factor of non-maskable interrupt is the plural. CGNMIFLG identifies the factor of non-maskable interrupts. In the case of INTWDT interrupt, CGNMIFLG<NMIFLG0> is set.

When INTWDT interrupt generates, simultaneously the watchdog timer out ($\overline{\text{WDTOUT}}$) output "Low". $\overline{\text{WDTOUT}}$ becomes "High" by the watchdog timer clearing that is writing clear code 0x4E to the WDCR register.

Note: This product does not have the watchdog timer output pin($\overline{\text{WDTOUT}}$).

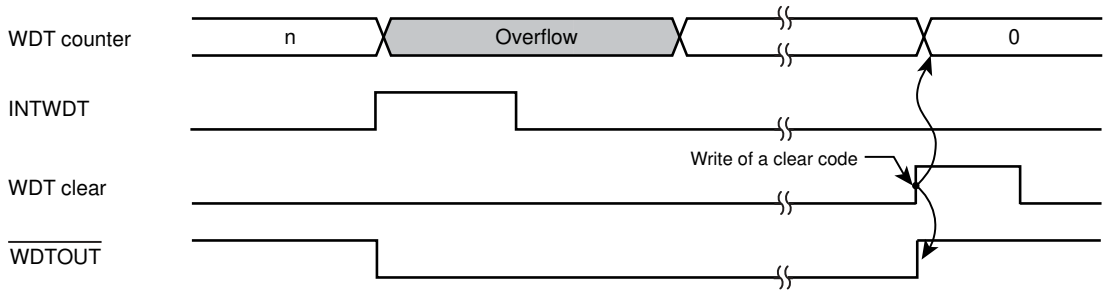


Figure 15-2 INTWDT interrupt generation

15.4.2 Internal reset generation

Figure 15-3 shows the internal reset generation (WDMOD<RESCR>="1").

MCU is reset by the overflow of the binary counter. In this case, reset status continues for 32 states. A clock is initialized so that input clock (fsys) is the same as a high-speed frequency clock (fosc). This means fsys = fosc.

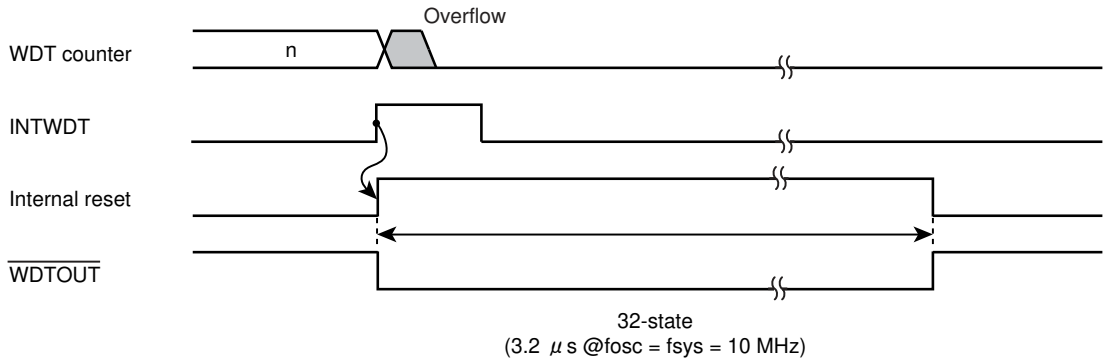


Figure 15-3 Internal reset generation

15.5 Control register

The watchdog timer (WDT) is controlled by two control registers WDMOD and WDCR.

15.5.1 Watchdog Timer Mode Register (WDMOD)

1. Specifying the detection time of the watchdog timer <WDTP[2:0]>.

   Set the watchdog timer detecting time to WDMOD<WDTP[2:0]>. After reset, it is initialized to WDMOD<WDTP[2:0]> = "000".

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 to protect from the error writing by the malfunction, first <WDTE> bit is set to "0", and then the disable code (0xB1) must be written to WDCR register.

   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>

   This register specifies whether WDTOUT is used for internal reset or interrupt. After reset, WDMOD<RESCR> is initialized to "1", the internal reset is generated by the overflow of binary counter.

15.5.2 Watchdog Timer Control Register(WDCR)

This is a register for disabling the watchdog timer function and controlling the clearing function of the binary counter.

15.5.3 Setting example

15.5.3.1 Disabling control

By writing the disable code (0xB1) to this WDCR register after setting WDMOD <WDTE> to "0," the watchdog timer can be disabled and the binary counter can be cleared.

		7	6	5	4	3	2	1	0	
WDMOD	←	0	−	−	−	−	−	−	−	Set <WDTE> to "0".
WDCR	←	1	0	1	1	0	0	0	1	Writes the disable code (0xB1).

15.5.3.2 Enabling control

Set WDMOD <WDTE> to "1".

		7	6	5	4	3	2	1	0	
WDMOD	←	1	−	−	−	−	−	−	−	Set <WDTE> to "1".

15.5.3.3 Watchdog timer clearing control

Writing the clear code (0x4E) to the WDCR register clears the binary counter and it restarts counting.

		7	6	5	4	3	2	1	0	
WDCR	←	0	1	0	0	1	1	1	0	Writes the clear code (0x4E).

15.5.3.4 Detection time of watchdog timer

In the case that 2^{21}/fsys is used, set "011" to WDMOD<WDTP[2:0]>.

		7	6	5	4	3	2	1	0
WDMOD	←	1	0	1	1	−	−	−	−

16. Real Time Clock (RTC)

16.1 Function

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

16.2 Block Diagram

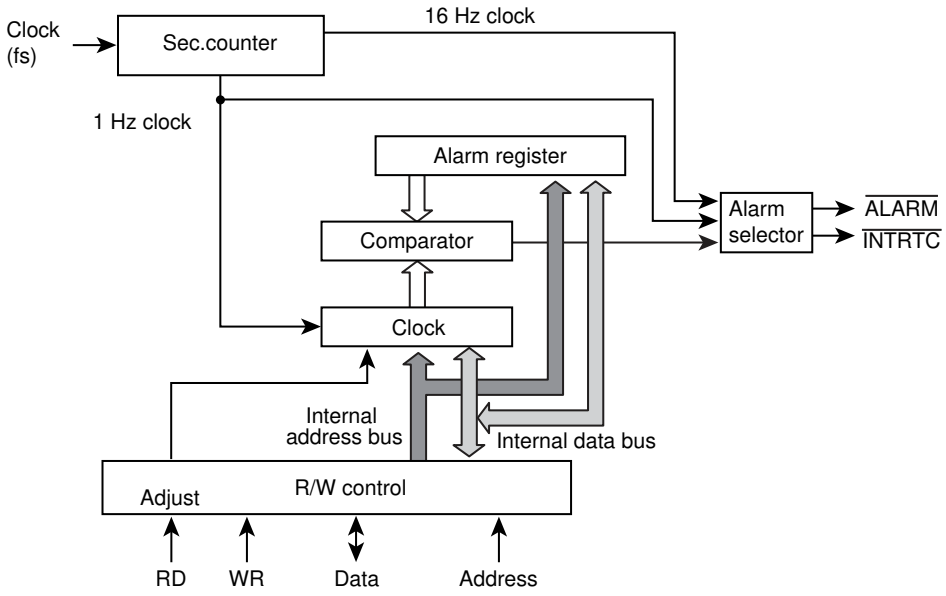


Figure 16-1 Block Diagram

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.**

16.3 Detailed Description Register

16.3.1 Register List

The registers and the addresses related to RTC are shown as below.

RTC has two functions, PAGE0 (clock) and PAGE1 (alarm), which share some parts of registers.

The PAGE can be selected by setting RTCPAGER<PAGE >.

Base Address = 0x4004_0100

Register name		Address(Base+)
Second column register (only PAGE0)	RTCSECR	0x0000
Minute column register	RTCMINR	0x0001
Hour column register	RTCHOURR	0x0002
- (note 1)	-	0x0003
Day of the week column register	RTCDAYR	0x0004
Day column register	RTCDATER	0x0005
Month column register (PAGE0) Selection register of 24-hour,12-hour (PAGE1)	RTCMONTHR	0x0006
Year column register (PAGE0) Leap year register (PAGE1)	RTCYEARR	0x0007
PAGE register	RTCPAGER	0x0008
- (note 1)	-	0x0009
- (note 1)	-	0x000A
- (note 1)	-	0x000B
Reset register	RTCRESTR	0x000C
Reserved	-	0x000D
- (note 1)	-	0x000E
- (note 1)	-	0x000F

Note 1: **"0" is read by reading the address. Writing is disregarded.**
Note 2: **Access to the "Reserved" areas is prohibited.**

16.3.2 Control Register

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 "16.4.3 Entering the Low Power Consumption Mode" for more information.

Table 16-1 PAGE0 (clock function) register

Symbol	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function
RTCSECR	−	40sec.	20sec.	10sec.	8sec.	4sec.	2sec.	1sec.	Second column
RTCMINR	−	40min.	20min.	10min.	8min.	4min.	2min.	1min.	Minute column
RTCHOURR	−	−	20hours PM/AM	10hour	8hour	4hour	2hour	1hours	Hour column
RTCDAYR	−	−	−	−	−	Day of the week			Day of the week column
RTCDATER	−	−	Day20	Day10	Day8	Day4	Day2	Day1	Day column
RTCMONTHR	−	−	−	Oct.	Aug.	Apr.	Feb.	Jan.	Month column
RTCYEARR	year 80	year 40	year20	year 10	year 8	year 4	year 2	year 1	Year column (lower two columns)
RTCPAGER	Interrupt enable	−	−	Adjustment function	Clock enable	Alarm enable	−	PAGE setting	PAGE register
RTCRESTR	1 Hz enable	16 Hz enable	Clock reset	Alarm reset	−	(FD/FY) − (FW) Always write "1"			Reset register

Note: Reading RTCSECR, RTCMINR, RTCHOURR, RTCDAYR, RTCMONTHR, RTCYEARR of PAGE0 captures the current state.

Table 16-2 PAGE1 (alarm function) registers

Symbol	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function
RTCSECR	−	−	−	−	−	−	−	−	−
RTCMINR	−	40min.	20min.	10min.	8min.	4min.	2min.	1min.	Minute column
RTCHOURR	−	−	20hours PM/AM	10hour	8hour	4hour	2hour	1hour	Hour column
RTCDAYR	−	−	−	−	−	Day of the week			Day of the week column
RTCDATER	−	−	Day20	Day10	Day8	Day4	Day2	Day1	Day column
RTCMONTHR	−	−	−	−	−	−	−	24/12	24-hour clock mode
RTCYEARR	−	−	−	−	−	−	Leap-year setting		Leap-year mode
RTCPAGER	Interrupt enable	−	−	Adjustment function	Clock enable	Alarm enable	−	PAGE setting	PAGE register
RTCRESTR	1 Hz Enable	16 Hz Enable	Clock reset	Alarm reset	−	(FD/FY) − (FW) Always write "1"			Reset register

Note 1: **Reading RTCMINR, RTCHOURR, RTCDAYR, RTCMONTHR, RTCYEARR 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.**

Note: **Regarding the Table 16-1 and the Table 16-2,"FD" indicates TMPM330FDFG, "FY" indicates TMPM330FYFG and "FW" indicates TMPM330FWFG.**

16.3.3 Detailed Description of Control Register

16.3.3.1 RTCSECR (Second column register (for PAGE0 only))

	7	6	5	4	3	2	1	0
bit symbol	-	SE						
After reset	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Functon
7	−	R	Read as 0.
6-0	SE	R/W	Setting digit register of second 000_0000 : 00sec. 001_0000 : 10sec. 010_0000 : 20sec. 000_0001 : 01sec. 001_0001 : 11sec. ・ 000_0010 : 02sec. 001_0010 : 12sec. 011_0000 : 30sec. 000_0011 : 03sec. 001_0011 : 13sec. ・ 000_0100 : 04sec. 001_0100 : 14sec. 100_0000 : 40sec. 000_0101 : 05sec. 001_0101 : 15sec. ・ 000_0110 : 06sec. 001_0110 : 16sec. 101_0000 : 50sec. 000_0111 : 07sec. 001_0111 : 17sec. ・ 000_1000 : 08sec. 001_1000 : 18sec. ・ 000_1001 : 09sec. 001_1001 : 19sec. 101_1001 : 59sec.

Note: **The setting other than listed above is prohibited.**

16.3.3.2 RTCMINR (Minute column register (PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	-	MI						
After reset	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Functon
7	−	R	Read as 0.
6-0	MI	R/W	Setting digit register of Minutes. 000_0000 : 00min. 001_0000 : 10min. 010_0000 : 20min. 000_0001 : 01min. 001_0001 : 11min. ・ 000_0010 : 02min. 001_0010 : 12min. 011_0000 : 30min. 000_0011 : 03min. 001_0011 : 13min. ・ 000_0100 : 04min. 001_0100 : 14min. 100_0000 : 40min. 000_0101 : 05min. 001_0101 : 15min. ・ 000_0110 : 06min. 001_0110 : 16min. 101_0000 : 50min. 000_0111 : 07min. 001_0111 : 17min. ・ 000_1000 : 08min. 001_1000 : 18min. ・ 000_1001 : 09min. 001_1001 : 19min. 101_1001 : 59min.

Note: **The setting other than listed above is prohibited.**

16.3.3.3 RTCHOURR (Hour column register(PAGE0/1))

(1) 24-hour clock mode (RTCMONTHR<MO0>= "1")

	7	6	5	4	3	2	1	0
Bit symbol	-	-	HO					
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Functon
7-6	−	R	Read as 0.
5-0	HO	R/W	Setting digit register of Hour. 00_0000 : 0 o'clock 01_0000 : 10 o'clock 10_0000 : 20 o'clock 00_0001 : 1 o'clock 01_0001 : 11 o'clock 10_0001 : 21 o'clock 00_0010 : 2 o'clock 01_0010 : 12 o'clock 10_0010 : 22 o'clock 00_0011 : 3 o'clock 01_0011 : 13 o'clock 10_0011 : 23 o'clock 00_0100 : 4 o'clock 01_0100 : 14 o'clock 00_0101 : 5 o'clock 01_0101 : 15 o'clock 00_0110 : 6 o'clock 01_0110 : 16 o'clock 00_0111 : 7 o'clock 01_0111 : 17 o'clock 00_1000 : 8 o'clock 01_1000 : 18 o'clock 00_1001 : 9 o'clock 01_1001 : 19 o'clock

Note: **The setting other than listed above is prohibited.**

(2) 12-hour clock mode (RTCMONTHR<MO0> = "0")

	7	6	5	4	3	2	1	0
Bit symbol	-	-	HO					
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Functon
7-6	−	R	Read as 0.
5-0	HO	R/W	Setting digit register of Hour. (AM) (PM) 00_0000 : 0 o'clock 10_0000 : 0 o'clock 00_0001 : 1 o'clock 10_0001 : 1 o'clock 00_0010 : 2 o'clock 10_0010 : 2 o'clock 00_0011 : 3 o'clock 10_0011 : 3 o'clock 00_0100 : 4 o'clock 10_0100 : 4 o'clock 00_0101 : 5 o'clock 10_0101 : 5 o'clock 00_0110 : 6 o'clock 10_0110 : 6 o'clock 00_0111 : 7 o'clock 10_0111 : 7 o'clock 00_1000 : 8 o'clock 10_1000 : 8 o'clock 00_1001 : 9 o'clock 10_1001 : 9 o'clock 01_0000 : 10 o'clock 11_0000 : 10 o'clock 01_0001 : 11 o'clock 11_0001 : 11 o'clock

Note: **The setting other than listed above is prohibited.**

16.3.3.4 RTCDAYR (Day of the week column register(PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	WE		
After reset	0	0	0	0	0	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-3	−	R	Read as 0.
2-0	WE	R/W	Setting digit register of day of the week. 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday

Note: **The setting other than listed above is prohibited.**

16.3.3.5 RTCDATER (Day column register (for PAGE0/1 only))

	7	6	5	4	3	2	1	0
Bit symbol	-	-	DA					
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Functon			
7-6	−	R	Read as 0.			
5-0	DA	R/W	Setting digit register of day.			
				01_0000 : 10th day	10_0000 : 20th day	11_0000 : 30th day
			00_0001 : 1st day	01_0001 : 11th day	10_0001 : 21th day	11_0001 : 31th day
			00_0010 : 2nd day	01_0010 : 12th day	10_0010 : 22th day	
			00_0011 : 3rd day	01_0011 : 13th day	10_0011 : 23th day	
			00_0100 : 4th day	01_0100 : 14th day	10_0100 : 24th day	
			00_0101 : 5th day	01_0101 : 15th day	10_0101 : 25th day	
			00_0110 : 6th day	01_0110 : 16th day	10_0110 : 26th day	
			00_0111 : 7th day	01_0111 : 17th day	10_0111 : 27th day	
			00_1000 : 8th day	01_1000 : 18th day	10_1000 : 28th day	
			00_1001 : 9th day	01_1001 : 19th day	10_1001 : 29th day	

Note 1: **The setting other than listed above is prohibited.**
Note 2: **Do not set for non-existent days (e.g.: 30th Feb.).**

16.3.3.6 RTCMONTHR (Month column register (for PAGE0 only))

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	MO				
After reset	0	0	0	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Functon
7-5	−	R	Read as 0.
4-0	MO	R/W	Setting digit register of Month. 0_0001 : January 0_0111 : July 0_0010 : February 0_1000 : August 0_0011 : March 0_1001 : September 0_0100 : April 1_0000 : October 0_0101 : May 1_0001 : November 0_0110 : June 1_0010 : December

Note: **The setting other than listed above is prohibited.**

16.3.3.7 RTCMONTHR (Selection of 24-hour clock or 12-hour clock24(for PAGE1 only))

	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	MO0
After reset	0	0	0	0	0	0	0	Undefined

Bit	Bit Symbol	Type	Function
7-1	−	R	Read as 0.
0	MO0	R/W	0: 12-hour 1: 24-hour

Note: **Do not change the RTCMONTHR<MO0> while the RTC is in operation.**

16.3.3.8 RTCYEARR (Year column register (for PAGE0 only))

	7	6	5	4	3	2	1	0
bit symbol	YE							
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-0	YE	R/W	Setting digit register of Year. 0000_0000 : 00 years 0000_0001 : 01 years 0000_0010 : 02 years 0000_0011 : 03 years 0000_0100 : 04 years 0000_0101 : 05 years 0000_0110 : 06 years 0000_0111 : 07 years 0000_1000 : 08 years 0000_1001 : 09 years 0001_0000 : 10 years ・ 0010_0000 : 20 years ・ 0011_0000 : 30 years ・ 0100_0000 : 40 years ・ 01001_0000 : 50 years ・ 0110_0000 : 60 years ・ 0111_0000 : 70 years ・ 1000_0000 : 80 years ・ 1001_0000 : 90 years ・ ・ 1001_1001 : 99 years

Note: **The setting other than listed above is prohibited.**

16.3.3.9 RTCYEARR (Leap year register (for PAGE1 only))

	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	LEAP	
After reset	0	0	0	0	0	0	Undefined	Undefined

Bit	Bit Symbol	Type	Functon
7-2	−	R	Read as 0.
1-0	LEAP	R/W	00 : leap year 01 : one year after leap year 10 : two years after leap year 11 : three years after leap year

16.3.3.10 RTCPAGER(PAGE register(PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	INTENA	-	-	ADJUST	ENATMR	ENAALM	-	PAGE
After reset	0	0	0	0	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Function
7	INTENA	R/W	INTRTC 0:Disable 1:Enable
6-5	−	R	Read as 0.
4	ADJUST	R/W	[Write] 0: Don't care 1: Sets ADJUST request 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". [Read] 0: ADJUST no request 1: ADJUST requested If "1" is read, it indicates that ADJUST is being executed. If "0" is read, it indicates that the execution is finished.
3	ENATMR	R/W	Clock 0: Disable 1: Enable
2	ENAALM	R/W	ALARM 0: Disable 1: Enable
1	−	R	Read as 0.
0	PAGE	R/W	PAGE selection 0:Selects Page0 1:Selects Page1

Note 1: **A read-modify-write operation cannot be porfomed.**

Note 2: **To set interrupt enable bits to <ENATMR>, <ENAALM> and <INTENA>, you must follow the order specified here. Make sure not to set them at the same time (make sure that there is time lag between interrupt enable and clock/ alarm enable).To change the setting of <ENATMR> and <ENAALM>, <INTENA> must be disabled first.**

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

16.3.3.11 RTCRESTR (Reset register (for PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	-	-	-	-
After reset	1	1	0	0	0	0(FD/FY) 1(FW)	0(FD/FY) 1(FW)	0(FD/FY) 1(FW)

Bit	Bit Symbol	Type	Function
7	DIS1HZ	R/W	1 Hz 0:Enable 1: Disable
6	DIS16HZ	R/W	16 Hz 0: Enable 1: Disable
5	RSTTMR	R/W	[Write] 0: Don't care 1: Sec.counter reset Resets the sec counter. The equest is sampled using low-speed clock. [Read] 0: No reset request 1: RESET requested If "1" is read, it indicates that RESET is being executed. If "0" is read, it indicates that the execution is finished.
4	RSTALM	R/W	0:Don't care 1: Alarm reset 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
3	−	R	Read as 0.
2-0	−	R	FD/FY(Note2) : Read as 0.
		R/W	FW(Note2) : Write "1".

Note 1: A read-modify-write operation cannot be performed.
Note 2: "FD" indicates TMPM330FDFG, "FY" indicates TMPM330FYFG and "FW" indicates TMPM330FWFG.

The setting of <DIS1HZ> and <DIS16MHZ>,RTCPAGER<ENAALM> used for alarm, 1Hz interrupt and 16Hz interrupt is shown as below.

<DIS1HZ>	<DIS16HZ>	RTCPAGER <ENAALM>	Interrupt source signal
1	1	1	Alarm
0	1	0	1 Hz
1	0	0	16 Hz
Others			Interrupt not generated.

16.4 Operational Description

The RTC incorporates a second counter that generates a 1Hz signal from a 32.768 kHz signal.

The second counter operation must be taken into account when using the RTC.

16.4.1 Reading clock data

1. Using 1Hz interrupt

   The 1Hz interrupt is generated being synchronized with counting up of the second 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.

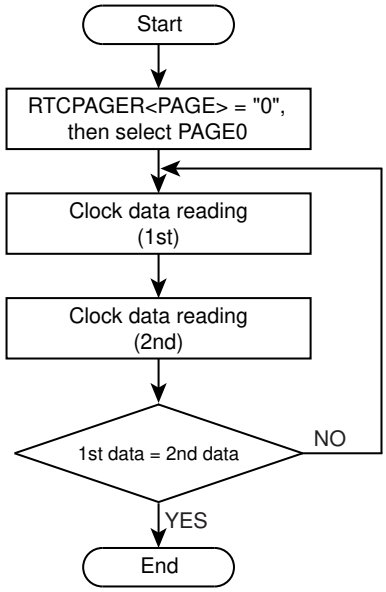


Figure 16-2 Flowchart of the clock data reading

16.4.2 Writing clock data

A carry during writing ruins correct data writing. The following procedure ensures the correct data writing.

1. Using 1 Hz interrupt

   The 1Hz interrupt is generated by being synchronized with counting up of the second 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 second 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.

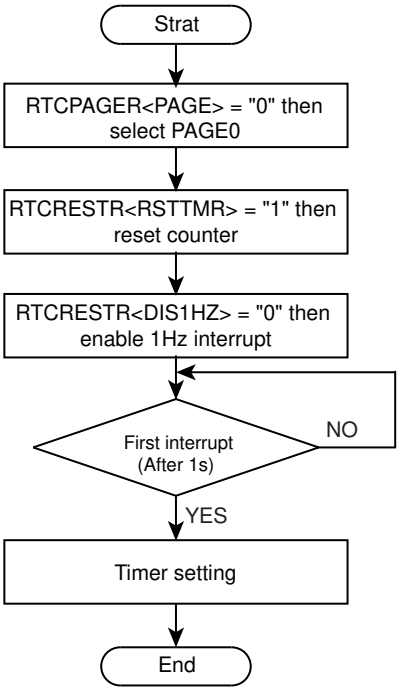


Figure 16-3 Flowchart of the clock data writing

3. Disabling the clock

Writing "0" to RTCPAGER<ENATMR> disables clock operation including a carry.

Stop the clock after the 1Hz-interrupt. The second counter keeps counting.

Set the clock again and enable the clock within one second before next 1Hz-interrupt

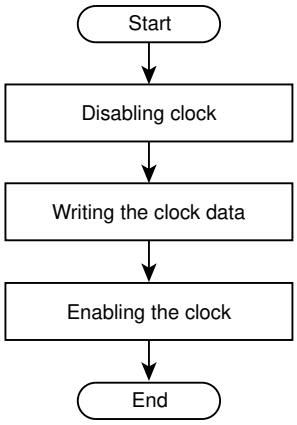


Figure 16-4 Flowchart of the disabling clock

16.4.3 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.

16.5 Alarm 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 $\overline{\text{ALARM}}$ pin.

1. "Low" pulse (when the alarm register corresponds with the clock)
2. 1Hz cycle "Low" pulse
3. 16Hz cycle "Low" pulse

In any cases shown above, the INTRTC outputs one cycle pulse of low-speed clock. It outputs the INTRTC interrupt request simultaneously.

The INTRTC interrupt signal is falling edge triggered.Specify the falling edge as the active state in the CG Interrupt Mode Control Register

16.5.1 "Low" pulse (when the alarm register corresponds with the clock)

"Low" pulse is output to the $\overline{\text{ALARM}}$ 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.

The alarm settings

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 ALARM pin at noon (12:00) 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
RTCDATER	←	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 interrupts

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 fs (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 INTRTC interrupt routine that is generated when the time set for the alarm matches the RTC count.**

16.5.2 1Hz cycle "Low" pulse1 Hz

The RTC outputs a "Low" pulse cycle of low-speed 1Hz clock to the $\overline{\text{ALARM}}$ pin by setting RTCPAGER<INTENA>="1" after setting RTCPAGER<ENAALM>= "0", RTCRESTR<DIS1HZ>= "0" and <DIS16HZ>= "1". It generates an INTRTC interrupt simultaneously.

16.5.3 16Hz cycle "Low" pulse16 Hz

The RTC outputs a "Low" pulse cycle of low-speed 16Hz clock to the $\overline{\text{ALARM}}$ pin by setting RTCPAGER<INTENA>="1" after setting RTCPAGER<ENAALM>= "0", RTCRESTR<DIS1HZ>= "1" and <DIS16HZ>= "0". It generates an INTRTC interrupt simultaneously.

17. Flash Memory Operation

This section describes the hardware configuration and operation of the flash memory.

17.1 Flash Memory

17.1.1 Features

1. Memory capacity

   The TMPM330FDFG/FYFG/FWFG 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 TMPM330FDFG/TMPM330FYFG contain 128 words and the TMPM330FWFG contains 64 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
		128 KB	64 KB	32 KB	16 KB			
TMPM330FDFG	512 KB	3	1	2	−	128	1.28 sec	0.4 sec
TMPM330FYFG	256 KB	1	1	2	−	128	0.64 sec	0.4 sec
TMPM330FWFG	128 KB	−	1	1	2	64	0.64 sec	0.2 sec

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. Programming method

   There are two types of the onboard programming mode for the user to program (rewrite) the device while it is mounted on the user's board:

   - The onboard programming mode

   a. User boot mode

      The user's original rewriting method can be supported.

   b. Single boot mode

      The rewriting method to use serial data transfer (Toshiba's unique method) can be supported.

4. 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.

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

5. Protect/ Security Function

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 the chapter "ROM protection" for details of ROM protection and security function.

Note: **If a password is set to 0xFF (erased data), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.**

17.1.2 Block Diagram of the Flash Memory Section

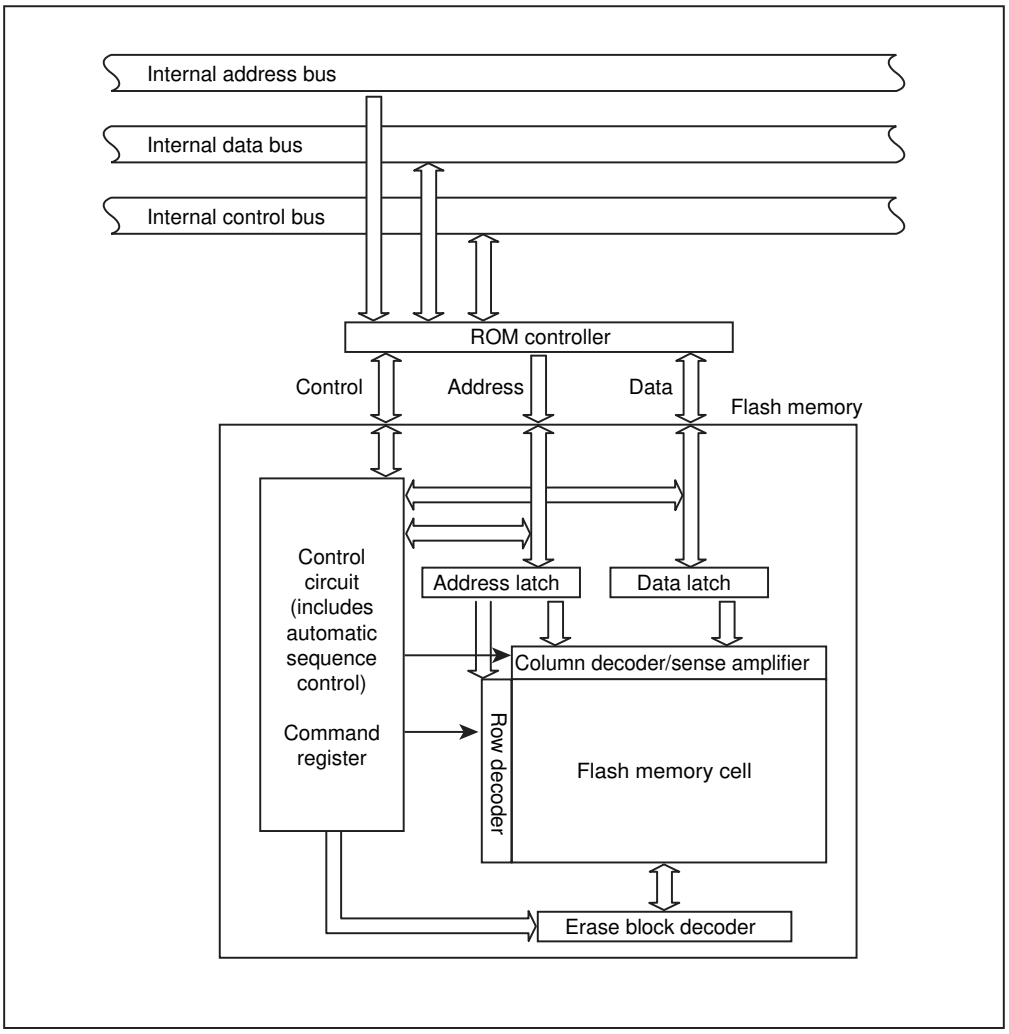


Figure 17-1 Block Diagram of the Flash Memory Section

17.2 Operation Mode

This device has three operation modes including the mode not to use the internal flash memory.

Table 17-1 Operation Modes

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.
Single boot mode		After reset is cleared, it starts up from the internal Boot ROM (Mask ROM). In the Boot ROM, an algorithm to enable flash memory rewriting on the user's set through the serial port of this device is programmed. By connecting to an external host computer through the serial port, the internal flash memory can be programmed by transferring data in accordance with predefined protocols.

Among the flash memory operation modes listed in the above table, the User Boot mode and the Single Boot mode are the programmable modes. These two modes, the User Boot mode and the Single Boot mode, are referred to as "Onboard Programming" modes where onboard rewriting of internal flash memory can be made on the user's card.

Either the Single Chip or Single Boot operation mode can be selected by externally setting the level of the $\overline{\text{BOOT}}$ (PH0) pin while the device is in reset status.

Table 17-2 Operation Mode Setting

Operation mode	Pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$ (PH0)
Single chip mode	0 → 1	1
Single boot mode	0 → 1	0

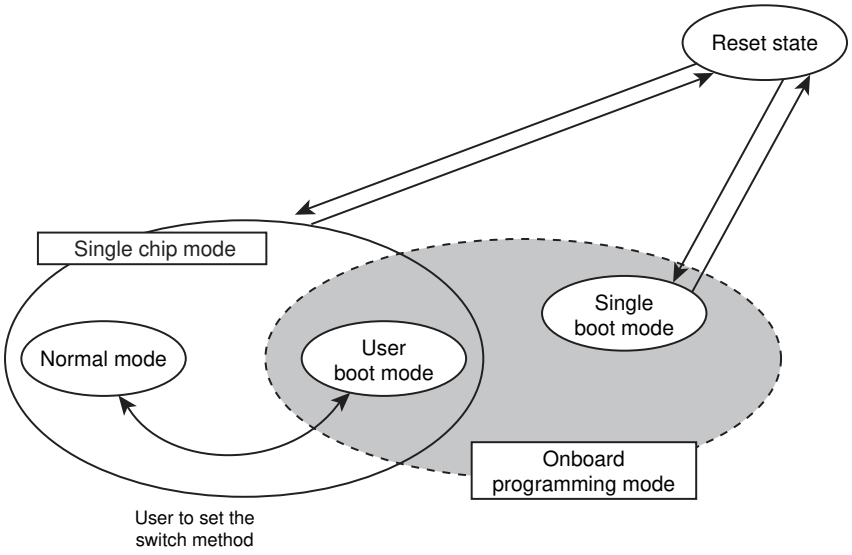


Figure 17-2 Mode Transition Diagram

17.2.1 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 $\overline{\text{RESET}}$ input is held at "0" for a minimum duration of 12 system clocks (0.3 μs with 40MHz 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 $\overline{\text{RESET}}$ inputs upon power on for a minimum duration of 700 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.**

17.2.2 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 TMPM330FDFG/FYFG/FWFG 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. Be sure not to cause any exceptions including a non-maskable while User Boot Mode.

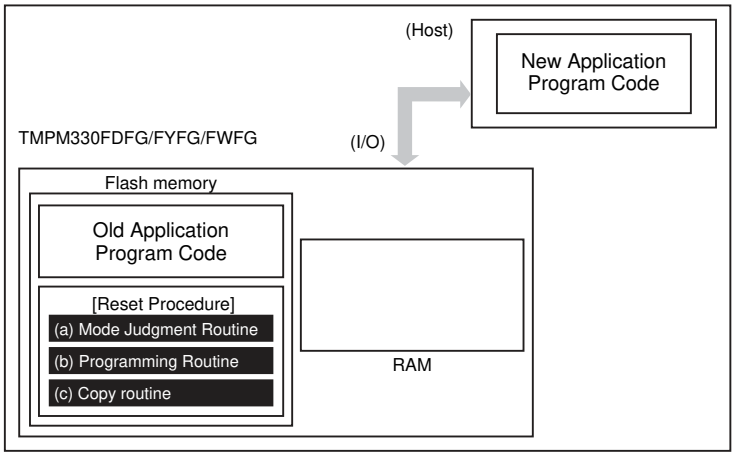
(1-A) and (1-B) are the examples of programming with routines in the internal flash memory and in the external memory. For a detailed description of the erase and program sequence, refer to "17.3 On-board Programming of Flash Memory (Rewrite/Erase)".

17.2.2.1 (1-A) Method 1: Storing a Programming Routine in the Flash Memory

(1) Step-1

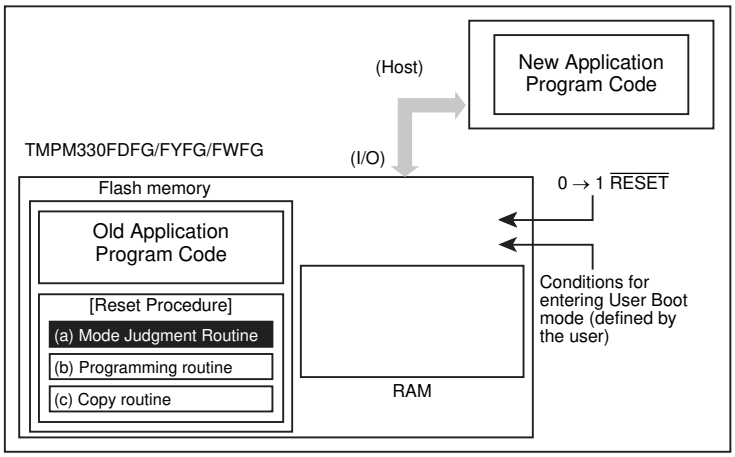
Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM330FDFG/FYFG/FWFG on a printed circuit board, write the following program routines 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 TMPM330FDFG/FYFG/FWFG flash memory to either the TMPM330FDFG/FYFG/FWFG on-chip RAM or external memory device.



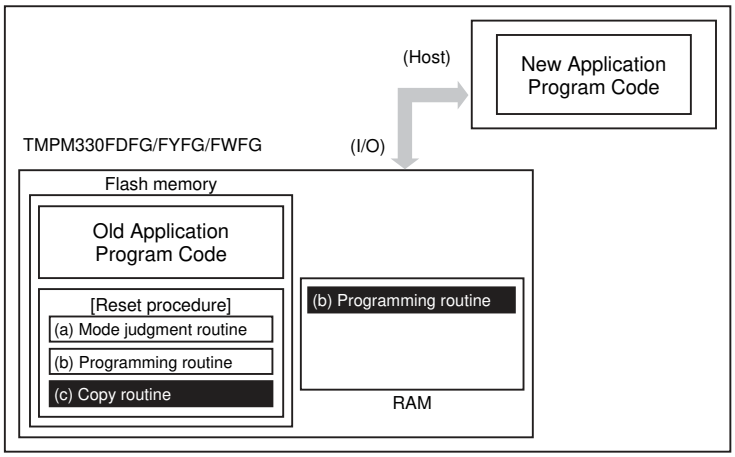
(2) Step-2

After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMPM330FDFG/FYFG/FWFG 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.)



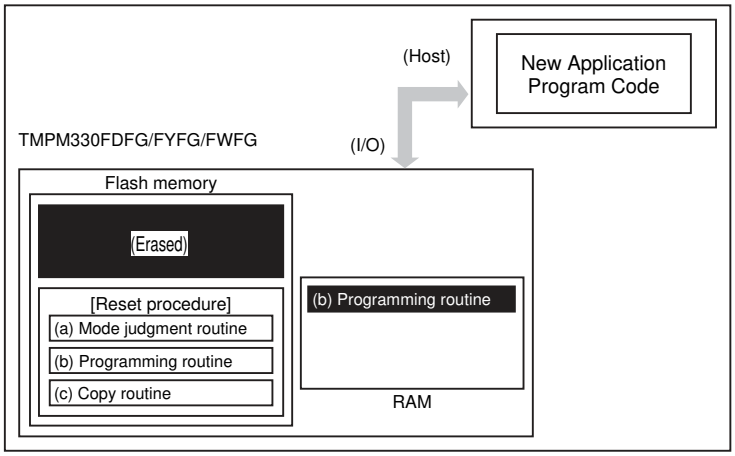
(3) Step-3

Once transition to User Boot mode is occurred, execute the copy routine (c) to copy the flash programming routine (b) to the TMPM330FDFG/FYFG/FWFG on-chip RAM.



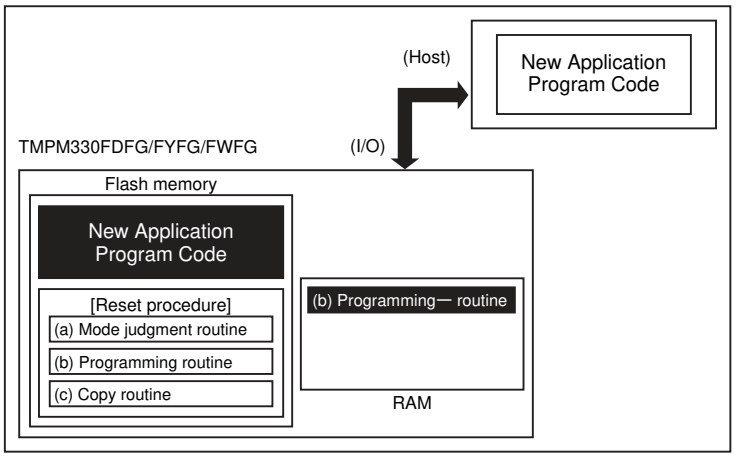
(4) Step-4

Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.



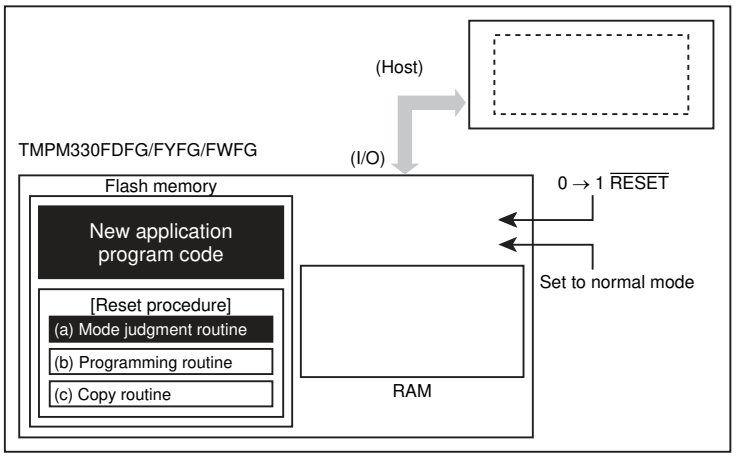
(5) Step-5

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user's program area must be set.



(6) Step-6

Set $\overline{\text{RESET}}$ to "0" to reset the TMPM330FDFG/FYFG/FWFG. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



17.2.2.2 (1-B) Method 2: Transferring a Programming Routine from an External Host

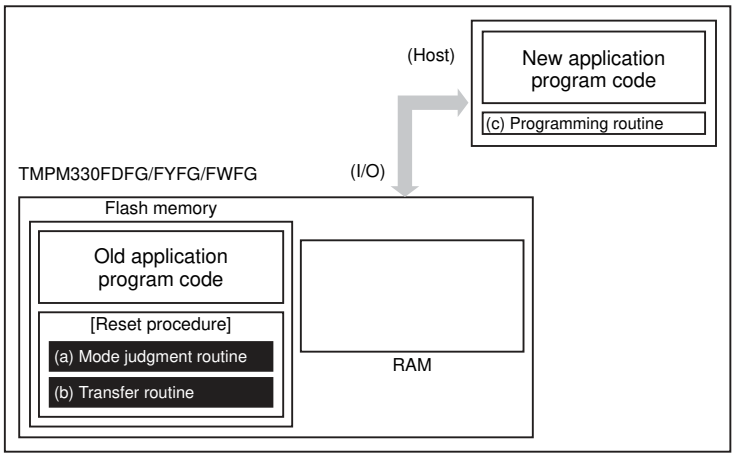
(1) Step-1

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 TMPM330FDFG/FYFG/FWFG 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

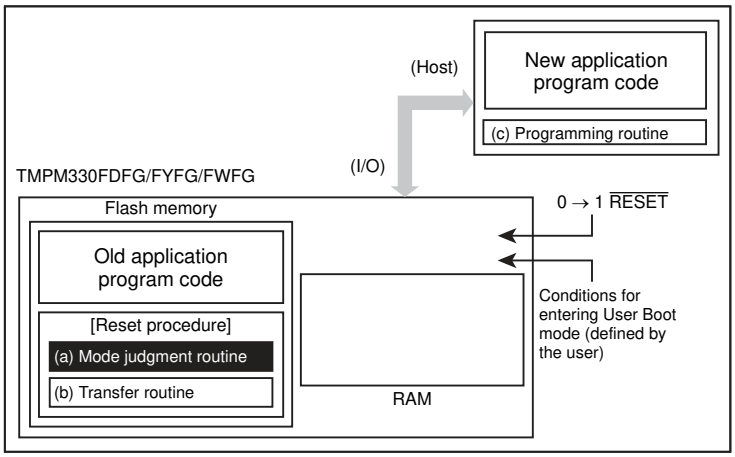
Also, prepare a programming routine shown below on the host controller:

(c) Programming routine: Code to download new program code from an external host controller and re-program the flash memory



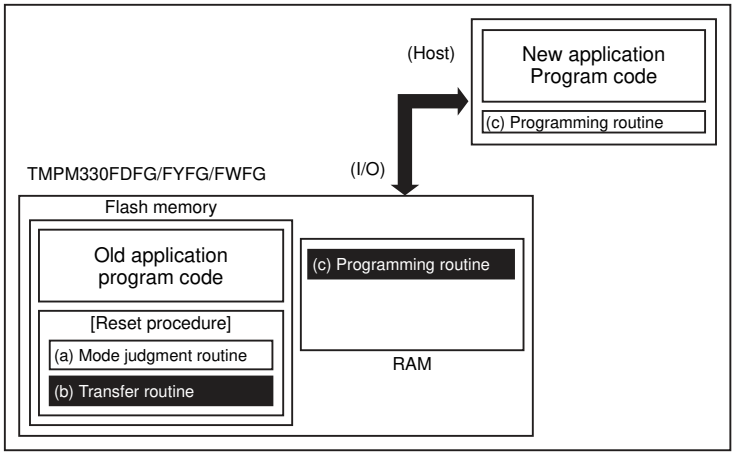
(2) Step-2

After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMPM330FDFG/FYFG/FWFG 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).



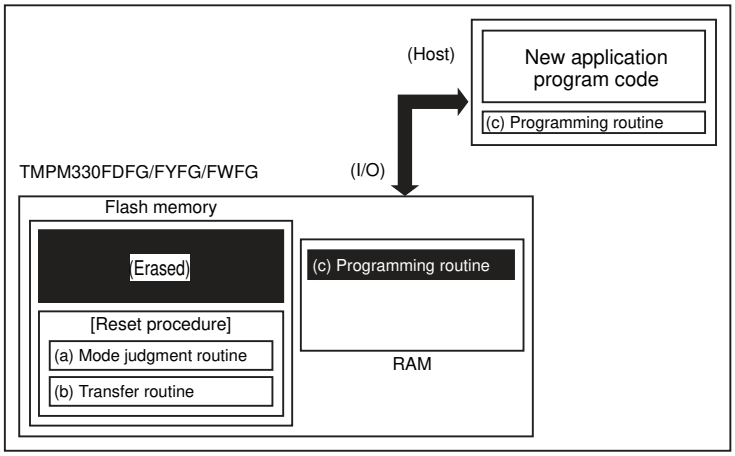
(3) Step-3

Once User Boot mode is entered, execute the transfer routine (b) to download the flash programming routine (c) from the host controller to the TMPM330FDFG/FYFG/FWFG on-chip RAM.



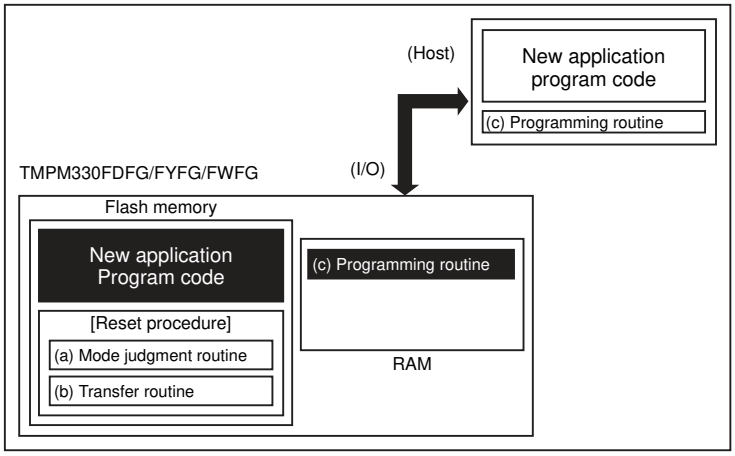
(4) Step-4

Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.



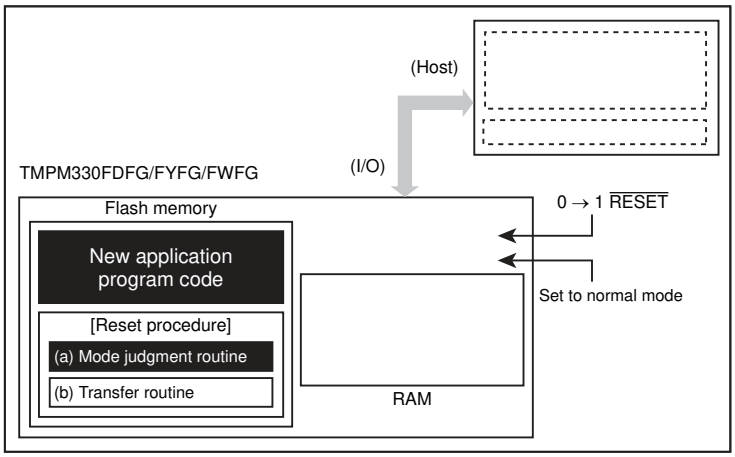
(5) Step-5

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user program area must be set.



(6) Step-6

Set $\overline{\text{RESET}}$ to "0" low to reset the TMPM330FDFG/FYFG/FWFG. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



17.2.3 Single Boot Mode

In Single Boot mode, the flash memory can be re-programmed by using a program contained in the TMPM330FDFG/FYFG/FWFG on-chip boot ROM. This boot ROM is a masked ROM. When Single Boot mode is selected upon reset, the boot ROM is mapped to the address region including the interrupt vector table while the flash memory is mapped to an address region different from it.

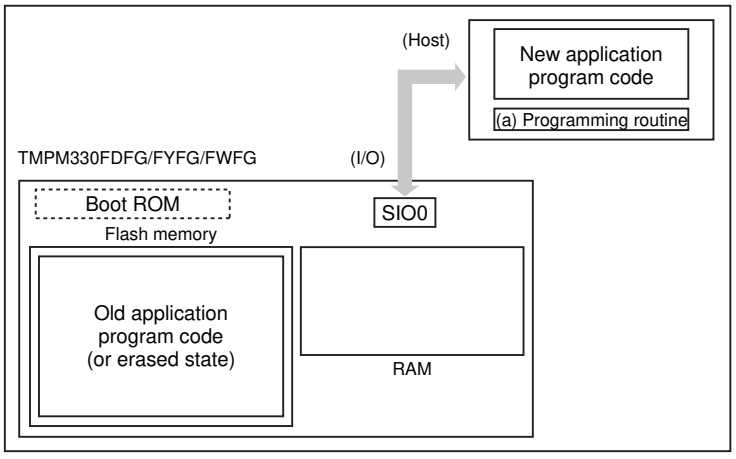
Single Boot mode allows for serial programming of the flash memory. Channel 0 of the SIO (SIO0) of the TMPM330FDFG/FYFG/FWFG is connected to an external host controller. Via this serial link, a programming routine is downloaded from the host controller to the TMPM330FDFG/FYFG/FWFG on-chip RAM. Then, the flash memory is re-programmed by executing the programming routine. The host sends out both commands and programming data to re-program the flash memory. Communications between the SIO0 and the host must follow the protocol described later. To secure the contents of the flash memory, the validity of the application's password is verified before a programming routine is downloaded into the on-chip RAM. If password matching fails, the transfer of a programming routine itself is aborted. As in the case of User Boot mode, all interrupts including the non-maskable interrupt (NMI) must be disabled in Single Boot mode while the flash memory is being erased or programmed. In Single Boot mode, the boot-ROM programs are executed in Normal mode.

Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations.

17.2.3.1 (2-A) Using the Program in the On-Chip Boot ROM

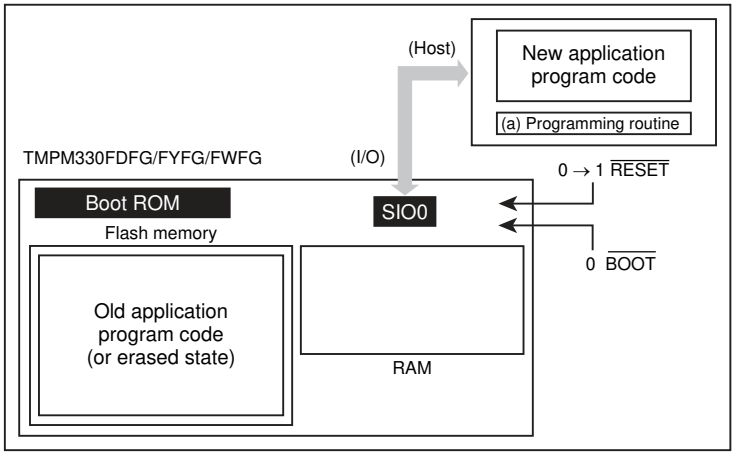
(1) Step-1

The flash block containing the older version of the program code need not be erased before executing the programming routine. Since a programming routine and programming data are transferred via the SIO (SIO0), the SIO0 must be connected to a host controller. Prepare a programming routine (a) on the host controller.



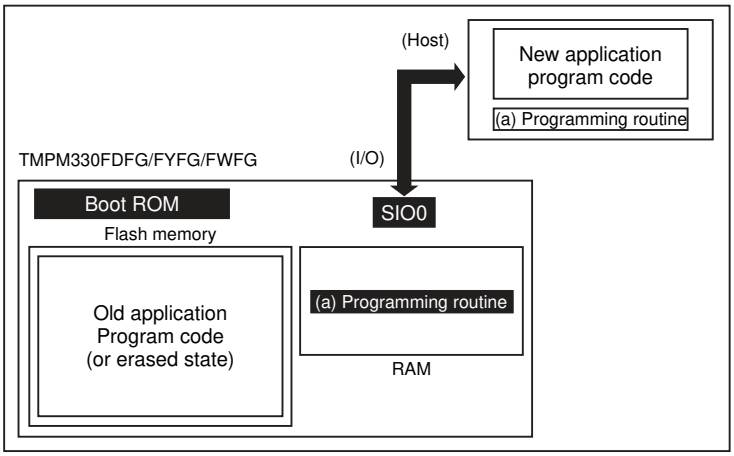
(2) Step-2

Set the $\overline{\text{RESET}}$ pin to "1" to cancel the reset of the TMPM330FDFG/FYFG/FWFG when the $\overline{\text{BOOT}}$ pin has already been set to "0". After reset, CPU reboots from the on-chip boot ROM. The 12-byte password transferred from the host controller via SIO0 is first compared to the contents of the special flash memory locations. (If the flash block has already been erased, the password is 0xFF).



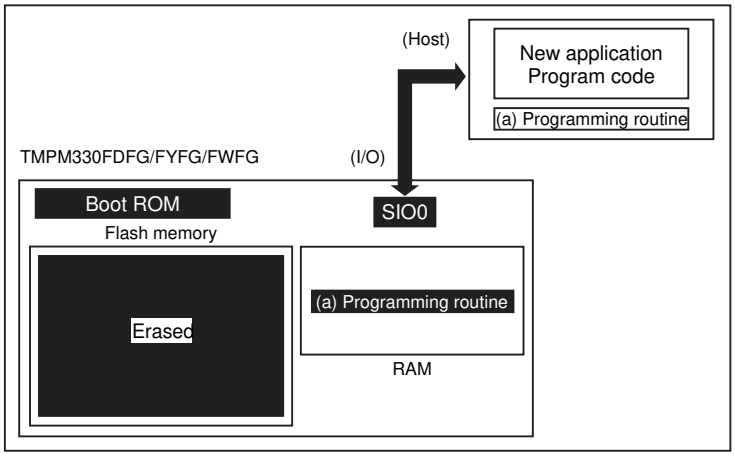
(3) Step-3

If the password was correct, the boot program downloads, via the SIO0, the programming routine (a) from the host controller into the on-chip RAM of the TMPM330FDFG/FYFG/FWFG. The programming routine must be stored in the range from 0x2000_0400 to the end address of RAM.



(4) Step-4

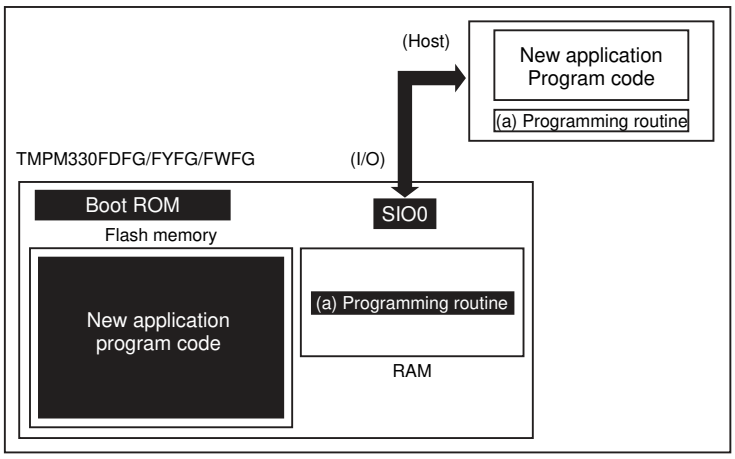
The CPU jumps to the programming routine (a) in the on-chip RAM to erase the flash block containing the old application program code. The Block Erase or Chip Erase command may be used.



(5) Step-5

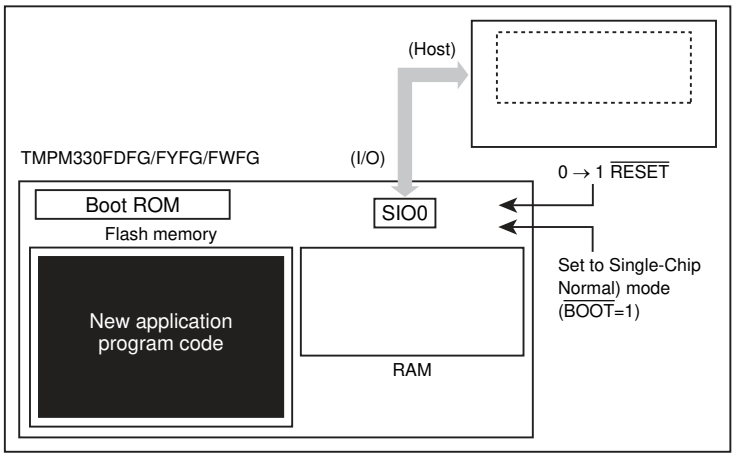
Next, the programming routine (a) downloads new application program code from the host controller and programs it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user's program area must be set.

In the example below, new program code comes from the same host controller via the same SIO0 channel as for the programming routine. However, once the programming routine has begun to execute, it is free to change the transfer path and the source of the transfer. Create board hardware and a programming routine to suit your particular needs.



(6) Step-6

When programming of the flash memory is complete, power off the board and disconnect the cable leading from the host to the target board. Turn on the power again so that the TMPM330FDFG/FYFG/FWFG re-boots in Single-Chip (Normal) mode to execute the new program.



17.2.4 Configuration for Single Boot Mode

To execute the on-board programming, boot the TMPM330FDFG/FYFG/FWFG with Single Boot mode following the configuration shown below.

$\overline{BOOT}$(PH0) = 0
$\overline{RESET}$ = 0 → 1

Set the $\overline{RESET}$ input to "0", and set the each $\overline{BOOT}$ (PH0) pins to values shown above, and then release RESET (high).

17.2.5 Memory Map

Figure 17-3 shows a comparison of the memory maps in Normal and Single Boot modes. In Single Boot mode, the internal flash memory is mapped to 0x3F80_0000 and later addresses, and the Internal boot ROM (Mask ROM) is mapped to 0x0000_0000 through 0x0000_1FFF.

The internal flash memory and RAM addresses of each device are shown below.

Product Name	Flash Size	RAM Size	Flash Address (Single Chip/ Single Boot Mode)	RAM Address
TMPM330FDFG	512 KB	32 KB	0x0000_0000 to 0x0007_FFFF 0x3F80_0000 to 0x3F87_FFFF	0x2000_0000 to 0x2000_7FFF
TMPM330FYFG	256 KB	16 KB	0x0000_0000 to 0x0003_FFFF (0x0007_FF00 to 0x0007_FF80)(Note) 0x3F80_0000 to 0x3F83_FFFF (0x3F87_FF00 to 0x3F87_FF80) (Note)	0x2000_0000 to 0x2000_3FFF
TMPM330FWFG	128 KB	8 KB	0x0000_0000 to 0x0001_FFFF 0x3F80_0000 to 0x3F81_FFFF	0x2000_0000 to 0x2000_1FFF

Note: In addition to 256KB flash area, the TMPM330FYFG provides 128-word data/password area (1 page) for Show Product Information command.

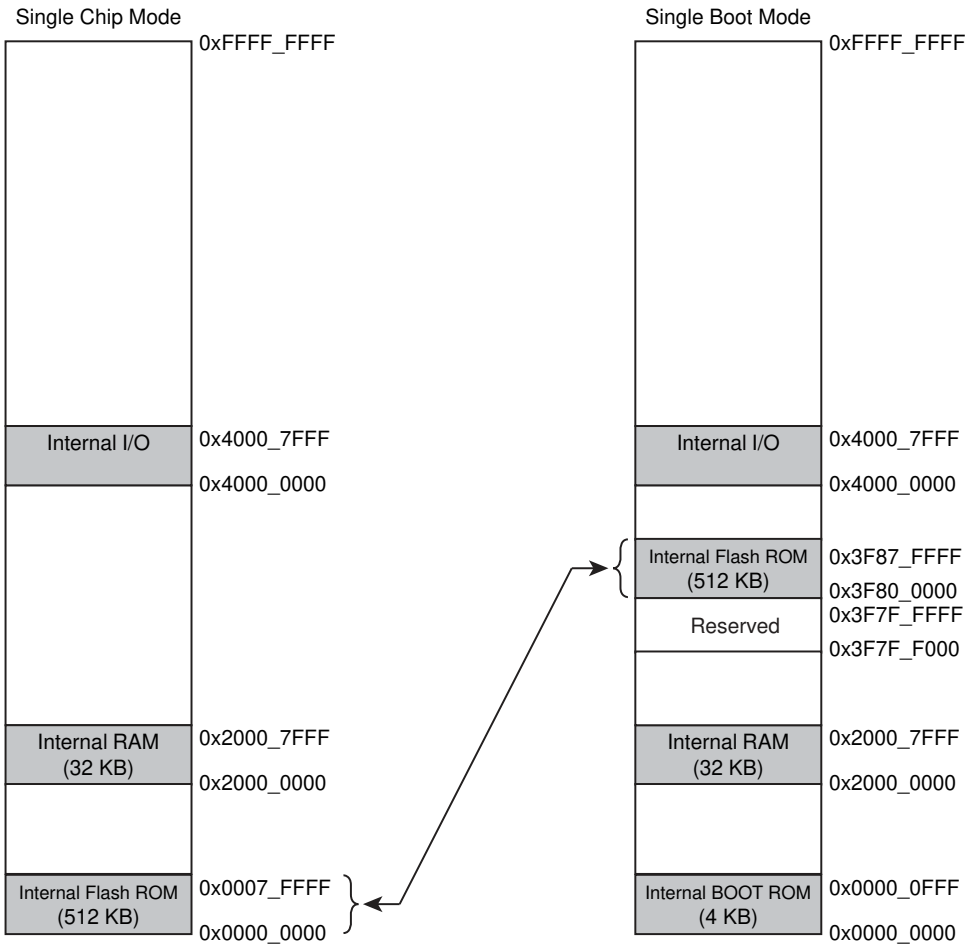


Figure 17-3 Memory Maps for TMPM330FDFG

17.2.6 Interface specification

In Single Boot mode, an SIO channel is used for communications with a programming controller. The same configuration is applied to a communication format on a programming controller to execute the on-board programming. Both UART (asynchronous) and I/O Interface (synchronous) modes are supported. The communication formats are shown below.

- UART communication

  Communication channel : SIO channel 0

  Serial transfer mode : UART (asynchronous), half -duplex, LSB fast

  Data length : 8 bit

  Parity bits : None

  STOP bits : 1 bit

  Baud rate : Arbitrary baud rate

- I/O interface mode

  Communication channel : SIO channel 0

  Serial transfer mode : I/O interface mode, full -duplex, LSB fast

  Synchronization clock (SCLK0) : Input mode

  Handshaking signal : PE4 configured as an output mode

  Baud rate : Arbitrary baud rate

Table 17-3 Required Pin Connections

Pins		Interface	
		UART	I/O Interface Mode
Power supply pins	RVDD3	o	o
	AVDD	o	o
	DVDD3	o	o
	RVSS	o	o
	AVSS	o	o
	DVSS3	o	o
Mode-setting pin	$\overline{\text{BOOT}}$ (PH0)	o	o
Reset pin	$\overline{\text{RESET}}$	o	o
Communication pins	TXD0 (PE0)	o	o
	RXD0 (PE1)	o	o
	SCLK0 (PE2)	×	o (Input mode)
	PE4	×	o(Output mode)

17.2.7 Data Transfer Format

Table 17-4 and Table 17-6 to Table 17-9 illustrate the operation commands and data transfer formats at each operation mode. In conjunction with this section, refer to "17.2.10 Operation of Boot Program".

Table 17-4 Single Boot Mode Commands

Code	Command
0x10	RAM transfer
0x20	Show Flash Memory SUM
0x30	Show Product Information
0x40	Chip and protection bit erase

17.2.8 Restrictions on internal memories

Single Boot Mode places restrictions on the internal RAM and ROM as shown in Table 17-5.

Table 17-5 Restrictions in Single Boot Mode

Memory	Details
Internal RAM	A program contained in the BOOT ROM uses the area, through 0x2000_0000 to 0x2000_03FF as a work area. Store the RAM transfer program from 0x2000_0400 through the end address of RAM.
Internal ROM	The following addresses are assigned for storing software ID information and passwords. Storing program in these addresses is not recommendable. TMPM330FDFG: 0x3F87_FF00 to 0x3F87_FF0F TMPM330FYFG: 0x3F87_FF00 to 0x3F87_FF0F TMPM330FWFG: 0x3F81_FF00 to 0x3F81_FF0F

17.2.9 Transfer Format for Single Boot Mode commands

The following tables shows the transfer format for each Single Boot Mode command. Use this section in conjunction with Chapter "17.2.10 Operation of Boot Program".

17.2.9.1 RAM Transfer

Table 17-6 Transfer Format for the RAM Transfer Command

	Byte	Data Transferred from the Controller to the TMPM330FDFG/FYFG/FWFG	Baud rate	Data Transferred from the TMPM330FDFG/ FYFG/FWFG to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode : 0x86 For I/O Interface mode : 0x30	Desired baud rate (Note 1)	−
	2 byte	−		ACK for the serial operation mode byte ・For UART mode -Normal acknowledge : 0x86 (The boot program aborts if the baud rate can not be set correctly.) ・For I/O Interface mode -Normal acknowledge :0x30
	3 byte	Command code (0x10)		−
	4 byte	−		ACK for the command code byte (Note 2) -Normal acknowledge : 0x10 -Negative acknowledge : 0xX1 -Communication error : 0xX8
	5 byte to 16 byte	Password sequence (12 bytes) 0x3F87_FF04 to 0x3F87_FF0F (FD/FY) 0x3F81_FF04 to 0x3F81_FF0F (FW)		−
	17 byte	Check SUM value for bytes 5 - 16		−
	18 byte	−		ACK for the checksum byte (Note 2) -Normal acknowledge : 0x10 -Negative acknowledge : 0xX1 -Communication error : 0xX8
	19 byte	RAM storage start address 31 to 24		−
	20 byte	RAM storage start address 23 to 16		−
	21 byte	RAM storage start address 15 to 8		−
	22 byte	RAM storage start address 7 to 0		−
	23 byte	RAM storage byte count 15 to 8		−
	24 byte	RAM storage byte count 7 to 0		−
	25 byte	Check SUM value for bytes 19 to 24		−
	26 byte	−		ACK for the checksum byte (Note 2) -Normal acknowledge : 0x10 -Negative acknowledge : 0xX1 -Communication error : 0xX8
	27 byte to m byte	RAM storage data		−
	m + 1 byte	Checksum value for bytes 27 ~ m		−
	m + 2 byte	−		ACK for the checksum byte (Note 2) -Normal acknowledge : 0x10 -Negative acknowledge : 0xX1 -Communication error : 0xX8
RAM	m + 3 byte	−		Jump to RAM storage start address

Note 1: In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.

Note 2: In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.

Note 3: The 19th to 25th bytes must be within the RAM address range from 0x2000_0400 through the end address of RAM.

Note 4: FD/ FY/ FW in the above table denotes the TMPM330FDFG, TMPM330FYFG and TMPM330FWFG respectively.

17.2.9.2 Show Flash Memory SUM

Table 17-7 Transfer Format for the Show Flash Memory SUM Command

	Byte	Data Transferred from the Controller to the TMPM330FDFG/FYFG/FWFG	Baud rate	Data Transferred from the TMPM330FDFG/ FYFG/FWFG to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode : 0x86 For I/O Interface mode: 0x30	Desired baud rate (Note 1)	−
	2 byte	−		ACK for the serial operation mode byte ・For UART mode -Normal acknowledge : 0x86 (The boot program aborts if the baud rate can not be set correctly.) ・For I/O Interface mode -Normal acknowledge : 0x30
	3 byte	Command code (0x20)		−
	4 byte	−		ACK for the command code byte (Note 2) -Normal acknowledge : 0x20 -Negative acknowledge : 0xX1 -Communication error : 0xX8
	5 byte	−		SUM (upper byte)
	6 byte	−		SUM (lower byte)
	7 byte	−		Checksum value for bytes 5 and 6
	8 byte	(Wait for the next command code.)		−

Note 1: **In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.**

Note 2: **In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.**

17.2.9.3 Transfer Format for the Show Product Information

Table 17-8 Transfer Format for the Show Product Information Command

	Byte	Data Transferred from the Controller to the TMPM330FDFG/FYFG/FWFG	Baud rate	Data Transferred from the TMPM330FDFG/FYFG/FWFG to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode : 0x86 For I/O Interface mode: 0x30	Desired baud rate (Note 1)	−
	2 byte	−		ACK for the serial operation mode byte ・For UART mode -Normal acknowledge : 0x86 (The boot program aborts if the baud rate can not be set correctly.) ・For I/O Interface mode -Normal acknowledge : 0x30
	3 byte	Command code (0x30)		−
	4 byte	−		ACK for the command code byte (Note 2) -Normal acknowledge : 0x30 -Negative acknowledge : 0xX1 -Communication error : 0xX8
	5 byte	−		Flash memory data at address 0x3F87_FF00 (FD/ FY) 0x3F81_FF00 (FW)
	6 byte	−		Flash memory data at address 0x3F87_FF01 (FD/ FY) 0x3F81_FF01 (FW)
	7 byte	−		Flash memory data at address 0x3F87_FF02 (FD/ FY) 0x3F81_FF02 (FW)
	8 byte	−		Flash memory data at address 0x3F87_FF03 (FD/ FY) 0x3F81_FF03 (FW)
	9 byte to 20 byte	−		Product name (12-byte ASCII code) From the 9th byte: TMPM330FD_ _' (FD/ FY) TMPM330FW_ _' (FW)
	21 byte to 24 byte	−		Password comparison start address (4 bytes) From the 21st byte: 0x04 , 0xFF, 0x87, 0x3F (FD/FY) 0x04 , 0xFF, 0x81, 0x3F (FW)
	25 byte to 28 byte	−		RAM start address (4 bytes) From the 25th byte: 0x00, 0x00, 0x00, 0x20 (FD/FY/FW)
	29 byte to 32 byte	−		Dummy data (4 bytes) From the 29th byte: 0x00, 0x00, 0x00, 0x00 (FD/FY/FW)
	33 byte to 36 byte	−		RAM end address (4 bytes) From the 33rd byte: 0xFF, 0x7F, 0x00, 0x20 (FD/FY) (Note 4) 0xFF, 0x1F, 0x00, 0x20 (FW)
	37 byte to 40 byte	−		Dummy data (4 bytes) From the 37th byte: 0x00, 0x00, 0x00, 0x00 (FD/FY/FW)
	41 byte to 44 byte	−		Dummy data (4 bytes) From the 41st byte: 0x00, 0x00, 0x00, 0x00 (FD/FY/FW)

Table 17-8 Transfer Format for the Show Product Information Command

	Byte	Data Transferred from the Controller to the TMPM330FDFG/FYFG/FWFG	Baud rate	Data Transferred from the TMPM330FDFG/FYFG/FWFG to the Controller
	45 byte to 46 byte	−		Dummy data or Fuse information (2 bytes) From the 45th byte: 0x00, 0x00 (FD/FY/FW)
	47 byte to 50 byte	−		Flash memory start address (4 bytes) From the 47th byte: 0x00, 0x00, 0x80, 0x3F (FD/FY/FW)
	51 byte to 54 byte	−		Flash memory end address (4 bytes) From the 51st byte: 0xFF, 0xFF, 0x87, 0x3F (FD/FY) (Note 5) 0xFF, 0xFF, 0x81, 0x3F (FW)
	55 byte to 56 byte	−		Flash memory block count (2 bytes) From the 55th byte: 0x06, 0x00 (FD/FY) 0x04, 0x00 (FW)
	57 byte to 60 byte	−		Start address of a group of the same-size (16K) flash blocks (4 bytes) From 57th byte: 0x00, 0x00, 0x00, 0x00 (FD/FY) 0x00, 0x00, 0x80, 0x3F (FW)
	61 byte to 64 byte	−		Size (in halfwords) of the same-size (16K) flash blocks (4 bytes) From 61st byte: 0x00, 0x20, 0x00, 0x00 (FD/FY/FW)
	65 byte	−		Number of flash blocks of the same size (16K) (1 byte) 0x00 (FD/FY) 0x02 (FW)
	66 byte to 69 byte	−		Start address of a group of the same-size (32K) flash blocks (4 bytes) From 66th byte: 0x00, 0x00, 0x80, 0x3F (FD/FY) 0x00, 0x80, 0x80, 0x3F (FW)
	70 byte to 73 byte	−		Size (in halfwords) of the same-size (32K) flash blocks (4 bytes) From 70th byte: 0x00, 0x40, 0x00, 0x00 (FD/FY/FW)
	74 byte	−		Number of flash blocks of the same size (32K) (1 byte) 0x02 (FD/FY) 0x01 (FW)
	75 byte to 78 byte	−		Start address of a group of the same-size (64K) flash blocks (4 bytes) From 75th byte: 0x00, 0x00, 0x81, 0x3F (FD/FY/FW)
	79 byte to 82 byte	−		Size (in halfwords) of the same-size (64K) flash blocks (4 bytes) From 79th byte: 0x00, 0x80, 0x00, 0x00 (FD/FY/FW)

Table 17-8 Transfer Format for the Show Product Information Command

	Byte	Data Transferred from the Controller to the TMPM330FDFG/FYFG/FWFG	Baud rate	Data Transferred from the TMPM330FDFG/FYFG/ FWFG to the Controller
	83 byte	−		Number of flash blocks of the same size (64K) (1 byte) 0x01 (FD/FY/FW)
	84 byte to 87 byte	−		Start address of a group of the same-size (128K) flash blocks (4 bytes) From 84th byte: 0x00, 0x00, 0x82, 0x3F (FD/FY) 0x00, 0x00, 0x00, 0x00 (FW)
	88 byte to 91 byte	−		Size (in halfwords) of the same-size (128K) flash blocks (4 bytes) From 88th byte: 0x00, 0x00, 0x01, 0x00 (FD/FY/FW)
	92 byte	−		Number of flash blocks of the same size (128K) (1 byte) 0x03 (FD/FY) (Note 6) 0x00 (FW)
	93 byte	−		Checksum value for bytes 5 ~ 92
	94 byte	(Wait for the next command code.)		−

Note 1: In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.

Note 2: In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.

Note 3: FD/ FY/ FW in the above table denotes the TMPM330FDFG, TMPM330FYFG and TMPM330FWFG respectively.

Note 4: The RAM actual end address of the TMPM330FYFG is 0x2000_3FFF.

Note 5: The flash memory actual end address of the TMPM330FYFG is 0x3F83_FFFF. In addition to 256KB flash area, the TMPM330FYFG provides 128-word data/ password area (0x3F87_FF00 to 0x3F87_FF80, 1 page) for Show Product Information command.

Note 6: The actual number of the TMPM330FYFG flash blocks is one.

17.2.9.4 Chip Erase and Protect Bit Erase

Table 17-9 Transfer Format for the Chip and Protection Bit Erase Command

	Byte	Data Transferred from the Controller to the TMPM330FDFG/FYFG/FWFG	Baud rate	Data Transferred from the TMPM330FDFG/ FYFG/FWFG to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode : 0x86 For I/O Interface mode : 0x30	Desired baud rate (Note 1)	−
	2 byte	−		ACK for the serial operation mode byte ・For UART mode -Normal acknowledge : 0x86 ・For I/O Interface mode -Normal acknowledge : 0x30 (The boot program aborts if the baud rate can not be set correctly.)
	3 byte	Command code (0x40)		−
	4 byte	−		ACK for the command code byte (Note 2) -Normal acknowledge : 0x40 -Negative acknowledge : 0xX1 -Communication error : 0xX8
	5 byte	Chip erase command code (0x54)		−
	6 byte	−		ACK for the command code byte (Note 2) -Normal acknowledge : 0x54 -Negative acknowledge : 0xX1 -Communication error : 0xX8
	7 byte	−		ACK for the chip erase command code byte -Normal acknowledge : 0x4F -Negative acknowledge : 0x4C
	8 byte	(Wait for the next command code.)		−

Note 1: **In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.**

Note 2: **In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.**

17.2.10 Operation of Boot Program

When Single Boot mode is selected, the boot program is automatically executed on startup. The boot program offers these four commands, of which the details are provided on the following subsections.

1. RAM Transfer command

   The RAM Transfer command stores program code transferred from a host controller to the on-chip RAM and executes the program once the transfer is successfully completed. The user program RAM space can be assigned to the range from 0x2000_0400 to the end address of RAM, whereas the boot program area (0x2000_0000 ~ 0x2000_03FF) is unavailable. The user program starts at the assigned RAM address.

   The RAM Transfer command can be used to download a flash programming routine of your own; this provides the ability to control on-board programming of the flash memory in a unique manner. The programming routine must utilize the flash memory command sequences described in Section 17.3. Before initiating a transfer, the RAM Transfer command verifies a password sequence coming from the controller against that stored in the flash memory.

Note: **If a password is set to 0xFF (erased data), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.**

2. Show Flash Memory SUM command

   The Show Flash Memory SUM command adds the entire contents of the flash memory together. The boot program does not provide a command to read out the contents of the flash memory. Instead, the Flash Memory SUM command can be used for software revision management.

3. Show Product Information command

   The Show Product Information command provides the product name, on-chip memory configuration and the like. This command also reads out the contents of the flash memory locations at addresses shown below. In addition to the Show Flash Memory Sum command, these locations can be used for software revision management.

Product name	Area
TMPM330FDFG TMPM330FYFG	0x3F87_FF00 to 0x3F87_FF03
TMPM330FWFG	0x3F81_FF00 to 0x3F81_FF03

4. Chip and Protection Bit Erase command

   This command erases the entire area of the flash memory automatically without verifying a password. All the blocks in the memory cell and their protection conditions are erased even when any of the blocks are prohibited from writing and erasing. When the command is completed, the FCSECBIT <SECBIT> bit is set to "1". This command serves to recover boot programming operation when a user forgets the password. Therefore password verification is not executed.

17.2.10.1 RAM Transfer Command

See Table 17-6 for the transfer format of this command.

1. The 1st byte specifies which one of the two serial operation modes is used. For a detailed description of how the serial operation mode is determined, see "17.2.10.6 Determination of a Serial Operation Mode" described later. If it is determined as UART mode, the boot program then checks if the SIO0 is programmable to the baud rate at which the 1st byte was transferred. During the first-byte interval, the RXE bit in the SC0MOD register is cleared.
   - To communicate in UART mode

     Send, from the controller to the target board, 0x86 in UART data format at the desired baud rate. If the serial operation mode is determined as UART, then the boot program checks if the SIO0 can be programmed to the baud rate at which the first byte was transferred. If that baud rate is not possible, the boot program aborts, disabling any subsequent communications.
   - To communicate in I/O Interface mode

     Send, from the controller to the target board, 0x30 in I/O Interface data format at 1/16 of the desired baud rate. Also send the 2nd byte at the same baud rate. Then send all subsequent bytes at a rate equal to the desired baud rate.

     In I/O Interface mode, the CPU sees the serial receive pin as if it were a general input port in monitoring its logic transitions. If the baud rate of the incoming data is high or the chip's operating frequency is high, the CPU may not be able to keep up with the speed of logic transitions. To prevent such situations, the 1st and 2nd bytes must be transferred at 1/16 of the desired baud rate; then the boot program calculates 16 times that as the desired baud rate. When the serial operation mode is determined as I/O Interface mode, the SIO0 is configured for SCLK Input mode. Beginning with the third byte, the controller must ensure that its AC timing restrictions are satisfied at the selected baud rate. In the case of I/O Interface mode, the boot program does not check the receive error flag; thus there is no such thing as error acknowledge (bit 3, 0x08).
2. The 2nd byte, transmitted from the target board to the controller, is an acknowledge response to the 1st byte. The boot program echoes back the first byte: 0x86 for UART mode and 0x30 for I/O Interface mode.
   - UART mode

     If the SIO0 can be programmed to the baud rate at which the 1st byte was transferred, the boot program programs the SC0BRCR and sends back 0x86 to the controller as an acknowledge. If the SIO0 is not programmable at that baud rate, the boot program simply aborts with no error indication. Following the 1st byte, the controller should allow for a time-out period of five seconds. If it does not receive 0x86 within the allowed time-out period, the controller should give up the communication. The boot program sets the RXE bit in the SC0MOD0 register to enable reception ("1") before loading the SIO transmit buffer with 0x86.
   - I/O Interface mode

     The boot program programs the SC0MOD0 and SC0CR registers to configure the SIO0 in I/O Interface mode (clocked by the rising edge of SCLK0), writes 0x30 to the SC0BUF. Then, the SIO0 waits for the SCLK0 signal to come from the controller. Following the transmission of the 1st byte, the controller should send the SCLK clock to the target board after a certain idle time (several microseconds). This must be done at 1/16 the desire baud rate. If the 2nd byte, which is from the target board to the controller, is 0x30, then the controller should take it as a go-ahead. The controller must then deliver the 3rd byte to the target board at a rate equal to the desired baud rate. The boot program sets the RXE bit in the SC0MOD register to enable reception before loading the SIO transmit buffer with 0x30.
3. The 3rd byte transmitted from the controller to the target board is a command. The code for the RAM Transfer command is 0x10.
4. The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the state

in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

If the 3rd byte is equal to any of the command codes listed in Table 17-4, the boot program echoes it back to the controller. When the RAM Transfer command was received, the boot program echoes back a value of 0x10 and then branches to the RAM Transfer routine. Once this branch is taken, password verification is done. Password verification is detailed in a later section "Password". If the 3rd byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

5. The 5th to 16th bytes transmitted from the controller to the target board, are a 12-byte password. Each byte is compared to the contents of following addresses in the flash memory. The verification is started with the 5th byte and the smallest address in the designated area. If the password verification fails, the RAM Transfer routine sets the password error flag.

Product name	Area
TMPM330FDFG TMPM330FYFG	0x3F87_FF04 to 0x3F87_FF0F
TMPM330FWFG	0x3F81_FF04 to 0x3F81_FF0F

6. The 17th byte is a checksum value for the password sequence (5th to 16th bytes). To calculate the checksum value for the 12-byte password, add the 12 bytes together, drop the carries and take the two’s complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in a later section "Checksum Calculation".

7. The 18th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th to 17th bytes. First, the RAM Transfer routine checks for a receive error in the 5th to 17th bytes. If there was a receive error, the boot program sends back 0x18 (bit 3) and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

   Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 5th to 16th bytes must result in 0x00 (with the carry dropped). If it is not 0x00, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 0x11 to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

   Finally, the RAM Transfer routine examines the result of the password verification. The following two cases are treated as a password error. In these cases, the RAM Transfer routine sends back 0x11 (bit 0) to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

   - Irrespective of the result of the password comparison, all the 12 bytes of a password in the flash memory are the same value other than 0xFF.
   - Not the entire password bytes transmitted from the controller matched those contained in the flash memory.

   When all the above verification has been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

8. The 19th to 22nd bytes, transmitted from the controller the target board, indicate the start address of the RAM region where subsequent data (e.g., a flash programming routine) should be stored. The 19th byte corresponds to bits 31.24 of the address and the 22nd byte corresponds to bits 7.0 of the address.

9. The 23rd and 24th bytes, transmitted from the controller to the target board, indicate the number of bytes that will be transferred from the controller to be stored in the RAM. The 23rd byte corresponds to bits 15.8 of the number of bytes to be transferred, and the 24th byte corresponds to bits 7.0 of the number of bytes.

10. The 25th byte is a checksum value for the 19th to 24th bytes. To calculate the checksum value, add all these bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in a later section "Checksum Calculation".

11. The 26th byte, transmitted from the target board to the controller, is an acknowledge response to the 19th to 25th bytes of data. First, the RAM Transfer routine checks for a receive error in the 19th to 25th bytes. If there was a receive error, the RAM Transfer routine sends back 0x18 and returns to the command wait state (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

    Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 19th to 25th bytes must result in 0x00 (with the carry dropped). If it is not 0x00, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 0x11 to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

    - The RAM storage start address must be within the range of 0x2000_0400 to the end address of RAM.

    When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

12. The 27th to mth bytes from the controller are stored in the on-chip RAM of the TMPM330FDFG/FYFG/FWFG. Storage begins at the address specified by the 19th.22nd bytes and continues for the number of bytes specified by the 23rd.24th bytes.

13. The (m+1) th byte is a checksum value. To calculate the checksum value, add the 27th to mth bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in a later section "Checksum Calculation".

14. The (m+2) th byte is a acknowledge response to the 27th to (m+1) th bytes. First, the RAM Transfer routine checks for a receive error in the 27th to (m+1) th bytes. If there was a receive error, the RAM Transfer routine sends back 0x18 (bit 3) and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

    Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 27th to (m+1) th bytes must result in 0x00 (with the carry dropped). If it is not 0x00, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 0x11 (bit 0) to the controller and returns to the command wait state (i.e., the 3rd byte) again. When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

15. If the (m+2) th byte was a normal acknowledge response, a branch is made to the address specified by the 19th to 22nd bytes.

17.2.10.2 Show Flash Memory SUM Command

See Table 17-7 for the transfer format of this command.

1. The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

2. The 3rd byte, which the target board receives from the controller, is a command. The code for the Show Flash Memory Sum command is 0x20.

3. The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

   If the 3rd byte is equal to any of the command codes listed in Table 17-4, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was received, the boot program echoes back a value of 0x20 and then branches to the Show Flash Memory Sum routine. If the 3rd byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the command wait state (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

4. The Show Flash Memory Sum routine adds all the bytes of the flash memory together. The 5th and 6th bytes, transmitted from the target board to the controller, indicate the upper and lower bytes of this total sum, respectively. For details on sum calculation, see a later section "17.2.10.8 Calculation of the Show Flash Memory Sum Command".

5. The 7th byte is a checksum value for the 5th and 6th bytes. To calculate the checksum value, add the 5th and 6th bytes together, drop the carry and take the two’s complement of the sum. Transmit this checksum value from the controller to the target board.

6. The 8th byte is the next command code.

17.2.10.3 Show Product Information Command

See Table 17-8 for the transfer format of this command.

1. The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

2. The 3rd byte, which the target board receives from the controller, is a command. The code for the Show Product Information command is 0x30.

3. The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

   If the 3rd byte is equal to any of the command codes listed in Table 17-4, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was received, the boot program echoes back a value of 0x30 and then branches to the Show Flash Memory Sum routine. If the 3rd byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

4. The 5th to 8th bytes, transmitted from the target board to the controller, are the data read from addresses shown below in the flash memory. Software version management is possible by storing a software ID in these locations.

Product name	Area
TMPM330FDFG TMPM330FYFG	0x3F87_FF00 to 0x3F87_FF03
TMPM330FWFG	0x3F81_FF00 to 0x3F81_FF03

5. The 9th to 20th bytes, transmitted from the target board to the controller, indicate the product name as shown below (where [ ] is a space) in ASCII code.

Product name	Code
TMPM330FDFG TMPM330FYFG	T, M, P, M, 3, 3, 0, F, D, _, [ ], _
TMPM330FWFG	T, M, P, M, 3, 3, 0, F, W, _, [ ], _

6. The 21st to 24th bytes, transmitted from the target board to the controller, indicate the start address of the flash memory area containing the password. Each product has own start address shown below.

Product name	Address
TMPM330FDFG TMPM330FYFG	0x04, 0xFF,0x 87, 0x3F
TMPM330FWFG	0x04, 0xFF,0x 81, 0x3F

7. The 25th to 28th bytes, transmitted from the target board to the controller, indicate the start address of the on-chip RAM, i.e., 0x00, 0x00, 0x00, 0x20.

8. The 29th to 32nd bytes, transmitted from the target board to the controller, are dummy data (0x00, 0x00, 0x00 and 0x00).

9. The 33rd to 36th bytes, transmitted from the target board to the controller, indicate the end address of the on-chip RAM. Each product has own end address shown below.

Product name	Address
TMPM330FDFG TMPM330FYFG	0xFF, 0x7F, 0x00, 0x20
TMPM330FWFG	0xFF, 0x1F, 0x00, 0x20

Note: **The RAM actual end address of the TMPM330FYFG is 0x2000_3FFF.**

10. The 37th to 40th bytes, transmitted from the target board to the controller, are 0x00, 0x00, 0x00 and 0x00. The 41st to 44th bytes, transmitted from the target board to the controller, are 0x00, 0x00, 0x00 and 0x00.

11. The 45th and 46th bytes transmitted are 0x00, 0x00.

12. The 47th to 50th bytes, transmitted from the target board to the controller, indicate the start address of the on-chip flash memory, are 0x00, 0x00, 0x80, and 0x3F.

13. The 51st to 54th bytes, transmitted from the target board to the controller, indicate the end address of the on-chip flash memory. Each product has own end address shown below.

Product name	Address
TMPM330FDFG TMPM330FYFG	0xFF, 0xFF, 0x87, 0x3F
TMPM330FWFG	0xFF, 0xFF, 0x81, 0x3F

Note: **The flash memory actual end address of the TMPM330FYFG is 0x3F83_FFFF. In addition to 256KB flash area, the TMPM330FYFG provides 128-word data area (0x3F87_FF00 . 0x3F87_FF80, 1 page) for Show Product Information command and the password area.**

14. The 55th to 56th bytes, transmitted from the target board to the controller, indicate the

number of flash blocks available. Each product transmits own number shown below.

Product name	Number of flash blocks
TMPM330FDFG	0x06, 0x00
TMPM330FYFG	0x06, 0x00
TMPM330FWFG	0x04, 0x00

15. The 57th to 83rd bytes, transmitted from the target board to the controller, contain information about the flash blocks. Flash blocks of the same size are treated as a group. Information about the flash blocks indicate the start address of a group, the size of the blocks in that group (in halfwords) and the number of the blocks in that group. The 57th to 65th bytes are the information about the 16-kbyte blocks. The 66th to 74th bytes are the information about the 32-kbyte blocks. The 75th to 83rd bytes are the information about the 64-kbyte blocks. The 84th to 92nd bytes are the information about the 128-kbyte blocks. See Table 17-8 for the values of bytes transmitted.
16. The 66th byte, transmitted from the target board to the controller, is a checksum value for the 5th to 92nd bytes. The checksum value is calculated by adding all these bytes together, dropping the carry and taking the two’s complement of the total sum.
17. The 94th byte is the next command code.

17.2.10.4 Chip and Protection Bit Erase Command

See Table 17-9 for the transfer format of this command.

1. The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.
2. From the Controller to the TMPM330FDFG/FYFG/FWFG

   The 3rd byte, which the target board receives from the controller, is a command. The code for the Show Product Information command is 0x40.
3. From the TMPM330FDFG/FYFG/FWFG to the Controller

   The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte.

   Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

   If the 3rd byte is equal to any of the command codes listed in Table 17-4, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was received, the boot program echoes back a value of 0x40. If the 3rd byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.
4. From the Controller to the TMPM330FDFG/FYFG/FWFG

   The 5th byte, transmitted from the target board to the controller, is the Chip Erase Enable command code (0x54).
5. From the TMPM330FDFG/FYFG/FWFG to the Controller

   The 6th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th byte.

   Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

   If the 5th byte is equal to any of the command codes to enable erasing, the boot program echoes it back to the controller. When the Chip and Protection Erase command was received, the boot program echoes back a value of 0x54 and then branches to the Chip Erase routine. If the 5th byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.
6. From the TMPM330FDFG/FYFG/FWFG to the Controller

   The 7th byte indicates whether the Chip Erase command is normally completed or not.

   At normal completion, completion code (0x4F) is sent.

   When an error was detected, error code (0x4C) is sent.
7. The 9th byte is the next command code.

17.2.10.5 Acknowledge Responses

The boot program represents processing states with specific codes. Table 17-10 to Table 17-13 show the values of possible acknowledge responses to the received data. The upper four bits of the acknowledge response are equal to those of the command being executed. Bit 3 of the code indicates a receive error. Bit 0 indicates an invalid command error, a checksum error or a password error. Bit 1 and bit 2 are always "0". Receive error checking is not done in I/O Interface mode.

Table 17-10 ACK Response to the Serial Operation Mode Byte

Return Value	Meaning
0x86	The SIO can be configured to operate in UART mode. (See Note)
0x30	The SIO can be configured to operate in I/O Interface mode.

Note: **If the serial operation mode is determined as UART, the boot program checks if the SIO can be programmed to the baud rate at which the operation mode byte was transferred. If that baud rate is not possible, the boot program aborts, without sending back any response.**

Table 17-11 ACK Response to the Command Byte

Return Value	Meaning
0x?8 (See Note)	A receive error occurred while getting a command code.
0x?1 (See Note)	An undefined command code was received. (Reception was completed normally.)
0x10	The RAM Transfer command was received.
0x20	The Show Flash Memory Sum command was received.
0x30	The Show Product Information command was received.
0x40	The Chip Erase command was received.

Note: **The upper four bits of the ACK response are the same as those of the previous command code.**

Table 17-12 ACK Response to the Checksum Byte

Return Value	Meaning
0xN8 (See Note)	A receive error occurred.
0xN1 (See Note)	A checksum or password error occurred.
0xN0 (See Note)	The checksum was correct.

Note: The upper four bits of the ACK response are the same as those of the operation command code. It is 1 (N ; RAM transfer command data [7:4]) when password error occurs.

Table 17-13 ACK Response to Chip and Protection Bit Erase Byte

Return Value	Meaning
0x54	The Chip Erase enabling command was received.
0x4F	The Chip Erase command was completed.
0x4C	The Chip Erase command was abnormally completed.

17.2.10.6 Determination of a Serial Operation Mode

The first byte from the controller determines the serial operation mode. To use UART mode for communications between the controller and the target board, the controller must first send a value of 0x86 at a desired baud rate to the target board. To use I/O Interface mode, the controller must send a value of 0x30 at 1/16 the desired baud rate. Figure 17-4 shows the waveforms for the first byte.

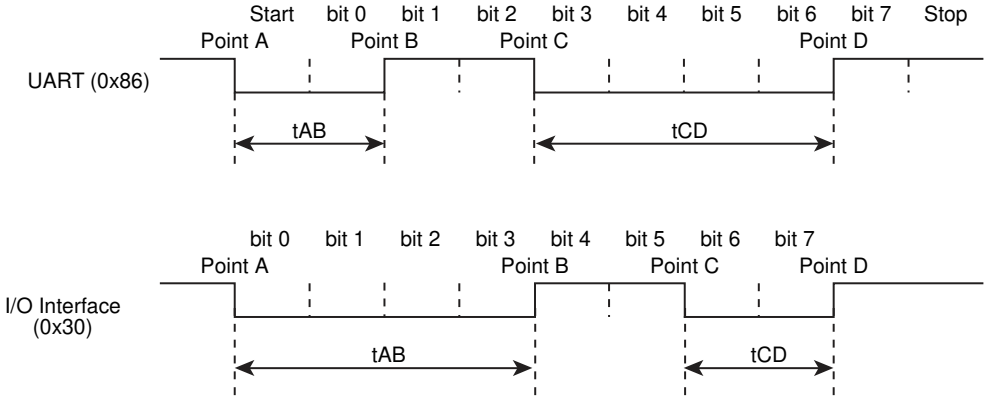


Figure 17-4 Serial Operation Mode Byte

After $\overline{\text{RESET}}$ is released, the boot program monitors the first serial byte from the controller, with the SIO reception disabled, and calculates the intervals of tAB, tAC and tAD. Figure 17-5 shows a flowchart describing the steps to determine the intervals of tAB, tAC and tAD. As shown in the flowchart, the boot program captures timer counts each time a logic transition occurs in the first serial byte. Consequently,the calculated tAB, tAC and tAD intervals are bound to have slight errors. If the transfer goes at a high baud rate, the CPU might not be able to keep up with the speed of logic transitions at the serial receive pin. In particular, I/O Interface mode is more prone to this problem since its baud rate is generally much higher than that for UART mode. To avoid such a situation, the controller should send the first serial byte at 1/16 the desired baud rate.

The flowchart in Figure 17-5 shows how the boot program distinguishes between UART and I/O Interface modes. If the length of tAB is equal to or less than the length of tCD, the serial operation mode is determined as UART mode. If the length of tAB is greater than the length of tCD, the serial operation mode is determined as I/O Interface mode. Bear in mind that if the baud rate is too high or the timer operating frequency is too low, the timer resolution will be coarse, relative to the intervals between logic transitions. This becomes a problem due to inherent errors caused by the way in which timer counts are captured by software; consequently the boot program might not be able to determine the serial operation mode correctly. To prevent this problem, reset UART mode within the programming routine.

For example, the serial operation mode may be determined to be I/O Interface mode when the intended mode is UART mode. To avoid such a situation, when UART mode is utilized, the controller should allow for a time-out period within which it expects to receive an echo-back (0x86) from the target board. The controller should give up the communication if it fails to get that echo-back within the allowed time. When I/O Interface mode is utilized, once the first serial byte has been transmitted, the controller should send the SCLK clock after a certain idle time to get an acknowledge response. If the received acknowledge response is not 0x30, the controller should give up further communications.

When the intended mode is I/O interface mode, the first byte does not have to be 0x30 as long as tAB is greater than tCD as shown above. 0x91, 0xA1 or 0xB1 can be sent as the first byte code to determine the falling edges of Point A and Point C and the rising edges of Point B and Point D. If tAB is greater than tCD and SIO is selected by the resolution of the operation mode determination, the second byte code is 0x30 even though the transmitted code on the first byte is not 0x30 (The first byte code to determine I/O interface mode is described as 0x30).

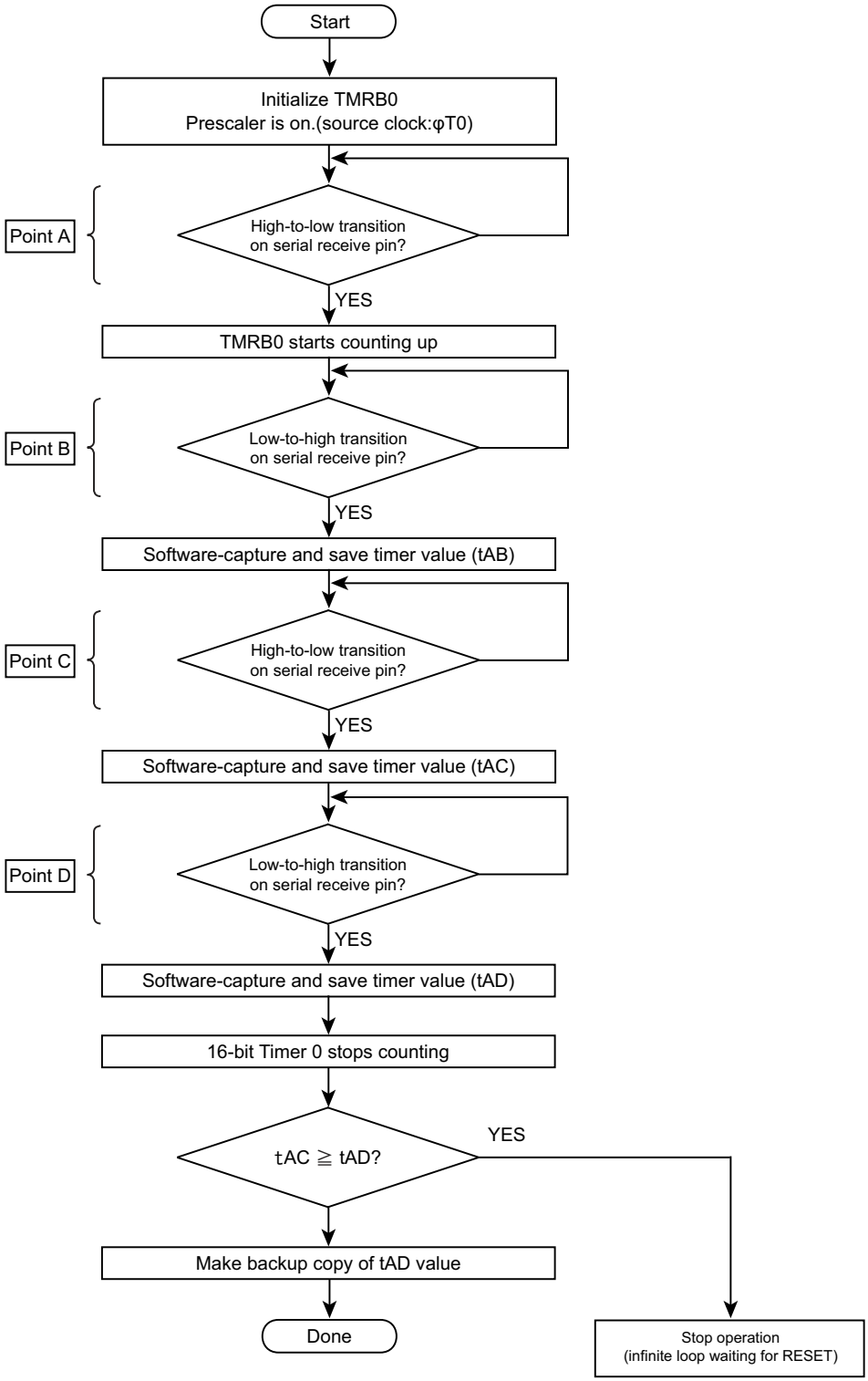


Figure 17-5 Serial Operation Mode Byte Reception Flowchart

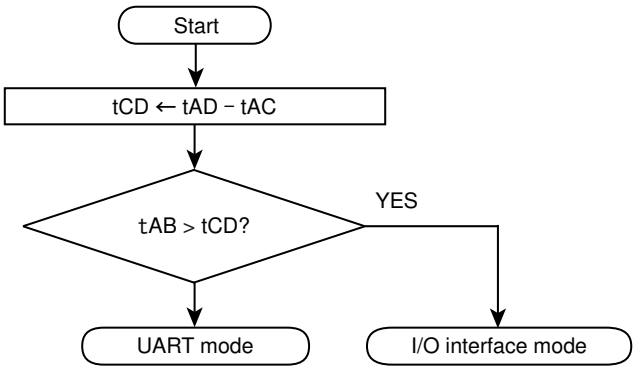


Figure 17-6 Serial Operation Mode Determination Flowchart

17.2.10.7 Password

The RAM Transfer command (0x10) causes the boot program to perform password verification. Following an echo-back of the command code, the boot program verifies the contents of the 12-byte password area within the flash memory. The following table shows the password area of each product.

Product name	Area
TMPM330FDFG TMPM330FYFG	0x3F87_FF04 to 0x3F87_FF0F
TMPM330FWFG	0x3F81_FF04 to 0x3F81_FF0F

Note: **If a password is set to 0xFF (erased data area), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.**

If all these address locations contain the same bytes of data other than 0xFF, a password area error occurs as shown in Figure 17-7. In this case, the boot program returns an error acknowledge (0x11) in response to the checksum byte (the 17th byte), regardless of whether the password sequence sent from the controller is all 0xFFs.

The password sequence received from the controller (5th to 16th bytes) is compared to the password stored in the flash memory. All of the 12 bytes must match to pass the password verification. Otherwise, a password error occurs, which causes the boot program to reply an error acknowledge in response to the checksum byte (the 17th byte).

The password verification is performed even if the security function is enabled.

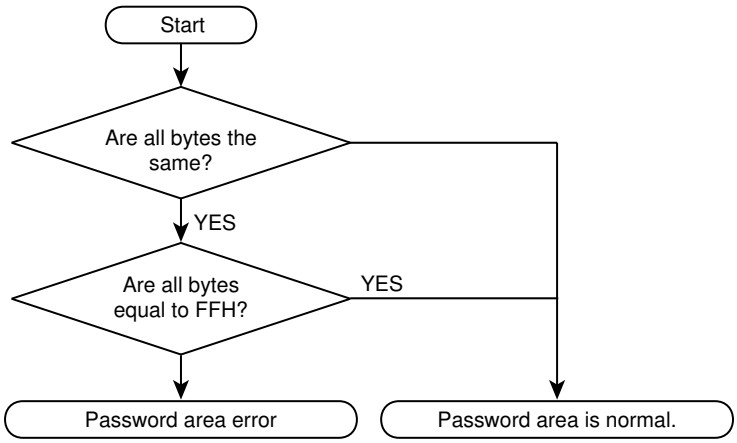


Figure 17-7 Password Area Verification Flowchart

17.2.10.8 Calculation of the Show Flash Memory Sum Command

The result of the sum calculation "byte + byte + byte + ・・・" is responded by a word quantity. The Show Flash Memory Sum command adds all 512 Kbytes of the flash memory together and provides the total sum as a halfword quantity. The sum is sent to the controller, with the upper eight bits first, followed by the lower eight bits.

Note: **In the TMP330FYFG, the range from 256KB through the password area and from the password area through 0x007_FFFF area are calculated as "0xFF".**

Example)

0xA1
0xB2
0xC3
0xD4

For the interest of simplicity, assume the depth of the flash memory is four locations. Then the sum of the four bytes is calculated as:
0xA1 + 0xB2 + 0xC3 + 0xD4 = 0x02EA
Hence, 0x02 is first sent to the controller, followed by 0xEA.

17.2.10.9 Checksum Calculation

The checksum byte for a series of bytes of data is calculated by adding the bytes together, dropping the carries, and taking the two’s complement of the total sum. The Show Flash Memory Sum command and the Show Product Information command perform the checksum calculation. The controller must perform the same checksum operation in transmitting checksum bytes.

Example) Assume the Show Flash Memory Sum command provides the upper and lower bytes of the sum as 0xE5 and 0xF6. To calculate the checksum for a series of 0xE5 and 0xF6:

Add the bytes together

0xE5 + 0xF6 = 0x1DB

Take the two’s complement of the sum, and that is the checksum byte.

0 − 0xDB = 0x25

17.2.11 General Boot Program Flowchart

Figure 17-8 shows an overall flowchart of the boot program.

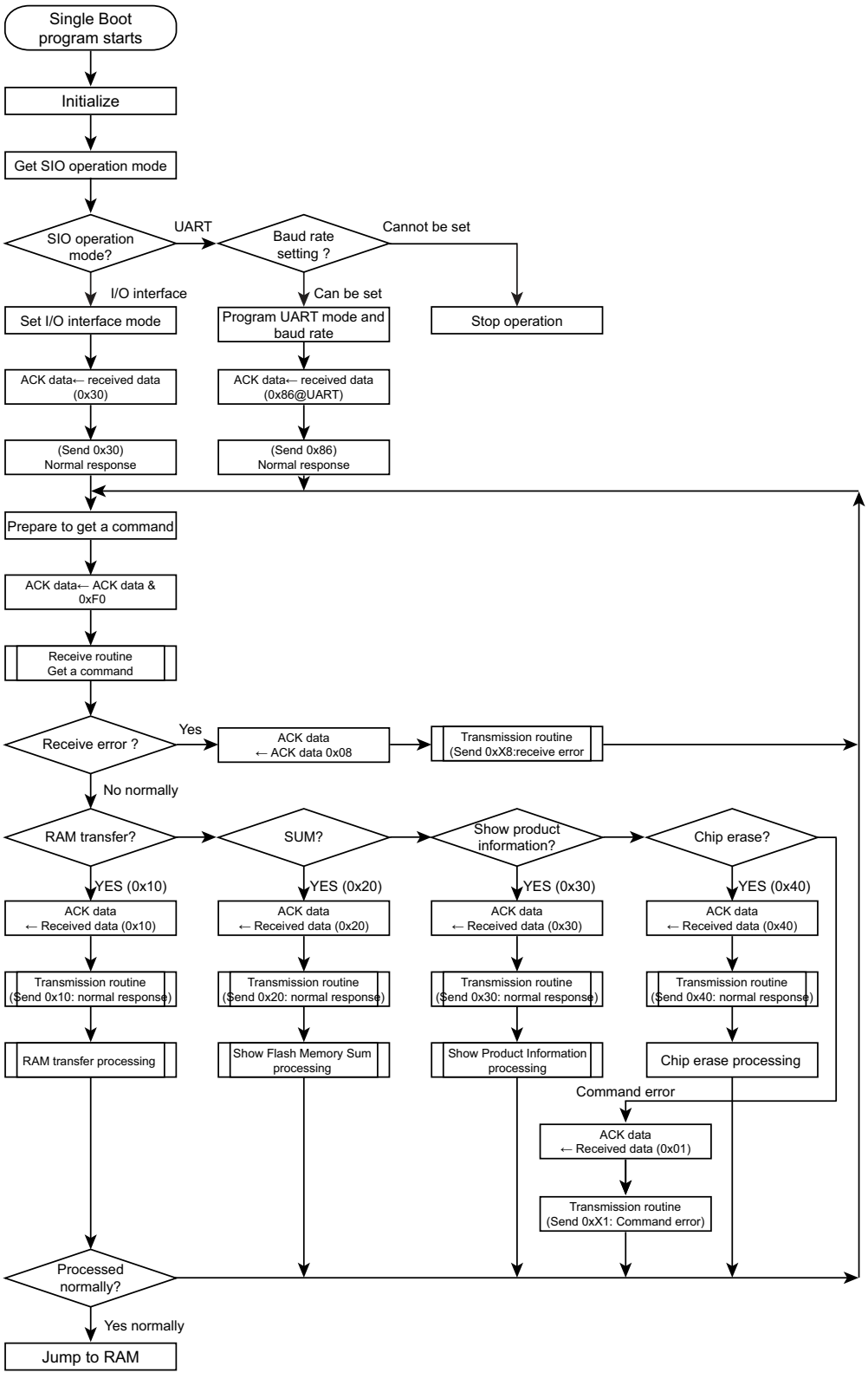


Figure 17-8 Overall Boot Program Flowchart

17.3 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 after shifting to the user boot mode.

17.3.1 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 17-14 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	The write or erase operation can be individually inhibited for each block.

17.3.1.1 Block Configuration

(1) TMPM330FDFG

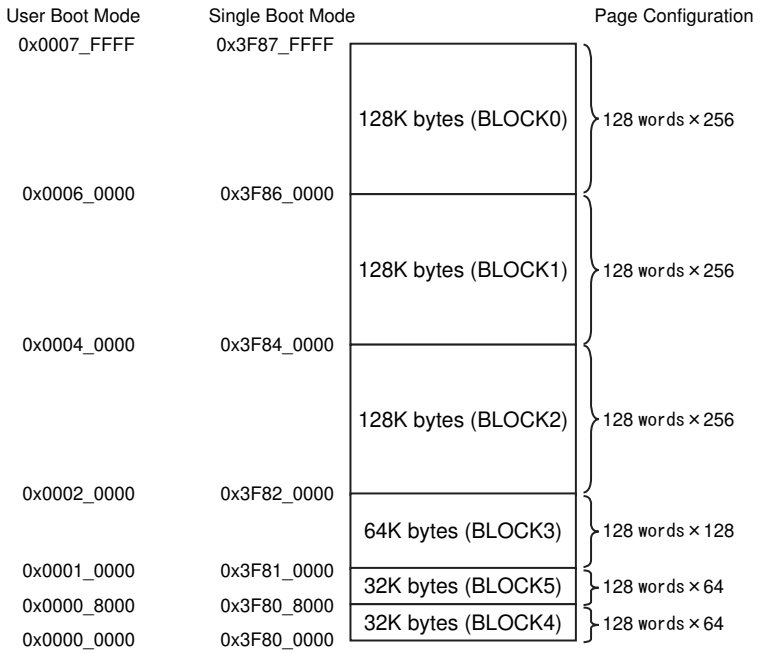


Figure 17-9 Block Configuration of Flash Memory (TMPM330FDFG)

(2) TMPM330FYFG

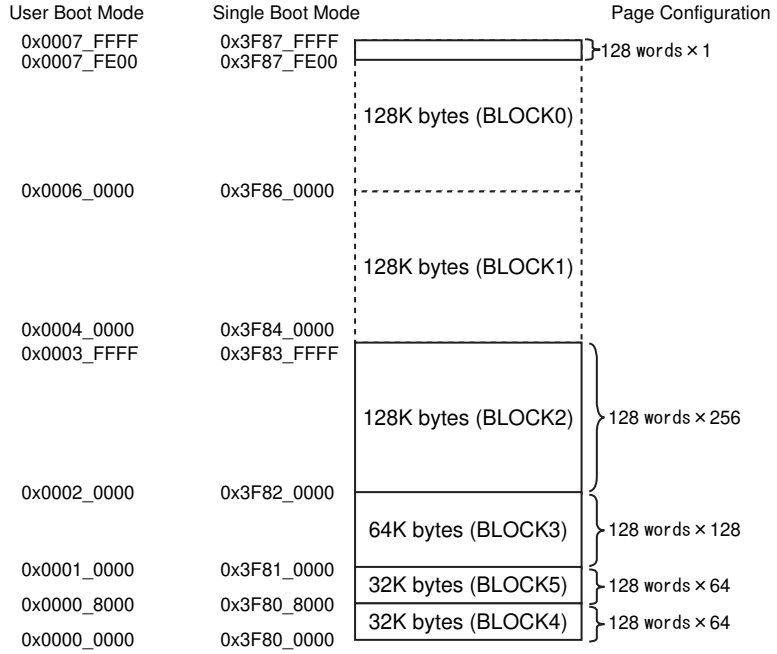


Figure 17-10 Block Configuration of Flash Memory (TMPM330FYFG)

Note: **In addition to 256KB flash area, the TMPM330FYFG provides 128-word data/password area (0x3F87_FE00 . 0x3F87_FFFF, 1 page) for Show Product Information command. To erase the content, execute the automatic chip erase command or assign block 0 with the automatic block erase command.**

(3) TMPM330FWFG

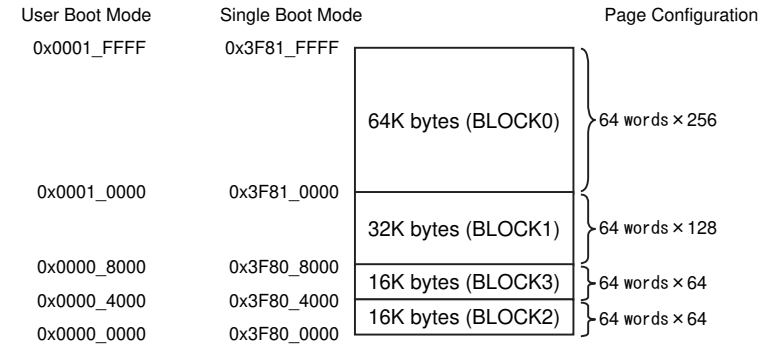


Figure 17-11 Block Configuration of Flash Memory (TMPM330FWFG)

17.3.1.2 Basic operation

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 reset and debug exceptions 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 debug port 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 FCFLCS <RDY/BSY> 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.**

17.3.1.3 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 automatic operation.

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 VIL 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 "17.2.1 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.

17.3.1.4 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 TMPM330FDFG/ TMPM330FYFG contain 128 words and the TMPM330FWFG contains 64 words in a page. A 128 word block is defined by a same [31:9] address and it starts from the address [8:0] = 0x00 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] = 0x00 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 FCFLCS [0] <RDY/BSY> .

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 third bus cycle is executed, the automatic page programming is in operation. This condition can be checked by monitoring FCFLCS<RDY/BSY>. 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, FCFLCS<RDY/BSY> 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> . 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> . 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> . 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 17-18 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 FCFLCS <BLPRO> to be described later. This status of the automatic programming operation to set protection bits can be checked by monitoring FCFLCS <RDY/BSY> . 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 FCFLCS <BLPRO> are set to "1" indicating that it is in the protected state. 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. FCFLCS <RDY/BSY> 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 <BLPRO> are set to "1" or not if FCSECBIT<FCSECBIT> is 0x1. Be sure to check the value of FCFLCS <BLPRO> before executing the automatic protection bit erase command. See the chapter "ROM protection" for details.

Note: **The TMPM330FYFG is configured with block 2 through 5. Block 0 and 1 require a programming of protection bits when using security function.**

- When all the FCFLCS <BLPRO> 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, FCFLCS 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 FCFLCS <BLPRO> include "0" (not all the protection bits are programmed):

  If the automatic protection bit is cleared to "0", the protection condition is canceled. With this device, protection bits can be programmed to an individual block and performed bit-erase operation in the four bits unit as shown in Table 17-19. The target bits are specified in the seventh bus write cycle.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.

17.3.1.5 Flash control/ status register

Base Address = 0x4004_0500

Register name		Address(Base+)
Security bit register	FCSECBIT	0x0000
Reserved	-	0x0004
Flash control register	FCFLCS	0x0020
Reserved	-	0x0024
Reserved	-	0x0028

Note: **Access to the "Reserved" areas is prohibited.**

(1) FCFLCS (Flash control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
After reset	0	0	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	RDY/BSY
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-22	−	R	Read as 0.
21-16	BLPRO5 to BLPRO0	R	Protection for Block5 to 0 (Note 3) 0: disabled 1: enabled 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.
15-1	−	R	Read as 0.
0	RDY/BSY	R	Ready/Busy (Note 1) 0: Auto operating 1:Auto operation terminated 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."

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 μs 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.

Note 3: The FCFLCS[21:20] of TMPM330FWFG have no function. They are read as "0".

(2) FCSECBIT (Security bit register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	SECBIT
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	SECBIT	R/W	Security bits 0:disabled 1:enabled

Note: **This register is initialized by cold reset.**

17.3.1.6 List of Command Sequences

Table 17-15 shows the addresses and the data of each command of flash memory.

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 (word) data transfer commands. (In the following table, only lower 8 bits data are shown.)

See Table 17-16 for the detail of the address bit configuration. Use a value of "Addr." in the Table 17-15 for the address [15:8] of the normal command in the Table 17-16.

Note: **Always set "0" to the address bits [1:0] in the entire bus cycle.**

Table 17-15 Flash Memory Access from the Internal CPU

Command sequence	First bus cycle	Second bus cycle	Third bus cycle	Fourth bus cycle	Fifth bus cycle	Sixth bus cycle	Seventh bus cycle
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	Data	Data	Data	Data	Data	Data	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

17.3.1.7 Address bit configuration for bus write cycles

Table 17-16 is used in conjunction with Table 17-15 "Flash Memory Access from the Internal CPU." Address setting can be performed according to the normal bus write cycle address configuration from the first bus cycle. "0" is recommended" in the Table 17-16 Address Bit Configuration for Bus Write Cycles can be changed as necessary.

Table 17-16 Address Bit Configuration for Bus Write Cycles

Address	Addr [31:19]	Addr [18]	Addr [17]	Addr [16]	Addr [15]	Addr [14]	Addr [13:11]	Addr [10]	Addr [9]	Addr [8]	Addr [7:0]

[TMPM330FDFG/FYFG/FWFG]

Normal commands	Normal bus write cycle address configuration			
	Flash area	"0" is recommended.	Command	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	Addr[1:0]="0" (fixed) , Others:0 (recommended)

[TMPM330FDFG/FYFG]

Block erase	BA: Block address (Set the sixth bus write cycle address for block erase operation)				
	Block selection (Table 17-17)		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)				
	Page selection				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	Protection bit selection (Table 17-18)	Fixed to "0".	Protection bit selection (Table 17-18)	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	Protection bit selection (Table 17-19)	"0" is recommended.		Addr[1:0]="0" (fixed) Others:0 (recommended)

[TMPM330FWFG]

Block erase	BA: Block address (Set the sixth bus write cycle address for block erase operation)				
	Block selection (Table 17-17)		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)				
	Page selection				Addr[1:0]="0" (fixed) Others:0 (recommended)
Protection bit programming	PBA: Protection bit address (Set the seventh bus write cycle address for protection bit programming				
	Flash area	Protection bit selection (Table 17-18)	Fixed to "0".	Protection bit selection (Table 17-18)	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	Protection bit selection (Table 17-19)	Fixed to "0".		Addr[1:0]="0" (fixed) Others:0 (recommended)

As block address, specify any address in the block to be erased.

Table 17-17 Block Address Table

Block	Address (User boot mode)	Address (Single boot mode)	Size (Kbyte)

[TMPM330FDFG/FYFG]

4	0x0000_0000 ~ 0x0000_7FFF	0x3F80_0000 ~ 0x3F80_7FFF	32
5	0x0000_8000 ~ 0x0000_FFFF	0x3F80_0000 ~ 0x3F80_FFFF	32
3	0x0001_0000 ~ 0x0001_FFFF	0x3F81_0000 ~ 0x3F81_FFFF	64
2	0x0002_0000 ~ 0x0003_FFFF	0x3F82_0000 ~ 0x3F83_FFFF	128
1	0x0004_0000 ~ 0x0005_FFFF	0x3F84_0000 ~ 0x3F85_FFFF	128
0	0x0006_0000 ~ 0x0007_FFFF	0x3F86_8000 ~ 0x3F87_FFFF	128

[TMPM330FWFG]

2	0x0000_0000 ~ 0x0000_3FFF	0x3F80_0000 ~ 0x3F80_3FFF	16
3	0x0000_4000 ~ 0x0000_7FFF	0x3F80_4000 ~ 0x3F80_7FFF	16
1	0x0000_8000 ~ 0x0000_FFFF	0x3F80_8000 ~ 0x3F80_FFFF	32
0	0x0001_0000 ~ 0x0001_FFFF	0x3F81_0000 ~ 0x3F81_FFFF	64

Note: **As for the addresses from the first to the fifth bus cycles, specify the upper addresses of the blocks to be erased.**

Table 17-18 Protection Bit Programming Address Table

Block	Protection bit	The seventh bus write cycle address						
		Address [18]	Address [17]	Address [16]	Address [15:11]	Address [10]	Address [9]	Address [8]

[TMPM330FDFG/FYFG]

Block0	<BLPRO[0]>	0	0	1	Fixed to "0".	0	0	"0" is recommended.
Block1	<BLPRO[1]>	0	0	1		0	1	
Block2	<BLPRO[2]>	0	0	1		1	0	
Block3	<BLPRO[3]>	0	0	1		1	1	
Block4	<BLPRO[4]>	0	1	0		0	0	
Block5	<BLPRO[5]>	0	1	0		0	1	

[TMPM330FWFG]

Block0	<BLPRO[0]>	0	0	Fixed to "0".	0	0
Block1	<BLPRO[1]>	0	0		0	1
Block2	<BLPRO[2]>	0	0		1	0
Block3	<BLPRO[3]>	0	0		1	1

Table 17-19 Protection Bit Erase Address Table

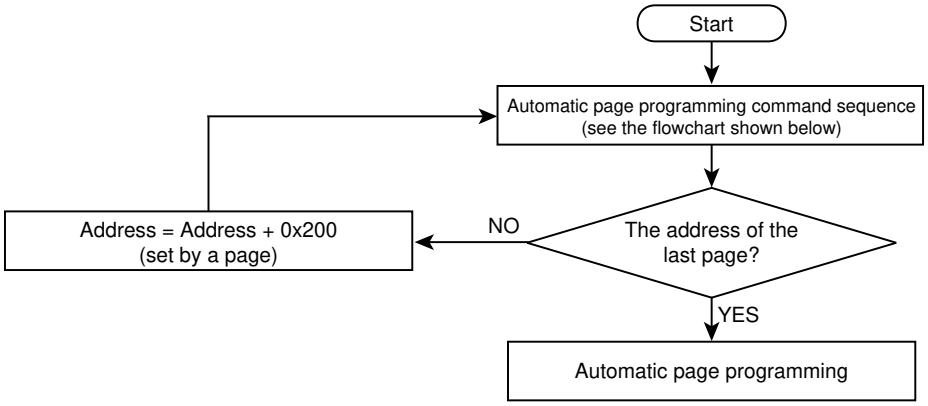
Block	Protection bit	The seventh bus write cycle address [18:17]	
		Address[18]	Address[17]
Block0 to 3	<BLPRO[0:3]>	0	0
Block4 to 5	<BLPRO[4:5]>	0	1

Note: **The protection bit erase command cannot erase by individual block.**

Table 17-20 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	0x12 (TMPM330FDFG) 0x12 (TMPM330FYFG) 0x11 (TMPM330FWFG)	Macro code

17.3.1.8 Flowchart



Automatic Page Programming Command Sequence (Address/ Command)

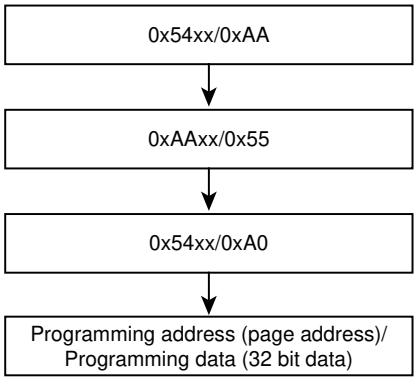


Figure 17-12 Automatic Programming

Note: Command sequence is executed by 0x54xx or 0x55xx.

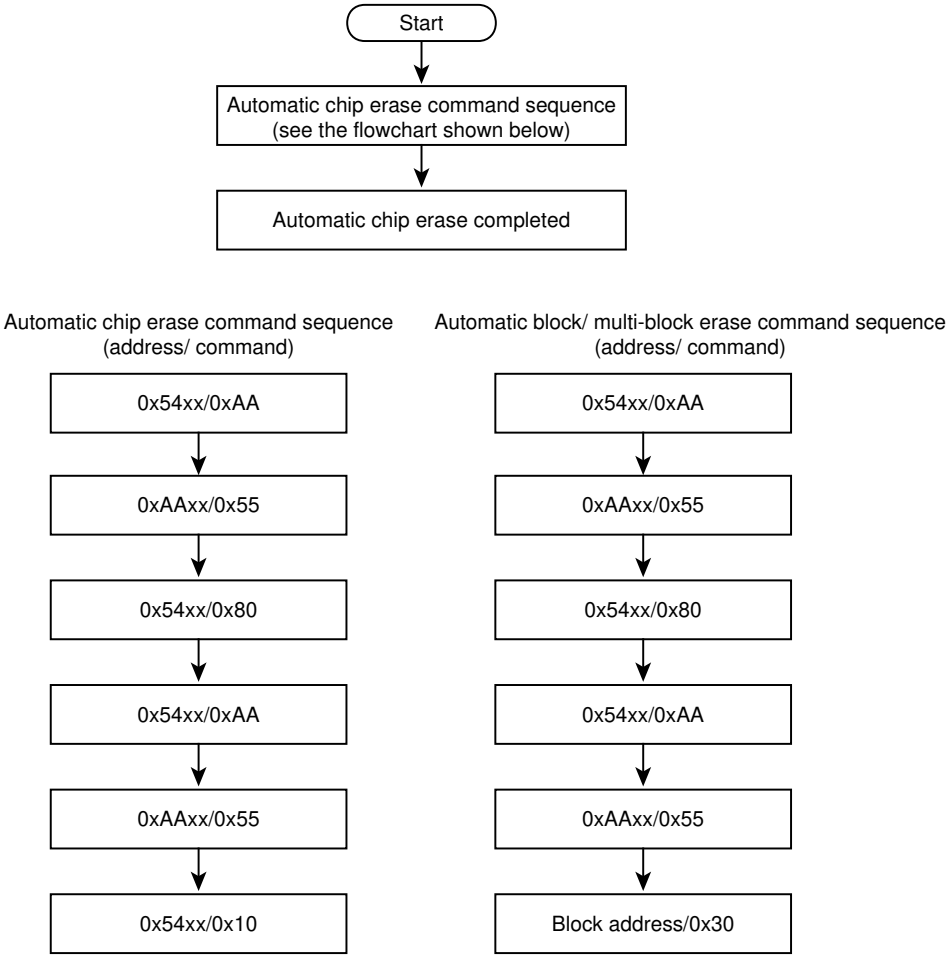


Figure 17-13 Automatic Erase

Note: **Command sequence is executed by 0x54xx or 0x55xx.**

18. ROM protection

18.1 Outline

The TMPM330FDFG/FYFG/FWFG 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.

18.2 Future

18.2.1 Write/ erase-protection function

The write/ erase-protection function enables the internal flash to prohibit the writing and erasing operation for each block.

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 <BLPRO[5:0]> bit. See the chapter "Flash" for programming details.

18.2.2 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<BLPRO> 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.**

Note: **The TMPM330FYFG is configured with block 2 through 5. Block 0 and 1 require a programming of protection bits when using security function.**

Table 18-1 shows details of the restrictions by the security function.

Table 18-1 Restrictions by the security function

Item	Details
1) ROM data readout	Data can be read from CPU.
2) Debug port	Communication of JTAG/SW 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.

18.3 Register

Base Address = 0x4004_0500

Register name		Address(Base+)
Security bit register	FCSECBIT	0x0000
Reserved	-	0x0004
Flash control register	FCFLCS	0x0020
Reserved	-	0x0024
Reserved	-	0x0028

Note: **Access to the "Reserved" areas is prohibited.**

18.3.1 FCFLCS (Flash control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
After reset	0	0	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	RDY/BSY
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-22	−	R	Read as 0.
21-16	BLPRO5 to BLPRO0	R	Protection for Block5 to 0 (Note 3) 0: disabled 1: enabled 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.
15-1	−	R	Read as 0.
0	RDY/BSY	R	Ready/Busy (Note 1) 0: Auto operating 1:Auto operation terminated 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."

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 ms 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.**

Note 3: **The FCFLCS[21:20] of TMPM330FWFG have no function. They are read as "0".**

18.3.2 FCSECBIT(Security bit register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	SECBIT
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	ã@î\
31-1	−	R	Read as 0.
0	SECBIT	R/W	Security bit 0: Disabled 1: Enabled

Note: This register is initialized only by power-on reset.

18.4 Writing and erasing

18.4.1 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 0 through 3 and block 4 through 5. 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 the capter "Flash" for details

18.4.2 Security bit

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.1.

Note: The above procedure is enabled only when using 32-bit data transfer command.

19. Electrical Characteristics

19.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		DVDD3	−0.3 to 3.9	V
		AVDD3	−0.3 to 3.9	
		RVDD3	−0.3 to 3.9	
Input voltage		V_{IN}	−0.3 to VDD + 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(10 s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	−40 to 125	°C
Operating Temperature	Except during Flash W/E	T_{OPR}	−20 to 85	°C
	During Flash W/E		0 to 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.**

19.2 DC Electrical Characteristics (1/3)

Ta = −20 to 85 °C

Parameter		Symbol	Condition		Min	Typ. (Note 1)	Max	Unit
Supply voltage	DVDD3 = AVDD3 = RVDD3 (Note2) DVSS = AVSS = 0V	DVDD3 AVDD3 RVDD3	f_{OSC} = 8 to 10 MHz fsys = 1 to 40 MHz fs = 30 to 34 kHz		2.7	−	3.6	V
Low-level input voltage	PC0 to 3, PD4 to 7	V_{IL1}	2.7 V ≤ AVDD3 ≤ 3.6 V		−0.3	−	0.3 AVDD3	V
	PD0 to 3	V_{IL2}					0.2 AVDD3	
	PA2 to 7, PB0, PB3 to 7, PE0, PE4, PF0, PG7, PI0 to 5, PJ4 to 5,PK1 to 2	V_{IL3}	2.7 V ≤ DVDD3 ≤ 3.6 V				0.3 DVDD3	
	PA0 to 1, PB1 to 2, PE1 to 3, PE5 to 6, PF1 to 7, PG0 to 6, PH0 to 7, PI6 to 7, PJ0 to 3, PJ6 to 7 $\overline{RESET}$, $\overline{NMI}$, MODE	V_{IL4}					0.2 DVDD3	
	PK0	V_{IL5}						
	X1	V_{IL6}					0.1 DVDD3	
High-level input voltage	PC0 to 3, PD4 to 7	V_{IH1}	2.7 V ≤ AVDD3 ≤ 3.6 V		0.7 AVDD3	−	AVDD3 + 0.3	V
	PD0 to 3	V_{IH2}			0.8 AVDD3			
	PA2 to 7, PB0, PB3 to 7, PE0, PE4, PF0, PG7, PI0 to 5, PJ4 to 5,PK1 to 2	V_{IH3}	2.7 V ≤ DVDD3 ≤ 3.6 V		0.7 DVDD3		DVDD3 + 0.3	
	PA0 to 1, PB1 to 2, PE1 to 3, PE5 to 6, PF1 to 7, PG0 to 6, PH0 to 7, PI6 to 7, PJ0 to 3, PJ6 to 7 $\overline{RESET}$, $\overline{NMI}$, MODE	V_{IH4}			0.8 DVDD3			
	PK0	V_{IH5}					3.6	
	X1	V_{IH6}			0.9 DVDD3		DVDD3 + 0.3	
Low-level output voltage		V_{OL}	I_{OL} = 2 mA	DVDD3 ≥ 2.7 V	−	−	0.4	V
High-level output voltage		V_{OH}	I_{OH} = −2 mA	DVDD3 ≥ 2.7 V	2.4	−	−	V
Input leakage current	Except PK0	I_{LI1}	0.0 ≤ V_{IN} ≤ DVDD3 0.0 ≤ V_{IN} ≤ AVDD3		−	0.02	±5	μA
	PK0	I_{LI2}	0.0 V ≤ V_{IN} ≤ 3.6 V					
Output leakage current	Except PK0	I_{LO1}	0.2 ≤ V_{IN} ≤ DVDD3 − 0.2 0.2 ≤ V_{IN} ≤ AVDD3 − 0.2		−	0.05	±10	
	PK0	I_{LO2}	0.2 V ≤ V_{IN} ≤ 3.4 V					
Pull-up resister at Reset		RRST	DVDD3 = 2.7 V to 3.6 V		30	50	150	kΩ
Hysteresis voltage		VTH	2.7 V ≤ DVDD3 ≤ 3.6 V		0.3	0.6	−	V
Programmable pull-up/pull-down resistor		PKH	DVDD3 = 2.7 V to 3.6 V		−	50	150	kΩ
Pin capacitance (Except power supply pins)		C_{IO}	fc = 1 MHz		−	−	10	pF

Note 1: **Ta = 25 °C, DVDD3 = RVDD3 = AVDD3 = 3.3 V, unless otherwise noted.**

Note 2: **The same voltage must be supplied to DVDD3, AVDD3, and RVDD3.**

19.3 DC Electrical Characteristics (2/3)

DVDD3 = AVDD3 = RVDD3 = 2.7 V to 3.6 V, Ta = −20 to 85 °C

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Low-level output current	I_{OL}	Per pin	−	−	2	mA
	ΣI_{OL1}	Per group GrL1 = 6 to 12,89pin <PE3 / PG3,7 / PH4 to 5 / PJ5 / PK2> GrL2 = 15 to 37pin <PB3 to 4 / PE0 to 2,4 to 6 / PF0 to 2,7, PG0 to 2,4 to 6 / PH0 to 3> GrL3 = 15=61pin <PA0,1 / PB0 to 2,5 to 7 / PF4 to 6 / PI0 to 5 / PJ1,6,7 / PK0,1> GrL4 = 15 to 37pin <PA2 to 7 / PF3 / PH6,7 / PI6,7 / PJ0,2 to 4>	−	−	18	mA
	ΣI_{OL2}	Total	−	−	35	mA
High-level output current	I_{OH}	Per pin	−	−	-2	mA
	ΣI_{OH1}	Per group GrL1 = 6 to 12,89pin <PE3 / PG3,7 / PH4,5 / PJ5 / PK2> GrL2 = 15 to 37pin <PB3,4 / PE0 to 2,4 to 6 / PF0 to 2,7, PG0 to 2,4 to 6 / PH0 to 3> GrL3 = 15=61pin <PA0,1 / PB0 to 2,5 to 7 / PF4 to 6 / PI0 to 5 / PJ1,6,7 / PK0,1> GrL4 = 15 to 37pin <PA2 to 7 / PF3 / PH6,7 / PI6,7 / PJ0,2 to 4>	−	−	-18	mA
	ΣI_{OH2}	Total	−	−	-35	mA

Note: **The same voltage must be supplied to DVDD3, AVDD3, and RVDD3.**

19.4 DC Electrical Characteristics (3/3)

19.4.1 TMPM330FDFG/TMPM330FYFG

DVDD3 = AVDD3 = RVDD3 = 2.7 V to 3.6 V, Ta = −20 to 85 °C

Parameter	Symbol	Condition	Min	Typ. (Note1)	Max	Unit
NORMAL (Note2) Gear 1/1	I_{DD}	fsys = 40 MHz (f_{OSC} = 10 MHz)	−	32	42	mA
IDLE (Note3)			−	8	13	
SLOW		fs = 32.768 kHz	−	2.5	6	
SLEEP			−	55	950	µA
STOP		−	−	55	900	

Note 1: **Ta = 25 °C, DVDD3 = AVDD3 = RVDD3 = 3.3 V, unless otherwise noted.**

Note 2: I_{DD} NORMAL: Measured with the dhrystone ver. 2.1 operated in FLASH. All functions operates excluding A/D**.**

Note 3: **I_{DD} IDLE: Measured with all functions stopped.The currents flow through DVDD3, AVDD3 and RVDD3 are included.**

19.4.2 TMPM330FWFG

DVDD3 = AVDD3 = RVDD3 = 2.7 V to 3.6 V, Ta = −20 to 85 °C

Parameter	Symbol	Condition	Min	Typ. (Note1)	Max	Unit
NORMAL (Note2) Gear 1/1	I_{DD}	fsys = 40 MHz (f_{OSC} = 10 MHz)	−	26	32	mA
IDLE (Note3)			−	8	12.5	
SLOW		fs = 32.768 kHz	−	2.5	5.5	
SLEEP			−	55	900	µA
STOP		−	−	45	850	

Note 1: **Ta = 25 °C, DVDD3 = AVDD3 = RVDD3 = 3.3 V, unless otherwise noted.**

Note 2: **I_{DD} NORMAL: Measured with the dhrystone ver. 2.1 operated in FLASH. All functions operates excluding A/D**

Note 3: **I_{DD} IDLE: Measured with all functions stopped.The currents flow through DVDD3, AVDD3 and RVDD3 are included.**

19.5 10-bit ADC Electrical Characteristics

DVDD3 = AVDD3 = RVDD3 = VREFH = 2.7 V to 3.6 V
AVSS = DVSS, Ta = −20 to 85 °C

Parameter		Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage(+)		VREFH	−	2.7	3.3	3.6	V
Analog input voltage		VAIN	−	AVSS	−	VREFH	V
Power supply current of analog reference voltage	AD conversion	IREF	DVSS = AVSS	−	2.5	5.5	mA
	Non-AD conversion			−	±0.02	±5	μA
Supply current	AD conversion	−	Except IREF	−	−	3	mA
INL error		−	AIN resistance ≤ 600 Ω AIN load capacitance ≤ 30 pF Conversion time ≥ 1.15 μs	−	±2	±3	LSB
DNL error				−	±1	±2	
Offset error				−	±2	±4	
Full-scale error				−	±2	±4	
INL error		−	AIN resistance ≤ 600 Ω AIN load capacitance ≤ 0.1 μF Conversion time ≥ 1.15 μs	−	±2	±3	
DNL error				−	±1	±2	
Offset error				−	±2	±4	
Full-scale error				−	±2	±4	
INL error		−	AIN resistance ≤ 10 kΩ AIN load capacitance ≥ 0.1 μF Conversion time ≥ 2.30 μs	−	±2	±3	
DNL error				−	±1	±2	
Offset error				−	±2	±4	
Full-scale error				−	±2	±4	

Note: **1LSB = (VREFH − AVSS)/1024 [V]**

Note: Peripheral functions are disable.

19.6 AC Electrical Characteristics

19.6.1 AC measurement condition

The AC characteristics data of this chapter is measured under the following conditions unless otherwise noted

- Output levels: High = 0.8 × DVDD3, Low = 0.2 × DVDD3
- Input levels: Refer to low-level input voltage and high-level input voltage in "DC Electrical Characteristics".
- Load capacity: CL = 30pF

19.6.2 Serial Channel (SIO/UART)

19.6.2.1 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.

(1) SCLK input mode

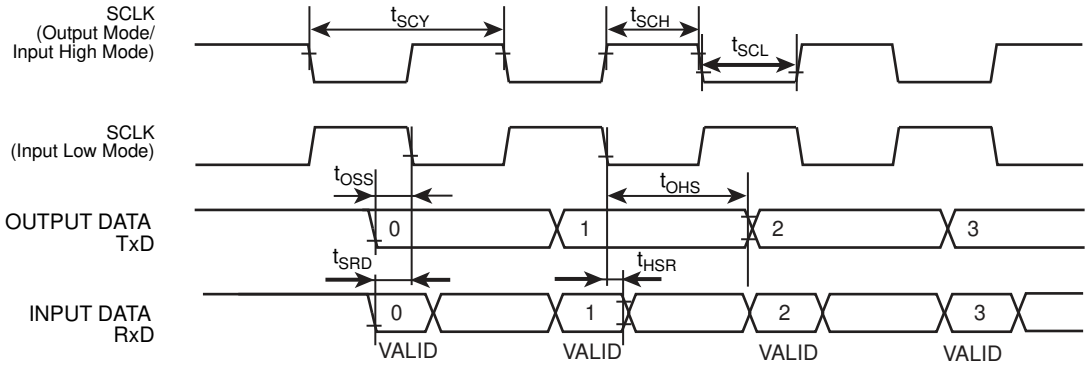
Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	3x	−	75	−	ns
SCLK Clock Low width (input)	t_{SCL}	3x	−	75	−	
SCLK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	−	150	−	
Output Data ← SCLK rise or fall (Note 1)	t_{OSS}	$t_{SCY}/2 - 3x - 45$	−	−45 (Note2)	−	
SCLK rise → Output Data hold or fall(Note1)	t_{OHS}	$t_{SCY}/2$	−	75	−	
Valid Data input ← SCLK rise or fall(Note1)	t_{SRD}	30	−	30	−	
SCLK rise → Input Data hold or fall(Note1)	t_{HSR}	x + 30	−	55	−	

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

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCLK cycle (programmable)	t_{SCY}	4x	−	100	−	ns
Output Data ← SCLK rise	t_{OSS}	$t_{SCY}/2 - 20$	−	30	−	
SCLK rise → Output hold Data hold	t_{OHS}	$t_{SCY}/2 - 20$	−	30	−	
Valid Data Input ← SCLK rise	t_{SRD}	45	−	45	−	
SCLK rise → Input Data hold	t_{HSR}	0	−	0	−	



19.6.3 Serial Bus Interface(I2C/SIO)

19.6.3.1 I2C Mode

In the table below, the letter x represents the I2C operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

n denotes the value of n programmed into the SCK (SCL output frequency select) field in the SBIxCR.

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	−	μs
SCL Low width (Input) (Note 1)	t_{LOW}	−	−	4.7	−	1.3	−	μs
SCL High width (Input) (Note 2)	t_{HIGH}	−	−	4.0	−	0.6	−	μs
Setup time for a repeated START condition	$t_{SU;\ STA}$	(Note5)	−	4.7	−	0.6	−	μs
Data hold time (Input) (Note 3, 4)	$t_{HD;\ DAT}$	−	−	0.0	−	0.0	−	μs
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}	(Note5)	−	4.7	−	1.3	−	μs

Note 1: **SCL clock Low width (output) = $(2^{n-1} + 58)/x$**

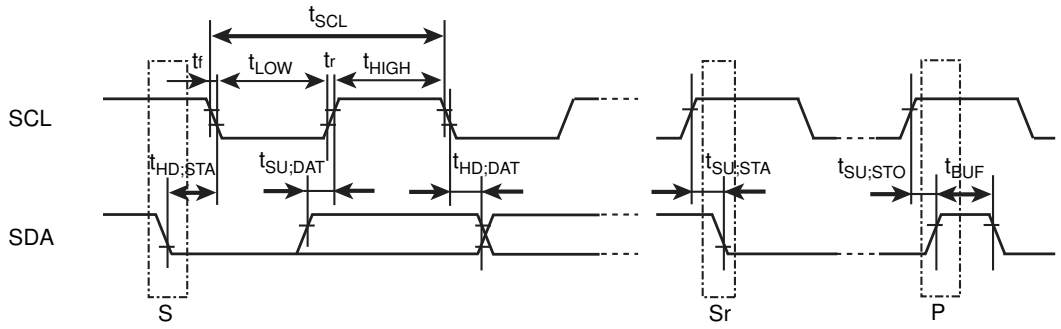
Note 2: **SCL clock High width (output) = $(2^{n-1} + 12)/x$ On I2C-bus specification, Maximum Speed of Standard Mode is 100kHz, Fast mode is 400khz. Internal SCL Frequency setting should comply with Note1 & Note2 shown above.**

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

19.6.3.2 Clock-Synchronous 8-Bit SIO mode

In the table below, the letter x represents the I2C operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

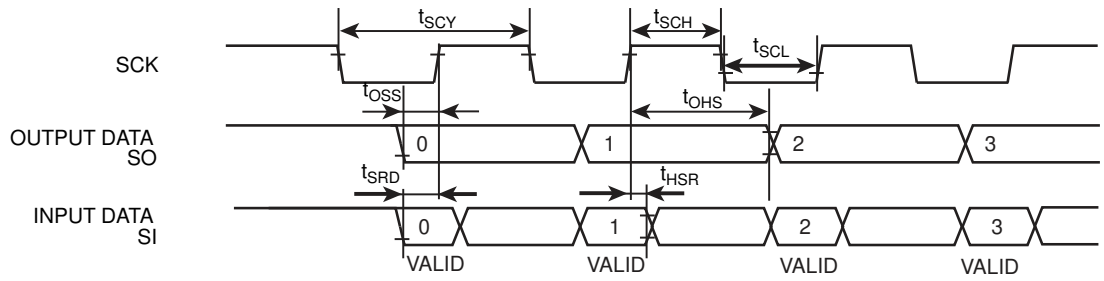
(1) SCK Input Mode (The electrical specifications below are for an SCK signal with a 50% duty cycle.)

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCK Clock High width (input)	t_{SCH}	4x	−	100	−	ns
SCK Clock Low width (input)	t_{SCL}	4x	−	100	−	
SCK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	−	200	−	
Output Data ← SCK rise	t_{OSS}	$t_{SCY}/2 - 3x - 45$	−	−20 (Note)	−	
SCK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 + x$	−	125	−	
Valid Data input ← SCK rise	t_{SRD}	30 − x	−	5	−	
SCK rise → Input Data hold	t_{HSR}	30	−	30	−	

Note: **Keep this value positive by adjusting SCK cycle.**

(2) SCK Output Mode (The electrical specifications below are for an SCK signal with a 50% duty cycle.)

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCK cycle (programmable)	t_{SCY}	16x	−	400	−	ns
Output Data ← SCK rise	t_{OSS}	$t_{SCY}/2 - 20$	−	180	−	
SCK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 - 20$	−	180	−	
Valid Data input ← SCK rise	t_{SRD}	45	−	45	−	
SCK rise → Input Data hold	t_{HSR}	0	−	0	−	



19.6.4 Event Counter

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		40 MHz		Unit
		Min	Max	Min	Max	
Clock Low pulse width	t_{VCKL}	2x + 100	−	150	−	ns
Clock High pulse width	t_{VCKH}	2x + 100	−	150	−	ns

19.6.5 Capture

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		40 MHz		Unit
		Min	Max	Min	Max	
Low pulse width	t_{CPL}	2x + 100	−	150	−	ns
High pulse width	t_{CPH}	2x + 100	−	150	−	ns

19.6.6 External Interrupt

In the table below, the letter x represents the fsys cycle time.

1. Except STOP release interrupts

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
INT0 to 7 low level pulse width	t_{INTAL}	x + 100	−	125	−	ns
INT0 to 7 high level pulse width	t_{INTAH}	x + 100	−	125	−	ns

2. STOP release interrupts

Parameter	Symbol	Min	Max	Unit
INT0 to 7 low level pulse width	t_{INTBL}	125	−	ns
INT0 to 7 high level pulse width	t_{INTBH}	125	−	ns

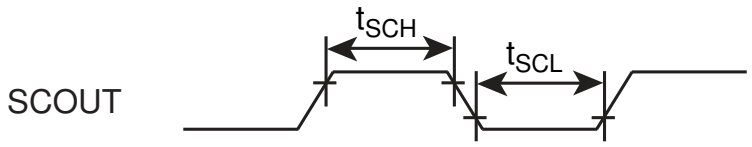
19.6.7 $\overline{\text{NMI}}$

Parameter	Symbol	Min	Max	Unit
$\overline{\text{NIM}}$ low level pulse width	t_{INTCL}	100	−	ns

19.6.8 SCOUT Pin AC Characteristic

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
High pulse width	t_{SCH}	0.5T − 5	−	7.5	−	ns
Low pulse width	t_{SCL}	0.5T − 5	−	7.5	−	ns

Note: **In the above table, the letter T represents the cycle time of the SCOUT output clock.**



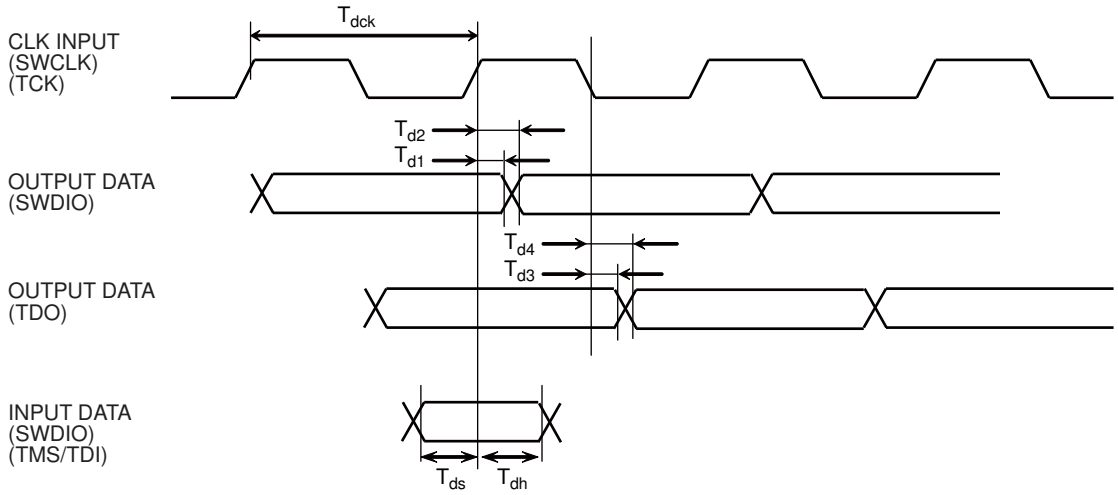
19.6.9 Debug Communication

19.6.9.1 SWD Interface

Parameter	Symbol	Min	Max	Unit
CLK cycle	T_{dck}	83.33	−	ns
CLK rise →Output data hold	T_{d1}	4	−	ns
CLK rise → Output data valid	T_{d2}	−	30	ns
Input data valid ←CLK rise	T_{ds}	20	−	ns
CLK rise → Input data hold	T_{dh}	15	−	ns

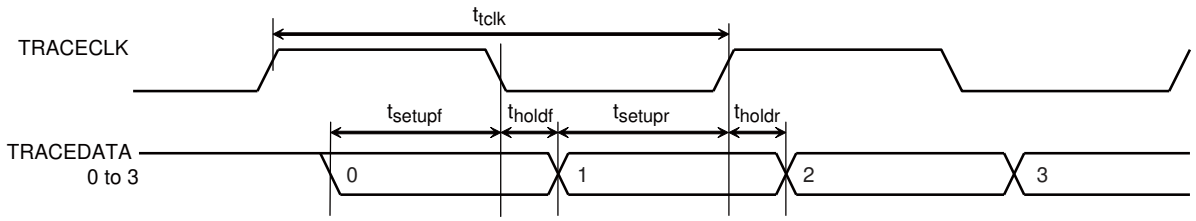
19.6.9.2 JTAG Interface

Parameter	Symbol	Min	Max	Unit
CLK cycle	T_{dck}	100	−	ns
CLK fall →Output data hold	T_{d3}	4	−	ns
CLK fall → Output data valid	T_{d4}	−	50	ns
Input data valid ←CLK rise	T_{ds}	20	−	ns
CLK rise → Input data hold	T_{dh}	15	−	ns



19.6.10 ETM Trace

Parameter	Symbol	Min	Max	Unit
TRACECLK cycle	t_{tclk}	50	−	ns
TRACEDATA valid ← TRACECLK rise	t_{setupr}	2	−	ns
TRACECLK rise → TRACEDATA hold	t_{holdr}	1	−	ns
TRACEDATA valid ← TRACECLK fall	t_{setupf}	2	−	ns
TRACECLK fall → TRACEDATA hold	t_{holdf}	1	−	ns



19.7 Flash Characteristics

19.7.1 Rewriting

Parameter	Condition	Min	Typ.	Max	Unit
Guarantee on Flash-memory rewriting	DVDD3 = AVDD3 = RVDD3 = 2.7 V to 3.6 V Ta = 0 to 70 °C	−	−	100	Times

19.8 Recommended Oscillation Circuit

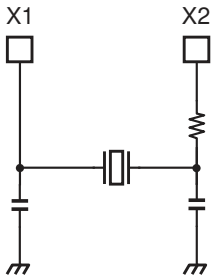


Figure 19-1 High-frequency oscillation connection

Note: **To obtain a stable oscillation, load capacity and the position of the oscillator must be configured properly. Since these factors are strongly affected by substratepatterns, please evaluate oscillation stability using the substrate you use.**

The TX03 has been evaluated by the oscillator vender below. Please refer this information when selecting external parts

19.8.1 Ceramic oscillator

The TX03 recommends the high-frequency oscillator by Murata Manufacturing Co., Ltd.
Please refer to the following URL for details.

http://www.murata.co.jp

19.8.2 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

19.9 Handling Precaution

19.9.1 Solderability

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	

19.9.2 Power-on sequence

The power supply must be raised (from 0V to 2.7V) at a speed of 0.37ms/V or slower. The power-on sequence must consider the time for the internal regulator and oscillator to be stableIn the .In the TX03, the internal regulator requires at least 700 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.

Figure 19-2 shows the power-on sequence.

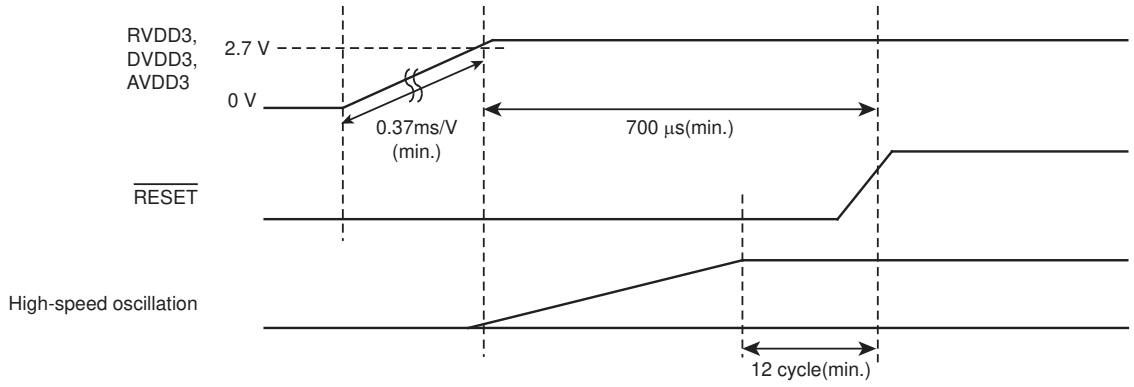


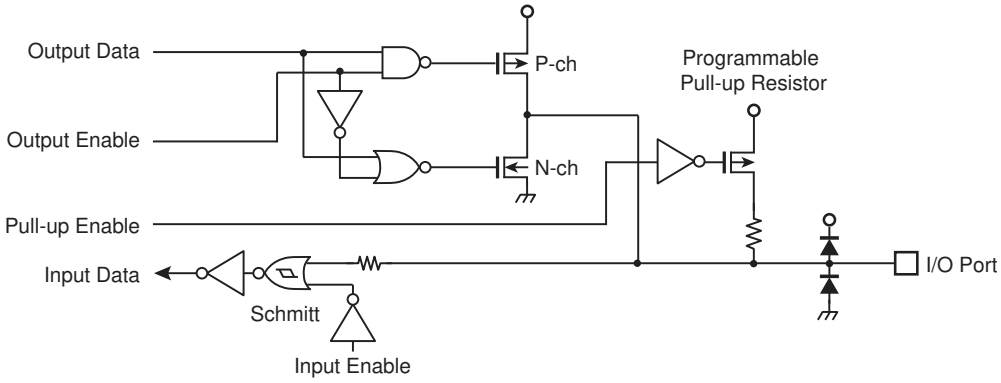
Figure 19-2 Power-on sequence

20. Port Section Equivalent Circuit Schematic

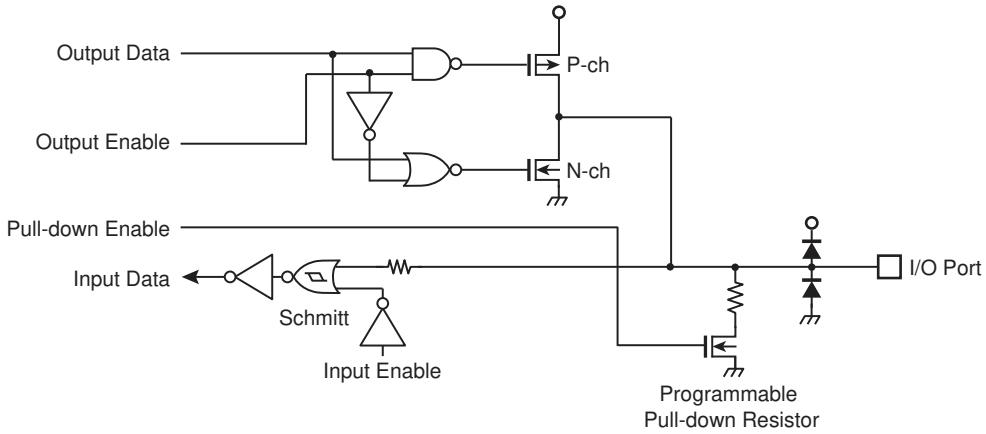
Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

The input protection resistance ranges from several tens of Ω to several hundreds of Ω. Damping resistors X2 and XT2 are shown with a typical value.

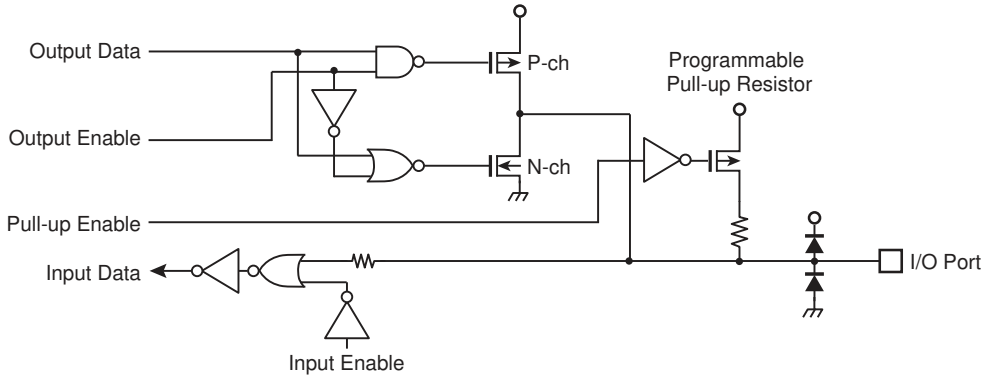
20.1 PA0, PB1 to 2, PE1 to 3, PE5 to 6, PF1 to 7, PG0 to 6, PH0 to 7, PI6 to 7, PJ0 to 3, PJ6 to 7



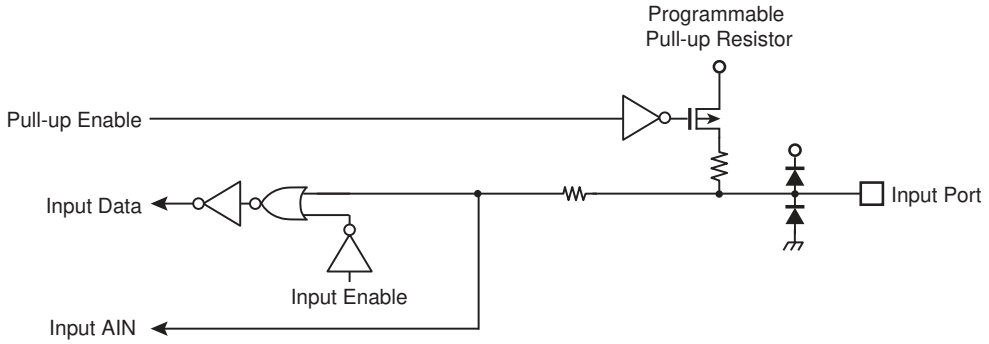
20.2 PA1



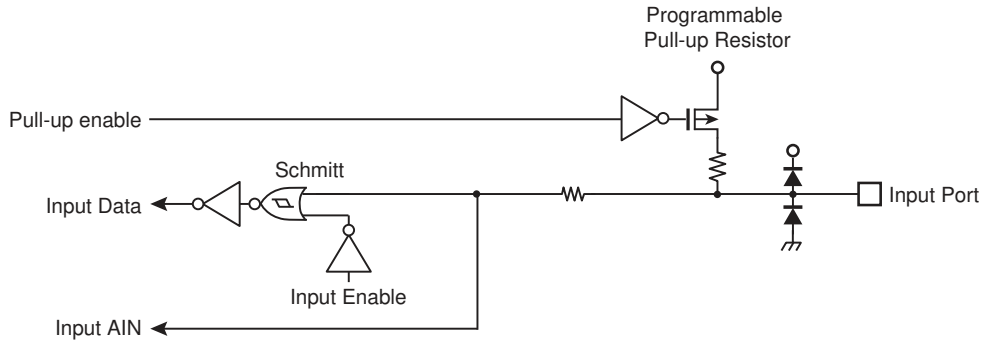
20.3 PA2 to 7, PB0, PB3 to 7, PE0, PE4, PF0, PG7, PI0 to 5, PJ4 to 5, PK1 to 2



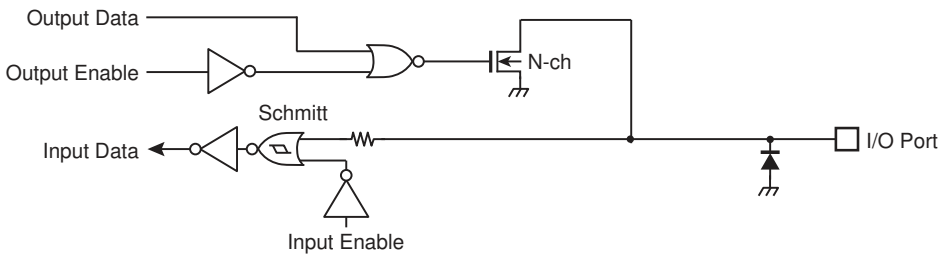
20.4 PC0 to 3, PD4 to 7



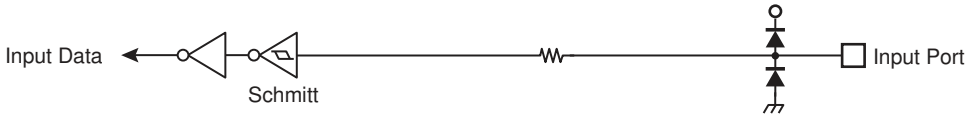
20.5 PD0 to 3



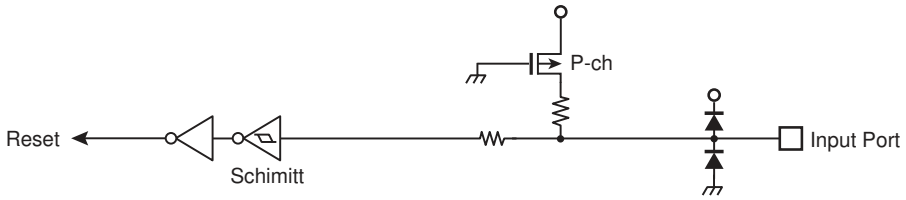
20.6 PK0



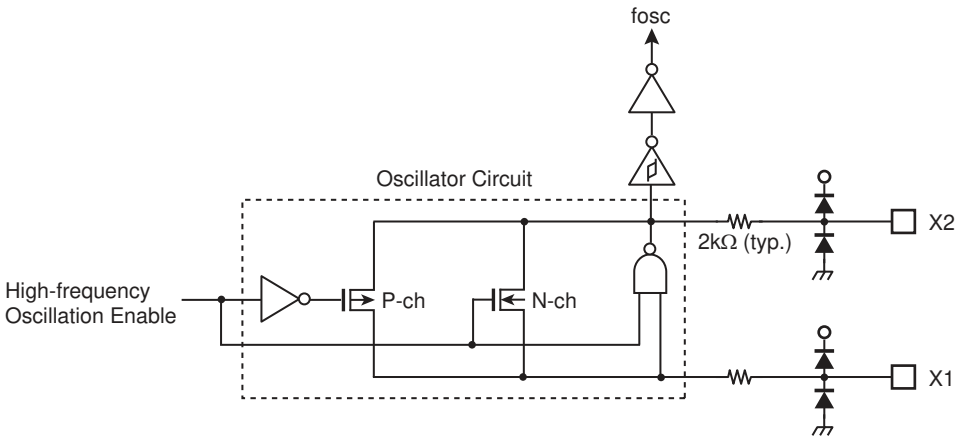
20.7 $\overline{\text{NMI}}$, MODE



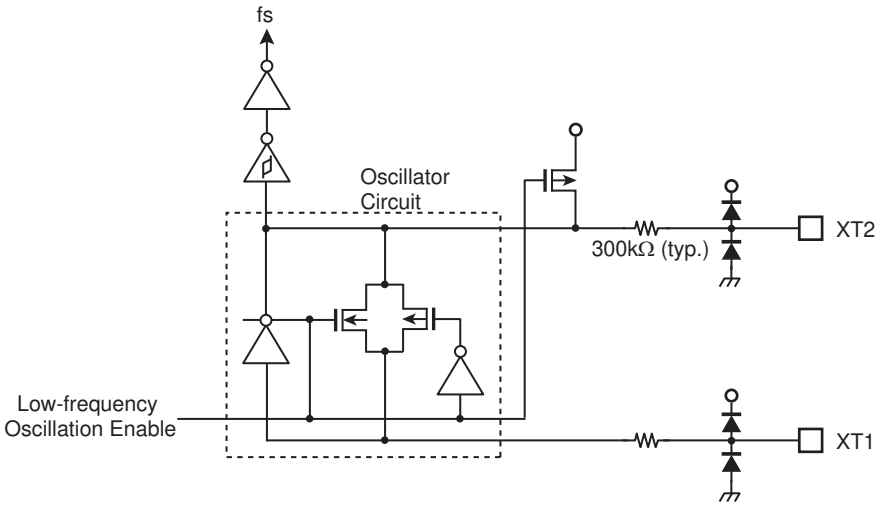
20.8 $\overline{\text{RESET}}$



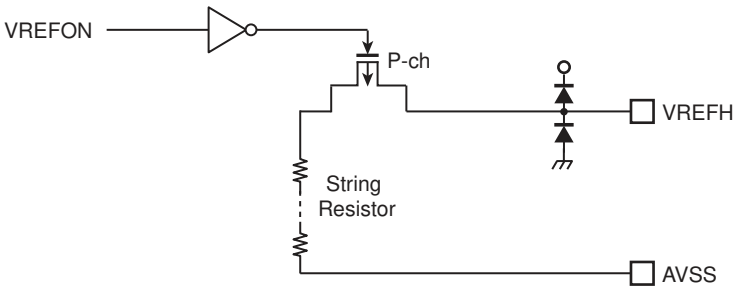
20.9 X1, X2



20.10 XT1, XT2



20.11 VREFH, AVSS

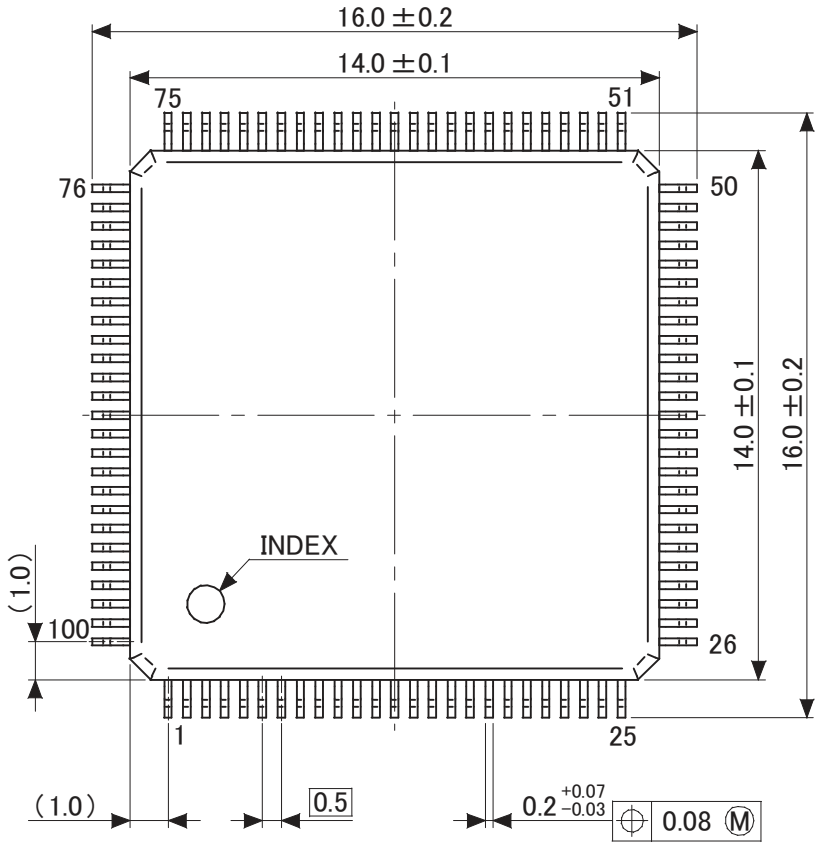


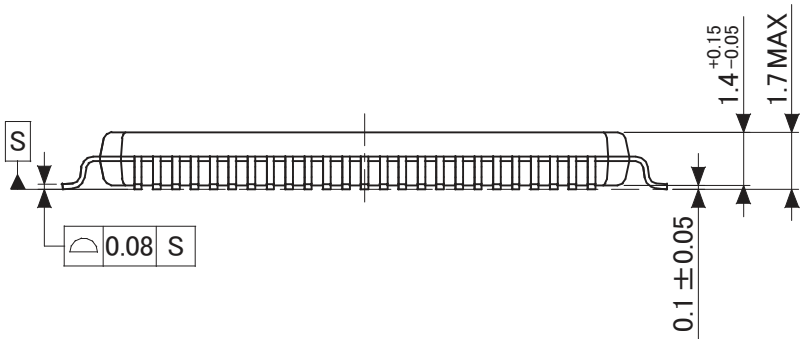
21. Package Dimensions

Type: LQFP100-P-1414-0.50H

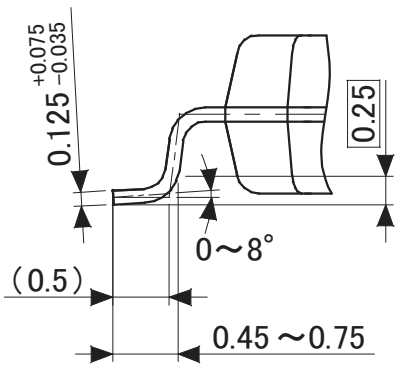
Unit: mm

Dimensions





Pin detail



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 customers use the Product, create designs including the Product, or incorporate the Product into their own applications, 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 with which the 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.