TOSHIBA

32 Bit RISC Microcontroller

TX03 Series

TMPM381/383

TOSHIBA CORPORATION

Semiconductor & Storage Products Company

TOSHIBA

Dear customers

Sep. 01, 2021

Toshiba Electronic Devices & Storage Corporation
Toshiba Electronic Device Solutions Corporation
5801-1, Horikawatyou, Saiwai Ward, Kawasaki City, Kanagawa prefecture, 212-8520
Tel: +81-44-548-2200
Fax: +81-44-548-8965

Error correction for technical datasheet of Universal Asynchronous Receiver-Transmitter

Thank you for using Toshiba microcontrollers.

We have found the mistakes about occurring transmission interrupt timing of the Universal Asynchronous Receiver-Transmitter (UART and FUART) and the Universal Asynchronous Receiver-Transmitter Circuit with 50% duty mode (UART) in our technical datasheet and reference manual. We will inform you about the mistakes in this document.

We apologize for any inconvenience, but we ask that you review the content.

If you have any questions, please contact our sales representative.

1. Applicable products

TMPM342FYXBG
TMPM343F10XBG
TMPM343FDXBG
TMPM366F20AFG
TMPM366FWFG
TMPM366FYFG
TMPM366FDFG
TMPM366FWXBG
TMPM366FYXBG
TMPM366FDXBG
TMPM367FDFG
TMPM367FDXBG
TMPM368FDFG
TMPM368FDXBG
TMPM369FDFG
TMPM369FDXBG
TMPM36BF10FG
TMPM36BFYFG
TMPM381FWDFG
TMPM381FWFG
TMPM383FSEFG
TMPM383FSUG
TMPM383FWEFG
TMPM383FWUG
TMPM3V4FSEFG
TMPM3V4FSUG
TMPM3V4FWEFG
TMPM3V4FWUG
TMPM3V6FWDFG
TMPM3V6FWFG
TMPM440FEXBG
TMPM440F10XBG
TMPM461F10FG
TMPM461F15FG
TMPM462F10FG
TMPM462F15FG
TMPM46BF10FG
TMPM4G6FDFG
TMPM4G6FEFG
TMPM4G6F10FG
TMPM4G7FDFG
TMPM4G7FEFG
TMPM4G7F10FG
TMPM4G8FDFG
TMPM4G8FDXBG
TMPM4G8FEFG
TMPM4G8FEXBG
TMPM4G8F10FG
TMPM4G8F10XBG
TMPM4G8F15FG
TMPM4G8F15XBG
TMPM4G9FDFG
TMPM4G9FDXBG
TMPM4G9FEFG
TMPM4G9FEXBG
TMPM4G9F10FG
TMPM4G9F10XBG
TMPM4G9F15FG
TMPM4G9F15XBG
TMPA900CMXBG
TMPA901CMXBG
TMPA910CRAXBG
TMPA910CRBXBG
TMPA911CRXBG
TMPA912CMXBG
TMPA913CHXBG

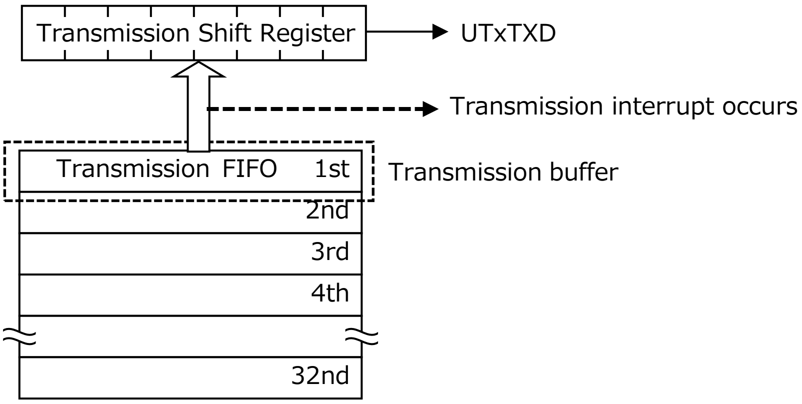
2. Details

The timing of occurring transmission interrupt is shown as below.

There is the mistake in the timing of occurring transmission interrupt when the transmission FIFO is not used only, and it will be corrected as below. There is no mistake in the transmission interrupt timing when using the transmission FIFO.

2.1. When the transmission FIFO is unused

Transmission interrupt occurs when a transmission data moves from the transmission buffer (the 1st level of transmission FIFO) to transmission shift register. (When the transmission buffer becomes empty.)



2.1.1. The timing of occurring transmission interrupt

The transmission interrupt when the transmission FIFO is not used occurs when the transmission buffer becomes empty because it notifies the timing of writing to the transmission buffer for the next data. The transmission interrupt is automatically cleared when the next data is written to the transmission buffer. Therefore, it is not necessary to clear the transmission interrupt by software when continuously transmitting data (set UARTxICR<TXIC> to "1").

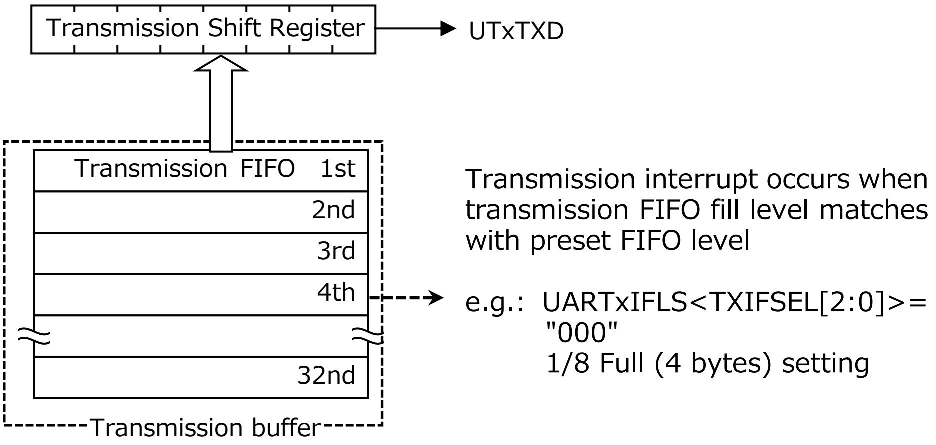
When the transmission is terminated, the final transmission data is transferred to the shift register, and the final transmission interrupt occurs when the transmission buffer becomes empty. If the next data is not written to the transmission buffer, the transmission interrupt can be intentionally cleared by executing clear by software in the interrupt handler (set UARTxICR <TXIC> to "1").

If you execute the transmission interrupt clear by software during data transmission (set UARTxICR <TXIC> to "1"), the transmission interrupt does not occur if you write the data to the transmission buffer at the same time as the STOP bit is generated. In order to generate the transmission interrupt reliably, do not clear the transmission interrupt by software, write data to the transmission buffer during data transmission, or write the data to the transmission buffer while transmission is stopped (when UARTxFR<BUSY>= "0").

When transmitting data continuously, it is recommended to transfer the data by using the transmission FIFO in the next section.

2.2. When transmission FIFO is used

Transmission interrupt occurs when transmission FIFO level matches with preset FIFO level which is specified by UARTxIFLS<TXIFSEL[2:0]>.


Transmission Shift Register
UTxTXD
Transmission FIFO
1st
2nd
3rd
4th
32nd
Transmission buffer
Transmission interrupt occurs when transmission FIFO fill level matches with preset FIFO level
e.g.: UARTxIFLS<TXIFSEL[2:0]>= "000"
1/8 Full (4 bytes) setting

2.2.1. The timing of occurring transmission interrupt

When using the transmission FIFO, the transmission interrupt occurs when transmission FIFO level matches with preset FIFO level.

For example, in case of UARTxIFLS<TXIFSEL[2:0]> = "000" (1/8 full 4 bytes setting), the transmission interrupt occurs when the transmission FIFO level matches with 4th level.

The transmission interrupt is cleared when data whose FIFO level is above the specified FIFO level is stored in the transmission FIFO and occurs again when the specified FIFO level is reached.

3. Description

The description about occurring transmission interrupt is different from each product. The chapter number of placement for each product are shown as below.

There is the mistake in the timing of occurring transmission interrupt when the transmission FIFO is not used only, and it will be corrected as below. There is no mistake in the transmission interrupt timing when using the transmission FIFO.

The details of revised description for the mistake will be explained in "4. Revised description" below, and the revised description is all target products in common.

3.1. Description Type A

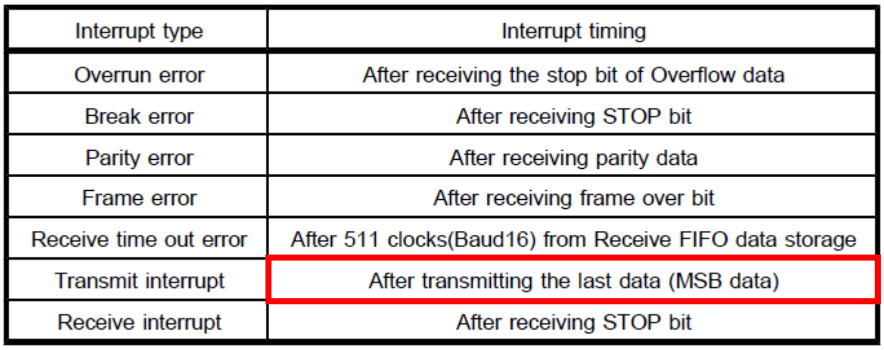
3.1.1. Applicable products and chapter of the description

Product name	Chapter of the description
TMPM342FYXBG	16.4.7
TMPM366F20AFG (Note)	15.4.7
TMPM366FWFG, TMPM366FYFG, TMPM366FDFG, TMPM366FWXBG, TMPM366FYXBG, TMPM366FDXBG	16.4.7
TMPM367FDFG, TMPM367FDXBG, TMPM368FDFG, TMPM368FDXBG, TMPM369FDFG, TMPM369FDXBG,	13.4.7
TMPM36BFYFG, TMPM36BF10FG	13.4.7
TMPA900CMXBG, TMPA901CMXBG, TMPA910CRAXBG, TMPA910CRBXBG, TMPA911CRXBG, TMPA912CMXBG, TMPA913CHXBG	3.13.1.1 (7)

Note: The chapter in a section of the Universal Asynchronous Receiver-Transmitter (UART).

Type A

Original description (Red box)



Interrupt type	Interrupt timing
Overrun error	After receiving the stop bit of Overflow data
Break error	After receiving STOP bit
Parity error	After receiving parity data
Frame error	After receiving frame over bit
Receive time out error	After 511 clocks(Baud16) from Receive FIFO data storage
Transmit interrupt	After transmitting the last data (MSB data)
Receive interrupt	After receiving STOP bit

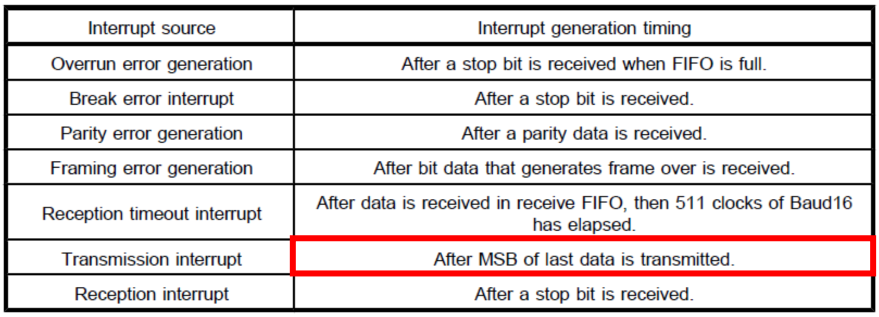
3.2. Description Type B(1)

3.2.1. Applicable products and chapter of the description

Product name	Chapter of the description
TMPM461F10FG, TMPM461F15FG, TMPM462F10FG, TMPM462F15FG	14.4.6.2

Type B(1)

Original description (Red box)



Interrupt source	Interrupt generation timing
Overrun error generation	After a stop bit is received when FIFO is full.
Break error interrupt	After a stop bit is received.
Parity error generation	After a parity data is received.
Framing error generation	After bit data that generates frame over is received.
Reception timeout interrupt	After data is received in receive FIFO, then 511 clocks of Baud16 has elapsed.
Transmission interrupt	After MSB of last data is transmitted.
Reception interrupt	After a stop bit is received.

3.3. Description Type B(2)

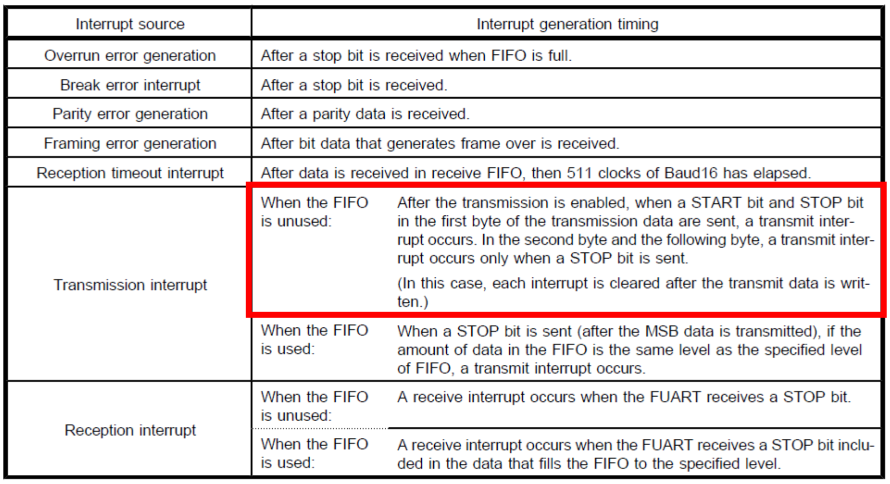
3.3.1. Applicable products and chapter of the description

Product name	Chapter of the description
TMPM343FDXBG, TMPM343F10XBG, TMPM366F20AFG (Note)	16.4.6.2
TMPM381FWFG, TMPM381FWDFG, TMPM383FSUG, TMPM383FSEFG, TMPM383FWUG, TMPM383FWEFG, TMPM3V4FSUG, TMPM3V4FSEFG, TMPM3V4FWUG, TMPM3V4FWEFG, TMPM3V6FWFG, TMPM3V6FWDFG	11.4.6.2
TMPM440FEXBG, TMPM440F10XBG	26.4.6.2

Note: The chapter in a section of the Universal Asynchronous Receiver-Transmitter Circuit with 50% duty mode (UART).

Type B(2)

Original description (Red box)



Interrupt source	Interrupt generation timing	
Overrun error generation	After a stop bit is received when FIFO is full.	
Break error interrupt	After a stop bit is received.	
Parity error generation	After a parity data is received.	
Framing error generation	After bit data that generates frame over is received.	
Reception timeout interrupt	After data is received in receive FIFO, then 511 clocks of Baud16 has elapsed.	
Transmission interrupt	When the FIFO is unused:	After the transmission is enabled, when a START bit and STOP bit in the first byte of the transmission data are sent, a transmit interrupt occurs. In the second byte and the following byte, a transmit interrupt occurs only when a STOP bit is sent. (In this case, each interrupt is cleared after the transmit data is written.)
	When the FIFO is used:	When a STOP bit is sent (after the MSB data is transmitted), if the amount of data in the FIFO is the same level as the specified level of FIFO, a transmit interrupt occurs.
Reception interrupt	When the FIFO is unused:	A receive interrupt occurs when the FUART receives a STOP bit.
	When the FIFO is used:	A receive interrupt occurs when the FUART receives a STOP bit included in the data that fills the FIFO to the specified level.

3.4. Description Type B(3)

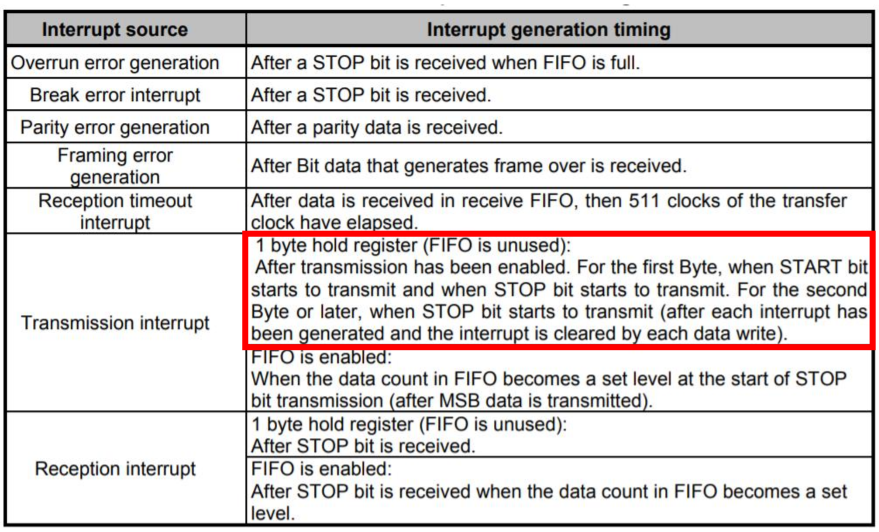
3.4.1. Applicable products and chapter of the description

Product name	Chapter of the description
TMPM4G6FDFG, TMPM4G6FEFG, TMPM4G6F10FG, TMPM4G7FDFG, TMPM4G7FEFG, TMPM4G7F10FG, TMPM4G8FDFG, TMPM4G8FDXBG, TMPM4G8FEFG, TMPM4G8FEXBG, TMPM4G8F10FG, TMPM4G8F10XBG, TMPM4G8F15FG, TMPM4G8F15XBG, TMPM4G9FDFG, TMPM4G9FDXBG, TMPM4G9FEFG, TMPM4G9FEXBG, TMPM4G9F10FG, TMPM4G9F10XBG, TMPM4G9F15FG, TMPM4G9F15XB	Reference Manual (Note) Full Universal Asynchronous Receiver Transmitter Circuit (FUART-B) 3.8.2

Note: In this reference manual, read UARTxIFLS with ***[FURTxIFLS]***, UARTxICR with ***[FURTxICR]***, UARTxFR with ***[FURTxFR]***.

Type B(3)

Original description (Red box)



Interrupt source	Interrupt generation timing
Overrun error generation	After a STOP bit is received when FIFO is full.
Break error interrupt	After a STOP bit is received.
Parity error generation	After a parity data is received.
Framing error generation	After Bit data that generates frame over is received.
Reception timeout interrupt	After data is received in receive FIFO, then 511 clocks of the transfer clock have elapsed.
Transmission interrupt	1 byte hold register (FIFO is unused): After transmission has been enabled. For the first Byte, when START bit starts to transmit and when STOP bit starts to transmit. For the second Byte or later, when STOP bit starts to transmit (after each interrupt has been generated and the interrupt is cleared by each data write).
	FIFO is enabled: When the data count in FIFO becomes a set level at the start of STOP bit transmission (after MSB data is transmitted).
Reception interrupt	1 byte hold register (FIFO is unused): After STOP bit is received.
	FIFO is enabled: After STOP bit is received when the data count in FIFO becomes a set level.

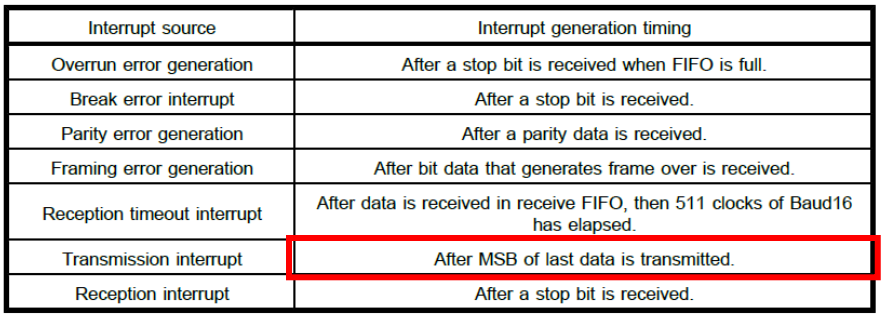
3.5. Description Type C

3.5.1. Applicable products and chapter of the description.

Product name	Chapter of the description
TMPM46BF10FG	19.4.6.2

Type C

Original description (Red box)



Interrupt source	Interrupt generation timing
Overrun error generation	After a stop bit is received when FIFO is full.
Break error interrupt	After a stop bit is received.
Parity error generation	After a parity data is received.
Framing error generation	After bit data that generates frame over is received.
Reception timeout interrupt	After data is received in receive FIFO, then 511 clocks of Baud16 has elapsed.
Transmission interrupt	After MSB of last data is transmitted.
Reception interrupt	After a stop bit is received.

4. Revised description

The description of the transmission interrupt occurrence timing differs depending on the products, but the correct description is as follows in common.

4.1. The timing of occurring transmission interrupt

The transmission interrupt when the transmission FIFO is not used occurs when the transmission buffer becomes empty because it notifies the timing of writing to the transmission buffer for the next data. The transmission interrupt is automatically cleared when the next data is written to the transmission buffer. Therefore, it is not necessary to clear the transmission interrupt by software when continuously transmitting data (set UARTxICR<TXIC> to "1").

When the transmission is terminated, the final transmission data is transferred to the shift register, and the final transmission interrupt occurs when the transmission buffer becomes empty. If the next data is not written to the transmission buffer, the transmission interrupt can be intentionally cleared by executing clear by software in the interrupt handler (set UARTxICR <TXIC> to "1").

If you execute the transmission interrupt clear by software during data transmission (set UARTxICR <TXIC> to "1"), the transmission interrupt does not occur if you write the data to the transmission buffer at the same time as the STOP bit is generated. In order to generate the transmission interrupt reliably, do not clear the transmission interrupt by software, write data to the transmission buffer during data transmission, or write the data to the transmission buffer while UART transmission is stopped (when UARTxFR<BUSY> = "0").

End of document

General precautions on the use of Toshiba MCUs

This Page explains general precautions on the use of Toshiba MCUs.

Note that if there is a difference between the general precautions and the description in the body of the document, the description in the body of document has higher priority.

1. The MCUs' operation at power-on

   At power-on, internal state of the MCUs is unstable. Therefore, state of the pins is undefined until reset operation is completed.

   When a reset is performed by an external reset pin, pins of the MCUs that use the reset pin are undefined until reset operation by the external pin is completed.

   Also, when a reset is performed by the internal power-on reset, pins of the MCUs that use the internal power-on reset are undefined until power supply voltage reaches the voltage at which power-on reset is valid.

2. Unused pins

   Unused input/output ports of the MCUs are prohibited to use. The pins are high-impedance.

   Generally, if MCUs operate while the high-impedance pins left open, electrostatic damage or latch-up may occur in the internal LSI due to induced voltage influenced from external noise.

   Toshiba recommend that each unused pin should be connected to the power supply pins or GND pins via resistors.

3. Clock oscillation stability

   A reset state must be released after the clock oscillation becomes stable. If the clock is changed to another clock while the program is in progress, wait until the clock is stable.

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

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

The SFR addresses 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 description

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

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

d. Register description

Registers are described as shown below.

- Register name <Bit Symbol>

  Example: 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
2015/12/25	1	First Release
2016/2/3	2	Contents Revised

Table of Contents

General precautions on the use of Toshiba MCUs

TMPM381/383

2. Product Information

3. Processor Core

4. Memory Map

5. Reset Operation

6. Clock/Mode control

7. Exceptions

8. Digital Noise Filter Circuit (DNF)

9. Input / Output port

10. 16-bit Timer / Event Counters (TMRB)

11. Universal Asynchronous Receiver-Transmitter Circuit (UART)

12. Serial Channel with 4bytes FIFO (SIO/UART)

13. Serial Bus Interface (I2C/SIO)

14. Synchronous Serial Port (SSP)

15. Remote Control Signal Preprocessor (RMC)

16. Analog/Digital Converter (ADC)

17. Real Time Clock (RTC)

18. Power-on-Reset Circuit (POR)

19. Low Voltage Detection Circuit (VLTD)

20. Oscillation Frequency Detector (OFD)

21. Watchdog Timer(WDT)

22. Flash Memory Operation

23. Debug Interface

24. Port Section Equivalent Circuit Schematic

25. Electrical Characteristics

26. Package Dimensions

TMPM381/383

The TMPM381/383 is a 32-bit RISC microprocessor with an ARM® Cortex® -M3 microprocessor core.

Features of the TMPM381/383 are shown as follows:

Product name	ROM (FLASH)	RAM	Package
TMPM381FWFG	128Kbyte	10Kbyte	LQFP100 (14x14mm, 0.5mm pitch)
TMPM381FWDFG	128 Kbyte	10Kbyte	QFP100 (14x20mm, 0.65mm pitch)
TMPM383FWUG	128 Kbyte	10Kbyte	LQFP64 (10x10mm, 0.5mm pitch)
TMPM383FWEFG	128 Kbyte	10Kbyte	QFP64 (14x14mm, 0.8mm pitch)
TMPM383FSUG	64 Kbyte	8Kbyte	LQFP64 (10x10mm, 0.5mm pitch)
TMPM383FSEFG	64 Kbyte	8Kbyte	QFP64 (14x14mm, 0.8mm pitch)

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 bits) can be executed with one clock.

      A division takes within 2 to 12 cycles

      [Low power consumption]

      Optimized design using a low power consumption library

      Standby function that stops the operation of the microcontroller 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 and data memory
   - On-chip Flash ROM

     TMPM381FWFG, TMPM381FWDFG : 128Kbyte

     TMPM383FWUG, TMPM383FWEFG : 128Kbyte

     TMPM383FSUG, TMPM383FSEFG : 64Kbyte

   - On-chip RAM

     TMPM381FWFG, TMPM381FWDFG : 10Kbyte

     TMPM383FWUG, TMPM383FWEFG : 10Kbyte

TMPM383FSUG, TMPM383FSEFG : 8Kbyte

3. Clock controller (CG)
   - Built-in Oscillator (9MHz)
   - External Oscillator (High Frequency 10MHz)
   - External Oscillator (Low Frequency 32.768kHz)
   - Installed 1 unit of Built-in PLL. (4x)
   - Clock gear function: divides high-speed clock into 1/1, 1/2, 1/4, 1/8 or 1/16.

4. Low power consumption function
   - IDLE, SLEEP, STOP,SLOW

5. External interrupt source
   - External interrupt pin

     TMPM381FWFG, TMPM381FWDFG : 16 pin

     TMPM383FWUG, TMPM383FWEFG : 8 pin

     TMPM383FSUG, TMPM383FSEFG : 8 pin
   - 7-level priority setting

6. Input/ output ports (PORT)
   - Input/Output pin

     TMPM381FWFG, TMPM381FWDFG : 83 pin

     TMPM383FWUG, TMPM383FWEFG : 47 pin

     TMPM383FSUG, TMPM383FSEFG : 47 pin
   - Output pin

     TMPM381FWFG, TMPM381FWDFG : 1 pin

     TMPM383FWUG, TMPM383FWEFG : 1 pin

     TMPM383FSUG, TMPM383FSEFG : 1 pin

7. 16-bit timer (TMRB) : 8 channels
   - 16-bit interval timer mode
   - 16-bit event counter mode
   - 16-bit PPG output
   - External trigger 16-bit programmable square-wave output mode (PPG)
   - Timer synchronous mode
   - Input capture function

8. Real time clock (RTC): 1 channel
   - Clock (hour, minute and second)
   - Calendar (month, week, date and leap year)
   - Clock adjustment function
   - +/-30s adjustment function
   - Alarm output
   - 1Hz clock output

9. Watchdog timer (WDT) : 1 channel

Reset or non-maskable interrupt (NMI) are occurred.

10. Serial channel (SIO/UART)

    TMPM381FWFG, TMPM381FWDFG : 3 channel

    TMPM383FWUG, TMPM383FWEFG : 2 channel

    TMPM383FSUG, TMPM383FSEFG : 2 channel

    - Selectable either UART or synchronous mode (4byte FIFO equipped)

11. Asynchronous serial communication interface (UART): 1 channel
    - Data length: 5, 6, 7, 8 bits
    - Transmit FIFO: 32-stage 8-bit width, transmit FIFO: 32-stage 12-bit width
    - 50% Duty mode

12. Serial bus interface (I2C/SIO) : 1 channels
    - Either I2C bus mode or synchronous mode can be selected.

13. Synchronous Serial interface (SSP) : 1 channel
    - Communication protocol that includes SPI: 3 types (SPI/SSI/Microwire)
    - 16byte FIFO equipped (8-bit width, 8 depth)

14. 12-bit AD converter (ADC): 1 unit

    TMPM381FWFG, TMPM381FWDFG : 18 channel

    TMPM383FWUG, TMPM383FWEFG : 10 channel

    TMPM383FSUG, TMPM383FSEFG : 10 channel

    - Fixed channel / Channel scan mode
    - Single / repeat mode
    - External trigger start and internal timer trigger start are possible.
    - Repeat conversion is capable.
    - AD monitoring
    - Minimum conversion time: 2 μs (ADC conversion clock 40 MHz)

15. Remote control signal preprocessor (RMC): 1 channel
    - Can receive up to 72-bit data at a time

16. Power_On reset function (POR) : 1 unit

17. VLTD function : 1 unit

18. Oscillation frequency detector (OFD) : 1channel

19. Debug interface
    - JTAG/SWD/SWV/DATA TRACE (Data 2 bits) are supported.

20. Endian
    - Little-endian

21. Maximum operating frequency : 40MHz

22. Operating voltage range
    - 4.5V to 5.5V (all function operation)
    - 3.9V to 4.5V (restrictions of 12-bit ADC)

23. Temperature range
    - −40°C to 85°C (except during Flash writing/erasing and debugging)
    - 0°C to 70°C (during Flash writing/erasing and debugging)

    12/2

24. Package

    TMPM381FWFG : LQFP100 (14mm x 14mm, 0.5mm pitch)

    TMPM381FWDFG : QFP100 (14mm x 20mm, 0.65mm pitch)

    TMPM383FWUG : LQFP64 (10mm x 10mm, 0.5mm pitch)

    TMPM383FWEFG : QFP64(14mm x 14mm, 0.65mm pitch)

    TMPM383FSUG : LQFP64 (10mm x 10mm, 0.5mm pitch)

    TMPM383FSEFG : QFP64(14mm x 14mm, 0.65mm pitch)

1.2 Block Diagram

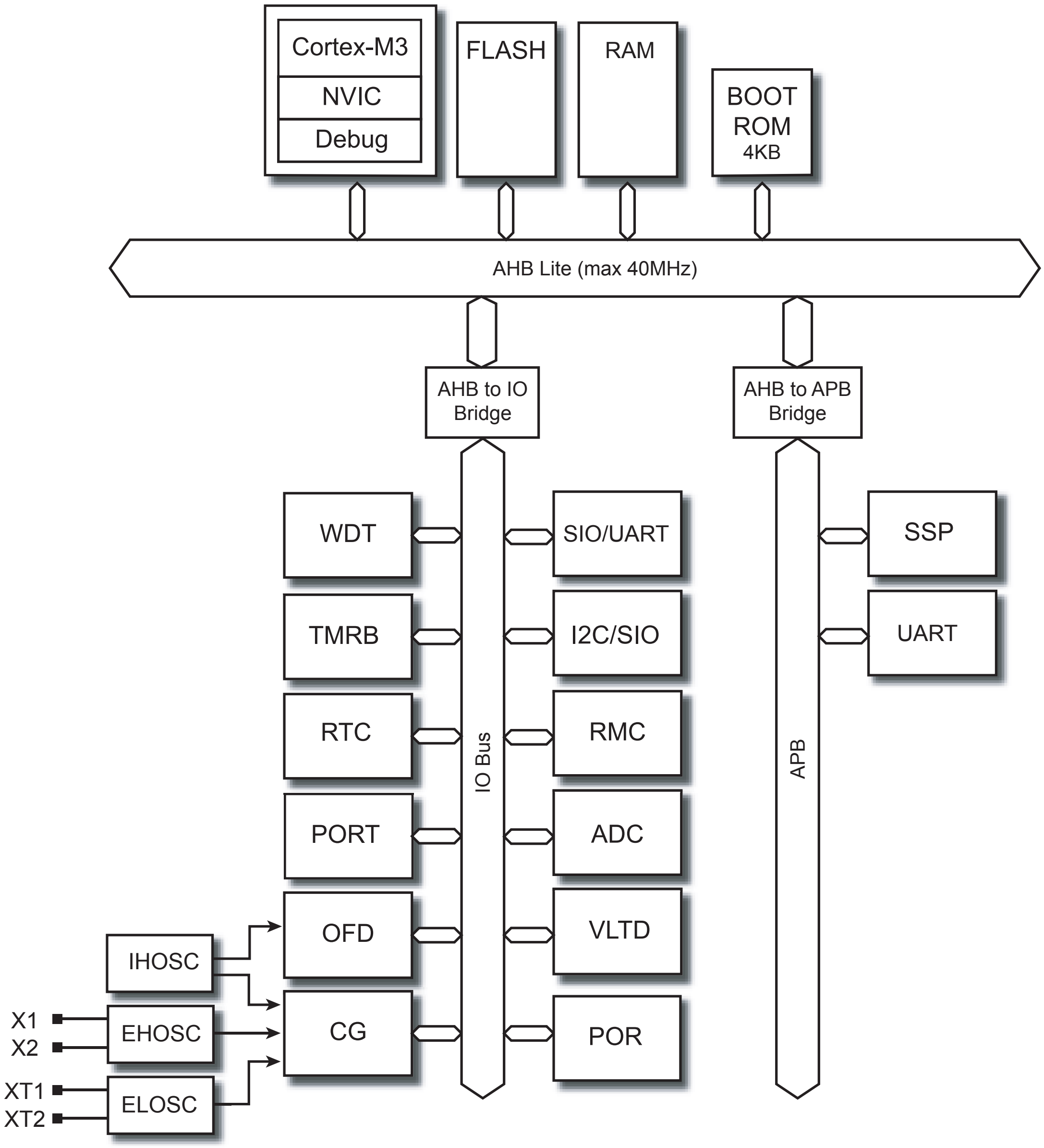


Figure 1-1 Block Diagram

1.3 Pin Layout (Top view)

1.3.1 TMPM381FWFG

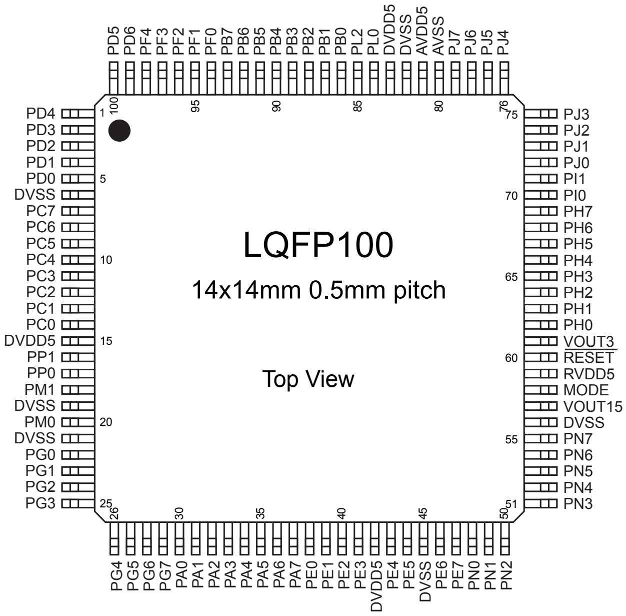


Figure 1-2 Pin Layout (LQFP100 14x14mm TOP VIEW)

1.3.2 TMPM381FWDFG

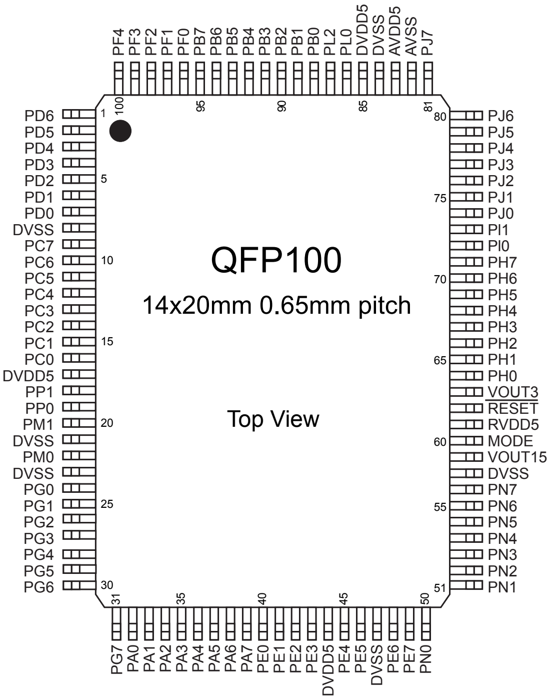


Figure 1-3 Pin Layout (QFP100 14x20mm TOP VIEW)

1.3.3 TMPM383FWUG, TMPM383FSUG

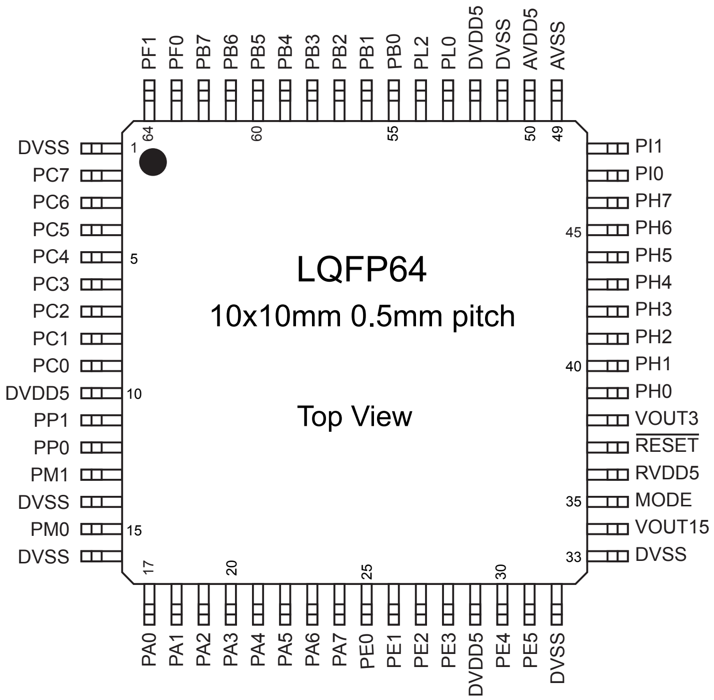


Figure 1-4 Pin Layout (LQFP64 10x10mm TOP VIEW)

1.3.4 TMPM383FWEFG, MPM383FSEFG

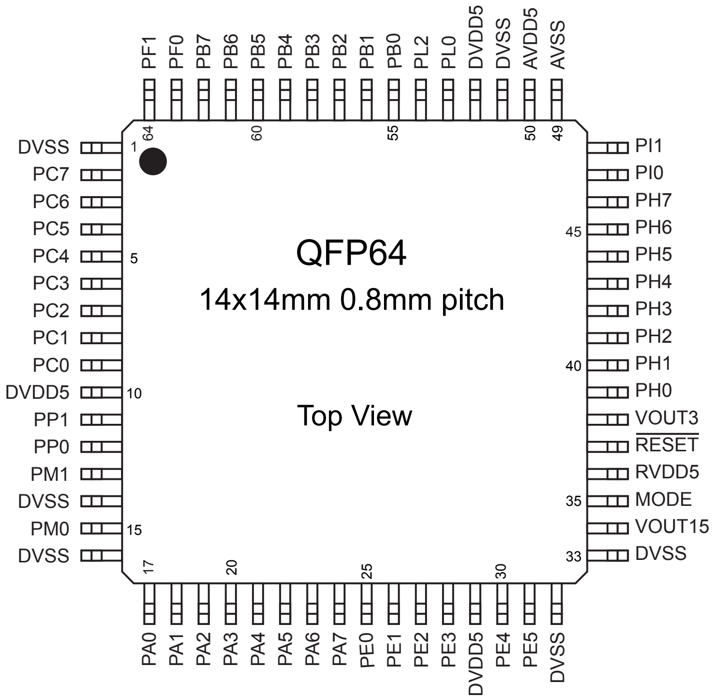


Figure 1-5 Pin Layout (QFP64 14x14mm TOP VIEW)

1.4 Pin names and Functions

1.4.1 Pin names and Functions for each peripheral function, control pin and power supply pin

1.4.1.1 Peripheral functions

Table 1-1 The number of pins and Pin names

Peripheral function	Pin name	Input or Output	Function
Clock / Mode control	SCOUT	Output	System clock output
External interrupt	INTx	Input	External interrupt input pin x External interrupt input pin x has a noise filter (Filter width 30ns typ.).
16 bit timer / even counter	TBxIN	Input	Input capture input pin
	TBxOUT	Output	output pin
SIO/UART	TXDx	Output	Data output pin
	RXDx	Input	Data input pin
	SCLKx	I/O	Clock input / output pin
	$\overline{\text{CTSx}}$	Input	Hand shake input pin
UART	UTxTXD	Output	Data output pin
	UTxTXD50A	Output	Data output pin
	UTxTXD50B	Output	Data output pin
	UTxRXD	Input	Data input pin
	UTxRXD50	Input	Data input pin
I2C/SIO	SDAx	I/O	Data input / output pin (I2C bus mode)
	SOx	Output	Data output pin (Clock-synchronous 8-bit SIO mode)
	SCLx	I/O	Clock input / output pin (I2C bus mode)
	SIx	Input	Data input pin (Clock-synchronous 8-bit SIO mode)
	SCKx	I/O	Clock input / output pin (Clock-synchronous 8-bit SIO mode)
SSP	SPxDO	Output	Data output pin
	SPxDI	Input	Data input pin
	SPxCLK	I/O	Clock input / output pin
	SPxFSS	I/O	Frame / slave select input / output pin
Remote control signal preprocessor	RXINx	Input	Data input pin
Analog digital convertor	AINx	Input	Analog input pin
Real time clock	$\overline{\text{ALARM}}$	Output	ALARM output pin

Note: x : Channel number

1.4.1.2 Debug function

Table 1-2 Pin name and functions

Pin name	Input or Output	Function
TMS	Input	JTAG test mode select input pin
TCK	Input	JTAG serial clock input pin
TDO	Output	JTAG serial data output pin
TDI	Input	JTAG serial data input pin
$\overline{\text{TRST}}$	Input	JTAG test reset input pin
SWDIO	I/O	Serial wire data input / output pin
SWCLK	Input	Serial wire clock input pin
SWV	Output	Serial wire viewer output pin
TRACECLK	Output	Trace clock output pin
TRACEDATA0	Output	Trace data output pin 0
TRACEDATA1	Output	Trace data output pin 1

1.4.1.3 Control function

Table 1-3 Pin name and functions

Pin name	Input or Output	Function
X1	Input	High frequency resonator connection pin
X2	Output	High frequency resonator connection pin
XT1	Input	Low frequency resonator connection pin
XT2	Output	Low frequency resonator connection pin
MODE	Input	MODE pin This pin must be fixed to Low level.
$\overline{\text{RESET}}$	Input	Reset signal input pin
$\overline{\text{BOOT}}$	Input	BOOT mode control pin BOOT mode control pin is sampled at the rising edge of reset signal input pin. TMPM381/383 changes Single Boot Mode when BOOT mode control pin is "Low". TMPM381/383 changes Single Chip Mode when BOOT mode control pin is "High". Refer to "Flash memory" section for a detail.

1.4.1.4 Power supply pins

Table 1-4 Pin name and functions

Power supply pin name	Function
VOUT15	Pin connected with the capacitor (4.7μF) for the regulator
VOUT3	Pin connected with the capacitor (4.7μF) for the regulator
RVDD5	Power supply pin for the regulator
DVDD5	Power supply pin for the digital circuit DVDD5 supplies the following pins. PA,PB,PC,PD,PE,PF,PG,PL,PM,PN,PP, MODE, $\overline{\text{RESET}}$, $\overline{\text{BOOT}}$
DVSS	GND pin for the digital circuit
AVDD5	Power supply pin for the analog circuit AVDD5 supplies the following pins. PH, PI, PJ
AVSS	GND pin for ADC

1.4.1.5 Capacitors between power supply pins

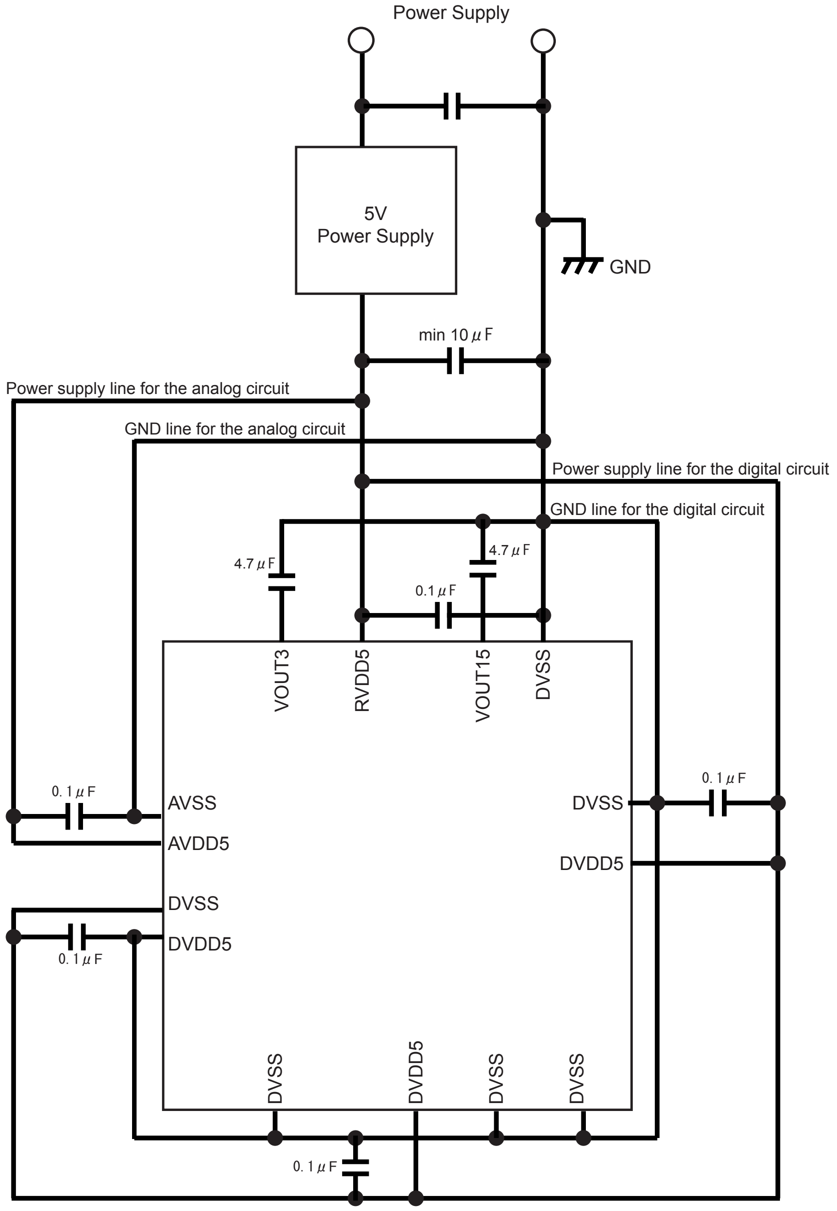


Figure 1-6 Capacitors for power supply pins connection circuit

Note 1: Place an output capacitor (at least 10 μF) between RVDD5 and DVSS at the closest position between them.

Note 2: At the closest position between RVDD5 and DVSS (mentioned above), separate a 5V analog power supply line from a 5V digital power supply line in the close vicinity of this capacitor. If the unseparated line is long, due to common impedance coupling, electrical power fluctuation is conducted to analog power supply from dig-ital power supply. This causes the noise of the analog circuit.

Note 3: When wiring power supply and GND lines, they must be placed close. If not, a power supply and GND line make a power supply loop through the capacitor of the power supply circuit. They will be the antenna receiving high frequency noise.

Note 4: Capacitors of VOUT3 and VOUT15 for regulators must be the same capacity, and they must be placed at the shortest distance from 56 pin DVSS.

Note 5: Place a capacitor between every DVDD5 and every DVSS at the closest distance.

Note 6: Place a capacitor between AVDD5 and AVSS at the closest distance.

1.4.2 Pin names and Function of TMPM381/383

1.4.2.1 The detail for pin names and function list

The mean of the symbol in the table is shown below.

1. Function A

   The function which is specified without setting of function register is shown in this cell.

2. Function B

   The function which is specified with setting of function register is shown in this cell. The number in this cell is corresponded with the number of function register.

3. Pin specification

   The mean of the symbol in the table is shown below.

   - SMT/CMOS : Type of input gate

     SMT : Schmitt input

     CMOS : CMOS input

   - OD : Programmable open drain output support

     Yes : supported

     N/A : Not supported

   - PU/PD : Programmable Pull-Up / Pull-Down

     PU : Programmable Pull-Up supported

     PD : Programmable Pull-Down supported

   - Pin No.

     LQFP100 : TMPM381FWFG

     QFP100 : TMPM381FWDFG

     LQFP64 : TMPM383FWUG, TMPM383FSUG

     QFP64 : TMPM383FWEFG, TMPM383FSEFG

1.4.2.2 PORT / Debug pin

Table 1-5 Pin names and functions <Sorted by PORT> (1/5)

Pin No.			PORT	Function A	Function B					Port Specification		
LQFP 100	QFP 100	LQFP64 QFP64			1	2	3	4	5	PU/ PD	OD	SMT/ CMOS
PORT A												
30	32	17	PA0		TB0IN	INT3				PU/ PD	Yes	SMT
31	33	18	PA1		TB0OUT	SCOUT				PU/ PD	Yes	SMT
32	34	19	PA2		TB1IN	INT4				PU/ PD	Yes	SMT
33	35	20	PA3		TB1OUT	RXIN0				PU/ PD	Yes	SMT
34	36	21	PA4		SCLK1	$\overline{\text{CTS1}}$				PU/ PD	Yes	SMT
35	37	22	PA5		TXD1	TB6OUT				PU/ PD	Yes	SMT
36	38	23	PA6		RXD1	TB6IN				PU/ PD	Yes	SMT
37	39	24	PA7		TB4IN	INT8				PU/ PD	Yes	SMT
PORT B												
86	88	55	PB0		TRACECLK					PU/ PD	Yes	SMT
87	89	56	PB1		TRACEDATA0					PU/ PD	Yes	SMT
88	90	57	PB2		TRACEDATA1					PU/ PD	Yes	SMT
89	91	58	PB3		TMS/SWDIO					PU/ PD	Yes	SMT
90	92	59	PB4		TCK/SWCLK					PU/ PD	Yes	SMT
91	93	60	PB5		TDO/SWV					PU/ PD	Yes	SMT
92	94	61	PB6		TDI					PU/ PD	Yes	SMT
93	95	62	PB7		$\overline{\text{TRST}}$					PU/ PD	Yes	SMT

Table 1-6 Pin names and functions <Sorted by PORT> (2/5)

Pin No.			PORT	Function A	Function B					Port Specification		
LQFP 100	QFP 100	LQFP64 QFP64			1	2	3	4	5	PU/ PD	OD	SMT/ CMOS
PORT C												
14	16	9	PC0			SP0DO	SDA0/SO0			PU/ PD	Yes	SMT
13	15	8	PC1			SP0DI	SCL0/SI0			PU/ PD	Yes	SMT
12	14	7	PC2			SP0CLK	SCK0			PU/ PD	Yes	SMT
11	13	6	PC3			SP0FSS				PU/ PD	Yes	SMT
10	12	5	PC4							PU/ PD	Yes	SMT
9	11	4	PC5						UT0TXD50 B	PU/ PD	Yes	SMT
8	10	3	PC6					UT0TXD	UT0TXD50 A	PU/ PD	Yes	SMT
7	9	2	PC7					UT0RXD	UT0RXD50	PU/ PD	Yes	SMT
PORT D												
5	7	-	PD0			TB5IN	INTC			PU/ PD	Yes	SMT
4	6	-	PD1			TB5OUT				PU/ PD	Yes	SMT
3	5	-	PD2				INTD			PU/ PD	Yes	SMT
2	4	-	PD3		INT9					PU/ PD	Yes	SMT
1	3	-	PD4		SCLK2	$\overline{\text{CTS2}}$				PU/ PD	Yes	SMT
100	2	-	PD5		TXD2					PU/ PD	Yes	SMT
99	1	-	PD6		RXD2					PU/ PD	Yes	SMT
PORT E												
38	40	25	PE0		TXD0					PU/ PD	Yes	SMT
39	41	26	PE1		RXD0					PU/ PD	Yes	SMT
40	42	27	PE2		SCLK0	$\overline{\text{CTS0}}$				PU/ PD	Yes	SMT
41	43	28	PE3		TB4OUT					PU/ PD	Yes	SMT
43	45	30	PE4		TB2IN	INT5				PU/ PD	Yes	SMT
44	46	31	PE5		TB2OUT					PU/ PD	Yes	SMT
46	48	-	PE6		TB3IN	INT6				PU/ PD	Yes	SMT
47	49	-	PE7		TB3OUT	INT7				PU/ PD	Yes	SMT

Table 1-7 Pin names and functions <Sorted by PORT> (3/5)

Pin No.			PORT	Function A	Function B					Port Specification		
LQFP 100	QFP 100	LQFP64 QFP64			1	2	3	4	5	PU/ PD	OD	SMT/ CMOS
PORT F												
94	96	63	PF0		TB7IN					PU/ PD	Yes	SMT
95	97	64	PF1		TB7OUT	$\overline{\text{ALARM}}$				PU/ PD	Yes	SMT
96	98	-	PF2							PU/ PD	Yes	SMT
97	99	-	PF3							PU/ PD	Yes	SMT
98	100	-	PF4							PU/ PD	Yes	SMT
PORT G												
22	24	-	PG0							PU/ PD	Yes	SMT
23	25	-	PG1							PU/ PD	Yes	SMT
24	26	-	PG2							PU/ PD	Yes	SMT
25	27	-	PG3							PU/ PD	Yes	SMT
26	28	-	PG4							PU/ PD	Yes	SMT
27	29	-	PG5							PU/ PD	Yes	SMT
28	30	-	PG6							PU/ PD	Yes	SMT
29	31	-	PG7							PU/ PD	Yes	SMT
PORT H												
62	64	39	PH0	AIN0	INT0					PU/ PD	Yes	SMT
63	65	40	PH1	AIN1	INT1					PU/ PD	Yes	SMT
64	66	41	PH2	AIN2	INT2					PU/ PD	Yes	SMT
65	67	42	PH3	AIN3						PU/ PD	Yes	SMT
66	68	43	PH4	AIN4						PU/ PD	Yes	SMT
67	69	44	PH5	AIN5						PU/ PD	Yes	SMT
68	70	45	PH6	AIN6						PU/ PD	Yes	SMT
69	71	46	PH7	AIN7						PU/ PD	Yes	SMT

Table 1-8 Pin names and functions <Sorted by PORT> (4/5)

Pin No.			PORT	Function A	Function B					Port Specification		
LQFP 100	QFP 100	LQFP64 QFP64			1	2	3	4	5	PU/ PD	OD	SMT/ CMOS
PORT I												
70	72	47	PI0	AIN8						PU/ PD	Yes	SMT
71	73	48	PI1	AIN9						PU/ PD	Yes	SMT
PORT J												
72	74	-	PJ0	AIN10						PU/ PD	Yes	SMT
73	75	-	PJ1	AIN11						PU/ PD	Yes	SMT
74	76	-	PJ2	AIN12						PU/ PD	Yes	SMT
75	77	-	PJ3	AIN13						PU/ PD	Yes	SMT
76	78	-	PJ4	AIN14						PU/ PD	Yes	SMT
77	79	-	PJ5	AIN15						PU/ PD	Yes	SMT
78	80	-	PJ6	AIN16	INTA					PU/ PD	Yes	SMT
79	81	-	PJ7	AIN17	INTB					PU/ PD	Yes	SMT
PORT L												
84	86	53	PL0	$\overline{\text{BOOT}}$						PU/ PD	Yes	SMT
85	87	54	PL2		INTF					PU/ PD	Yes	SMT
PORT M												
20	22	15	PM0	X1						PU/ PD	Yes	SMT
18	20	13	PM1	X2						PU/ PD	Yes	SMT

Table 1-9 Pin names and functions <Sorted by PORT> (5/5)

Pin No.			PORT	Function A	Function B					Port Specification		
LQFP 100	QFP 100	LQFP64 QFP64			1	2	3	4	5	PU/ PD	OD	SMT/ CMOS
PORT N												
48	50	-	PN0							PU/ PD	Yes	SMT
49	51	-	PN1							PU/ PD	Yes	SMT
50	52	-	PN2							PU/ PD	Yes	SMT
51	53	-	PN3							PU/ PD	Yes	SMT
52	54	-	PN4							PU/ PD	Yes	SMT
53	55	-	PN5							PU/ PD	Yes	SMT
54	56	-	PN6							PU/ PD	Yes	SMT
55	57	-	PN7			INTE				PU/ PD	Yes	SMT
PORT P												
17	19	12	PP0	XT1						PU/ PD	Yes	SMT
16	18	11	PP1	XT2						PU/ PD	Yes	SMT

1.4.2.3 Control pin

Table 1-10 The number of pin and pin names

Pin No.			Control function Pin name
LQFP 100	QFP 100	LQFP64 QFP64	
20	22	15	X1
18	20	13	X2
17	19	12	XT1
16	18	11	XT2
58	60	35	MODE
60	62	37	$\overline{\text{RESET}}$
84	86	53	$\overline{\text{BOOT}}$

1.4.2.4 Power Supply pin

Table 1-11 The number of pin and pin names

Pin No.			Power supply Pin name
LQFP 100	QFP 100	LQFP64 QFP64	
59	61	36	RVDD5
57	59	34	VOUT15
61	63	38	VOUT3
15,42,83	17,44,85	10,29,52	DVDD5
6,19,21,45,56,82	8,21,23,47,58,84	1,14,16,32,33,51	DVSS
81	83	50	AVDD5
80	82	49	AVSS

2. Product Information

This chapter describes peripheral function-related channels or number of units, information of pins and product-specific function information. Use this chapter in conjunction with Chapter Peripheral Function.

2.1 Built-in Functions of the M381 and M383

The table below shows the differences of the built-in functions between the M381 and M383.

Table 2-1 Functional comparison

Function		TMPM381FWFG TMPM381FWDFG	TMPM383FWUG/TMPM383FSFG TMPM383FWEFG/TMPM383FSEFG
Package		LQFP100-P-1414-0.5H (TMPM381FWFG) QFP100-P-1420-0.65A (TMPM381FWDFG)	LQFP64-P-1010-0.50E (TMPM383FWUG/TMPM383FSFG) QFP64-P-1414-0.80A (TMPM383FWEFG/TMPM383FSEFG)
Flash / RAM		128KB /10KB	128KB/10KB (TMPM383FWUG/TMPM383FWEFG) 64KB/8KB (TMPM383FSFG/TMPM383FSEFG)
Interrupt	Internal	47	45 (INTRX2 and INTTX2 are not available.)
	External	16 (INT0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)	8 (INT0,1,2,3,4,5,8,F)
Port	Port A	8 (PA0,1,2,3,4,5,6,7)	8 (PA0,1,2,3,4,5,6,7)
	Port B	8 (PB0,1,2,3,4,5,6,7)	8 (PB0,1,2,3,4,5,6,7)
	Port C	8 (PC0,1,2,3,4,5,6,7)	8 (PC0,1,2,3,4,5,6,7)
	Port D	8 (PD0,1,2,3,4,5,6,7)	-|
	Port E	8 (PE0,1,2,3,4,5,6,7)	6 (PE0,1,2,3,4,5)
	Port F	5 (PF0,1,2,3,4)	2 (PF0,1)
	Port G	8 (PG0,1,2,3,4,5,6,7)	-
	Port H	8 (PH0,1,2,3,4,5,6,7)	8 (PH0,1,2,3,4,5,6,7)
	Port I	8 (PI0,1,2,3,4,5,6,7)	8 (PI0,1,2,3,4,5,6,7)
	Port J	8 (PJ0,1,2,3,4,5,6,7)	-
	Port L	8 (PL0,1,2,3,4,5,6,7)	8 (PL0,1,2,3,4,5,6,7)
	Port M	8 (PM0,1,2,3,4,5,6,7)	8 (PM0,1,2,3,4,5,6,7)
	Port N	8 (PN0,1,2,3,4,5,6,7)	-
	Port P	8 (PP0,1,2,3,4,5,6,7)	8 (PP0,1,2,3,4,5,6,7)
Peripheral circuit	TMRB	8ch	8ch (ch3 and ch5 can only use the interval timer function.)
	UART	1ch	1ch
	SIO/UART	3ch (ch0,1,2)	2ch (ch0,1)
	I2C/SIO	1ch	1ch
	SSP	1ch	1ch
	RMC	1ch	1ch
	ADC	1unit/18ch (AIN0 to 17)	1unit/10ch (AIN0 to 9)
	RTC	1ch	1ch
	POR		
	VLTD		
	OFD		
	WDT	1ch	1ch
Debug I/F		JTAG/SWD/SWV/DATA TRACE(Data 2 bits)	JTAG/SWD/SWV/DATA TRACE (Data 2 bits)

2.2 Information of Each Peripheral Function

2.2.1 Exception

2.2.1.1 Differences of the Interrupt Factors

Table 2-2 Differences of the interrupt factors between the M381 and M383

No.	Interrupt factor	
	M381	M383
38	INT6 External interrupt pin 6	-
39	INT7 External interrupt pin 7	-
40	INTRX2 Serial channel receive interrupt (ch2)	-
41	INTTX2 Serial channel receive interrupt (ch2)	-
59	INT9 External interrupt pin 9	-
60	INTA External interrupt pin A	-
61	INTB External interrupt pin B	-
74	INTC External interrupt pin C	-
75	INTD External interrupt pin D	-
76	INTE External interrupt pin E	-

2.2.2 16-bit Timer / Event Counters (TMRB)

The TMPM381/383 incorporates 8-channel TMRB (TMRB0 to 7).

External clock input pin/capture trigger input pins (TB3IN and TB5IN), timer flip-flop output pins (TB3OUT and TB5OUT) are not available in Channel 3 (TMRB3) and Channel 5 (TMRB5) of the M383. Therefore, the timer functions (counter source clock selection, capture operation, and timer flip-flop output using TB3IN/TB5IN) cannot be used.

However, if the internal clock is selected in TMRB3 and TMRB5 of the M383, 16-bit internal timer function can be used. By selecting TB2OUT as the capture timing, capture operation is also possible.

TB5OUT cannot be used as the external output pin in TMRB5 of the M383. TB5OUT can be used as the timer flip-flop output. This output is utilized as the capture trigger input to TMRB6 and TMRB7 in the trigger function between timers.

Table 2-3 Differences of channels of the TMRB (M381)

	External pin		Trigger function between the timers		Interrupt		Internal connection	
Channel	External clock /capture trigger input pin	Timer flip-flop output pin	Capture trigger	Synchronous start trigger channel	Capture interrupt	TMRB interrupt	Start of ADC conversion	Transfer clock for SIO/UART, RMC
TMRB0	TB0IN	TB0OUT	TB7OUT	-	INTCAP00 INTCAP01	INTTB00 INTTB01	-	-
TMRB1	TB1N	TB1OUT	TB7OUT	TB0PRUN TB0RUN	INTCAP10 INTCAP11	INTTB10 INTTB11	-	RMC
TMRB2	TB2IN	TB2OUT	TB7OUT	TB0PRUN TB0RUN	INTCAP20 INTCAP21	INTTB20 INTTB21	-	-
TMRB3	TB3IN	TB3OUT	TB2OUT	TB0PRUN TB0RUN	INTCAP30 INTCAP31	INTTB30 INTTB31	-	-
TMRB4	TB4IN	TB4OUT	TB2OUT	-	INTCAP40 INTCAP41	INTTB40 INTTB41	-	SIO0 SIO1
TMRB5	TB5IN	TB5OUT	TB2OUT	TB4PRUN TB4RUN	INTCAP50 INTCAP51	INTTB50 INTTB51	INTTB51	-
TMRB6	TB6IN	TB6OUT	TB5OUT	TB4PRUN TB4RUN	INTCAP60 INTCAP61	INTTB60 INTTB61	-	-
TMRB7	TB7IN	TB7OUT	TB5OUT	TB4PRUN TB4RUN	INTCAP70 INTCAP71	INTTB70 INTTB71	-	SIO2

Table 2-4 Differences of channels of the TMRB (M383)

	External pin		Trigger function between the timers		Interrupt		Internal connection	
Channel	External clock /capture trigger input pin	Timer flip-flop output pin	Capture trigger	Synchronous start trigger channel	Capture interrupt	TMRB interrupt	ADC conversion starts	Transfer clock for SIO/ UART, RMC
TMRB0	TB0IN	TB0OUT	TB7OUT	-	INTCAP00 INTCAP01	INTTB00 INTTB01	-	-
TMRB1	TB1N	TB1OUT	TB7OUT	TB0PRUN TB0RUN	INTCAP10 INTCAP11	INTTB10 INTTB11	-	RMC
TMRB2	TB2IN	TB2OUT	TB7OUT	TB0PRUN TB0RUN	INTCAP20 INTCAP21	INTTB20 INTTB21	-	-
TMRB3	-	-	TB2OUT	TB0PRUN TB0RUN	INTCAP30 INTCAP31	INTTB30 INTTB31	-	-
TMRB4	TB4IN	TB4OUT	TB2OUT	-	INTCAP40 INTCAP41	INTTB40 INTTB41	-	SIO0 SIO1
TMRB5	-	-	TB2OUT	TB4PRUN TB4RUN	INTCAP50 INTCAP51	INTTB50 INTTB51	INTTB51	-
TMRB6	TB6IN	TB6OUT	TB5OUT	TB4PRUN TB4RUN	INTCAP60 INTCAP61	INTTB60 INTTB61	-	-
TMRB7	TB7IN	TB7OUT	TB5OUT	TB4PRUN TB4RUN	INTCAP70 INTCAP71	INTTB70 INTTB71	-	-

2.2.3 Serial Channel (SIO/UART)

The M381 incorporates three-channel SIO/UART (SIO0,1,2); the M383 incorporates two-channel (SIO0,1) SIO/UART.

Table 2-5 SIO/UART Pin specifications(M381)

Channel	Pin specification				Interrupt		Internal connection
	TXDx	RXDx	SCLKx	$\overline{\text{CTSx}}$	Reception	Transmission	Transfer clock input
SIO0	PE0	PE1	PE2	PE2	INTRX0	INTTX0	TB4OUT (TMRB4)
SIO1	PA5	PA6	PA4	PA4	INTRX1	INTTX1	TB4OUT (TMRB4)
SIO2	PD6	PD5	PD4	PD4	INTRX2	INTTX2	TB7OUT (TMRB7)

Table 2-6 SIO/UART Pin specifications (M383)

Channel	Pin specification				Interrupt		Internal connection
	TXDx	RXDx	SCLKx	$\overline{\text{CTSx}}$	Reception	Transmission	Transfer clock input
SIO0	PE0	PE1	PE2	PE2	INTRX0	INTTX0	TB4OUT (TMRB4)
SIO1	PA5	PA6	PA4	PA4	INTRX1	INTTX1	TB4OUT (TMRB4)
SIO2	-	-	-	-	-	-	-

2.2.4 Universal Asynchronous Serial Communication Circuit (UART)

TMPM381/383 incorporates 2 channels of the UART.

Table 2-7 UART Pin specifications

Channel	Pin specification				
	Normal mode		50% duty mode		
	PCFR4[7:6]=11 PCFR5[7:5]=000		PCFR4[7:6]=00 PCFR5[7:5]=111		
	UT0TXD	UT0RXD	UT0TXD50B	UT0TXD50A	UT0RXD50
UART0	PC6	PC7	PC5	PC6	PC7

2.2.5 I2C Bus (I2C/SIO)

TMPM381/383 incorporates 1 channels of I2C/SIO.

Table 2-8 I2C Pin specifications

Channel	端子仕様		
	SDA0 SO0	SCL0 SI0	SCK
SBI0	PC0	PC1	PC2

2.2.6 Synchronous Serial Interface (SSP)

TMPM381/383 incorporates 1 channels of SSP.

Table 2-9 SSP pin specifications

Channel	Pin specification			
	SP0DO	SP0DI	SP0CLK	SP0FSS
SSP0	PC0	PC1	PC2	PC3

2.2.7 Analog/Digital Converter (ADC)

The M381 incorporates one unit 18-channel AD converter; the M383 incorporates one unit 10-channel AD converter.

Table 2-10 ADC pin specifications

Unit	Product	AIN0~7	AIN8~9	AIN10~17
ADC	M381	PH0~7	PI0~1	PJ0~7
	M383	PH0~7	PI0~1	-

2.2.8 Debug Interface

TMPM381/383 supports serial wire debug ports, JTAG debug ports and trace outputs.

TMPM381/383 does not support the usage of "JTAG+SW (without TRST)" which is in the "Debug Interface" chapter.

Table 2-11 Debug pin specifications

	TMS SWDIO	TCK SWCLK	TDO SWV	TDI	$\overline{\text{TRST}}$
JTAG Serial wire	PB3	PB4	PB5	PB6	PB7

	TRACECLK	TRACEDATA0	TRACEDATA1		
Trace output	PB0	PB1	PB2		

3. 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.

3.1 Information on the processor core

The following table shows the revision of the processor core in the TMPM381/383.

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
TMPM381/383	r2p1

3.2 Configurable Options

The Cortex-M3 core has optional blocks. The optional blocks of the revision r2p1 are ETM™ and MPU. The following table shows the configurable options in the TMPM381/383.

Feature	Configure option
FPB	Two literal comparators Six instruction comparators
DWT	Four comparators
ITM	Present
MPU	Absent
ETM	Present
AHB-AP	Present
AHB Trace Macrocell Interface	Absent
TPIU	Present
WIC	Absent
Debug Port	JTAG / Serial wire
Bit Band	Present
constant AHB control	Absent

3.3 Exceptions/ Interruptions

Exceptions and interruptions are described in the following section.

3.3.1 Number of Interrupt Inputs

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

TMPM381/383 has 63/45 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, 0x02 is read out.

3.3.2 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.

TMPM381/383 has 3 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.

3.3.3 SysTick

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

For the detail of SysTick exception, refer to the section of "SysTick" in the exception and the register of SysTick in the NVIC register.

3.3.4 SYSRESETREQ

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

TMPM381/383 provides the same operation when SYSRESETREQ signal are output.

Note: The reset operation by <SYSRESETREQ> cannot be used in SLOW mode.

3.3.5 LOCKUP

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

TMPM381/383 does not use this signal. To return from LOCKUP status, it is necessary to use non-maskable interrupt (NMI) or reset.

3.3.6 Auxiliary Fault Status register

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

However, TMPM381/383 is not defined this function. If auxiliary fault status register is read, always "0x0000_0000" is read out.

3.4 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.

TMPM381/383 does not use event output signals and event input signals. Please do not use SEV instruction and WFE instruction.

3.5 Power Management

The Cortex-M3 core provides power management system which uses SLEEPING signal and SLEEPDEEP signal. 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.

TMPM381/383 does not use SLEEPDEEP signal so that <SLEEPDEEP> bit must not be set. And also event signal is not used so that please do not use WFE instruction.

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

3.6 Exclusive access

In Cortex-M3 core, the DCode bus system supports exclusive access. However TMPM381/383 does not use this function.

4. Memory Map

4.1 Memory Map

The memory maps for TMPM381/383 are based on the ARM Cortex-M3 processor core memory map. The internal ROM, internal RAM and special function registers (SFR) of TMPM381/383 are mapped to the Code, SRAM and peripheral regions of the Cortex-M3 respectively. The special function register (SFR) means the control registers of all input/output ports and peripheral functions.

The CPU register area is the processor core’s internal register region.

For more information on each region, see the "ARM documentation set for the ARM Cortex-M3".

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

A memory map of TMPM381/383 is shown below:

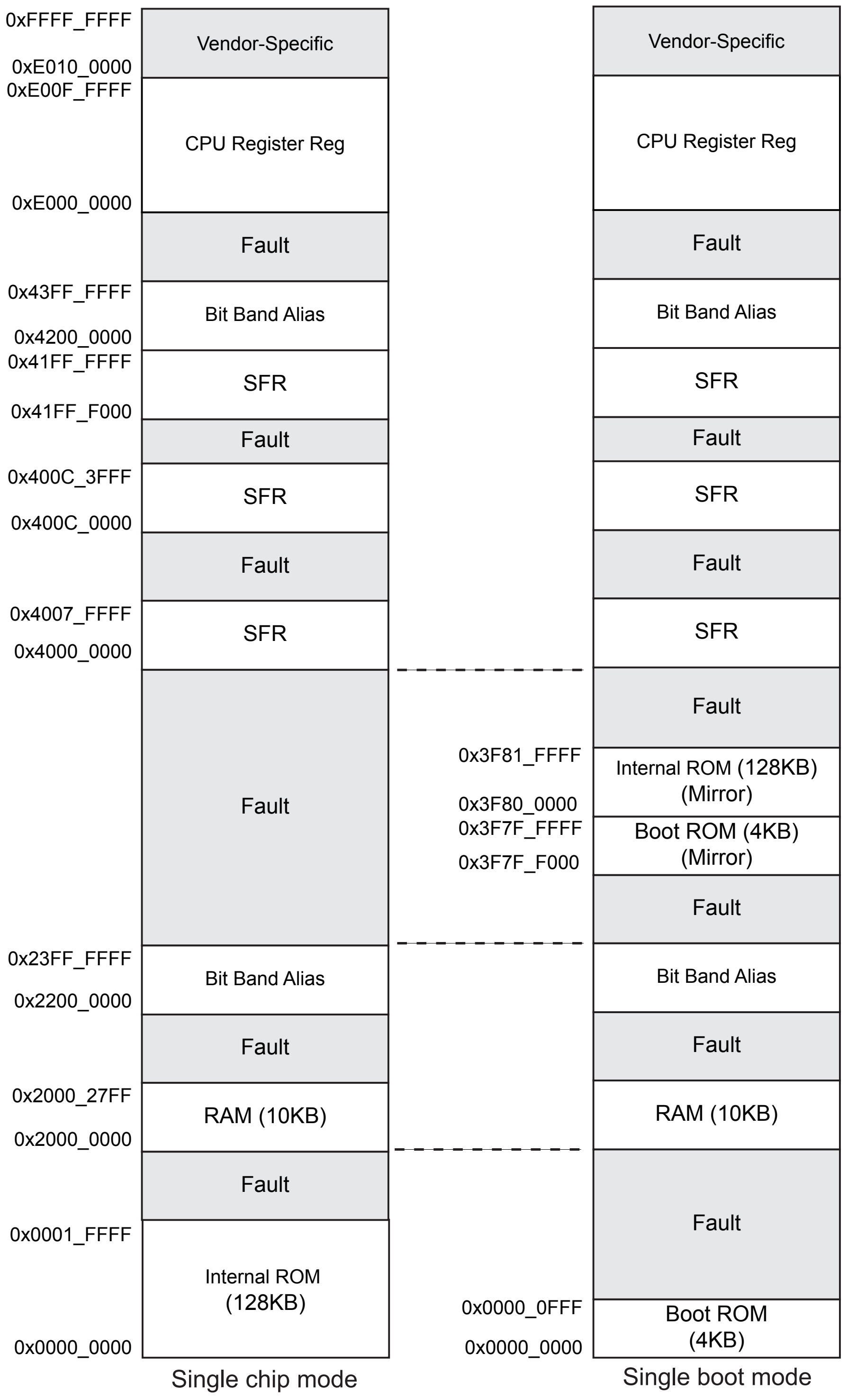


Figure 4-1 Memory Map (FLASH 128KB)

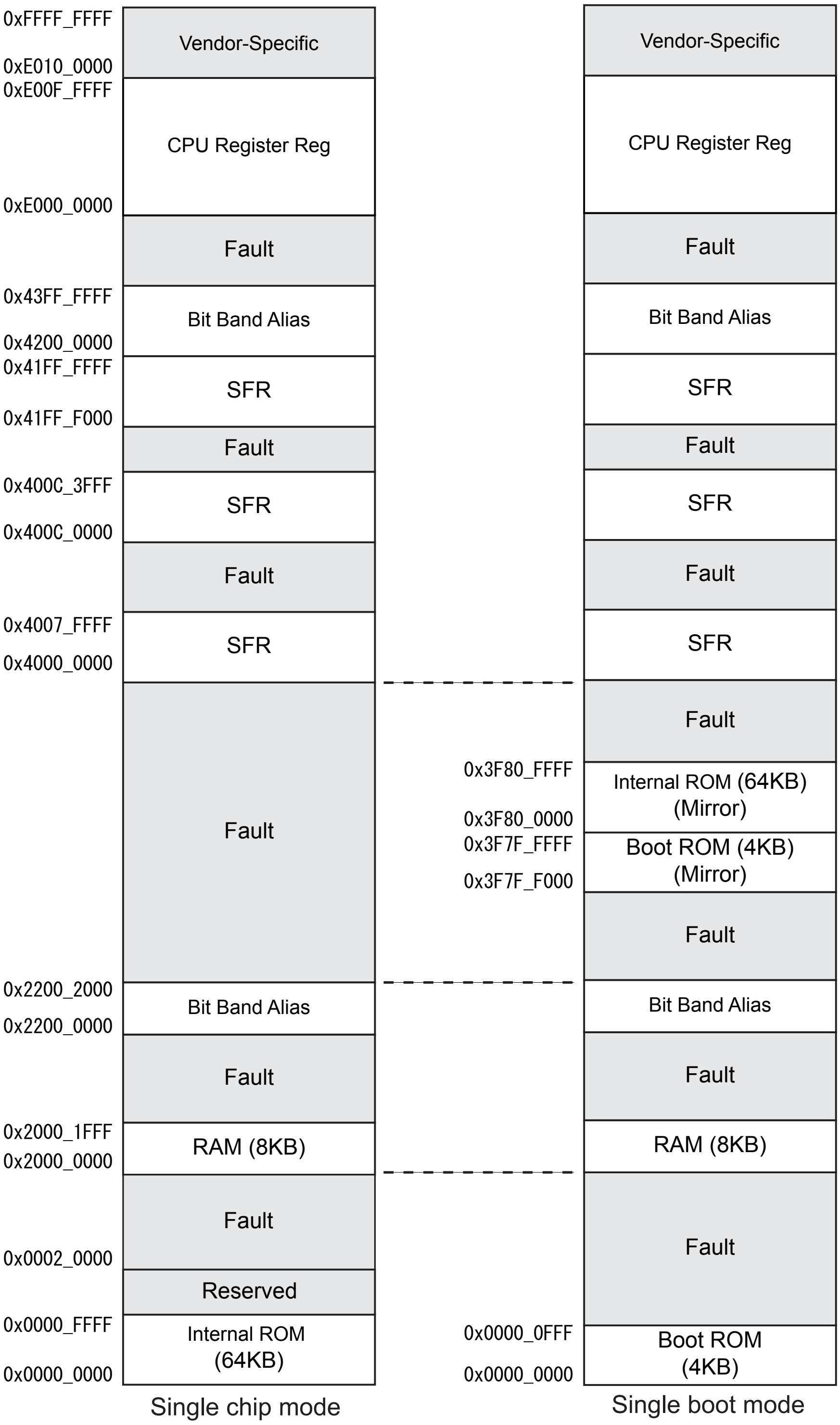


Figure 4-2 MemoryMap (FLASH 64KB)

4.2 Bus Matrix

This MCU contains one bus master .

Bus masters connect to slave ports (S0 to S2) of Bus Matrix. In the bus matrix, master ports (M0 to M4) connect to peripheral functions via connections described as (o) or (・) in the following figure. (・) shows a connection to a mirror area.

While multiple slaves are connected on the same bus master line in the Bus Matrix, if multiple slave accesses are generated at the same time, a priority is given to access from a master with the smallest slave number.

4.2.1 Structure

4.2.1.1 Single chip mode

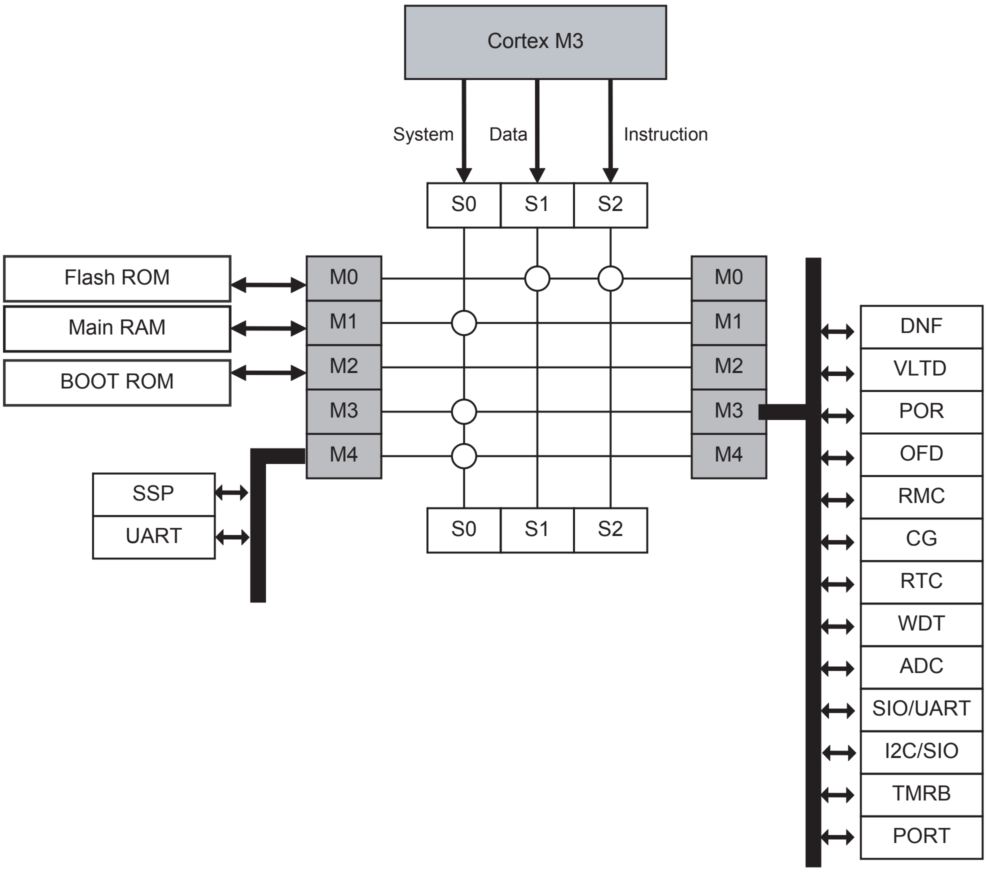


Figure 4-3 Bus Matrix of TMPM381/383

4.2.1.2 Single boot mode

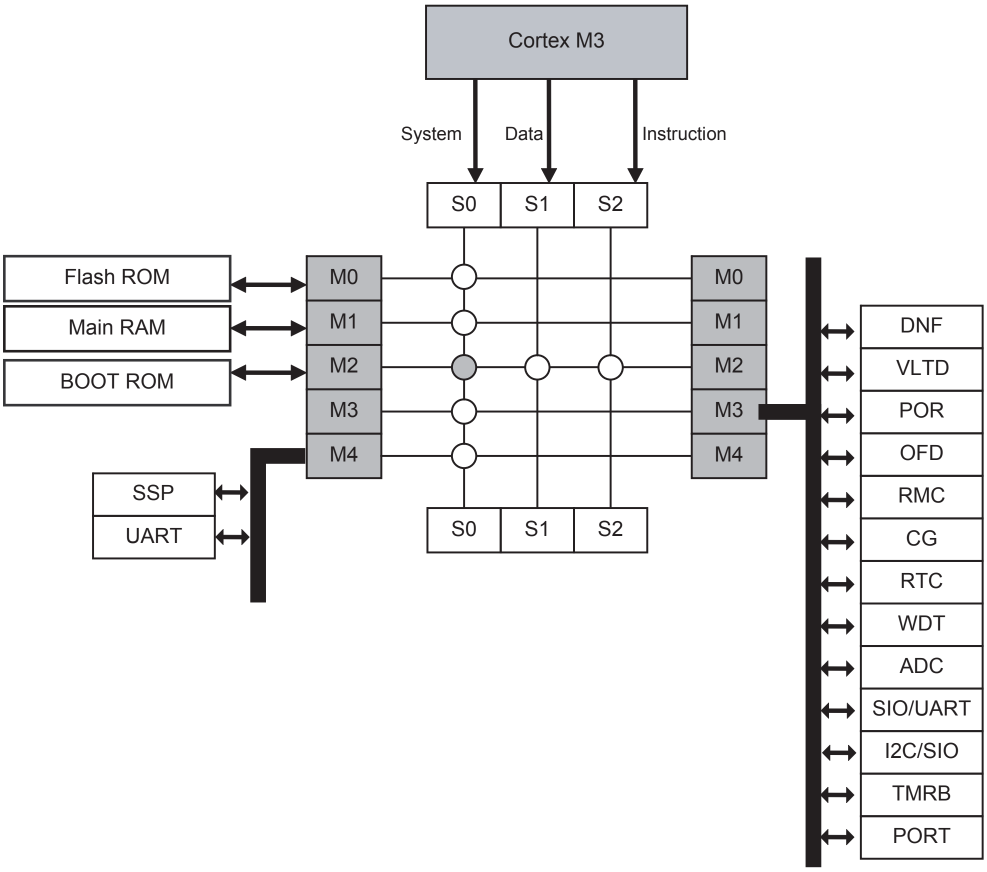


Figure 4-4 Bus Matrix of TMPM381/383

4.2.2 Connection table

4.2.2.1 Code area / SRAM area

(1) Single chip mode

Start Address	Slave	Master	Core S-Bus	Core D-Bus	Core I-Bus
			S0	S1	S2
0x0000_0000	Flash ROM	M0	Fault	o	o
0x0002_0000	Fault	-	Fault	Fault	Fault
0x2000_0000	Main RAM	M1	o	Fault	Fault
0x2000_2800	Fault	-	Fault	Fault	Fault
0x2200_0000	Bit band alias	-	o	Fault	Fault
0x2400_0000	Fault	-	Fault	Fault	Fault

(2) Single boot mode

Start Address		Master	Core S-Bus	Core D-Bus	Core I-Bus
	Slave		S0	S1	S2
0x0000_0000	Boot ROM	M2	Fault	o	o
0x0000_1000	Fault	-	Fault	Fault	Fault
0x2000_0000	Main RAM	M1	o	Fault	Fault
0x2000_2800	Fault	-	Fault	Fault	Fault
0x2200_0000	Bit band alias	-	o	Fault	Fault
0x2400_0000	Fault	-	Fault	Fault	Fault
0x3F7F_F000	Boot ROM(mirror)	M2	o	Fault	Fault
0x3F80_0000	Flash ROM(mirror)	M0	o	Fault	Fault
0x3F82_0000	Fault	-	Fault	Fault	Fault

Note: Please do not access the address range given in Reserved.

4.2.2.2 Peripheral area / External bus area

Start Address	Slave	Master	Core S-Bus	Core D-Bus	Core I-Bus
			S0	S1	S2
0x4000_0000	PORT	M3	o	Fault	Fault
0x4001_0000	TMRB		o	Fault	Fault
0x4002_0000	I2C/SIO		o	Fault	Fault
0x4002_0080	SIO/UART		o	Fault	Fault
0x4003_0000	ADC		o	Fault	Fault
0x4004_0000	WDT		o	Fault	Fault
0x4004_0100	RTC		o	Fault	Fault
0x4004_0200	CG		o	Fault	Fault
0x4004_0300	Fault	-	Fault	Fault	Fault
0x4004_0400	RMC	M3	o	Fault	Fault
0x4004_0500	Fault	-	Fault	Fault	Fault
0x4004_0700	Reserved	-	-	-	-
0x4004_0800	OFD	M3	o	Fault	Fault
0x4004_0900	VLTD		o	Fault	Fault
0x4004_0A00	Fault	-	Fault	Fault	Fault
0x4006_0000	DNF	M3	o	Fault	Fault
0x4006_0010	Fault	-	Fault	Fault	Fault
0x400C_0000	SSP	M4	o	Fault	Fault
0x400C_2000	UART		o	Fault	Fault
0x400C_3000	Fault	-	Fault	Fault	Fault
0x41FF_F000	Flash(SFR)	-	o	Fault	Fault
0x4200_0000	Bit band alias	-	o	Fault	Fault
0x4400_0000	Fault	-	Fault	Fault	Fault

4.3 Address lists of peripheral functions

Do not access to addresses in the peripheral area except control registers. For details of control registers, refer to Chapter of each peripheral functions.

Peripheral Function		Base Address
Input / Output port	Port A	0x4000_0000
	Port B	0x4000_0040
	Port C	0x4000_0080
	Port D	0x4000_00C0
	Port E	0x4000_0100
	Port F	0x4000_0140
	Port G	0x4000_0180
	Port H	0x4000_01C0
	Port I	0x4000_0200
	Port J	0x4000_0240
	Port L	0x4000_02C0
	Port M	0x4000_0300
	Port N	0x4000_0340
	Port P	0x4000_0380
16-bit Timer / Event Counters (TMRB)	ch0	0x4001_0000
	ch1	0x4001_0040
	ch2	0x4001_0080
	ch3	0x4001_00C0
	ch4	0x4001_0100
	ch5	0x4001_0140
	ch6	0x4001_0180
	ch7	0x4001_01C0
Serial Bus interface (I2C/SIO)	ch0	0x4002_0000
Serial Channel (SIO/UART)	ch0	0x4002_0080
	ch1	0x4002_00C0
	ch2	0x4002_0100
Analog / Digital Converter (ADC)		0x4003_0000
Watchdog Timer(WDT)		0x4004_0000
Real Time Clock (RTC)		0x4004_0100
Clock/Mode control(CG)		0x4004_0200
Remote control signal preprocessor (RMC)		0x4004_0400
Oscillation Frequency Detector (OFD)		0x4004_0800
Low Voltage Detection Circuit (VLTD)		0x4004_0900
Digital Noise Filter Circuit (DNF)		0x4006_0000
Synchronous Serial Port (SSP)		0x400C_0000
Asynchronous Serial Channel (UART)		0x400C_2000
Flash Control(Flash SFR)		0x41FF_F000

5. Reset Operation

The following are sources of reset operation.

- Power-on-reset circuit (POR)
- Voltage Detection Circuit (VLTD)
- RESET pin ($\overline{\text{RESET}}$)
- Watch-dog timer (WDT)
- Oscillation frequency circuit (OFD)
- Application interrupt by CPU and a signal from the reset register bit <SYSRESETREQ>

To recognize a source of reset, check CGRSTFLG in the clock generator register described in Chapter of " Exception ".

Detail about the power-on-reset circuit, the power detection circuit, the watch-dog timer and the oscillation frequency detection circuit, refer to each chapter.

A reset by <SYSRESETREQ> is referred to " Cortex-M3 Technical Reference Manual " .

Note: Once reset operation is done, internal RAM data is not assured.

5.1 Cold Reset

When turning-on power, it is necessary to take a stable time of built-in regulator, built-in Flash memory and internal high-speed oscillator into consideration. TMPM381/383 has a function to insert a stable time automatically.

5.1.1 Reset by VLTD circuit (not using $\overline{\text{RESET}}$ pin)

Once power voltage is beyond the release voltage of power-on-reset, power counter starts operation, and then after tPWUP (approximately 0.9ms in fosc=9MHz) internal reset signal is released.

TMPM381/383 has a function to enable low voltage detection circuit (VLTD) operation. Within tPWUP, if the supply voltage becomes upper than the detection voltage <VDLVL[1:0]>, internal reset signal is released by power counter stops operation.

Power-on-reset circuit operation is referred to Section of " Power-on-reset circuit (POR) ". And low voltage detection circuit operation is referred to Section of " Low voltage detection circuit (VLTD) " .

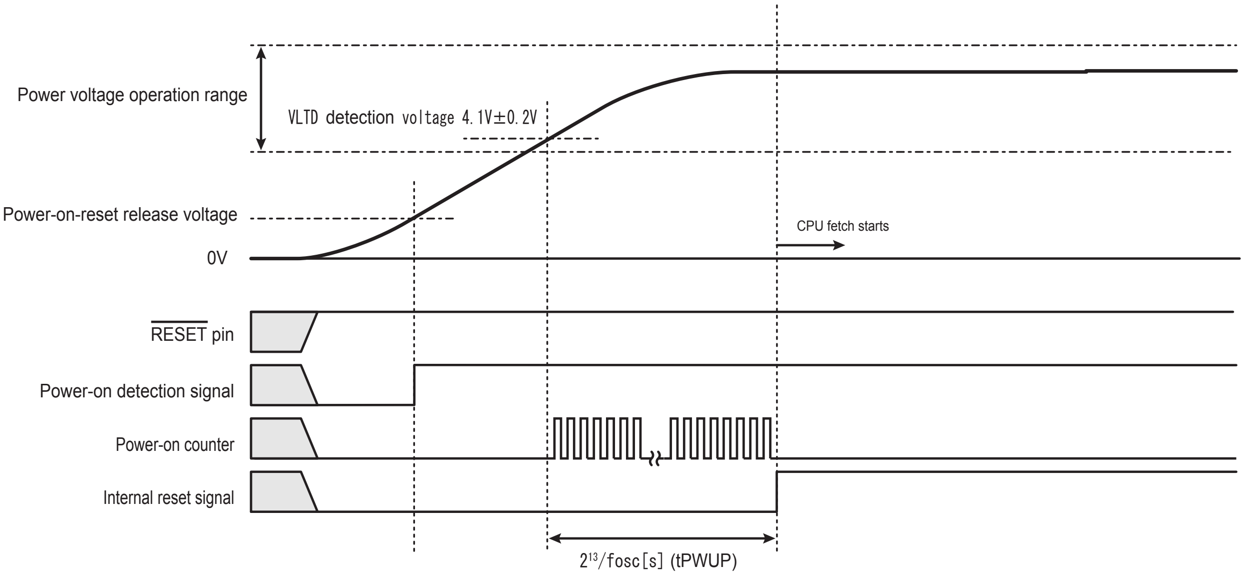

Power voltage operation range
VLTD detection voltage 4.1V±0.2V
Power-on-reset release voltage
0V
CPU fetch starts
RESET pin
Power-on detection signal
Power-on counter
Internal reset signal
2^13/fosc[s] (tPWUP)

Figure 5-1 Reset Operation by VLTD Circuit

5.1.2 Reset by $\overline{\text{RESET}}$ pin

The reset using the $\overline{\text{RESET}}$ pin will be effective after the power-on counter finishes. And if $\overline{\text{RESET}}$ pin is set to " High " within tPWUP after power-on reset signal becomes " High " , the reset process will be the same as the power-on described in 5.1.1.

TMPM381/383 has a function to enable low voltage detection circuit (VLTD) operation. Before $\overline{\text{RESET}}$ pin is set to " High " , if the supply voltage becomes upper than the detection voltage <VDLVL[1:0]>, internal reset signal is released after $\overline{\text{RESET}}$ pin is set to " Low " .

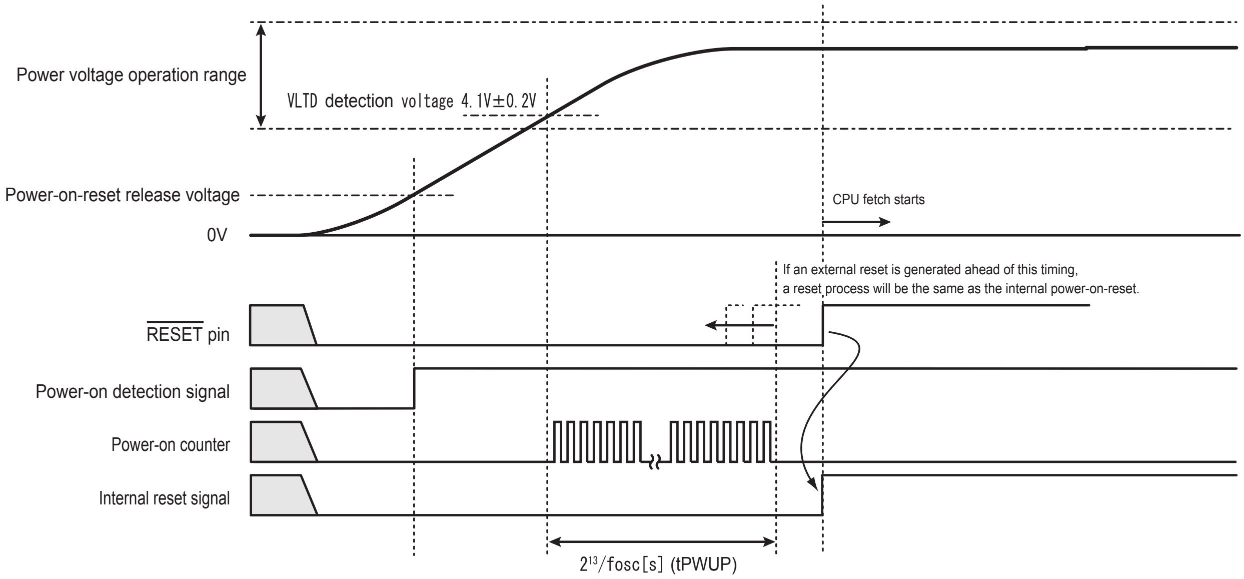

Power voltage operation range
VLTD detection voltage 4.1V±0.2V
Power-on-reset release voltage
0V
CPU fetch starts
If an external reset is generated ahead of this timing,
a reset process will be the same as the internal power-on-reset.
$\overline{\text{RESET}}$ pin
Power-on detection signal
Power-on counter
Internal reset signal
2^{13}/fosc[s] (tPWUP)

Figure 5-2 Reset Operation by $\overline{\text{RESET}}$ pin

5.2 Warm-up

5.2.1 Reset Duration

To do reset TMPM381/383, the following condition is required; power supply voltage is in the operational range; RESET pin is kept " Low " at least for 12 system clocks by internal high frequency oscillator. After RESET pin becomes " High " , internal reset will be released.

5.3 After reset

After reset, the control register of processor core and the peripheral function control register (SFR) are almost initialized. System debug component registers (FPB, DWT, and ITM) of the internal core, CGRSTFLG in the clock generator and FCSECBIT in the Flash related register are only initialized by cold reset.

When reset is released, MCU starts operation by a clock of internal high-speed oscillator. External clock and PLL multiple circuit should be set if necessary.

6. Clock/Mode control

6.1 Outline

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.

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	-	-	-	FCSTOP	-	-	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
31-24	−	R	Read as "0".
23	−	R/W	Write "0".
22-21	−	R	Read as "0".
20	FCSTOP	R/W	ADC clock 0: Active 1: Stop Enables to stop providing AD converter clock. AD converter clock is provided after reset. Confirming that AD converter is stopped or finished in advance is required when setting "1" (stop).
19-18	−	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.
15-14	−	R	Read as "0".
13-12	FPSEL[1:0]	R/W	φT0 source clock 00: The clock divided fgear by a prescaler 01: The clock divided fc by a prescaler 10: fsys 11: fsys Specifies the source clock to φT0. In the SLOW mode, must be set "10" or "11".
11	−	R	Read as "0".
10-8	PRCK[2:0]	R/W	Prescaler clock 000: fperiph 100: fperiph/16 001: fperiph/2 101: fperiph/32 010: fperiph/4 110: Reserved 011: fperiph/8 111: Reserved Specifies the prescaler clock to peripheral functions.
7-3	−	R	Read as "0".
2-0	GEAR[2:0]	R/W	High-speed clock (fc) gear 000: fc 100: fc/2 001: Reserved 101: fc/4 010: Reserved 110: fc/8 011: Reserved 111: fc/16

6.2.3 CGOSCCR (Oscillation control register)

	31	30	29	28	27	26	25	24
bit symbol	WUODR							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	WUODR				WUPSEL2	HOSCON	OSCSEL	XEN2
After reset	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
bit symbol	WUODRL		-	-	-	-	XTEN	XEN1
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	WUPSEL1	PLLON	WUEF	WUEON
After reset	0	0	1	1	0	0	0	0

Bit	Bit Symbol	Type	Function
31-20	WUODR[11:0]	R/W	Warm-up counter setup value. Warm-up timer value.
19	WUPSEL2	R/W	High-speed warm-up clock. 0: internal (f_{IHOSC}) 1: external (f_{EHOSC}) Select warm-up counter by high-speed oscillator. The warm-up counter is counted up by the selected clock. When using STOP/SLEEP mode, please select clock-source that is same as <OSCSEL> to <WUPSEL2> before entering to STOP/SLEEP mode.
18	HOSCON	R/W	Port M or X1/X2 0: Port M 1: X1/X2 (f_{EHOSC}) Specifies Port M or X1/X2. When external oscillator is used, set PMCR/PMPUP/PMPDN/PMIE of Port M to disable. After reset, PMCR/PMPUP/PMPDN/PMIE are set to disable.
17	OSCSEL	R/W	High-speed oscillator (Note 3) 0: internal high-speed oscillator 1: external high-speed oscillator
16	XEN2	R/W	internal high-speed oscillator operation 0: Stop 1: Oscillation
15-14	WUODRL[1:0]	R/W	Warm-up counter setup value. Setup the 16-bit timer for warm-up timer of lower 2-bits counter value. This is used for low-speed clock. If high-speed oscillator is selected, <WUODRL[1:0]> is set "00".
13-12	−	R/W	Write "0".
11-10	−	R	Read as "0".
9	XTEN	R/W	External low-speed oscillator operation 0: Stop 1: Oscillation
8	XEN1	R/W	External high-speed oscillator operation 0: Stop 1: Oscillation
7-4	−	R/W	Write "0011"
3	WUPSEL1	R/W	Selects warm-up counter 0: high-speed 1: low-speed
2	PLLON	R/W	PLL (multiplying circuit) operation (Note 4) 0: Stop 1: Oscillation
1	WUEF	R	Status of warm-up timer (WUP) 0: WUP finish 1: WUP active Enables to monitor the status of the warm-up timer.

Bit	Bit Symbol	Type	Function
0	WUEON	W	Operation of warm-up timer (WUP) 0: don’t care 1: WUP start Enables to start the warm-up timer. Read as "0".

Note 1: Refer to Section "6.3.4 Warm-up function" about the Warm-up setup.

Note 2: Refer to "6.3.5 Clock Multiplication Circuit (PLL)" about setting PLL.

Note 3: If CGOSCCR<OSCSEL> is "1", PMCR/PMPUP/PMPDN/PMIE can not be modified.

Note 4: When using f_{IHOSC} as system clock, do not use PLL multiplying.

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
31-20	−	R	Read as "0".
19-17	−	R/W	Write "0".
16	DRVE	R/W	Controls the port in STOP mode 0: Not drive ports 1: Drive ports
15-10	−	R	Read as "0".
9	RXTEN	R/W	Low-speed oscillator operation after releasing STOP mode. 0: Stop 1: Oscillation
8	RXEN	R/W	High-speed oscillator operation after releasing 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

Note: Access to the "Reserved" is prohibited.

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	1	1	0	1	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	PLLSEL
After reset	0	0	0	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-12	−	R/W	Write "1101"
11	−	R	Read as "0".
10-8	−	R/W	Write "000"
7-1	−	R/W	Write "0001111"
0	PLLSEL	R/W	Use of PLL 0: fosc use 1: f_{PLL} use Specifies use or disuse of the clock multiplied by the PLL. "fosc (internal high-speed oscillator)" is automatically set after reset.Resetting is required when using the PLL.

Note: When using f_{IHOSC} as system clock, do not use PLL multiplying.

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	Selects system clock 0: high-speed 1: low-speed Specifies system clock. When change value of <SYSCK>, oscillation must stable High-speed oscillator (f_{EHOSC} or f_{IHOSC}) and Low-speed oscillator. According to the used oscillator, corresponding CGOSCCR<XEN1>, <XEN2> or <XTEN> must be set to "1" in advance.
0	SYSCKFLG	R	System clock status 0: high-speed 1: low-speed Shows the status of the system clock. When switching the oscillator with <SYSCK>, generates time lag to complete. If the read value from <SYSCKFLG> is the same as the value specified in <SYSCK>, the switching has been completed.

6.3 Clock control

6.3.1 Clock Type

Each clock is defined as follows:

f_{EHOSC} : Clock generated by external high-speed oscillator.

f_{IHOSC} : Clock input from internal high-speed oscillator.

fs : Clock generated by external low-speed oscillator.

fosc : f_{IHOSC} or f_{EHOSC} specified by CGOSCCR<OSCSEL>.

f_{PLL} : Clock multiplied by 4 by PLL.

fc : Clock specified by CGPLLSEL<PLLSEL> (high-speed clock)

fgear : Clock specified by CGSYSCR<GEAR[2:0]>

fsys : Clock specified by CGSKSEL<SYSCK>

fperiph : Clock specified by CGSYSCR<FPSEL[1:0]>

φT0 : Clock specified by CGSYSCR<PRCK[2:0]> (prescaler clock)

The gear clock fgear and the prescaler clock φT0 are dividable as follows.

Gear clock : fc, fc/2, fc/4, fc/8, fc/16

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

6.3.2 Initial Values after Reset

Reset operation initializes the clock configuration as follows.

internal high-speed oscillator : oscillating

external high-speed oscillator : stop

external low-speed oscillator : stop

PLL (phase locked loop circuit) : stop

Gear clock : fc (no frequency dividing)

Reset operation causes all the clock configurations to be the same as f_{IHOSC}.

fc = f_{IHOSC}

fsys = fc = f_{IHOSC}

fperiph = fc = f_{IHOSC}

φT0 = fperiph = f_{IHOSC}

6.3.3 Clock system Diagram

Figure 6-1 shows the clock system diagram.

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

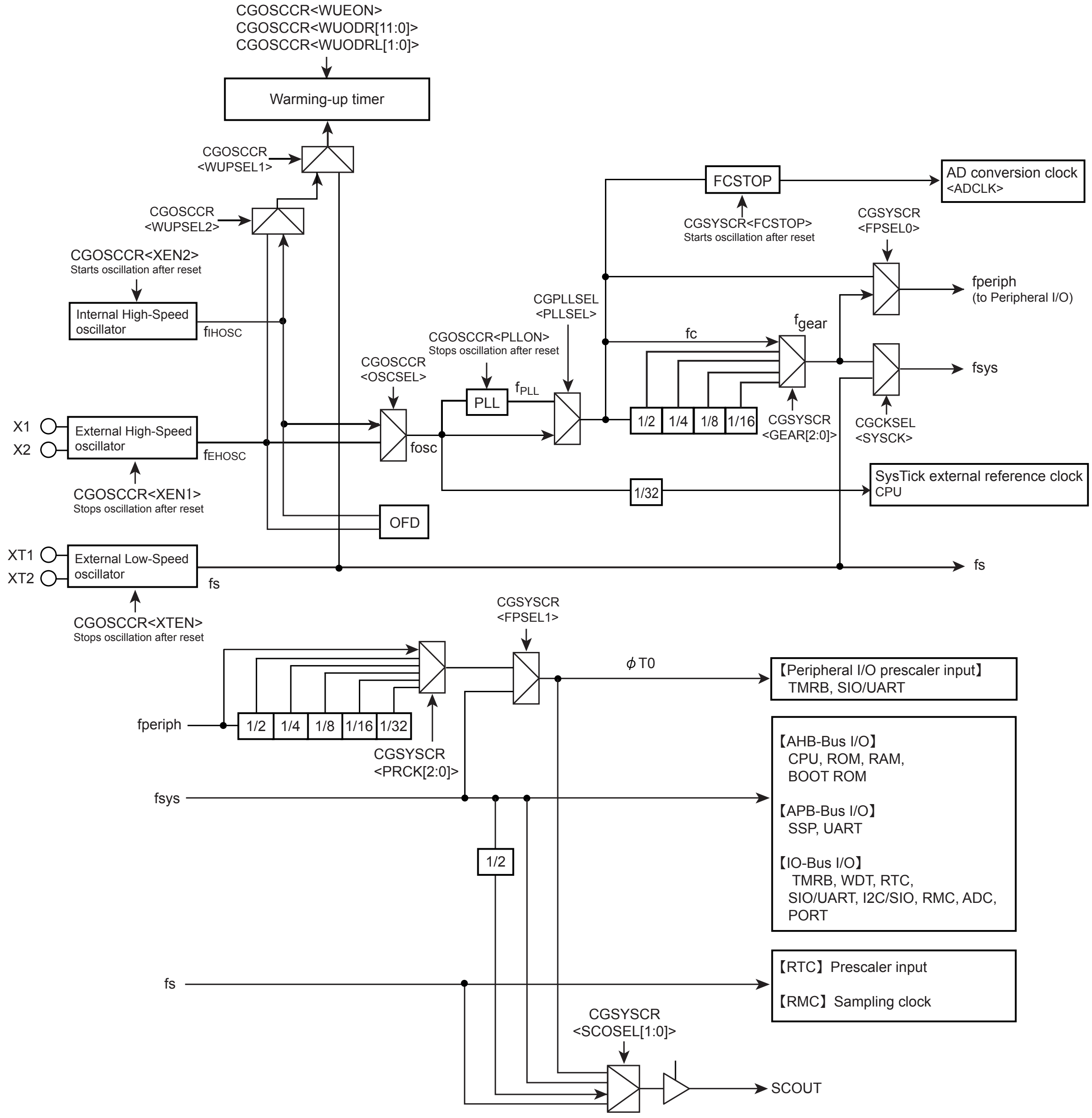


Figure 6-1 Clock Block Diagram

6.3.4 Warm-up function

The warm-up function secures the stability time for the oscillator of fs and the PLL with the warm-up timer when releasing STOP mode. Refer to "6.6.7 Warm-up" for a detail.

How to configure the warm-up function.

1. Specify the count up clock

   Specify the count up clock for the warm-up counter in the CGOSCCR <WUPSEL2> <WUPSEL1>.

2. Specify the warm-up counter value

   The warm-up time can be selected by setting the CGSCCR<WUODR[11:0]><WUODRL[1:0]>.

   The value can be calculated by following formula with round lower 4 bit off, set to the bit of <WUODR[11:0]> for high-speed oscillation and set to the bit of <WUODR[11:0]><WUODRL[1:0]> for low-speed oscillation.

$$\text{number of warm-up cycle} = \frac{\text{warm-up time to set}}{\text{input frequency cycle (s)}}$$

Note: Set CGOSCCR<WUODRL[1:0]> to "00" for high-speed oscillation.

<example 1> When using high-speed oscillator 8MHz, and set warm-up time 5ms.

$$\frac{\text{warm-up time to set}}{\text{input frequency cycle (s)}} = \frac{\text{5ms}}{\text{1/8MHz}} = \text{4000 cycle} = \text{0x9C40}$$

Round lower 4 bit off, set 0x9C4 to CGOSCCR<WUODR[11:0]>

3. confirm the start and completion of warm-up

   The CGOSCCR<WUEON><WUEF> is used to confirm the start and completion of warm-up through software (instruction). When CGOSCCR<WUEON> is set to "1", the warm-up start a count up. The completion of warm-up can be confirmed with CGOSCCR<WUEF>.

Note 1: Setting warm-up count value to CGOSCCR<WUDOR[11:0]><WUDORL[1:0]>, wait until this value is reflected, then transit to standby mode by executing a command "WFI".

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

The example of warm-up function setup.

Table 6-1 The example of warm-up setting (When an external high-speed oscillator is selected)

	CGOSCCR<WUPSEL1> = "0"	: Selects the warm-up clock (Specifies high-speed oscillator).
	CGOSCCR<WUPSEL2> = "1"	: Selects the warm-up clock (Specifies external oscillator (f_{EHOSC})).
	CGOSCCR<WUODR[11:0]> = "0x9C4" CGOSCCR<WUODRL[1:0]> = "00"	: Specifies the warm-up time.
↻	CGOSCCR<WUODR[11:0]> read	: Confirm warm-up time reflecting Repeat until the read data is "0x9C4".
	CGOSCCR<XEN1> = "1"	: high-speed oscillator (f_{EHOSC}) enable
	CGOSCCR<WUEON> = "1"	: Start the warm-up timer (WUP)
↻	CGOSCCR<WUEF> read	: Wait until the state becomes "0" (warm-up is finished)

6.3.5 Clock Multiplication Circuit (PLL)

This circuit outputs the f_{PLL} clock that is multiplied by 4 of the high-speed oscillator output clock (fosc). As a result, the input frequency to oscillator can be low frequency, and the internal clock be made high-speed.

6.3.5.1 How to configure the PLL function

The PLL is disabled after reset.

To enable the PLL, set CGOSCCR<PLLON> to "1". After 200μs for lock-up time elapses, set CGPLLSEL<PLLSEL> to "1", f_{PLL} which is multiplied by 4 from fosc is used.

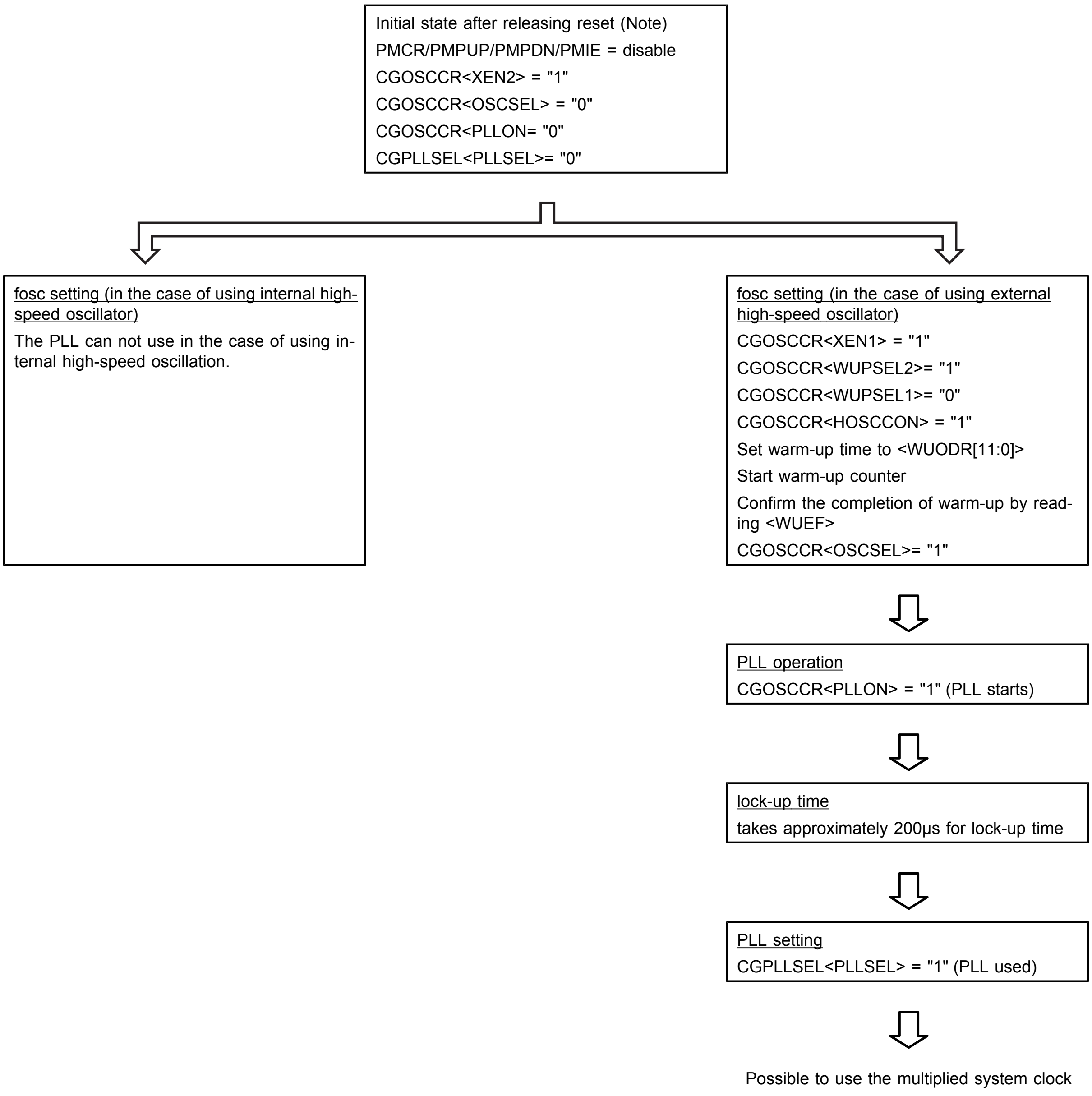
The PLL requires a certain amount of time to be stabilized, which should be secured using the warm-up function or other methods.

Note: When using f_{IHOSC} as system clock, do not use PLL multiplying.

6.3.5.2 The sequence of PLL setting

The sequence of PLL setting is shown below.

The sequence of PLL setting



Initial state after releasing reset (Note)
PMCR/PMPUP/PMPDN/PMIE = disable
CGOSCCR<XEN2> = "1"
CGOSCCR<OSCSEL> = "0"
CGOSCCR<PLLON= "0"
CGPLLSEL<PLLSEL>= "0"

fosc setting (in the case of using internal high-speed oscillator)
The PLL can not use in the case of using internal high-speed oscillation.

fosc setting (in the case of using external high-speed oscillator)
CGOSCCR<XEN1> = "1"
CGOSCCR<WUPSEL2>= "1"
CGOSCCR<WUPSEL1>= "0"
CGOSCCR<HOSCCON> = "1"
Set warm-up time to <WUODR[11:0]>
Start warm-up counter
Confirm the completion of warm-up by reading <WUEF>
CGOSCCR<OSCSEL>= "1"

PLL operation
CGOSCCR<PLLON> = "1" (PLL starts)

lock-up time
takes approximately 200μs for lock-up time

PLL setting
CGPLLSEL<PLLSEL> = "1" (PLL used)

Possible to use the multiplied system clock

Note: Internal high-speed oscillator and voltage supply need to be stable.

6.3.6 System clock

The internal high-speed oscillation clock, the external high-speed oscillation clock or the external low-speed oscillation clock can be used as a source clock of the system clock.

Source clock	Frequency	using PLL
Internal high-speed oscillation (IHOSC)	9MHz (Target)	Can not use PLL
External high-speed oscillation (EHOSC)	8 to 10MHz	Not use or 4 multiplying
External low-speed oscillation (fs)	30 to 34 kHz	-

When the internal high-speed oscillation or the external high-speed oscillation is used as a source clock, the clock divided by CGSYSCR<GEAR[2:0]> is used as the system clock. Although the setting can be changed while operating, the actual switching takes place after a slight delay.

Table 6-2 shows the example of the operation frequency by the setting PLL and the clock gear.

Table 6-2 The range of an operation frequency when a clock gear and PLL are used
(Unit : MHz, "-" : Reserved, "*" : Don’t care)

Input frequency		PLL multiplying	Min. operation frequency (fc)	Max. operation frequency (fc)	ADC Max. operation frequency	Clock gear (CG) PLL = ON					Clock gear (CG) PLL = OFF				
						1/1	1/2	1/4	1/8	1/16	1/1	1/2	1/4	1/8	1/16
EHOSC	8	4	1	32	32	32	16	8	4	2	8	4	2	1	-
	10			40	40	40	20	10	5	2.5	10	5	2.5	1.25	-
IHOSC	10			10	10	-	-	-	-	-	10	5	2.5	1.25	-
fs	0.032768	-	0.032768	0.032768	-	*	*	*	*	*	*	*	*	*	*

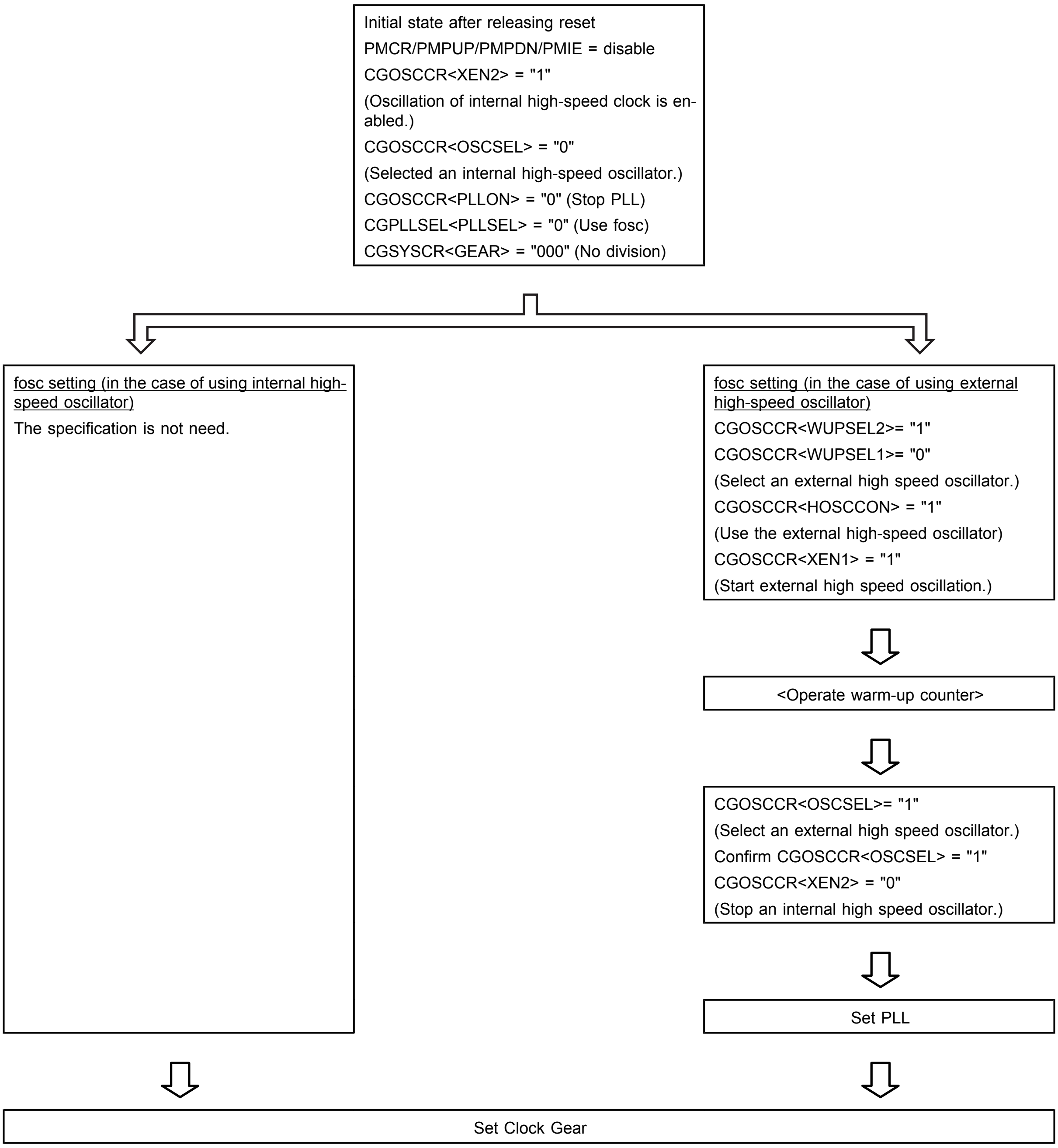
↑ Initial value after reset

Note: Do not use 1/16 when SysTick is used.

6.3.6.1 The sequence of System clock setting

The system clock is selected by CGOSCCR. After setting CGOSCCR, the PLL is set by CGPLLSEL and CGOSCCR and the clock gear is set by CGSYSCR.

The sequence of PLL setting



Initial state after releasing reset
PMCR/PMPUP/PMPDN/PMIE = disable
CGOSCCR<XEN2> = "1"
(Oscillation of internal high-speed clock is enabled.)
CGOSCCR<OSCSEL> = "0"
(Selected an internal high-speed oscillator.)
CGOSCCR<PLLON> = "0" (Stop PLL)
CGPLLSEL<PLLSEL> = "0" (Use fosc)
CGSYSCR<GEAR> = "000" (No division)

fosc setting (in the case of using internal high-speed oscillator)
The specification is not need.

fosc setting (in the case of using external high-speed oscillator)
CGOSCCR<WUPSEL2>= "1"
CGOSCCR<WUPSEL1>= "0"
(Select an external high speed oscillator.)
CGOSCCR<HOSCCON> = "1"
(Use the external high-speed oscillator)
CGOSCCR<XEN1> = "1"
(Start external high speed oscillation.)

<Operate warm-up counter>

CGOSCCR<OSCSEL>= "1"
(Select an external high speed oscillator.)
Confirm CGOSCCR<OSCSEL> = "1"
CGOSCCR<XEN2> = "0"
(Stop an internal high speed oscillator.)

Set PLL

Set Clock Gear

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[1:0]> can be divided according to the setting in the CGSYSCR<PRCK[2:0]>. After the TMPM381/383 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

This product enables to output the system clock from a pin. The SCOUT pin can output the low speed clock fs, the system clock fsys and fsys/2, and the prescaler input clock for peripheral functions φT0.

Note 1: The phase difference (AC timing) between the system clock output by the SCOUT and the internal clock is not guaranteed.

Note 2: When fsys is output from SCOUT pin, SCOUT pin outputs the unexpected waveform just after changing clock gear. In the case of influencing to system by the unexpected waveform, the output of SCOUT pin should be disabled when changing the clock gear.

When the port is used as SCOUT pin, refer to "Input/Output port".

Table 6-3 shows the pin status in each mode when the SCOUT pin is set to the SCOUT output.

Table 6-3 SCOUT Output Status in Each Mode

Mode SCOUT selection CGSYSCR	NORMAL	SLOW	Low power consumption mode		
			IDLE	SLEEP	STOP (Note)
<SCOSEL[1:0]> = "00"	Output the fs clock				Fixed to "0" or "1".
<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				

Note: When TMPM381/383 is change to the STOP mode, please set SCSTBYCR<PTKEEP> to "1" first and hold the state of a port.

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."

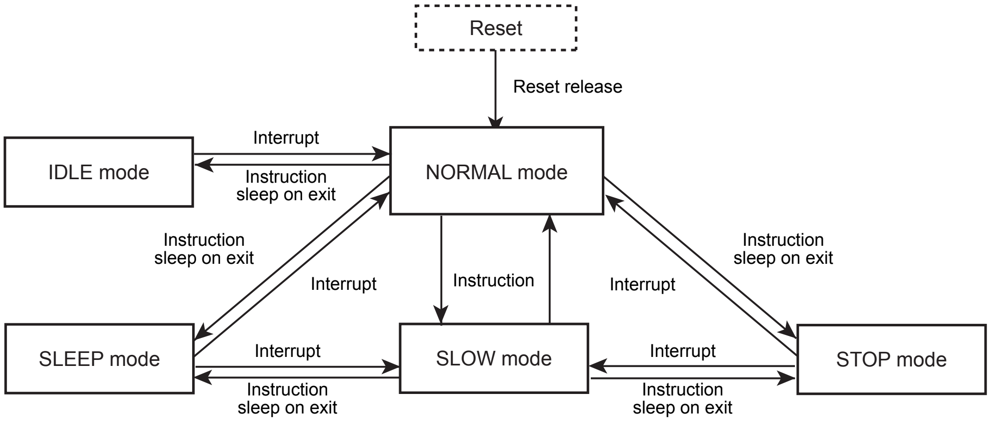


Figure 6-2 Mode Transition Diagram

6.5 Operation mode

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

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 oscillated.

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.

Note 1: In the SLOW mode, CGSYSCR<FPSEL[1:0]> must be set "10" or "11".

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

6.6 Low Power Consumption Modes

The TMPM381/383 has the 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: This product 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: This product 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 IDLE, SLEEP, STOP 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)
- Analog Digital converter (ADC)
- Watch dog timer (WDT)

Note: WDT should be stopped before entering IDLE mode.

6.6.2 SLEEP mode

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

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

6.6.3 STOP mode

Except some peripheral circuits, all the internal circuits including the internal oscillator are brought to a stop in STOP mode.

When releasing STOP mode, the operation mode changes to the operation mode before entering STOP mode.

The STOP mode enables to select the pin status by setting the CGSTBYCR<DRVE>. Table 6-4 shows the pin status in the STOP mode.

Note: Warm-up is need at the time of a return. It is necessary to set warm-up time in the mode (NORMAL mode or SLOW mode) before entering STOP mode. Please refer to "6.6.8 Clock Operations in Mode Transition" about warm-up time.

Table 6-4 Pin States in the STOP mode

Function	Pin Name	I/O	<DRVE> = 0	<DRVE> = 1
Oscillator	X1, XT1	Input	×	×
	X2, XT2	Output	"High" level output.	"High" level output.
PORT	Px	Input	×	Depends on PxIE[m]
		Output	×	Depends on PxCR[m]
Debug	TMS/SWDIO TDO/SWV	Input	Depends on PxIE[m]	
		Output	Depend on PxCR[m] and enable when data is valid	
Interrupt	INT	Input	Depends on PxIE[m]	
SSP	SPCLK, SPFSS, SPDO	Output	×	Depend on PxCR[m] and enable when data is valid
except above		Input	×	Depends on PxIE[m]
		Output	×	Depend on PxCR[m]

o : Valid input or output.

× : Invalid input or output.

Note: x: port number / m: corresponding bit

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 6-5 shows the mode setting in the <STBY[2:0]>.

Table 6-5 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 6-6 show the operational status in each mode.

Table 6-6 Operational Status in Each Mode

Block	NORMAL	IDLE	SLOW	SLEEP	STOP
Processor core	o	×	o	×	×
I/O port	o	o	o	o	o (Note 2)
SSP	o	×	#	#	#
UART	o	o	#	#	#
12/10-bit ADC	o	ON/OFF selectable for each module	#	×	×
SIO/UART	o		#	#	#
I2C/SIO	o		#	#	#
WDT	o		#	#	#
TMRB	o		o	#	#
RMC	o	o	o	o	×
RTC	o	o	o	o	×
CG	o	o	o	o	×
PLL	o	o	×	×	×
OFD	o (Note 3)	o (Note 3)	#	#	#
External high-speed oscillator (f_{EHOSC})	o	o	o (Note 1)	×	×
Internal high-speed oscillator (f_{IHOSC})	o	o	o (Note 1)	×	×
External low-speed oscillator (fs)	o	o	o	o	×

o : Operation is available when in the target mode.

× : The clock to module stops automatically after transiting to the target mode.

# : It is necessary that the select module must be stopped by software before entering in the target mode.

Note 1: The high-speed oscillator (f_{EHOSC} or f_{IHOSC}) does not stop automatically in SLOW mode and must be stopped by setting CGOSCCR1<XEN1> or <XEN2> after switched from NORMAL mode to SLOW mode. The high-speed oscillator (f_{EHOSC} or f_{IHOSC}) does not oscillate automatically in SLOW mode and must be enabled by setting CGOSCCR<XEN1> or <XEN2> before switch to NORMAL mode.

Note 2: The state depends on the CGSTBYCR<DRVE>.

Note 3: When selecting f_{IHOSC} to system clock, OFD can’t be use.

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 6-7.

Table 6-7 Release Source in Each Mode

Low power consumption mode			IDLE	SLEEP	STOP
Release source	Interrupt	INT0 to F (Note1)	o	o	o
		INTRTC	o	o	×
		INTRMCRX	o	o	×
		INTSSP0 to 1	×	×	×
		INTSBI0 to 1	o	×	×
		INTRX0 to 2, INTTX0 to 2	o	×	×
		INTUART0	o	×	×
		INTADCP0 to 1	o	×	×
		INTADTMR /INTADSFT	o	×	×
		INTTB00 to 70 / 01 to 71	o	×	×
		INTCAP00 to 70 / 01 to 71	o	×	×
	SysTick Interrupt		o	×	×
	Non-Maskable Interrupt (INTWDT)		o	×	×
	Reset (WDT)		o	×	×
	Reset (POR)		o	o	o
	Reset (OFD)		o	×	×
	Reset (VLTD)		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: When releasing from IDLE, SLEEP, STOP mode by interrupting level mode, hold the level until the interrupt handling starts. If the level is changed before that, the correct interrupt handling can not be started.

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 interrupt is set to detect interrupt request before entering the low power consumption mode.

  Regarding to setting the interrupt to be used to release the STOP mode, refer to "Exceptions".

- Release by SysTick interrupt

  SysTick interrupt can only be used in the IDLE mode.

- Release by Non-Maskable Interrupt (NMI)

  There are some kinds of NMI sources: WDT interrupt (INTWDT).

  INTWDT can be used only in the IDLE mode.

  INTWDT can not be used in the SLEEP mode or STOP mode. Before entering in the SLEEP mode or STOP mode, INTWDT must be disabled.

- Release by reset

  Any low power consumption mode can be released by a reset from the $\overline{\text{RESET}}$ pin or POR.

  Only IDLE mode can be released by a reset from OFD or WDT. The SLEEP mode and STOP mode can not be released by a reset from OFD or WDT. Before entering in the SLEEP mode or STOP mode, the reset from OFD and WDT must be disabled.

  After that, the mode switches to the NORMAL mode and all the registers are initialized as is the case with normal reset.

  If a reset is used for releasing the STOP mode, the reset signal keeps until the oscillation is stabilized due to do the no warm-up operation.

Refer to "Interrupts" for details.

6.6.7 Warm-up

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

In the mode transition from STOP to the NORMAL/SLOW or from SLEEP to NORMAL, the warm-up counter and the internal oscillator are activated automatically. And then the system clock output is started after the elapse of warm-up time.

It is necessary to set a warm-up source clock in the CGOSCCR<WUPSEL2><WUPSEL1> and to set the warm-up time in the CGOSCCR<WUODR[11:0]><WUODRL[1:0]> before executing the instruction to enter the STOP/SLEEP mode.

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

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

In regard to warm-up time, please refer to "6.6.8 Clock Operations in Mode Transition".

Table 6-8 shows whether the warm-up setting of each mode transition is required or not.

Table 6-8 Warm-up setting in mode transition

Mode transition	Warm-up setting
NORMAL → IDLE	Not required
NORMAL → SLEEP	Not required(Note1)
NORMAL → SLOW	Not required(Note1)
NORMAL → STOP	Not required
IDLE → NORMAL	Not required
SLEEP → NORMAL	Auto-warm-up
SLEEP → SLOW	Not required
SLOW → NORMAL	Not required(Note2)
SLOW → SLEEP	Not required
SLOW → STOP	Not required
STOP → NORMAL	Auto-warm-up (Note 3)
STOP → SLOW	Auto-warm-up

Note 1: If the external low-speed oscillator is disabled, enable the low-speed oscillator and then activate the warm-up by software.

Note 2: If the internal or external high-speed oscillator is disabled, enable the high-speed oscillator and then activate the warm-up by software.

Note 3: When the STOP mode is released by a reset of $\overline{\text{RESET}}$ pin or POR, do not warm-up operation automatically. The reset as same as a cold reset must be input.

6.6.8 Clock Operations in Mode Transition

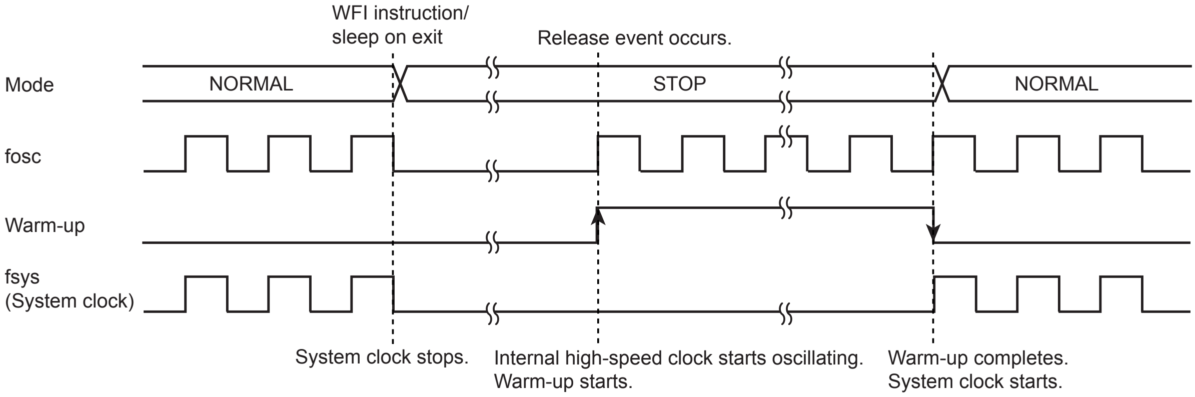
The clock operations in mode transition are described as follows.

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.

The CGOSCCR<WUODR[11:0]> is set to the stable time of an internal or external high-speed oscillator. If PLL is used, the warm-up time must be added a lock-up time (approximate 200μs).

Returning to the NORMAL mode by reset does not induce the automatic warm-up. The reset signal as same as a cold reset should be input.

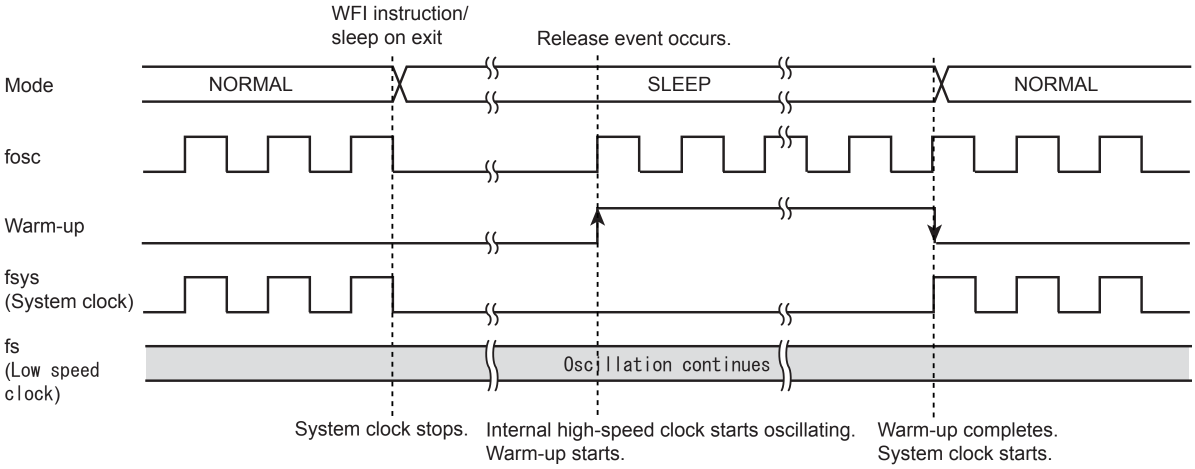


6.6.8.2 Transition of operation modes: NORMAL → SLEEP → NORMAL

When returning to the NORMAL mode from the SLEEP mode, the warm-up is activated automatically.

The CGOSCCR<WUODR[11:0]> and <WUODRL[1:0]> are set to the stable time of an external low-speed oscillator.

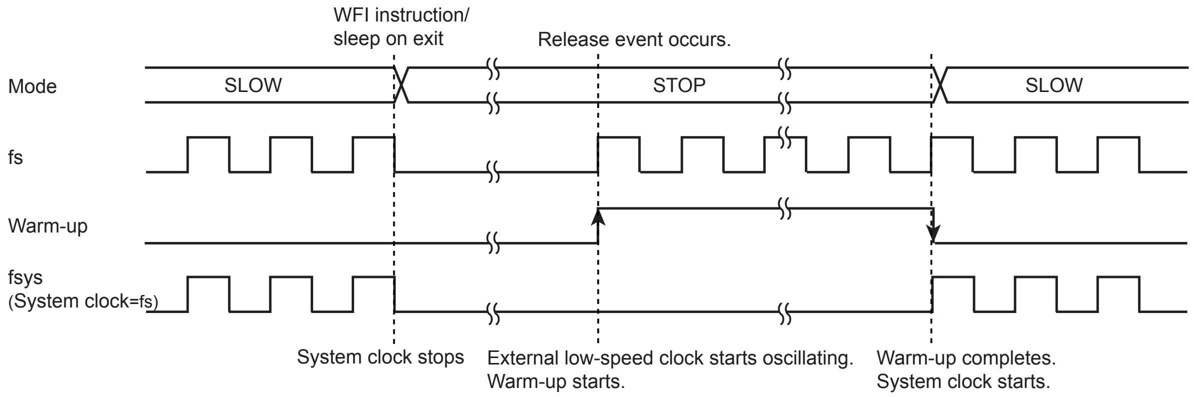
Returning to the NORMAL mode by reset does not induce the automatic warm-up. The reset signal as same as a cold reset should be input.



6.6.8.3 Transition of operation modes: SLOW → STOP → SLOW

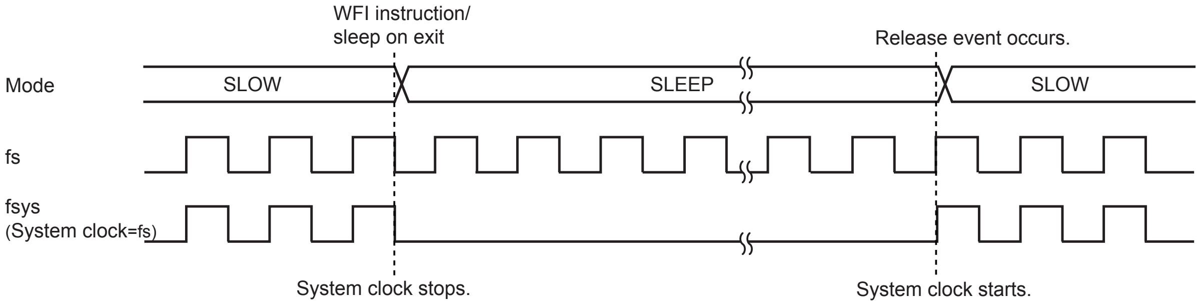
When returning to the SLOW mode from the STOP mode, the warm-up is activated automatically.

The CGOSCCR<WUODR[11:0]> and <WUODRL[1:0]> are set to the stable time of an external low-speed oscillator.



6.6.8.4 Transition of operation modes: SLOW → SLEEP → SLOW

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



6.6.9 Precaution on Transition to the Low-power Consumption Mode

6.6.9.1 Case when the MCU Enters IDLE, SLEEP or STOP Mode

(1) When the WFI instruction is executed to enter IDLE mode, SLEEP mode or STOP mode, if an interrupt request for release from the low-power consumption mode occurs, the MCU does not enter IDLE mode, SLEEP mode or STOP mode. This is because the interrupt request has a higher priority than the WFI instruction. Therefore, the following process must be added depending on enabling or disabling the interrupt:

a. Case when the interrupts are disabled (masked only by PRIMASK)

Write eight or more NOP instructions immediately after the WFI instruction, and then write the instruction to be executed.

b. Case when the interrupts are enabled

Write the interrupt process routine because the MCU branches to the interrupt service routine.

(2) Before the MCU entering SLEEP mode or STOP mode, select the clock with CGOSCCR<WUPSEL1><WUPSEL2>, which is the same as the clock selected with CGOSCCR<OSCSEL>, to use the same source clock for both the warm-up-counter and fosc.

(3) A non-maskable interrupt can be used to release only in IDLE mode.

(4) Do not use non-maskable interrupts as a release factor of SLEEP mode or STOP mode. Before the MCU entering SLEEP mode or STOP mode, inhibit non-maskable interrupts, specify as follows: Stop the watch-dog timer, Stop the voltage detection.

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.

Processing	Description	See
Detection by CG/CPU	The CG/CPU detects the exception request.	Section 7.1.2.1
⇩		
Handling by CPU	The CPU handles the exception request.	Section 7.1.2.2
⇩		
Branch to ISR	The CPU branches to the corresponding interrupt service routine (ISR).	
⇩		
Execution of ISR	Necessary processing is executed.	Section 7.1.2.3
⇩		
Return from exception	The CPU branches to another ISR or returns to the previous program.	Section 7.1.2.4

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, POR, OFD, VLTD, SYSRETREQ
2	Non-Maskable Interrupt	−2	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)

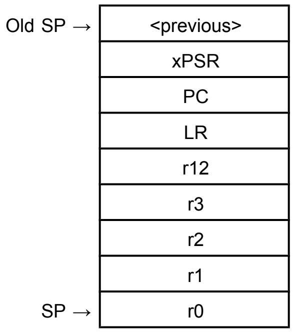
When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called "pre-emption".

(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 five 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.

- Reset exception by POR

  A reset exception occurs when the power is turned on. For details, see the chapter on the POR.

- Reset exception by OFD

  The oscillation frequency detection (OFD) has a reset generating feature. For details, see the chapter on the OFD.

- Reset exception by VLTD

  The Voltage Level Detector (VLTD) has a reset generating feature. For details, see the chapter on the VLTD.

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.

- 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 systick timer counts based on an external reference clock obtained by 32 dividing the fosc which is selected by the bits <OSCSEL> of the register CGOSCCR.

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 (route 1).

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).

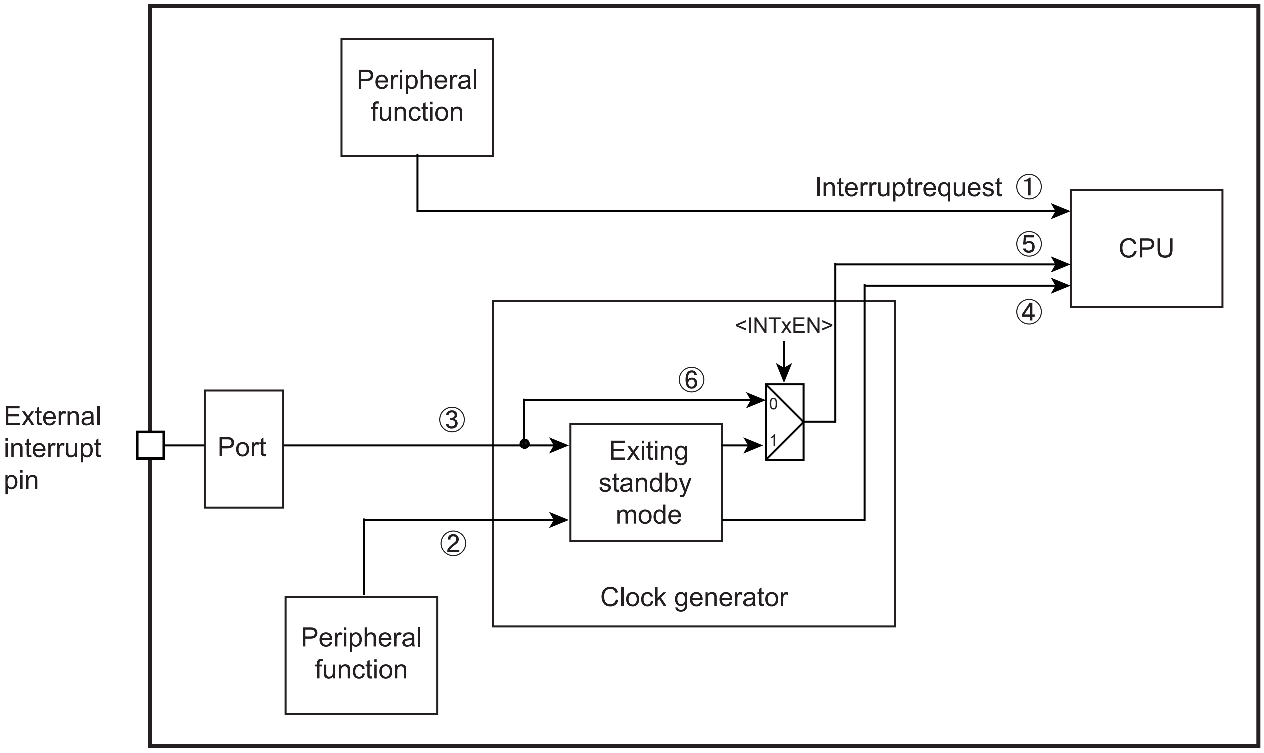


Figure 7-1 Interrupt Route

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 Lists of Interrupt Sources

No.	Interrupt Source		The active level to release the low power consumption mode					CG interrupt mode control register
			"Low" level	"High" level	Rising edge	Falling edge	Both edge	
0	INT0	External interrupt pin 0	o	o	o	o	o	CGIMCGA
1	INT1	External interrupt pin 1	o	o	o	o	o	
2	INT2	External interrupt pin 2	o	o	o	o	o	
3	INT3	External interrupt pin 3	o	o	o	o	o	
4	INT4	External interrupt pin 4	o	o	o	o	o	CGIMCGB
5	INT5	External interrupt pin 5	o	o	o	o	o	
6	INTRX0	SC receive interrupt (Channel 0)						
7	INTTX0	SC transmit interrupt (Channel 0)						
8	INTRX1	SC receive interrupt (Channel 1)						
9	INTTX1	SC transmit interrupt (Channel 1)						
10	INTSSP0	SSP serial interface (channel 0)						
11	Reserved	Reserved						
12	Reserved	Reserved						
13	Reserved	Reserved						
14	INTSBI0	Serial bus interface 0						
15	Reserved	Reserved						
16	Reserved	Reserved						
17	INTRTC	RTC interrupt	×	×	×	o	×	CGIMCGE
18	Reserved	Reserved						
19	INTRMCRX	Remote Controller reception interrupt	×	×	o	×	×	CGIMCGE
20	INTTB00	16-bit TMRB compare match detection 0 / overflow(channel 0)						
21	INTTB01	16-bit TMRB compare match detection 1 (channel 0)						
22	INTTB10	16-bit TMRB compare match detection 0 / overflow(channel 1)						
23	INTTB11	16-bit TMRB compare match detection 1 (channel 1)						
24	INTTB40	16-bit TMRB compare match detection 0 / overflow(channel 4)						
25	INTTB41	16-bit TMRB compare match detection 1 (channel 4)						
26	INTTB50	16-bit TMRB compare match detection 0 / overflow(channel 5)						
27	INTTB51	16-bit TMRB compare match detection 1 (channel 5)						
28	Reserved	Reserved						
29	Reserved	Reserved						
30	INTCAP00	16-bit TMRB input capture 0 (channel 0)						
31	INTCAP01	16-bit TMRB input capture 1 (channel 0)						
32	INTCAP10	16-bit TMRB input capture 0 (channel 1)						
33	INTCAP11	16-bit TMRB input capture 1 (channel 1)						
34	INTCAP40	16-bit TMRB input capture 0 (channel 4)						
35	INTCAP41	16-bit TMRB input capture 1 (channel 4)						
36	INTCAP50	16-bit TMRB input capture 0 (channel 5)						

Table 7-3 Lists of Interrupt Sources

No.	Interrupt Source		The active level to release the low power consumption mode					CG interrupt mode control register
			"Low" level	"High" level	Rising edge	Falling edge	Both edge	
37	INTCAP51	16-bit TMRB input capture 1 (channel 5)						
38	INT6	External interrupt pin 6	o	o	o	o	o	CGIMCGB
39	INT7	External interrupt pin 7	o	o	o	o	o	
40	INTRX2	SC receive interrupt (Channel 2)						
41	INTTX2	SC transmit interrupt (Channel 2)						
42	INTADCP0	ADC conversion monitoring function interrupt 0						
43	INTADCP1	ADC conversion monitoring function interrupt 1						
44	INTUART0	UART interrupt (channel 0)						
45	Reserved	Reserved						
46	INTTB20	16-bit TMRB compare match detection 0 / overflow(channel 2)						
47	INTTB21	16-bit TMRB compare match detection 1 (channel 2)						
48	INTTB30	16-bit TMRB compare match detection 0 / overflow(channel 3)						
49	INTTB31	16-bit TMRB compare match detection 1 (channel 3)						
50	INTCAP20	16-bit TMRB input capture 0 (channel 2)						
51	INTCAP21	16-bit TMRB input capture 1 (channel 2)						
52	INTCAP30	16-bit TMRB input capture 0 (channel 3)						
53	INTCAP31	16-bit TMRB input capture 1 (channel 3)						
54	INTADSFT	ADC conversion started by software is finished						
55	Reserved	Reserved						
56	INTADTMR	ADC conversion triggered by timer is finished						
57	Reserved	Reserved						
58	INT8	External interrupt pin 8	o	o	o	o	o	CGIMCGC
59	INT9	External interrupt pin 9	o	o	o	o	o	
60	INTA	External interrupt pin A	o	o	o	o	o	
61	INTB	External interrupt pin B	o	o	o	o	o	
62	Reserved	Reserved						
63	Reserved	Reserved						
64	Reserved	Reserved						
65	Reserved	Reserved						
66	INTTB60	16-bit TMRB compare match detection 0 / overflow(channel 6)						
67	INTTB61	16-bit TMRB compare match detection 1 (channel 6)						
68	INTTB70	16-bit TMRB compare match detection 0 / overflow(channel 7)						
69	INTTB71	16-bit TMRB compare match detection 1 (channel 7)						
70	INTCAP60	16-bit TMRB input capture 0 (channel 6)						
71	INTCAP61	16-bit TMRB input capture 1 (channel 6)						
72	INTCAP70	16-bit TMRB input capture 0 (channel 7)						
73	INTCAP71	16-bit TMRB input capture 1 (channel 7)						

Table 7-3 Lists of Interrupt Sources

No.	Interrupt Source		The active level to release the low power consumption mode					CG interrupt mode control register
			"Low" level	"High" level	Rising edge	Falling edge	Both edge	
74	INTC	External interrupt pin C	o	o	o	o	o	CGIMCGD
75	INTD	External interrupt pin D	o	o	o	o	o	
76	INTE	External interrupt pin E	o	o	o	o	o	
77	INTF	External interrupt pin F	o	o	o	o	o	

Note: The active level marked with "o" is used for release of low power consumption mode. The active level marked with "×" cannot be used.

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.

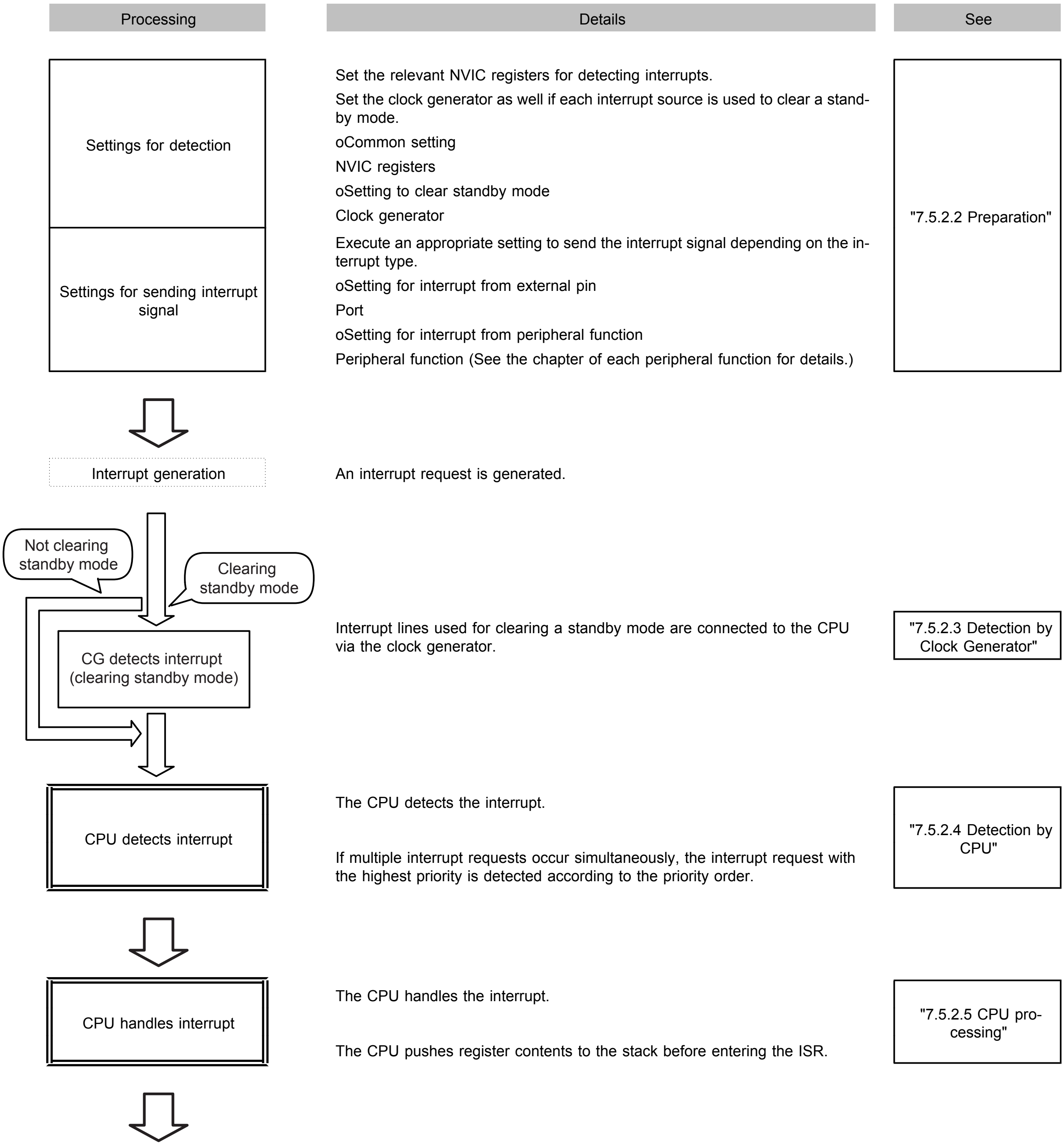
An interrupt request detected by the clock generator is notified to the CPU with a signal in "High" level.

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
Settings for detection	Set the relevant NVIC registers for detecting interrupts. Set the clock generator as well if each interrupt source is used to clear a standby mode. oCommon setting NVIC registers oSetting to clear standby mode Clock generator	"7.5.2.2 Preparation"
Settings for sending interrupt signal	Execute an appropriate setting to send the interrupt signal depending on the interrupt type. oSetting for interrupt from external pin Port oSetting for interrupt from peripheral function Peripheral function (See the chapter of each peripheral function for details.)	
Interrupt generation	An interrupt request is generated.	
CG detects interrupt (clearing standby mode)	Interrupt lines used for clearing a standby mode are connected to the CPU via the clock generator.	"7.5.2.3 Detection by Clock Generator"
CPU detects interrupt	The CPU detects the interrupt. If multiple interrupt requests occur simultaneously, the interrupt request with the highest priority is detected according to the priority order.	"7.5.2.4 Detection by CPU"
CPU handles interrupt	The CPU handles the interrupt. The CPU pushes register contents to the stack before entering the ISR.	"7.5.2.5 CPU processing"

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 "7.6.3.2 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(Base+)
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 Set-Enable Register 3	0x0108
Interrupt Clear-Enable Register 1	0x0180
Interrupt Clear-Enable Register 2	0x0184
Interrupt Clear-Enable Register 3	0x0188
Interrupt Set-Pending Register 1	0x0200
Interrupt Set-Pending Register 2	0x0204
Interrupt Set-Pending Register 3	0x0208
Interrupt Clear-Pending Register 1	0x0280
Interrupt Clear-Pending Register 2	0x0284
Interrupt Clear-Pending Register 3	0x0288
Interrupt Priority Register	0x0400 ~ 0x045C
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(Base+)
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
CG Interrupt Mode Control Register E	CGIMCGE	0x0030

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 (fosc/32) 1: CPU clock (fsys)
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.

Note: In this product, the systick timer counts based on an external reference clock obtained by 32 dividing the fosc which is selected by the bits <OSCSEL> of the register CGOSCCR.

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".

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: **When using a multi-shot timer, the calibration value is subtracted 1 from this value and use it.**

7.6.2.5 Interrupt Control Registers

Each interrupt source has the interrupt set-enable register, interrupt clear-enable register, interrupt set-pending register and interrupt clear-pending register.

Each bit corresponds to each interrupt source.

(1) Interrupt Set-Enable Register

This register enables interrupts and identifies whether the interrupt is enabled/disabled.

When set this register to "1", the corresponding interrupt is enabled.

Writing "0" has no meaning.

When this register is read, whether the corresponding interrupt is enabled or disabled is identified.

To clear the bit of this register, set "1" to the corresponding bit of the interrupt clear-enable register.

Bit symbol	Type	Function
SETENA	R/W	Interrupt No. [77:0] [Write] 1: Enables interrupts. [Read] 0: Interrupts are disabled. 1: Interrupts are enabled.

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

(a) Interrupt Set-Enable Register 1

	31	30	29	28	27	26	25	24
Bit symbol	SETENA (Interrupt 31)	SETENA (Interrupt 30)	-	-	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 17)	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	SETENA (Interrupt 14)	-	-	-	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

(b) Interrupt Set-Enable Register 2

	31	30	29	28	27	26	25	24
Bit symbol	-	-	SETENA (Interrupt 61)	SETENA (Interrupt 60)	SETENA (Interrupt 59)	SETENA (Interrupt 58)	-	SETENA (Interrupt 56)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	SETENA (Interrupt 54)	SETENA (Interrupt 53)	SETENA (Interrupt 52)	SETENA (Interrupt 51)	SETENA (Interrupt 50)	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 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

(c) Interrupt Set-Enable 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	-	-	SETENA (Interrupt 77)	SETENA (Interrupt 76)	SETENA (Interrupt 75)	SETENA (Interrupt 74)	SETENA (Interrupt 73)	SETENA (Interrupt 72)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	SETENA (Interrupt 71)	SETENA (Interrupt 70)	SETENA (Interrupt 69)	SETENA (Interrupt 68)	SETENA (Interrupt 67)	SETENA (Interrupt 66)	-	-
After reset	0	0	0	0	0	0	0	0

(2) Interrupt Clear-Enable Register

This register disables interrupts and identifies whether the interrupt is enabled/disabled.

When set this register to "1", the corresponding interrupt is disabled.

Writing "0" has no meaning.

When this register is read, whether the corresponding interrupt is enabled or disabled is identified.

Bit symbol	Type	Function
CLRENA	R/W	Interrupt No. [77:0] [Write] 1: Disables interrupts. [Read] 0: Interrupts are disabled. 1: Interrupts are enabled.

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

(a) Interrupt Clear-Enable Register 1

	31	30	29	28	27	26	25	24
Bit symbol	CLRENA (Interrupt 31)	CLRENA (Interrupt 30)	-	-	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 17)	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	CLRENA (Interrupt 14)	-	-	-	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

(b) Interrupt Clear-Enable Register 2

	31	30	29	28	27	26	25	24
Bit symbol	-	-	CLRENA (Interrupt 61)	CLRENA (Interrupt 60)	CLRENA (Interrupt 59)	CLRENA (Interrupt 58)	-	CLRENA (Interrupt 56)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	CLRENA (Interrupt 54)	CLRENA (Interrupt 53)	CLRENA (Interrupt 52)	CLRENA (Interrupt 51)	CLRENA (Interrupt 50)	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 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

(c) Interrupt Clear-Enable 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	-	-	CLRENA (Interrupt 77)	CLRENA (Interrupt 76)	CLRENA (Interrupt 75)	CLRENA (Interrupt 74)	CLRENA (Interrupt 73)	CLRENA (Interrupt 72)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	CLRENA (Interrupt 71)	CLRENA (Interrupt 70)	CLRENA (Interrupt 69)	CLRENA (Interrupt 68)	CLRENA (Interrupt 67)	CLRENA (Interrupt 66)	-	-
After reset	0	0	0	0	0	0	0	0

(3) Interrupt Set-Pending Register

This register forcibly suspends interrupts and identifies whether interrupts are suspended.

When this register is set to "1", the corresponding interrupt is suspended. However, this register is in-valid for the interrupt which has already been suspended or disabled.

Writing "0" has no meaning.

When this register is read, whether the corresponding interrupt is suspended or not.

Bit symbol	Type	Function
SETPEND	R/W	Interrupt No. [77:0] [Write] 1: Interrupts are suspended. [Read] 0: No pending interrupts. 1: Pending interrupts exist.

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

(a) Interrupt Set-Pending Register 1

	31	30	29	28	27	26	25	24
Bit symbol	SETPEND (Interrupt 31)	SETPEND (Interrupt 30)	-	-	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 17)	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
Bit symbol	-	SETPEND (Interrupt 14)	-	-	-	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

(b) Interrupt Set-Pending Register 2

	31	30	29	28	27	26	25	24
Bit symbol	-	-	SETPEND (Interrupt 61)	SETPEND (Interrupt 60)	SETPEND (Interrupt 59)	SETPEND (Interrupt 58)	-	SETPEND (Interrupt 56)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
Bit symbol	-	SETPEND (Interrupt 54)	SETPEND (Interrupt 53)	SETPEND (Interrupt 52)	SETPEND (Interrupt 51)	SETPEND (Interrupt 50)	SETPEND (Interrupt 49)	SETPEND (Interrupt 48)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
Bit symbol	SETPEND (Interrupt 47)	SETPEND (Interrupt 46)	-	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

(c) Interrupt Set-Pending Register 3

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
Bit symbol	-	-	SETPEND (Interrupt 77)	SETPEND (Interrupt 76)	SETPEND (Interrupt 75)	SETPEND (Interrupt 74)	SETPEND (Interrupt 73)	SETPEND (Interrupt 72)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
Bit symbol	SETPEND (Interrupt 71)	SETPEND (Interrupt 70)	SETPEND (Interrupt 69)	SETPEND (Interrupt 68)	SETPEND (Interrupt 67)	SETPEND (Interrupt 66)	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

(4) Interrupt Clear-Pending Register

This register clears pending interrupts and identifies whether interrupts are suspended.

When set this register to "1", the corresponding pending interrupt is cleared. However, this register is invalid for the interrupt which has already been started.

Writing "0" has no meaning.

When this register is read, it indicates whether the corresponding interrupt is suspended.

Bit symbol	Type	Function
CLRPEND	R/W	Interrupt No. [77:0] [Write] 1: Clears pending interrupts [Read] 0: No pending interrupts. 1: Pending interrupts exists.

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

(a) Interrupt Clear-Pending Register 1

	31	30	29	28	27	26	25	24
Bit symbol	CLRPEND (Interrupt 31)	CLRPEND (Interrupt 30)	-	-	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 17)	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
Bit symbol	-	CLRPEND (Interrupt 14)	-	-	-	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

(b) Interrupt Clear-Pending Register 2

	31	30	29	28	27	26	25	24
Bit symbol	-	-	CLRPEND (Interrupt 61)	CLRPEND (Interrupt 60)	CLRPEND (Interrupt 59)	CLRPEND (Interrupt 58)	-	CLRPEND (Interrupt 56)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
Bit symbol	-	CLRPEND (Interrupt 54)	CLRPEND (Interrupt 53)	CLRPEND (Interrupt 52)	CLRPEND (Interrupt 51)	CLRPEND (Interrupt 50)	CLRPEND (Interrupt 49)	CLRPEND (Interrupt 48)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
Bit symbol	CLRPEND (Interrupt 47)	CLRPEND (Interrupt 46)	-	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

(c) Interrupt Clear-Pending Register 3

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
Bit symbol	-	-	CLRPEND (Interrupt 77)	CLRPEND (Interrupt 76)	CLRPEND (Interrupt 75)	CLRPEND (Interrupt 74)	CLRPEND (Interrupt 73)	CLRPEND (Interrupt 72)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
Bit symbol	CLRPEND (Interrupt 71)	CLRPEND (Interrupt 70)	CLRPEND (Interrupt 69)	CLRPEND (Interrupt 68)	CLRPEND (Interrupt 67)	CLRPEND (Interrupt 66)	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

7.6.2.6 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_10	PRI_9	PRI_8
0xE000_E40C	-	PRI_14	-	-
0xE000_E410	PRI_19	-	PRI_17	-
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	-	-
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_44
0xE000_E430	PRI_51	PRI_50	PRI_49	PRI_48
0xE000_E434	-	PRI_54	PRI_53	PRI_52
0xE000_E438	PRI_59	PRI_58	-	PRI_56
0xE000_E43C	-	-	PRI_61	PRI_60
0xE000_E440	PRI_67	PRI_66	-	-
0xE000_E444	PRI_71	PRI_70	PRI_69	PRI_68
0xE000_E448	PRI_75	PRI_74	PRI_73	PRI_72
0xE000_E44C	-	-	PRI_77	PRI_76

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.

7.6.2.7 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.

7.6.2.8 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.

7.6.2.9 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.

7.6.2.10 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 CG Interrupt Mode Control Register

This CG interrupt mode control register specifies the active level to release the low power consumption mode and enables/disables the releasing low power consumption mode. A detected active level can also be read from this register.

Bit symbol	Type	Function
EMCGx[2:0]	R/W	Sets the active level for release of low power consumption mode. The factor of active level can be selected from Table 7-3. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Settings other than the above: Prohibited
EMSTx[1:0]	R	Detected active level (This bit is valid only when EMCGx[2:0]="100".) 00: − 01: Rising edge 10: Falling edge 11: Both edges
INTxEN	R/W	Release the low power consumption mode 0: Disabled 1: Enabled

Table 7-4 The Active Level to release low-power consumption mode

Interrupt factor		Active level setting register	The active level to release the low-power consumption mode				
			"Low" level	"High" level	Rising edge	Falling edge	Both edge
INT0	External interrupt pin 0	CGIMCGA <EMCGx[2:0]>	o	o	o	o	o
INT1	External interrupt pin 1	CGIMCGA <EMCG1[2:0]>	o	o	o	o	o
INT2	External interrupt pin 2	CGIMCGA <EMCG2[2:0]>	o	o	o	o	o
INT3	External interrupt pin 3	CGIMCGA <EMCG3[2:0]>	o	o	o	o	o
INT4	External interrupt pin 4	CGIMCGB <EMCG4[2:0]>	o	o	o	o	o
INT5	External interrupt pin 5	CGIMCGB <EMCG5[2:0]>	o	o	o	o	o
INT6	External interrupt pin 6	CGIMCGB <EMCG6[2:0]>	o	o	o	o	o
INT7	External interrupt pin 7	CGIMCGB <EMCG7[2:0]>	o	o	o	o	o
INT8	External interrupt pin 8	CGIMCGC <EMCG8[2:0]>	o	o	o	o	o
INT9	External interrupt pin 9	CGIMCGC <EMCG9[2:0]>	o	o	o	o	o
INTA	External interrupt pin A	CGIMCGC <EMCGA[2:0]>	o	o	o	o	o
INTB	External interrupt pin B	CGIMCGC <EMCGB[2:0]>	o	o	o	o	o
INTC	External interrupt pin C	CGIMCGD <EMCGC[2:0]>	o	o	o	o	o
INTD	External interrupt pin D	CGIMCGD <EMCGD[2:0]>	o	o	o	o	o
INTE	External interrupt pin E	CGIMCGD <EMCGE[2:0]>	o	o	o	o	o
INTF	External interrupt pin F	CGIMCGD <EMCGF[2:0]>	o	o	o	o	o
INTRTC	RTC interrupt	CGIMCGE <EMCGRTC[2:0]>	×	×	×	o	×
INTRMCRX	Remote Controller reception interrupt	CGIMCGH <EMCGRMCRX[2:0]>	×	×	o	×	×

Note: The active level marled with "o" for release of low power consumption mode can be used. The active level marked with "×" cannot be used.

(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

Note 1: The active level specified by <EMCGx[2:0]> varies depending on the interrupt request. Refer to Table 7-4.

Note 2: <EMSTx> is valid only when <EMCGx[2:0]> is set to "100" for both on rising and falling edges. In the other cases, the value is undefined. The active level used for release of low-power consumption mode can be checked by reading <EMSTx>. If interrupts are cleared with the CGICRCG register, <EMSTx> is also cleared.

Note 3: Do not specify <INTxEN> when the edge is selected. Select the edge first and then specify <INTxEN>. Setting them simultaneously is prohibited.

Note 4: "0" is read from bits 31, 23, 15 and 7.

Note 5: Undefined value is read from bits 25, 17, 9 and 1.

(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

Note 1: The active level specified by <EMCGx[2:0]> varies depending on the interrupt request. Refer to Table 7-4.

Note 2: <EMSTx> is valid only when <EMCGx[2:0]> is set to "100" for both on rising and falling edges. In the other cases, the value is undefined. The active level used for release of low-power consumption mode can be checked by reading <EMSTx>. If interrupts are cleared with the CGICRCG register, <EMSTx> is also cleared.

Note 3: Do not specify <INTxEN> when the edge is selected. Select the edge first and then specify <INTxEN>. Setting them simultaneously is prohibited.

Note 4: "0" is read from bits 31, 23, 15 and 7.

Note 5: Undefined value is read from bits 25, 17, 9 and 1.

(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		-	INT09EN
After reset	0	0	1	0	0	0	undefined	0
	7	6	5	4	3	2	1	0
Bit symbol	-	EMCG8			EMST8		-	INT08EN
After reset	0	0	1	0	0	0	undefined	0

Note 1: The active level specified by <EMCGx[2:0]> varies depending on the interrupt request. Refer to Table 7-4.

Note 2: <EMSTx> is valid only when <EMCGx[2:0]> is set to "100" for both on rising and falling edges. In the other cases, the value is undefined. The active level used for release of low-power consumption mode can be checked by reading <EMSTx>. If interrupts are cleared with the CGICRCG register, <EMSTx> is also cleared.

Note 3: Do not specify <INTxEN> when the edge is selected. Select the edge first and then specify <INTxEN>. Setting them simultaneously is prohibited.

Note 4: "0" is read from bits 31, 23, 15 and 7.

Note 5: Undefined value is read from bits 25, 17, 9 and 1.

(4) CGIMCGD(CG Interrupt Mode Control Register D)

	31	30	29	28	27	26	25	24
Bit symbol	-	EMCGF			EMSTF		-	INTFEN
After reset	0	0	1	0	0	0	undefined	0
	23	22	21	20	19	18	17	16
Bit symbol	-	EMCGE			EMSTE		-	INTEEN
After reset	0	0	1	0	0	0	undefined	0
	15	14	13	12	11	10	9	8
Bit symbol	-	EMCGD			EMSTD		-	INTDEN
After reset	0	0	1	0	0	0	undefined	0
	7	6	5	4	3	2	1	0
Bit symbol	-	EMCGC			EMSTC		-	INTCEN
After reset	0	0	1	0	0	0	undefined	0

Note 1: The active level specified by <EMCGx[2:0]> varies depending on the interrupt request. Refer to Table 7-4.

Note 2: <EMSTx> is valid only when <EMCGx[2:0]> is set to "100" for both on rising and falling edges. In the other cases, the value is undefined. The active level used for release of low-power consumption mode can be checked by reading <EMSTx>. If interrupts are cleared with the CGICRCG register, <EMSTx> is also cleared.

Note 3: Do not specify <INTxEN> when the edge is selected. Select the edge first and then specify <INTxEN>. Setting them simultaneously is prohibited.

Note 4: "0" is read from bits 31, 23, 15 and 7.

Note 5: Undefined value is read from bits 25, 17, 9 and 1.

(5) CGIMCGE(CG Interrupt Mode Control Register E)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	1	0	0	0	undefined	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	1	0	0	0	undefined	0
	15	14	13	12	11	10	9	8
Bit symbol	-	EMCGRMCRX			EMSTRMCRX		-	INTRMCRXEN
After reset	0	0	1	0	0	0	undefined	0
	7	6	5	4	3	2	1	0
Bit symbol	-	EMCGRTC			EMSTRTC		-	INTRTCEN
After reset	0	0	1	0	0	0	undefined	0

Note 1: The active level specified by <EMCGx[2:0]> varies depending on the interrupt request. Refer to Table 7-4.

Note 2: <EMSTx> is valid only when <EMCGx[2:0]> is set to "100" for both on rising and falling edges. In the other cases, the value is undefined. The active level used for release of low-power consumption mode can be checked by reading <EMSTx>. If interrupts are cleared with the CGICRCG register, <EMSTx> is also cleared.

Note 3: Do not specify <INTxEN> when the edge is selected. Select the edge first and then specify <INTxEN>. Setting them simultaneously is prohibited.

Note 4: "0" is read from bits 31, 23, 15 and 7.

Note 5: Undefined value is read from bits 25, 17, 9 and 1.

7.6.3.2 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: INT6 0_0111: INT7 0_1000: INT8 0_1001: INT9 0_1010: INTA 0_1011: INTB 0_1100: INTC 0_1101: INTD 0_1110: INTE 0_1111: INTF 1_0000: INTRTC 1_0001: INTRMCRX 1_00010 to 1_1111: setting prohibited. Read as "0".

7.6.3.3 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	-	-	-	-	-	-	-	NMIFLG0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2-1	−	R	Reads as undefined.
0	NMIFLG0	R	NMI source generation flag 0: not applicable 1: generated from WDT

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

7.6.3.4 CGRSTFLG (Reset Flag Register)

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

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as 0.
5	OFDRSTF	R/W	OFD reset flag 0: "0" is written 1: Reset from OFD
4	DBGRSTF	R/W	Debug reset flag (Note1) 0: "0" is written 1: Reset from SYSRESETREQ
3	VLTDRSTF	R/W	VLTD reset flag 0: "0" is written 1: Reset from VLTD
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 $\overline{\text{RESET}}$ pin.
0	PONRSTF	R/W	Power On reset flag 0: "0" is written 1: Reset from Power On reset

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 product has power-on reset circuit and this register is initialized only by power-on reset.Therefore,"1" is set to the <PONRSTF> bit in initial reset state right after power-on.**This bit is not set by the second and subsequent resets.

8. Digital Noise Filter Circuit (DNF)

The digital noise canceler circuit can eliminate noise of input signals from external interrupt pins at the certain range.

8.1 Configuration

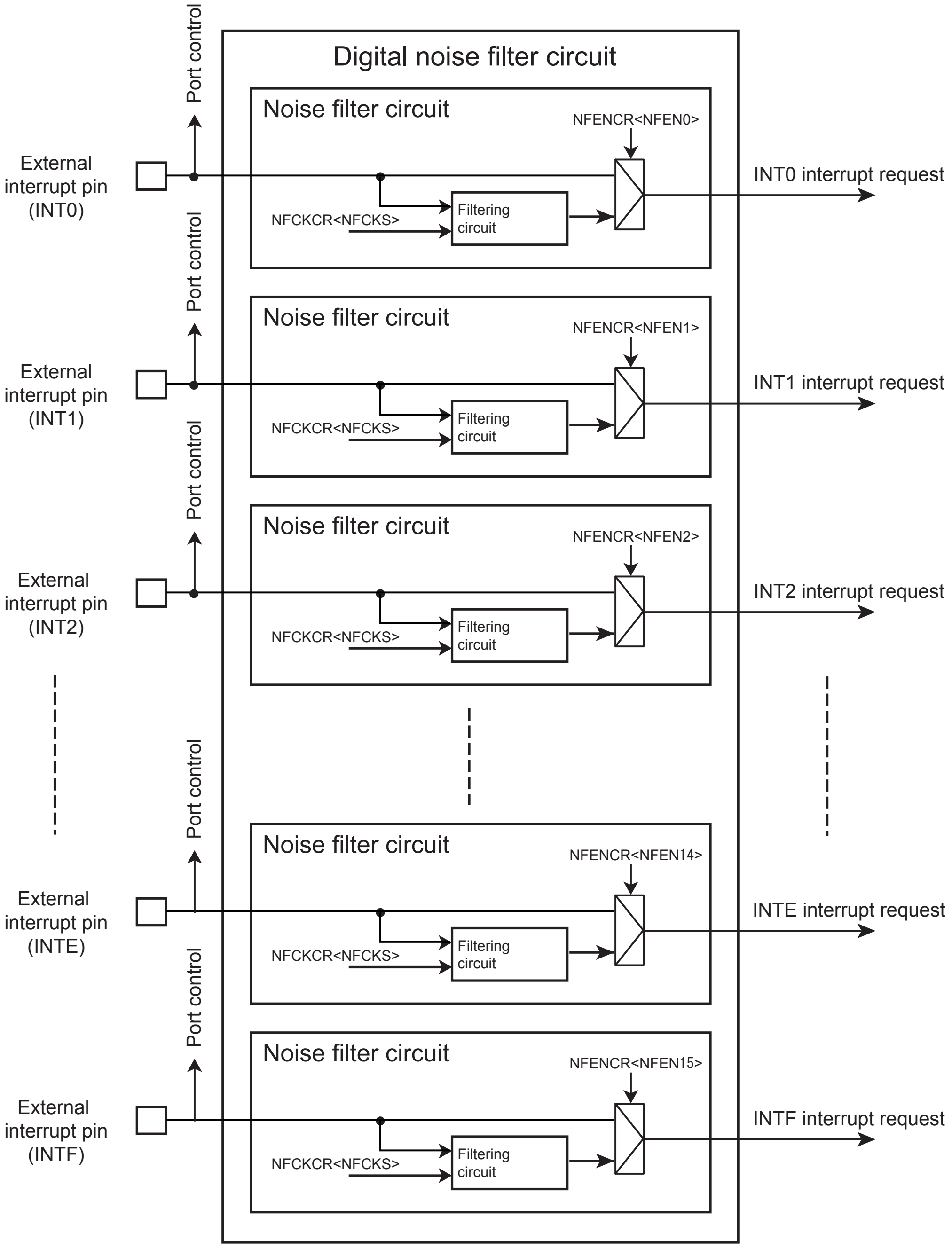


Figure 8-1 Circuit diagram of digital noise filter

8.2 Registers

8.2.1 Register List

Base Address = 0x4006_0000

Register name		Address(Base+)
Noise filter control register	NFCKCR	0x0000
Noise filter enable register	NFENCR	0x0004

8.2.1.1 NFCKCR (Noise Filter 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	-	-	-	-	-	NFCKS		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as "0".
2-0	NFCKS[2:0]	R/W	Noise filter clock selection 000: Clock control circuit stops 001: fsys/2 clock output 010: fsys/4 clock output 011: fsys/8 clock output 100: fsys/16 clock output 101: fsys/32 clock output 110: fsys/64 clock output 111: fsys/128 clock output

Note: NFCKCR<NFCKS> setting is specified in NFENCR<NFEN[2:0]>="000".

Note: If external inputs are used to release STOP mode, the noise filter circuit cannot be used. Make sure to disable the noise filter enable bit of NFENCR register and stop the clock by NFCKCR register.

8.2.1.2 NFENCR (Noise Filter 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	NFENF	NFENE	NFEND	NFENC	NFENB	NFENA	NFEN9	NFEN8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	NFEN7	NFEN6	NFEN5	NFEN4	NFEN3	NFEN2	NFEN1	NFEN0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15	NFENF	R/W	INTF noise filter is enabled/disabled. 0: Disabled 1: Enabled
14	NFENE	R/W	INTE noise filter is enabled/disabled. 0: Disabled 1: Enabled
13	NFEND	R/W	INTD noise filter is enabled/disabled. 0: Disabled 1: Enabled
12	NFENC	R/W	INTC noise filter is enabled/disabled. 0: Disabled 1: Enabled
11	NFENB	R/W	INTB noise filter is enabled/disabled. 0: Disabled 1: Enabled
10	NFENA	R/W	INTA noise filter is enabled/disabled. 0: Disabled 1: Enabled
9	NFEN9	R/W	INT9 noise filter is enabled/disabled. 0: Disabled 1: Enabled
8	NFEN8	R/W	INT8 noise filter is enabled/disabled. 0: Disabled 1: Enabled
7	NFEN7	R/W	INT7 noise filter is enabled/disabled. 0: Disabled 1: Enabled
6	NFEN6	R/W	INT6 noise filter is enabled/disabled. 0: Disabled 1: Enabled
5	NFEN5	R/W	INT5 noise filter is enabled/disabled. 0: Disabled 1: Enabled
4	NFEN4	R/W	INT4 noise filter is enabled/disabled. 0: Disabled 1: Enabled

Bit	Bit Symbol	Type	Function
3	NFEN3	R/W	INT3 noise filter is enabled/disabled. 0: Disabled 1: Enabled
2	NFEN2	R/W	INT2 noise filter is enabled/disabled. 0: Disabled 1: Enabled
1	NFEN1	R/W	INT1 noise filter is enabled/disabled. 0: Disabled 1: Enabled
0	NFEN0	R/W	INT0 noise filter is enabled/disabled. 0: Disabled 1: Enabled

Note: Disabled (Pre-noise filtering output signal and noise filter circuit counter are cleared when releasing STOP mode.)

Note: Enabled (Post-noise filtering output signal)

Note: Some pulses shorter than fsys cannot be filtered noise. Especially, in the case that fsys frequency is low, noise filtering operation may not be effective.

Note: Before external interrupt signals are enabled, clear the interrupt events and then set the corresponding bit of NFENCR register to be enabled.

Note: If external inputs are used to release STOP mode, refer to "8.3.4 Precautions on Use of STOP Mode"

8.3 Operation Description

8.3.1 Configuration

The noise filter circuit consists of the noise filter circuit and interrupt request generation circuit.

It eliminates high level or low level noise from external inputs and then CG detects the rising/falling edge or signal level (high or low) to determine the signal state in each interrupt signal.

8.3.2 Operation

The noise filter eliminates high and low level noise from the external interrupt input INTx.

A noise filtering time is determined by the input level continuation time specified in NFCKCR<NFCKS>. If the time is less than 7 clocks, the input is determined as noise. If the time is over 8 clocks, the input is determined as an invalid signal. However, the determination of an input signal for 7 to 8 clocks varies depending on the edge timing.

8.3.3 Noise Filter Usable Operation Mode

The noise filter circuit can be used only in the NORMAL mode and IDLE mode.

8.3.4 Precautions on Use of STOP Mode

If STOP mode is used, the noise filter circuit cannot be used due to a stop of fsys clock. If external input are used to release STOP mode, set the following procedure: Set the interrupt enable bit to be disabled; set the noise filter enable/disable bit of NFENCR register; and stop the noise filter clock of NFCKCR register.

8.3.5 Minimum Noise Filtering Time

The noise filter circuit determines input levels to send the external interrupt signals if high level or low level inputs are continued to input over 8 clock periods specified in NFCKCR register.

Table 8-1 Minimum noise filtering time

NFCKCR<NFCKS>	fsys [MHz]			Unit
	20	32	40	
001	0.7	0.44	0.35	μs
010	1.4	0.88	0.7	
011	2.8	1.75	1.4	
100	5.6	3.5	2.8	
101	11.2	7.0	5.6	
110	22.4	14.0	11.2	
111	44.8	28.0	22.4	

9. Input / Output port

This chapter describes port-related registers, their setting and circuits.

9.1 Registers

When the port registers are used, the following registers must be set.

All registers are 32-bits. The configurations are different depend on the number of port bits and assignation of the function.

"x" means the name of ports and "n" means the function number in the following description.

Register Name		Setting Value	
PxDATA	Data register	0 or 1	This register reads / writes port data.
PxCR	Output control register	0 : Output Disable 1 : Output Enable	This register controls output.
PxFRn	Function register n	0 : PORT 1 : Function	This register sets the function. The assigned function can be enabled by setting "1". This register exists for the each function assigned to the port. In case of having some function, only one function can be enabled.
PxOD	Open-drain control register	0 : CMOS 1 : Open-drain	This register controls programmable open-drain outputs. Programmable open-drain outputs are set with PxOD. When output data is "1", output buffer is disabled and becomes a pseudo-open-drain output.
PxPUP	Pull-up control register	0 : Pull-up Disable 1 : Pull-up Enable	This register controls programmable pull-ups.
PxPDN	Pull-down control register	0 : Pull-down Disable 1 : Pull-down Enable	This register controls programmable pull-downs.
PxIE	Input control register	0 : Input Disable 1 : Input Enable	This register controls inputs. Some time is required after enabling PxIE until external data is reflected in PxDATA.

9.1.1 Register list

For detail of the base address, refer to "Address lists of peripheral functions" of "Memory Map" chapter.

Register name	Address (Base+)	PORT A	PORT B	PORT C	PORT D	PORT E
Data register	0x0000	PADATA	PBDATA	PCDATA	PDDATA	PEDATA
Output control register	0x0004	PACR	PBCR	PCCR	PDCR	PECR
Function register 1	0x0008	PAFR1	PBFR1	−	PDFR1	PEFR1
Function register 2	0x000C	PAFR2	−	PCFR2	PDFR2	PEFR2
Function register 3	0x0010	−	−	PCFR3	PDFR3	−
Function register 4	0x0014	−	−	PCFR4	−	−
Function register 5	0x0018	−	−	PCFR5	−	−
Open-drain control register	0x0028	PAOD	PBOD	PCOD	PDOD	PEOD
Pull-up control register	0x002C	PAPUP	PBPUP	PCPUP	PDPUP	PEPUP
Pull-down control register	0x0030	PAPDN	PBPDN	PCPDN	PDPDN	PEPDN
Input control register	0x0038	PAIE	PBIE	PCIE	PDIE	PEIE

Register name	Address (Base+)	PORT F	PORT G	PORT H	PORT I	PORT J
Data register	0x0000	PFDATA	PGDATA	PHDATA	PIDATA	PJDATA
Output control register	0x0004	PFCR	PGCR	PHCR	PICR	PJCR
Function register 1	0x0008	PFFR1	−	PHFR1	−	PJFR1
Function register 2	0x000C	PFFR2	−	−	−	−
Function register 3	0x0010	−	−	−	−	−
Function register 4	0x0014	−	−	−	−	−
Open-drain control register	0x0028	PFOD	PGOD	PHOD	PIOD	PJOD
Pull-up control register	0x002C	PFPUP	PGPUP	PHPUP	PIPUP	PJPUP
Pull-down control register	0x0030	PFPDN	PGPDN	PHPDN	PIPDN	PJPDN
Input control register	0x0038	PFIE	PGIE	PHIE	PIIE	PJIE

Register name	Address (Base+)	PORT L	PORT M	PORT N	PORT P
Data register	0x0000	PLDATA	PMDATA	PNDATA	PPDATA
Output control register	0x0004	PLCR	PMCR	PNCR	PPCR
Function register 1	0x0008	PLFR1	−	−	−
Function register 2	0x000C	−	−	PNFR2	−
Function register 3	0x0010	−	−	−	−
Function register 4	0x0014	−	−	−	−
Open-drain control register	0x0028	PLOD	PMOD	PNOD	PPOD
Pull-up control register	0x002C	PLPUP	PMPUP	PNPUP	PPPUP
Pull-down control register	0x0030	PLPDN	PMPDN	PNPDN	PPPDN
Input control register	0x0038	PLIE	PMIE	PNIE	PPIE

Note: The address shown as "-" is not accessed.

9.1.2 Port function and setting list

The list of the function and setting register for each port is shown belows:

- "Table 9-1 PORT A Setting List"
- "Table 9-2 PORT B Setting List"
- "Table 9-3 PORT C Setting List"
- "Table 9-4 PORT D Setting List"
- "Table 9-5 PORT E Setting List"
- "Table 9-6 PORT F Setting List"
- "Table 9-7 PORT G Setting List"
- "Table 9-8 PORT H Setting List"
- "Table 9-9 PORT I Setting List"
- "Table 9-10 PORT J Setting List"
- "Table 9-11 PORT L Setting List"
- "Table 9-12 PORT M Setting List"
- "Table 9-13 PORT N Setting List"
- "Table 9-14 PORT P Setting List"

The cell of PxFRn shows the function register which must be set to select a function. If this register is set to "1", the corresponding function is enabled.

A bit in the cell filled with a hatch is read as "0" and the writing a data to this bit is invalid.

"0" or "1" in the table is shown the value which is set to the register. "0/1" is shown that the optional value can be set to the register.

Some function input / output pins are assigned to some ports. Only one port can be assigned by function registers.

9.1.2.1 PORT A

Table 9-1 PORT A Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PADATA	PACR	PAFRn	PAOD	PAPUP	PAPDN	PAIE
PA0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB0IN	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
	INT3	Input	FT4	0/1	0	PAFR2	0/1	0/1	0/1	1
PA1	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB0OUT	Output	FT1	0/1	1	PAFR1	0/1	0/1	0/1	0
	SCOUT	Output	FT1	0/1	1	PAFR2	0/1	0/1	0/1	0
PA2	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB1IN	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
	INT4	Input	FT4	0/1	0	PAFR2	0/1	0/1	0/1	1
PA3	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB1OUT	Output	FT1	0/1	1	PAFR1	0/1	0/1	0/1	0
	RXIN0	Input	FT1	0/1	0	PAFR2	0/1	0/1	0/1	1
PA4	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SCLK1	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
		Output		0/1	1	PAFR1	0/1	0/1	0/1	0
	$\overline{\text{CTS1}}$	Input	FT1	0/1	0	PAFR2	0/1	0/1	0/1	1
PA5	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TXD1	Output	FT1	0/1	1	PAFR1	0/1	0/1	0/1	0
	TB6OUT	Output	FT1	0/1	1	PAFR2	0/1	0/1	0/1	0
PA6	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	RXD1	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
	TB6IN	Input	FT1	0/1	0	PAFR2	0/1	0/1	0/1	1
PA7	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB4IN	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
	INT8	Input	FT4	0/1	0	PAFR2	0/1	0/1	0/1	1

9.1.2.2 PORT B

Table 9-2 PORT B Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PBDATA	PBCR	PBFRn	PBOD	PBPUP	PBPDN	PBIE
PB0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TRACECLK	Output	FT1	0/1	1	PBFR1	0/1	0/1	0/1	0
PB1	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TRACEDATA0	Output	FT1	0/1	1	PBFR1	0/1	0/1	0/1	0
PB2	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TRACEDATA1	Output	FT1	0/1	1	PBFR1	0/1	0/1	0/1	0
PB3	After reset(TMS/ SWDIO)			0	1	PBFR1	0	1	0	1
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TMS	Input	FT2	0/1	0	PBFR1	0/1	0/1	0/1	1
	SWDIO	I/O	FT2	0/1	1	PBFR1	0/1	0/1	0/1	1
PB4	After reset(TCK/ SWCLK)			0	0	PBFR1	0	0	1	1
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TCK	Input	FT2	0/1	0	PBFR1	0/1	0/1	0/1	1
	SWCLK	Input	FT2	0/1	0	PBFR1	0/1	0/1	0/1	1
PB5	After reset(TDO/ SWV)			0	1	PBFR1	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TDO	Output	FT2	0/1	1	PBFR1	0/1	0/1	0/1	0
	SWV	Output	FT2	0/1	1	PBFR1	0/1	0/1	0/1	0
PB6	After reset(TDI)			0	0	PBFR1	0	1	0	1
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TDI	Input	FT2	0/1	0	PBFR1	0/1	0/1	0/1	1
PB7	After reset($\overline{\text{TRST}}$)			0	0	PBFR1	0	1	0	1
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	$\overline{\text{TRST}}$	Output	FT2	0/1	1	PBFR1	0/1	0/1	0/1	0

9.1.2.3 PORT C

Table 9-3 PORT C Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PCDATA	PCCR	PCFRn	PCOD	PCPUP	PCPDN	PCIE
PC0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SP0DO	Output	FT3	0/1	1	PCFR2	0/1	0/1	0/1	0
	SO0	Output	FT1	0/1	1	PCFR3	0/1	0/1	0/1	0
	SDA0	I/O	FT1	0/1	1	PCFR3	0/1	0/1	0/1	1
PC1	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SP0DI	Input	FT1	0/1	0	PCFR2	0/1	0/1	0/1	1
	SI0	Input	FT1	0/1	0	PCFR3	0/1	0/1	0/1	1
	SCL0	I/O	FT1	0/1	1	PCFR3	0/1	0/1	0/1	1
PC2	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SP0CLK	Input	FT3	0/1	0	PCFR2	0/1	0/1	0/1	1
		Output		0/1	1	PCFR2	0/1	0/1	0/1	0
	SCK0	Input	FT1	0/1	0	PCFR3	0/1	0/1	0/1	1
		Output		0/1	1	PCFR3	0/1	0/1	0/1	0
PC3	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SP0FSS	Input	FT3	0/1	0	PCFR2	0/1	0/1	0/1	1
		Output		0/1	1	PCFR2	0/1	0/1	0/1	0
PC4	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
PC5	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	UT0TXD50B	Output	FT1	0/1	1	PCFR5	0/1	0/1	0/1	0
PC6	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	UT0TXD	Output	FT1	0/1	1	PCFR4	0/1	0/1	0/1	0
	UT0TXD50A	Output	FT1	0/1	1	PCFR5	0/1	0/1	0/1	0
PC7	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	UT0RXD	Input	FT1	0/1	0	PCFR4	0/1	0/1	0/1	1
	UT0RXD50	Input	FT1	0/1	0	PCFR5	0/1	0/1	0/1	1

9.1.2.4 PORT D

Table 9-4 PORT D Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PDDATA	PDCR	PDFRn	PDOD	PDPUP	PDPDN	PDIE
PD0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB5IN	Input	FT1	0/1	0	PDFR2	0/1	0/1	0/1	1
	INTC	Input	FT4	0/1	0	PDFR3	0/1	0/1	0/1	1
PD1	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB5OUT	Output	FT1	0/1	1	PDFR2	0/1	0/1	0/1	0
PD2	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	INTD	Input	FT4	0/1	0	PDFR3	0/1	0/1	0/1	1
PD3	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	INT9	Input	FT4	0/1	0	PDFR1	0/1	0/1	0/1	1
PD4	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SCLK2	Input	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
		Output		0/1	1	PDFR1	0/1	0/1	0/1	0
	$\overline{\text{CTS2}}$	Input	FT1	0/1	0	PDFR2	0/1	0/1	0/1	1
PD5	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TXD2	Output	FT1	0/1	1	PDFR1	0/1	0/1	0/1	0
PD6	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	RXD2	Input	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1

9.1.2.5 PORT E

Table 9-5 PORT E Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PEDATA	PECR	PEFRn	PEOD	PEPUP	PEPDN	PEIE
PE0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TXD0	Output	FT1	0/1	1	PEFR1	0/1	0/1	0/1	0
PE1	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	RXD0	Input	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
PE2	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SCLK0	Input	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
		Output		0/1	1	PEFR1	0/1	0/1	0/1	0
	$\overline{\text{CTS0}}$	Input	FT1	0/1	0	PEFR2	0/1	0/1	0/1	1
PE3	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB4OUT	Output	FT1	0/1	1	PEFR1	0/1	0/1	0/1	0
PE4	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB2IN	Input	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
	INT5	Input	FT4	0/1	0	PEFR2	0/1	0/1	0/1	1
PE5	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB2OUT	Output	FT1	0/1	1	PEFR1	0/1	0/1	0/1	0
PE6	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB3IN	Input	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
	INT6	Input	FT4	0/1	0	PEFR2	0/1	0/1	0/1	1
PE7	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB3OUT	Output	FT1	0/1	1	PEFR1	0/1	0/1	0/1	0
	INT7	Input	FT4	0/1	0	PEFR2	0/1	0/1	0/1	1

9.1.2.6 PORT F

Table 9-6 PORT F Setting List

PO RT	Reset status	Input/Output	PORT Type	Control registers						
				PFDATA	PFCR	PFFRn	PFOD	PFPUP	PFPDN	PFIE
PF0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB7IN	Input	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
PF1	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB7OUT	Output	FT1	0/1	1	PFFR1	0/1	0/1	0/1	0
	$\overline{\text{ALARM}}$	Output	FT1	0/1	1	PFFR2	0/1	0/1	0/1	0
PF2	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
PF3	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
PF4	After reset			0	0	0	0	0/1	0/1	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0

9.1.2.7 PORT G

Table 9-7 PORT G Setting List

PO RT	Reset status	Input/Output	PORT Type	Control registers						
				PGDATA	PGCR	PGFRn	PGOD	PGPUP	PGPDN	PGIE
PG0	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
PG1	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
PG2	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
PG3	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
PG4	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
PG5	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
PG6	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
PG7	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0

9.1.2.8 PORT H

Table 9-8 PORT H Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PHDATA	PHCR	PHFRn	PHOD	PHPUP	PHPDN	PHIE
PH0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	INT0	Input	FT4	0/1	0	PHFR1	0/1	0/1	0/1	1
	AIN0	Input	FT5	0/1	0		0/1	0	0	0
PH1	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	INT1	Input	FT4	0/1	0	PHFR1	0/1	0/1	0/1	1
	AIN1	Input	FT5	0/1	0		0/1	0/1	0/1	1
PH2	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	INT2	Input	FT4	0/1	0	PHFR1	0/1	0/1	0/1	1
	AIN2	Input	FT5	0/1	0		0/1	0	0	0
PH3	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	AIN3	Input	FT5	0/1	0		0/1	0	0	0
PH4	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	AIN4	Input	FT5	0/1	0		0/1	0	0	0
PH5	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	AIN5	Input	FT5	0/1	0		0/1	0	0	0
PH6	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	AIN6	Input	FT5	0/1	0		0/1	0	0	0
PH7	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	AIN7	Input	FT5	0/1	0		0/1	0	0	0

Note: To use the Port H as an analog input of the AD converter, disable input on PHIE and disable pull-up on PHPUP.

9.1.2.9 PORT I

Table 9-9 PORT I Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PIDATA	PICR	PIFRn	PIOD	PIPUP	PIPDN	PIIE
PI0	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AIN8	Input	FT5	0/1	0		0/1	0	0	0
PI1	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AIN9	Input	FT5	0/1	0		0/1	0	0	0

Note: To use the Port I as an analog input of the AD converter, disable input on PIIE and disable pull-up on PIPUP.

9.1.2.10 PORT J

Table 9-10 PORT J Setting List

PO RT	Reset status	Input/Output	PORT Type	Control registers						
				PJDATA	PJCR	PJFRn	PJOD	PJPUP	PJPDN	PJIE
PJ0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	AIN10	Input	FT5	0/1	0		0/1	0	0	0
PJ1	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	AIN11	Input	FT5	0/1	0		0/1	0	0	0
PJ2	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	AIN12	Input	FT5	0/1	0		0/1	0	0	0
PJ3	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	AIN13	Input	FT5	0/1	0		0/1	0	0	0
PJ4	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	AIN14	Input	FT5	0/1	0		0/1	0	0	0
PJ5	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	AIN15	Input	FT5	0/1	0		0/1	0	0	0
PJ6	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	INTA	Input	FT4	0/1	0	PJFR1	0/1	0/1	0/1	1
	AIN16	Input	FT5	0/1	0		0/1	0/1	0/1	1
PJ7	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	INTB	Input	FT4	0/1	0	PJFR1	0/1	0/1	0/1	1
	AIN17	Input	FT5	0/1	0		0/1	0/1	0/1	1

Note: To use the Port J as an analog input of the AD converter, disable input on PJIE and disable pull-up on PJPUP.

9.1.2.11 PORT L

Table 9-11 PORT L Setting List

PO RT	Reset status	Input/Output	PORT Type	Control registers						
				PLDATA	PLCR	PLFRn	PLOD	PLPUP	PLPDN	PLIE
PL0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0 (Note1)
PL2	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	INTF	Input	FT4	0/1	0	PLFR1	0/1	0/1	0/1	1

Note 2: PL0 works as a BOOT function. It is enabled to be input and pulled-up while RESET pin is "Low". At the rising edge of the reset signal, if PL0 is "High", the device enters single chip mode and boots from the on-chip flash memory. If PL0 is "Low", the device enters single BOOT mode and boots from the internal BOOT program.

9.1.2.12 PORT M

Table 9-12 PORT M Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PMDATA	PMCR	PMFRn	PMOD	PMPUP	PMPDN	PMIE
PM0	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	X1	Input	FT5	0/1	0		0	0	0	0
PM1	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	X2	Output	FT5	0/1	1		0	0	0	0

9.1.2.13 PORT N

Table 9-13 PORT N Setting List

PO RT	Reset status	Input/Output	PORT Type	Control registers						
				PNDATA	PNCR	PNFRn	PNOD	PNPUP	PNPDN	PNIE
PN0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
PN1	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
PN2	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
PN3	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
PN4	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
PN5	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
PN6	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
PN7	After reset			0	0	0	0	0	0	0
	Input Port	Input		0	0	0	0/1	0/1	0/1	1
	Output Port	Output		0	1	0	0/1	0/1	0/1	0
	INTE	Input	FT4	0/1	0	PNFR2	0/1	0/1	0/1	1

9.1.2.14 PORT P

Table 9-14 PORT P Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PPDATA	PPCR	PPFRn	PPOD	PPPUP	PPPDN	PPIE
PP0	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	XT1	Input	FT5	0/1	0		0	0	0	0
PP1	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	XT2	Output	FT5	0/1	1		0	0	0	0

9.2 Block Diagrams of Ports

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".

9.2.1 Type FT1

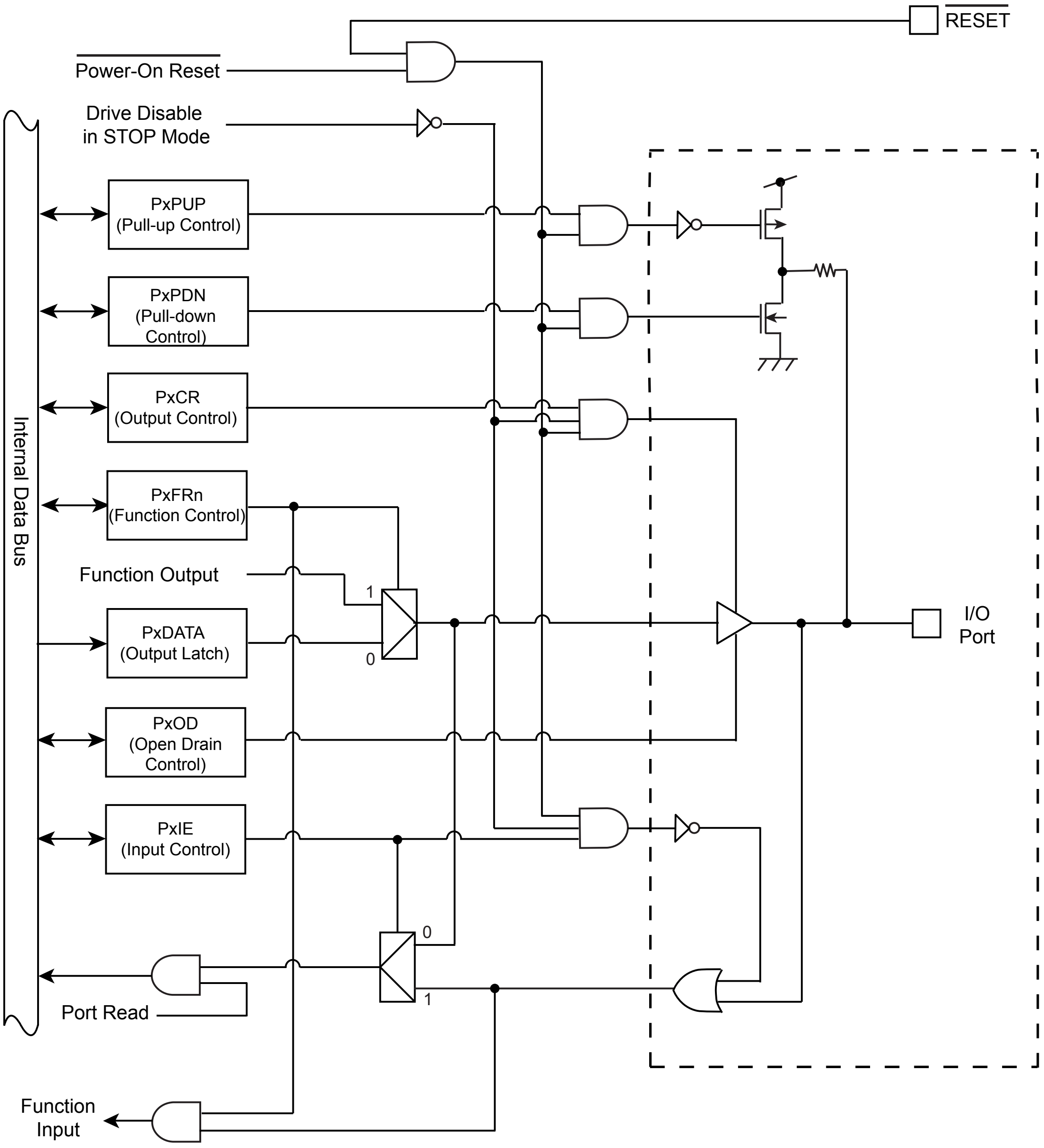


Figure 9-1 Port Type FT1

9.2.2 Type FT2

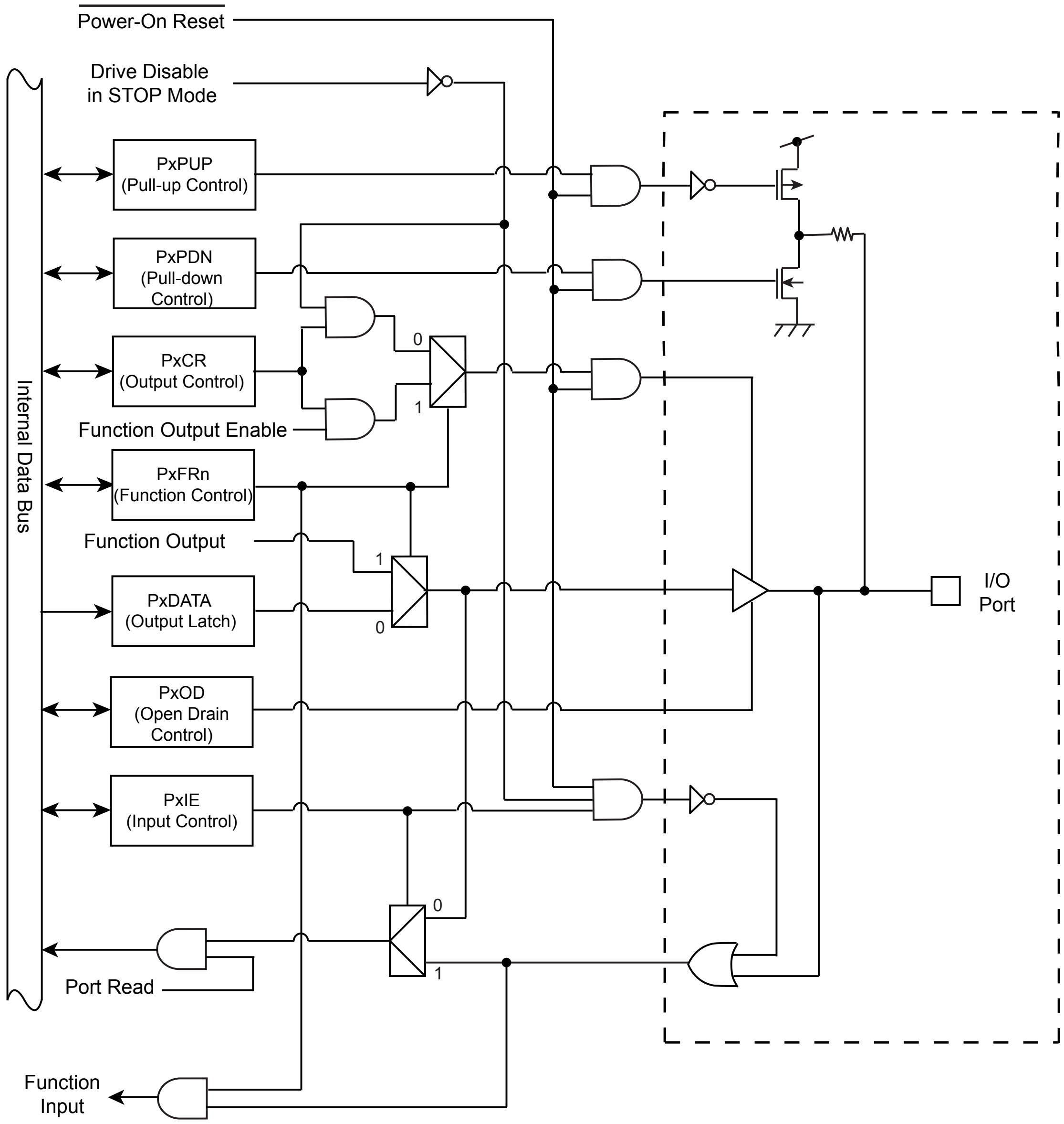


Figure 9-2 Port Type FT2

9.2.3 Type FT3

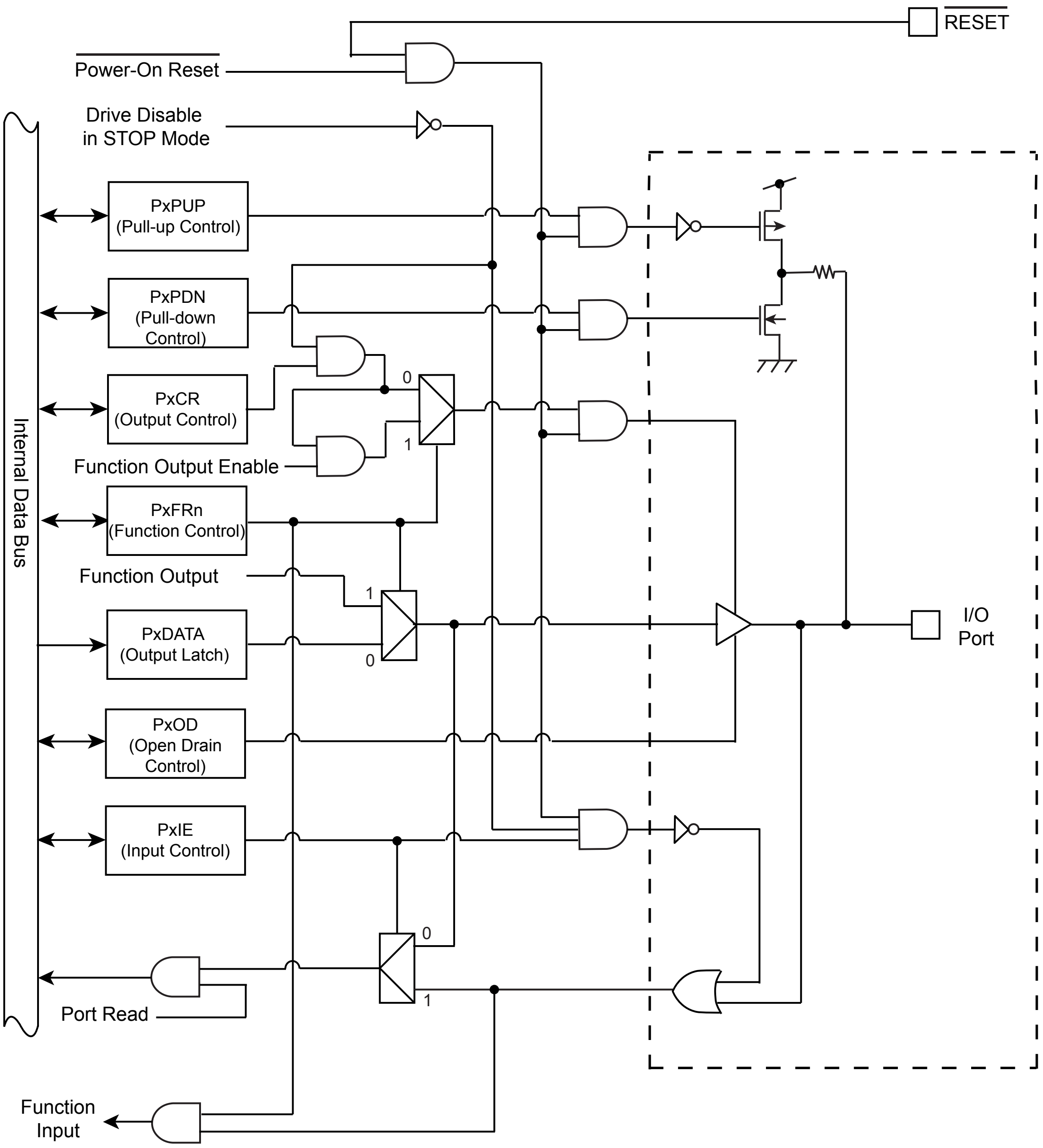


Figure 9-3 Port Type FT3

9.2.4 Type FT4

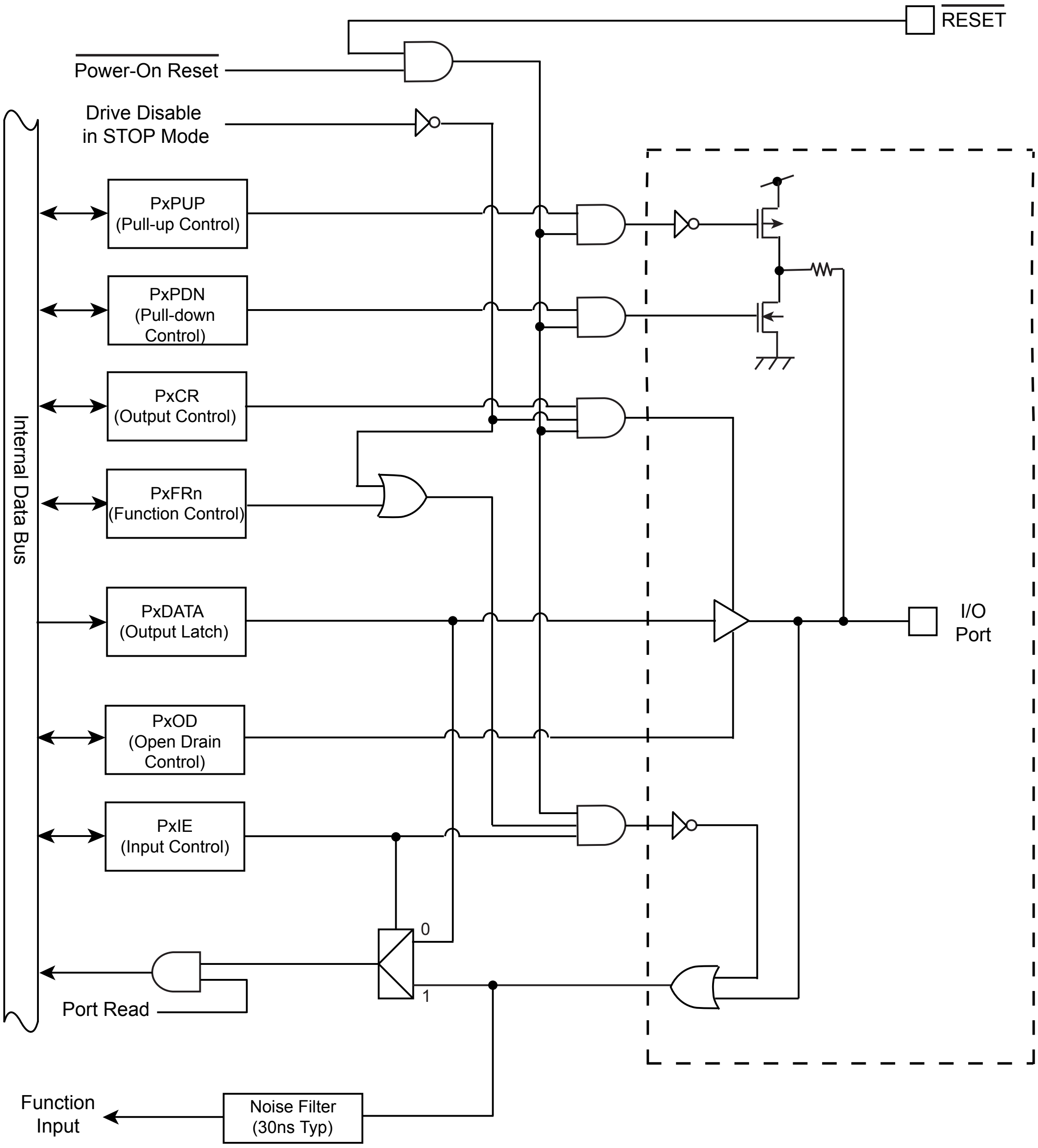


Figure 9-4 Port Type FT4

9.2.5 Type FT5

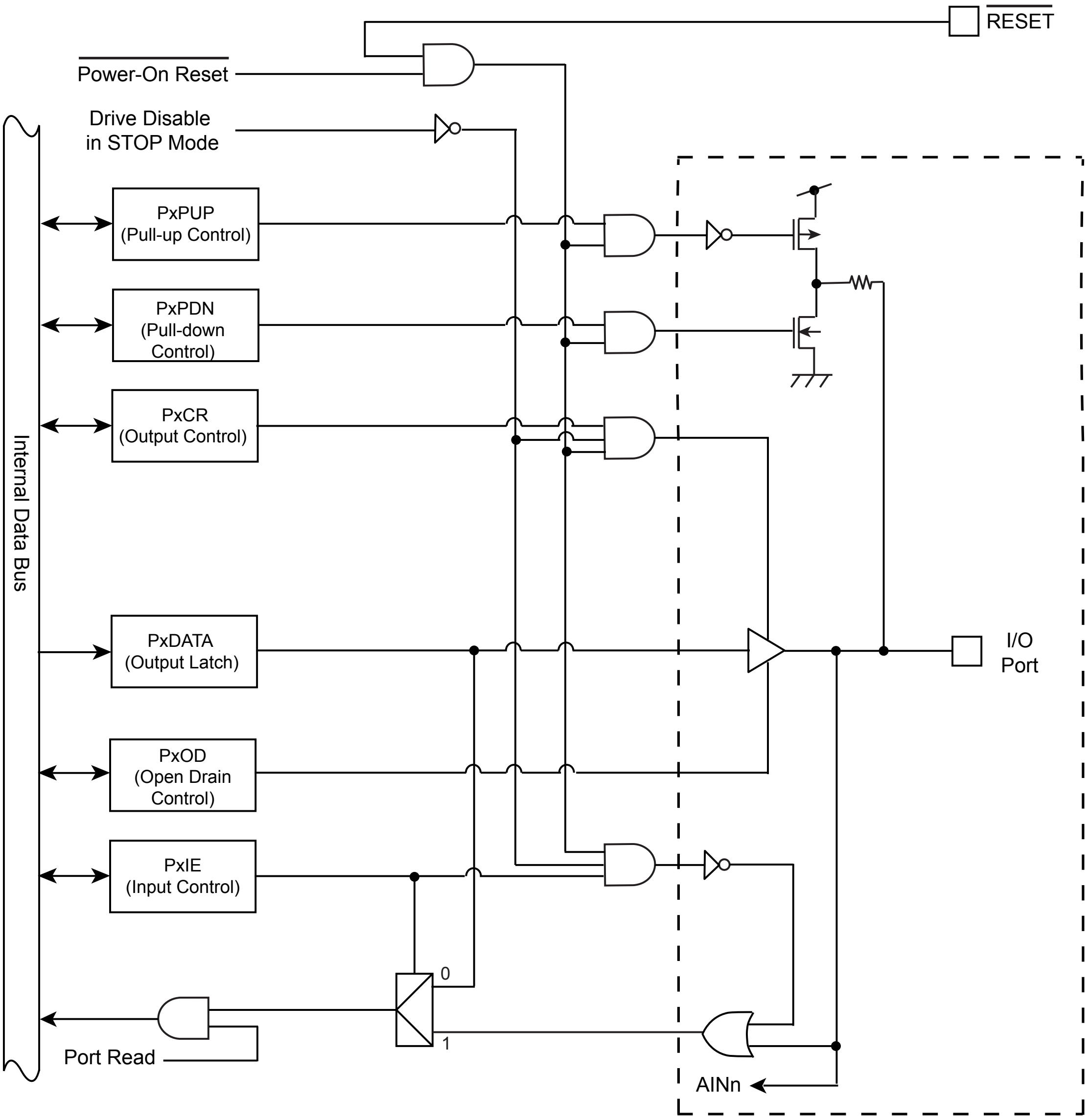


Figure 9-5 Port Type FT5

9.2.6 Type FT6

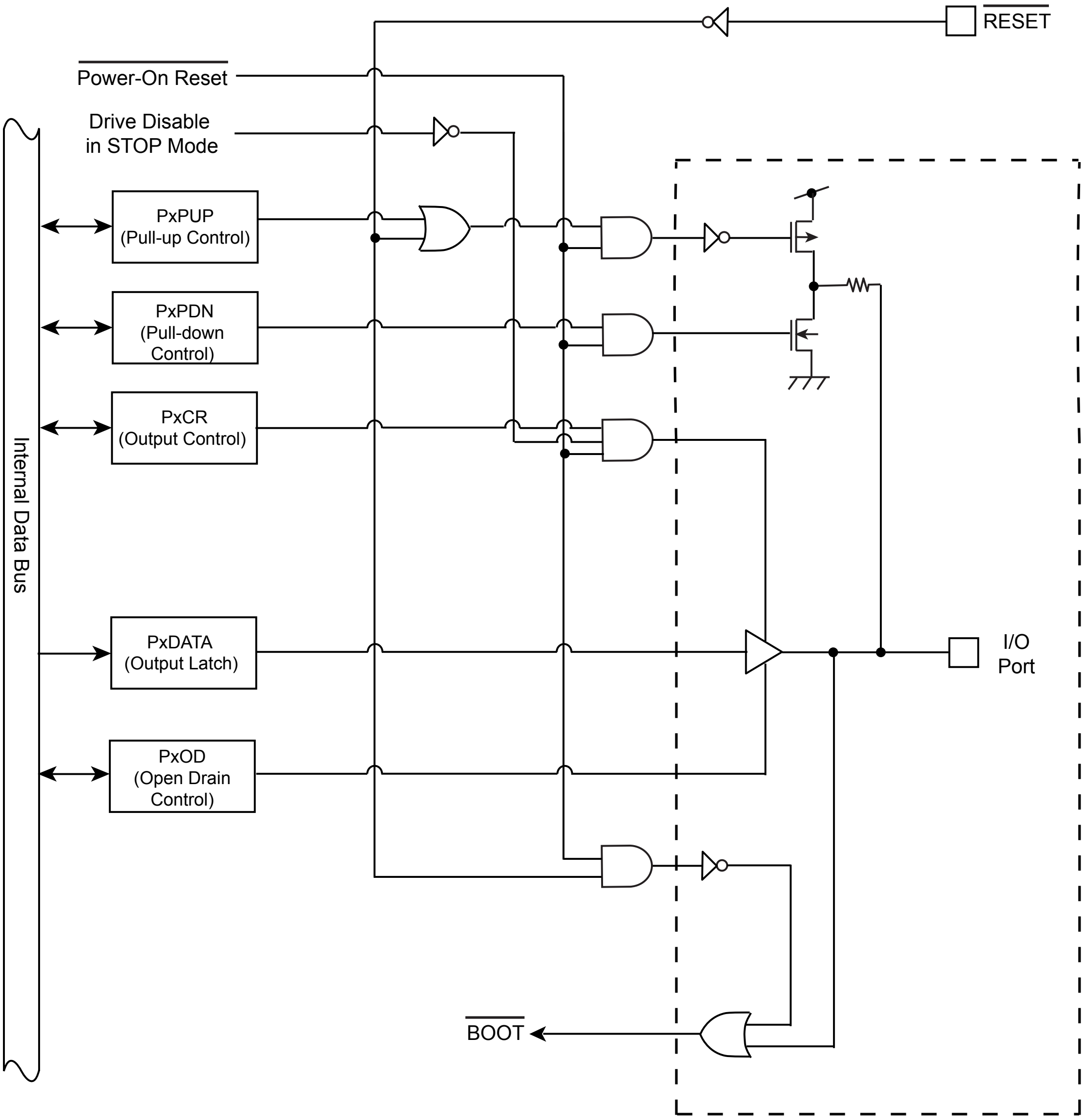


Figure 9-6 Port Type FT6

10. 16-bit Timer / Event Counters (TMRB)

10.1 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)
- External trigger programmable pulse generation mode (PPG)
- Timer synchronous mode

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

- One shot pulse output by an external trigger
- Frequency measurement
- Pulse width measurement

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

10.2 Differences in the Specifications

Each channel functions independently and the channels operate in the same way except for the differences in their specification as shown in Table 10-1.

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

1. The flip-flop output of TMRB 2, 5 and 7 can be used as the capture trigger of other channels.
   - TB2OUT → available for TMRB3 through TMRB5
   - TB5OUT → available for TMRB6 through TMRB7
   - TB7OUT → available for TMRB0 through TMRB2

2. The start trigger of the timer synchronous mode (with TBxRUN)
   - TMRB0 → can start TMRB1 through TMRB3 synchronously
   - TMRB4 → can start TMRB5 through TMRB7 synchronously

3. The start trigger of the timer prescaler synchronous mode (with TBxPRUN)
   - TMRB0 → can start TMRB1 through TMRB3 synchronously
   - TMRB4 → can start TMRB5 through TMRB7 synchronously

For details of each product, refer to Chapter "Product Information".

Table 10-1 Differences in the Specifications of TMRB Modules

Specification	External pins		Trigger function between timers		Interrupt		Internal Connects	
Channel	External clock / capture trigger input pins	Timer Flip-Flop output pins	Capture trigger	Synchronous start trigger channel	Capture interrupt	TMRB interrupt	Start AD conversion	Timer Flip-Flop Connect with SIO/UART, RMC (TXTRG : Transfer clock)
TMRB0	TB0IN	TB0OUT	TB7OUT	−	INTCAP00 INTCAP01	INTTB00 INTTB01	−	−
TMRB1	TB1IN	TB1OUT	TB7OUT	TB0PRUN TB0RUN	INTCAP10 INTCAP11	INTTB10 INTTB11	−	RMC
TMRB2	TB2IN	TB2OUT	TB7OUT	TB0PRUN TB0RUN	INTCAP20 INTCAP21	INTTB20 INTTB21	−	−
TMRB3	TB3IN	TB3OUT	TB2OUT	TB0PRUN TB0RUN	INTCAP30 INTCAP31	INTTB30 INTTB31	−	−
TMRB4	TB4IN	TB4OUT	TB2OUT	−	INTCAP40 INTCAP41	INTTB40 INTTB41	−	SIO0 SIO1
TMRB5	TB5IN	TB5OUT	TB2OUT	TB4PRUN TB4RUN	INTCAP50 INTCAP51	INTTB50 INTTB51	INTTB51	−
TMRB6	TB6IN	TB6OUT	TB5OUT	TB4PRUN TB4RUN	INTCAP60 INTCAP61	INTTB60 INTTB61	−	−
TMRB7	TB7IN	TB7OUT	TB5OUT	TB4PRUN TB4RUN	INTCAP70 INTCAP71	INTTB70 INTTB71	−	SIO2

10.3 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.

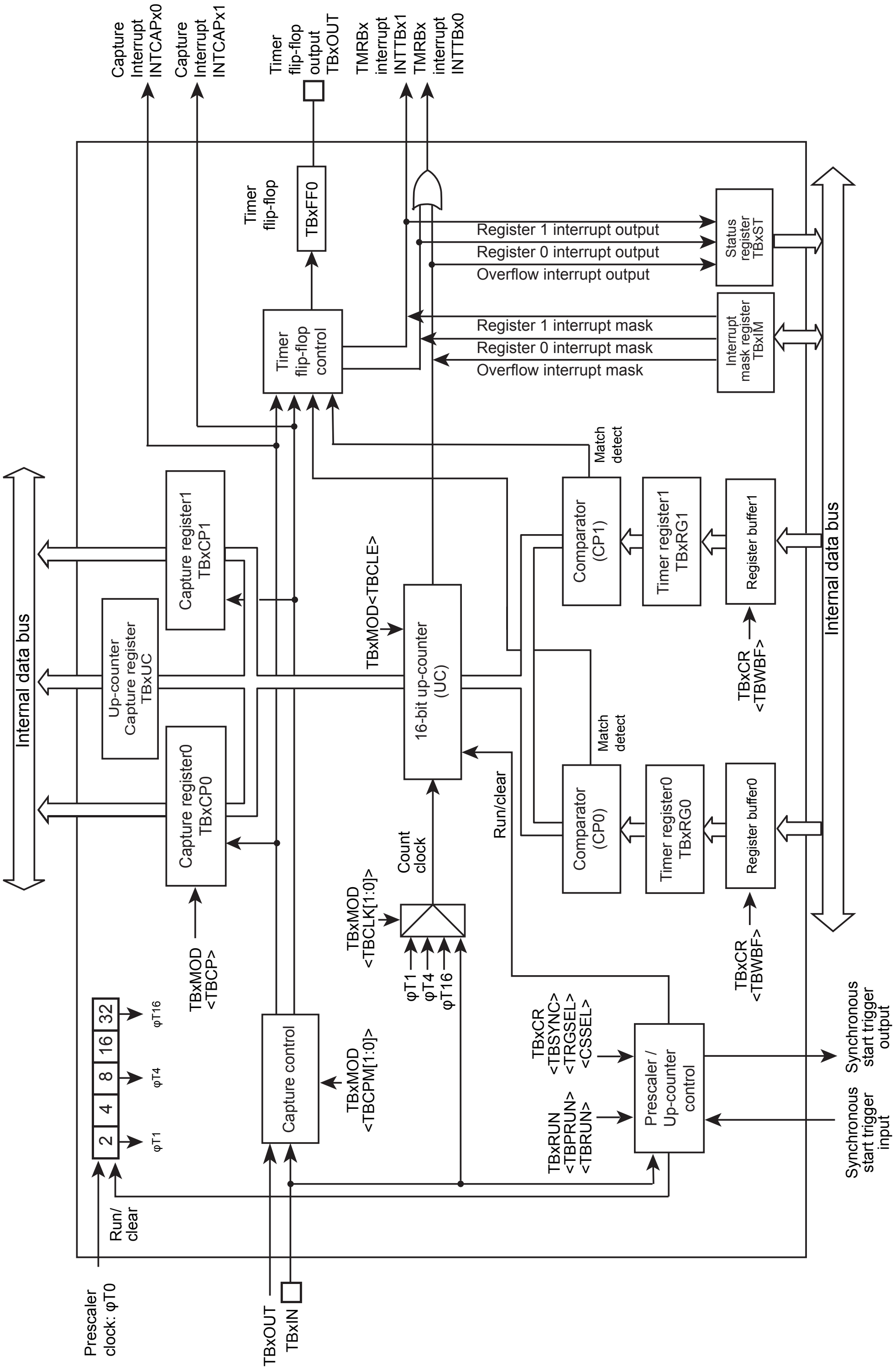


Figure 10-1 TMRBx Block Diagram

10.4 Registers

10.4.1 Register list according to channel

The followings are the TMRB control registers and addresses.

For detail of the base address, refer to "Address lists of peripheral functions" of "Memory Map" chapter.

Register name		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

10.4.2 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	TBHALT	-	-	-	-	-	-
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: Disable 1: Enable 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	TBHALT	R/W	Clock operation during debug HALT 0: Operation 1: Stop When using debug tool, in case of operation mode transition to HALT mode, TMRB clock is operated or stopped by this bit.
5-0	−	R	Read as "0".

10.4.3 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

Note: When the counter is stopped (<TBRUN>=0) and TBxUC<TBUC[15:0]> is read, the value which was captured when the counter was operated is read.

10.4.4 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	-	TRGSEL	CSSEL
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: Disable 1: Enable
6	−	R/W	Write as "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	−	R	Read as "0"
1	TRGSEL	R/W	Select external trigger 0: Rising edge 1: Falling edge Select the edge of the external trigger (Signal to TBxIN pin)
0	CSSEL	R/W	Select count start 0: Soft start 1: External trigger

10.4.5 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	-	TBRSWR	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	TBRSWR	R/W	Controls the timing to write to timer registers 0 and 1 when double buffering is enabled. 0: Timer registers 0 and 1 can be written separately, even in case writing preparation is ready for only one resister. 1: In case both resisters are not ready to be written, Timer registers 0 and 1 can not be written
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 (Note2) 00: Disable 01: TBxIN↑ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN pin input. 10: TBxIN↑ TBxIN↓ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN pin input. Takes count values into capture register 1 (TBxCP1) upon falling of TBxIN pin input. 11: TBxOUT↑ TBxOUT↓ Takes count values into capture register 0 (TBnCP0) upon rising of 16-bit timer match output (TBxOUT) and into capture register 1 (TBnCP1) upon falling of TBxOUT.
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: TBxIN pin input 01: φT1 10: φT4 11: φT16

Note 1: Do not modify TBxMOD during operating TMRBx.

Note 2: Specifications are different depending on the product. For details, refer to "Product Information".

10.4.6 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".

10.4.7 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 match 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.

10.4.8 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.

Note: Even if TBxIM setting is done, TBxST is set.

10.4.9 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.

Note: When the counter is operated and TBxUC is read, the value of the up counter is captured and read.

10.4.10 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.

10.4.11 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.

10.4.12 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	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBCP0							
After reset	0	0	0	0	0	0	0	0

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.

10.4.13 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	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBCP1							
After reset	0	0	0	0	0	0	0	0

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.

10.5 Description of Operations for Each Circuit

The channels operate in the same way, except for the differences in their specifications as shown in Table 10-1.

10.5.1 Prescaler

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

The prescaler input clock φT0 is fs, 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[1:0]> 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. Below tables show prescaler output clock resolutions.

Table 10-2 Prescaler Output Clock Resolutions (fc = 40MHz)

Select peripheral clock CGSYSCR <FPSEL[1:0]>	Select gear clock CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
00 (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)
	111 (fc/16)	000 (fperiph/1)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)
		001 (fperiph/2)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)
		010 (fperiph/4)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)
		011 (fperiph/8)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
		100 (fperiph/16)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)	fc/2^{13} (204.8 μs)
		101 (fperiph/32)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)	fc/2^{14} (409.6 μs)

Table 10-2 Prescaler Output Clock Resolutions (fc = 40MHz)

Select peripheral clock CGSYSCR <FPSEL[1:0]>	Select gear clock CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
01 (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)
	111 (fc/16)	000 (fperiph/1)	−	−	fc/2^{5} (0.8 μs)
		001 (fperiph/2)	−	−	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^{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 10-3 Prescaler Output Clock Resolutions (fs = 32.768kHz, <SYSCK> = "1")

Select peripheral clock CGSYSCR <FPSEL[1:0]>	Select gear clock CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
10 11 (fs)	−	−	fs/2^{1} (61.0 μs)	fs/2^{3} (244.1 μs)	fs/2^{5} (976.6 μs)

10.5.2 Up-counter (UC)

UC is a 16-bit binary counter.

- Source clock

  UC source clock, specified by TBxMOD<TBCLK[2:0]>, can be selected from either three types - φT1, φT4 and φT16 - of prescaler output clock or the external clock of the TBxIN pin.

- Counter 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.

10.5.3 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.

10.5.4 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>.

10.5.5 Capture registers (TBxCP0, TBxCP1)

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

10.5.6 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.

10.5.7 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.

10.5.8 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.

10.5.9 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.

10.6 Description of Operations for Each Mode

10.6.1 16-bit interval Timer Mode

In the case of generating constant period interrupt, set the interval time to the Timer register (TBxRG0) to generate the INTTBx0 interrupt.Same as TBxRG0, set the interval time to the Timer register (TBxRG1) to generate the INTTBx1 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

10.6.2 16-bit Event Counter Mode

It is possible to make it the event counter by using an input clock as an external clock (TBxIN pin input).

The up-counter counts up on the rising edge of TBxIN 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 TMRBx.
Set PORT registers.										Allocates corresponding port to TBxIN.
TBxFFCR	←	X	X	0	0	0	0	1	1	Disable to TBxFF0 reverse trigger.
TBxMOD	←	X	1	0	0	0	0	0	0	Changes to TBxIN as an input clock.
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.
TBxMOD	←	X	0	0	0	0	0	0	0	Software capture is done.

Note: X; Don’t care −; No change

10.6.3 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

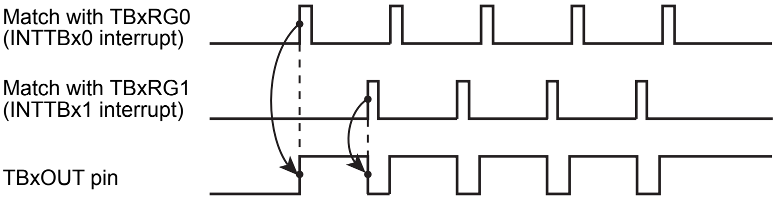


Figure 10-2 Example of Output of Programmable Pulse Generation (PPG)

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.

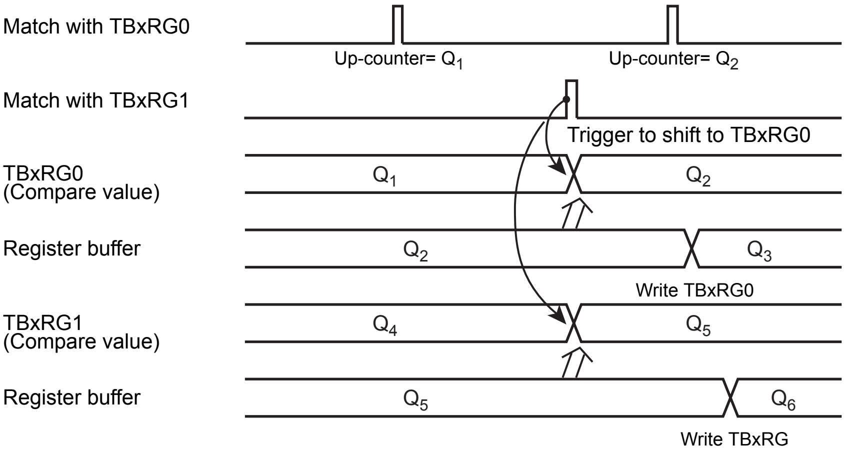


Figure 10-3 Register Buffer Operation

The block diagram of this mode is shown below.

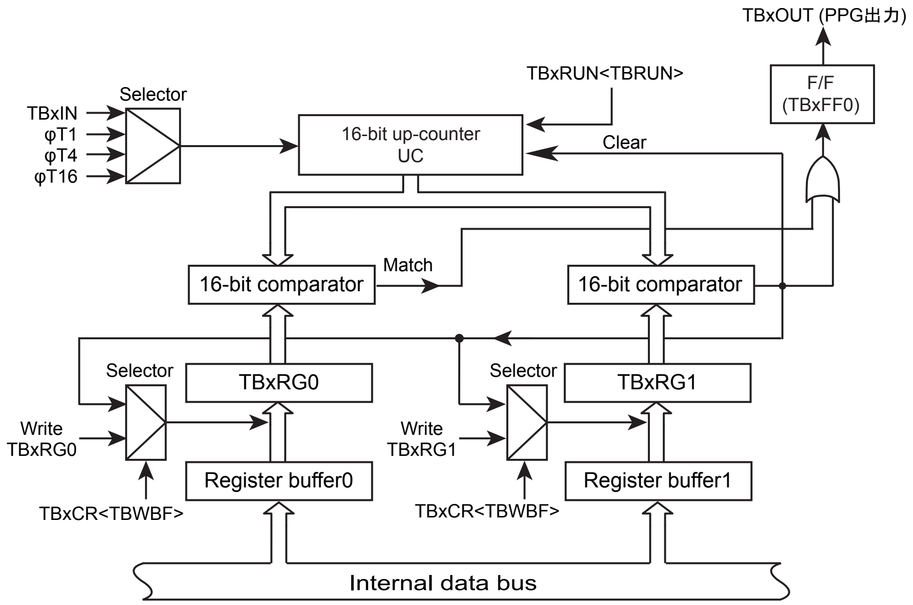


Figure 10-4 Block Diagram of 16-bit PPG Mode

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	0	0	Disable 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)				
Set PORT registers.										Allocates corresponding port to TBxOUT.
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note 1: m ; corresponding bit of port

Note 2: X; Don’t care

−; No change

10.6.4 External Trigger PPG (Programmable Square Wave) Output Mode

An external trigger count start mode provides one-shot pulse output with a short delay.

1. Set the 16-bit up-counter to count-up on the rising edge of TBxIN pin (TBxCR<TRGSEL, CSSEL>= "01") while the up-counter (UC) is stopping (TBxRUN<TBRUN> = "0"). Set a delay time (d) to the timer register (TBxRG0). In the timer register TBxRG1, set a value (d+P) which is added one-shot pulse width (p) to the delay time of TBxRG0.

2. To trigger enable to reverse the timer flip-flop, first set "11" to the timer flip-flop control register (TBxFFCR<TBE1T1,TBE0T1>). Then the timer flip-flop(TBxFF0) is reversed, when UC matches TBxRG0/TBxRG1.

3. Set "1" to TBxRUN<TBRUN> to enable the count-up on the rising edge of external trigger.

4. After the generation of one-shot pulse on the rising edge of TBxIN pin, set a reverse of the timer flip-flop (TBxFF0) to disable with the interrupt of INTTBx1 or set "0" to TBxRUN<TBRUN> to clear the value and stop the 16-bit up-counter operation.

The symbols (d) and (p) in the above description corresponds to the symbol d and p in the following figure.

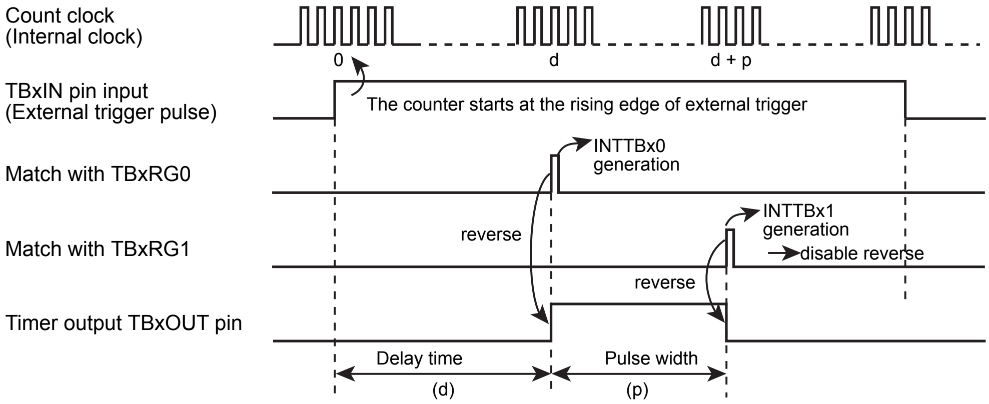


Figure 10-5 One-shot pulse generation using an external trigger count start (with a delay)

10.6.5 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 pairs of 4-channel TMRB. If one channel starts, remaining 3 channels can be start synchronously. In the TMPM381/383, 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.

Note: Except timer synchronous mode, TBxCR<TBSYNC> should be cleared to "0". In case of setting timer synchronous mode, other channels are waiting start until they are started by TMRB0 or TMRB4.

10.7 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

10.7.1 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 UC 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 10-6 One-shot Pulse Output (With Delay)".

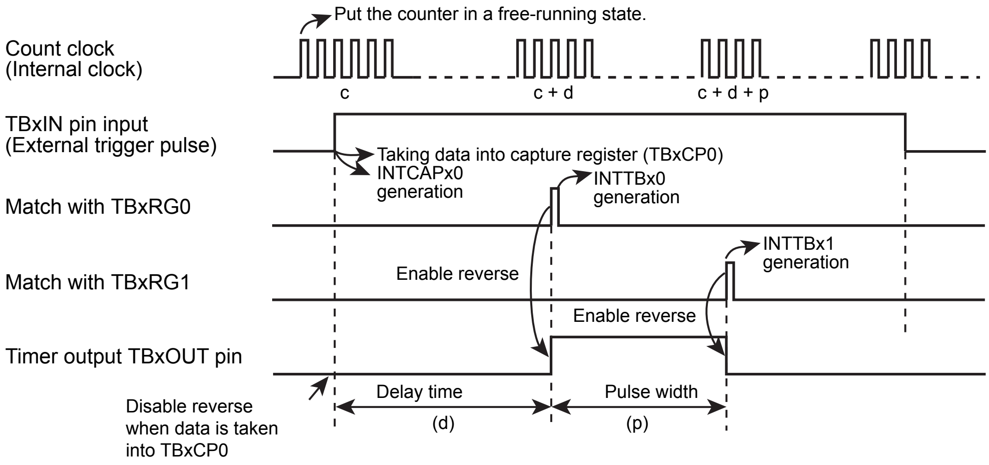


Figure 10-6 One-shot Pulse Output (With Delay)

The followings show the settings in the case that 2 ms width one-shot pulse is output after 3ms by triggering TBxIN input at the rising edge. (φT1 is selected for counting.)

		7	6	5	4	3	2	1	0	
[Main processing] Capture setting by TBxIN0										
Set PORT registers.										Allocates corresponding port to TBxIN0.
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	1	0	1	0	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.
Set PORT registers.										Allocates corresponding port to TBxOUT.
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 UC consistent with TBxRG0 and 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.

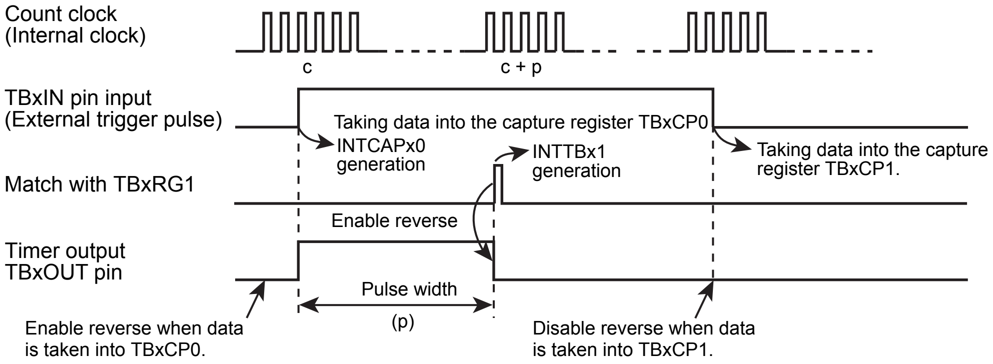


Figure 10-7 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

10.7.2 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 TMRB0 and TMRB7. TB7OUT of the 16-bit timer TMRB7 is used to specify the measurement time.

TMRB0 count clock selects TB0IN input and performs count operation by using external clock input. If TB0MOD<TBCPM[1:0]> is set "11", TMRB0 count clock takes the counter value into the TB0CP0 at the rising edge of TB7OUT and takes the counter value into TB0CP1 at the falling edge of TB7OUT.

This setting allows a count value of the 16-bit up-counter UC to be taken into the capture register (TB0CP0) upon rising of a timer flip-flop output (TB7OUT) of the 16-bit timer (TMRB7), and an UC counter value to be taken into the capture register (TB0CP1) upon falling of TB7OUT of the 16-bit timer (TMRB7).

A frequency is then obtained from the difference between TB0CP0 and TB0CP1 based on the measurement, by generating the INTTB7 16-bit timer interrupt.

For example, if the difference between TB0CP0 and TB0CP1 is 100 and the level width setting value of TB0OUT is 0.5 s, the frequency is 200 Hz (100 ÷ 0.5 s = 200 Hz).

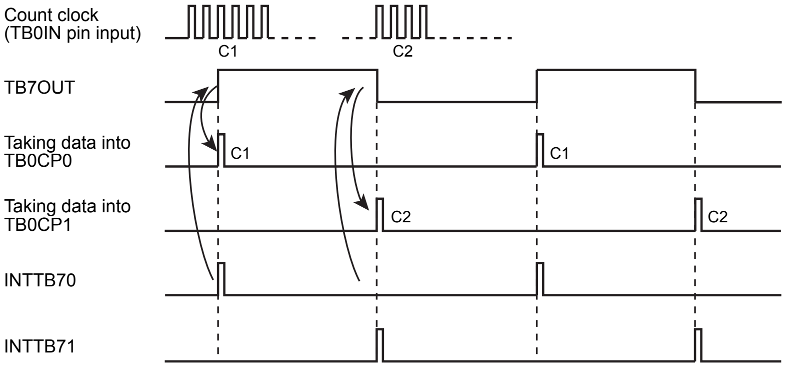


Figure 10-8 Frequency Measurement

10.7.3 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 TBxIN 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 TBxIN 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 width exceeding the UC maximum count time which is dependent 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 10-9 Pulse Width Measurement" and this difference is multiplied by the cycle of the prescaler output clock to obtain the "Low" level width.

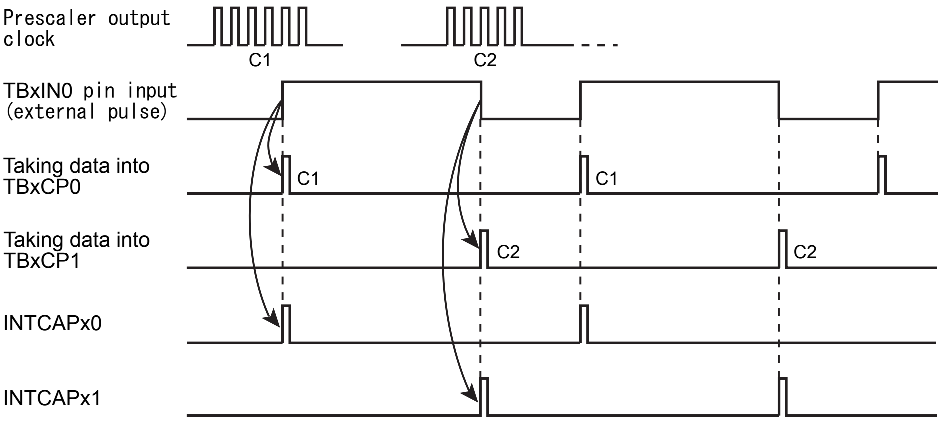


Figure 10-9 Pulse Width Measurement

11. Universal Asynchronous Receiver-Transmitter Circuit (UART)

11.1 Outline

The universal asynchronous receiver-transmitter circuit provides the following features:

- Transmit/receive data format
  - Data length: selectable from 5, 6, 7, or 8 bits
  - With/without a parity bit
  - STOP bit length: selectable from 1 bit or 2 bits
- FIFO
  - Transmission: 8-bit width/ 32-deep, reception: 12-bit width/ 32-deep
  - Enable/disable decision possible
- Interrupt function
  - Multiple interrupt event outputs
  - Each interrupt can be enabled/disabled.
- Baud-rate generator

  Transmit and receive common clock can be generated from fsys.
- 50% duty mode
  - Corresponding to the communication of the 50% duty signal.
  - "0" data can be output by distributing to the two terminals alternately.

Note:"x" of pin names and registers indicates the channel number.

11.2 Structure

Figure 11-1 shows a block diagram of UART.

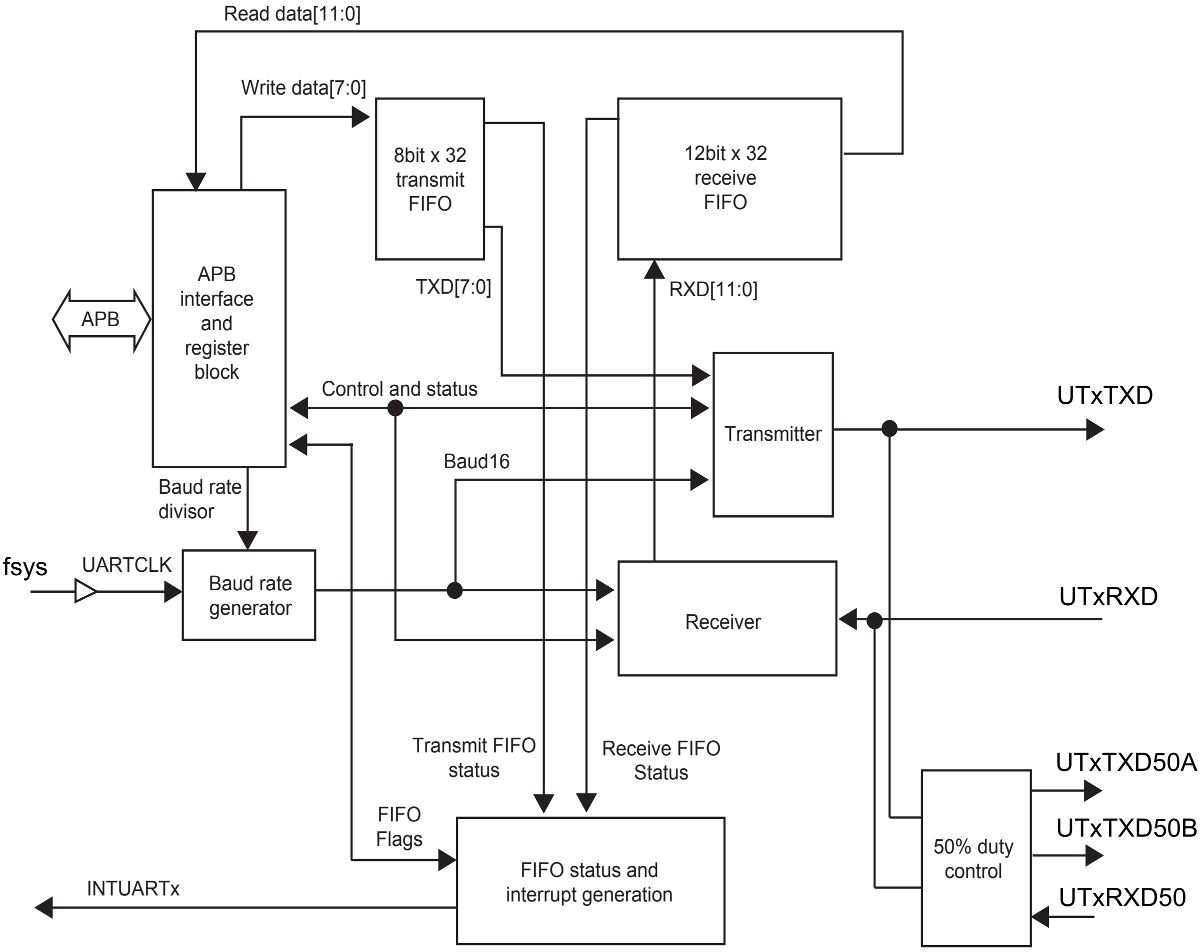


Figure 11-1 Block diagram of UART

11.3 Registers

11.3.1 List of Registers

The following table lists the control registers and their addresses.

For base addresses, refer to "A list of peripheral function base addresses" in the chapter on "Memory Map."

Register name		Address (Base+)
Data register	UARTxDR	0x0000
Receive status register	UARTxRSR	0x0004
Error clear register	UARTxECR	0x0004
Reserved	-	0x0008 to 0x0017
Flag register	UARTxFR	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Integer baud rate register	UARTxIBRD	0x0024
Fractional baud rate register	UARTxFBRD	0x0028
Line control register	UARTxLCR_H	0x002C
Control register	UARTxCR	0x0030
interrupt FIFO level select register	UARTxIFLS	0x0034
Interrupt mask set/clear register	UARTxIMSC	0x0038
Raw interrupt status register	UARTxRIS	0x003C
Masked interrupt status register	UARTxMIS	0x0040
Interrupt clear register	UARTxICR	0x0044
Reserved	-	0x0048
50% duty control register	UARTxHCCR	0x0050
Reserved	-	0x0054 to 0x0FFF

Note 1: When the control registers are re-set, disable the UART.

Note 2: If the UART operation is disabled during the reception or transmission, the UART will stop after on-going transmission is complete.

Note 3: Do not access the addresses described as "Reserved".

11.3.2 UARTxDR (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	-	-	-	-	OE	BE	PE	FE
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	DATA							
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-12	−	R	Read as "0".
11	OE	R	Overrun error 0: No error 1: Error If the FIFO has been full when receiving data, this bit is set to "1". When the FIFO has empty space and new data can be written to FIFO, this bit is cleared to "0".
10	BE	R	Break error 0: No error 1: Error If break condition (the UTxRXD input is kept to "Low" longer than accumulated time of a start bit, data bit, parity bit, and stop bit) is detected, this bit is set to "1". If the FIFO is enabled, this error is stored at the top of the FIFO. If a break error occurs, "0" is stored in the FIFO as data. Next data reception is enabled after the UTxRXD input is "1" (marking state) and the start bit is received.
9	PE	R	Parity error 0: No error 1: Error When this bit is set to "1", this indicates that a received data parity does not match the parity programmed with UARTxLCR_H<EPS> and <SPS>. If the FIFO is enabled, this error is stored at the top of FIFO.
8	FE	R	Framing error 0: No error 1: Error When this bit is set to "1", this indicates that received data does not include a valid stop bit. (A valid stop bit length is "1".) If the FIFO is enabled, this error is stored at the top of FIFO.
7-0	DATA[7:0]	R/W	[Read] Receive data [Write] Transmit data

Note: Error status can be identified by reading UARTxRSR as well.

11.3.3 UARTxRSR (Receive Status Register)

Both UARTxRSR and UARTxECR registers are mapped on the same address.

	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	-	-	-	-	OE	BE	PE	FE
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-4	−	R	Read as "0".
3	OE	R	Overrun error 0: No error 1: Error When data is received, if FIFO has already been full, this bit is set to "1". This bit is cleared to "0" by writing data to UARTxECR. If FIFO is full, further data cannot be written. Thus, the content of FIFO is valid and only the content of shift register is overwritten. CPU must be read data in order to empty FIFO.
2	BE	R	Break error 0: No error 1: Error If break condition (UTxRXD input is held "Low" for longer than a full-word transmission time defined as start, data, parity, and stop bits) is detected, this bit is set to "1". This bit is cleared to "0" by writing data to UARTxECR. If FIFO is enabled, this error is input to the top of the FIFO. If a break error occurs, "0" is input to FIFO as data. In the next data reception, UTxRXD input is set to "1" (marking status), this bit is enabled after a start bit is received.
1	PE	R	Parity error 0: No error 1: Error When this bit is "1", this indicates that the parity of received data does not match the parity set in UARTxLCR_H<EPS> and <SPS>. This bit is cleared to "0" by writing data to UARTxECR. If the FIFO is enabled, this error is input to the top of the FIFO.
0	FE	R	Framing error 0: No error 1: Error When this bit is set to "1", this indicates that a valid stop bit is not included in the received data. (A valid stop bit length is "1".) This bit is cleared to "0" by writing data to UARTxECR. If the FIFO is enabled, this error is input to the top of FIFO.

Note 1: An overrun error is immediately set when the error occurs.

Note 2: UARTxRSR is updated when data is read from UARTxDR. Therefore, received data must be read from UARTxDR before error status is read from UARTxRSR. This read sequence cannot be reversed. In addition, error status can be read by reading UARTxDR.

11.3.4 UARTxECR (Error 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	-	-	-	-	OE	BE	PE	FE
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-4	−	R	Read as "0".
3	OE	W	When data is written to UARTxECR, each framing, parity, break, and overrun errors are cleared. This clearing is executed regard less of the data value. The address of this register is the same as those of the UARTxSR register.
2	BE	W	
1	PE	W	
0	FE	W	

11.3.5 UARTxFR (UART 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	TXFE	RXFF	TXFF	RXFE	BUSY	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-9	−	R	Read as an undefined value.
8	-	R	Read as an undefined value.
7	TXFE	R	When UARTxLCR_H<FEN> = "1" 0: The transmit FIFO is not empty. 1: The transmit FIFO is empty.
			When UARTxLCR_H<FEN> = "0" 0: The transmit hold register is not empty. 1: The transmit hold register is empty.
6	RXFF	R	When UARTxLCR_H<FEN> = "1" 0: The receive FIFO is not full. 1: The receive FIFO is full.
			When UARTxLCR_H<FEN> = "0" 0: The receive hold register is not full. 1: The receive hold register is full.
5	TXFF	R	When UARTxLCR_H<FEN> = "1" 0: The transmit FIFO is not full. 1: The transmit FIFO is full.
			When UARTxLCR_H<FEN> = "0" 0: The transmit hold register is not full. 1: The transmit hold register is full.
4	RXFE	R	When UARTxLCR_H<FEN> = "1" 0: The receive FIFO is not empty. 1: The receive FIFO is empty.
			When UARTxLCR_H<FEN> = "0" 0: The receive hold register is not empty. 1: The receive hold register is empty.
3	BUSY	R	UART busy 0: UART transmission is stopping. 1: UART transmission is ongoing. This bit is set to "1" when the transmit FIFO is not empty regardless of whether UART operation is enabled or not.
2	-	R	Read as an undefined value.
1	-	R	Read as an undefined value.
0	-	R	Read as an undefined value.

Note:<TXFE> does not indicate the status of shift register.

11.3.6 UARTxIBRD (UART Integer Baud-rate 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	BAUDDIVINT							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	BAUDDIVINT							
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-16	−	R	Read as an undefined value.
15-0	BAUDDIVINT [15:0]	R/W	Integer baud-rate divisor (0x0001 to 0xFFFF) The integer part of a baud-rate divisor value

Note 1: The value written to UARTxIBRD will be valid upon currently ongoing transmission or reception is complete.
Note 2: The value written to UARTxIBRD will be valid when data is written to UARTxLCR_H.
Note 3: Set <BAUDDIVINT[15:0]> before UARTxCR<UARTEN> is set to "1".
Note 4: "0x0000" cannot be set.

11.3.7 UARTxFBRD (UART Fractional Baud-rate 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	-	-	BAUDDIVFRAC					
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-6	−	R	Read as "0".
5 -0	BAUD DIVFRAC [5:0]	R/W	Fractional baud-rate divisor (0x01 to 0x3F) The fractional part of a baud-rate divisor value.

Note 1: The value written to UARTxFBRD will be valid upon currently ongoing transmission or reception is complete.

Note 2: The value written to UARTxFBRD will be valid when data is written to UARTxLCR_H.

Note 3: Set <BAUDDIVFRAC[5:0]> before "1" is set to UARTxCR>UARTEN>.

Note 4: The minimum value of a baud-rate divisor is 1 and the maximum value is 65535. Therefore, the integer part of a baud-rate divisor cannot be set to 0. The fractional part of a baud-rate divisor must be set 0 if the integer part of a baud-rate divisor is 65535.

11.3.8 UARTxLCR_H (UART Line 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	SPS	WLEN		FEN	STP2	EPS	PEN	BRK
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-8	−	R	Read as "0".
7	SPS	R/W	Selects a stick parity 0: A stick parity is disabled. 1: When <EPS> = "0", "1" is sent/received as a parity bit. When <EPS> = "1", "0" is sent/received as a parity bit. <SPS> has no meaning when <PEN> is set to "0" and the parity check and generation are disabled. For details of the truth table of <SPS>, <EPS>, and <PEN>, refer to Table 11-1.
6-5	WLEN[1:0]	R/W	Word length 00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits These bits indicate the number of data bits transmitted/received in the frame.
4	FEN	R/W	Enables/disables the FIFO. 0: The FIFO is disabled (The FIFO becomes a 1-deep hold register.) 1: The FIFO is enabled
3	STP2	R/W	Selects a transmission stop bit length 0: 1 bit 1: 2 bits In reception, a 2-bit length stop bit is not checked.
2	EPS	R/W	Even parity selection 0: Odd parity 1: Even parity Controls a parity bit in transmission/reception. When <PEN> is set to "0", if parity check and generation are disabled, this bit has no meaning.
1	PEN	R/W	Parity enable 0: Disabled (Parity is disabled. A parity bit is not added.) 1: Enabled (Parity check and generation are enabled.)
0	BRK	R/W	Enables/disables break transmission 0: No break transmission 1: Performs break transmission When <BRK> is set to "1", "Low" level signal is output to UTxTXD output after currently ongoing transmission is complete. To establish break state, <BRK> must be keep "1" at least for two-frame transmission period. If break state is established, the contents of the transmit FIFO is not influenced. When break state is not transmitted, set "0" to <BRK>.

Note: When the contents of UARTxIBRD or UARTxFBRD are updated, UARTxLCR_H always must be written at the end of writing process.

Table 11-1 Truth table of UARTxLCR_H <SPS>, <EPS> and <PEN>

Parity enable <PEN>	Even parity selection <EPS>	Stick parity selection <SPS>	Parity selection (Transmission or check)
0	×	×	No transmission and check
1	1	0	Even parity transmission or even parity reception
1	0	0	Odd parity transmission or odd parity reception
1	0	1	"1" is sent/received as a parity bit.
1	1	1	"0" is sent/received as a parity bit.

Note: X ; don’t care

11.3.9 UARTxCR (UART 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	-	-	-	-	-	-	RXE	TXE
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	-	-	UARTEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-16	−	R	Read as an undefined value.
15	-	R/W	Write as "0".
14	-	R/W	Write as "0".
13-12	-	R	Read as an undefined value.
11	-	R/W	Write as "0".
10	-	R/W	Write as "0".
9	RXE	R/W	Enables/disables the reception 0: Disabled 1: Enabled When <RXE> is set to "1", reception is enabled. If reception is disabled during the reception, reception stops after currently ongoing data reception is complete.
8	TXE	R/W	Enables/disables the transmission 0: Disabled 1: Enabled When <TXE> is set to "1", transmission is enabled. If transmission is disabled during transmission, transmission stops after currently ongoing transmission is complete.
7	-	R/W	Write as "0".
6-3	-	R	Read as an undefined value.
2	-	R/W	Write as "0".
1	-	R/W	Write as "0".
0	UARTEN	R/W	Enables/disables the UART 0: Disabled 1: Enabled When <UARTEN> is set to "0", the UART is disabled. If the UART is disabled during transmission or reception, the UART stops after currently ongoing data transmission or reception is complete.

11.3.10 UARTxIFLS (UART Interrupt FIFO Level 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	-	-	RXIFLSEL			TXIFLSEL		
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-6	−	R	Read as an undefined value.
5-3	RXIFLSEL[2:0]	R/W	Selects the reception interrupt FIFO level 000: The Receive FIFO ≥ full of 1/8 001: The receive FIFO ≥ full of 1/4 010: The receive FIFO ≥ full of 1/2 011: The receive FIFO ≥ full of 3/4 100: The receive FIFO ≥ full of 7/8 Other than the above settings: Reserved Selects the receive FIFO interrupt level. The FIFO level is not a trigger of interrupts. Interrupts is generated when the transition is made through the specified FIFO level. For example, if the FIFO level is set to full of 1/8 (4 bytes), when the 4th byte data is stored in the receive FIFO, an interrupt occurs (after a STOP bit is received).
2-0	TXIFLSEL[2:0]	R/W	Selects the transmission interrupt FIFO level 000: The transmit FIFO ≤ full of 1/8 001: The transmit FIFO ≤ full of 1/4 010: The transmit FIFO ≤ full of 1/2 011: The transmit FIFO ≤ full of 3/4 100: The transmit FIFO ≤ full of 7/8 Other than the above settings: Reserved Selects the transmit FIFO interrupt level. Interrupts are not generated at reaching to the specified interrupt level. They are generated when the transition is made through the specified interrupt level. For example, if the FIFO level is set to full of 1/8 (4 bytes), when the 5th data is read from the transmit FIFO (when transmission of a STOP bit is started), when data in the FIFO increases to 4bytes, an interrupt occurs.

11.3.11 UARTxIMSC (UART Interrupt Disable/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	-	-	-	-	-	OEIM	BEIM	PEIM
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	FEIM	RTIM	TXIM	RXIM	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-11	−	R	Read as an undefined value.
10	OEIM	R/W	Overrun error interrupt mask 0: Disabled 1: Enabled
9	BEIM	R/W	Break error interrupt mask 0: Disabled 1: Enabled
8	PEIM	R/W	Parity error interrupt mask 0: Disabled 1: Enabled
7	FEIM	R/W	Framing error interrupt mask 0: Disabled 1: Enabled
6	RTIM	R/W	Receive timeout interrupt mask 0: Disabled 1: Enabled
5	TXIM	R/W	Transmit interrupt mask 0: Disabled 1: Enabled
4	RXIM	R/W	Receive interrupt mask 0: Disabled 1: Enabled
3	-	R/W	Write as "0".
2	-	R/W	Write as "0".
1	-	R/W	Write as "0".
0	-	R/W	Write as "0".

11.3.12 UARTxRIS (UART Raw 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	-	-	-	-	-	OERIS	BERIS	PERIS
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	FERIS	RTRIS	TXRIS	RXRIS	-	-	-	-
After reset	0	0	0	0	Undefined	Undefined	Undefined	Undefined

Bit	Bit symbol	Type	Function
31-11	−	R	Read as an undefined value.
10	OERIS	R	Overrun error interrupt status 0: No interrupt request 1: An interrupt is requested.
9	BERIS	R	Break error interrupt status 0: No interrupt request 1: An interrupt is requested.
8	PERIS	R	Parity error interrupt status 0: No interrupt request 1: An interrupt is requested.
7	FERIS	R	Framing error interrupt status 0: No interrupt request 1: An interrupt is requested.
6	RTRIS	R	Receive time out interrupt status 0: No interrupt request 1: An interrupt is requested.
5	TXRIS	R	Transmit interrupt status 0: No interrupt request 1: An interrupt is requested.
4	RXRIS	R	Receive interrupt status 0: No interrupt request 1: An interrupt is requested.
3	-	R	Read as an undefined value.
2	-	R	Read as an undefined value.
1	-	R	Read as an undefined value.
0	-	R	Read as an undefined value.

11.3.13 UARTxMIS (UART Masked 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	-	-	-	-	-	OEMIS	BEMIS	PEMIS
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	FEMIS	RTMIS	TXMIS	RXMIS	-	-	-	-
After reset	0	0	0	0	Undefined	Undefined	Undefined	Undefined

Bit	Bit symbol	Type	Function
31-11	−	R	Read as an undefined value.
10	OEMIS	R	Overrun error mask interrupt status 0: No interrupt request 1: An interrupt is requested.
9	BEMIS	R	Break error mask interrupt status 0: No interrupt request 1: An interrupt is requested.
8	PEMIS	R	Parity error mask interrupt status 0: No interrupt request 1: An interrupt is requested.
7	FEMIS	R	Framing error mask interrupt status 0: No interrupt request 1: An interrupt is requested.
6	RTMIS	R	Receive time out mask interrupt status 0: No interrupt request 1: An interrupt is requested.
5	TXMIS	R	Transmit mask interrupt status 0: No interrupt request 1: An interrupt is requested.
4	RXMIS	R	Receive mask interrupt status 0: No interrupt request 1: An interrupt is requested.
3	-	R	Read as an undefined value.
2	-	R	Read as an undefined value.
1	-	R	Read as an undefined value.
0	-	R	Read as an undefined value.

11.3.14 UARTxICR (UART Interrupt 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	-	-	-	-	-	OEIC	BEIC	PEIC
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	FEIC	RTIC	TXIC	RXIC	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-11	−	W	Write as "0".
10	OEIC	W	Overrun error interrupt clear 0: Invalid 1: Clear
9	BEIC	W	Break error interrupt clear 0: Invalid 1: Clear
8	PEIC	W	Parity error interrupt clear 0: Invalid 1: Clear
7	FEIC	W	Framing error interrupt clear 0: Invalid 1: Clear
6	RTIC	W	Receive time out interrupt clear 0: Invalid 1: Clear
5	TXIC	W	Transmit interrupt clear 0: Invalid 1: Clear
4	RXIC	W	Receive interrupt clear 0: Invalid 1: Clear
3	-	W	Write as "0".
2	-	W	Write as "0".
1	-	W	Write as "0".
0	-	W	Write as "0".

Note: The UARTxICR register is interrupt clear register for write-only. If the bits of this register are set to "1", the corresponding interrupt is cleared. Writing "0" is invalid.

11.3.15 UARTxHCCR (50% Duty 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	HCLPB	HCZR			-	HCST	HCMD	HCCR
After reset	0	0	1	1	0	0	0	0

Bit	Bit symbol	Type	Function
31-8	−	R	Write as "0".
7	HCLPB	R/W	Loop back test enable 0: Disable 1: Enable
6-4	HCZR[2:0]	R/W	"0" period detection control 000: Reserved 001: 1/16 width more than "0" detection 010: 2/16 width more than "0" detection 011: 3/16 width more than "0" detection 100: 4/16 width more than "0" detection 101: 5/16 width more than "0" detection 110: 6/16 width more than "0" detection 111: 7/16 width more than "0" detection
3	-	R/W	Write as "0".
2	HCST	R/W	Terminal selection to start the start bit 0: UTxTXD50A 1: UTxTXD50B
1	HCMD	R/W	Transmission terminal mode selection 0: 1 terminal mode 1: 2 terminal mode
0	HCCR	R/W	50% duty mode enable 0: Disable 1: Enable

Note 1: The loop-back test control is the test function in the development stage. Do not use for the final products.

11.4 Operation Description

11.4.1 Transmit FIFO and Receive FIFO

11.4.1.1 Transmit FIFO

The transmit FIFO is an 8-bit width and 32-deep memory buffer. The CPU data written via APB interface is stacked to this FIFO until the data is read by the transmit. When the transmit FIFO is disabled, it can be served as a 1-byte hold register.

11.4.1.2 Receive FIFO

The receive FIFO is a 12-bit width and 32-deep memory buffer. Error bits corresponding to receive data are stacked into the receive FIFO using receive logic until the data is read by the CPU via APB interface. When the receive FIFO is disabled, it can serve as a 1-byte hold register.

Bit	11	10	9	8	7	6	5	4	3	2	1	0
Receive FIFO	Overrun Error	Break Error	Parity Error	Framing Error	Receive data							
Receive data 8 bits	Overrun Error	Break Error	Parity Error	Framing Error	Receive data							
Receive data 7 bits	Overrun Error	Break Error	Parity Error	Framing Error		Receive data						
Receive data 6 bits	Overrun Error	Break Error	Parity Error	Framing Error			Receive data					
Receive data 5 bits	Overrun Error	Break Error	Parity Error	Framing Error				Receive data				

Note: Empty bits in the receive data are undefined.

11.4.2 Transmit Data and Receive data

Data written in UARTDR is stacked into the transmit FIFO when the FIFO is enabled.

If the FIFO is disabled, the data is transferred to the transmit hold register.

Transfer is started by writing data. The data includes a start bit. If a parity bit is enabled, the data is transferred with a parity and stop bit.

Received data is a 12-bit width including 4-bit status bits (break error, framing error, parity error, and overrun error) and is stacked into the receive FIFO. If the FIFO is disabled, received data and the status are transferred to the receive hold register.

11.4.3 Baud-rate Generator

The baud-rate generator outputs internal clocks: Baud16 and IrLPBaud16. Baud16 generates the timing for UART transmission/reception control. IrLPBaud16 generates a pulse width of IrDA encode transmit bit stream in the low power mode.

The baud-rate is calculated by the following equation using the $f_{UARTCLK}$ input from the UART and a baud-rate divisor.

Baud-rate = ($f_{UARTCLK}$) / (16 x baud-rate divisor)

11.4.3.1 Calculating A Baud-rate Divisor

The baud-rate divisor is calculated as follows:

Baud-rate divisor BAUDDIV = ($f_{UARTCLK}$) / (16 x baud rate)

where $f_{UARTCLK}$ is a clock frequency of UART.

BAUDDIV consists of the integer part (BAUDDIVINT) and fractional part (BAUDDIVFRAC).

Example: Calculation of the divisor

If the required baud-rate is 230400 and $f_{UARTCLK}$ = 4 MHz

Baud rate divisor = $(4 \times 10^6)/ (16 \times 230400) = 1.085$

Accordingly, the integer part of baud-rate (BAUDDIVINT) = 1 and the fractional part of baud-rate = 0.085

Therefore, BAUDDIVFRAC is calculated as follows:

BAUDDIVFRAC = ((0.085 × 64) + 0.5) = 5.94 = 5 (Figures below the decimal point are omitted.)

Generated baud-rate divisor is calculated using the above integer part and fractional part as below:

BAUDDIV = 1 + 5/64 = 1.078

At this time, generated baud-rate is calculated as follows:

Generated baud-rate = $(4 \times 10^6)/ (16 \times 1.078) = 231911$

A margin of error = (231911 - 230400)/ 230400 × 100 = 0.656%

In addition, the maximum margin of error is 1/64 × 100 = 1.56% when the UARTxFBRD register is used. This margin of error is generated when UARTxFBRD = 1.

11.4.4 Transmit Logic

The transmit logic performs parallel-to-serial conversion on data read out from the transmit FIFO. Control logic outputs the signal beginning with a start bit, data bits starting with LSB, parity bit, and the stop bit, according to the specified configuration in the control registers.

11.4.5 Receive Logic

The receive logic performs serial-to-parallel conversion on the received bit stream after the start bit has been detected. Error checking for overrun, parity, frame and line break detection are also performed. Data related to an error bit for overrun, parity, framing and break is written to the receive FIFO.

11.4.6 Interrupt Generation Logic

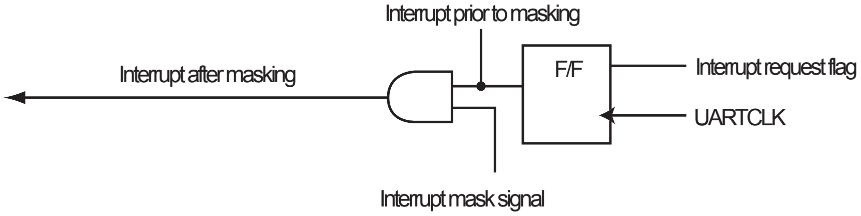
The UART outputs a maskable interrupt according to interrupt events.

11.4.6.1 UART Interrupt Generation Circuit

(1) Interrupt request flag generation circuit

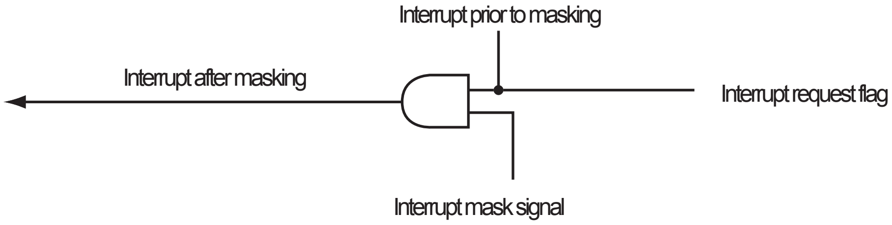
1. Generation circuit for break, parity and framing error flags

   An interrupt request flag changes in real-time associated with F/F. Each flag is cleared when data is written to the corresponding interrupt clear register.



2. Generation circuit for overrun error flag

   An interrupt request flag changes with overrun errors in real-time. The register status is not maintained. An overrun flag is cleared by reading the receive FIFO.



(2) UART interrupt

Each masked interrupt status is ORed and it is output from the UART as INTUARTx.

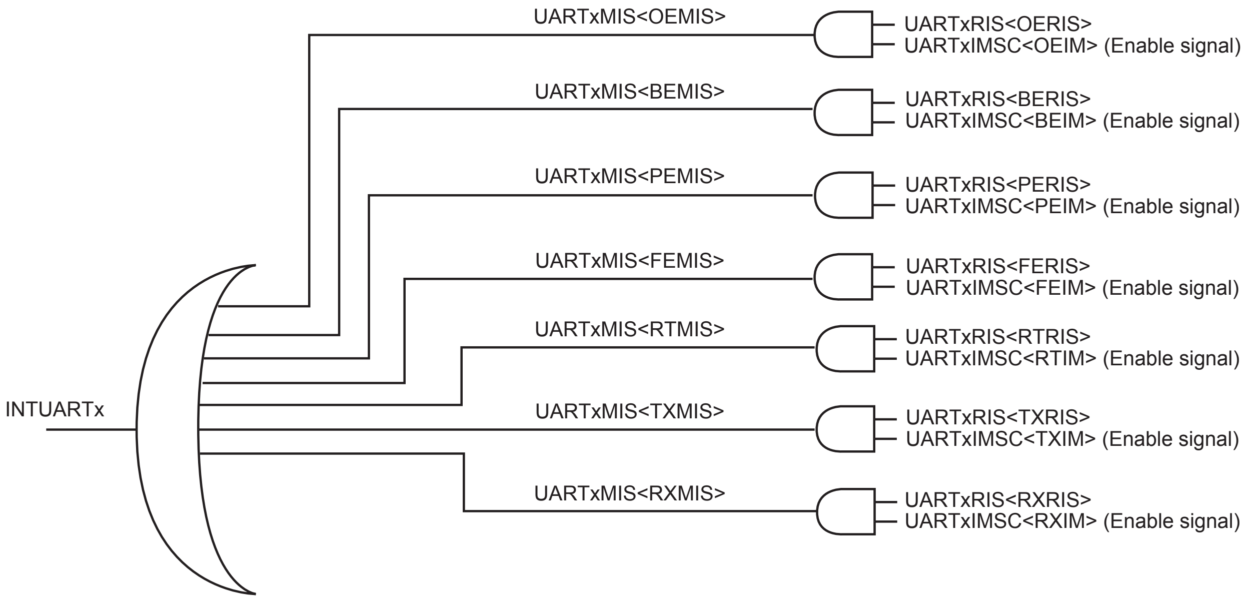


Figure 11-2 UART interrupt block

11.4.6.2 Interrupt Generation Timing

Interrupt source	Interrupt generation timing	
Overrun error generation	After a stop bit is received when FIFO is full.	
Break error interrupt	After a stop bit is received.	
Parity error generation	After a parity data is received.	
Framing error generation	After bit data that generates frame over is received.	
Reception timeout interrupt	After data is received in receive FIFO, then 511 clocks of Baud16 has elapsed.	
Transmission interrupt	When the FIFO is unused:	After the transmission is enabled, when a START bit and STOP bit in the first byte of the transmission data are sent, a transmit interrupt occurs. In the second byte and the following byte, a transmit interrupt occurs only when a STOP bit is sent. (In this case, each interrupt is cleared after the transmit data is written.)
	When the FIFO is used:	When a STOP bit is sent (after the MSB data is transmitted), if the amount of data in the FIFO is the same level as the specified level of FIFO, a transmit interrupt occurs.
Reception interrupt	When the FIFO is unused:	A receive interrupt occurs when the FUART receives a STOP bit.
	When the FIFO is used:	A receive interrupt occurs when the FUART receives a STOP bit included in the data that fills the FIFO to the specified level.

Note: In this table, a stop bit means a last stop bit. (A stop bit length is selectable with UARTxLCR_H<STP2>)

11.4.7 50% Duty Mode

11.4.7.1 Outline

The following functions are available in the 50% duty mode.

- Communication pins:

  For transmission UTxTXD50A and UTxTXD50B

  For reception UTxRXD50
- 1-pin transmission mode
- 2-pin transmission mode
- Selection of the start bit pin
- Detection control of data "0" in the reception period
- The loop-back test function

Note: In 50% duty mode, do not perform the full-duplex transmission except when the loop-back test is enabled.

11.4.7.2 Control

When "1" is set to UARTxHBSCR<HCCR>, the 50% duty control circuit is enabled, and the UART enters the 50% duty mode. Duty conversion is performed on transmission/reception signals in the 50% duty control circuit.

Port pin functions can select whether duty conversion is performed on the signal.

Figure 11-1 shows the circuit configuration.

Note: Do not modify the UARTxHBSCR register during the communication.

11.4.7.3 Operational Description

(1) Transmission

(a) 1-pin transmission mode

This mode transmits the same data through UTxTXD50A and UTxTXD50B.

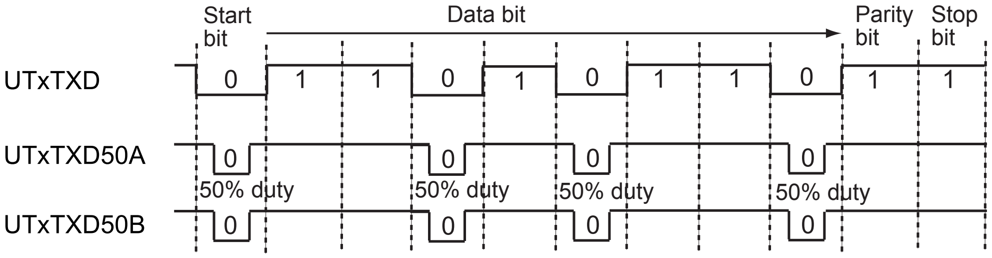


Figure 11-3 Example of waveforms in 1-pin transmission mode

(b) 2-pin transmission mode

When "1" is set to UARTxHBSCR<HCMD>, 2-pin transmission mode is enabled. Data "0" is transmitted through UTxTXD50A and UTxTXD50B alternatively.

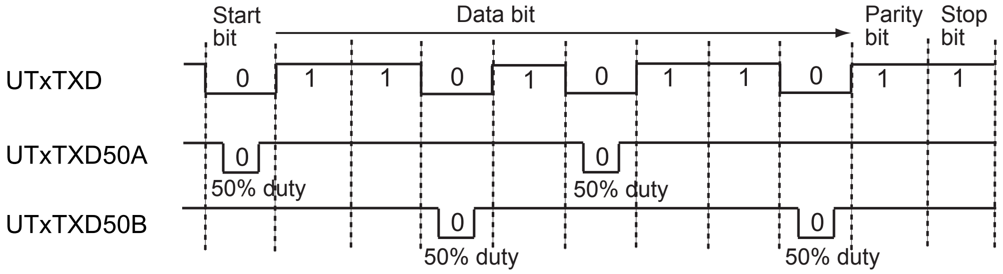


Figure 11-4 Example of waveforms in 2-pin transmission mode

(2) Reception

In reception, data is received through UTxRXD50 regardless of 1-pin mode or 2-pin mode.

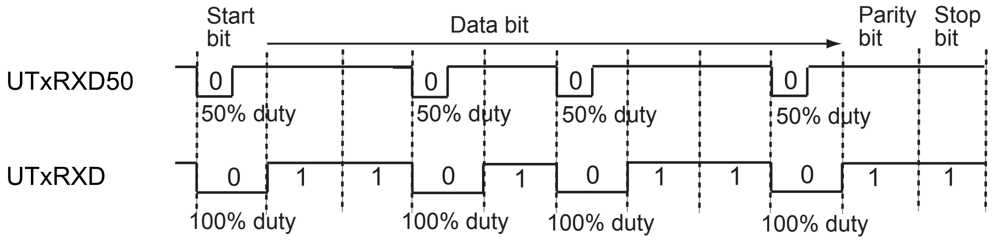


Figure 11-5 Example of waveforms in reception

(3) Selection of the start bit pin

A start bit can be selected either from UTxTXD50A or UTxTXD50B with UARTxHBSCR<HCST> in 2-pin transmission mode. Figure 11-6 shows an example where the start bit is set to UTxTXD50B. Figure 11-4 shows an example where the start bit is set to UTxTXD50A.

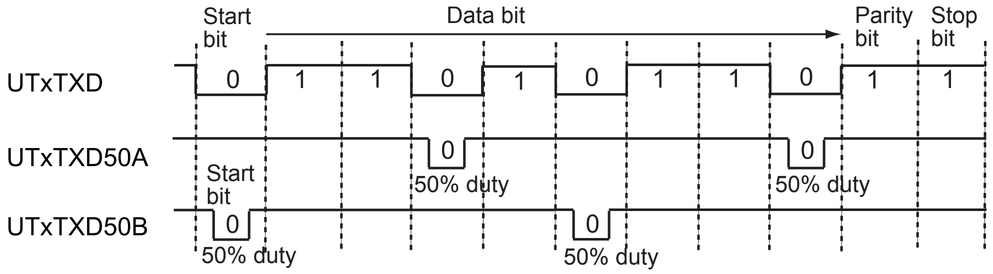


Figure 11-6 Example of the start bit

(4) Detection control of data "0" in the reception period

The detection width can be set with UARTxHBSCR<HCZR[2:0]>. The width is one that the data is recognized as "0" against width of one bit in 100% duty.

After a falling edge of data of "0" is detected, if "0" level continues over than the specified detection width, the data is captured as data of "0".

If data of "0" level is less than the specified width, the data is not recognized as "0".

Note that even when data of "0" level continues over than the specified detection width, if the noise which is over the width of UARTCLK occurs, the data is not recognized as "0".

The following figure shows an example where the detection width is set to 3/16 width.

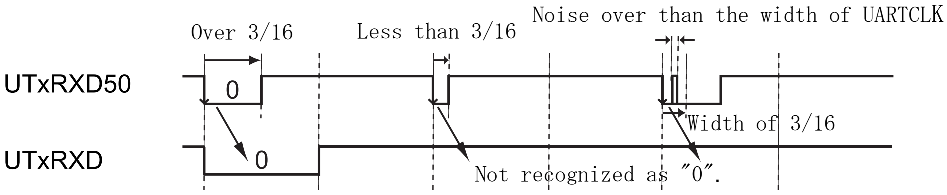


Figure 11-7 Example of data "0" detection period

(5) The loop-back test control

When "1" is set to UARTxHBSCR<HCLPB>, the loop-back test control is enabled. At this time, UTxTXD50A and UTxTXD50B are internally connected to UTxRXD50; therefore, TMPM381/383 can singly check transmit/receive operation.

UTxTXD50A and UTxTXD50B are ANDed. The result is sent to UTxRXD50.

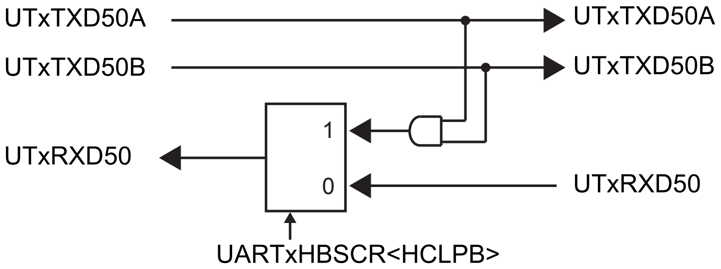


Figure 11-8 Connection in the loop-back test

Note 1: The loop-back test control is the test function in the development stage. Do not use for the final products.

Note 2: In 50% duty mode, full-duplex communication is only allowed when the loop-back test is enabled. Do not perform full-duplex communication except when the loop-back test is enabled.

Note 3: When the loop-back test is enabled, do not enter the external signal to the UART.

12. Serial Channel with 4bytes FIFO (SIO/UART)

12.1 Overview

Serial channel (SIO/UART) has the modes shown below.

- Synchronous communication mode (I/O interface mode)
- 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 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128.
  - Make it possible to divide from the prescaler output clock frequency into 1 to 16.
  - Make it possible to divide from the prescaler output clock frequency into N+m/16 (N=2 to 15, m=1 to 15). (only UART mode)
  - The usable system clock (fsys) (only UART mode).
- Buffer
  - The usable double buffer function.
  - Make it possible to clear the transmit buffer.
- FIFO

  The usable 4 byte FIFO including transmit and receive.
- I/O Interface Mode
  - Transfer Mode: the half duplex (transmit/receive), the full duplex
  - Clock: Output (fixed rising edge) /Input (selectable either rising or falling edge)
  - Make it possible to specify the interval time of continuous transmission.
  - The state of TXDx pin after output of the last bit can be selected as follow:

    Keep a "High" level, "Low" level or the state of the last bit
  - The state of TXDx pin when an under run error is occurred in clock input mode can be selected as follow:

    Keep a "High" level or "Low" level
  - The last bit hold time of TXDx pin can be specified in clock input mode.
- 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 $\overline{\text{CTSx}}$ pin
  - Noise cancel for RXDx pin

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

12.2 Configuration

Serial channel block diagram and serial clock generator circuit diagram are shown in belows:

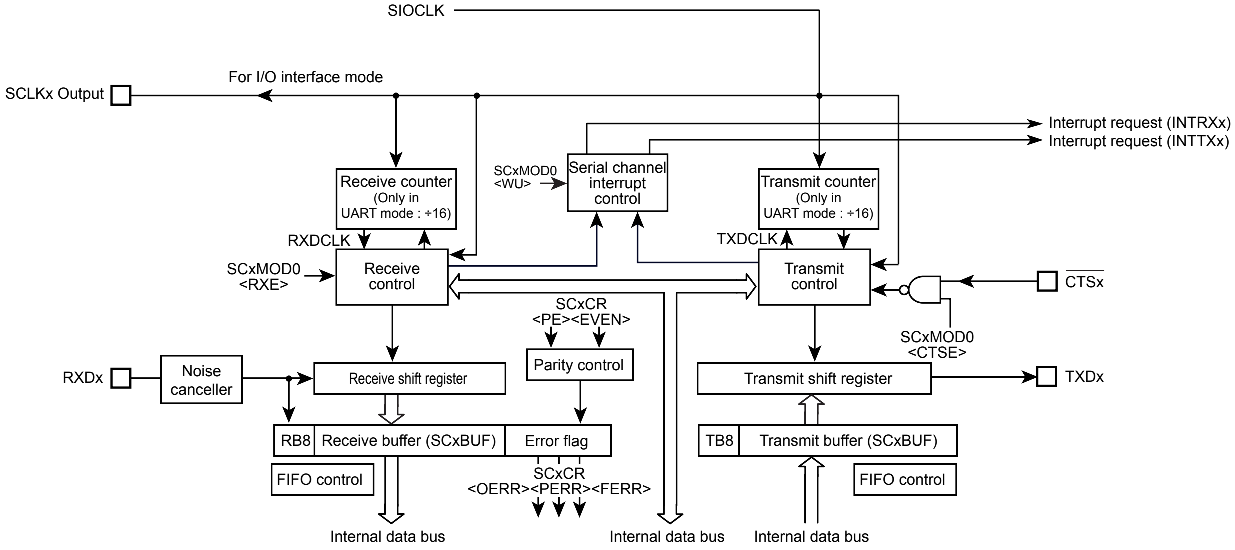


Figure 12-1 Serial Channel Block Diagram

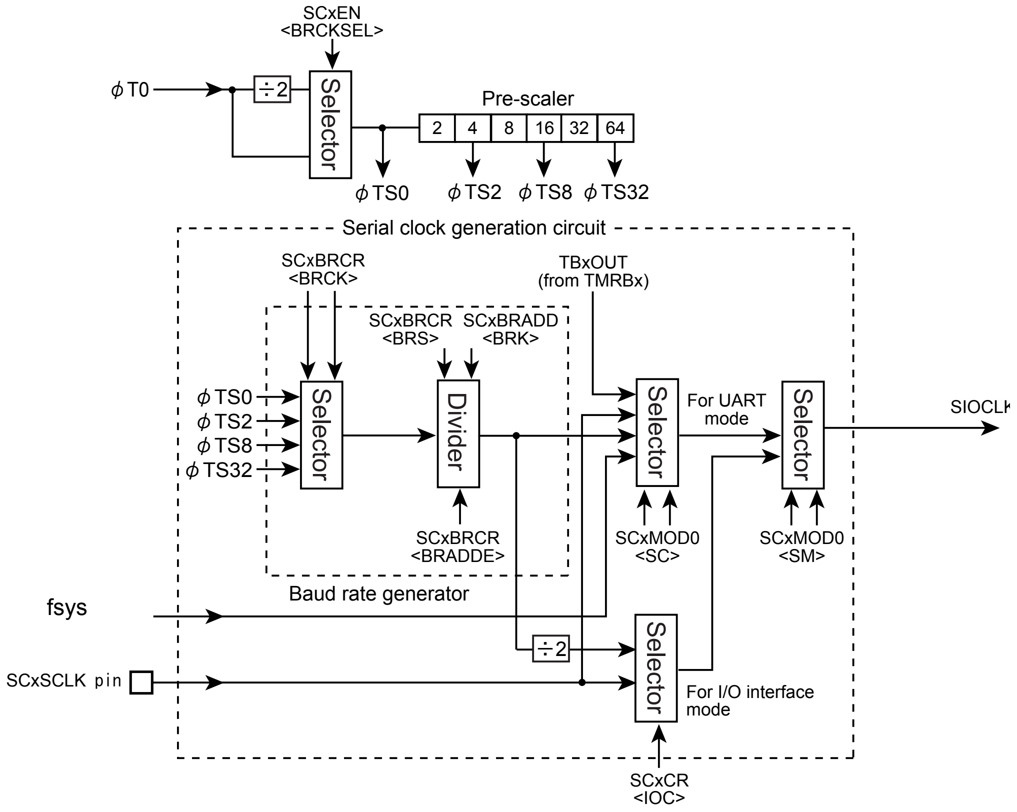


Figure 12-2 Serial clock generation circuit block diagram

12.3 Registers Description

12.3.1 Registers List

The table below shows control registers and their addresses.

For detail of the base address, refer to "Address lists of peripheral functions" of "Memory Map" chapter.

Register name		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
Receive FIFO configuration register	SCxRFC	0x0020
Transmit FIFO configuration register	SCxTFC	0x0024
Receive FIFO status register	SCxRST	0x0028
Transmit FIFO status register	SCxTST	0x002C
FIFO configuration register	SCxFCNF	0x0030

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

12.3.2 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	-	-	-	-	-	-	BRCKSEL	SIOE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	BRCKSEL	R/W	Selects input clock for prescaler. 0: φT0/2 1: φT0
0	SIOE	R/W	Serial channel operation 0: Disabled 1: Enabled Specified the Serial channel operation. To use the Serial channel, set <SIOE> = "1". When the operation is disabled, no clock is supplied to the other registers in the Serial channel module. This can reduce the power consumption. If the Serial channel operation is executed and then disabled, the settings will be maintained in each register.

12.3.3 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 or FIFO [read] RB: Receive buffer or FIFO

12.3.4 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	-	EHOLD			-	TXDEMP	TIDLE	
After reset	0	0	0	0	0	1	1	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
31-15	−	R	Read as "0".
14-12	EHOLD[2:0]	R/W	The last bit hold time of a TXDx pin in clock input mode (For only I/O interface mode) Set the last bit hold time and SCLK cycle to keep the last bit hold time equal or less than SCLK cycle/2. 000: 2/fsys 100: 32/fsys 001: 4/fsys 101: 64/fsys 010: 8/fsys 110: 128/fsys 011: 16/fsys 111: Reserved
11	-	R	Read as "0".
10	TXDEMP	R/W	The state of TXDx pin when an under run error is occurred in clock input mode. (For only I/O interface mode) 0: "Low" level output 1: "High" level output
9-8	TIDLE[1:0]	R/W	The state of TXDx pin after output of the last bit (For only I/O interface mode) When <TIDLE[1:0]> is set to "10", set "000" to <EHOLD[2:0]>. 00: Keep a "Low" level output 01 :Keep a "High" level output 10: Keep a last bit 11: Reserved
7	RB8	R	Receive data bit 8 (For only UART mode) 9th bit of the received data in the 9-bit UART mode.
6	EVEN	R/W	Parity (For only UART mode) Selects even or odd parity. The parity bit may be used only in the 7- or 8-bit UART mode. 0: Odd 1: Even Selects even or odd parity.
5	PE	R/W	Add parity (For only UART mode) Controls disabled or enabled parity. The parity bit may be used only in the 7- or 8-bit UART mode. 0: Disabled 1: Enabled
4	OERR	R	Over-run error flag (Note) 0: Normal operation 1: Error
3	PERR	R	Parity / Under-run error flag (Note) 0: Normal operation 1: Error
2	FERR	R	Framing error flag (Note) 0: Normal operation 1: Error

Bit	Bit Symbol	Type	Function
1	SCLKS	R/W	Selecting input clock edge (For I/O Interface mode) Set to "0" in the clock output mode. 0: Data in the transmit buffer is sent to TXDx pin every one bit on the falling edge of RXDx pin. Data from RXDx pin is received in the receive buffer every one bit on the rising edge of RXDx pin. In this case, the state of a RXDx pin starts from "High" level. (Rising edge mode) 1: Data in the transmit buffer is sent to TXDx pin every one bit on the rising edge of SCLKx pin. Data from RXDx pin is received in the receive buffer every one bit on the falling edge of SCLKx pin. In this case, the state of a SCLKx starts from "Low" level.
0	IOC	R/W	Selecting clock (For I/O Interface mode) 0: Clock output mode (A transfer clock is output from SCLKx pin.) 1: Clock input mode (A transfer clock is input to SCLKx pin.)

Note: <OERR>, <PERR> and <FERR> are cleared to "0" when read.

12.3.5 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 only UART mode) Writes the 9th bit of transmit data in the 9-bit UART mode.
6	CTSE	R/W	Handshake function control (For only UART mode) 0: CTS disabled 1: CTS enabled Controls handshake function. Setting "1" enables handshake function using $\overline{\text{CTSx}}$ pin.
5	RXE	R/W	Receive control (Note1)(Note2) 0: Disabled 1: Enabled
4	WU	R/W	Wake-up function (For only UART mode) 0: Disabled 1: Enabled This function is available only at 9-bit UART mode. In other mode, this function has no meaning. When it is enabled, interrupt is occurred only when RB9 = "1" in a 9-bit UART mode.
3-2	SM[1:0]	R/W	Specifies transfer mode. 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode
1-0	SC[1:0]	R/W	Serial transfer clock (For only UART mode) 00: TMRB output 01: Baud rate generator 10: System clock (fsys) 11: External clock (SCLKx pin input) (For the I/O interface mode, the transfer clock in I/O interface mode is selected by SCxCR<IOC>.)

Note 1: Specify the all mode control registers first and then the <RXE>.

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

12.3.6 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 operation in the IDLE mode.
6-5	FDPX[1:0]	R/W	Transfer mode setting 00: Transfer prohibited 01: Half duplex (Receive) 10: Half duplex (Transmit) 11: Full duplex Configures the transfer mode in the I/O interface mode. And when FIFO is enabled, specify the configuration of FIFO. In UART mode, specify the only configuration of FIFO.
4	TXE	R/W	Transmit control (Note1)(Note2) 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 mode) 000: None 001: 1 x SCLK cycle 010: 2 x SCLK cycle 011: 4 x SCLK cycle 100: 8 x SCLK cycle 101: 16 x SCLK cycle 110: 32 x SCLK cycle 111: 64 x SCLK cycle This parameter is valid only for the I/O interface mode when SCLK output mode is selected. In other modes, this parameter 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 control registers first and then enable the <TXE>.

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

12.3.7 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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-8</td><td>−</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>7</td><td>TBEMP</td><td>R</td><td>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".
When writing the transmit data to the transmit buffer, this bit is cleared to "0".</td></tr>
<tr><td>6</td><td>RBFLL</td><td>R</td><td>Receive buffer full flag
0: Empty
1: Full
If double buffering is disabled, this flag is insignificant.
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". When reading the receive buffer, this bit is cleared to "0".</td></tr>
<tr><td>5</td><td>TXRUN</td><td>R</td><td>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.
<table>
<tr><th>&lt;TXRUN&gt;</th><th>&lt;TBEMP&gt;</th><th>Status</th></tr>
<tr><td>1</td><td>−</td><td>Transmission in progress</td></tr>
<tr><td rowspan="2">0</td><td>1</td><td>Transmission is completed.</td></tr>
<tr><td>0</td><td>Wait state with data in transmit buffer</td></tr>
</table></td></tr>
<tr><td>4</td><td>SBLEN</td><td>R/W</td><td>STOP bit length (for UART mode)
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&gt;.</td></tr>
<tr><td>3</td><td>DRCHG</td><td>R/W</td><td>Setting transfer direction
0: LSB first
1: MSB first
Specifies the direction of data transfer.
In the UART mode, set this bit to LSB first.</td></tr>
<tr><td>2</td><td>WBUF</td><td>R/W</td><td>Enable 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 i n the I/O interface mode and to transmit in the UART mode.
When receiving data in the I/O interface mode (in clock input mode) and UART mode, double buffering is enabled regardless of the <WBUF&gt;.</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></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 a software reset is executed, the following bits are initialized and the transmit/receive circuit and FIFO become initial state (Note1)(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.

12.3.8 SCxBRCR (Baud Rate Generator 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	-	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 (Only for UART mode) 0: disabled 1: enabled
5-4	BRCK[1:0]	R/W	Select input clock to the baud rate generator. 00:φTS0 01:φTS2 10:φTS8 11:φTS32
3-0	BRS[3:0]	R/W	Division ratio "N" 0000: N = 16 0001: N = 1 0010: N = 2 ... 1111: N = 15

Note 1: 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 2: 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.

12.3.9 SCxBRADD (Baud Rate Generator 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	-	-	-	-	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 mode) 0000: Prohibited 0001: K = 1 0010: K = 2 ... 1111: K = 15

Table 12-1 lists the settings of baud rate generator division ratio.

Table 12-1 Setting division ratio

	<BRADDE> = "0"	<BRADDE> = "1" (Note1) (Only in the UART mode)
<BRS>	Specify "N"	
<BRK>	No setting required	Specify "K" (Note2)
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: Specifying "K = 0" is prohibited.

12.3.10 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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-8</td><td>−</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>7-5</td><td>-</td><td>R/W</td><td>Be sure to write "000".</td></tr>
<tr><td>4</td><td>RFST</td><td>R/W</td><td>Bytes used in receive FIFO.
0: Maximum
1: Same as FILL level of receive FIFO
The number of receive FIFO bytes to be used is selected. (Note1)
0: The maximum number of bytes of the FIFO configured (see also <CNFG>).
1: Same as the fill level for receive interrupt generation specified by SC0RFC <RIL[1:0]>.</td></tr>
<tr><td>3</td><td>TFIE</td><td>R/W</td><td>Specify transmit interrupt for transmit FIFO.
0: Disabled
1: Enabled
When transmit FIFO is enabled, transmit interrupts are enabled or disabled by this parameter.</td></tr>
<tr><td>2</td><td>RFIE</td><td>R/W</td><td>Specify receive interrupt for receive FIFO.
0: Disabled
1: Enabled
When receive FIFO is enabled, receive interrupts are enabled or disabled by this parameter.</td></tr>
<tr><td>1</td><td>RXTXCNT</td><td>R/W</td><td>Automatic disable of RXE/TXE.
0: None
1: Auto disable
Controls automatic disabling of transmission and reception.
Setting "1" enables to operate as follows.
<table>
<tr><td>Half duplex Receive</td><td>When the receive shift register, receive buffers and receive FIFO are filled up to the specified number of valid bytes, SCxMOD0<RXE> is automatically set to "0" to inhibit further reception.</td></tr>
<tr><td>Half duplex Transmit</td><td>When the transmit shift register, transmit buffers and the transmit FIFO are empty, SCxMOD1<TXE> is automatically set to "0" to inhibit further transmission.</td></tr>
<tr><td>Full duplex</td><td>When either of the above two conditions is satisfied, <TXE> and <RXE> are automatically set to "0" to inhibit further transmission and reception.</td></tr>
</table></td></tr>
<tr><td>0</td><td>CNFG</td><td>R/W</td><td>FIFO enable.
0: Disabled
1: Enabled
Enables FIFO.(Note2)
When <CNFG> is set to "1", FIFO is enabled. If FIFO is enabled, the SCOMOD1 <FDPX[1:0]> setting automatically configures FIFO as follows:
<table>
<tr><td>Half duplex Receive</td><td>Receive FIFO 4bytes</td></tr>
<tr><td>Half duplex Transmit</td><td>Transmit FIFO 4bytes</td></tr>
<tr><td>Full duplex</td><td>Receive FIFO 2bytes and Transmit FIFO 2bytes</td></tr>
</table></td></tr>
</table>

Note 1: Regarding Transmit FIFO, the maximum number of bytes being configured is always available. (See also <CNFG>.)

Note 2: The FIFO can not be used in 9 bit UART mode.

12.3.11 SCxRFC (Receive 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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-8</td><td>−</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>7</td><td>RFCS</td><td>W</td><td>Receive FIFO clear (Note1)
1: Clear
When SCxRFC<RFCS> is set to "1", the receive FIFO is cleared and SCxRST<RLVL[2:0]> is "000". And also the read pointer is initialized.
Read as "0".</td></tr>
<tr><td>6</td><td>RFIS</td><td>R/W</td><td>Select interrupt generation condition.
0: When FIFO fill level (SCxRST<RLVL[2:0]>) = Receive FIFO fill level to generate receive interrupt (<RIL [1:0]>)
1: When FIFO fill level (SCxRST<RLVL[2:0]>) ≥ Receive FIFO fill level to generate receive interrupt (<RIL [1:0]>)
For the detail of interrupt condition, refer to "12.12.1.2 FIFO"</td></tr>
<tr><td>5-2</td><td>-</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>1-0</td><td>RIL[1:0]</td><td>R/W</td><td>FIFO fill level to generate receive interrupts.
.
<table>
<tr><th></th><th>Half duplex</th><th>Full duplex</th></tr>
<tr><td>00</td><td>4 bytes</td><td>2 bytes</td></tr>
<tr><td>01</td><td>1 byte</td><td>1 byte</td></tr>
<tr><td>10</td><td>2 bytes</td><td>2 bytes</td></tr>
<tr><td>11</td><td>3 bytes</td><td>1 byte</td></tr>
</table></td></tr>
</table>

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

12.3.12 SCxTFC (Transmit 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	-	-	-	-	-	-	-	TBCLR
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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-9</td><td>−</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>8</td><td>TBCLR</td><td>W</td><td>Transmit buffer clear
0: Don’t care
1: Clear
When SCxTFC<TBCLR> is set to "1", the transmit buffer is cleared.
Read as "0".</td></tr>
<tr><td>7</td><td>TFCS</td><td>W</td><td>Transmit FIFO clear (Note1)
0: Don’t care
1: Clear
When SCxTFC<TFCS> is set to "1", the transmit FIFO is cleared and SCxTST<TLVL[2:0]> is "000". And also the write pointer is initialized.
Read as "0".</td></tr>
<tr><td>6</td><td>TFIS</td><td>R/W</td><td>Selects interrupt generation condition.
0: When FIFO fill level (SCxTST<TLVL[2:0]>) = Transmit FIFO fill level to generate transmit interrupt (<TIL [1:0]>)
1: When FIFO fill level (SCxTST<TLVL[2:0]>) ≤ Transmit FIFO fill level to generate transmit interrupt (<TIL [1:0]>)
For the detail of interrupt condition, refer to "12.12.2.2 FIFO"</td></tr>
<tr><td>5-2</td><td>−</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>1-0</td><td>TIL[1:0]</td><td>R/W</td><td>Fill level which transmit interrupt is occurred.
<table>
<tr><th></th><th>Half duplex</th><th>Full duplex</th></tr>
<tr><td>00</td><td>Empty</td><td>Empty</td></tr>
<tr><td>01</td><td>1 byte</td><td>1 byte</td></tr>
<tr><td>10</td><td>2 bytes</td><td>Empty</td></tr>
<tr><td>11</td><td>3 bytes</td><td>1 byte</td></tr>
</table></td></tr>
</table>

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

Note 2: In case that SCxEN<SIOE>="0" (Stop SIO/UART operation) or the operation mode is changed to IDLE mode with SCxMOD<I2SC>="0" (Stop SIO/UART operation in IDLE mode), SCxTFC is initialized again.After you perform the following operations, configure the SCxTFC register again.

12.3.13 SCxRST (Receive 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	Receive FIFO Overrun. (Note) 0: Not generated 1: Generated
6-3	−	R	Read as "0".
2-0	RLVL[2:0]	R	Status of Receive FIFO fill level. 000: Empty 001: 1 byte 010: 2 bytes 011: 3 bytes 100: 4 bytes

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

12.3.14 SCxTST (Transmit 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	1	0	0	0	0	0	0	0

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

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

12.4 Operation in Each Mode

Table 12-2 shows the modes.

Table 12-2 Modes

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

The Mode 0 is a synchronous communication and can be used to extend I/O. This mode transmits and receives data in synchronization with SCLK clock. SCLK clock can be used for both input and output modes. The direction of data transfer can be selected from LSB first or 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 directions can be selected as only 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.

12.5 Data Format

12.5.1 Data Format List

Figure 12-3 shows data format.

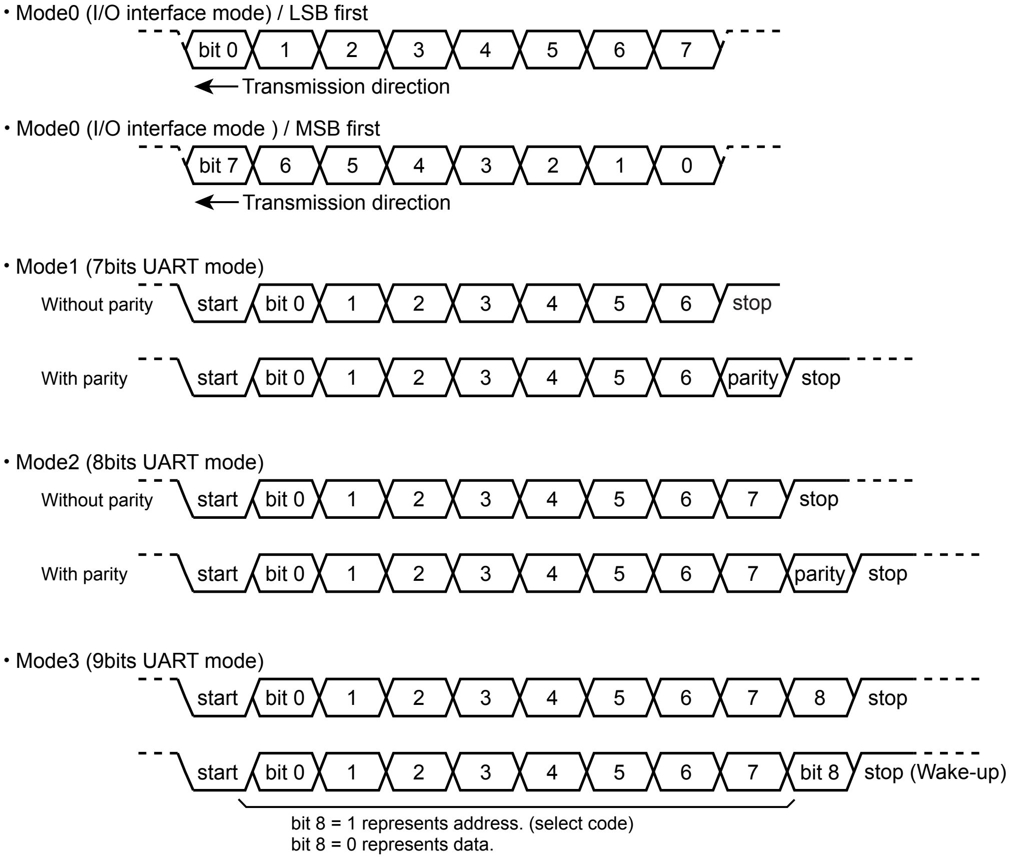

• Mode0 (I/O interface mode) / LSB first
bit 0 1 2 3 4 5 6 7
Transmission direction
• Mode0 (I/O interface mode) / MSB first
bit 7 6 5 4 3 2 1 0
Transmission direction
• Mode1 (7bits UART mode)
Without parity
start bit 0 1 2 3 4 5 6 stop
With parity
start bit 0 1 2 3 4 5 6 parity stop
• Mode2 (8bits UART mode)
Without parity
start bit 0 1 2 3 4 5 6 7 stop
With parity
start bit 0 1 2 3 4 5 6 7 parity stop
• Mode3 (9bits UART mode)
start bit 0 1 2 3 4 5 6 7 8 stop
start bit 0 1 2 3 4 5 6 7 bit 8 stop (Wake-up)
bit 8 = 1 represents address. (select code)
bit 8 = 0 represents data.

Figure 12-3 Data Format

12.5.2 Parity Control

The parity bit can be added with a transmitted data only in the 7- or 8-bit UART mode. And the received parity bit can be compared with a generated one.

Setting "1" to SCxCR<PE> enables the parity. SCxCR<EVEN> selects either even or odd parity.

12.5.2.1 Transmission

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

12.5.2.2 Reception

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>, 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 SCxCR<PERR> is set to "1".

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

12.5.3 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.

12.6 Clock Control

12.6.1 Prescaler

There is a 7-bit prescaler to divide a prescaler input clock φT0 by 1, 2, 4, 8, 16, 32, 64 and 128.

Use the CGSYSCR and SCxEN<BRCKSEL> in the clock/mode control block to select the input clock of the prescaler.

The prescaler becomes active only when the baud rate generator is selected as a transfer clock by SCxMOD0<SC[1:0]> = "01".

12.6.2 Serial Clock Generation Circuit

The serial clock generation 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.

12.6.2.1 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 1, 4, 16 and 64.

This input clock is selected by setting the SCxEN<BRCKSEL> and SCxBRCR<BRCK>.

SCxEN<BRCKSEL>	SCxBRCR<BRCK>	Baud rate generator input clock φTx
0	00	φT0/2
0	01	φT0/8
0	10	φT0/32
0	11	φT0/128
1	00	φT0
1	01	φT0/4
1	10	φT0/16
1	11	φT0/64

(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 1/(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[3:0]>	Divide by K SCxBRADD<BRK[3:0]>
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.

The input clock to the divider of baud rate generator is φTx, the baud rate in the case of 1/N and N + (16-K)/16 is shown below.

- Divide by N

$$\text{Baud rate} = \frac{\phi Tx}{N}$$

- N + (16-K)/16 division

$$\text{Baud rate} = \frac{\phi Tx}{N + \frac{(16 - K)}{16}}$$

12.6.2.2 Clock Selection Circuit

A clock can be selected by setting the modes and the register.

Modes can be specified by setting the SCxMOD0<SM[1:0]>

The clock in I/O interface mode is selected by setting SCxCR<IOC><SCLKS>.

The clock in UART mode is selected by setting SCxMOD0<SC[1:0]>.

(1) Transfer Clock in I/O interface mode

Table 12-3 shows clock selection in I/O interface mode.

Table 12-3 Clock Selection in I/O Interface Mode

Mode SCxMOD0<SM[1:0]>	Input/Output selection SCxCR<IOC>	Clock edge selection SCxCR<SCLKS>	Clock of use
"00" (I/O interface mode)	"0" (Clock output mode)	"0" (Transmit : falling edge, Receive : rising edge)	Divided by 2 of the baud rate generator output.
	"1" (Clock input mode)	"0" (Transmit : falling edge, Receive : rising edge)	SCLKx pin input
		"1" (Transmit : rising edge, Receive : falling edge)	SCLKx pin input

To use SCLKx input, the following conditions must be satisfied.

- <u>If double buffer is used</u>

  - SCLK cycle > 6/fsys
- <u>If double buffer is not used</u>

  - SCLK cycle > 8/fsys

(2) Transfer clock in the UART mode

Table 12-4 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 12-4 Clock Selection in UART Mode

Mode SCxMOD0<SM[1:0]>	Clock selection SCxMOD0<SC[1:0]>
UART Mode ("01", "10", "11")	"00" : TMRB output
	"01" : Baud rate generator
	"10" : fsys
	"11" : SCLKx pin input

To use SCLKx pin input, the following conditions must be satisfied.

- SCLK cycle > 2/fsys

To enable the timer output, a timer flip-flop output inverts when the value of the counter and that of TBxRG1 match. The SIOCLK clock frequency is "Setting value of TBxRG1 × 2".

Baud rates can be obtained by using the following formula.

Baud rate calculation

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by CGSYSCR<PRCK[1:0]>}}{\underbrace{(\text{TBxRG1} \times 2)}_{\text{One clock cycle is a period that the timer flip-flop is inverted twice.}} \times \underbrace{2}_{\text{In the case the timer prescaler clock } \Phi\text{T1 (2divition ratio) is selected.}} \times 16}$$

12.6.3 Transmit/Receive Buffer and FIFO

12.6.3.1 Configuration

Figure 12-4 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.

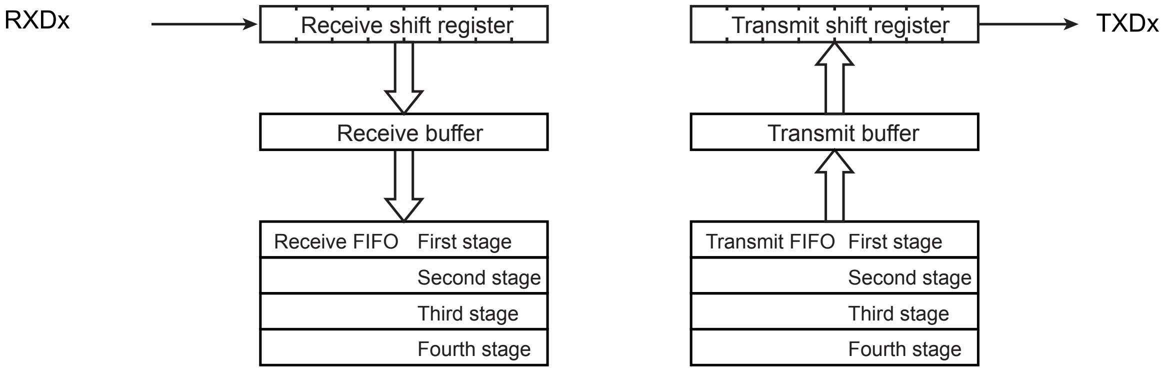


Figure 12-4 The Configuration of Buffer and FIFO

12.6.3.2 Transmit/Receive Buffer

Transmit buffer and receive buffer are double-buffered. The buffer configuration is specified by SCxMOD2<WBUF>.

When serial channel is operated as receive, if it is operated as clock input mode in the I/O interface mode or it is operated as the UART mode, it's double buffered regardless of <WBUF> settings.

In other modes, it's according to the <WBUF> settings.

Table 12-5 shows correlation between modes and buffers.

Table 12-5 Mode and buffer Composition

Mode		SCxMOD2<WBUF>	
		"0"	"1"
UART mode	Transmit	Single	Double
	Receive	Double	Double
I/O interface mode (Clock input mode)	Transmit	Single	Double
	Receive	Double	Double
I/O interface mode (Clock output mode)	Transmit	Single	Double
	Receive	Single	Double

12.6.3.3 Initialize Transmit Buffer

When transmission is stopped with a data in the transmit buffer, it is necessary to initialize the transmit buffer before new transmit data is written to transmit buffer.

The transmit buffer must be initialized when the transmit operation is stopped. To stop the transmit operation can be confirmed by reading SCxMOD2<TXRUN>. After confirming to stop the transmit operation, SCxTFC<TBCLR> is set to "1" and initialize the transmit buffer.

When a transmit FIFO is enabled, the initialize operation is depend on the data in a transmit FIFO. If transmit FIFO has data, a data is transferred from a transmit FIFO to a transmit buffer. If is does not have data, SCxMOD2<RBEMP> is set to "1".

Note: In the I/O interface mode with clock input mode is input asynchronously. When transmit operation is stopped, do not input the clock.

12.6.3.4 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 Transmit/Receive FIFO buffer, Transmit/Receive FIFO must be cleared after setting the SIO transfer mode (half duplex/ full duplex) and enabling FIFO (SCxFCNF<CNFG> = "1").**

Table 12-6 shows correction between modes and FIFO.

Table 12-6 Mode and FIFO Composition

	SCxMOD1<FDPX[1:0]>	Receive FIFO	Transmit FIFO
Half duplex Receive	"01"	4byte	-
Half duplex Transmit	"10"	-	4byte
Full duplex	"11"	2byte	2byte

12.7 Status Flag

The SCxMOD2 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". When reading the receive buffer is read, this bit is cleared to "0".

<TBEMP> shows that the transmit buffer is 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".

12.8 Error Flag

Three error flags are provided in the SCxCR. 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.

Mode	Flag		
	<OERR>	<PERR>	<FERR>
UART mode	Over-run error	Parity error	Framing error
I/O Interface mode (Clock input mode)	Over-run error	Under-run error (When a double buffer and FIFO are used)	Fixed to 0
		Fixed to 0 (When a double buffer and FIFO are not used)	
I/O Interface mode (Clock output mode)	Undefined	Undefined	Fixed to 0

12.8.1 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 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 mode with clock output mode, the SCLKx pin output stops upon setting the flag.

Note: **To switch from the I/O interface mode with clock output mode to other modes, read the SCxCR and clear the overrun flag.**

12.8.2 PERR Flag

This flag indicates a parity error in the UART mode and an under-run error or completion of transmit 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 received parity bit.

In the I/O interface mode, <PERR> is set to "1" under the following conditions when a double buffer is enabled.

In the clock input mode, <PERR> is set to "1" when the clock is input after completing data output of the transmit shift register with no data in the transmit buffer.

In the clock output mode, <PERR> is set to "1" after completing output of all data and the clock output stops.

Note: To switch from the I/O interface mode with clock output mode to other modes, read the SCxCR and clear the under-run flag.

12.8.3 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>, the stop bit status is determined by only 1’st STOP bit.

This bit is fixed to "0" in the I/O interface mode.

12.9 Receive

12.9.1 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 eighth pulse.

12.9.2 Receive Control Unit

12.9.2.1 I/O interface mode

In the clock output mode with SCxCR <IOC> set to "0", the RXDx pin is sampled on the rising edge of SCLKx pin.

In the clock input mode with SCxCR <IOC> set to "1", the RXDx pin is sampled on the rising or falling edge of SCLKx pin depending on the SCxCR <SCLKS>.

12.9.2.2 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.

12.9.3 Receive Operation

12.9.3.1 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 cleared to "0" by reading the receive buffer. When the double buffer is disabled, the receive buffer full flag has no meaning.

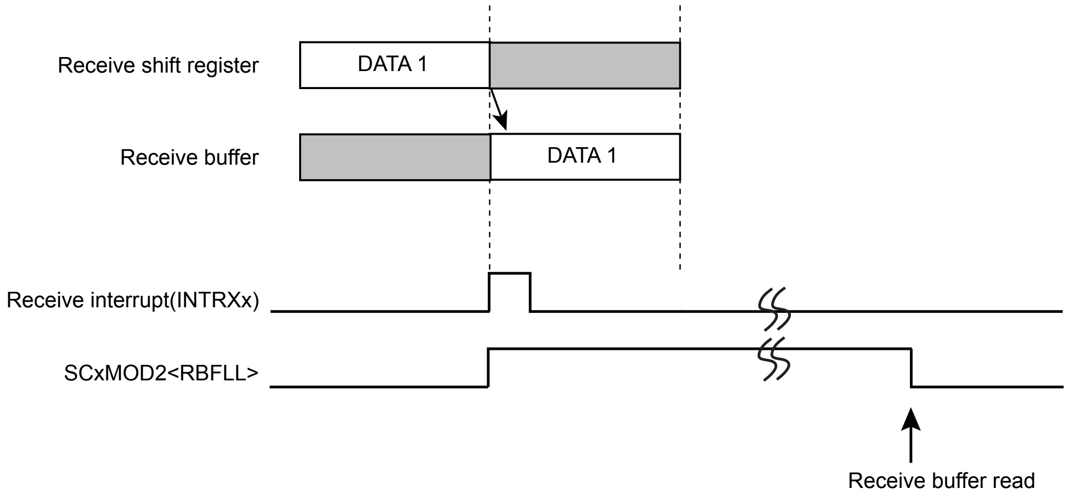


Figure 12-5 Receive Buffer Operation

12.9.3.2 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[1:0] >.

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 configurations and operations in the half duplex Receive mode are described as follows.

SCxMOD1<FDPX[1:0]> = "01"	:Transfer mode is set to half duplex mode
SCxFCNF<RFST><TFIE><RFIE>	:Automatically inhibits continuous reception after reaching the fill level.
<RXTCNT><CNFG> = "10111"	:The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.
SCxRFC<RIL[1:0]> = "00"	:The fill level of FIFO in which generated receive interrupt is set to 4 bytes
SCxRFC<RFCS><RFIS> = "01"	: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 operations completed.

In the above condition, if the continuous reception after reaching the fill level is enabled, it becomes possible to receive a data continuously by reading the data in the FIFO.

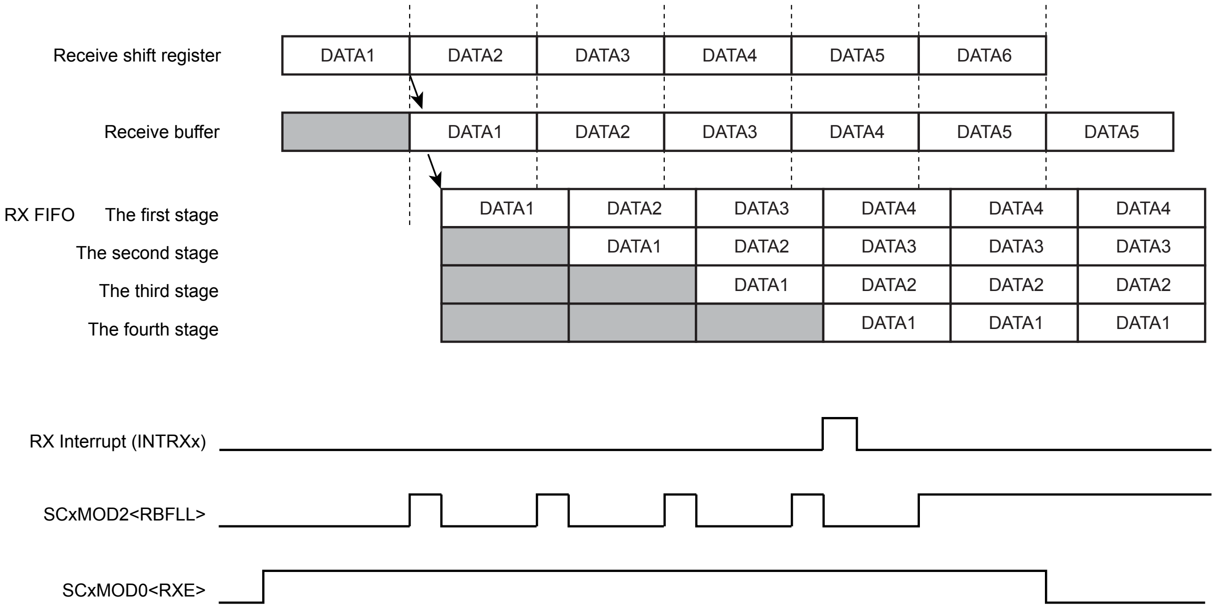


Figure 12-6 Receive FIFO Operation

12.9.3.3 I/O interface mode with clock output mode

In the I/O interface mode with clock output mode setting, clock stops when all received data is stored in the receive buffer and FIFO. So, in this mode, the over-run 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 clock 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, clock output is restarted.

(2) Case of double buffer

Stop clock output after receiving the data into a receive shift register and a receive buffer.

When a data is read, clock output is restarted.

(3) Case of FIFO

Stop clock 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 the received buffer and clock output restarts.

And if SCxFCNF<RXTXCNT>is set to "1", clock stops and receive operation stops with clearing SCxMOD0<RXE>.

12.9.3.4 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. 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 enabled, the 9-bit UART mode is prohibited because up to 8-bit data can be stored in receive 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>.

12.9.3.5 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".

12.9.3.6 Overrun Error

When receive FIFO is disabled, the overrun error occurs without completing reading data before receiving the next data. When an overrun error occurs, a content of receive buffer and SCxCR<RB8> is not lost, but a content of receive shift register is lost.

When receive FIFO is enabled, overrun error is occurred and set overrun flag by no reading receive FIFO before moving the next data into received buffer when receive FIFO is full. In this case, the contents of receive FIFO are not lost.

In the I/O interface mode with clock output mode, the clock output automatically stops, so this flag has no meaning.

Note: When the mode is changed from I/O interface mode with clock output mode to the other modes, read SCxCR and clear overrun flag.

12.10 Transmit

12.10.1 Transmit 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.

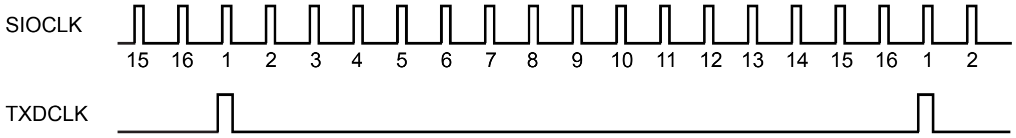


Figure 12-7 Generation of Transmission Clock in UART mode

12.10.2 Transmit Control

12.10.2.1 In I/O Interface Mode

In the clock output mode with SCxCR<IOC> set to "0", each bit of data in the transmit buffer is outputted to the TXDx pin on the falling edge of SCLKx pin.

In the clock input mode with SCxCR<IOC> set to "1", each bit of data in the transmit buffer is outputted to the TXDx pin on the rising or falling edge of the SCLKx pin according to the SCxCR<SCLKS>.

12.10.2.2 In 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.

12.10.3 Transmit Operation

12.10.3.1 Operation of Transmit Buffer

If double buffering is disabled, the CPU writes data only to transmit shift register 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".

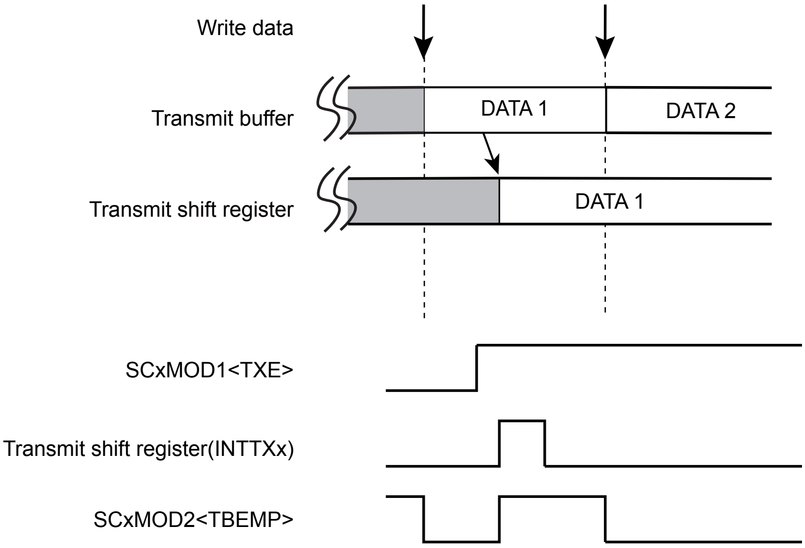


Figure 12-8 Operation of Transmit Buffer (Double-buffer is enabled)

12.10.3.2 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".

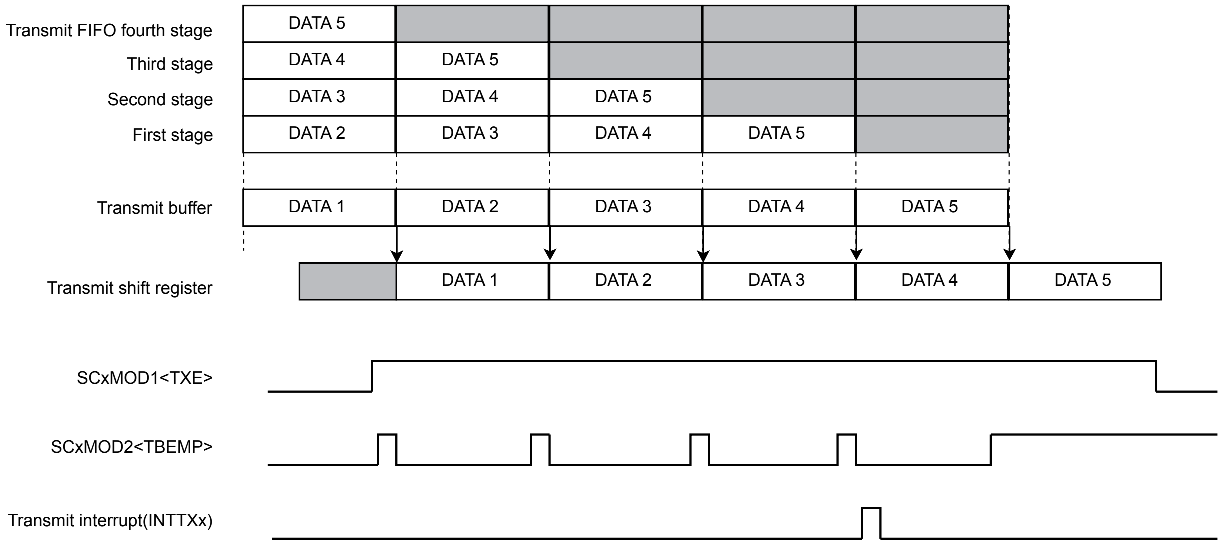
Note: To use Transmit FIFO buffer, Transmit FIFO must be cleared after setting the SIO transfer mode (half duplex/ full duplex) and enabling FIFO (SCxFCNF<CNFG>="1").

Settings and operations to transmit 5 bytes data stream by setting the transfer mode to half duplex are shown as below.

SCxMOD1<FDPX[1:0]> = "10"	:Transfer mode is set to half duplex.
SCxFCNF<RFST><TFIE><RFIE> <RXTXCNT><CNFG> = "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<TIL[1:0]> = "00"	:Sets the interrupt generation fill level to "0".
SCxTFC<TFCS><TFIS> = "11"	:Clears receive FIFO and sets the condition of interrupt generation.
SCxFCNF<CNFG> = "1"	:Enable FIFO

After above settings are configured, data transmission can be initiated by writing 5 bytes of data to the transmit buffer and FIFO, and setting the SCxMOD1<TXE> bit to "1". When the last transmit data is moved to the transmit buffer, the transmit 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 writing transmit data.



12.10.3.3 Transmit in I/O interface Mode with Clock Output Mode

In the I/O interface mode with clock output mode, the clock output automatically stops when all data transmission is completed and underrun error will not occur.

The timing of suspension and resume of clock output is different depending on the buffer and FIFO usage.

(1) Single Buffer

The clock output stops each time one frame of data is transferred. Handshaking for each data with the other side of communication can be enabled. The clock output resumes when the next data is written in the buffer.

(2) Double Buffer

The clock output stops upon completion of data transmission in the transmit shift register and the transmit buffer. The clock 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, clock output resumes.

If SCxFCNF<RXTXCNT> is configured, SCxMOD0<TXE> bit is cleared at the same time as clock stops and the transmission stops.

12.10.3.4 Level of TXDx pin after the last bit is output in I/O interface mode

The level of TXDx pin after the data hold time is passed after the last bit is output is specified by SCxCR<TIDLE>.

When SCxCR<TIDLE> is "00", the level of TXDx pin is output "Low" level. When SCxCR<TIDLE> is "01", the level of TXDx pin is output "High" level. When SCxCR<TIDLE> is "10", the level of TXDx pin is output the level of the last bit.

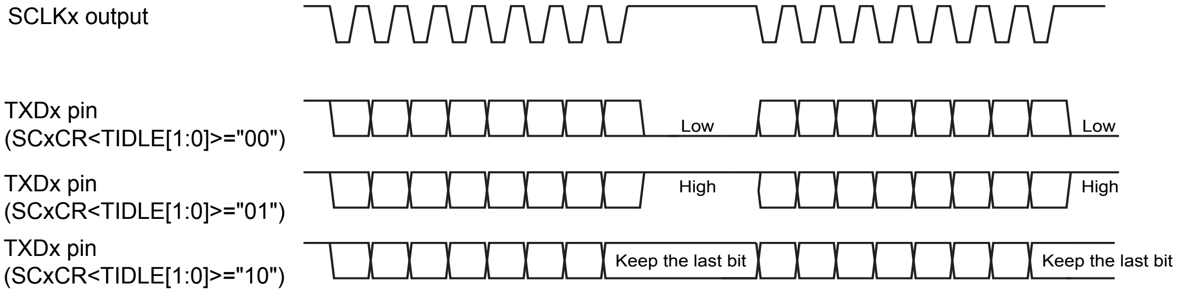


Figure 12-9 Level of TXDx pin After the last bit is output

12.10.3.5 Under-run error

In the I/O interface mode with clock input mode and if FIFO is empty 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".

The level of a TXDx pin can be specified by SCxCR<TXDEMP>. When SCxCR<TXDEMP> is "0", a TXDx pin outputs "Low" level during data output period. When SCxCR<TXDEMP> is "1", a TXDx pin outputs "High" level.

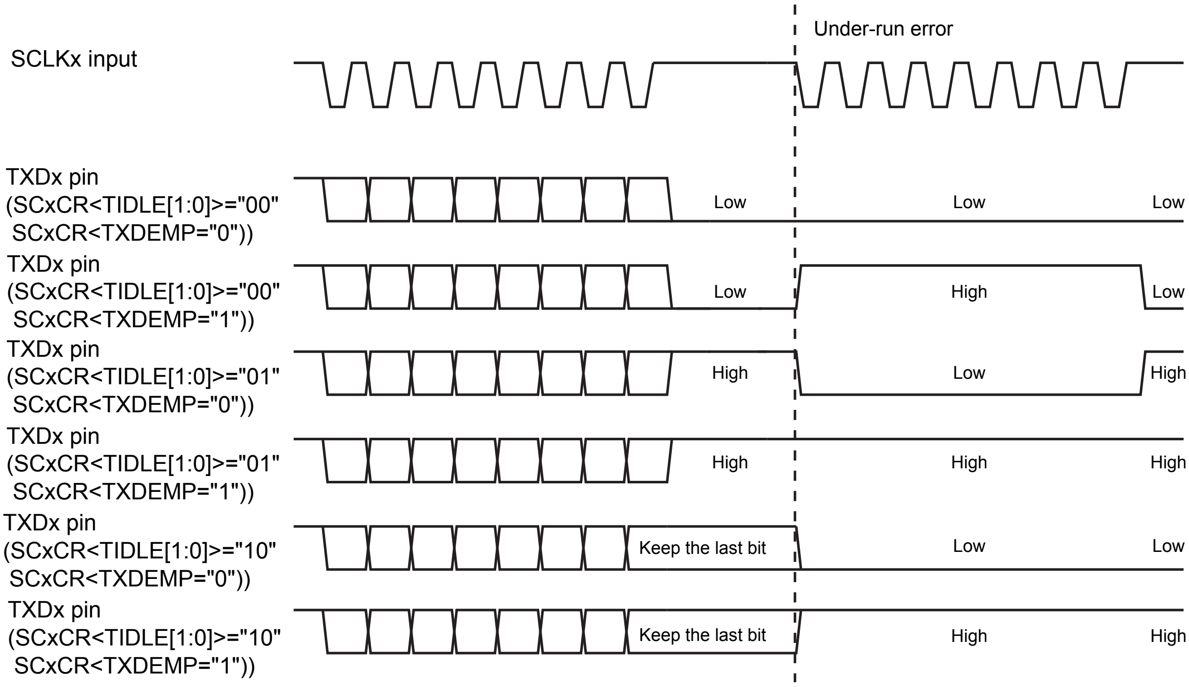

Under-run error
SCLKx input
TXDx pin (SCxCR<TIDLE[1:0]>="00" SCxCR<TXDEMP="0"))
Low
Low
Low
TXDx pin (SCxCR<TIDLE[1:0]>="00" SCxCR<TXDEMP="1"))
Low
High
Low
TXDx pin (SCxCR<TIDLE[1:0]>="01" SCxCR<TXDEMP="0"))
High
Low
High
TXDx pin (SCxCR<TIDLE[1:0]>="01" SCxCR<TXDEMP="1"))
High
High
High
TXDx pin (SCxCR<TIDLE[1:0]>="10" SCxCR<TXDEMP="0"))
Keep the last bit
Low
Low
TXDx pin (SCxCR<TIDLE[1:0]>="10" SCxCR<TXDEMP="1"))
Keep the last bit
High
High

Figure 12-10 Level of TXDx pin when Under-run Error is Occurred

In the I/O interface mode with SCLK output setting, the clock output automatically stops, so SCxCR<PERR> has no meaning.

Note: Before switching the I/O interface mode with clock output mode to other modes, read the SCxCR and clear the under-run flag.

12.10.3.6 Data Hold Time In the I/O interface mode with clock input mode

In the I/O interface mode with clock input mode, a data hold time of the last bit can be adjusted by SCxCR<EHOLD[2:0]>. Specify a data hold time and the period of the SCLK to satisfy the following formula.

The data hold time of the last bit ≤ The period of SCLK / 2

12.11 Handshake function

The function of the handshake is to enable frame-by-frame data transmission by using the $\overline{\text{CTSx}}$ (Clear to send) pin and to prevent over-run errors. This function can be enabled or disabled by SCxMOD0<CTSE>.

When the $\overline{\text{CTSx}}$ pin is set to "High" level, the current data transmission can be completed but the next data transmission is suspended until the $\overline{\text{CTSx}}$ pin returns to the "Low" level. 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 "High" level during transmission, the next data transmission is suspended after the current transmission is completed.

Note 2: Data transmission starts on the first falling edge of the TXDCLK clock after $\overline{\text{CTS}}$ is set to "Low" level.

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.

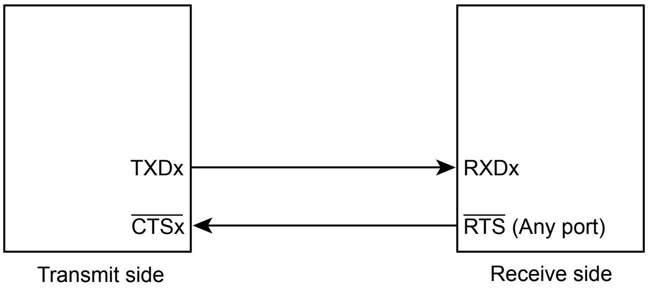


Figure 12-11 Handshake Function

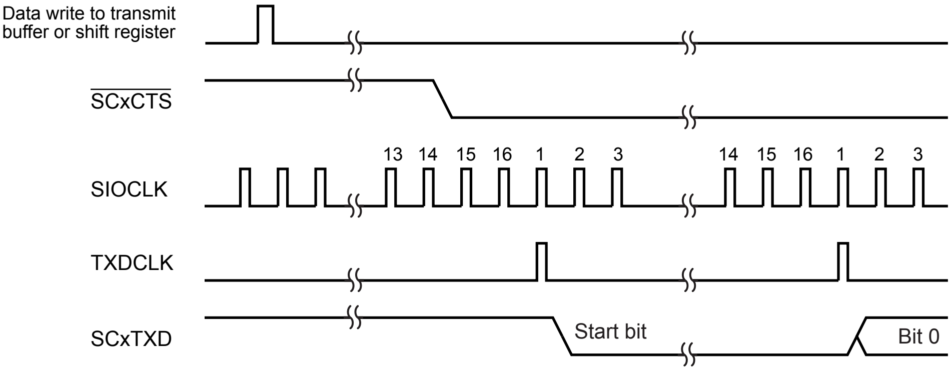


Figure 12-12 $\overline{\text{CTSx}}$ Signal timing

12.12 Interrupt/Error Generation Timing

12.12.1 Receive Interrupts

Figure 12-13 shows the data flow of receive operation and the route of read.

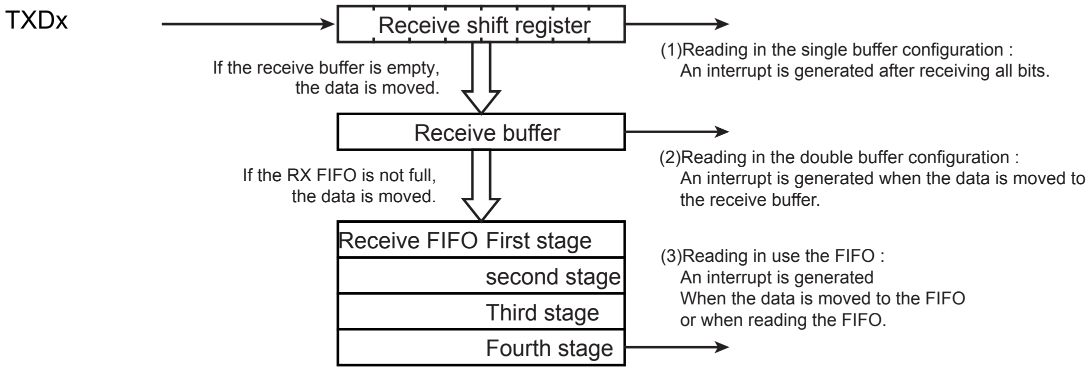

TXDx
Receive shift register
If the receive buffer is empty,
the data is moved.
(1)Reading in the single buffer configuration :
An interrupt is generated after receiving all bits.
Receive buffer
If the RX FIFO is not full,
the data is moved.
(2)Reading in the double buffer configuration :
An interrupt is generated when the data is moved to
the receive buffer.
Receive FIFO First stage
second stage
Third stage
Fourth stage
(3)Reading in use the FIFO :
An interrupt is generated
When the data is moved to the FIFO
or when reading the FIFO.

Figure 12-13 Receive Buffer/FIFO Configuration Diagram

12.12.1.1 Single Buffer / Double Buffer

Receive interrupts are generated at the time depends on the transfer mode and the buffer configurations, which are given as follows.

Table 12-7 Receive Interrupt Conditions in use of Single Buffer / Double Buffer

Buffer Configurations	UART modes	IO interface modes
Single Buffer	−	Immediately after the raising / falling edge of the last SCLKx pin (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	A receive interrupt occurs when data is transferred from the receive shift register to the receive buffter.Specific timings are : ・If data does not exist in the receive buffer, a receive interrupt occurs in the vicinity of the center of the 1st stop bit. ・If data exists in both the receive shift register and the receive buffer, a receive interrupt occurs when the buffer is read.	A receive interrupt occurs when data is transferred from the receive shift register to the receive buffer.Specific timings are: ・If data does not exit int the receive buffer, a receive interrupt occurs immediately after on rising/falling edge of SCxSCLK pin of the last bit. (The setting of rising edge or falling edge is specified with SCxCR<SCLKS>.) ・If data exists in both the receive shift register and the receive buffer, a receive interrupt occurs when the buffer is read.

Note: Interrupts are not generated when an over-run error is occurred.

12.12.1.2 FIFO

When the FIFO is used, a receive interrupt occurs on depending on the timing described in Table 12-8 and the condition specified with SCxRFC<RFIS>.

Table 12-8 Receive Interrupt Conditions in use of FIFO

SCxRFC<RFIS>	Interrupt conditions	Interrupt generation timing
"0"	When FIFO fill level (SCxRST<RLVL[2:0]>) = Receive FIFO fill level to generate receive interrupt <RIL[1:0]>	・When transfer a received data from receive buffer to receive FIFO ・When read a receive data from receive FIFO
"1"	When FIFO fill level (SCxRST<RLVL[2:0]>) ≥ Receive FIFO fill level to generate receive interrupt <RIL[1:0]>	・When read a receive data from receive FIFO

12.12.2 Transmit interrupts

Figure 12-14 shows the data flow of transmit operation and the route of read.

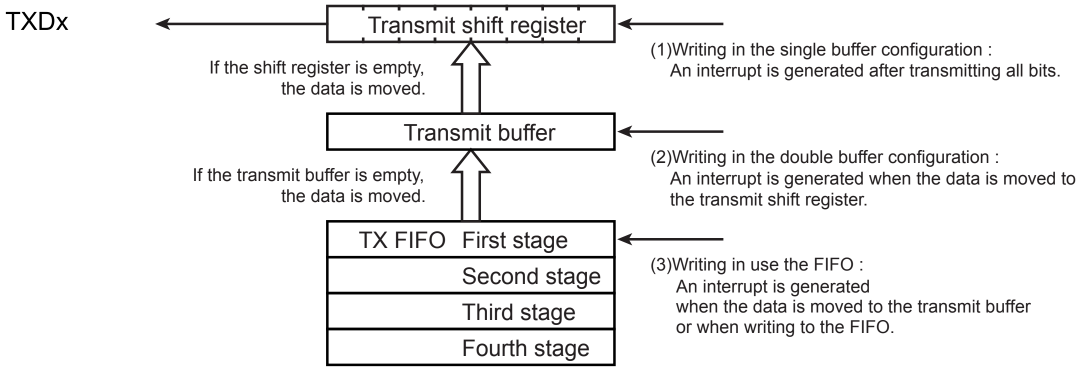

TXDx
Transmit shift register
If the shift register is empty,
the data is moved.
Transmit buffer
If the transmit buffer is empty,
the data is moved.
TX FIFO First stage
Second stage
Third stage
Fourth stage
(1)Writing in the single buffer configuration :
An interrupt is generated after transmitting all bits.
(2)Writing in the double buffer configuration :
An interrupt is generated when the data is moved to
the transmit shift register.
(3)Writing in use the FIFO :
An interrupt is generated
when the data is moved to the transmit buffer
or when writing to the FIFO.

Figure 12-14 Transmit Buffer / FIFO Configuration Diagram

12.12.2.1 Singe Buffer / Double Buffer

Transmit interrupts are generated at the time depends on the transfer mode and the buffer configurations, which are given as follows.

Table 12-9 Transmit Interrupt conditions in use of Single Buffer/Double Buffer

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 SCLKx pin (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. When SCxMOD1<TXE> is "1" and the transmit shift register is empty, if data is transferred to the transmit shift register from the transmit buffer immediately after the data is written to the transmit buffer, a transmit interrupt occurs.	

12.12.2.2 FIFO

When the FIFO is used, a transmit interrupt occurs depending on the timing described in Table 12-10 and the condition specified with SCxTFC<TFIS>.

Table 12-10 Transmit Interrupt conditions in use of FIFO

SCxTFC<TFIS>	Interrupt condition	Interrupt generation timing
"0"	When FIFO fill level (SCxTST<TLVL[2:0]>) = Transmit FIFO fill level to generate transmit interrupt <TIL[1:0]>	・When transmitted data is transferred from transmit FIFO to transmit buffer ・When transmit data is write into transmit FIFO
"1"	When FIFO fill level (SCxTST<TLVL[2:0]>) ≤ Transmit FIFO fill level to generate transmit interrupt <TIL[1:0]>	・When transmit data is write into transmit FIFO

12.12.3 Error Generation

12.12.3.1 UART Mode

Error	9 bits	7 bits 8 bits 7 bits + Parity 8 bits + Parity
Framing Error over-run Error	Around the center of stop bit	
Parity Error	−	Determination:Around the center of parity bit Flag change:Around the center of stop bit

12.12.3.2 I/O Interface Mode

over-run Error	Immediately after the raising / falling edge of the last SCLKx pin (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Under-run Error	Immediately after the rising or falling edge of the next SCLKx pin. (Rising or falling is determined according to SCxCR<SCLKS> setting.)

Note: Over-run error and Under-run error have no meaning in clock output mode.

12.13 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 and the transmit circuit become initial state. Other states are maintained.

12.14 Operation in Each Mode

12.14.1 Mode 0 (I/O interface mode)

The I/O interface mode is selected by setting SCxMOD<SM[1:0]> to "00".

Mode 0 consists of two modes, the clock output mode to output synchronous clock (SCLK) and the clock input mode to accept synchronous clock (SCLK) from an external source.

The operation with disabling a FIFO in each mode is described below. Regarding a FIFO, refer to a receive FIFO and a transmit FIFO which are described before.

12.14.1.1 Transmit

(1) Clock Output Mode

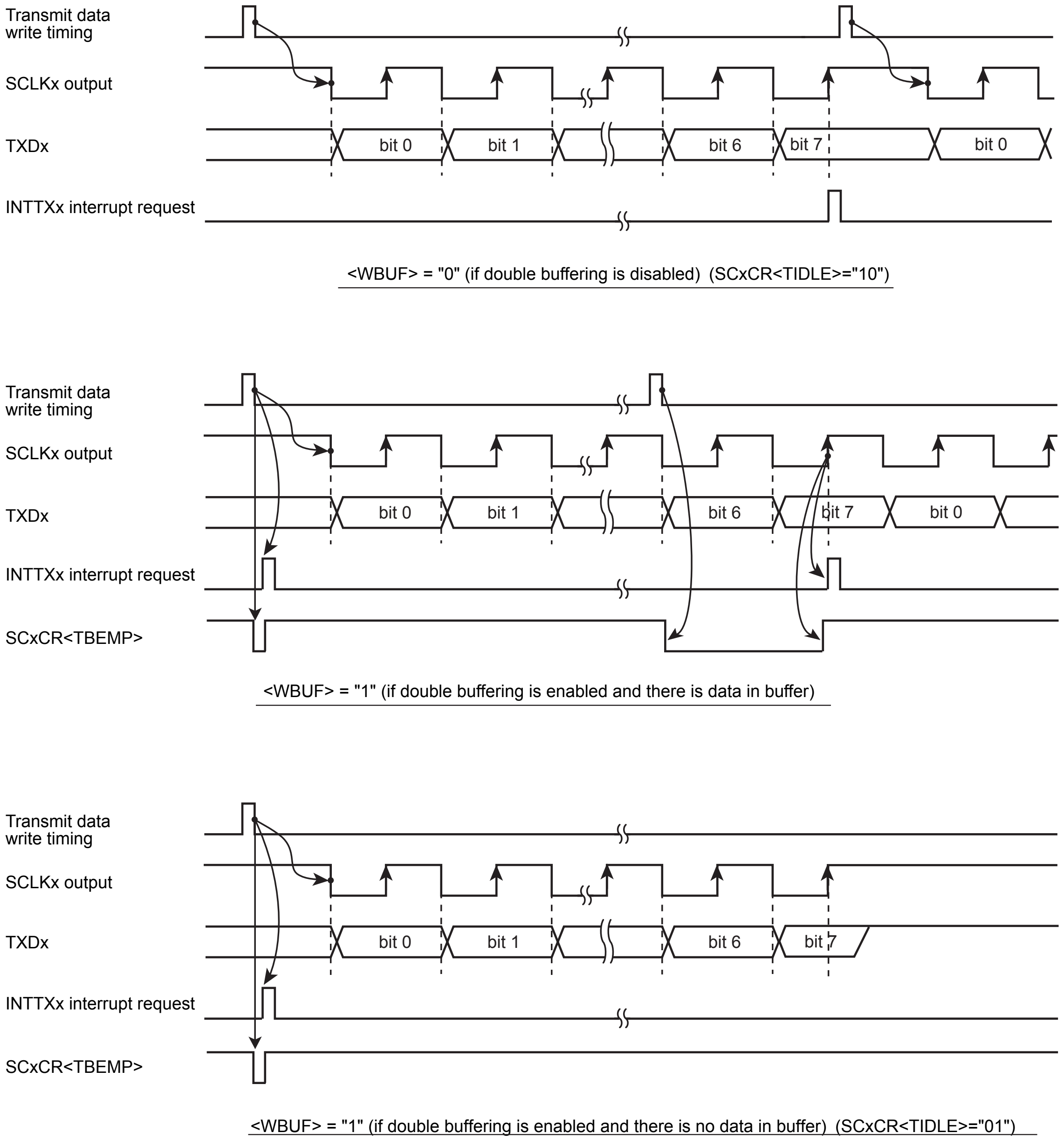
- If the transmit double buffer is disabled (SCxMOD2<WBUF> = "0")

  Data is output from the TXDx pin and the clock is output from the SCLKx 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 in the shift register is empty or when data transmission from the transmit buffer (shift register) is completed. Simultaneously, SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

  If the transmit buffer has no data to be moved to the transmit shift register, INTTXx interrupt is not generated and the clock output stops.



<WBUF> = "0" (if double buffering is disabled) (SCxCR<TIDLE>="10")

<WBUF> = "1" (if double buffering is enabled and there is data in buffer)

<WBUF> = "1" (if double buffering is enabled and there is no data in buffer) (SCxCR<TIDLE>="01")

Figure 12-15 Transmit Operation in the I/O Interface Mode (Clock Output Mode)

(2) Clock Input Mode

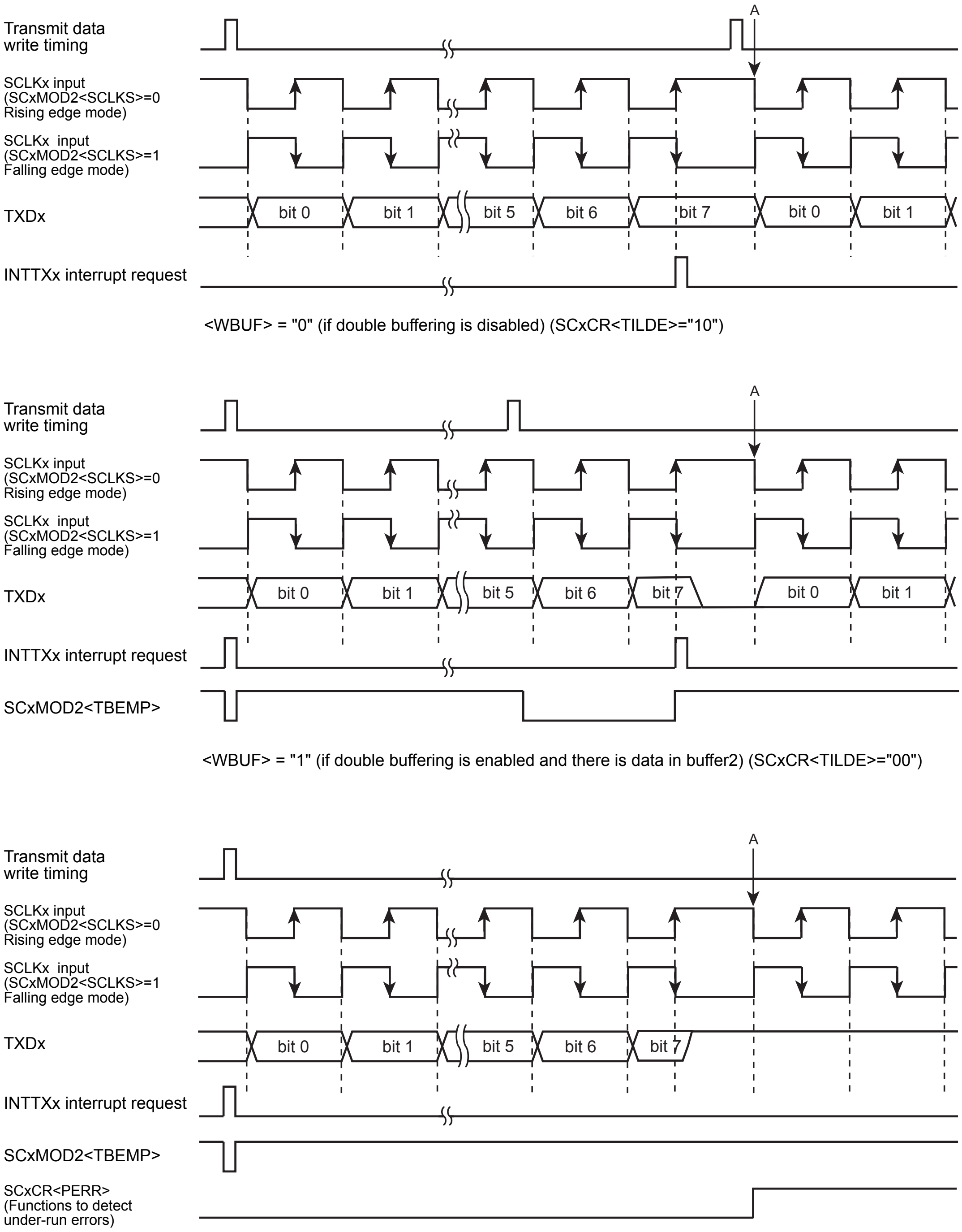
- If double buffering is disabled (SCxMOD2<WBUF> = "0")

  If the clock is input in the condition where data is written in the transmit buffer, 8-bit data is output from the TXDx pin. When all data is output, an interrupt INTTXx is generated. The next transmit data must be written before the timing of point "A" as shown in Figure 12-16.

- 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 clock input becomes active or when data transmission from the transmit shift register is completed. Simultaneously, SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

  If the clock input becomes active while no data is in the transmit buffer, although the internal bit counter is started, an under-run error occurs and the level which is specified by SCxCR<TXDEMP> is output to TXDx pin.


A
Transmit data write timing
SCLKx input (SCxMOD2<SCLKS>=0 Rising edge mode)
SCLKx input (SCxMOD2<SCLKS>=1 Falling edge mode)
TXDx
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
INTTx interrupt request
A
Transmit data write timing
SCLKx input (SCxMOD2<SCLKS>=0 Rising edge mode)
SCLKx input (SCxMOD2<SCLKS>=1 Falling edge mode)
TXDx
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
INTTXx interrupt request
SCxMOD2<TBEMP>
A
Transmit data write timing
SCLKx input (SCxMOD2<SCLKS>=0 Rising edge mode)
SCLKx input (SCxMOD2<SCLKS>=1 Falling edge mode)
TXDx
bit 0
bit 1
bit 5
bit 6
bit 7
INTTXx interrupt request
SCxMOD2<TBEMP>
SCxCR<PERR> (Functions to detect under-run errors)

<WBUF> = "0" (if double buffering is disabled) (SCxCR<TILDE>="10")

<WBUF> = "1" (if double buffering is enabled and there is data in buffer2) (SCxCR<TILDE>="00")

<WBUF> = "1" (if double buffering is enabled and there is no data in buffer2) (SCxCR<TXDEMP><TILDE>="100")

Figure 12-16 Transmit Operation in the I/O Interface Mode (Clock Input Mode)

12.14.1.2 Receive

(1) Clock Output Mode

The clock output starts by setting the receive enable bit SCxMOD0<RXE> to "1".

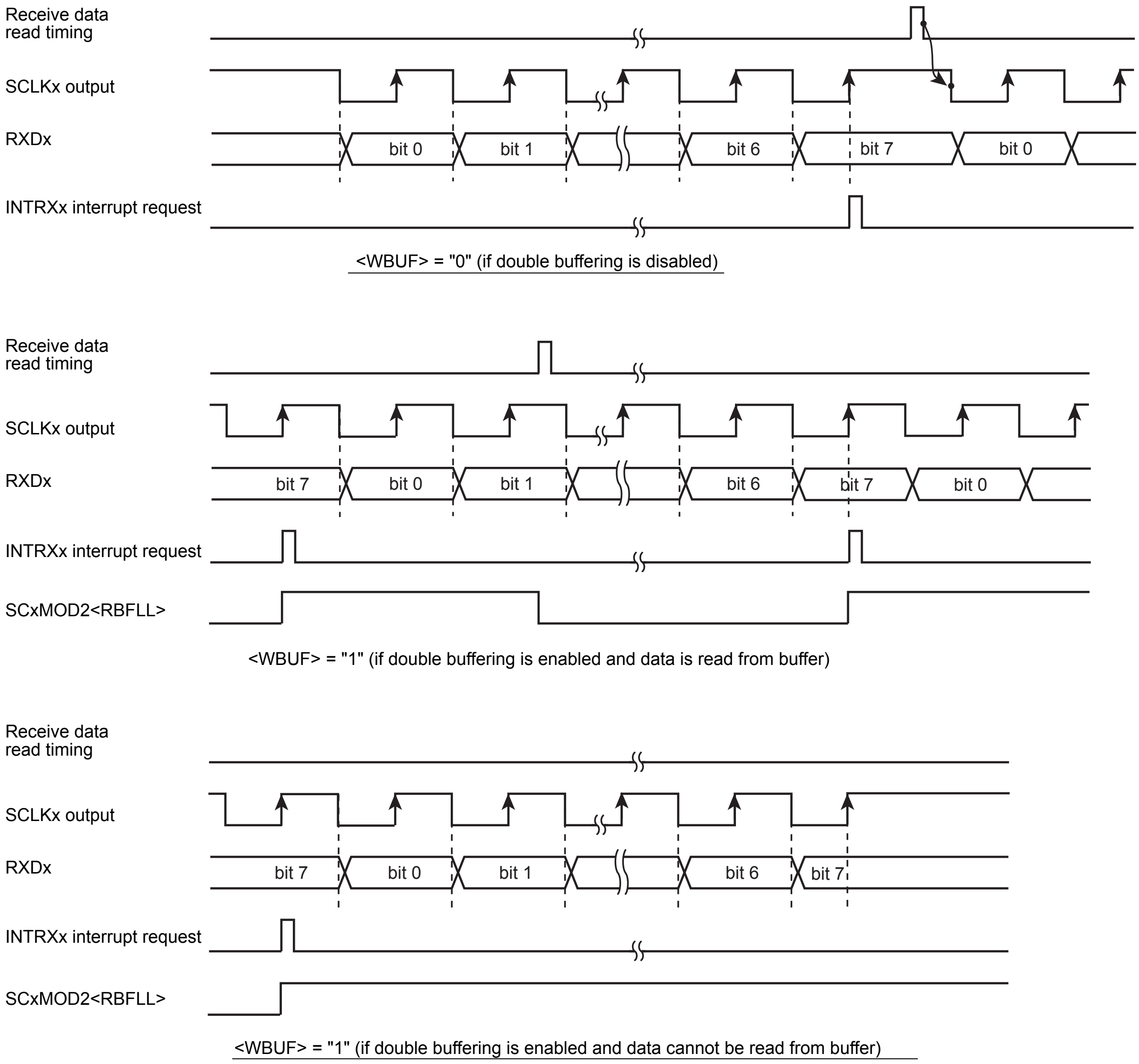
- If double buffer is disabled (SCxMOD2<WBUF> = "0")

  A clock is output from the SCLKx 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, SCxMOD2<RBFLL> is set to "1" and the INTRXx is generated.

  When a data is in the receive buffer, if the data is not read from the receive buffer before completing reception of the next 8 bits, the INTRXx interrupt is not generated and the clock 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.



<WBUF> = "0" (if double buffering is disabled)

<WBUF> = "1" (if double buffering is enabled and data is read from buffer)

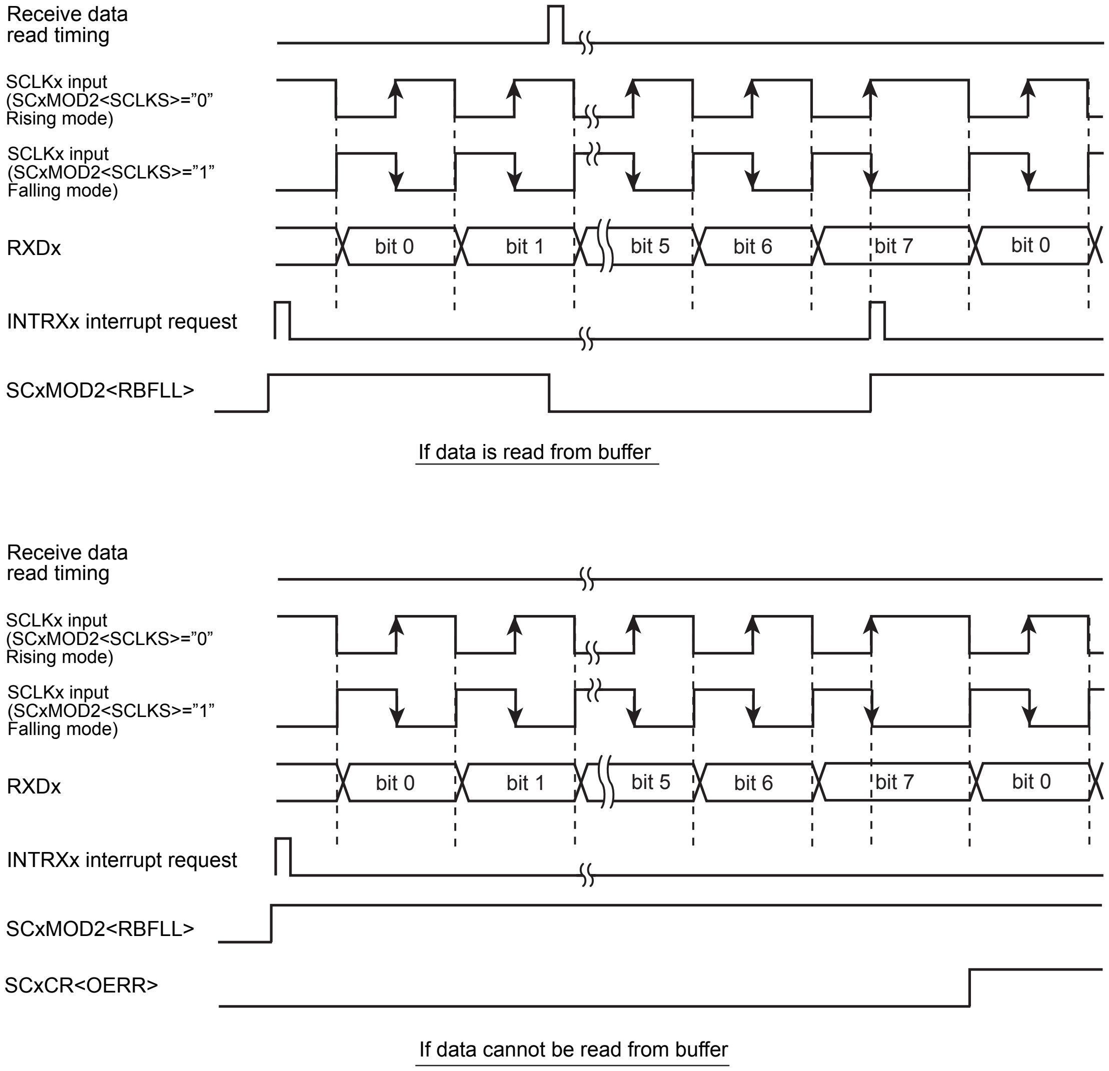
<WBUF> = "1" (if double buffering is enabled and data cannot be read from buffer)

Figure 12-17 Receive Operation in the I/O Interface Mode (Clock Output Mode)

(2) clock input mode

In the clock input mode, receiving double buffering is always enabled, the received data can be moved to the receive buffer from the shift register, and the receive buffer can receive the next data successively.

The INTRXx receive interrupt is generated each time received data is moved to the receive buffer.



If data is read from buffer

If data cannot be read from buffer

Figure 12-18 Receive Operation in the I/O Interface Mode (Clock Input Mode)

12.14.1.3 Transmit and Receive (Full-duplex)

(1) Clock Output Mode

- If double buffers are disabled (SCxMOD2<WBUF> = "0")

  Clock is output when the CPU writes data to the transmit buffer.

  Subsequently, a data is shifted into receive buffer and the INTRXx is generated. Concurrently, a data written to the transmit buffer is output from the TXDx pin, the INTTXx is generated when transmission of all data has been completed. Then, the clock output stops.

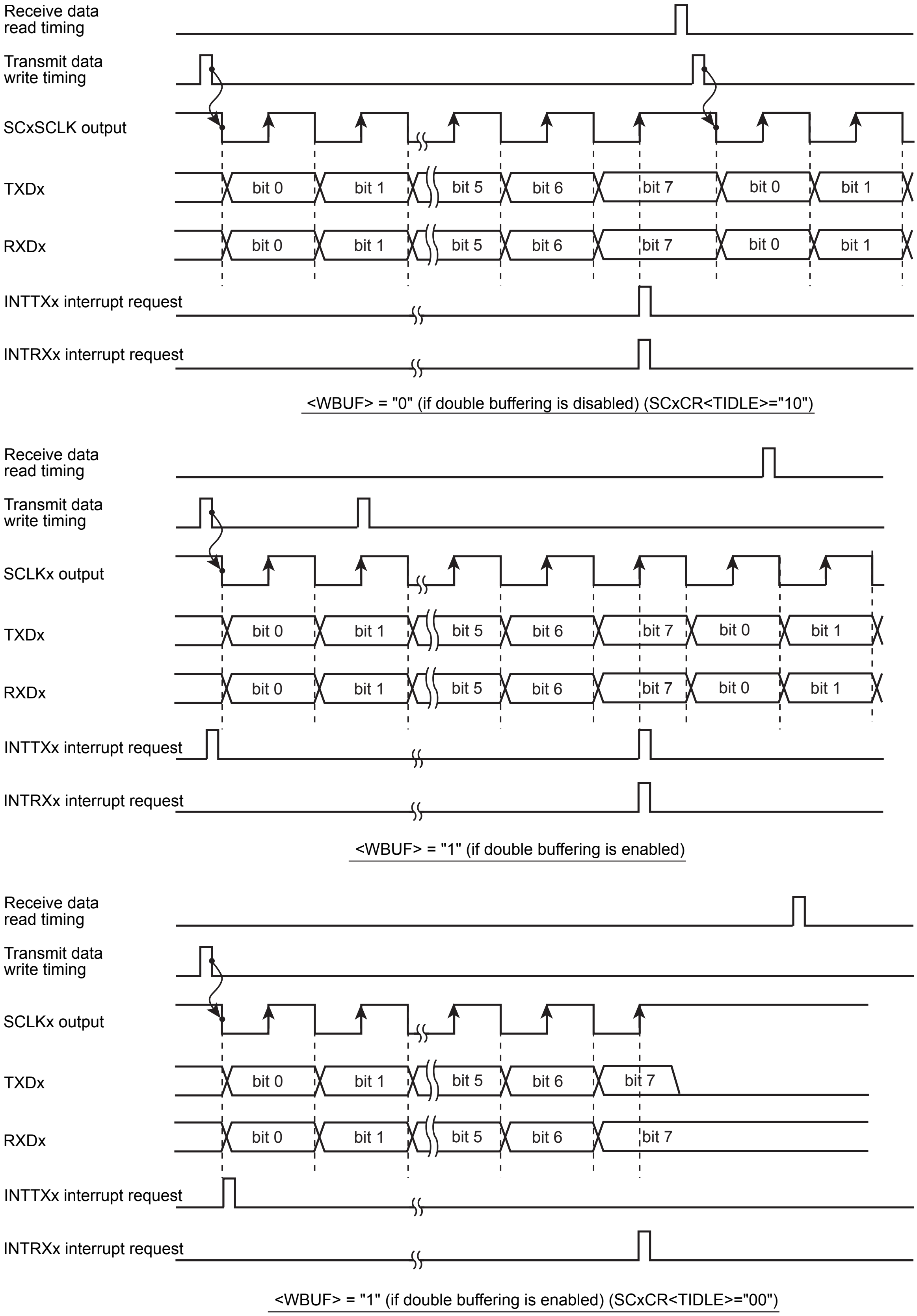
  The next round of data transmission and reception starts when the data is read from the receive buffer and the next transmit data is written to the transmit buffer by the CPU. The order of reading the receive buffer and writing to the transmit buffer can be freely determined. Data transmission is resumed only when both conditions are satisfied.

- If double buffers are enabled (SCxMOD2<WBUF> = "1")

  Clock is outputted when the CPU writes data to the transmit buffer.

  A data is shifted into the receive shift register, moved to the receive buffer, and the INTRXx is generated. While a data is received, a transmit data is output from the TXDx pin. When all data are sent out, the INTTXx 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<RBFLL> = "1"), the clock output stops. When both conditions, receive data is read and transmit data is written, are satisfied, the clock output is resumed and the next round of data transmission and reception is started.



<WBUF> = "0" (if double buffering is disabled) (SCxCR<TIDLE>="10")

<WBUF> = "1" (if double buffering is enabled)

<WBUF> = "1" (if double buffering is enabled) (SCxCR<TIDLE>="00")

Figure 12-19 Transmit/Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) Clock Input Mode

- If double buffers are disabled. (SCxMOD2<WBUF> = "0")

  When receiving data, double buffer is always enabled regardless of the SCxMOD2 <WBUF> settings.

  A data written in the transmit buffer is outputted from the TXDx pin and a data is shifted into the receive buffer when the clock input becomes active. The INTTXx is generated upon completion of data transmission. The INTRXx 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 clock input for the next data (data must be written before the point A in Figure 12-20). Data must be read before completing reception of the next data.

- If double buffers are enabled. (SCxMOD2<WBUF> = "1")

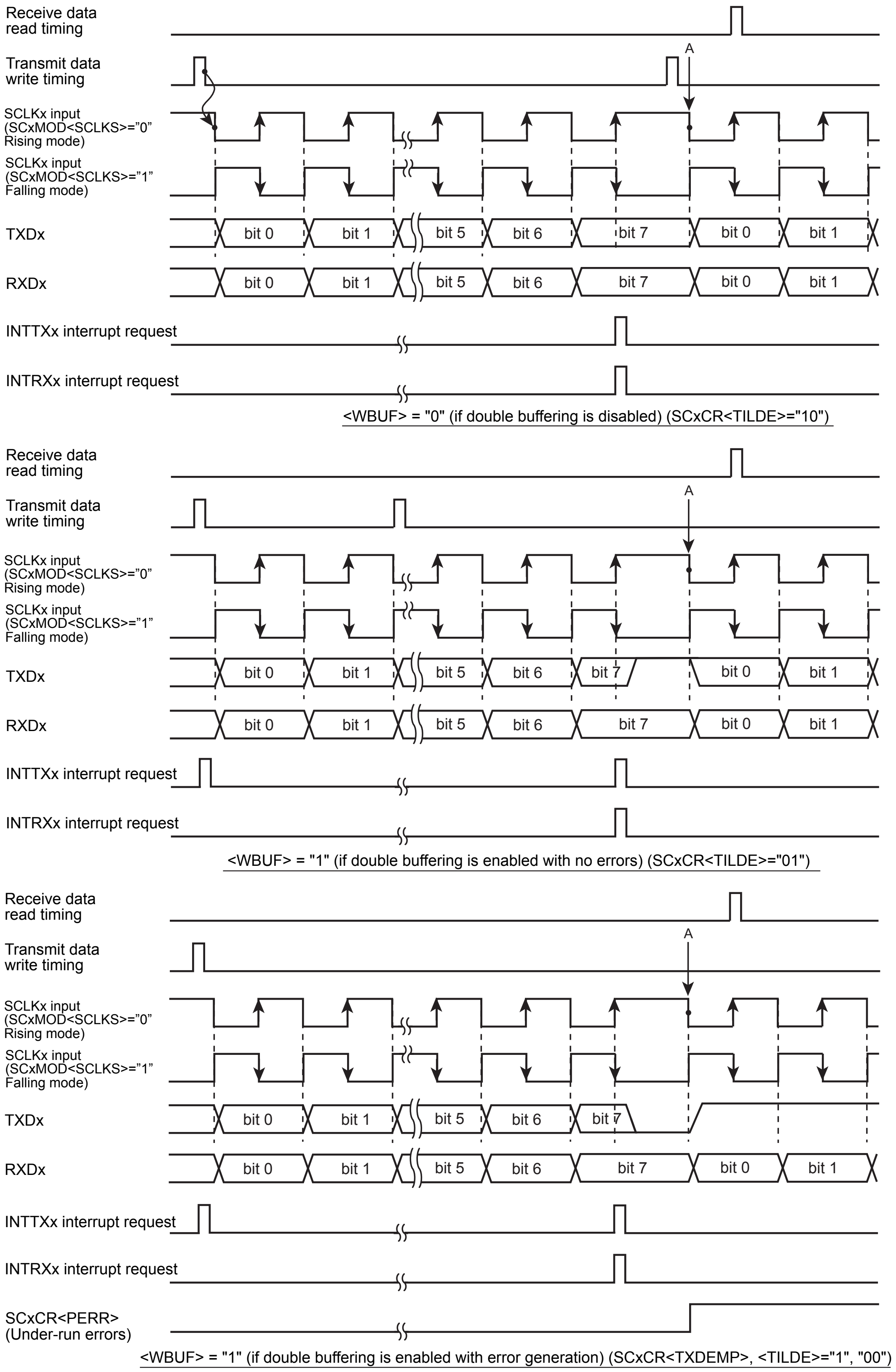
  The INTTXx is generated at the timing the transmit buffer data is moved to the transmit shift register after completing data transmission from the transmit shift register. At the same time, data received is shifted to the shift register, it is moved to the receive buffer, and the INTRXx is generated.

  Note that transmit data must be written into the transmit buffer before the clock input for the next data (data must be written before the point A in Figure 12-20). Data must be read before completing reception of the next data.

  Upon the clock input for the next data, 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 data is received, an overrun error occurs.

  If there is no data written to transmit buffer when clock for the next data is input, an underrun error occurs. The level which is specified by SCxCR<TXDEMP> is output to TXDx pin.



<WBUF> = "0" (if double buffering is disabled) (SCxCR<TILDE>="10")

<WBUF> = "1" (if double buffering is enabled with no errors) (SCxCR<TILDE>="01")

<WBUF> = "1" (if double buffering is enabled with error generation) (SCxCR<TXDEMP>, <TILDE>="1", "00")

Figure 12-20 Transmit/Receive Operation in the I/O Interface Mode (Clock Input Mode)

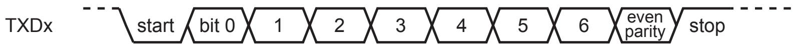
12.14.2 Mode 1 (7-bit UART mode)

The 7-bit UART mode is selected by setting SCxMOD<SM[1:0]> to "01".

In this mode, parity bits can be added to the transmit data stream; 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>. 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.



(Transmission rate of 2400 bps @fsys = 9.8304 MHz)

Clocking conditions	system clock:	High-speed (fc)
	High-speed clock gear:	x 1 (fc)
	Prescaler clock:	fperiph/2 (fperiph = fsys)

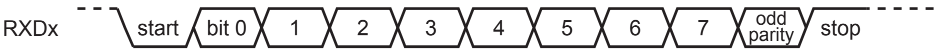
		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	-	0	0	1	0	1	Set 7-bit UART mode
SCxCR	←	x	1	1	x	x	x	0	0	Even parity enabled
SCxBRCR	←	0	0	1	0	0	1	0	0	Set 2400bps
SCxBUF	←	*	*	*	*	*	*	*	*	Set transmit data

x: don’t care - : no change

12.14.3 Mode 2 (8-bit UART mode)

The 8-bit UART mode is 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:



(Reception rate of 9600 bps @fc = 9.8304 MHz)

Clocking conditions	System clock:	High-speed (fc)
	High-speed clock gear:	x 1 (fc)
	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

12.14.4 Mode 3 (9-bit UART mode)

The 9-bit UART mode is 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 SCxMOD0<TB8> for transmitting data. The data is stored in SCxCR<RB8> for receiving data.

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>.

12.14.4.1 Wakeup function

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting SCxMOD0<WU> to "1".

In this case, the interrupt INTRXx will be generated only when SCxCR<RB8> is set to "1".

Note: The TXDx pin of the slave controller must be set to the open drain output mode using the PxOD.

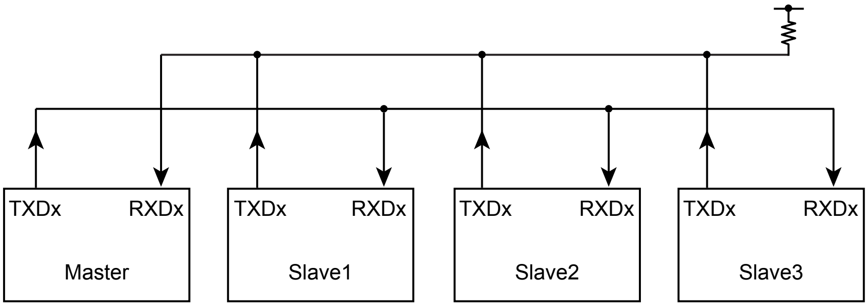


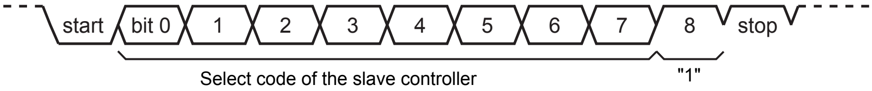
Figure 12-21 Serial Links to Use Wake-up Function

12.14.4.2 Protocol

1. Select the 9-bit UART mode for the master and slave controllers.

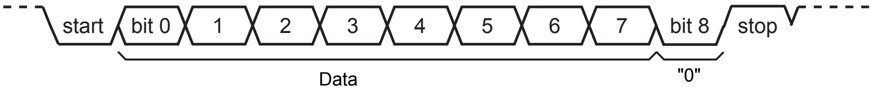
2. Set SCxMOD<WU> to "1" for the slave controllers to make them ready to receive data.

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> to "0".

5. The master controller transmits data to the designated slave controller (the controller of which SCxMOD<WU> 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> 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> set to "0" can transmit data to the master controller to inform that the data has been successfully received.

13. Serial Bus Interface (I2C/SIO)

The TMPM381/383 contains 1 Serial Bus Interface (I2C/SIO) channel, 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 13-1 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	SCL :PC1 SDA :PC0	PCFR3[1:0] = 11	PCCR[1:0] = 11	PCIE[1:0] = 11	PCOD[1:0] = 11
	SIO mode	SCK :PC2 SI :PC1 SO :PC0	PCFR3[2:0] = 111	PCCR[2:0] = 101(SCK0 output) PCCR[2:0] = 001(SCK0 input)	PCIE[2:0] = 010(SCK0 output) PCIE[2:0] = 110(SCK0 input)	PCOD[2:0] = xxx

13.1 Configuration

The configuration is shown in Figure 13-1.

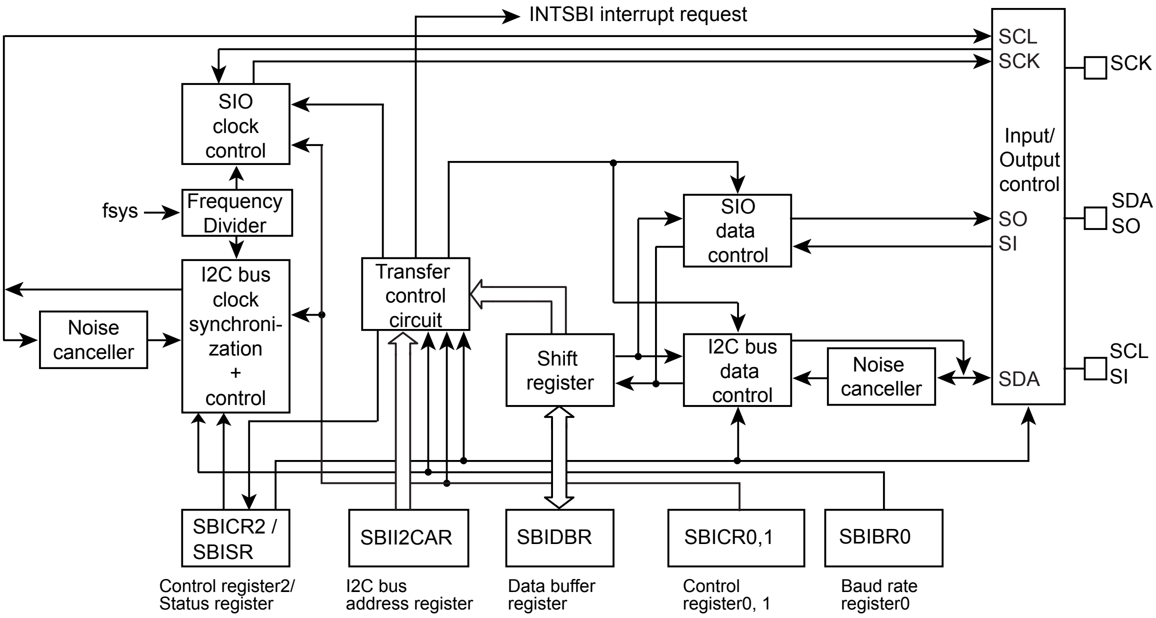


Figure 13-1 (I2C/SIO) Block Interface

13.2 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 "13.4 Control Registers in the I2C Bus Mode" and "13.7 Control register of SIO mode".

13.2.1 Registers for each channel

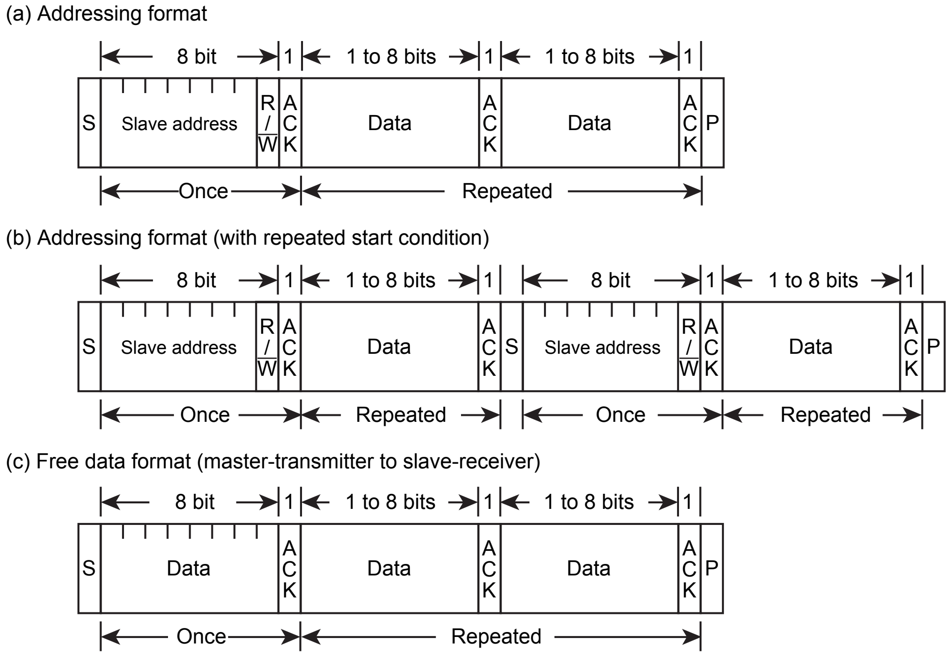
The tables below show the registers and register addresses for each channel.

Base Address = 0x4002_0000

Register name		Address(Base+)
Control register 0	SBICR0	0x0000
Control register 1	SBICR1	0x0004
Data buffer register	SBIDBR	0x0008
I2C bus address register	SBII2CAR	0x000C
Control register 2	SBICR2 (writing)	0x0010
Status register	SBISR (reading)	
Baud rate register 0	SBIBR0	0x0014

13.3 I2C Bus Mode Data Format

Figure 13-2 shows the data formats used in the I2C bus mode.



(a) Addressing format

(b) Addressing format (with repeated start condition)

(c) Free data format (master-transmitter to slave-receiver)

Note) S : Start condition
R/W : Direction bit
ACK : Acknowledge bit
P : Stop condition

Figure 13-2 I2C Bus Mode Data Formats

13.4 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.

13.4.1 SBICR0(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. For the first time in case of setting to enable, the relevant SBI registers can be read or written. Since all clocks except SBICR0 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.

Note: To use the serial bus interface, enable this bit first.

13.4.2 SBICR1(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)

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-8</td><td>-</td><td>R</td><td>Read as 0.</td></tr>
<tr><td>7-5</td><td>BC[2:0]</td><td>R/W</td><td>Select the number of bits per transfer (Note 1)
<table>
<tr><th rowspan="2">&lt;BC&gt;</th><th colspan="2">When &lt;ACK&gt; = 0</th><th colspan="2">When &lt;ACK&gt; = 1</th></tr>
<tr><th>Number of clock cycles</th><th>Data length</th><th>Number of clock cycles</th><th>Data length</th></tr>
<tr><td>000</td><td>8</td><td>8</td><td>9</td><td>8</td></tr>
<tr><td>001</td><td>1</td><td>1</td><td>2</td><td>1</td></tr>
<tr><td>010</td><td>2</td><td>2</td><td>3</td><td>2</td></tr>
<tr><td>011</td><td>3</td><td>3</td><td>4</td><td>3</td></tr>
<tr><td>100</td><td>4</td><td>4</td><td>5</td><td>4</td></tr>
<tr><td>101</td><td>5</td><td>5</td><td>6</td><td>5</td></tr>
<tr><td>110</td><td>6</td><td>6</td><td>7</td><td>6</td></tr>
<tr><td>111</td><td>7</td><td>7</td><td>8</td><td>7</td></tr>
</table></td></tr>
<tr><td>4</td><td>ACK</td><td>R/W</td><td>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.</td></tr>
<tr><td>3</td><td>-</td><td>R</td><td>Read as 1.</td></tr>
<tr><td>2-1</td><td>SCK[2:1]</td><td>R/W</td><td rowspan="2">Select internal SCL output clock frequency (Note 2).
<table>
<tr><td>000</td><td>n = 5</td><td>385 kHz</td><td rowspan="8">System Clock: fsys
(= 40MHz)
Clock gear : fc/1
Frequency = $\frac{fsys}{2^n + 72}$ [Hz]</td></tr>
<tr><td>001</td><td>n = 6</td><td>294 kHz</td></tr>
<tr><td>010</td><td>n = 7</td><td>200 kHz</td></tr>
<tr><td>011</td><td>n = 8</td><td>122 kHz</td></tr>
<tr><td>100</td><td>n = 9</td><td>68 kHz</td></tr>
<tr><td>101</td><td>n = 10</td><td>36 kHz</td></tr>
<tr><td>110</td><td>n = 11</td><td>19 kHz</td></tr>
<tr><td>111</td><td></td><td>reserved</td></tr>
</table></td></tr>
<tr><td rowspan="2">0</td><td>SCK[0]</td><td>W</td></tr>
<tr><td>SWRMON</td><td>R</td><td>On reading <SWRMON>: Software reset status monitor
0:Software reset operation is in progress.
1:Software reset operation is not in progress.</td></tr>
</table>

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 "13.5.1 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 SBICR2 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.**

Note 5: **When <BC[2:0]>="001" and <ACK>="0" in master mode, SCL line may be fixed to "L" by falling edge of SCL line after generation of STOP condition and the other master devices can not use the bus. In the case of bus which is connected with several master devices, the number of bits per transfer should be set equal or more than 2 before generation of STOP condition.**

13.4.3 SBICR2(Control register 2)

This register serves as SBISR 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 INTSBI 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. For details, refer to "13.5.16 Software 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.**

13.4.4 SBISR (Status Register)

This register serves as SBICR2 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	INTSBI 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 address 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"

13.4.5 SBIBR0(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".

13.4.6 SBIDBR (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 SBIBDR 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.**

13.4.7 SBII2CAR (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 SBII2CAR 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.**

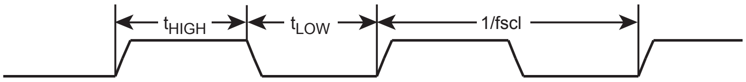
Note 2: **Do not set SBII2CAR to "0x00" in slave mode. (If SBII2CAR is set to "0x00", it’s recognized that the slave address matches the START byte ("0x01") of the I2C standard received in slave mode.)**

13.5 Control in the I2C Bus Mode

13.5.1 Serial Clock

13.5.1.1 Clock source

SBICR1<SCK[2:0]> specifies the maximum frequency of the serial clock to be output from the SCL pin in the master mode.



$$t_{LOW} = 2^{n-1}/\text{fsys} + 58/\text{fsys}$$
$$t_{HIGH} = 2^{n-1}/\text{fsys} + 14/\text{fsys}$$
$$\text{fscl} = 1/(t_{LOW} + t_{HIGH})$$
$$= \frac{\text{fsys}}{2^n + 72}$$

SBICR1<SCK[2:0]>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Figure 13-3 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.**

13.5.1.2 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.

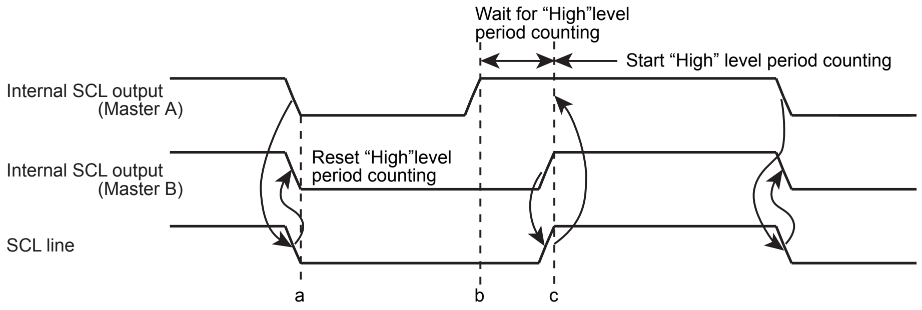

Wait for "High"level period counting
Start "High" level period counting
Internal SCL output (Master A)
Internal SCL output (Master B)
Reset "High"level period counting
SCL line
a
b
c

Figure 13-4 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.

13.5.2 Setting the Acknowledgement Mode

Setting SBICR1<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. However, the second byte of the general call is necessary to be controlled by software to generate an acknowledgement signal depending on the contents of the second byte.

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.

13.5.3 Setting the Number of Bits per Transfer

SBICR1<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.

13.5.4 Slave Addressing and Address Recognition Mode

Setting "0" to SBII2CAR<ALS> and a slave address in SBII2CAR<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.

13.5.5 Operating mode

The setting of SBICR2<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.

13.5.6 Configuring the SBI as a Transmitter or a Receiver

Setting SBICR2<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 SBII2CAR.
- 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/$\overline{\text{W}}$) 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.

13.5.7 Configuring the SBI as a Master or a Slave

Setting SBICR2<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.

13.5.8 Generating Start and Stop Conditions

When SBISR<BB> is "0", writing "1" to SBICR2<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.

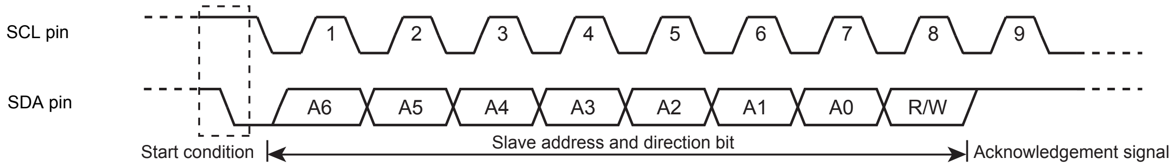


Figure 13-5 Generating the Start Condition and a Slave Address

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.

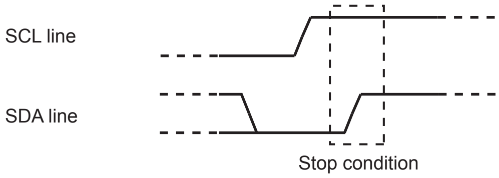


Figure 13-6 Generating the Stop Condition

SBISR<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).

13.5.9 Interrupt Service Request and Release

In master mode, a serial bus interface request (INTSBI) is generated when the transfer of the number of clock cycles set by <BC> and <ACK> is completed.

In slave mode, INTSBI 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 SBII2CAR<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"), INTSBI is generated when the received slave address matches the values specified at SBII2CAR or when a general-call (eight bits data following the start condition is all "0") is received.

When an interrupt request (INTSBI) is generated, SBICR2<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 SBIDBR. 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 occurs while a slave address and direction bit are transferred in the master mode, <PIN> is cleared to "0" and INTSBI occurs. This does not relate to whether a slave address matches <SA>.

13.5.10 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.

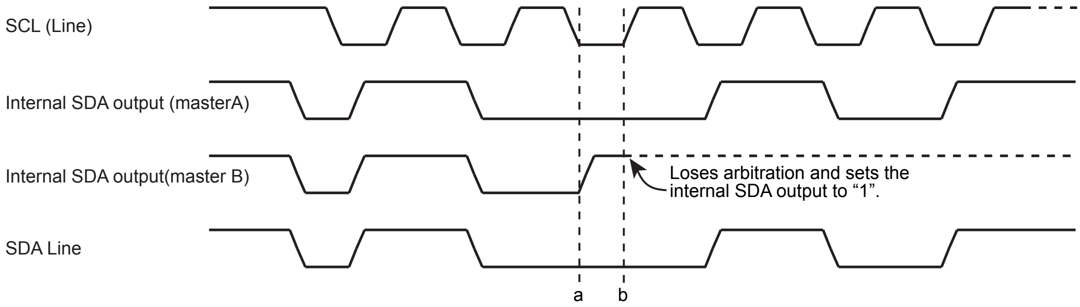

SCL (Line)
Internal SDA output (masterA)
Internal SDA output(master B)
SDA Line
Loses arbitration and sets the internal SDA output to "1".
a
b

Figure 13-7 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 SBISR<AL> is set to "1".

When an arbitration lost occurs, SBIxSR<MST> and <TRX> are cleared to "0", causing the SBI to operate as a slave receiver and it stops the clock output during data transfer.If the master device which sends a slave address and direction bit generates Arbitration lost, it receives a slave address and direction bit which are sent by other master devices as slave device.Regardless of whether a received slave address matches <SA>, <PIN> is cleared to "0" and INTSBI occurs.

<AL> is cleared to "0" when data is written to or read from SBIDBR or data is written to SBICR2.

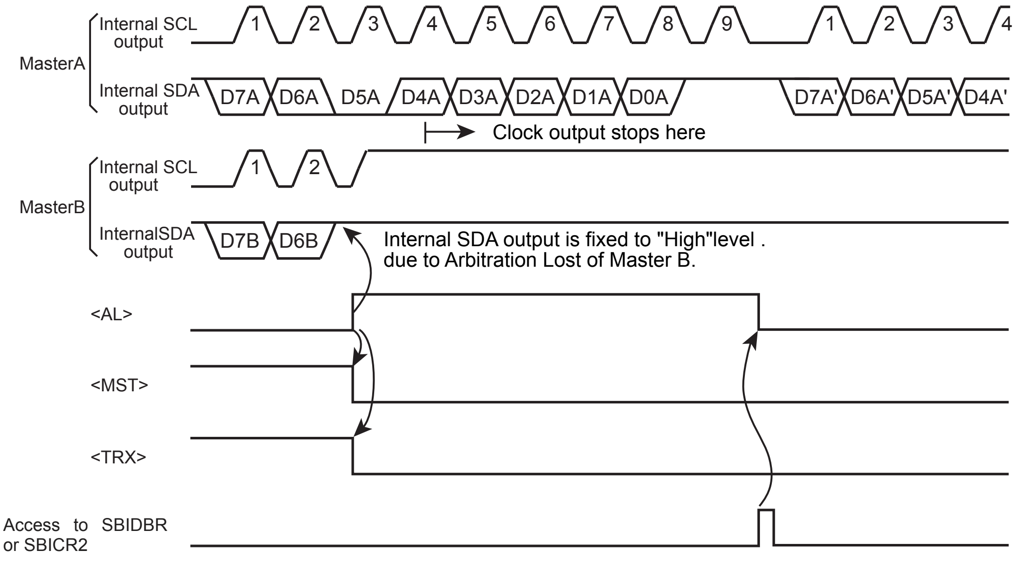


Figure 13-8 Example of Master B Lost Arbitration (D7A = D7B, D6A = D6B)

13.5.11 Slave Address Match Detection Monitor

When the SBI operates as a slave device in the address recognition mode (SBII2CAR<ALS>="0"), SBISR<AAS> is set to "1" on receiving the general-call address or the slave address that matches the value specified at SBII2CAR.

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 SBIDBR.

13.5.12 General-call Detection Monitor

When the SBI operates as a slave device, SBISR<ADO> is set to "1" when it receives the general-call address; i.e., the eight bits following the start condition are all zeros.

<ADO> is cleared to "0" when the start or stop condition is detected on the bus.

13.5.13 Last Received Bit Monitor

SBISR<LRB> is set to the SDA line value that was read at the rising of the SCL line.

In the acknowledgment mode, reading SBISR<LRB> immediately after generation of the INTSBI interrupt request causes ACK signal to be read.

13.5.14 Data Buffer Register (SBIDBR)

Reading or writing SBIDBR 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.

13.5.15 Baud Rate Register (SBIBR0)

The SBIBR0<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.

13.5.16 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 SBICR2<SWRST[1:0]> generates a reset signal that initializes the serial bus interface circuit. When writing SBIxCR2<SWRST[1:0]>, set SBIxCR2<MST><TRX><BB><PIN> to "0000" and SBIxCR2<SBIM[1:0]> to "10" for I2C bus mode. 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.

13.6 Data Transfer Procedure in the I2C Bus Mode

13.6.1 Device Initialization

Firstly, set SBICR1<ACK><SCK[2:0]>. Set "1" to <ACK> to specify the acknowledgement mode. Set "000" to SBICR1<BC[2:0]> .

Secondly, set <SA[6:0]> (a slave address) and <ALS> to SBII2CAR . (In the addressing format mode, set <ALS>="0").

Finally, to configure the Serial Bus Interface as a slave receiver, ensure that the serial bus interface pin is at "High" first. Then write "000" to SBICR2<MST><TRX><BB>, "1" to <PIN>, "10" to <SBIM[1:0]> and "00" to <SWRST[1:0]>.

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	
SBICR1	←	0	0	0	1	0	X	X	X	Specifies ACK and SCL clock.
SBII2CAR	←	X	X	X	X	X	X	X	X	Specifies a slave address and an address recognition mode.
SBICR2	←	0	0	0	1	1	0	0	0	Configures the SBI as a slave receiver.

Note: X; Don’t care

13.6.2 Generating the Start Condition and a Slave Address

13.6.2.1 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 SBICR1<ACK> to select the acknowledgment mode. Write to SBIDBR a slave address and a direction bit to be transmitted.

When <BB> = "0", writing "1111" to SBICR2<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 SBIDBR with the first eight clocks, and releases the SDA line in the ninth clock to receive an acknowledgment signal from the slave device.

The INTSBI 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 INTSBI 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 SBIDBR. 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.	←	SBISR								
Reg.	←	Reg. e 0x20								
if Reg.	≠	0x00								Ensures that the bus is free.
Then										
SBICR1	←	X	X	X	1	0	X	X	X	Selects the acknowledgement mode.
SBIDBR	←	X	X	X	X	X	X	X	X	Specifies the desired slave address and direction.
SBICR2	←	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

13.6.2.2 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 SBII2CAR 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 INTSBI 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".

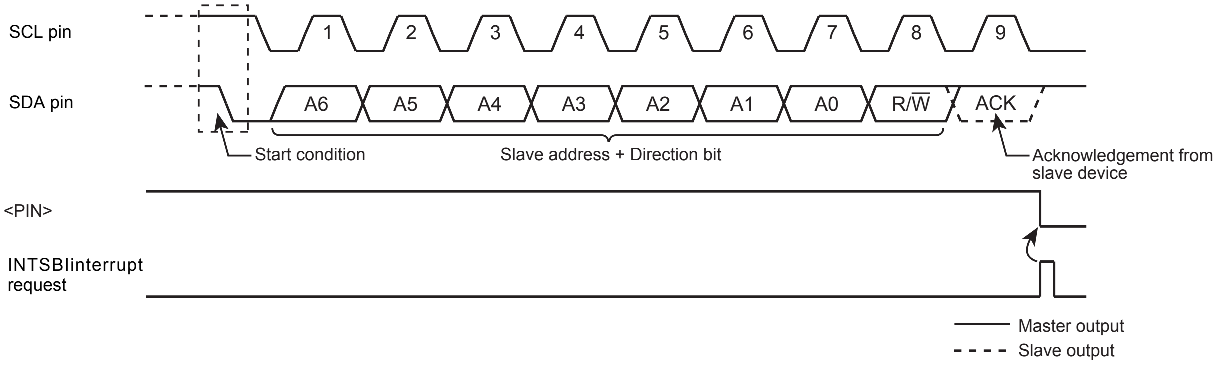


Figure 13-9 Generation of the Start Condition and a Slave Address

13.6.3 Transferring a Data Word

At the end of a data word transfer, the INTSBI interrupt is generated to test <MST> to determine whether the SBI is in the master or slave mode.

13.6.3.1 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 SBIDBR. If the data has different length, <BC[2:0]> and <ACK> are programmed and the transmit data is written into SBIDBR.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 INTSBI 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.

INTSBI 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.

SBICR1	←	X	X	X	X	0	X	X	X	Specifies the number of bits to be transmitted and specify whether ACK is required.
SBIDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

End of interrupt processing.

Note: **X; Don’t care**

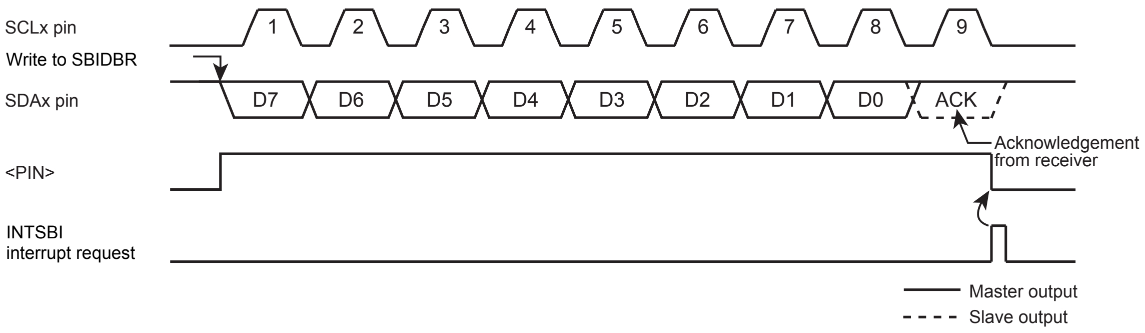


Figure 13-10 <BC[2:0]>= "000",<ACK>= "1" (Transmitter Mode)

(2) Receiver mode (<TRX> = "0")

If the next data to be transmitted has eight bits, the transmit data is written into SBIDBR.

If the data has different length, <BC[2:0]> and <ACK> are programmed and the received data is read from SBIDBR 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 INTSBI 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 SBIDBR, one-word transfer clock and an acknowledgement signal are output.

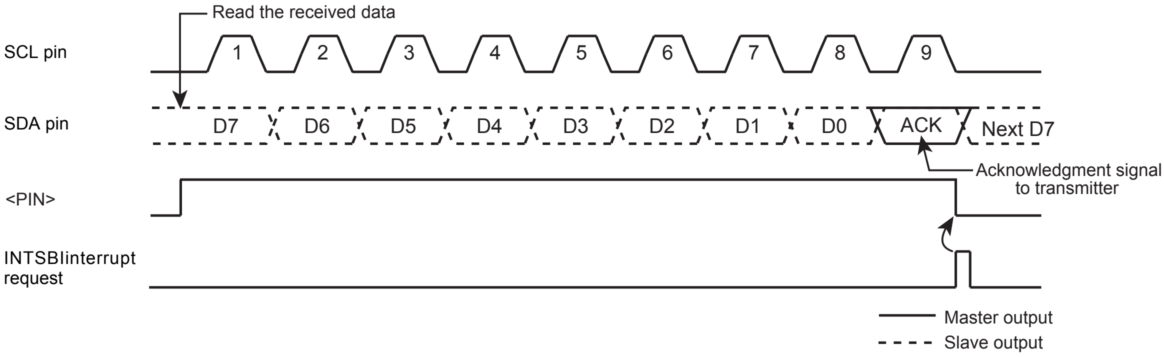


Figure 13-11 <BC[2:0]>= "000",<ACK>= "1" (Receiver Mode)

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.

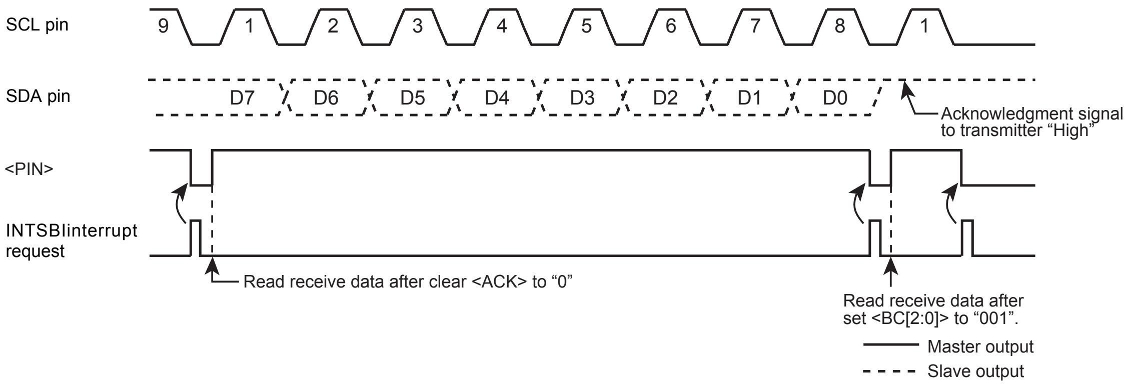

SCL pin
9
1
2
3
4
5
6
7
8
1
SDA pin
D7
D6
D5
D4
D3
D2
D1
D0
Acknowledgment signal to transmitter "High"
<PIN>
INTSBIinterrupt request
Read receive data after clear <ACK> to "0"
Read receive data after set <BC[2:0]> to "001".
Master output
Slave output

Figure 13-12 Terminating Data Transmission in the Master Receiver Mode

Example: When receiving N data word

INTSBI interrupt (after data transmission)

		7	6	5	4	3	2	1	0	
SBICR1	←	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.	←	SBIDBR								Reads dummy data.
End of interrupt										

INTSBI interrupt (first to (N-2)th data reception)

		7	6	5	4	3	2	1	0	
Reg.	←	SBIDBR								Reads the first to (N-2)th data words.
End of interrupt										

INTSBI interrupt ((N-1)th data reception)

		7	6	5	4	3	2	1	0	
SBICR1	←	X	X	X	0	0	X	X	X	Disables generation of acknowledgement clock.
Reg.	←	SBIDBR								Reads the (N-1)th data word.
End of interrupt										

INTSBI 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.	←	SBIDBR								Reads the Nth data word.
End of interrupt										

INTSBI interrupt (after completing data reception)

Processing to generate the stop condition.	Terminates the data transmission.
End of interrupt	

Note: X; Don't care

13.6.3.2 Slave mode (<MST> = "0")

In the slave mode, the SBI generates the INTSBI 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 INTSBI 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 SBIDBR or when <PIN> is set to "1", the SCLx pin is released after a period of t_{LOW}.

However, the second byte of the general call is necessary to be controlled by software to generate an acknowledgement signal depending on the contents of the second byte.

In the slave mode, the normal slave mode processing or the processing as a result of Arbitration Lost is carried out.

SBISR<AL>, <TRX>, <AAS> and <ADO> are tested to determine the processing required.

"Table 13-2 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.

INTSBI 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.
SBICR1     ←  X  X  X  1  0  X  X  X     Sets the number of bits to be transmitted.
SBIDBR     ←  X  X  X  X  X  X  X  X     Sets the transmit data.
```

Note: X; Don’t care

Table 13-2 Processing in Slave Mode

<TRX>	<AL>	<AAS>	<ADO>	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 SBIDBR.
	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 SBIDBR.
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 SBIDBR (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 SBIDBR.

13.6.4 Generating the Stop Condition

When SBISR<BB> is "1", writing "1" to SBICR2<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	
SBICR2	←	1	1	0	1	1	0	0	0	Generates the stop condition.

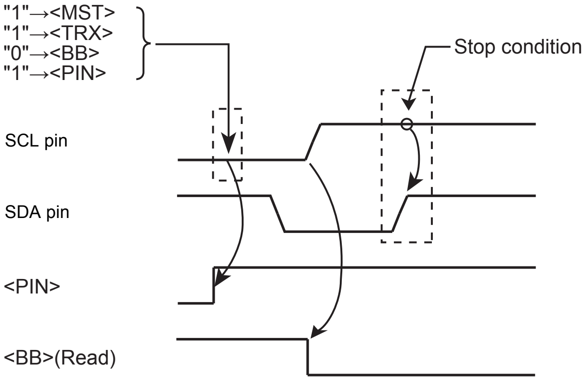


Figure 13-13 Generating the Stop Condition

13.6.5 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 SBICR2<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 SBISR<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 "13.6.2 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
SBICR2             ←      0   0   0   1   1   0   0   0      Releases the bus.
if SBISR<BB> ≠ 0                                             Checks that the SCL pin is released.
Then
if SBISR<LRB> ≠ 1                                            Checks that no other device is pulling the SCL pin to the "Low".
Then
4.7 μs Wait
SBICR1             ←      X   X   X   1   0   X   X   X      Selects the acknowledgment mode.
SBIDBR             ←      X   X   X   X   X   X   X   X      Sets the desired slave address and direction.
SBICR2             ←      1   1   1   1   1   0   0   0      Generates the start condition.
```

Note: X; Don’t care

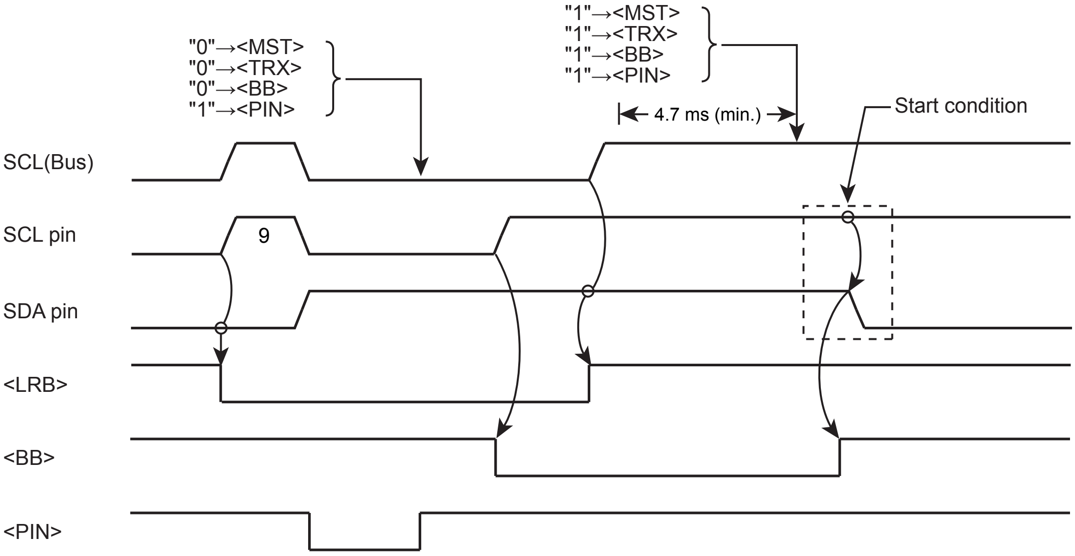


Figure 13-14 Timing Chart of Generating a Restart

13.7 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.

13.7.1 SBICR0(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 SBICR0 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.

13.7.2 SBICR1(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 (=40MHz) 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 SBICR2 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 SBICR2 register and the SBISR register are the same.**

Note 2: **Set <SIOS> to "0" and <SIOINH> to "1" before programming the transfer mode and the serial clock.**

13.7.3 SBIDBR (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 SBIDBR 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.**

13.7.4 SBICR2(Control register 2)

This register serves as SBISR 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.**

13.7.5 SBISR (Status Register)

This register serves as SBICR2 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.**

13.7.6 SBIBR0 (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".

13.8 Control in SIO mode

13.8.1 Serial Clock

13.8.1.1 Clock source

Internal or external clocks can be selected by programming SBICR1<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.

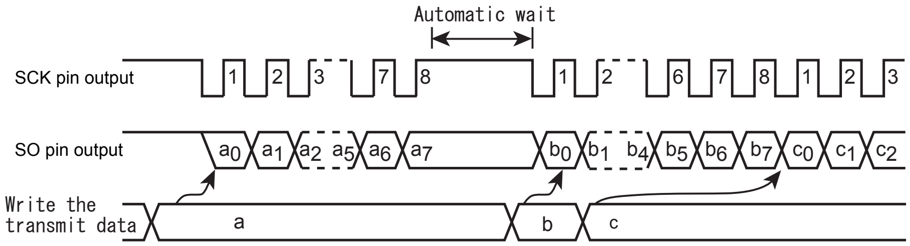


Figure 13-15 Automatic Wait

(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.

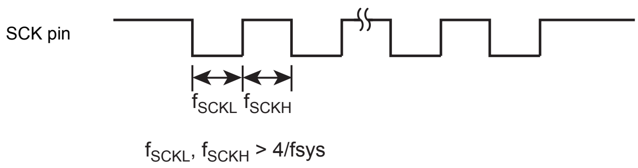


Figure 13-16 Maximum Transfer Frequency of External Clock Input

13.8.1.2 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).

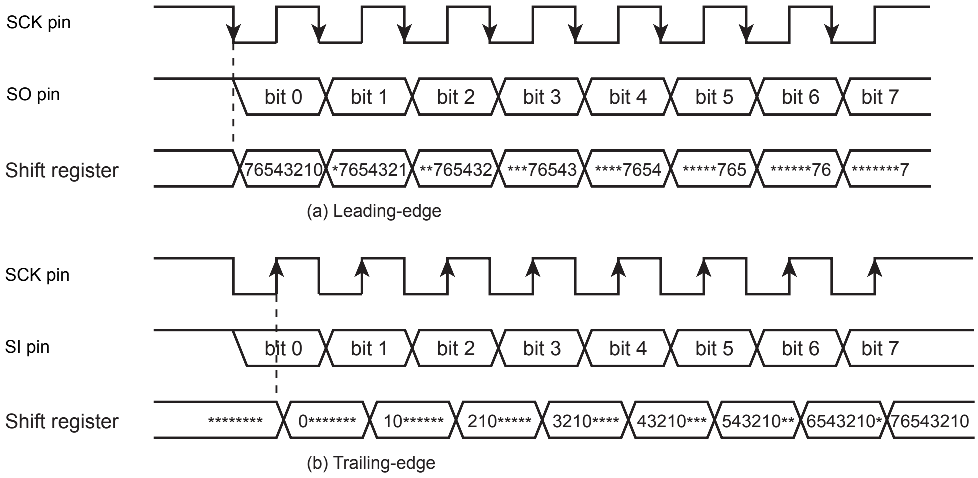


Figure 13-17 Shift Edge

13.8.2 Transfer Modes

The transmit mode, the receive mode or the transmit/receive mode can be selected by programming SBICR1<SIOM[1:0]>.

13.8.2.1 8-bit transmit mode

Set the control register to the transmit mode and write the transmit data to SBIDBR.

After writing the transmit data, writing "1" to SBICR1<SIOS> starts the transmission. The transmit data is moved from SBIDBR 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, SBIDBR becomes empty, and the INTSBI (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 SBIDBR is loaded with the next transmit data.

In the external clock mode, SBIDBR 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 SBIDBR 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 SBISR<SIOF> to "1" to the falling edge of SCK.

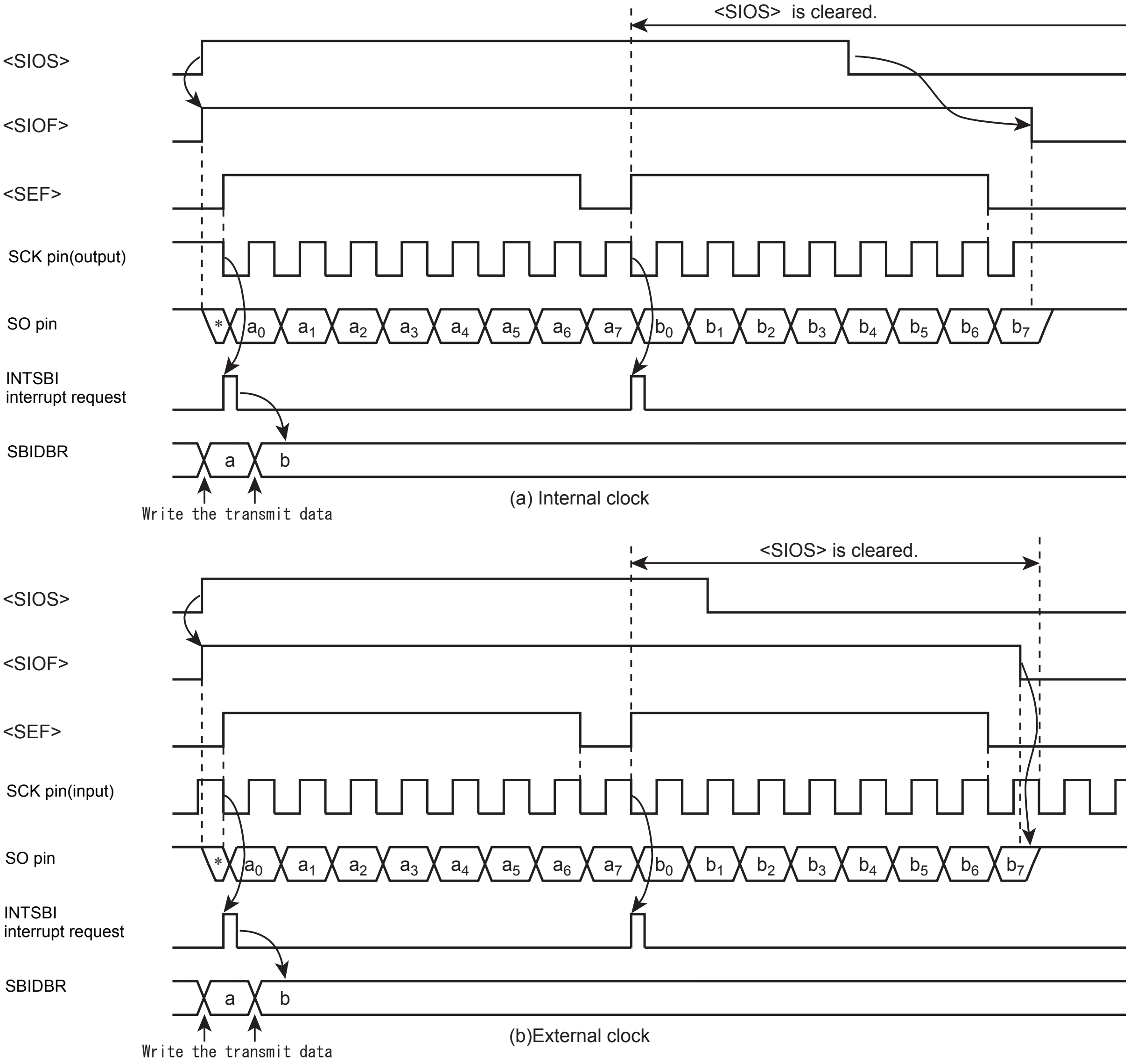
Transmission can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBI interrupt service program. If <SIOS> is cleared, remaining data is output before transmission ends. The program checks SBISR<SIOF> to determine whether transmission has come to an end. <SIOF> is cleared 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	
SBICR1	←	0	1	0	0	0	X	X	X	Selects the transmit mode.
SBIDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBICR1	←	1	0	0	0	0	X	X	X	Starts transmission.

INTSBI interrupt

SBDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.



(a) Internal clock

(b)External clock

Figure 13-18 Transmit Mode

Example: Example of programming (external clock) to terminate transmission by <SIO>

		7	6	5	4	3	2	1	0	
if SBISR<SIOF> ≠ 0										Recognizes the completion of the transmission.
Then										
if SCK ≠ 1										Recognizes "1" is set to the SCK pin by monitoring the port.
Then										
SBICR1	←	0	0	0	0	0	1	1	1	Completes the transmission by setting <SIOS> = 0.

13.8.2.2 8-bit receive mode

Set the control register to the receive mode. Then writing "1" to SBICR1<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 SBIDBR and the INTSBI (buffer-full) interrupt request is generated to request reading the received data. The interrupt service program then reads the received data from SBIDBR.

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 SBIDBR.

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 INTSBI interrupt service program. If <SIOS> is cleared, reception continues until all the bits of received data are written to SBIDBR. The program checks SBISR<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 SBIDBR 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	
SBICR1	←	0	1	1	1	0	X	X	X	Selects the receive mode.
SBICR1	←	1	0	1	1	0	X	X	X	Starts reception.

INTSBI interrupt

Reg.	←	SBIDBR	Reads the received data.

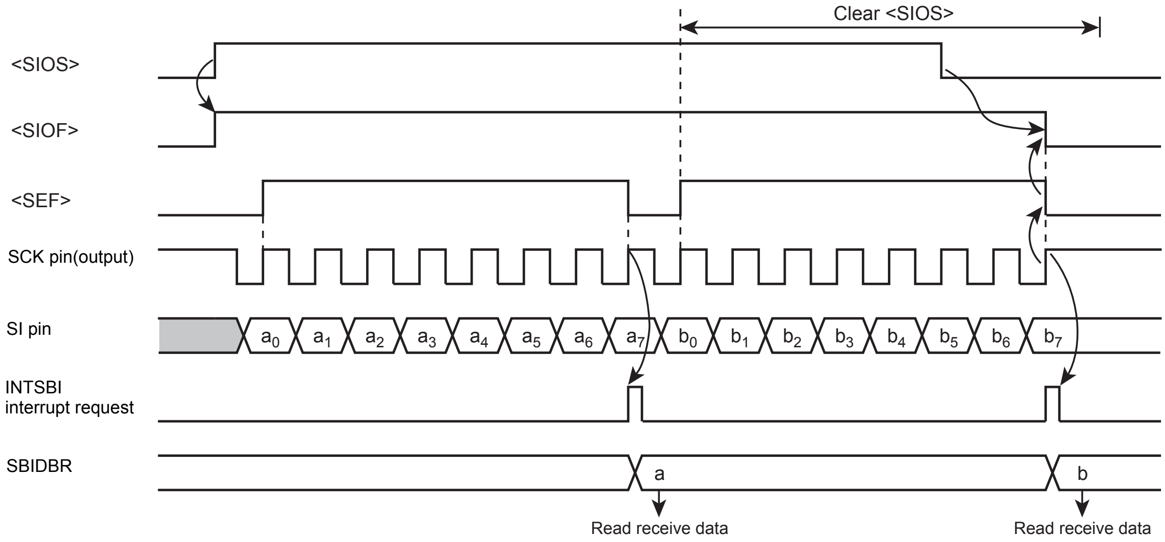


Figure 13-19 Receive Mode (Example: Internal Clock)

13.8.2.3 8-bit transmit/receive mode

Set the control register to the transfer/receive mode. Then writing the transmit data to SBIDBR and setting SBICR1<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 SBIDBR and the INTSBI interrupt request is generated.The interrupt service program reads the received data from the data buffer register and writes the next transmit data. Because SBIDBR 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 SBICR1<SIOINH> to "1" in the INTSBI interrupt service program. If <SIOS> is cleared, transmission and reception continue until the received data is fully transferred to SBIDBR. The program checks SBISR<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 SBIDBR 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.**

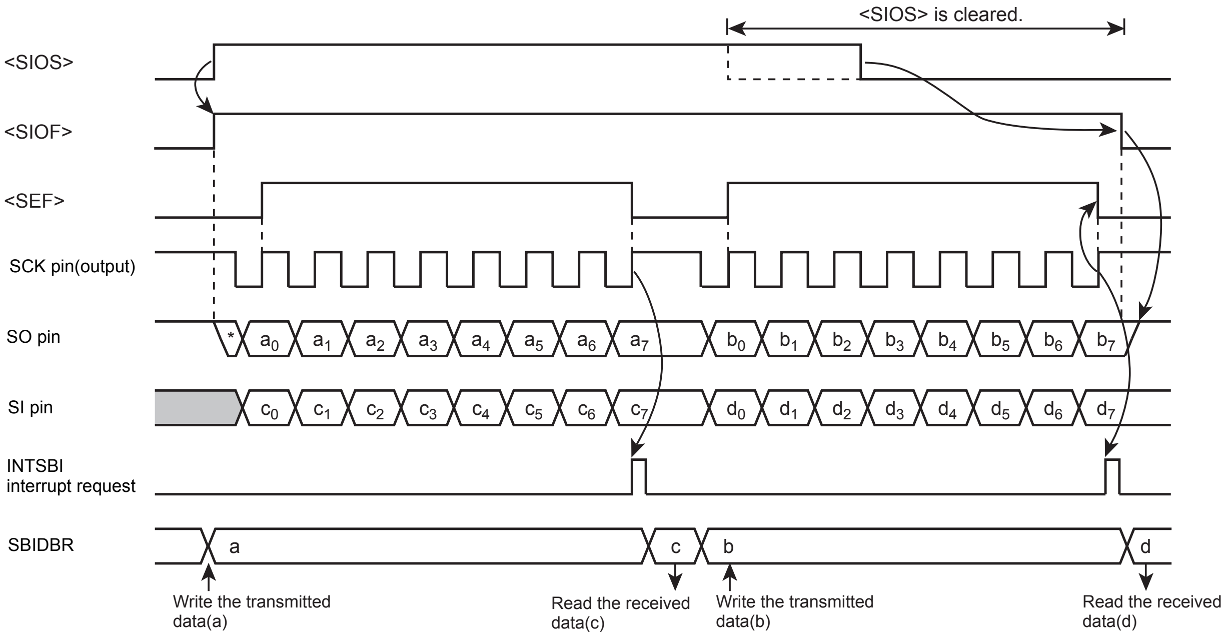

<SIOS> is cleared.
<SIOS>
<SIOF>
<SEF>
SCK pin(output)
SO pin
*
a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7
b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7
SI pin
c_0 c_1 c_2 c_3 c_4 c_5 c_6 c_7
d_0 d_1 d_2 d_3 d_4 d_5 d_6 d_7
INTSBI
interrupt request
SBIDBR
a
c
b
d
Write the transmitted data(a)
Read the received data(c)
Write the transmitted data(b)
Read the received data(d)

Figure 13-20 Transmit/Receive Mode (Example: Internal Clock)

		7	6	5	4	3	2	1	0	
SBICR1	←	0	1	1	0	0	X	X	X	Selects the transmit mode.
SBIDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBICR1	←	1	0	1	0	0	X	X	X	Starts reception/transmission.

INTSBI interrupt

Reg.	←	SBIDBR								Reads the received data.
SBIDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

13.8.2.4 Data retention time of the last bit at the end of transmission

Under the condition SBICR1<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.

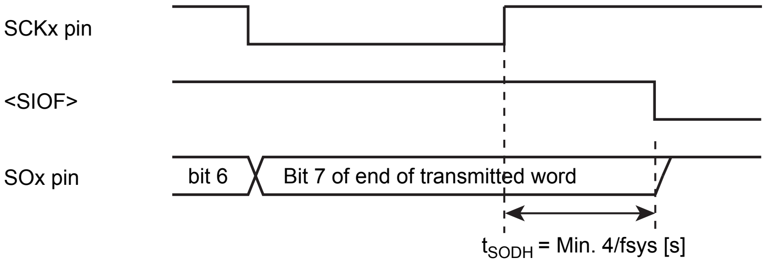

SCKx pin
<SIOF>
SOx pin
bit 6
Bit 7 of end of transmitted word
t_{SODH} = Min. 4/fsys [s]

Figure 13-21 Data retention time of the last bit at the end of transmission

14. Synchronous Serial Port (SSP)

14.1 Overview

This LSI contains the SSP (Synchronous Serial Port) with 1 channel. This channel has the following features.

Communication protocol		Three types of synchronous serial ports including the SPI ・ Motorola SPI (SPI) frame format ・ TI synchronous (SSI) frame format ・ National Microwire (Microwire) frame format
Operation mode		Master/slave mode
Transmit FIFO		16bits wide / 8 tiers deep
Receive FIFO		16bits wide / 8 tiers deep
Transmitted/received data size		4 to 16 bits
Interrupt type		Transmit interrupt Receive interrupt Receive overrun interrupt Time-out interrupt
Communication speed	In master mode	fsys / 2 (max. 10Mbps)
	In slave mode	fsys (40MHz) / 12 (max. 3.3Mbps)
Internal test function		The internal loopback test mode is available.
Control pin		SPCLK,SPFSS,SPDO,SPDI

14.2 Block Diagram

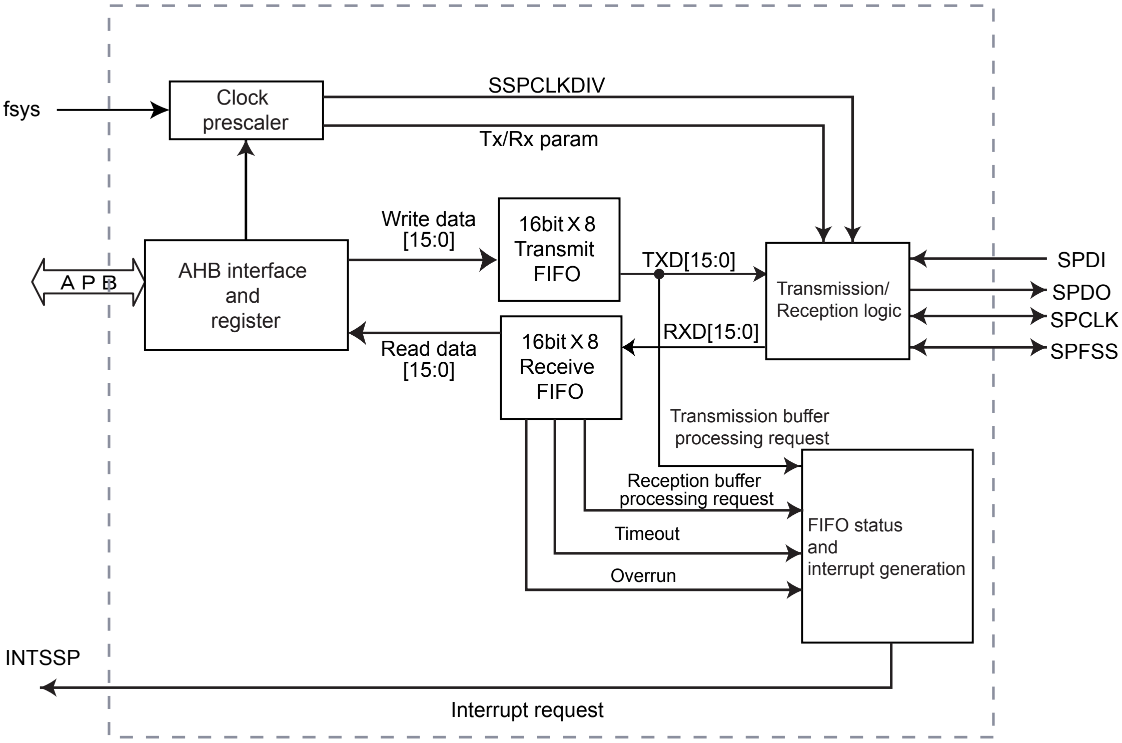


Figure 14-1 SSP block diagram

14.3 Register

14.3.1 Register List

Base Address = 0x400C_0000

Register Name		Address (Base+)
Control register 0	SSPCR0	0x0000
Control register 1	SSPCR1	0x0004
Receive FIFO (read) and transmit FIFO (write) data register	SSPDR	0x0008
Status register	SSPSR	0x000C
Clock prescale register	SSPCPSR	0x0010
Interrupt enable/disable register	SSPIMSC	0x0014
Pre-enable interrupt status register	SSPRIS	0x0018
Post-enable interrupt status register	SSPMIS	0x001C
Interrupt clear register	SSPICR	0x0020
Reserved	-	0x0028 to 0x0FFC

Note 1: These registers in the above table allows to access only word (32 bits) basis.

Note 2: Access to the "Reserved" area is prohibited.

14.3.2 SSPCR0(Control register 0)

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-16</td><td>−</td><td>W</td><td>Write as "0".</td></tr>
<tr><td>15-8</td><td>SCR[7:0]</td><td>R/W</td><td>For serial clock rate setting.
Parameter : 0x00 to 0xFF.

Bits to generate the SSP transmit bit rate and receive bit rate.
This bit rate can be obtained by the following equation.
Bit rate = f_{sys} / (<CPSDVSR> × (1+ <SCR>))
<CPSDVSR> is an even number between 2 to 254, which is programmed by the SSPCPSR register, and <SCR> takes a value between 0 to 255.</td></tr>
<tr><td>7</td><td>SPH</td><td>R/W</td><td>SPCLK phase:
0 : Captures data at the 1st clock edge.
1 : Captures data at the 2nd clock edge.
This is applicable to Motorola SPI frame format only. Refer to Section "Motorola SPI frame format"</td></tr>
<tr><td>6</td><td>SPO</td><td>R/W</td><td>SPCLK polarity:
0:SPCLK is in Low state.
1:SPCLK is in High state.
This is applicable to Motorola SPI frame format only. Refer to Section "Motorola SPI frame format"</td></tr>
<tr><td>5-4</td><td>FRF[1:0]</td><td>R/W</td><td>Frame format:
00: SPI frame format
01: SSI serial frame format
10: Microwire frame format
11: Reserved, undefined operation</td></tr>
<tr><td>3-0</td><td>DSS[3:0]</td><td>R/W</td><td>Data size select:
<table>
<tr><td>0000:</td><td>Reserved, undefined operation</td><td>1000:</td><td>9 bits data</td></tr>
<tr><td>0001:</td><td>Reserved, undefined operation</td><td>1001:</td><td>10 bits data</td></tr>
<tr><td>0010:</td><td>Reserved, undefined operation</td><td>1010:</td><td>11 bits data</td></tr>
<tr><td>0011:</td><td>4 bits data</td><td>1011:</td><td>12 bits data</td></tr>
<tr><td>0100:</td><td>5 bits data</td><td>1100:</td><td>13 bits data</td></tr>
<tr><td>0101:</td><td>6 bits data</td><td>1101:</td><td>14 bits data</td></tr>
<tr><td>0110:</td><td>7 bits data</td><td>1110:</td><td>15 bits data</td></tr>
<tr><td>0111:</td><td>8 bits data</td><td>1111:</td><td>16 bits data</td></tr>
</table></td></tr>
</table>

Note: Set a clock prescaler to SSPCR0<SCR[7:0]> = 0x00 , SSPCPSR<CPSDVSR[7:0]> = 0x02, when slave mode is selected.

14.3.3 SSPCR1(Control register1)

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

Bit	Bit Symbol	Type	Function
31-4	−	W	Write as "0".
3	SOD	R/W	Slave mode SPDO output control: 0: Enable 1: Disable Slave mode output disable. This bit is relevant only in the slave mode (<MS>="1").
2	MS	R/W	Master/slave mode select: (Note) 0: Device configured as a master. 1: Device configured as a slave.
1	SSE	R/W	SSP enable/disable 0: Disable 1: Enable
0	LBM	R/W	Loop back mode 0: Normal serial port operation enabled. 1: Output of transmit serial shifter is connected to input of receive serial shifter internally.

Note: This bit is for switching between master and slave. Be sure to configure in the following steps in slave mode and in transmission.

1)	Set to slave mode	:<MS>=1
2)	Set transmit data in FIFO	:<DATA>=0x****
3)	Set SSP to Enable.	:<SSE>=1

14.3.4 SSPDR(Data register)

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

Bit	Bit Symbol	Type	Function
31-16	−	W	Write as "0".
15-0	DATA[15:0]	R/W	Transmit/receive FIFO data: 0x0000 to 0xFFFF Read: Receive FIFO Write: Transmit FIFO If the data size used in the program is less than 16bits, write the data to fit LSB.The transmit control circuit ignores unused bits of MSB side. The receive control circuit receives the data to fit LSB automatically.

14.3.5 SSPSR(Status register)

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

Bit	Bit Symbol	Type	Function
31-5	−	W	Write as "0".
4	BSY	R	Busy flag: 0: Idle 1: Busy <BSY>="1" indicates that the SSP is currently transmitting and/or receiving a frame or the transmit FIFO is not empty.
3	RFF	R	Receive FIFO full flag: 0: Receive FIFO is not full. 1: Receive FIFO is full.
2	RNE	R	Receive FIFO empty flag: 0: Receive FIFO is empty. 1: Receive FIFO is not empty.
1	TNF	R	Transmit FIFO full flag: 0: Transmit FIFO is full. 1: Transmit FIFO is not full.
0	TFE	R	Transmit FIFO empty flag: 0: Transmit FIFO is not empty. 1: Transmit FIFO is empty.

14.3.6 SSPCPSR (Clock prescale register)

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

Bit	Bit Symbol	Type	Function
31-8	−	W	Write as "0".
7-0	CPSDVSR[7:0]	R/W	Clock prescale divisor: Set an even number from 2 to 254. Clock prescale divisor: Must be an even number from 2 to 254, depending on the frequency of fsys. The least significant bit always returns zero when read.

Note: Set a clock prescaler to SSPCR0<SCR[7:0]> = 0x00 , SSPCPSR<CPSDVSR[7:0]> = 0x02, when slave mode is selected.

14.3.7 SSPIMSC (Interrupt enable/disable register)

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

Bit	Bit Symbol	Type	Function
31-4	−	W	Write as "0".
3	TXIM	R/W	Transmit FIFO interrupt enable: 0: Disable 1: Enable Enable or disable a conditional interrupt to occur if the transmit FIFO is half empty or less.
2	RXIM	R/W	Receive FIFO interrupt enable: 0: Disable 1: Enable Enable or disable a conditional interrupt to occur if the receive FIFO is half full or less.
1	RTIM	R/W	Receive time-out interrupt enable: 0: Disable 1: Enable Enable or disable a conditional interrupt to indicate that data exists in the receive FIFO to the time-out period and data is not read.
0	RORIM	R/W	Receive overrun interrupt enable: 0: Disable 1: Enable Enable or disable a conditional interrupt to indicate that data was written when the receive FIFO was in the full condition.

14.3.8 SSPRIS (Pre-enable interrupt status register)

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

Bit	Bit Symbol	Type	Function
31-4	−	W	Write as "0".
3	TXRIS	R	Pre-enable transmit interrupt flag: 0: Interrupt not present 1: Interrupt present
2	RXRIS	R	Pre-enable receive interrupt flag: 0: Interrupt not present 1: Interrupt present
1	RTRIS	R	Pre-enable timeout interrupt flag: 0: Interrupt not present 1: Interrupt present
0	RORRIS	R	Pre-enable overrun interrupt flag: 0: Interrupt not present 1: Interrupt present

14.3.9 SSPMIS (Post-enable interrupt status register)

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

Bit	Bit Symbol	Type	Function
31-4	−	W	Write as "0".
3	TXMIS	R	Post-enable transmit interrupt flag: 0: Interrupt not present 1: Interrupt present
2	RXMIS	R	Post-enable receive interrupt flag: 0: Interrupt not present 1: Interrupt present
1	RTMIS	R	Post-enable time-out interrupt flag: 0: Interrupt not present 1: Interrupt present
0	RORMIS	R	Post-enable overrun interrupt flag: 0: Interrupt not present 1: Interrupt present

14.3.10 SSPICR (Interrupt clear register)

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

Bit	Bit Symbol	Type	Function
31-2	−	W	Write as "0".
1	RTIC	W	Clear the time-out interrupt flag: 0: Invalid 1: Clear
0	RORIC	W	Clear the overrun interrupt flag: 0: Invalid 1: Clear

14.4 Overview of SSP

This LSI contains the SSP with 1channels.

The SSP is an interface that enables serial communications with the peripheral devices with three types of synchronous serial interface functions.

The SSP performs serial-parallel conversion of the data received from a peripheral device.

The transmit buffers data in the independent 16-bit wide and 8-layered transmit FIFO in the transmit mode, and the receive buffers data in the 16-bit wide and 8-layered receive FIFO in receive mode. Serial data is transmitted via SPDO and received via SPDI.

The SSP contains a programmable prescaler to generate the serial output clock SPCLK from the input clock fsys. The operation mode, frame format, and data size of the SSP are programmed in the control registers SSPCR0 and SSPCR1.

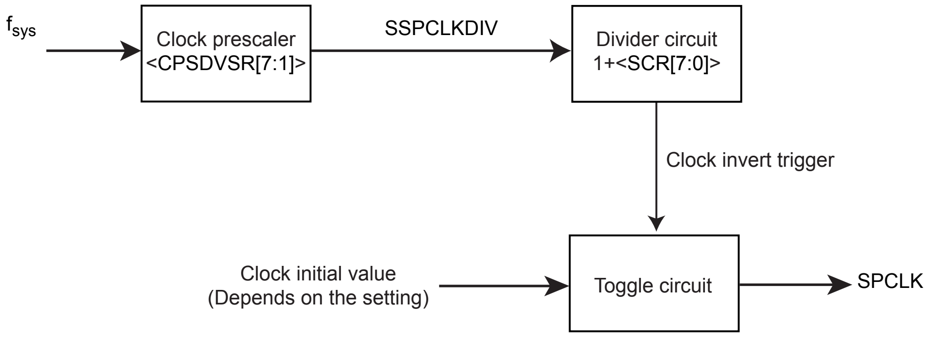
14.4.1 Clock prescaler

When configured as a master, a clock prescaler comprising two free-running serially linked counters is used to provide the serial output clock SPCLK.

You can program the clock prescaler through the SSPCPSR register, to divide fsys by a factor of 2 to 254 in steps of two. Because the least significant bit of the SSPCPSR register is not used, division by an odd number is not possible.

The output of the prescaler is further divided by a factor of 1 to 256, which is obtained by adding 1 to the value programmed in the SSPCR0 register, to give the master output clock SPCLK.

Bitrate = fsys / (<CPSDVSR>×(1+<SCR>))



14.4.2 Transmit FIFO

This is a 16-bit wide, 8-layered transmit FIFO buffer, which is shared in master and slave modes.

14.4.3 Receive FIFO

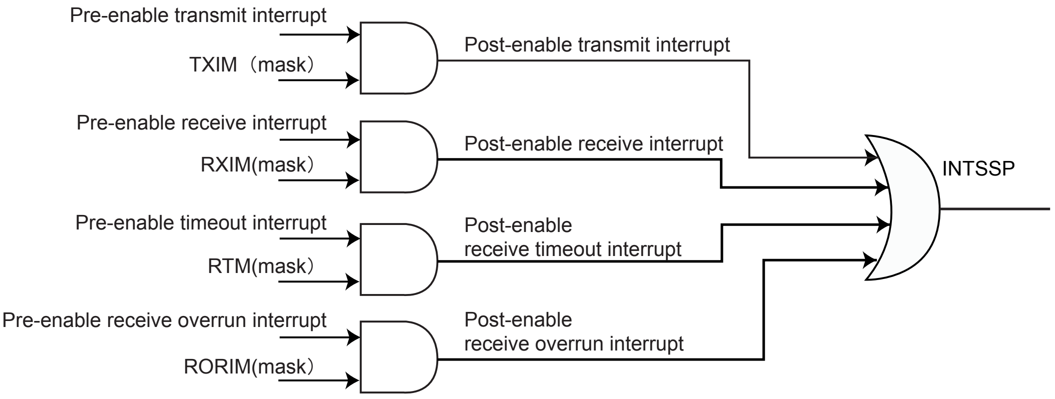
This is a 16-bit wide 8-layered receive FIFO buffer, which is shared in master and slave modes.

14.4.4 Interrupt generation logic

The interrupts, each of which can be masked separately, are generated.

Transmit interrupt	A conditional interrupt to occur when the transmit FIFO has free space more than (including half) of the entire capacity. (Number of valid data items in the transmit FIFO ≤ 4)
Receive interrupt	A conditional interrupt to occur when the receive FIFO has valid data more than half (including half) the entire capacity. (Number of valid data items in the receive FIFO ≥ 4)
Time-out interrupt	A conditional interrupt to indicate that the data exists in the receive FIFO to the time-out period.
Overrun interrupt	Conditional interrupts indicating that data is written to receive FIFO when it is full.

Also, The individual masked sources are combined into a single interrupt. When any of the above interrupts is asserted, the combined interrupt INTSSP is asserted.


Pre-enable transmit interrupt
TXIM（mask）
Post-enable transmit interrupt
Pre-enable receive interrupt
RXIM(mask）
Post-enable receive interrupt
Pre-enable timeout interrupt
RTM(mask）
Post-enable
receive timeout interrupt
Pre-enable receive overrun interrupt
RORIM(mask）
Post-enable
receive overrun interrupt
INTSSP

a. Transmit interrupt

The transmit interrupt is asserted when there are four or fewer valid entries in the transmit FIFO. The transmit interrupt is also generated when the SSP operation is disabled (SSPCR1 <SSE> = "0").

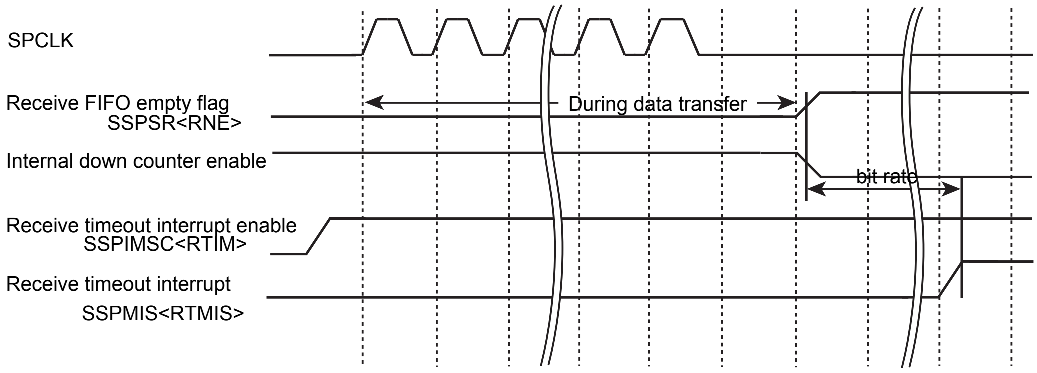
The first transmitted data can be written in the FIFO by using this interrupt.

b. Receive interrupt

The receive interrupt is asserted when there are four or more valid entries in the receive FIFO.

c. Time-out interrupt

The time-out interrupt is asserted when the receive FIFO is not empty and the SSP has remained idle for a fixed 32-bit period (bit rate). This mechanism ensures that the user is aware that data is still present in the receive FIFO and requires servicing. This operation occurs in both master and slave modes. When the time-out interrupt is generated, read all data from the receive FIFO. Even if all the data is not read, data can be transmitted / received if the receive FIFO has a free space and the number of data to be transmitted does not exceed the free space of the receive FIFO. When transfer starts, the timeout interrupt will be cleared. If data is transmitted / received when the receive FIFO has no free space, the time-out interrupt will not be cleared and an overrun interrupt will be generated.


SPCLK
Receive FIFO empty flag
SSPSR<RNE>
During data transfer
Internal down counter enable
bit rate
Receive timeout interrupt enable
SSPIMSC<RTIM>
Receive timeout interrupt
SSPMIS<RTMIS>

d. Overrun interrupt

When the next data (9th data item) is received when the receive FIFO is already full, an overrun interrupt is generated immediately after transfer. The data received after the overrun interrupt is generated (including the 9th data item) will become invalid and be discarded. However, if data is read from the receive FIFO while the 9th data item is being received (before the interrupt is generated), the 9th received data will be written in the receive FIFO as valid data. To perform transfer properly when the overrun interrupt has been generated, write "1" to SSPICR<RORIC> register, and then read all data from the receive FIFO. Even if all the data is not read, data can be transmitted / received if the receive FIFO has free space and the number of data to be transmitted does not exceed the free space of the receive FIFO. Note that if the receive FIFO is not read (provided that the receive FIFO is not empty) within a certain 32-bit period (bit rate) after the overrun interrupt is cleared, a time-out interrupt will be generated.

14.5 SSP operation

14.5.1 Initial setting for SSP

Settings for the SSP communication protocol must be made with the SSP disabled.

Control registers SSPCR0 and SSPCR1 need to configure this SSP as a master or slave operating under one of the following protocols. In addition, make the settings related to the communication speed in the clock prescale registers SSPCPSR and SSPCR0 <SCR>.

This SSP supports the following protocols:

- SPI
- SSI
- Microwire

14.5.2 Enabling SSP

The transfer operation starts when the operation is enabled with the transmitted data written in the transmit FIFO, or when transmitted data is written in the transmit FIFO with the operation enabled.

However, if the transmit FIFO contains only four or fewer entries when the operation is enabled, a transmit interrupt will be generated. This interrupt can be used to write the initial data.

Note: When the SSP is in the SPI slave mode and the SPFSS pin is not used, be sure to transmit data of one byte or more in the FIFO before enabling the operation. If the operation is enabled with the transmit FIFO empty, the transfer data will not be output correctly.

14.5.3 Clock ratios

When setting a frequency for fsys , the following conditions must be met.

- In master mode

  f_{SPCLK} (maximum) → f_{sys} /2

  f_{SPCLK} (minimum) → f_{sys} /(254×256)

- In slave mode

  f_{SPCLK} (maximum) → f_{sys} /12

  f_{SPCLK} (minimum) → f_{sys} /(254×256)

Note: The maximum baud-rate in the master mode is equal or less than 10Mbps.

14.6 Frame Format

Each frame format is between 4 and 16 bits wide depending on the size of data programmed, and is transmitted starting from the MSB.

- Serial clock (SPCLK)

  Signals remain "Low" in the SSI and Microwire formats and as inactive in the SPI format while the SSP is in the idle state. In addition, data is output at the set bit rate only during data transmission.

- Serial frame (SPFSS)

  In the SPI and Microwire frame formats, signals are set to "Low" active and always asserted to "Low" during frame transmission.

  In the SSI frame format, signals are asserted only during 1 bit rate before each frame transmission. In this frame format, output data is transmitted at the rising edge of SPCLK and the input data is received at its falling edge.

Refer to Section "14.6.1" to "14.6.3" for details of each frame format.

14.6.1 SSI frame format

In this mode, the SSP is in idle state, SPCLK and SPFSS are forcedly set to "Low", and the transmit data line SPDO becomes Hi-Z. When data is written in the transmit FIFO, the master outputs "High" pulses of 1 SPCLK to the SPFSS line. The transmitted data will be transferred from the transmit FIFO to the transmit serial shift register. Data of 4 to 16 bits will be output from the SPDO pin at the next rising edge of SPCLK.

Likewise, the received data will be input starting from the MSB to the SPDI pin at the falling edge of SPCLK. The received data will be transferred from the serial shift register into the receive FIFO at the rising edge of SPCLK after its LSB data is latched.

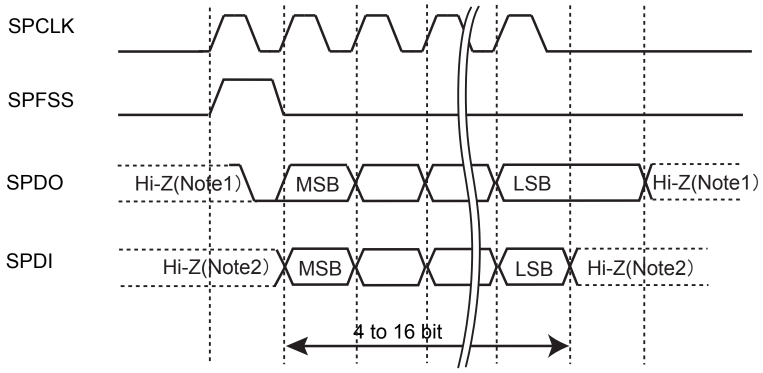


Figure 14-2 SSI frame format (transmission/reception during single transfer)

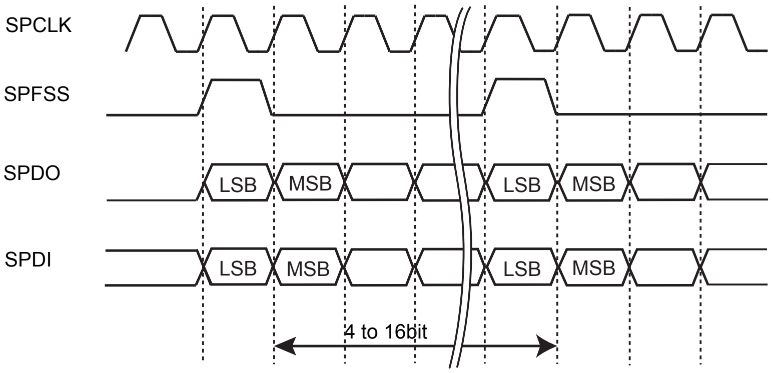


Figure 14-3 SSI frame format (transmission/reception during continuous transfer)

Note 1: When transmission is disable, SPDO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to valid the voltage level.

Note 2: SPDI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to valid the voltage level.

14.6.2 SPI frame format

The SPI interface has 4 lines. SPFSS is used for slave selection. One of the main features of the SPI format is that the <SPO> and <SPH> bits in the SSPCR0 register can be used to set the SPCLK operation timing.

SSPCR0 <SPO> is used to set the level at which SPCLK in idle state is held.

SSPCR0 <SPH> is used to select the clock edge at which data is latched.

	SSPCR0<SPO>	SSPCR0<SPH>
0	"Low" state	Capture data at the 1st clock edge.
1	"High" state	Capture data at the 2nd clock edge.

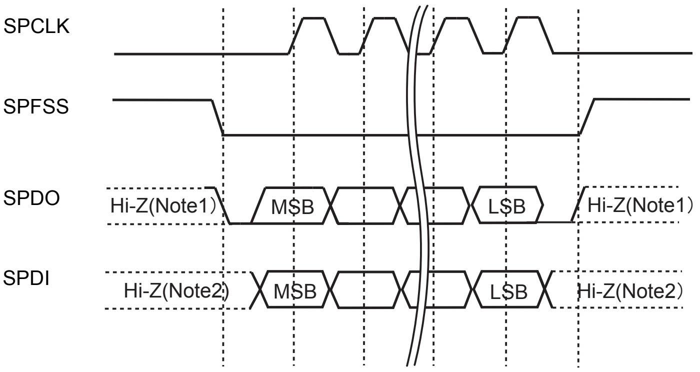


Figure 14-4 SPI frame format (single transfer, <SPO>="0" & <SPH>="0")

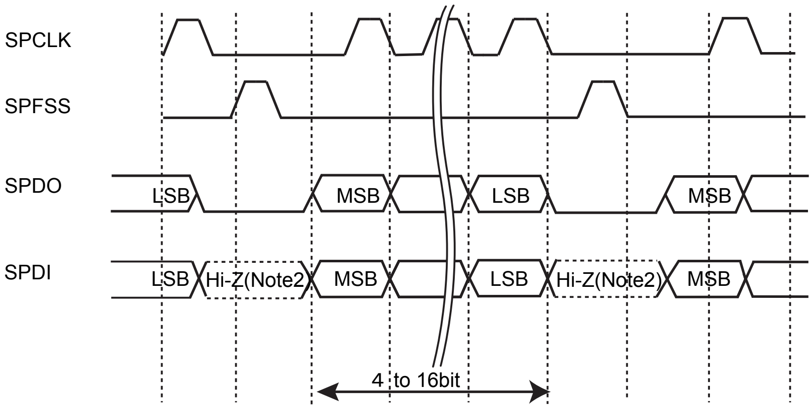


Figure 14-5 SPI frame format (continuous transfer,<SPO>="0" & <SPH>="0")

Note 1: When transmission is disable, SPDO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to valid the voltage level.

Note 2: SPDI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to valid the voltage level.

With this setting <SPO>="0", during the idle period:

- The SPCLK signal is set to "Low".
- SPFSS is set to "High".
- The transmit data line SPDO is set to "Low".

If the SSP is enabled and valid data exists in the transmit FIFO, the SPFSS master signal driven by "Low" notifies of the start of transmission. This enables the slave data in the SPDI input line of the master.

When a half of the SPCLK period has passed, valid master data is transferred to the SPDO pin. Both the master data and slave data are now set. When another half of SPCLK has passed, the SPCLK master clock pin becomes "High". After that, the data is captured at the rising edge of the SPCLK signal and transmitted at its falling edge.

In the single transfer, the SPFSS line will return to the idle "High" state when all the bits of that data word have been transferred, and then one cycle of SPCLK has passed after the last bit was captured.

However, for continuous transfer, the SPFSS signal must be pulsed at HIGH between individual data word transfers. This is because change is not enabled when the slave selection pin freezes data in its peripheral register and the <SPH> bit is logical 0.

Therefore, to enable writing of serial peripheral data, the master device must drive the SPFSS pin of the slave device between individual data transfers. When the continuous transfer is completed, the SPFSS pin will return to the idle state when one cycle of SPCLK has passed after the last bit is captured.

14.6.3 Microwire frame format

The Microwire format uses a special master/slave messaging method, which operates in half-duplex mode. In this mode, when a frame begins, an 8-bit control message is transmitted to the slave. During this transmission, no incoming data is received by the SSP. After the message has been transmitted, the slave decodes it, and after waiting one serial clock after the last bit of the 8-bit control message has been sent, it responds with the requested data. The returned data can be 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits.

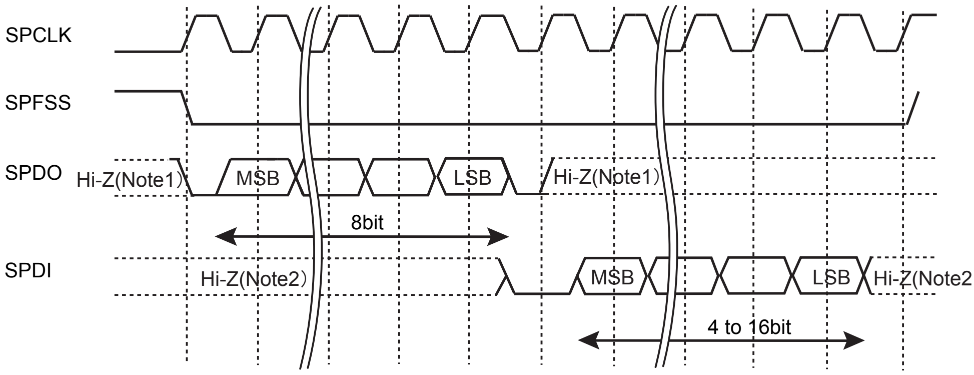


Figure 14-6 Microwire frame format (single transfer)

Note 1: When transmission is disabled, SPDO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to fix the voltage level.

Note 2: SPDI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to fix the voltage level.

Though the Microwire format is similar to the SPI format, it uses the master/slave message transmission method for half-duplex communications. Each serial transmission is started by an 8-bit control word, which is sent to the off-chip slave device. During this transmission, the SSP does not receive input data. After the message has been transmitted, the off-chip slave decodes it, and after waiting one serial clock after the last bit of the 8-bit control message has been sent, responds with the requested data. The returned data can be 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits. With this configuration, during the idle period:

- The SPCLK signal is set to "Low".
- SPFSS is set to "High".
- The transmit data line SPDO is set to "Low".

A transmission is triggered by writing a control byte to the transmit FIFO. The falling edge of SPFSS causes the value stored in the bottom entry of the transmit FIFO to be transferred to the serial shift register for the transmit logic, and the MSB of the 8-bit control frame to be shifted out onto the SPDO pin.

SPFSS remains "Low" and the SPDI pin remains tristated during this transmission. The off-chip serial slave device latches each control bit into its serial shifter on the rising edge of each SPCLK.

After the last bit is latched by the slave device, the control byte is decoded during a one clock wait-state, and the slave responds by transmitting data back to the SSP. Each bit is driven onto SPDI line on the falling edge of SPCLK.

The SSP in turn latches each bit on the rising edge of SPCLK. At the end of the frame, for single transfers, the SPFSS signal is pulled "High" one clock period after the last bit has been latched in the receive serial shifter, which causes the data to be transferred to the receive FIFO.

Note: The off-chip slave device can tristate the receive line either on the falling edge of SPCLK after the LSB has been latched by the receive shifter, or when the SPFSS pin goes "High".

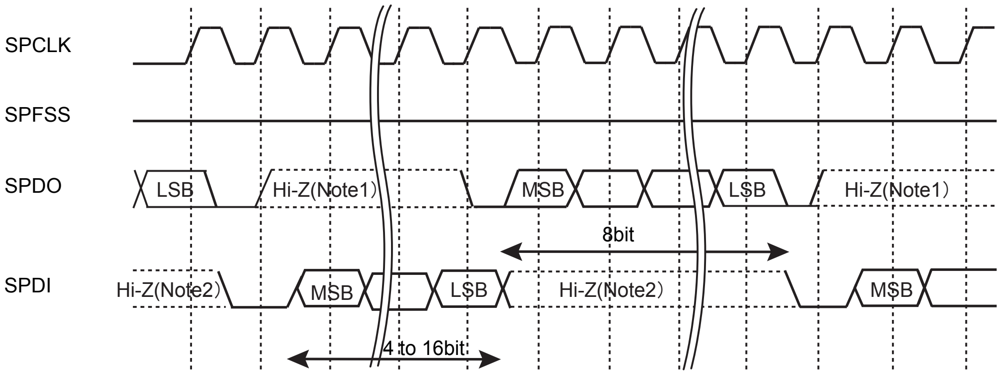


Figure 14-7 Microwire frame format (continuous transfer)

Note 1: When transmission is disabled, SPDO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to fix the voltage level.

Note 2: SPDI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to fix the voltage level.

For continuous transfers, data transmission begins and ends in the same manner as a single transfer. However, the SPFSS line is continuously asserted (held Low) and transmission of data occurs back to back.

The control byte of the next frame follows directly after the LSB of the received data from the current frame. Each of the received values is transferred from the receive shifter on the falling edge of SPCLK, after the LSB of the frame has been latched into the SSP.

Note: [Example of connection] The SSP does not support dynamic switching between the master and slave in the system. Each sample SSP is configured and connected as either a master or slave.

15. Remote Control Signal Preprocessor (RMC)

15.1 Basic operation

Remote control signal preprocessor (hereafter referred to as RMC) receives a remote control signal of which carrier is removed.

15.1.1 Reception of Remote Control Signal

- A sampling clock can be selected from either low frequency clock (32.768kHz) or Timer output.
- Noise canceling time can be adjusted.
- Leader detection
- Batch reception up to 72bit of data

15.2 Block Diagram

Figure 15-1 shows the block diagram of RMC.

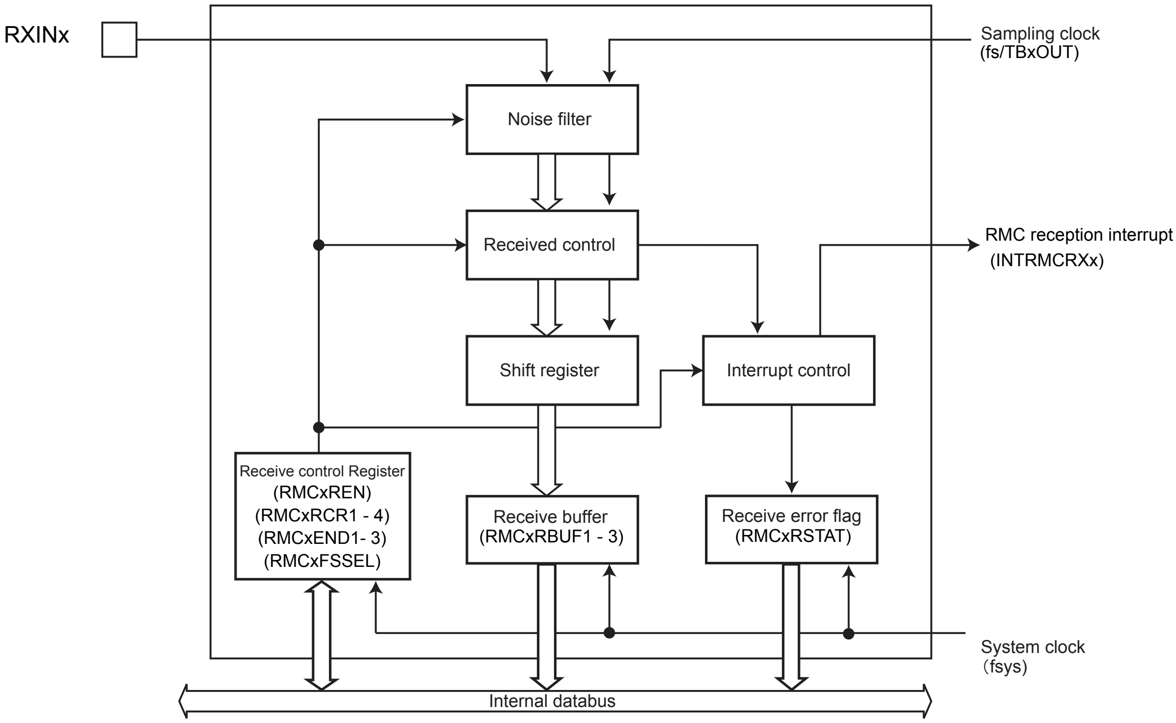


Figure 15-1 Block diagram of RMC

15.3 Registers

15.3.1 Register List

Addresses and names of RMC control registers are shown below.

Base Address = 0x4004_0400

Register		Address(Base+)
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 Register 2	RMCxRCR2	0x0018
Receive Control Register 3	RMCxRCR3	0x001C
Receive Control Register 4	RMCxRCR4	0x0020
Receive Status Register	RMCxRSTAT	0x0024
Receive End bit Number Register 1	RMCxEND1	0x0028
Receive End bit Number Register 2	RMCxEND2	0x002C
Receive End bit Number Register 3	RMCxEND3	0x0030
Source Clock selection Register	RMCxFSSEL	0x0034

15.3.2 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	-	-	-	-	-	-	-	RMCEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	−	R/W	Write as "1".
0	RMCEN	R/W	Controls RMC operation. 0: Disabled 1: Enabled To allow RMC to function, enable the <RMCEN> 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.

15.3.3 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 RMCxREN<RMCREN> bit after setting the RMCxRCR1, RMCxRCR2, and RMCxRCR3.

15.3.4 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)

15.3.5 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)

15.3.6 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 1 byte 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.

15.3.7 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

15.3.8 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

15.3.9 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".

15.3.10 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]

15.3.11 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. In this case, the received data in the data buffer may not be ensured.

15.3.12 RMCxEND1(Receive End bit Number 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	-	RMCEND1						
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	-	R	Read as 0.
6-0	RMCEND1[6:0]	R/W	Specifies that the number of receive data bit 000_0000 : No specifically the receive data bit 000_0001 to 100_1000 : Specifies that the number of receive data bit(1 to 72bit) 100_1001 to 111_1111 : Don’t set the value

15.3.13 RMCxEND2(Receive End bit Number 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	-	RMCEND2						
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	-	R	Read as 0.
6-0	RMCEND2[6:0]	R/W	Specifies that the number of receive data bit 000_0000 : No specifically the receive data bit 000_0001 to 100_1000 : Specifies that the number of receive data bit(1 to 72bit) 100_1001 to 111_1111 : Don’t set the value

15.3.14 RMCxEND3(Receive End bit Number 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	-	RMCEND3						
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	-	R	Read as 0.
6-0	RMCEND3[6:0]	R/W	Specifies the number of receive data bit 000_0000 : No specifically the receive data bit 000_0001 to 100_1000 : Specifies that the number of receive data bit(1 to 72bit) 100_1001 to 111_1111 : Don’t set the value

Note 1: As specified to RMCxEND1, RMCxEND2 and RMCxEND3, it is able to set three kinds of the receive data bit.

Note 2: To use the RMCxEND1, RMCxEND2 and RMCxEND3 is in combination with the maximum data bit cycle.

15.3.15 RMCxFSSEL(Source 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	-	-	-	-	-	-	-	RMCCLK
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as 0.
0	RMCCLK	R/W	Specifies that Sampling clock of RMC function 0 : Low frequency Clock (32.768kHz) 1 : Timer output (TBxOUT) For the Sampling of RMC function, It is able to set the Low Frequency Clock (32.768kHz) or Timer output (TBxOUT). For the information of TBxOUT used for sampling clock, refer to Chapter "Product Information". The Setting range of Timer output by TBxOUT is from 30 to 34kHz.

Note: To Change the sampling clock by using the RMCxFSSEL, disable the RMC operation first by using the RMCxEN <RMCEN>. Then, enable it again, and set the RMCxFSSEL before setting other RMC registers.

15.4 Operation Description

15.4.1 Reception of Remote Control Signal

15.4.1.1 Sampling clock

A remote control signal is sampled by using low-speed 32.768kHz clock (fs) or TBxOUT which is output of 16bit Timer/Event counters.

The sampling clock is configurable using RMCxFSSEL<RMCCLK>. For the information of TBxOUT used for sampling clock, refer to Chapter "Product Information".

15.4.1.2 Basic operation

RMC set RMCxRSTAT<RMCRLDR> when a leader is detected.

At this time, if you set the RMCxRCR2<RMCLIEN>, leader detection will generate a leader detection interrupt. When a leader detection interrupt occurs, RMCxRSTAT<RMCRLIF> 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>, 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 > is set.

Data reception stops when the maximum data bit cycle is detected and "Low"-width matches the setting value, and then, an interrupt occurs. If <RMCEND1>, <RMCEND2> and <RMCEND3> of the register RMCxEND1, RMCxEND2 and RMCEND3 have been configured, data reception stops and an interrupt occurs only in the case that the number of bits received before maximum data bit cycle is detected. The condition of RMC can be checked by reading the remote control receive status register.

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.

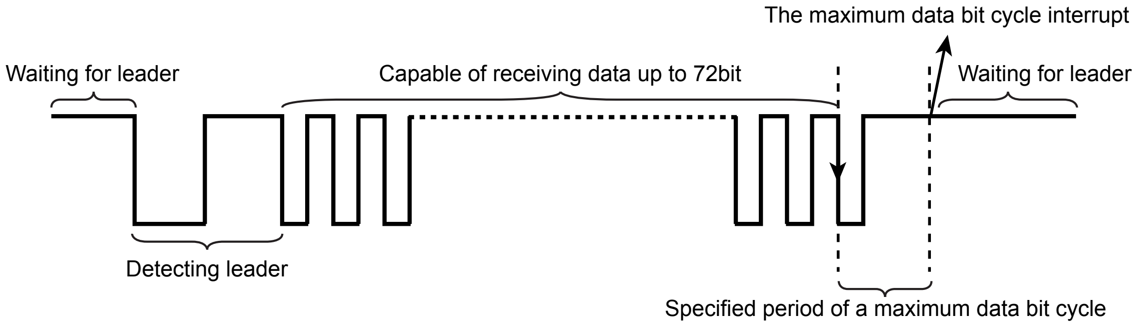

The maximum data bit cycle interrupt
Waiting for leader
Capable of receiving data up to 72bit
Waiting for leader
Detecting leader
Specified period of a maximum data bit cycle

Figure 15-2 Data reception completed by detecting the max data bit cycle

15.4.1.3 Preparation

Before starting receiving process, configure how to receive remote control signal using the Remote Control Signal Receive Control Registers (RMCxRCR1, RMCxRCR2 and RMCxRCR3, RMCxRCR4).

(1) Settings of Noise Cancelling Time

Configure noise cancelling time with the RMCxRCR4 <RMCNC[3:0]>.

Noise canceling is applied to remote control signals sampled by the sampling clock.

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".

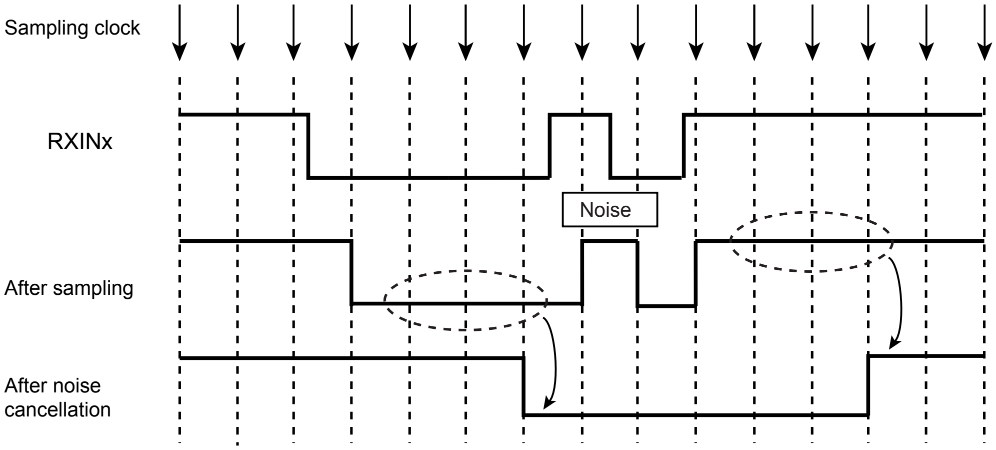


Figure 15-3 Noise Cancel (In the case of RMCxRCR4="0011" (3 Cycles))

(2) Settings of Detecting Leader

Set the leader cycle and a "Low" width of the leader to RMCxRCR1 <RMCLLMIN[7:0]> <RMCLLMAX[7:0]> <RMCLCMIN[7:0]> <RMCLCMAX[7:0]>. When you configure those above, 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]> = 0000_0000 <RMCLLMIN[7:0]> = don't care
No leader	<RMCLCMAX[7:0]> = 0000_0000 <RMCLCMIN[7:0]> = don't care <RMCLLMAX[7:0]> = don't care <RMCLLMIN[7:0]> = don't care

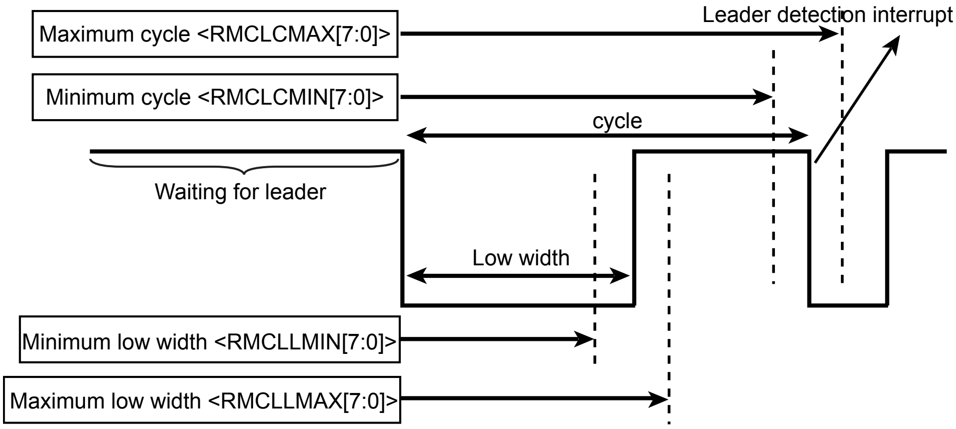

Maximum cycle <RMCLCMAX[7:0]>
Minimum cycle <RMCLCMIN[7:0]>
Leader detection interrupt
cycle
Waiting for leader
Low width
Minimum low width <RMCLLMIN[7:0]>
Maximum low width <RMCLLMAX[7:0]>

Figure 15-4 Leader wave form and the RMCxRCR1 register settings

If you want to generate an interrupt when detecting a leader, configure the RMCxRCR2 <RMCLIEN>.

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 of a cycle modulation is determined as 0 or 1.

There are two kinds of determinations:

As for data bit determination of a remote control signal in a phase method, see"15.4.1.8 Receiving a Remote Control Signal in a Phase Method".

1. Determination by threshold.

   Configure a threshold value to RMCxRCR3<RMCDATL[6:0]> 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>, 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 following shows the determination method of data bit.

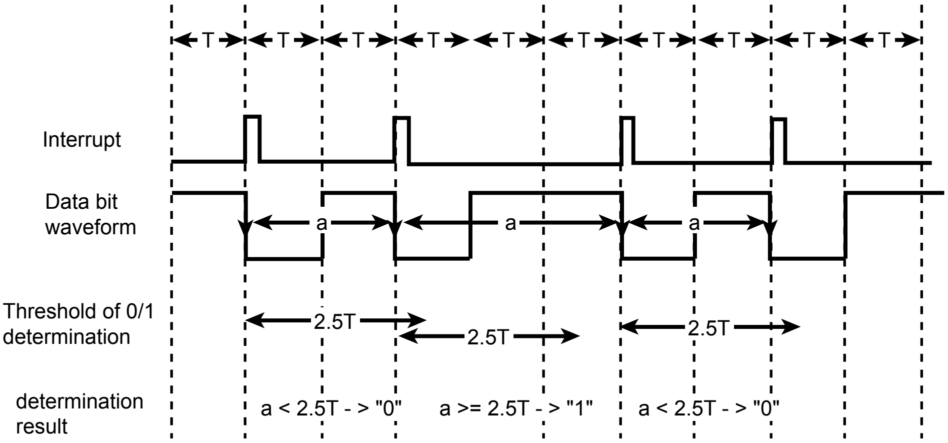


Figure 15-5 Determination method of data bit (In case that threshold is 2.5T)

(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. Completion by the maximum data bit cycle

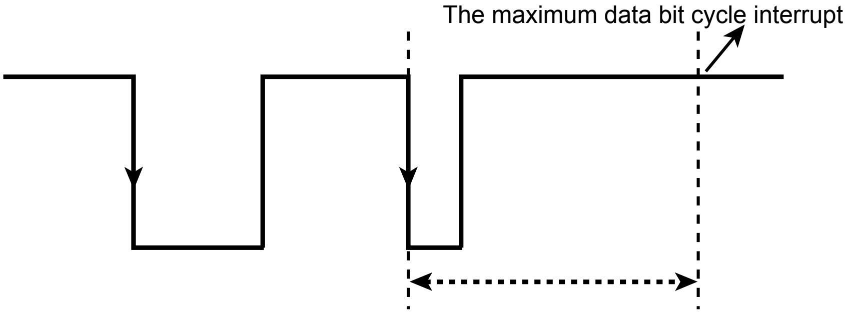
To complete reception by detecting a maximum data bit cycle, you need to configure the RMCxRCR2 <RMCDMAX[7:0]>.

If the falling edge of the data bit cycle isn't monitored after time specified as threshold in the <RMCDMAX[7:0]>, a maximum data bit cycle is detected. The detection completes reception and generates an interrupt.After interrupt inputs generated, RMCxRSTAT< RMCDMAXIF > is set to "1".

To complete reception by setting the number of receive data is set a RMCxEND 1 to 3 register of each <RMCEND1>, <RMCEND2>, <RMCEND3>.In this case when the number of set reception bit agreed with the number of bit which received at the time of the outbreak of MAX on the number of receive data is set a RMCxEND 1 to 3 register of each <RMCEND1>, <RMCEND2>, <RMCEND3>, it occurs by an MAX interrupt in data bit period.

As specified to RMCxEND3 to 1, it is able to set three kinds of the receive data bit.

When it can receive the Maximum Data bit , the number of bit is not match the setting value in <RMCEND1>, <RMCEND2>, <RMCEND3>, it wait for Leader Reception.



Threshold:<RMCDMAX[7:0]>
If the falling edge of the data bit cycle is not monitored after time specified as threshold,a maximum data bit cycle is detected.
The detection completes reception and generates an interrupt.

Figure 15-6 Completion by the maximum data bit cycle

2. Completion by detecting "Low" width

   To complete reception by detecting the "Low" width, you need to configure the RMCxRCR2 <RMCLL[7:0]>.

   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> is set to "1."

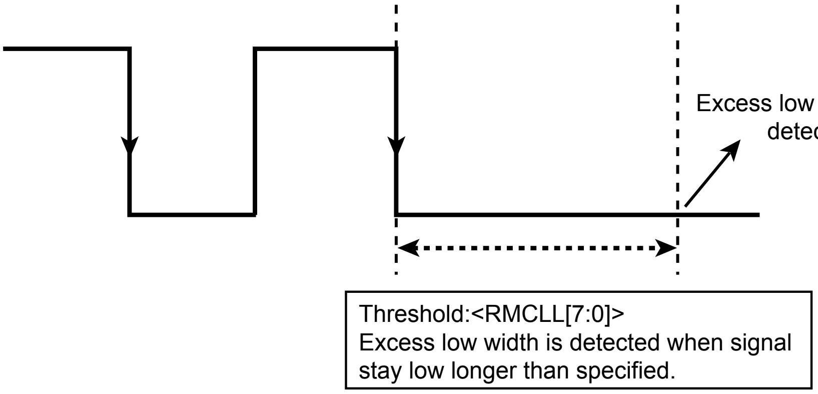

Excess low width
detection interrupt.
Threshold:<RMCLL[7:0]>
Excess low width is detected when signal stay low longer than specified.

Figure 15-7 Completion by detecting "Low" width

15.4.1.4 Enabling Reception

By enabling the RMCxREN <RMCREN> 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.

15.4.1.5 Stopping Reception

RMC stops reception by clearing the RMCxREN <RMCREN> to "0" (reception disabled).

Clearing this bit during reception stops reception immediately and the received data is discarded.

15.4.1.6 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]>. 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.

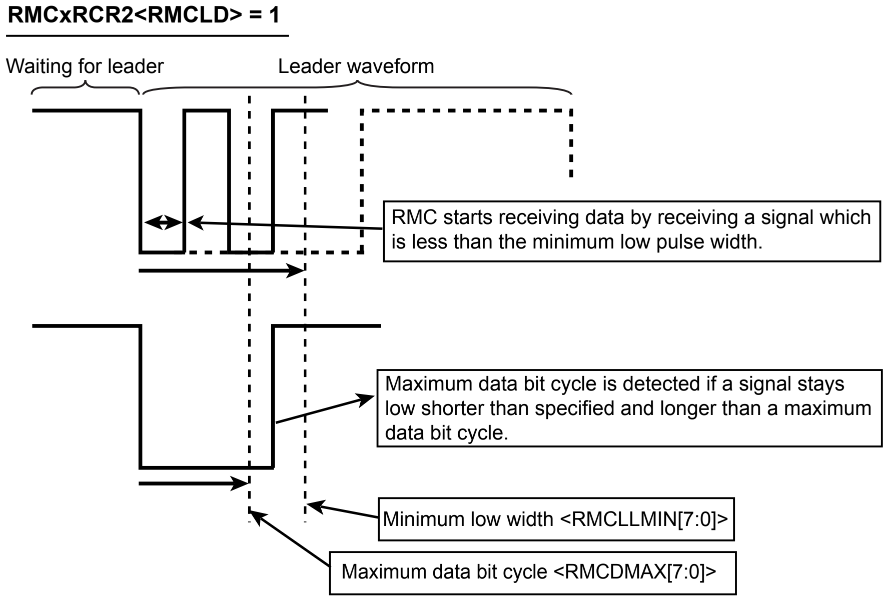

RMCxRCR2<RMCLD> = 1
Waiting for leader
Leader waveform
RMC starts receiving data by receiving a signal which is less than the minimum low pulse width.
Maximum data bit cycle is detected if a signal stays low shorter than specified and longer than a maximum data bit cycle.
Minimum low width <RMCLLMIN[7:0]>
Maximum data bit cycle <RMCDMAX[7:0]>

Figure 15-8 Receiving Remote Control Signal without Leader in Waiting Leader (In the case of RMCxRCR2<RMCLD>="1")

15.4.1.7 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> 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] >=0000 _ 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.

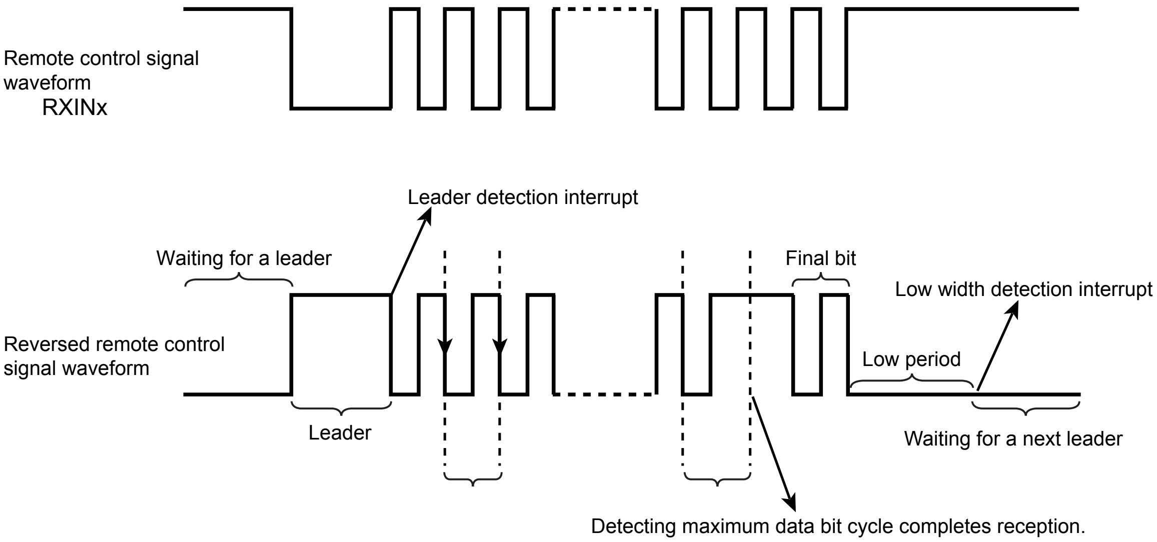

Remote control signal waveform
RXINx
Leader detection interrupt
Waiting for a leader
Final bit
Low width detection interrupt
Reversed remote control signal waveform
Low period
Leader
Waiting for a next leader
Detecting maximum data bit cycle completes reception.

Figure 15-9 A Leader only with "Low" Width

15.4.1.8 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]> and <RMCDATH[6:0]>.

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".

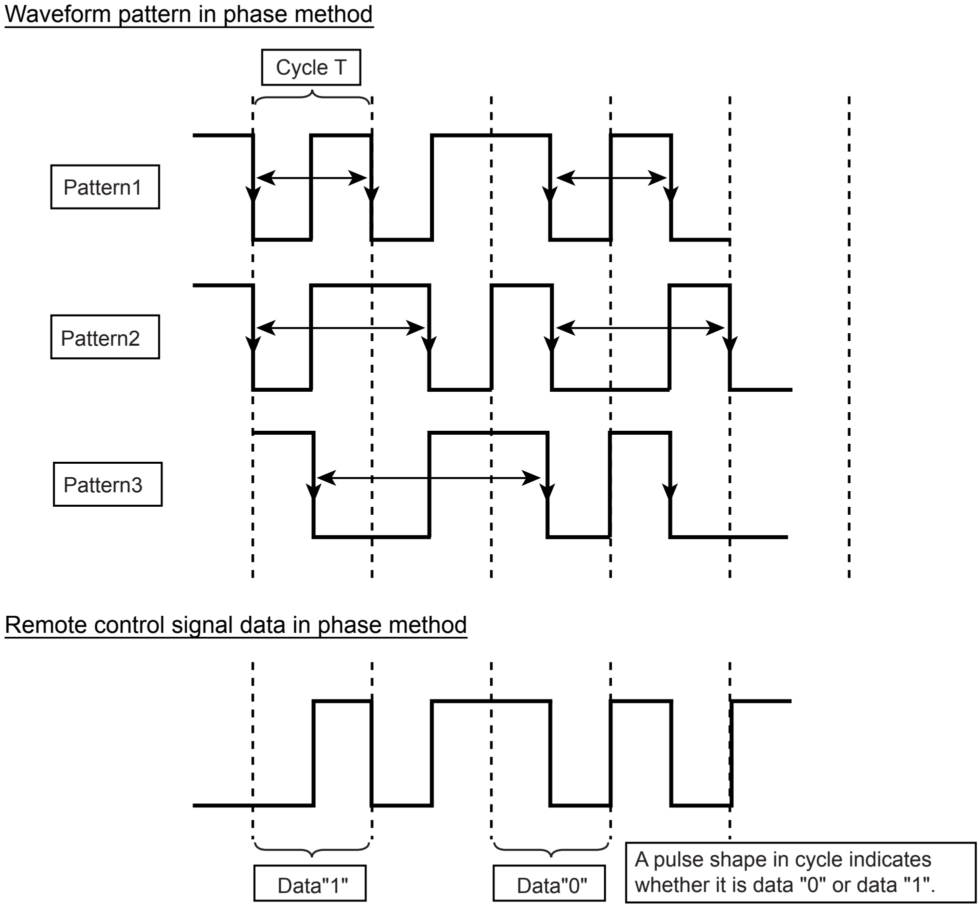

Waveform pattern in phase method
Cycle T
Pattern1
Pattern2
Pattern3
Remote control signal data in phase method
Data"1"
Data"0"
A pulse shape in cycle indicates whether it is data "0" or data "1".

Figure 15-10 Waveform pattern in phase method and the example of data

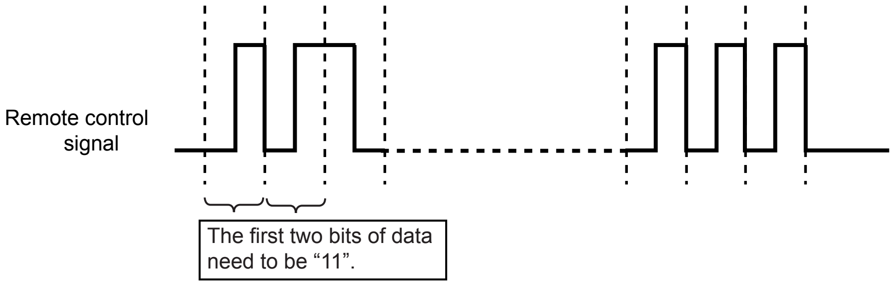


Figure 15-11 The waveform pattern in phase method

16. Analog/Digital Converter (ADC)

The TMPM381/383 contains a 12-/10- (selectable) bit successive-approximation analog-to-digital converter (ADC).

External analog input pins (AIN0 to AIN17) can also be used as input/output ports.

16.1 Functions and features

1. It can select analog input and start AD conversion when receiving trigger signal from TMRB(interrupt).

2. It can select analog input, in the Software Trigger Program and the Constant Trigger Program.

3. The ADCs has twelve register for AD conversion result.

4. The ADCs generate interrupt signal at the end of the program which was started by MRB trigger.

5. The ADCs generate interrupt signal at the end of the program which are the Software Trigger Program and the Constant Trigger Program.

6. The ADCs have the AD conversion monitoring function. When this function is enabled, an interrupt is generated when a conversion result matches the specified comparison value.

16.2 Block Diagram

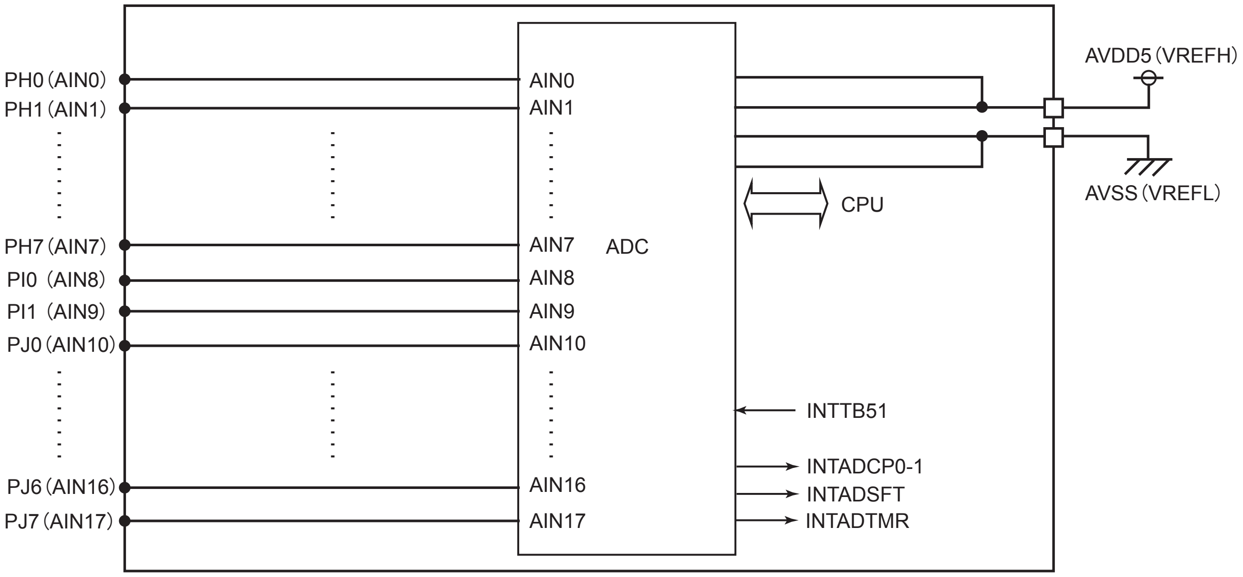


Figure 16-1 AD converters Block Diagram

16.3 List of Registers

Base Address = 0x4003_0000

Register Name		Address(Base+)
Clock Setting Register	ADCLK	0x0000
Mode Setting Register 0	ADMOD0	0x0004
Mode Setting Register 1	ADMOD1	0x0008
Mode Setting Register 2	ADMOD2	0x000C
Monitoring Setting Register 0	ADCMPCR0	0x0010
Monitoring Setting Register 1	ADCMPCR1	0x0014
Conversion Result Compare Register 0	ADCMP0	0x0018
Conversion Result Compare Register 1	ADCMP1	0x001C
Conversion Result Register 0	ADREG0	0x0020
Conversion Result Register 1	ADREG1	0x0024
Conversion Result Register 2	ADREG2	0x0028
Conversion Result Register 3	ADREG3	0x002C
Conversion Result Register 4	ADREG4	0x0030
Conversion Result Register 5	ADREG5	0x0034
Conversion Result Register 6	ADREG6	0x0038
Conversion Result Register 7	ADREG7	0x003C
Conversion Result Register 8	ADREG8	0x0040
Conversion Result Register 9	ADREG9	0x0044
Conversion Result Register 10	ADREG10	0x0048
Conversion Result Register 11	ADREG11	0x004C
Reserved	--	0x0050 to 0x00AC
Timer Trigger Program Registers 0 to 3	ADTSET03	0x00B0
Timer Trigger Program Registers 4 to 7	ADTSET47	0x00B4
Timer Trigger Program Registers 8 to 11	ADTSET811	0x00B8
Software Trigger Program Registers 0 to 3	ADSSET03	0x00BC
Software Trigger Program Registers 4 to 7	ADSSET47	0x00C0
Software Trigger Program Registers 8 to 11	ADSSET811	0x00C4
Constant Conversion Program Registers0 to 3	ADASET03	0x00C8
Constant Conversion Program Registers 4 to 7	ADASET47	0x00CC
Constant Conversion Program Registers 8 to 11	ADASET811	0x00D0
Mode Setting Register 3	ADMOD3	0x00D4

Note: Access to the "Reserved" area is prohibited.

16.4 Register Descriptions

AD conversion is performed at the clock frequency selected in the ADC Clock Setting Register.

16.4.1 ADCLK (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	0	1	0	1	1	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-3	TSH[3:0]	R/W	Write as "1001".
2-0	ADCLK[2:0]	R/W	AD prescaler output (SCLK) select 000: fc (Note) 001 to 111: Reserved

Note: The AD conversion times are 1T = 74×(1/SCLK) in the 12-bit mode and T = 68×(1/SCLK) in the 10-bit mode.

16.4.2 ADMOD0 (Mode Setting 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	-	-	-	-	-	-	DACON	ADSS
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	DACON	R/W	ADC operation control 1 0: Stop 1: Operate Setting <DACON> to "1", when using the ADC.
0	ADSS	W	Software triggered conversion 0: Don't care 1: Start Setting <ADSS> to "1" starts AD conversion (software triggered conversion). Receiving trigger signal from TMRB(interrupt) starts AD conversion also. For detail setting, please read the chapter about TMRB.

16.4.3 ADMOD1 (Mode Setting 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	ADEN	-	-	-	-	-	-	ADAS
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	ADEN	R/W	AD conversion control 0: Disable 1: Enable Setting <ADEN> to "1", when using the ADC. After Setting <ADEN> to "1", setting <ADAS> to "1" starts AD conversion and repeat conversion.
6-1	−	R	Read as "0".
0	ADAS	R/W	Constant AD conversion control 0: Disable 1: Enable

16.4.4 ADMOD2 (Mode Setting 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	-	-	-	-	-	-	ADSFN	ADBFN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	ADSFN	R	Software conversion busy flag 0: Conversion completed 1: Conversion in progress The <ADSFN> is a software AD conversion busy flag. After <ADSS> was set to "1", when AD conversion is actually started, <ADSFN> is set to "1". When finished AD conversion, <ADSFN> is cleared to "0".
0	ADBFN	R	AD conversion busy flag 0: Conversion not in progress 1: Conversion in progress The <ADBFN> is an AD conversion busy flag. When AD conversion is started regardless of conversion factor (Timer, Software, Constant), <ADBFN> is set to "1". When finished AD conversion, <ADBFN> is cleared to "0".

16.4.5 ADMOD3 (Mode Setting 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	-	-	-	-	BITS		-	RCUT
After reset	0	0	0	0	0	1	0	1
	7	6	5	4	3	2	1	0
bit symbol	-	-	PMODE			-	-	-
After reset	0	1	0	1	1	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-12	−	R/W	Write as "0".
11-10	BITS[1:0]	R/W	12-bit/10-bit resolution mode selection 00: 10-bit 01: 12-bit 10 to 11: Reserved
9	−	R/W	Write as "0".
8	RCUT	R/W	ADC operation control 2 0: Operate 1: Stop Write "0" under AD conversion. By setting ADMOD3<RCUT> to "1", consumption current will be reduced.
7	−	R/W	Write as "0".
6	−	R/W	Write as "1".
5-3	PMODE[2:0]	R/W	Write as "100".
2-0	−	R/W	Write as "0".

Note: ADMOD3<PMODE[2:0]> must be set to "100". And do not change other bits in ADxMOD3 register.

16.4.6 ADCMPCR0(Monitoring Setting Register 0)

After fixing the conversion result, the interrupt signal (INTADCP) is generated.

	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	-	-	-	-	CMPCNT0			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP0EN	-	-	ADBIG0	REGS0			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as "0".
11-8	CMPCNT0[3:0]	R/W	Comparison count for determining the result 0: After every comparison 1: After two comparisons · · 15: After 16 comparisons The ADCMPCR0 and ADCMPCR1 registers are used to enable or disable comparison between an AD conversion result and the specified comparison value, to select the register to be compared with an AD conversion result and to set how many times comparison should be performed to determine the result.
7	CMP0EN	R/W	Monitoring function 0:Disable 1:Enable By setting <CMP0EN>="0"(disable), accumulated number of decision counts is cleared.
6-5	-	R	Read as "0".
4	ADBIG0	R/W	Comparison condition 0:Larger than or equal to compare register 1:Smaller than or equal to compare register Compares whether a result of analog input is larger or smaller than the compare register. Every time AD conversion, which is set in the <REGS0[3:0]>, is complete, large/small decision is performed. If the result is matched the setting in <ADBIG0>, the counter increments.
3-0	REGS0[3:0]	R/W	AD conversion result register to be compared 0000: ADREG0 / 0100: ADREG4 / 1000: ADREG8 0001: ADREG1 / 0101: ADREG5 / 1001: ADREG9 0010: ADREG2 / 0110: ADREG6 / 1010: ADREG10 0011: ADREG3 / 0111: ADREG7 / 1011: ADREG11

16.4.7 ADCMPCR1(Monitoring Setting Register 1)

After fixing the conversion result, the interrupt signal (INTADCP) is generated.

	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	-	-	-	-	CMPCNT1			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP1EN	-	-	ADBIG1	REGS1			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as "0".
11-8	CMPCNT1[3:0]	R/W	Comparison count for determining the result 0: After every comparison 1: After two comparisons · · 15: After 16 comparisons The ADCMPCR0 and ADCMPCR1 registers are used to enable or disable comparison between an AD conversion result and the specified comparison value, to select the register to be compared with an AD conversion result and to set how many times comparison should be performed to determine the result.
7	CMP1EN	R/W	Monitoring function 0:Disable 1:Enable By setting <CMP1EN>="0"(disable), accumulated number of decision counts is cleared.
6-5	-	R	Read as "0".
4	ADBIG1	R/W	Comparison condition 0:Larger than or equal to compare register 1:Smaller than or equal to compare register Compares whether a result of analog input is larger or smaller than the compare register. Every time AD conversion, which is set in the <REGS1[3:0]>, is complete, large/small decision is performed. If the result is matched the setting in <ADBIG1>, the counter increments.
3-0	REGS1[3:0]	R/W	AD conversion result register to be compared 0000: ADREG0 0001: ADREG1 0010: ADREG2 0011: ADREG3 0100: ADREG4 0101: ADREG5 0110: ADREG6 0111: ADREG7 1000: ADREG8 1001: ADREG9 1010: ADREG10 1011: ADREG11

16.4.8 ADCMP0(Conversion Result Compare 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	AD0CMP0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AD0CMP0				-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	AD0CMP0[11:0]	R/W	The value to be compared with an AD conversion result Specify the value to be compared with an AD conversion result.
3-0	-	R	Read as "0".

Note: Set the AD monitoring function to be prohibited (<CMP0EN>="0" , <CMP1EN> = "0") in advance- when this register is modified.

16.4.9 ADCMP1(Conversion Result Compare 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	AD0CMP1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AD0CMP1				-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	AD0CMP1[11:0]	R/W	The value to be compared with an AD conversion result Specify the value to be compared with an AD conversion result.
3-0	-	R	Read as "0".

Note: Set the AD monitoring function to be prohibited (<CMP0EN>="0" , <CMP1EN> = "0") in advance- when this register is modified.

16.4.10 ADREG0(Conversion Result 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	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".
15-4	ADR0[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR0	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG0 is read and is cleared when the ADREG0 is read.
0	ADR0RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR0RF> is a flag that is set when an AD conversion result is stored in the ADREG0 register and is cleared when the low-order byte of ADREG0 is read.

16.4.11 ADREG1(Conversion Result 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	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".
15-4	ADR1[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR1	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG1 is read and is cleared when the ADREG1 is read.
0	ADR1RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR1RF> is a flag that is set when an AD conversion result is stored in the ADREG1 register and is cleared when the ADREG1 is read.

16.4.12 ADREG2(Conversion Result 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	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".
15-4	ADR2[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR2	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG2 is read and is cleared when the ADREG2 is read.
0	ADR2RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR2RF> is a flag that is set when an AD conversion result is stored in the ADREG2 register and is cleared when the ADREG2 is read.

16.4.13 ADREG3(Conversion Result 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	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".
15-4	ADR3[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR3	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG3 is read and is cleared when the ADREG3 is read.
0	ADR3RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR3RF> is a flag that is set when an AD conversion result is stored in the ADREG3 register and is cleared when the ADREG3 is read.

16.4.14 ADREG4(Conversion Result 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	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".
15-4	ADR4[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR4	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG4 is read and is cleared when the ADREG4 is read.
0	ADR4RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR4RF> is a flag that is set when an AD conversion result is stored in the ADREG4 register and is cleared when the ADREG4 is read.

16.4.15 ADREG5(Conversion Result 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	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".
15-4	ADR5[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR5	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG5 is read and is cleared when the ADREG5 is read.
0	ADR5RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR5RF> is a flag that is set when an AD conversion result is stored in the ADREG5 register and is cleared when the ADREG5 is read.

16.4.16 ADREG6(Conversion Result Register 6)

	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".
15-4	ADR6[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR6	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG6 is read and is cleared when the ADREG6 is read.
0	ADR6RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR6RF> is a flag that is set when an AD conversion result is stored in the ADREG6 register and is cleared when the ADREG6 is read.

16.4.17 ADREG7(Conversion Result Register 7)

	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".
15-4	ADR7[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR7	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG7 is read and is cleared when the ADREG7 is read.
0	ADR7RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR7RF> is a flag that is set when an AD conversion result is stored in the ADREG7 register and is cleared when the ADREG7 is read.

16.4.18 ADREG8(Conversion Result Register 8)

	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	ADR8							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR8				-	-	OVR8	ADR8RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR8[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR8	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG8 is read and is cleared when the ADREG8 is read.
0	ADR8RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR8RF> is a flag that is set when an AD conversion result is stored in the ADREG8 register and is cleared when the ADREG8 is read.

16.4.19 ADREG9(Conversion Result Register 9)

	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	ADR9							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR9				-	-	OVR9	ADR9RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR9[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR9	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG9 is read and is cleared when the ADREG9 is read.
0	ADR9RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR9RF> is a flag that is set when an AD conversion result is stored in the ADREG9 register and is cleared when the ADREG9 is read.

16.4.20 ADREG10(Conversion Result Register 10)

	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	ADR10							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR10				-	-	OVR10	ADR10RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR10[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR10	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG10 is read and is cleared when the ADREG10 is read.
0	ADR10RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR10RF> is a flag that is set when an AD conversion result is stored in the ADREG10 register and is cleared when the ADREG10 is read.

16.4.21 ADREG11(Conversion Result Register 11)

	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	ADR11							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR11				-	-	OVR11	ADR11RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR11[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR11	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG11 is read and is cleared when the ADREG11 is read.
0	ADR11RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR11RF> is a flag that is set when an AD conversion result is stored in the ADREG11 register and is cleared when the ADREG11 is read.

16.4.22 ADTSET03 / ADTSET47 / ADTSET811 (Timer Trigger Program Registers)

AD conversion can be started by INTTB51 generated from Timer5(TMRB5) as a trigger. There are twelve 8-bit registers for programming timer triggers. Setting the <ENSTm> to "1" enables the ADTSETm register. The <AINSTm[4:0]> are used to select the AIN pin to be used. The numbers of the Timer Trigger Program Registers correspond to those of the AD Conversion Result Registers. When finished this AD conversion, interrupt : INTADTMR is generated.

(m=0 to 11)

Table 16-1 Select the AIN pin

<AINST00 [4:0]> ~ <AINST53 [4:0]>	AD Channel
0_0000	AIN0
0_0001	AIN1
0_0010	AIN2
0_0011	AIN3
0_0100	AIN4
0_0101	AIN5
0_0110	AIN6
0_0111	AIN7
0_1000	AIN8
0_1001	AIN9
0_1010	AIN10
0_1011	AIN11
0_1100	AIN12
0_1101	AIN13
0_1110	AIN14
0_1111	AIN15
1_0000	AIN16
1_0001	AIN17
1_0010 to 1_1111	reserved

ADTSET03: Timer Trigger Program Registers 03

	31	30	29	28	27	26	25	24
bit symbol	ENST3	-	-	AINST3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST2	-	-	AINST2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST1	-	-	AINST1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST0	-	-	AINST0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST3	R/W	ADREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST3[4:0]	R/W	AIN select Refer to "Table 16-1 Select the AIN pin".
23	ENST2	R/W	ADREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST2[4:0]	R/W	AIN select Refer to "Table 16-1 Select the AIN pin".
15	ENST1	R/W	ADREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST1[4:0]	R/W	AIN select Refer to "Table 16-1 Select the AIN pin".
7	ENST0	R/W	ADREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST0[4:0]	R/W	AIN select Refer to "Table 16-1 Select the AIN pin".

ADTSET47: Timer Trigger Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENST7	-	-	AINST7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST6	-	-	AINST6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST5	-	-	AINST5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST4	-	-	AINST4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST7	R/W	ADREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST7[4:0]	R/W	AIN select Refer to "Table 16-1 Select the AIN pin".
23	ENST6	R/W	ADREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST6[4:0]	R/W	AIN select Refer to "Table 16-1 Select the AIN pin".
15	ENST5	R/W	ADREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST5[4:0]	R/W	AIN select Refer to "Table 16-1 Select the AIN pin".
7	ENST4	R/W	ADREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST4[4:0]	R/W	AIN select Refer to "Table 16-1 Select the AIN pin".

ADTSET811: Timer Trigger Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENST11	-	-	AINST11				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST10	-	-	AINST10				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST9	-	-	AINST9				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST8	-	-	AINST8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST11	R/W	ADREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST11[4:0]	R/W	AIN select Refer to "Table 16-1 Select the AIN pin".
23	ENST10	R/W	ADREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST10[4:0]	R/W	AIN select Refer to "Table 16-1 Select the AIN pin".
15	ENST9	R/W	ADREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST9[4:0]	R/W	AIN select Refer to "Table 16-1 Select the AIN pin".
7	ENST8	R/W	ADREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST8[4:0]	R/W	AIN select Refer to "Table 16-1 Select the AIN pin".

16.4.23 ADSSET03 / ADSSET47 / ADSSET811(Software Trigger Program Registers)

AD conversion can be started by software. There are twelve 8-bit registers for programming software triggers. The numbers of the Software Trigger Program Registers correspond to those of the Conversion Result Registers. Setting the <ENSSm> to "1" enables the ADSSETm register. The <AINSSm[4:0]> are used to select the AIN pin to be used. When finished this AD conversion, interrupt :INTADSFT is generated.

(m=0 to 11)

Table 16-2 Select the AIN pin

<AINSS00 [4:0]> ~ <AINSS53 [4:0]>	AD Channel
0_0000	AIN0
0_0001	AIN1
0_0010	AIN2
0_0011	AIN3
0_0100	AIN4
0_0101	AIN5
0_0110	AIN6
0_0111	AIN7
0_1000	AIN8
0_1001	AIN9
0_1010	AIN10
0_1011	AIN11
0_1100	AIN12
0_1101	AIN13
0_1110	AIN14
0_1111	AIN15
1_0000	AIN16
1_0001	AIN17
1_0010 to 1_1111	reserved

ADSSET03: Software Trigger Program Registers 03

	31	30	29	28	27	26	25	24
bit symbol	ENSS3	-	-	AINSS3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS2	-	-	AINSS2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS1	-	-	AINSS1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS0	-	-	AINSS0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS3	R/W	ADREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS3[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
23	ENSS2	R/W	ADREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS2[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
15	ENSS1	R/W	ADREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS1[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
7	ENSS0	R/W	ADREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS0[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".

ADSSET47: Software Trigger Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENSS7	-	-	AINSS7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS6	-	-	AINSS6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS5	-	-	AINSS5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS4	-	-	AINSS4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS7	R/W	ADREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS7[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
23	ENSS6	R/W	ADREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS6[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
15	ENSS5	R/W	ADREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS5[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
7	ENSS4	R/W	ADREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS4[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".

ADSSET811: Software Trigger Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENSS11	-	-	AINSS11				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS10	-	-	AINSS10				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS9	-	-	AINSS9				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS8	-	-	AINSS8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS11	R/W	ADREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS11[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
23	ENSS10	R/W	ADREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS10[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
15	ENSS9	R/W	ADREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS9[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
7	ENSS8	R/W	ADREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS8[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".

16.4.24 ADASET03 / ADASET47 / ADASET811(Constant Conversion Program Registers)

The ADCs allow conversion triggers to be constantly enabled. There are twelve 8-bit registers for programming constant triggers. Setting the <ENSAm> to "1" enables the ADASETm register. The <AINSAm[4:0]> are used to select the AIN pin to be used. The numbers of the Constant Trigger Program Registers correspond to those of the Conversion Result Registers.

(m=0 to 11)

Table 16-3 Select the AIN pin

<AINSA00 [4:0]> ~ <AINSA53 [4:0]>	AD Channel
0_0000	:AIN0
0_0001	:AIN1
0_0010	:AIN2
0_0011	:AIN3
0_0100	:AIN4
0_0101	:AIN5
0_0110	:AIN6
0_0111	:AIN7
0_1000	:AIN8
0_1001	:AIN9
0_1010	:AIN10
0_1011	:AIN11
0_1100	:AIN12
0_1101	:AIN13
0_1110	:AIN14
0_1111	:AIN15
1_0000	:AIN16
1_0001	:AIN17
1_0010 to 1_1111	:reserved

ADASET03: Constant Conversion Program Registers03

	31	30	29	28	27	26	25	24
bit symbol	ENSA3	-	-	AINSA3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA2	-	-	AINSA2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA1	-	-	AINSA1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA0	-	-	AINSA0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA3	R/W	ADREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA3[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
23	ENSA2	R/W	ADREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA2[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
15	ENSA1	R/W	ADREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA1[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
7	ENSA0	R/W	ADREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA0[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".

ADASET47: Constant Conversion Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENSA7	-	-	AINSA7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA6	-	-	AINSA6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA5	-	-	AINSA5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA4	-	-	AINSA4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA7	R/W	ADREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA7[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
23	ENSA6	R/W	ADREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA6[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
15	ENSA5	R/W	ADREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA5[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
7	ENSA4	R/W	ADREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA4[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".

ADASET811: Constant Conversion Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENSA11	-	-	AINSA11				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA10	-	-	AINSA10				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA9	-	-	AINSA9				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA8	-	-	AINSA8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA11	R/W	ADREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA11[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
23	ENSA10	R/W	ADREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA10[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
15	ENSA9	R/W	ADREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA9[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
7	ENSA8	R/W	ADREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA8[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".

16.5 Operation Descriptions

16.5.1 Analog Reference Voltages

For the High-level and Low-level analog reference voltages, the AVDD5and AVSSpins are used . If ADMOD3<RCUT> is set to "1", current flowing between AVDD5 and AVSS can be controlled to reduce consumption current. When AD converter is used, ADxMOD3<RCUT> is set to "0".

Note 1: During AD conversion, do not change the output data of port H/I/J, to avoid the influence on the conversion result.

Note 2: AD conversion results might be unstable by the following conditions.
Input operation is executed.
Output operation is executed.
Output current of port varies.
Take a countermeasure such as averaging the multiple conversion results, to get precise value.

16.5.2 Starting AD Conversion

AD conversion is started by software or one of the following three trigger signals.

- PMD(MPT) trigger
- Timer trigger
- Software trigger

These start triggers are given priorities as shown below.

PMD trigger 0 > ··· · > PMD trigger 3 > Timer trigger > Software trigger > constant trigger

If the PMD trigger occurs while an AD conversion is in progress, the PMD trigger is handled stop the ongoing program and start AD conversion correspond to PMD trigger number.

If a higher-priority trigger occurs while an AD conversion is in progress, the higher-priority trigger is handled after the ongoing program is completed.

It has some delay from generation of trigger to start of AD conversion. The delay depends on the trigger. The following timing chart and table show the delay.

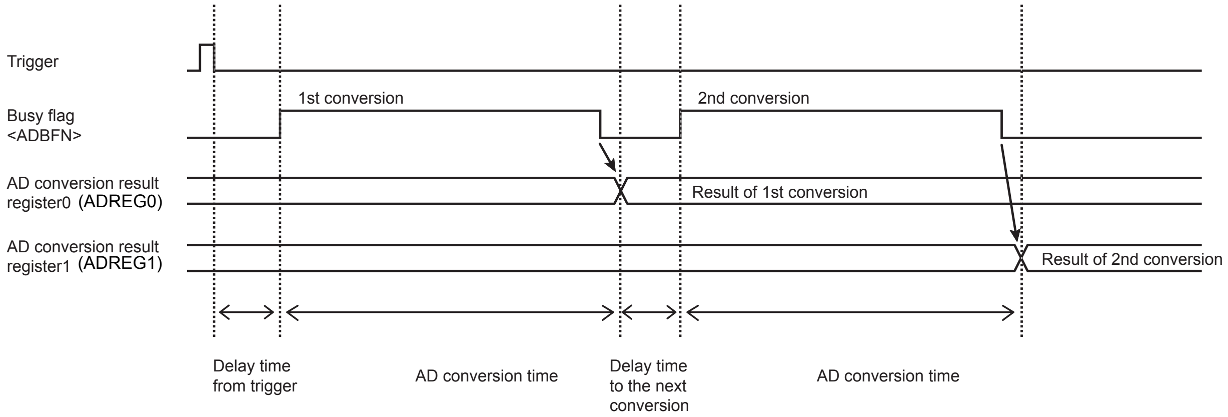


Figure 16-2 Timing chart of AD conversion

Table 16-4 AD conversion time (SCLK = 40MHz, unit : μs)

		fsys = 40MHz	
	Trigger	MIN	MAX
Delay time from trigger (Note 1)	PMD (MPT)	0.225	0.3
	TMRB	0.225	0.5
	Software, Constant	0.25	0.525
AD conversion time	−	1.85	
Delay time to the next conversion (Note2)	PMD (MPT)	0.175	0.225
	TMRB, Software, Constant	0.175	0.425

Note 1: Delay time from trigger to start of AD conversion.

Note 2: Delay time to the 2nd or after conversion in plural conversions with one trigger.

16.5.3 AD Conversion Monitoring Function

The ADCs have the AD conversion monitoring function. When this function is enabled, an interrupt is generated when a conversion result matches the specified comparison value.

To enable the monitoring function, set ADCMPCR0<CMP0EN> or ADCMPCR1<CMP1EN> to "1". In the monitoring function, if the value of AD conversion result register to which the monitoring function is assigned corresponds to the comparison condition specified by ADCMCR<ADBIG0>, the interrupt (INTADCPA for ADCMPCR0, INTADCPB for ADCMPCR1) is generated. The comparison is executed at the timing of storing the conversion result into the register.

Note 1: The AD conversion result store flag (ADRxRF) is not cleared by the comparison function.

Note 2: The comparison function differs from reading the conversion result by software. Therefore, if the next conversion is completed without reading the previous result, the overrun flag (OVRx) is set.

16.6 Timing chart of AD conversion

The following shows a timing chart of software trigger conversion, constant conversion and acceptance of trigger.

16.6.1 Software trigger Conversion

In the software trigger conversion, the interrupt is generated after completion of conversion programmed by ADSSET03, ADSSET47 and ADSSET811.(Figure 16-3)

If the ADMOD1<ADEN> is cleared to "0" during AD conversion, the ongoing conversion stops without storing to the result register.(Figure 16-4)

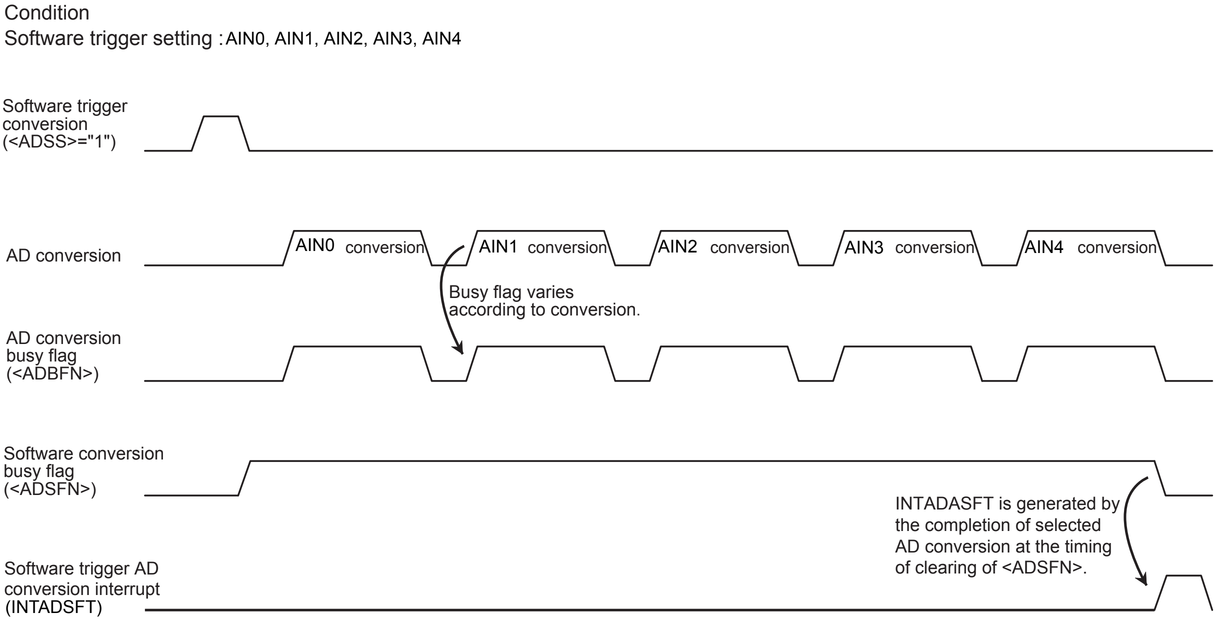

Condition
Software trigger setting :AIN0, AIN1, AIN2, AIN3, AIN4
Software trigger conversion (<ADSS>="1")
AD conversion
AIN0 conversion
AIN1 conversion
AIN2 conversion
AIN3 conversion
AIN4 conversion
Busy flag varies according to conversion.
AD conversion busy flag (<ADBFN>)
Software conversion busy flag (<ADSFN>)
INTADASFT is generated by the completion of selected AD conversion at the timing of clearing of <ADSFN>.
Software trigger AD conversion interrupt (INTADSFT)

Figure 16-3 Software trigger AD conversion

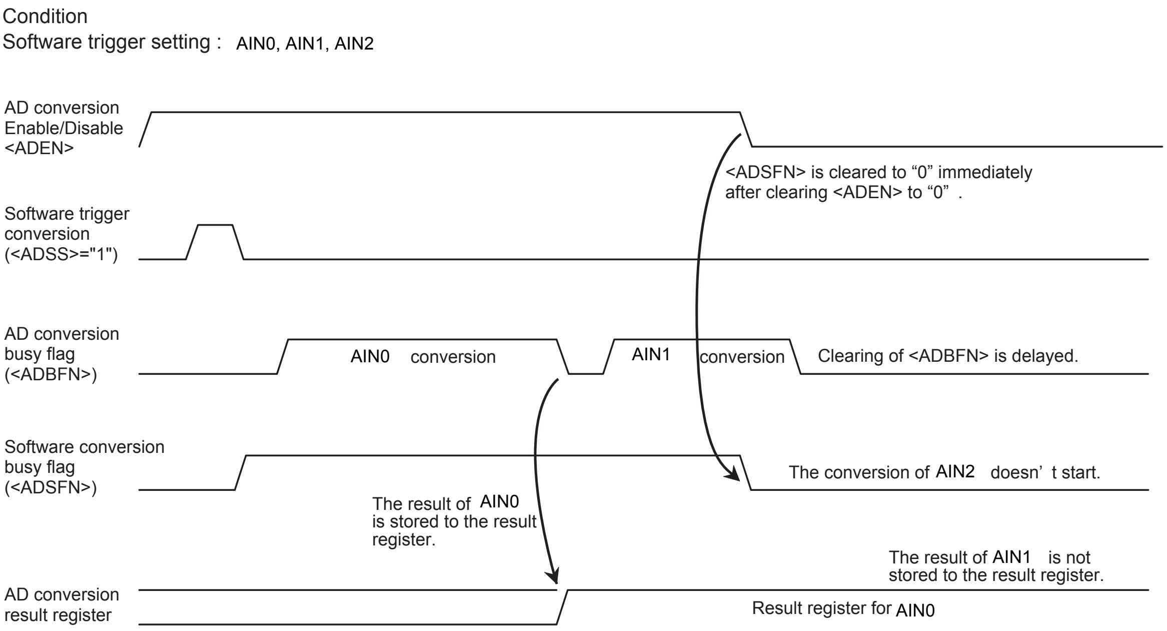

Condition
Software trigger setting : AIN0, AIN1, AIN2
AD conversion Enable/Disable <ADEN>
<ADSFN> is cleared to “0” immediately after clearing <ADEN> to “0” .
Software trigger conversion (<ADSS>="1")
AD conversion busy flag (<ADBFN>)
AIN0 conversion
AIN1 conversion
Clearing of <ADBFN> is delayed.
Software conversion busy flag (<ADSFN>)
The conversion of AIN2 doesn’ t start.
The result of AIN0 is stored to the result register.
The result of AIN1 is not stored to the result register.
AD conversion result register
Result register for AIN0

Figure 16-4 Writing "0" to <ADEN> during the software trigger AD conversion

16.6.2 Constant Conversion

In the constant conversion, if the next conversion completes without reading the previous result from the conversion result register, the overrun flag is set to "1". In this case, the previous conversion result in the conversion result register is overwritten by the next result. The overrun flag is cleared by reading of the conversion result.(Figure 16-5)

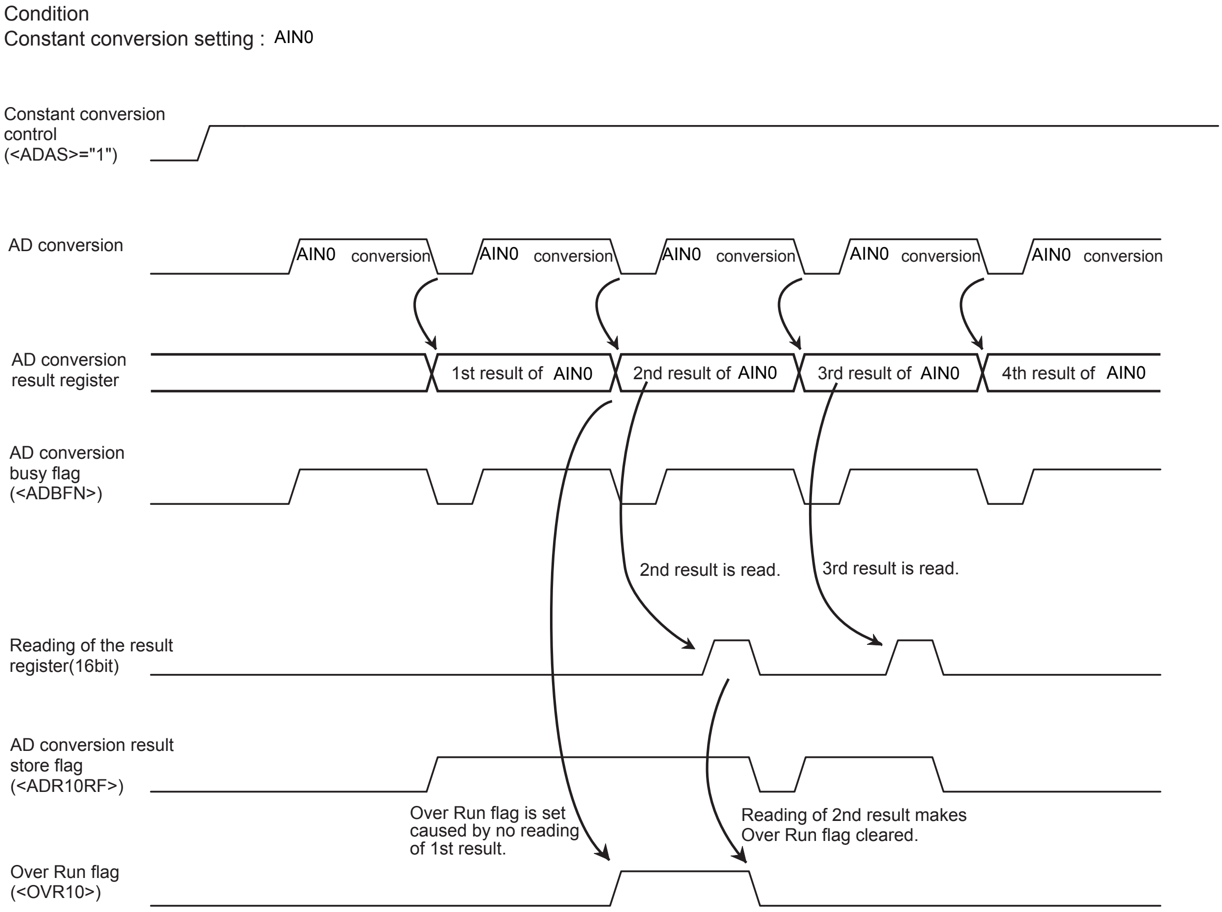

Condition
Constant conversion setting : AIN0
Constant conversion control (<ADAS>="1")
AD conversion
AIN0 conversion
AIN0 conversion
AIN0 conversion
AIN0 conversion
AIN0 conversion
AD conversion result register
1st result of AIN0
2nd result of AIN0
3rd result of AIN0
4th result of AIN0
AD conversion busy flag (<ADBFN>)
2nd result is read.
3rd result is read.
Reading of the result register(16bit)
AD conversion result store flag (<ADR10RF>)
Over Run flag is set caused by no reading of 1st result.
Reading of 2nd result makes Over Run flag cleared.
Over Run flag (<OVR10>)

Figure 16-5 Constant conversion

16.6.3 AD conversion by trigger

If the PMD trigger is occurred during the software trigger conversion, the ongoing conversion stops immediately.(Figure 16-6) If the timer trigger is occurred during the software trigger conversion, the ongoing conversion stops after the completion of ongoing conversion. (Figure 16-7) After the completion of conversion by trigger, the software trigger conversion starts from the beginning programmed by ADSSET03, ADSSET47 and ADSSET811.(Figure 16-8)

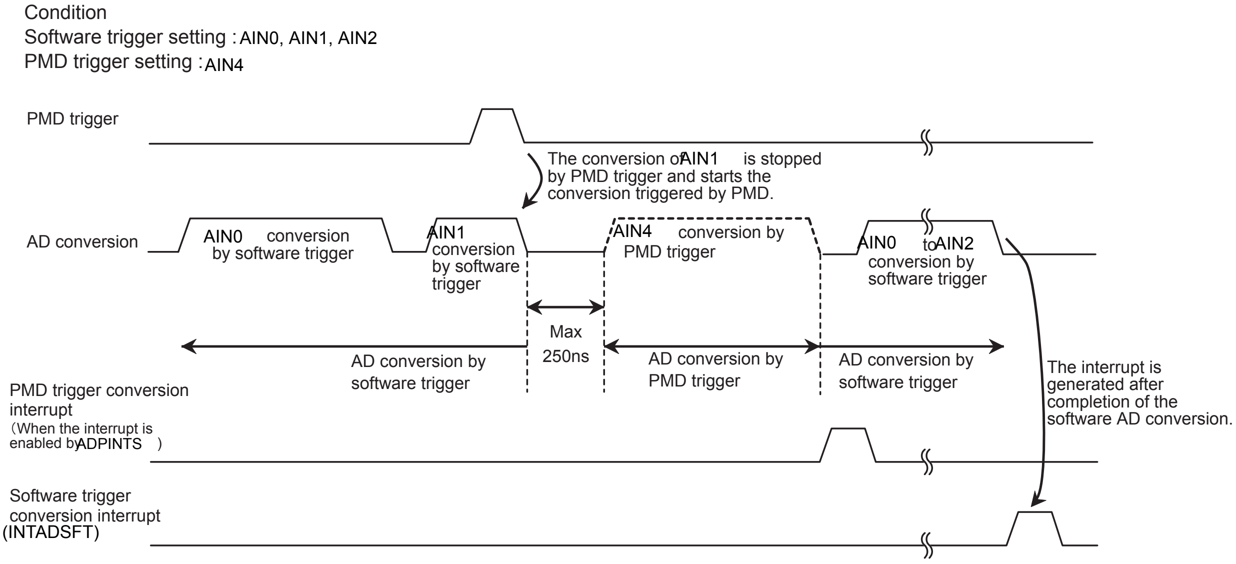

Condition
Software trigger setting : AIN0, AIN1, AIN2
PMD trigger setting : AIN4
PMD trigger
The conversion of AIN1 is stopped by PMD trigger and starts the conversion triggered by PMD.
AD conversion
AIN0 conversion by software trigger
AIN1 conversion by software trigger
AIN4 conversion by PMD trigger
AIN0 to AIN2 conversion by software trigger
Max 250ns
AD conversion by software trigger
AD conversion by PMD trigger
AD conversion by software trigger
The interrupt is generated after completion of the software AD conversion.
PMD trigger conversion interrupt (When the interrupt is enabled by ADPINTS)
Software trigger conversion interrupt (INTADSFT)

Figure 16-6 AD conversion by PMD trigger

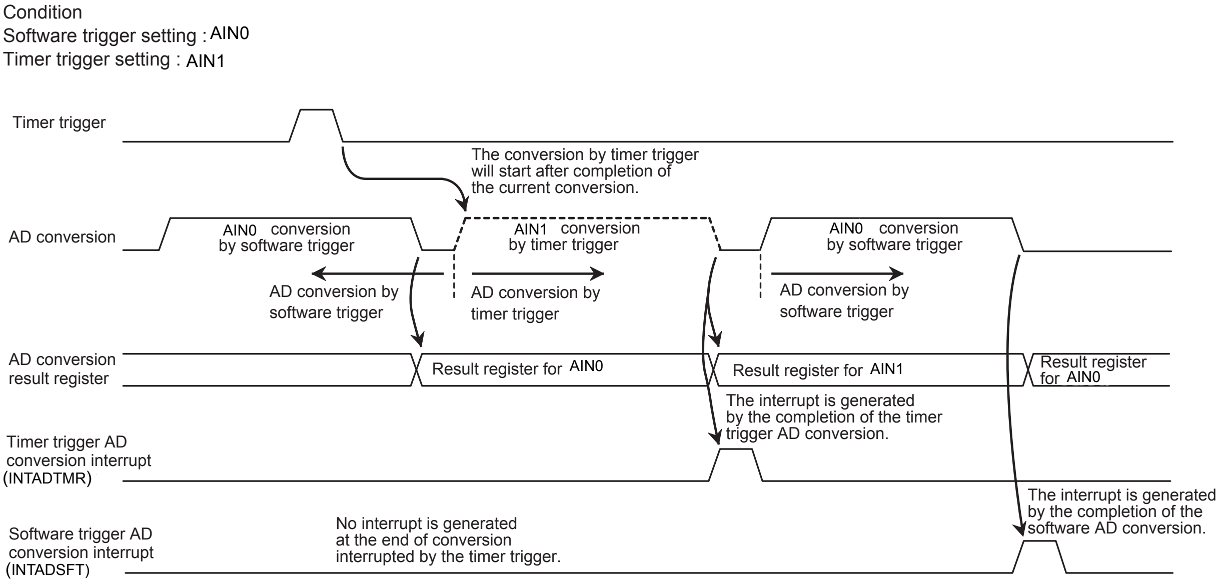

Condition
Software trigger setting : AIN0
Timer trigger setting : AIN1
Timer trigger
The conversion by timer trigger will start after completion of the current conversion.
AD conversion
AIN0 conversion by software trigger
AIN1 conversion by timer trigger
AIN0 conversion by software trigger
AD conversion by software trigger
AD conversion by timer trigger
AD conversion by software trigger
AD conversion result register
Result register for AIN0
Result register for AIN1
Result register for AIN0
The interrupt is generated by the completion of the timer trigger AD conversion.
Timer trigger AD conversion interrupt (INTADTMR)
The interrupt is generated by the completion of the software AD conversion.
Software trigger AD conversion interrupt (INTADSFT)
No interrupt is generated at the end of conversion interrupted by the timer trigger.

Figure 16-7 AD conversion by timer trigger (1)

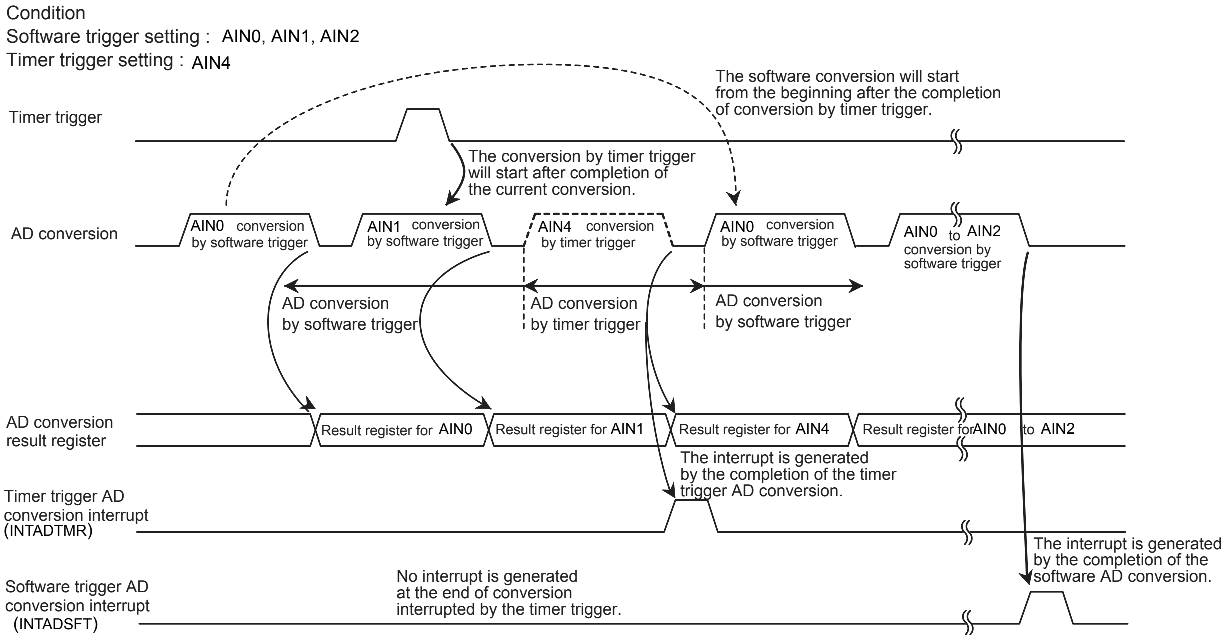

Condition
Software trigger setting : AIN0, AIN1, AIN2
Timer trigger setting : AIN4
Timer trigger
The software conversion will start from the beginning after the completion of conversion by timer trigger.
The conversion by timer trigger will start after completion of the current conversion.
AD conversion
AIN0 conversion by software trigger
AIN1 conversion by software trigger
AIN4 conversion by timer trigger
AIN0 conversion by software trigger
AIN0 to AIN2 conversion by software trigger
AD conversion by software trigger
AD conversion by timer trigger
AD conversion by software trigger
AD conversion result register
Result register for AIN0
Result register for AIN1
Result register for AIN4
Result register for AIN0 to AIN2
The interrupt is generated by the completion of the timer trigger AD conversion.
Timer trigger AD conversion interrupt (INTADTMR)
The interrupt is generated by the completion of the software AD conversion.
Software trigger AD conversion interrupt (INTADSFT)
No interrupt is generated at the end of conversion interrupted by the timer trigger.

Figure 16-8 AD conversion by timer trigger (2)

17. Real Time Clock (RTC)

17.1 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

17.2 Block Diagram

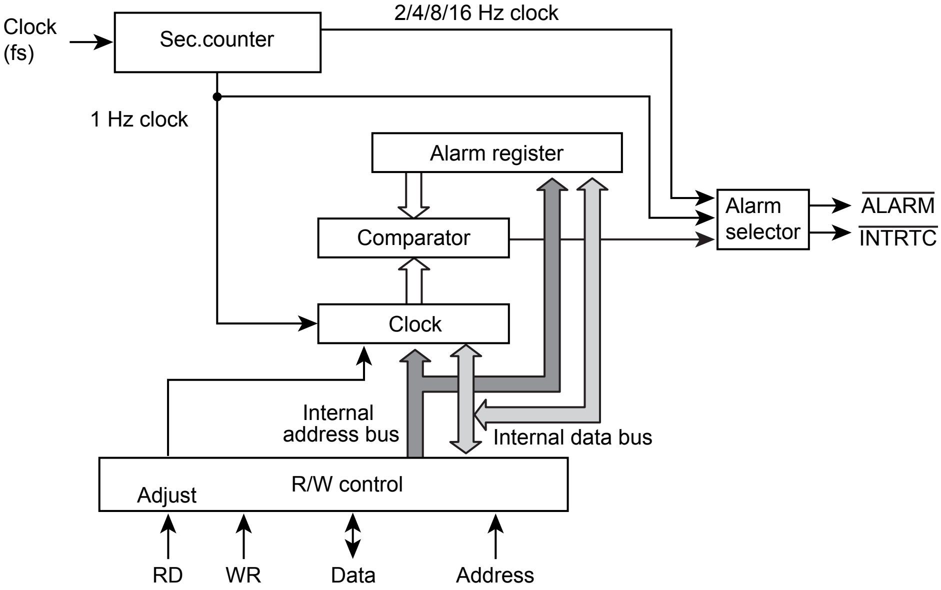


Figure 17-1 Block Diagram

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.

17.3 Detailed Description Register

17.3.1 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.

17.3.2 Control Register

Reset operation initializes the following registers.

- RTCPAGER<PAGE>, <ADJUST>, <INTENA>
- RTCRESTR<RSTALM>, <RSTTMR>, <DIS16HZ>, <DIS1HZ>, <DIS2HZ>, <DIS4HZ>, <DIS8HZ>

Other clock-related registers are not initialized by reset operation.

Before using 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 "17.4.3 Entering the Low Power Consumption Mode" for more information.

Note: In this product, external oscillation connect pins (XT1/XT2) are dual-purpose ports with general-purpose ports. If these pins are used as an external oscillation connect pin, refer to the setting procedure in Chapter Clock/mode Control.

Table 17-1 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	−	2Hz enable	4Hz enable	8Hz enable	Reset register

Note: Reading RTCSECR, RTCMINR, RTCHOURR, RTCDAYR, RTCMONTHR, RTCYEARR of PAGE0 captures the current state.

Table 17-2 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	−	2Hz enable	4Hz enable	8Hz enable	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.

17.3.3 Detailed Description of Control Register

17.3.3.1 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	Function
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.

17.3.3.2 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	Function
7	−	R	Read as 0.
6-0	MI	R/W	Setting digit register of Minutes. 000_0000 : 00min. 001_0000 : 10min. 010_0000 : 20min. 111_1111 : don’t care 000_0001 : 01min. 001_0001 : 11min. · (Only PAGE1) 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.

17.3.3.3 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	Function
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 11_1111 : don't care (Only PAGE1) 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	Function
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 11_1111 : don't care (Only PAGE1) 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.

17.3.3.4 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 111: don’t care (Only PAGE1)

Note: The setting other than listed above is prohibited.

17.3.3.5 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	Function
7-6	−	R	Read as 0.
5-0	DA	R/W	Setting digit register of day. 00_0001 : 1st day 00_0010 : 2nd day 00_0011 : 3rd day 00_0100 : 4th day 00_0101 : 5th day 00_0110 : 6th day 00_0111 : 7th day 00_1000 : 8th day 00_1001 : 9th day 01_0000 : 10th day 01_0001 : 11th day 01_0010 : 12th day 01_0011 : 13th day 01_0100 : 14th day 01_0101 : 15th day 01_0110 : 16th day 01_0111 : 17th day 01_1000 : 18th day 01_1001 : 19th day 10_0000 : 20th day 10_0001 : 21th day 10_0010 : 22th day 10_0011 : 23th day 10_0100 : 24th day 10_0101 : 25th day 10_0110 : 26th day 10_0111 : 27th day 10_1000 : 28th day 10_1001 : 29th day 11_0000 : 30th day 11_0001 : 31th day 11_1111 : don’t care (Only PAGE1)

Note 1: The setting other than listed above is prohibited.

Note 2: Do not set for non-existent days (e.g. 30th Feb.).

17.3.3.6 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	Function
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.

17.3.3.7 RTCMONTHR (Selection of 24-hour clock or 12-hour clock (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.

17.3.3.8 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 year 0001_0000 : 10 years 0110_0000 : 60 years 0000_0001 : 01 years ・ ・ 0000_0010 : 02 years 0010_0000 : 20 years 0111_0000 : 70 years 0000_0011 : 03 years ・ ・ 0000_0100 : 04 years 0011_0000 : 30 years 1000_0000 : 80 years 0000_0101 : 05 years ・ ・ 0000_0110 : 06 years 0100_0000 : 40 years 1001_0000 : 90 years 0000_0111 : 07 years ・ ・ 0000_1000 : 08 years 01001_0000 : 50 years ・ 0000_1001 : 09 years ・ 1001_1001 : 99 years

Note: The setting other than listed above is prohibited.

17.3.3.9 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	Function
7-2	−	R	Read as 0.
1-0	LEAP	R/W	00 : A leap year 01 : one year after a leap year 10 : two years after a leap year 11 : three years after a leap year

17.3.3.10 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 performed.

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

17.3.3.11 RTCRESTR (Reset register (for PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	-	DIS2HZ	DIS4HZ	DIS8HZ
After reset	1	1	0	0	0	1	1	1

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 request 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	DIS2HZ	R/W	2 Hz 0: Enable 1: Disable
1	DIS4HZ	R/W	4 Hz 0: Enable 1: Disable
0	DIS8HZ	R/W	8 Hz 0: Enable 1: Disable

Note: A read-modify-write operation cannot be performed.

The setting of <DIS1HZ>, <DIS2HZ>, <DIS4HZ>, <DIS8HZ>, <DIS16MHZ> and RTCPAGER<ENAALM> used for alarm, 1Hz interrupt, 2Hz interrupt, 4Hz interrupt, 8Hz interrupt and 16Hz interrupt are shown as below.

Table 17-3 Select interrupt source signal

<DIS1HZ>	<DIS2HZ>	<DIS4HZ>	<DIS8HZ>	<DIS16HZ>	RTCPAGER <ENAALM>	Interrupt source signal
1	1	1	1	1	1	ALARM
0	1	1	1	1	0	1 Hz
1	0	1	1	1	0	2 Hz
1	1	0	1	1	0	4Hz
1	1	1	0	1	0	8Hz
1	1	1	1	0	0	16 Hz
Others						Interrupt not generated

17.4 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.

Note: After reset, a low-speed clock stops oscillation. XT1 and XT2 ports are initialized to PP0 and PP1 ports. Re-set-up RTC register.

17.4.1 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.

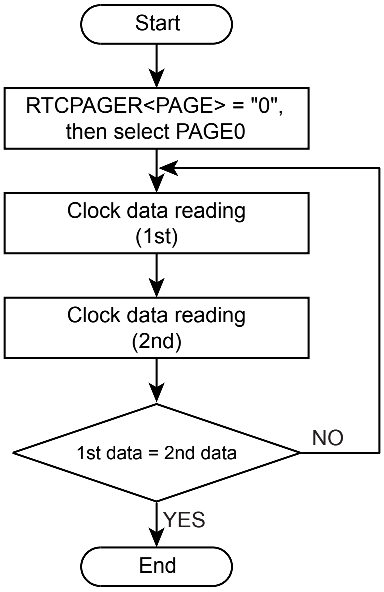


Figure 17-2 Flowchart of the clock data reading

17.4.2 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.

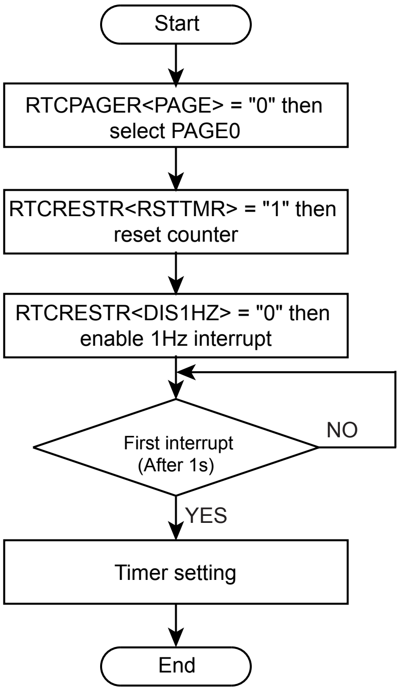

Start
RTCPAGER<PAGE> = "0" then select PAGE0
RTCRESTR<RSTTMR> = "1" then reset counter
RTCRESTR<DIS1HZ> = "0" then enable 1Hz interrupt
First interrupt (After 1s)
NO
YES
Timer setting
End

Figure 17-3 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

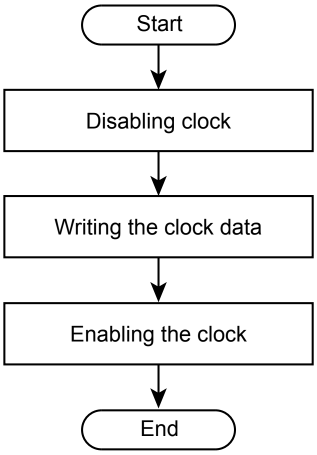


Figure 17-4 Flowchart of the disabling clock

17.4.3 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.

17.5 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. Outputting Low-pulse (1 Hz, 2 Hz, 4 Hz, 8 Hz or 16 Hz)

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

17.5.1 "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.

17.5.2 Outputting Low-pulse (1 Hz, 2 Hz, 4 Hz, 8 Hz or 16 Hz)

When RTCPAGER<ENAALM> and RTCRESTR are set as Table 17-3 and then RTCPAGER<INTENA> = "1" is set, one cycle low-speed (1 Hz, 2 Hz, 4 Hz, 8 Hz or 16 Hz) low pulse is output to ALARM pin. At the same time, INTRTC interrupt is also output.

18. Power-on-Reset Circuit (POR)

The power-on-reset circuit (POR) generates a power-on reset signal when power-on.

Power supply voltage is indicated as RVDD5.

18.1 Structure

Power-on-reset circuit consists of the reference voltage generation circuit, comparators, the VLTD reset circuit and the power-on counter.

This circuit compares a voltage divided by the ladder resistor with a reference voltage generated in the reference voltage generation circuit in the comparator.

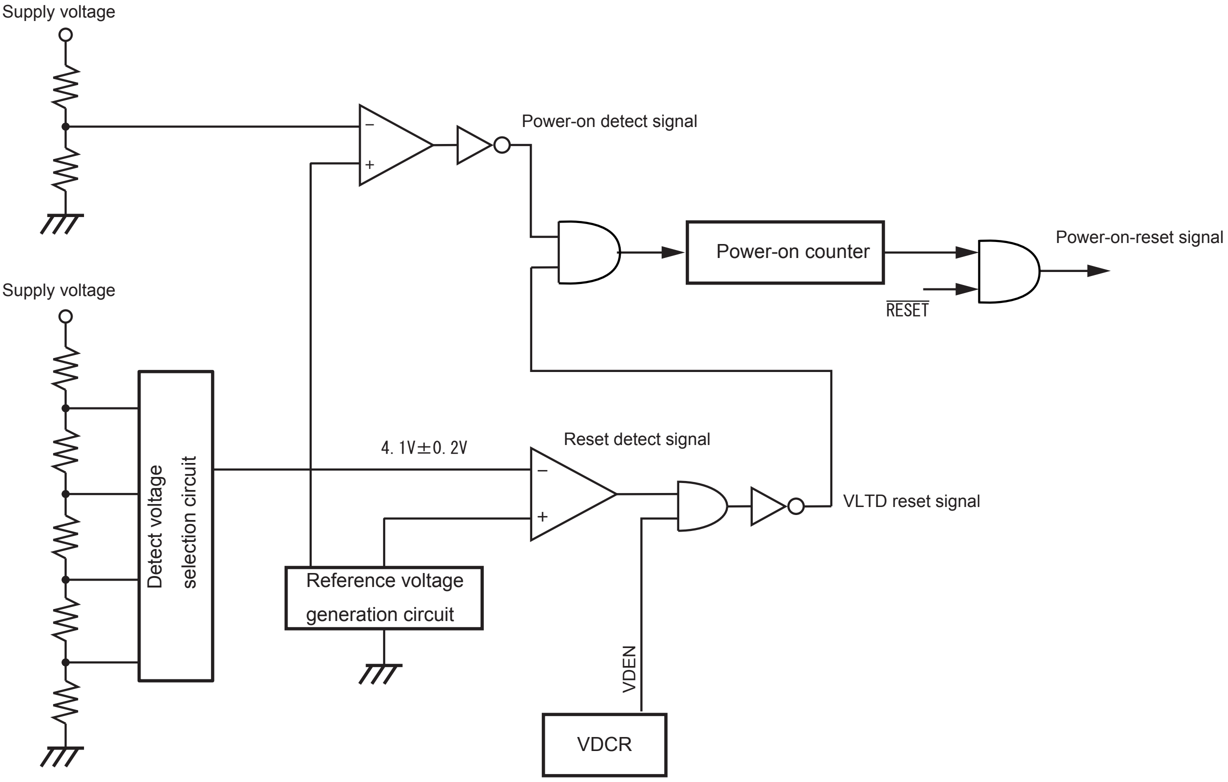


Figure 18-1 Power-on-reset circuit

For details of VDCR in VLTD reset circuit, refer to Section "Voltage detection circuit (VLTD)".

18.2 Function

At power-on, a power-on detection signal generates while power supply voltage is lower than the releasing voltage. Power-on detection signal is released at the timing when RVDD5 is over 3.0 ±0.2 V.

If the power-on detection signal is released and the VLTD signal is also released, the power-on counter starts to operate. After waiting time (approximately 0.9 ms) has elapsed, the power-on reset signal is released.

During the power-on reset signal generation, the CPU and the peripheral functions are reset.

When a reset input is not used, supply voltage must be raised to the recommended operational voltage range until the power-on reset releasing. If power supply voltage does not reach to the recommended operational voltage range during this period, TMPM381/383 cannot operate properly.

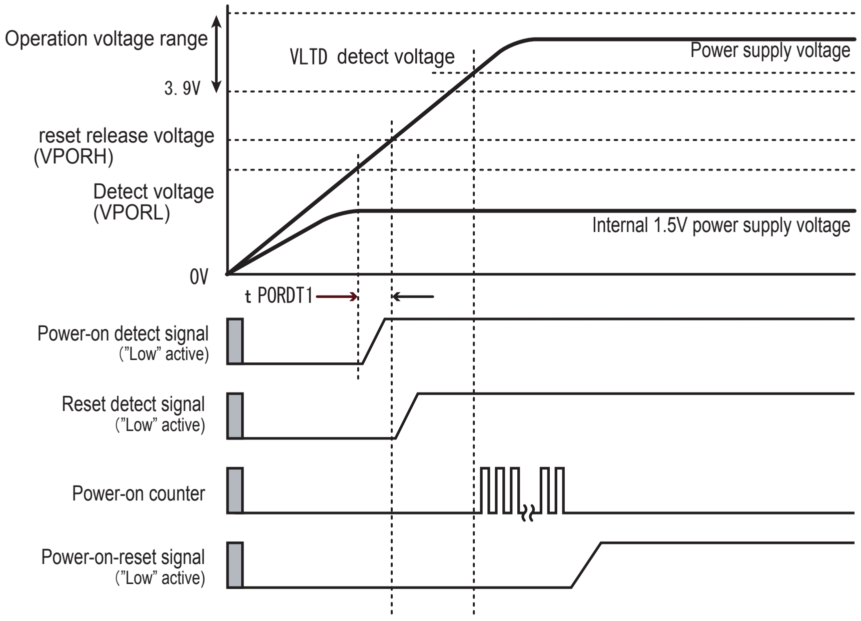


Figure 18-2 Power-on-reset operation timing

Symbol	Parameter	Min	Typ	Max	Unit
tPWUP	Power-on Counter	-	$2^{13}/f_{OSC2}$	-	s
tDVDD	Rising time of power line	-	-	3	ms
VPORH	Power-on Reset releasing voltage	2.8	3	3.2	V
VPORL	Power-on Reset detection voltage	2.6	2.8	3.0	V
tPORDT1	Power-on Reset release response time		30		μs

Note: Since the power-on releasing voltage and the power-on reset detection voltage relatively change,the detection voltage is never reversed.

Note: Due to the fluctuation of supply voltage, the power-on reset circuit may not operate properly. Users should give due consideration based on the electrical characteristic in the device designing.

19. Low Voltage Detection Circuit (VLTD)

The low voltage detection circuit generates a reset signal by detecting a decreasing voltage.

Note: Due to the fluctuation of supply voltage, the voltage detection circuit may not operate properly. Users should give due consideration based on the electrical characteristic in the device designing.

19.1 Structure

The low voltage detection circuit consists of a reference voltage generation circuit, a detection voltage level selection circuit, a comparator and control registers.

The supply voltage is divided by the ladder resistor and input to the detection voltage selection circuit. The detection voltage selection circuit selects a voltage according to the specified detection voltage (4.1 ± 0.2V), and the comparator compares it with the reference voltage. When the supply voltage becomes lower than the detection voltage (4.1 ± 0.2V), a voltage detection reset signal is generated.

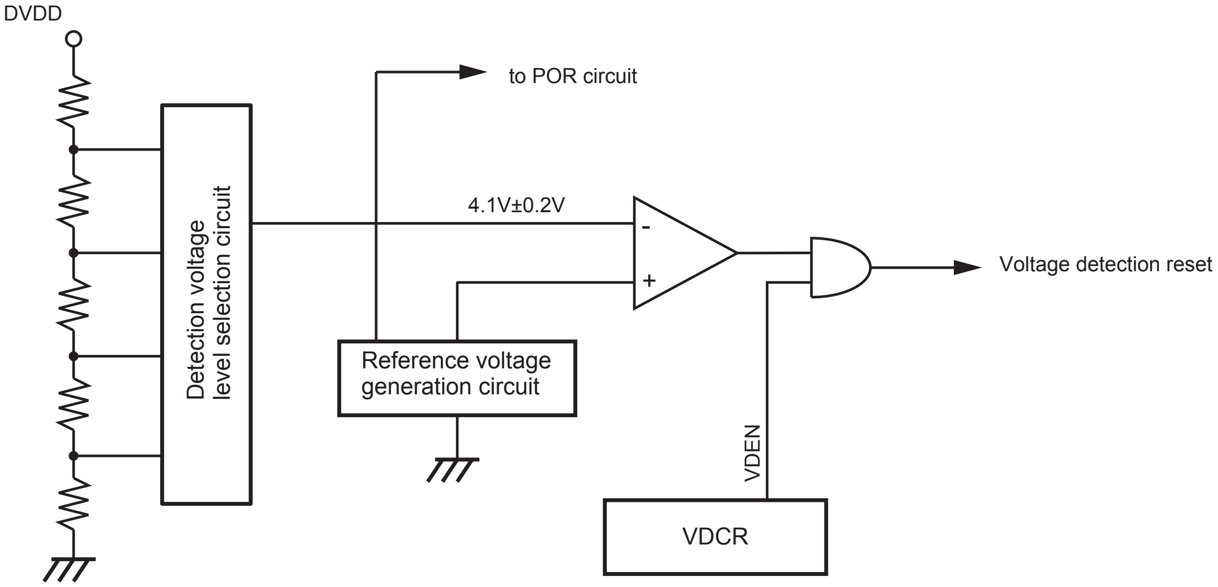

DVDD
to POR circuit
Detection voltage level selection circuit
4.1V±0.2V
-
+
Voltage detection reset
Reference voltage generation circuit
VDEN
VDCR

Figure 19-1 Voltage Detection Circuit)

19.2 Registers

19.2.1 Register List

Base Address = 0x4004_0900

Register name		Address(Base+)
Voltage detection control register	VDCR	0x0000

19.2.2 VDCR (Voltage detection 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	−	−	−	−	−	−		VDEN
After reset	0	0	0	0	0	0	1	1

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as "0".
2-1	−	R/W	Write "01"
0	VDEN	R/W	Low voltage detection operation 0: Disabled 1: Enabled

Note: VDCR is initialized by a power-on reset or an external reset input.

19.3 Operation Description

19.3.1 Control

The voltage detection circuit is controlled by voltage detection control registers.

19.3.2 Function

Enabling/disabling the voltage detection can be programmed by VDCR<VDEN>. After the voltage detection operation is enabled, When the supply voltage becomes lower than the detection voltage (4.1 ± 0.2V), a voltage detection reset signal is generated.

19.3.2.1 Enabling/disabling the voltage detection operation

Setting VDCR<VDEN> to "1" enables the voltage detection operation. Setting it to "0" disables the operation. VDCR<VDEN> is set to "1" immediately after a power-on reset or a reset by an external reset input is released.

Note: When the supply voltage is lower than the detection voltage (4.1 ± 0.2V), setting VDCR<VDEN> to "1" generates reset signal at the time.

19.3.2.2 Selecting the detection voltage level

A detection voltage is 4.1 ± 0.2V.

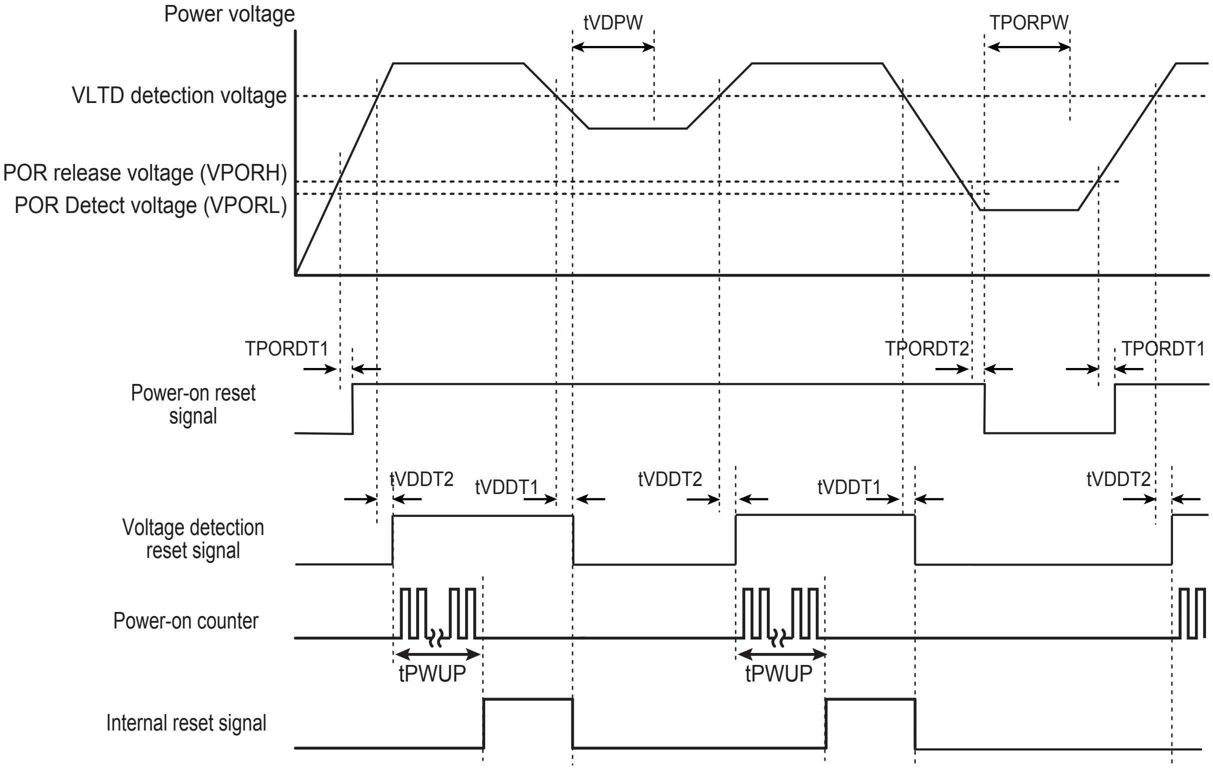


Figure 19-2 Voltage Detection Timing

Symbol	Parameter	Min	Typ	Max	Unit
tVDEN	Setup time after enabling voltage detection	-	40	-	μs
tVDDT1	Voltage detection response time	-	40	-	
tVDDT2	Voltage detection releasing time‘	-	40	-	
tVDPW	Voltage detection minimum pulse width	45	-	-	
LDLVL	Detection voltage	3.9	4.1	4.3	V

20. Oscillation Frequency Detector (OFD)

The oscillation frequency detector circuit (OFD) detects abnormal clock frequency. To use the OFD, abnormal states of clock such as a harmonic, a sub harmonic or stopped state can be detected.

The OFD monitors the target clock frequency using reference frequency and generates a reset signal if abnormal state is detected. And if the reference frequency stops, a reset signal is generated, too.

TMPM381/383 uses internal high-speed oscillator clock as a reference and the target clock is an external high-speed oscillator clock.

Note: It is not guaranteed that OFD can detect all defects at any time, and it is not a circuit to measure error frequency.

20.1 Block diagram

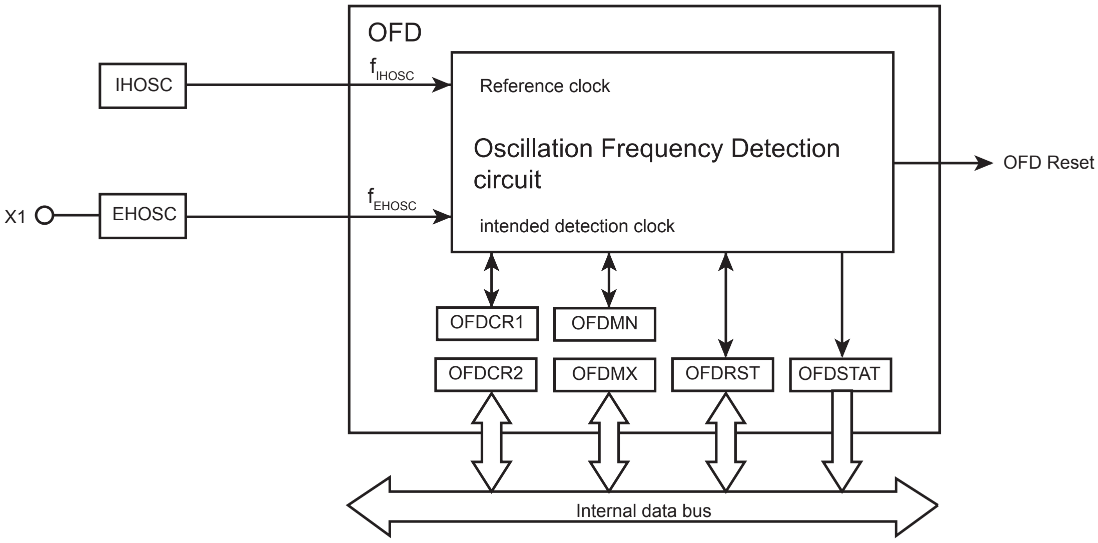


Figure 20-1 Oscillation Frequency Detector Block diagram

20.2 Registers

20.2.1 Register List

Base Address = 0x4004_0800

Register name		Address (Base+)
Control register 1	OFDCR1	0x0000
Control register 2	OFDCR2	0x0004
Lower detection frequency setting register	OFDMN	0x0008
Reserved	-	0x000C
Higher detection frequency setting register	OFDMX	0x0010
Reserved	-	0x0014
Reset control register	OFDRST	0x0018
Status register	OFDSTAT	0x001C

Note: Access to the "Reserved" area is prohibited.

20.2.1.1 OFDCR1 (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	OFDWEN							
After reset	0	0	0	0	0	1	1	0

Bit	Bit Symbol	Type	Description
31-8	−	R	Read as 0.
7-0	OFDWEN[7:0]	R/W	Controls register write 0x06: Disable 0xF9: Enable Setting 0xF9 enables to write registers except OFDCR1. When writing a value except 0x06 or 0xF9, 0x06 is written. If writing register is disabled, reading from each register is enabled.

20.2.1.2 OFDCR2 (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	OFDEN							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-8	−	R	Read as 0.
7-0	OFDEN[7:0]	R/W	Controls frequency detecting. 0x00: Disable 0xE4: Enable Writing a value except 0x00 or 0xE4 is invalid and a value will not be changed.

20.2.1.3 OFDMN (Lower detection frequency 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	OFDMN							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-8	−	R	Read as 0.
7-0	OFDMN[7:0]	R/W	Sets lower detection frequency.

Note: Writing to the register of OFDMN is protected while OFD circuit is operating.

20.2.1.4 OFDMX (Higher detection frequency 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	OFDMX							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-8	−	R	Read as 0.
7-0	OFDMX[7:0]	R/W	Sets higher detection frequency.

Note: Writing to the register of OFDMX is protected while OFD circuit is operating.

20.2.1.5 OFDRST (Reset 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	-	-	-	-	-	-	-	OFDRSTEN
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Description
31-1	−	R	Read as 0.
0	OFDRSTEN	R/W	Controls generating a reset. 0: Disable 1: Enable

Note: Writing to the register of OFDRST is protected while OFD circuit is operating.

20.2.1.6 OFDSTAT (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	-	-	-	-	-	-	OFDBUSY	FRQERR
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-2	−	R	Read as 0.
1	OFDBUSY	R	OFD operation 0: Stop 1: Run
0	FRQERR	R	Error detecting flag 0: No Error 1: Error

20.3 Operational Description

20.3.1 Setting

All register except OFDCR1 can not be written by reset. They are can be written by writing "0xF9" to OFDCR1.

To protect the mistaken writing, should be written "0x06" to OFDCR1 after setting all registers. And the register should be modified when OFD is stopped.

20.3.2 Operation

From the operation start-up to detection start-up, time length as two cycle of detecting clock is needed.

OFDSTAT<OFDBSY> can confirm whether it is operating. Detecting cycle is (reference clock frequency) / 2^8MHz.

When generating reset is enabled, the reset is generated if the following condition is satisfied.

- When a target clock frequency is over than the range of a frequency which is specified by OFDMX and OFDMN.
- When the reference clock stops.

When generating reset is disabled, OFDSTAT<FRQERR> can be confirmed the condition

Note: There are several factors of reset. Clock generator register CGRSTFLG can confirm the factors. For details of CGRSTFLG see chapter "exception."

20.3.3 Detection Frequency

The detection frequency have a detection frequency range and an undetectable frequency range because of oscillation accuracy. Therefore, it is undefined whether to be detected between detection frequency range and undetectable it.

Figure 20-2 shows the detection or undetectable frequency range when the target clock error is ±10% and the reference clock error is ± 5%.

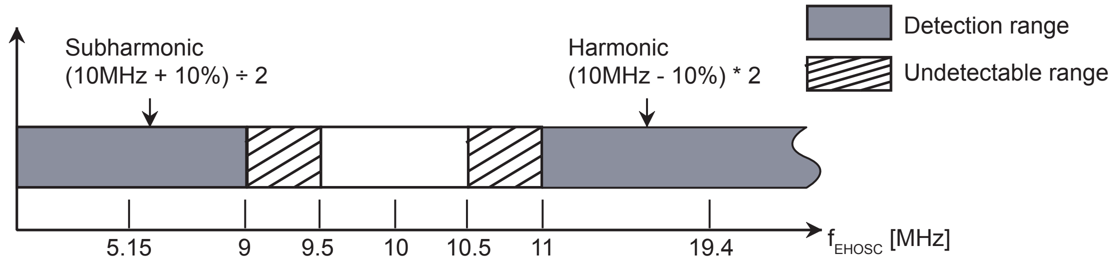


Figure 20-2 Example of detection frequency range (in case of 10MHz)

The higher and lower limit of the detection frequency is calculated from the maximum error of the target clock and the reference.

How to calculate the setup value of OFDMN/OFDMX is shown below when the target clock error is ±10% and the reference clock error is ± 5%.

target clock (f_{EHOSC})	10MHz ± 10%	Max.11MHz	-------- a
		Min. 9MHz	-------- b
reference clock (f_{IHOSC})	10MHz ± 5%	Max.10.5MHz	-------- c
		Min. 9.5MHz	-------- d

higher limit of the detection frequency = 1 ÷ {(d ÷ 2^8) ÷ (a ÷ 4)}

lower limit of the detection frequency = 1 ÷ {(c ÷ 2^8) ÷ (b ÷ 4)}

higher limit of the detection frequency = 1 ÷ {(9.5 × 10^6 ÷ 2^8) ÷ (11 × 10^6 ÷ 4)} = 74.10 = 74 (truncate after the decimal places)= 0x4a

lower limit of the detection frequency = 1 ÷ {(10.5 × 10^6 ÷ 2^8) ÷ (9 × 10^6 ÷ 4)} = 54.85 = 55 (round up after the decimal places)= 0x37

Setting "0x4a" to the register OFDMX and "0x37" to the register OFDMN, when the external oscillation of higher than 11MHz or lower than 9MHz is detected, the oscillation frequency detector outputs a reset signal.

20.3.4 Available Operation Mode

The oscillation frequency detection is available only in NORMAL and IDLE mode. Before shifting to another mode, disable the oscillation frequency detection.

20.3.5 Example of Operational Procedure

The example of operational procedure is shown below.

After reset, confirms various reset factor by CGRSTFLG. If the reset factor is not by the oscillation frequency detect, enable external oscillation, set register to use OFD and enable operation. Reset output must be disabled at this time.

After waiting the OFD operation is started, confirms abnormal status flag, and if there is not abnormal status, change to external oscillation clock.

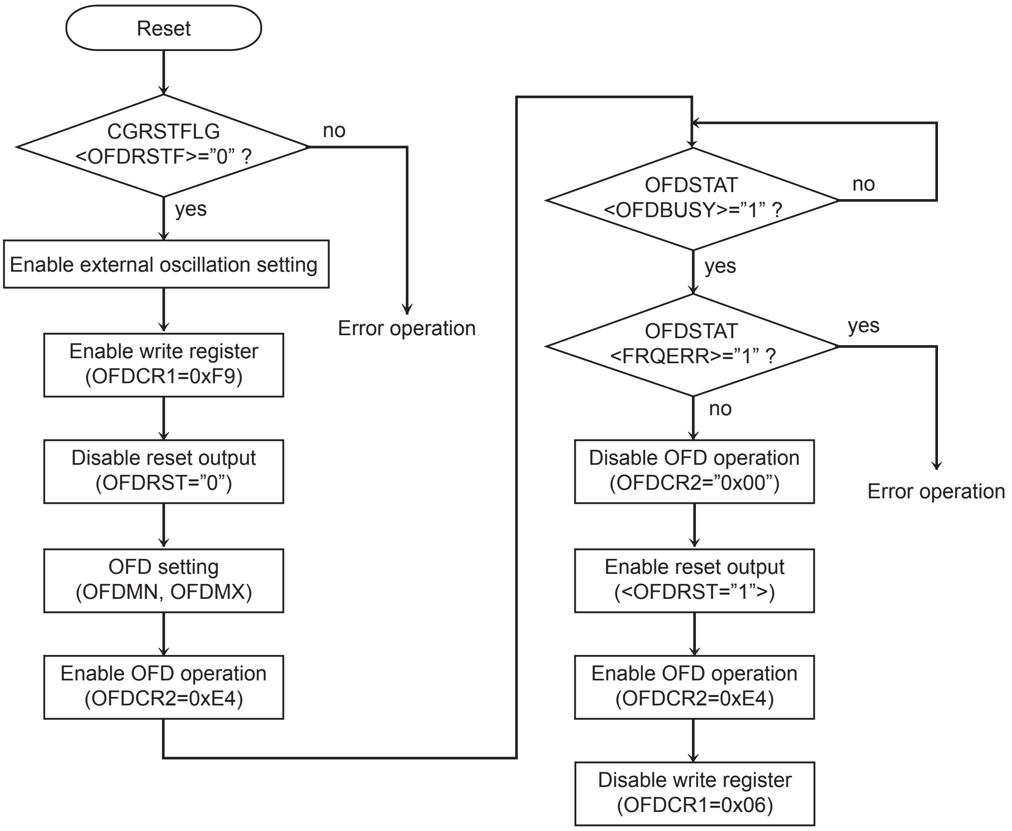

Reset
CGRSTFLG <OFDRSTF>="0" ?
no
yes
Enable external oscillation setting
Error operation
Enable write register (OFDCR1=0xF9)
Disable reset output (OFDRST="0")
OFD setting (OFDMN, OFDMX)
Enable OFD operation (OFDCR2=0xE4)
OFDSTAT <OFDBUSY>="1" ?
no
yes
OFDSTAT <FRQERR>="1" ?
yes
no
Error operation
Disable OFD operation (OFDCR2="0x00")
Enable reset output (<OFDRST="1">)
Enable OFD operation (OFDCR2=0xE4)
Disable write register (OFDCR1=0x06)

Figure 20-3 Example of operational procedure

21. 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 ($\overline{\text{WDTOUT}}$).

21.1 Configuration

Figure 21-1shows the block diagram of the watchdog timer.

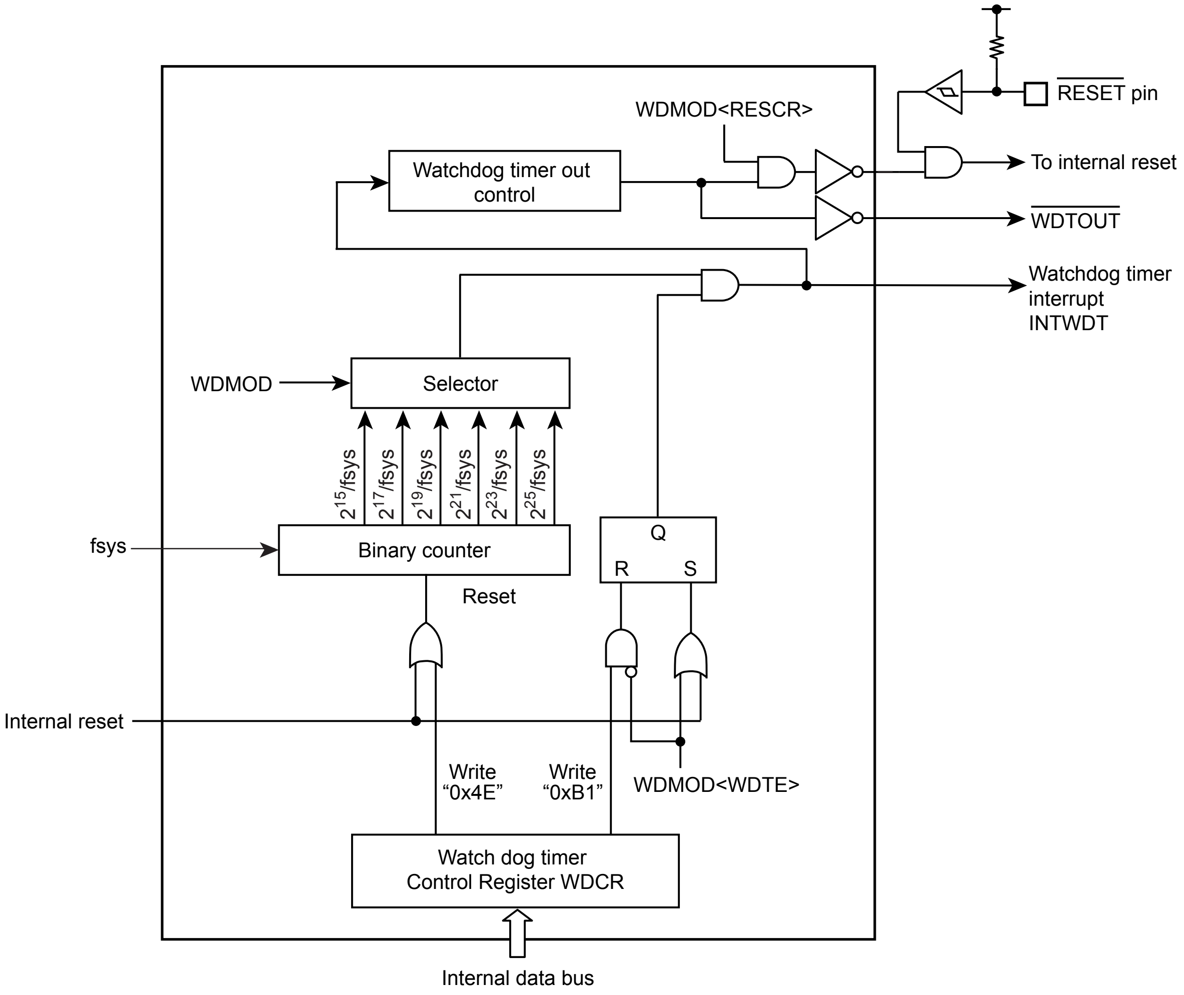


Figure 21-1 Block Diagram of the Watchdog Timer

21.2 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

21.2.1 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 21-1) 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 21-1 Detection time of watchdog timer (fc = 40MHz)

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
111 (fc/16)	13.12 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s	13.42 s

21.2.2 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

21.3 Operations

21.3.1 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 ($\overline{\text{WDTOUT}}$).

21.3.2 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 as the high-speed frequency clock is stopped. Before transition to below modes, the watchdog timer should be disabled.In IDLE mode, its operation depends on the WDMOD <I2WDT> setting.

- STOP mode
- SLEEP mode
- SLOW mode

Also, the binary counter is automatically stopped during debug mode.

21.4 Operation when malfunction (runaway) is detected

21.4.1 INTWDT interrupt generation

In the Figure 21-2 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($\overline{\text{WDTOUT}}$).

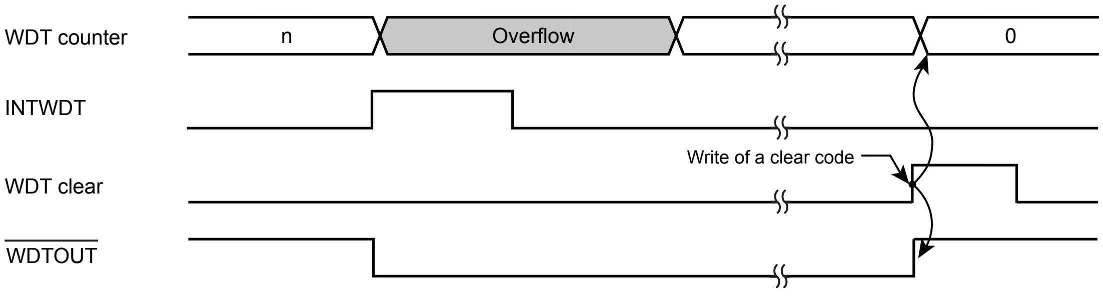


Figure 21-2 INTWDT interrupt generation

21.4.2 Internal reset generation

Figure 21-3 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 internal high-speed frequency clock (fosc). This means fsys = fosc.

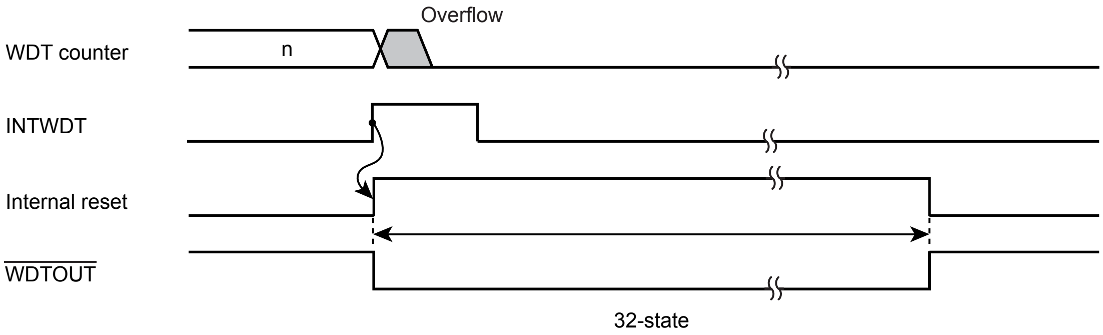


Figure 21-3 Internal reset generation

21.5 Control register

The watchdog timer (WDT) is controlled by two control registers WDMOD and WDCR.

21.5.1 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.

21.5.2 Watchdog Timer Control Register(WDCR)

This is a register for disabling the watchdog timer function and controlling the clearing function of the binary counter.

21.5.3 Setting example

21.5.3.1 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).

21.5.3.2 Enabling control

Set WDMOD <WDTE> to "1".

		7	6	5	4	3	2	1	0	
WDMOD	←	1	–	–	–	–	–	–	–	Set <WDTE> to "1".

21.5.3.3 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).

21.5.3.4 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	–	–	–	–

22. Flash Memory Operation

This section describes the hardware configuration and operation of Flash memory. In this section, "1-word" means 32 bits.

22.1 Features

22.1.1 Memory Size and Configuration

Table 22-1 and Figure 22-1 Figure 22-2show a built-in memory size and configuration of TMPM381/383.

Table 22-1 Memory size and configuration

Memory size	Block configuration				# of words per page	# of pages	Write time		Erase time	
	128 KB	64 KB	32 KB	16 KB			1 page	Total area	Block erase	Chip erase
128 KB	−	−	4	−	32	1024	1.25ms	1.28 sec	0.1sec	0.2 sec
64 KB	−	−	2	−	32	512	1.25ms	0.64 sec	0.1sec	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 used by a user.

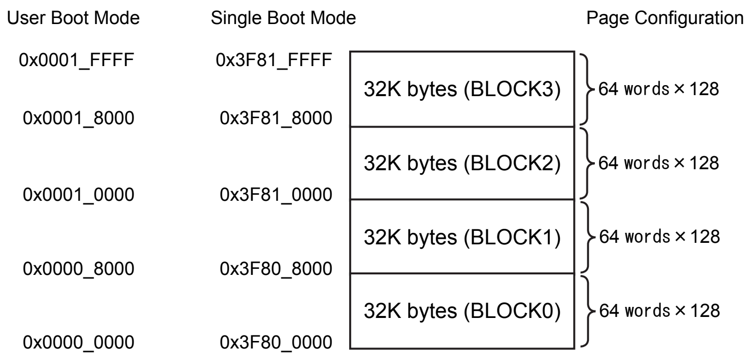


Figure 22-1 Block configuration (FLAH 128KB)

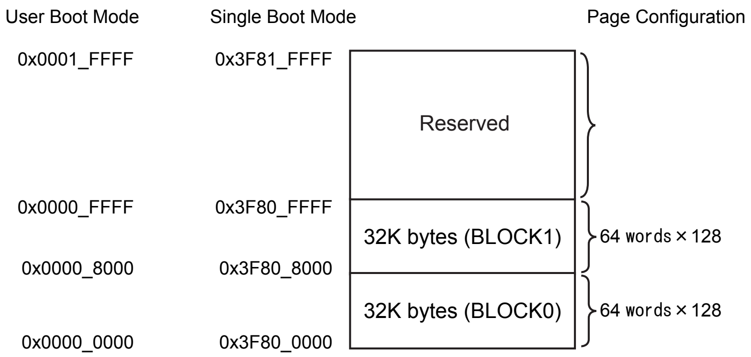


Figure 22-2 Block configuration (FLASH 64KB)

Flash memory configuration units ares described as "block" and "page".

- Page

  One page is 32 words. Same address [31:7] is used in a page. First address of the group is [6:0] = 0 and the last address of the group is [6:0] = 0x7F.

- Block

  One block is 32KB and flash memory is consists of four(128KB) or two(64KB) blocks.

Write operation is performed per page. The write time per page is 1.25ms. (Typ.)

Erase is performed per block (auto block erase command use) or performed on entire flash memory (use of auto chip erase command). Erase time varies on commands. If auto block command is used, the erase time will be 0.1 sec per block (Typ.). If the auto chip erase command is used to erase entire area, the time will be 0.2 sec (Typ.).

In addition, the protect function can be used per block. For detail of the protect function, refer to "22.1.5 Protect/Security Function".

22.1.2 Function

Flash memory built-in this device is generally compliant with the JEDEC standards except for some specific functions. Therefore, if a user is currently using a flash memory as an external memory, it is easy to implement the functions into this device. Furthermore, to provide easy write or erase operation, this product contains a dedicated circuit to perform write or chip erase automatically.

JEDEC compliant functions	Modified, added, or deleted functions
・Automatic programming ・Automatic chip erase ・Automatic block erase ・Data polling/toggle bit	<Modified> Block write/erase protect (only software protection is supported) <Deleted> Erase resume - suspend function

22.1.3 Operation Mode

22.1.3.1 Mode Description

This device provides the single chip mode and single boot mode. The single chip mode contains the normal mode and user boot mode. Figure 22-3 shows the mode transition.

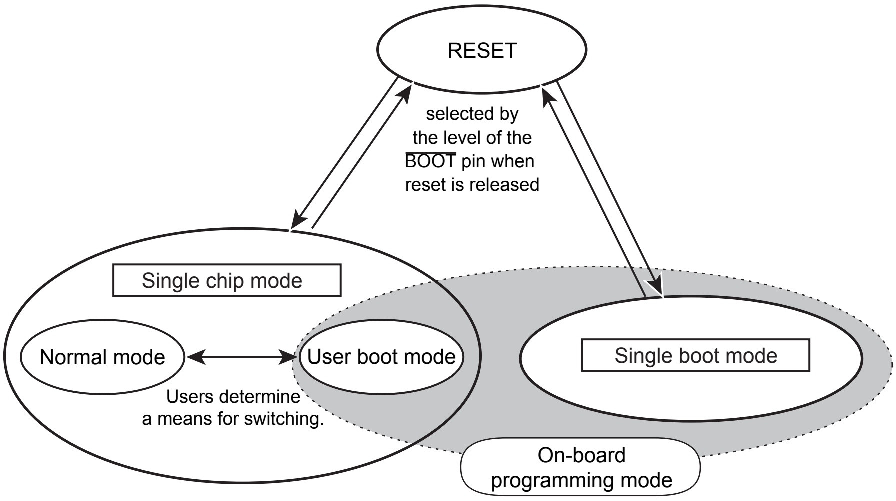


Figure 22-3 Mode transition

(1) Single chip mode

The single chip mode is a mode where the device can boot-up from Flash memory after reset. The mode contains two sub-modes in below.

- Normal mode

  The mode where user application program is executed.

- User boot mode

  The mode where flash memory is re-programmed on the user’s set.

Users can switch the normal mode to user boot mode freely. For example, a user can set if PA0 of port A is "1", the mode is the normal mode. If PA0 of port A is "0", the mode is the user boot mode. The user must prepare a routine program in the application program to determine the switching.

(2) Single boot mode

The mode where flash memory can boot-up from the built-in BOOT ROM (Mask ROM) after reset.

The BOOT ROM contains the algorithm that can rewrite Flash memory via serial port of this device on the user’s set. With connecting the serial port to external host, data transfer is performed in above-mentioned protocol and re-programmed Flash memory.

(3) On-board programming mode

The user boot mode and single boot mode are the modes where flash memory can be re-programmable on the user’s set. These two modes are called "on-board programming mode".

22.1.3.2 Mode Determination

Either the single chip or single boot operation mode can be selected by the level of the $\overline{\text{BOOT}}$ pin when reset is released.

Table 22-2 Operation mode setting

Operation mode	Pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$
Single chip mode	0 → 1	1
Single boot mode	0 → 1	0

22.1.4 Memory Map

Figure 22-4 Figure 22-5 shows a comparison of the memory map in the single chip mode and single boot mode. In the single boot mode, built-in Flash memory is mapped to 0x3F80_0000 and subsequent addresses, and the built-in BOOT ROM is mapped to 0x0000_0000 through 0x0000_0FFF.

Flash memory and RAM addresses are shown below.

FLASH size	RAM size	FLASH address	RAM address
128 KB	10 KB	0x0000_0000 to 0x0001_FFFF(single chip mode) 0x3F80_0000 to 0x3F81_FFFF(single boot mode)	0x2000_0000 to 0x2000_27FF
64KB	8KB	0x0000_0000 to 0x0000_FFFF(single chip mode) 0x3F80_0000 to 0x3F80_FFFF(single boot mode)	0x2000_0000 to 0x2000_1FFF

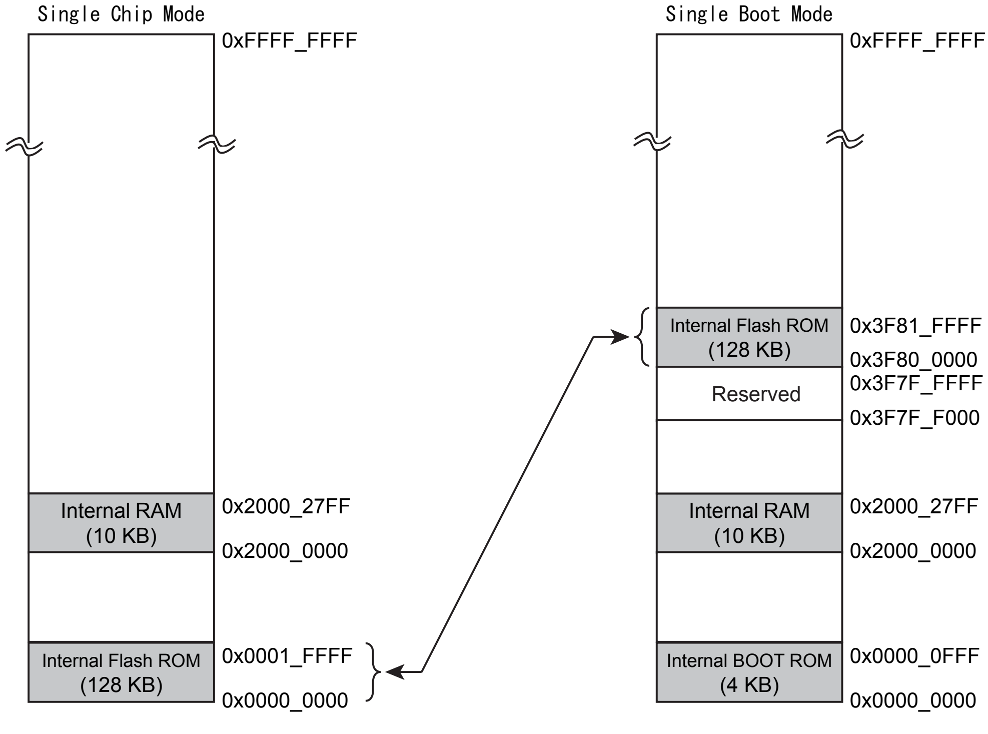


Figure 22-4 Comparison of memory map (FLASH 128KB)

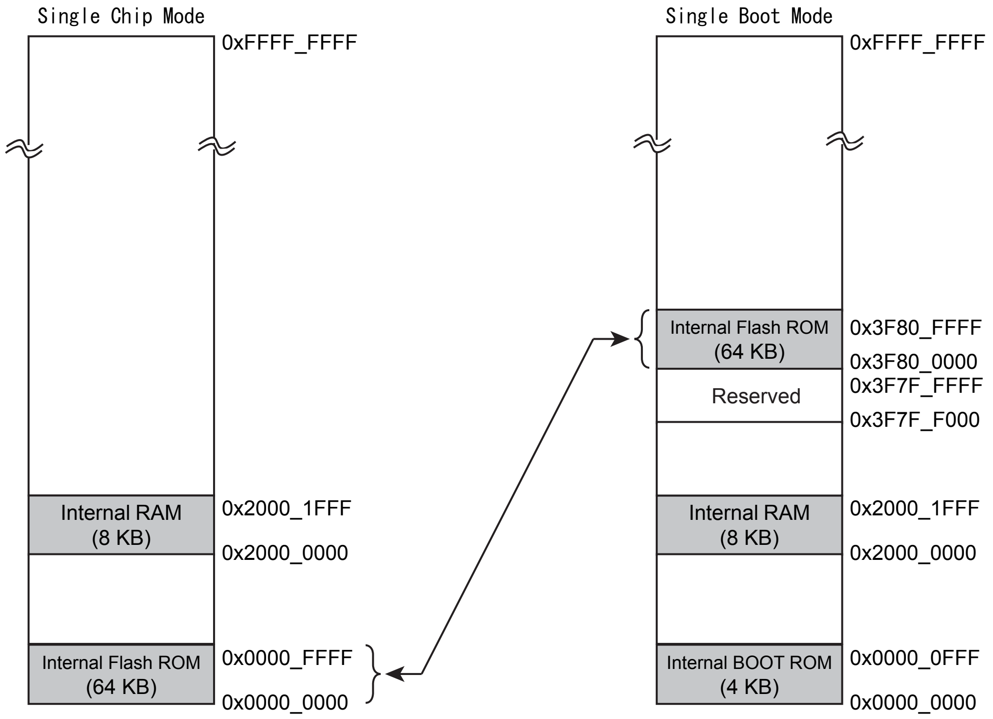


Figure 22-5 Comparison of memory map (FLASH 64KB)

22.1.5 Protect/Security Function

This device has the protect and security functions for Flash memory.

1. Protect function

   The write/erase operation can be inhibited per block.

2. Security function

   The read operation from a flash writer can be inhibited.

   Usage restrictions on debug functions

22.1.5.1 Protect Function

This function inhibits the write/erase operation per block.

To enable the protect function, a protect bit corresponding to a block is set to "1" using the protect bit program command. If a protect bit is set to "0" using the protect bit erase command, a block protect can be cancelled. The protect bit can be monitored with FCPSRA<BLK[3:0]>.

A program of protect bit can be programmed by 1-bit unit and can be erased by 4-bit unit. For detail of programming/erasing of protect bits, refer to "22.2.5 Command Description".

22.1.5.2 Protect Bit Mask Function

TMPM381/383 can temporarily release the protected function by masking the protect bits. To enable the protect function, set the corresponding bit of FCPMRA<BLKM[3:0]> to "0".

Note that when the system reset occurs, FCPMRA<BLKM[3:0]> is set to "1".

The contents of the protect bit are maintained in the non-maskable state.

FCPMRA<BLKM[3:0]> should be written as follows:

Note: Use a 32-bit transfer instruction when the following writing operations, item1 and 2.

1. Write the specified code (0xa74a9d23) to FCPMRA
2. Write data within 16 clocks after the operation of item 1.

Note: When the security function is enabled,even if All the protection bit is masked ,Automatic Chip Erase can not be used.

22.1.5.3 Security Function

Table 22-3 shows operations when the security function is enabled.

Table 22-3 Operations when the security function is enabled.

Item	Description
Read flash memory	CPU can read flash memory.
Debug port	JTAG, serial wire or trace communication is disabled.
Command execution to Flash memory	Command write to flash memory is not accepted. If a user tries to erase a protect bit, chip erase is executed and all protect bits are erased.

The security function is enabled under the following conditions;

1. FCSECBIT<SECBIT> is set to "1".
2. All protect bits (FCPSRA<BLK>) are set to "1".

FCSECBIT<SECBIT> is set to "1" by the power-on reset. Rewriting of FCSECBIT <SECBIT> is described in below.

Note: Use a 32-bit transfer instruction when the following writing operations, item1 and 2.

1. Write the specified code (0xa74a9d23) to FCSECBIT
2. Write data within 16 clocks after the operation of item 1.

22.1.6 Register

22.1.6.1 Register List

Base Address = 0x41FF_F000

Register name		Address(Base+)
Security bit register	FCSECBIT	0x0010
Flash Interface control register	FCCR	0x001C
Flash status register	FCSR	0x0020
Flash protect status register A	FCPSRA	0x0030
Flash protect mask register A	FCPMRA	0x0038

22.1.6.2 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 bit 0: Security function setting is disabled. 1: Security function setting is enabled.

Note: This register is initialized by power-on reset.

22.1.6.3 FCCR (Flash Interface 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	-	-	-	-	-	-	-	FLBOFF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	FLBOFF	R/W	Control of Flash Interface with instruction Buffer (Note 1) 0: Enable Instruction Buffer 1: Disable Instruction Buffer (with Buffer clear) This bit is a functional bit for controlling the Flash Interface . To use Instruction Buffer, set "0". To not use Instruction Buffer ,set to "1". In TMPM381/383, it must be set "0" for Flash accessing. b

Note 1: In TMPM381/383, after Flash programming or Flash Erasing ,it should be Clearing Instruction buffer by this functional bit or insert a reset signal .

Instruction Buffer Clearing operation as following.

after executing FCCR<FLBOFF>="1" , set FCCR<FLBOFF>="0" again on RAM.

22.1.6.4 FCSR (Flash 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	-	-	-	-	-	-	-	RDY_BSY
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	RDY_BSY	R	Ready/Busy (Note 1) 0: Busy (during auto operation) 1: Ready (auto operation ends) This bit is a function bit to monitor flash memory from CPU. While flash memory is in auto operation, this bit outputs "0" to indicate that flash memory is busy. Once auto operation is finished, this bit becomes ready state and outputs "1". Then next command is accepted. If a result of auto operation is failed, this bit outputs "0" continuously. The bit returns to "1" by hardware reset.

Note 1: Make sure that flash memory is ready before commands are issued. If a command is issued during busy, not only the command is not sent but also subsequent commands may not be accepted. In that case, use hardware reset to return. Hardware reset needs 0.5 μs or more reset period regardless of system clock. At this time, it takes approximately 2 ms until enabling to read after reset.

22.1.6.5 FCPSRA (Flash protect 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	-	-	-	-	BLK3	BLK2	BLK1	BLK0
After reset	0	0	0	0	(Note 1)	(Note 1)	(Note 1)	(Note 1)

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3-0	BLK3- BLK0	R	Protection status of Block3 to 0 0: Not protected 1: Protected Protect bit values correspond to protect status of each block. If corresponding bit indicates "1", corresponding block is in the protection status. A block in the protection status cannot be re-programmable.

Note 1: A value will correspond to the protection status.

Note 2: BLK3, BLK2 can not be used in FLASH 64KB version.

22.1.6.6 FCPMRA (Flash protect 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	-	-	-	-	BLKM3	BLKM2	BLKM1	BLKM0
After reset	0	0	0	0	1	1	1	1

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3-0	BLKM3- BLKM0	R	Masks protect bits of block3 through block 0. 0: Release the protect function (Protect bit is mask.) 1: - (Protect bit is not masked.) When the corresponding bit is "0", this block is released. When the corresponding bit is "1", the corresponding bit is set to the protect state(each bit of FCPSRA) and this block in the Flash memory becomes protect state.

Note 1: This register is initialized by system reset.

Note 2: Do not modify FCPMRA<BLKM[3:0]> while data is being written/erased to/from the Flash memory.

Note 3: When FCPMRA <BLKM[3:0]> is modified, read the register again to check whether the Flash is re-written. Then access the Flash memory.

Note 4: BLKM3, BLKM2 can not be used in FLASH 64KB version.

22.2 Detail of Flash Memory

In on-board programming, the CPU executes commands for reprogramming or erasing Flash memory. This reprogramming/erase control program should be prepared by the user beforehand. Since Flash memory content cannot be read while Flash memory is being written or erased, it is necessary to run the reprogram/erase control program on the built-in RAM. Do not generate interrupt/fault except reset to avoid abnormal program termination.

22.2.1 Function

Flash memory is generally compliant with the JEDEC standards except for some specific functions. However; a method of address designation of operation command is different from standard commands.

If write/erase operation is executed, commands are input to flash memory using 32-bit (1-word) store instruction command. After command input, write or erase operation is automatically executed in inside.

Table 22-4 Flash memory function

Main function	Description
Automatic page program	Writes data automatically.
Automatic chip erase	Erases the entire area of Flash memory automatically.
Automatic block erase	Erases a selected block automatically.
Write/erase protect	The write or erase operation can be individually inhibited for each block.

Note: In TMPM381/383, after Flash programming or Flash Erasing ,it should be Clearing Instruction buffer. Pls refer "22.1.6.3 FCCR (Flash Interface control register)" for a clear method.

Note: Check the FCSR<RDY_BSY> to make sure each command sequence end such as Flash writing, Flash Erase, Protection bit program, Protection bit Erase. and then hold for 200 μs or more before reading data from Flash memory or starting instruction fetch.

22.2.2 Operation Mode of Flash Memory

Flash memory provides mainly two types of operation modes;

- The mode to read memory data (Read mode)
- The mode to erase or rewrite memory data automatically (Automatic operation mode)

After power-on, after rest or after automatic operation mode is finished normally, Flash memory becomes read mode. Instruction stored in Flash memory or data read is executed in the read mode.

If commands is input during the read mode, the operation mode becomes the automatic operation. If the command process is normally finished, the operation mode returns to the read mode except the ID-Read command. During the automatic operation, data read and instruction execution stored in Flash memory cannot be performed.

If command process is abnormally finished then the operation mode should forcibly return to read mode. In this case, use the read command, read/reset command or hardware reset.

22.2.3 Hardware Reset

A hardware reset means a power-on reset or warm reset to use returning to the read mode when the automatic programming/erase operation is forcibly cancelled, or automatic operation abnormally ends.

If the hardware reset occurs during the automatic operation, Flash memory stops the automatic operation and returns to the read mode. If a hardware reset is generated during Flash memory automatic program/erase operation, the hardware reset needs 0.5 μs or more reset period regardless of system clock. At this time, it takes approximately 2 ms until enabling to read after reset. Note that if a hardware reset occurs during the automatic operation, data write operation is not executed properly. Set write operation again.

For detail of the reset operation, refer to "Reset". After a given reset input, CPU will read the reset vector data and then starts the routine after reset.

22.2.4 How to Execute Command

The command execution is performed by writing command sequences to Flash memory with a store instruction. Flash memory executes each automatic operation command according to the combination of input addresses and data. For detail of the command execution, refer to "22.2.5 Command Description".

An execution of store instruction to the Flash memory is called "bus write cycle". Each command consists of some bus write cycles. In Flash memory, when address and data of bus write cycle are performed in the specified order, the automatic command operation is performed. When the cycle is performed in non-specified order, Flash memory stops command execution and returns to the read mode.

If you cancel the command during the command sequence or input a different command sequence, execute the read command or read/reset command. Then Flash memory stops command execution and returns to the read mode. The read command and read/reset command are called "software reset".

When write command sequence ends, the automatic operation starts and FCSR<RDY_BSY> is set to "0". When the automatic operation normally ends, FCSR<RDY_BSY> = "1" is set and Flash memory returns to the read mode.

New command sequences are not accepted during the automatic operation. If you want to stop the command operation, use a hardware reset. In case that the automatic operation abnormally ends (FCSR<RDY_BSY> remains "0"), Flash memory remains locked and will not return to the read mode. To returns to the read mode, use a hardware reset. If the hardware reset stops the command operation, commands are not normally executed.

Notes on the command execution;

1. To recognize command, command sequencer need to be in the read mode before command starting. Confirm FCSR<RDY_BSY> = 1 is set prior to the first bus write cycle of each command. Consecutively, it is recommended that the read command is executed.
2. Execute each command sequence from outside of Flash memory.
3. Execute sequentially each bus write cycle by data transfer instruction in one-word (32-bit).
4. Do not access Flash memory during the each command sequence. Do not generate any interrupt or fault except reset.
5. Upon issuing a command, if any address or data is incorrectly written, make sure to return to the read mode by using software reset.

22.2.5 Command Description

This section explains each command content. For detail of specific command sequences, refer to "22.2.6 Command Sequence".

22.2.5.1 Automatic Page Program

(1) Operation Description

The automatic page program writes data per page. When the program writes data to multiple pages, a page command need to be executed in page by page. Writing across pages is not possible.

Writing to Flash memory means that data cell of "1" becomes data of "0". It is not possible to become data cell of "1" from data of "0". To become data cell of "1" from "0", the erase operation is required.

The automatic page program is allowed only once to each page already erased. Either data cell of "1" or "0" cannot be written data twice or more. If rewriting to a page that has already been written once, the automatic page program is needed to be set again after the automatic block erase or automatic chip erase command is executed.

Note 1: Page program execution to the same page twice or more without erasing operation may damage the device.

Note 2: Writing to the protected block is not possible.

(2) How to Set

The 1st to 3rd bus write cycles indicate the automatic page program command.

In the 4th bus write cycle, the first address and data of the page are written. On and after 5th bus cycle, one page data will be written sequentially. Data is written in one-word unit (32-bit).

If a part of the page is written, set "0xFFFFFFFF" as data, which means not required to write, for entire one page.

No automatic verify operation is performed internally in the device. So, be sure to read the data programmed to confirm that it has been correctly written.

If the automatic page program is abnormally terminated, that page has been failed to write. It is recommended not to use the device or not to use the block including the failed address.

22.2.5.2 Automatic Chip Erase

(1) Operation Description

The automatic chip erase is executed to the memory cell of all addresses. If protected blocks are contained, these blocks will not be erased. If all blocks are protected, the automatic chip erase operation will not performed and will return to the read mode after a command sequence is input.

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic chip erase command. After the command sequence is input, the automatic chip erase operation starts.

No automatic verify operation is performed internally in the device. So, be sure to read the data to confirm that it has been correctly erased.

22.2.5.3 Automatic Block Erase

(1) Operation Description

The automatic erase command performs erase operation to the specified block. If the specified block is protected, erase operation is not executed.

(2) How to Set

The 1st to 5th bus write cycles indicate the automatic block erase command. In the 6th bus write cycle, the block to be erased is specified. After the command sequence is input, the automatic block erase operation starts.

No automatic verify operation is performed internally in the device. So, be sure to read the data to confirm that it has been correctly erased.

22.2.5.4 Automatic Protect Bit Program

(1) Operation Description

The automatic protect bit program writes "1" to a protect bit at a time. To set "0" to a protect bit, use the automatic protect bit erase command.

For detail of the protect function, refer to "22.1.5 Protect/Security Function".

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic protect bit program command. In the 7th bus write cycle, the protect bit to be written is specified. After the command sequence is input, the automatic protect bit program starts. Check whether write operation is normally terminated with FCPSRA<BLK>.

22.2.5.5 Auto Protect Bit Erase

(1) Operation Description

The automatic protect bit erase command operation depends on the security status. For detail of security status, refer to "22.1.5 Protect/Security Function".

- Non-security status

  Clear the specified protect bit to "0". Protect bit erase is performed in 4-bit unit.
- Security status

  Erase all protect bits after all addresses of Flash memory are erased.

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic protect bit erase command. In the 7th bus write cycle, the protect bit to be erased is specified. After the command sequence is input, the automatic protect bit erase operation starts.

In the non-security status, specified protect bit is erased. Check whether erase operation is normally terminated with FCPSRA<BLK>.

In the security status, all addresses and all protect of Flash memory bits are erased. Confirm if data and protect bits are erased normally. If necessary, execute the automatic protect bit erase, automatic chip erase or automatic block erase.

All cases are the same as other commands, FCSR<RDY_BSY> becomes "0" during the automatic protect bit erase command operation. After the operation is complete, FCSR<RDY_BSY> becomes "1" and Flash memory will return to the read mode. To abort the operation, a hardware reset is required.

22.2.5.6 ID-Read

(1) Operation Description

The ID-Read command can read information including Flash memory type and three types of codes such as a maker code, device code and macro code.

(2) How to Set

The 1st to 3rd bus write cycles indicate the ID-Read command. In the 4th bus write cycle, the code to be read is specified. After the 4th bus write cycle, read operation in the arbitrary flash area acquires codes.

The ID-Read can be executed successively. The 4th bus write cycle and reading ID value can be executed repeatedly.

The ID-Read command does not automatically return to the read mode. To return to the read mode, execute the read command, read/reset command or hardware reset.

22.2.5.7 Read Command and Read/reset Command (Software Reset)

(1) Operation Description

A command to return Flash memory to the read mode.

When the ID-Read command is executed, macro stops at the current status without automatically return to the read mode. To return to the read mode from this situation, use the read command or read/reset command. It is also used to cancel the command when commands are input to the middle.

(2) How to Set

The 1st bus cycle indicates the read command. The 1st to 3rd bus write cycles indicate the read/reset command. After either command sequence is executed, Flash memory returns to the read mode.

22.2.6 Command Sequence

22.2.6.1 Command Sequence List

Table 22-5 shows addresses and data of bus write cycle in each command.

All command cycles except the 5th bus cycle of ID-Read command are bus write cycles. A bus write cycle is performed by 32-bit (1-word) data transfer instruction. (Following table shows only lower 8 bits of data.)

For detail of addresses, refer to Table 22-6. Use below values to "command" described in a column of Addr[15:9] in the Table 22-6.

Note 1) Always set to "0" to the address bit [1:0].

Note 2) Set below values to the address bit [19] according to Flash memory size.
Memory size is 1MB or less : Always set to "0"
Memory size is over 1MB : If bus write to 1MB area or less, the bit is set to "0".
If bus write to over 1MB area, the bit is set to "1".

Table 22-5 Command Sequence

Command	1st bus cycle	2nd bus cycle	3rd bus cycle	4th bus cycle	5th bus cycle	6th bus cycle	7th bus cycle
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	Data	Data	Data	Data	Data	Data	Data
Read	0xXX	−	−	−	−	−	−
	0xF0	−	−	−	−	−	−
Read/reset	0xX55X	0xXAAX	0xX55X	−	−	−	−
	0xAA	0x55	0xF0	−	−	−	−
ID-Read	0xX55X	0xXAAX	0xX55X	IA	0xXX	−	−
	0xAA	0x55	0x90	0x00	ID	−	−
Automatic page program	0xX55X	0xXAAX	0xX55X	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic chip erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	−
	0xAA	0x55	0x80	0xAA	0x55	0x10	−
Automatic block erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	BA	−
	0xAA	0x55	0x80	0xAA	0x55	0x30	−
Automatic protect bit program	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
Automatic protect bit erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	0xXX
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

Supplementary explanation

- IA: ID Address
- ID: ID data
- PA: Program page address
- PD: Program data (32-bit data)

  After the 4th bus cycle, input data in the order of the addresses per page
- BA: Block address (see Table 22-7)

- PBA: Protect bit address (see Table 22-8)

22.2.6.2 Address Bit Configuration in the Bus Cycle

Table 22-6 is used in conjunction with "Table 22-5 Command Sequence".

Set the address setting according to the normal bus write cycle address configuration from the first bus cycle.

Table 22-6 Address bit configuration in the bus write cycle

Address	Addr [31:15]	Addr [14]	Addr [13:12]	Addr [11:9]	Addr [8:7]	Addr [6:4]	Addr [3:0]
Normal Command	Normal bus write cycle address configuration						
	Flash area	"0" is recommended.		Command	Addr[1:0] = "0" (fixed) Other bits = "0" (recommended)		
ID-READ	IA: ID address (Setting of the 4th bus write cycle address for ID-READ)						
	Flash area	"0" is recom-mended.	ID Address	Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)			
Block erase	BA: Block address(Setting of the 6th bus write cycle address for block erase)						
	Block address (Table 22-7)	Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)					
Automatic page program	PA: Program page address (Setting of the 4th bus write cycle address for page program)						
	Page address					Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)	
Protect bit program	PBA: Protect bit address (Setting of the 7th bus write cycle address for protect bit program)						
	Flash area	Fix to "0"			Protect bit selection (Table 22-8)	Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)	

22.2.6.3 Block Address (BA)

Table 22-7 shows block addresses. Specify any address included in the block to be erased in the 6th bus write cycle of the automatic block erase command.

Table 22-7 Block address

Block	Address (User boot mode)	Address (Single boot mode)	Size (Kbyte)
3	0x0001_8000 to 0x0001_FFFF	0x3F81_8000 to 0x3F81_FFFF	32
2	0x0001_0000 to 0x0001_7FFF	0x3F81_0000 to 0x3F81_7FFF	32
1	0x0000_8000 to 0x0000_FFFF	0x3F80_8000 to 0x3F80_FFFF	32
0	0x0000_0000 to 0x0000_7FFF	0x3F80_0000 to 0x3F80_7FFF	32

22.2.6.4 How to Specify Protect Bit (PBA)

The protect bit is specified in 1-bit unit in programming and in 4-bit unit in erasing.

Table 22-8 shows a protect bit selection table of the automatic protect bit program. The column of address example indicates an address described in upper side is used in the use boot mode and the lower side is used in the single boot mode.

Four protect bits are erased by the automatic protect bit erase command in all.

Table 22-8 Protect bit program address

Block	Protect bit	Address of 7th bus write cycle			Address example [31:0]
		Address [14:9]	Address [8]	Address [7]	
Block0	<BLK[0]>	Fix to "0"	0	0	0x0000_0000 0x3F80_0000
Block1	<BLK[1]>		0	1	0x0000_0080 0x3F80_0080
Block2	<BLK[2]>	Fix to "0"	1	0	0x0000_0100 0x3F80_0100
Block3	<BLK[3]>		1	1	0x0000_0180 0x3F80_0180

22.2.6.5 ID-Read Code (IA, ID)

Table 22-9 shows how to specify a code and the content using ID-Read command.

The column of address example indicates an address described in the upper side is used in the use boot mode and the lower side is used in the single boot mode

Table 22-9 ID-Read Command codes and contents

Code	ID[7:0]	IA[13:12]	Address Example [31:0]
Manufacture code	0x98	0b00	0x0000_0000 0x3F80_0000
Device code	0x5A	0b01	0x0000_1000 0x3F80_1000
−	Reserved	0b10	−
Macro code	0x33	0b11	0x0000_3000 0x3F80_3000

22.2.6.6 Example of Command Sequence

(1) use boot mode

Command		Bus cycle						
		1	2	3	4	5	6	7
Read	Address	0x0000_0000	–	–	–	–	–	–
	Data	0x0000_00F0	–	–	–	–	–	–
Read/reset	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	–	–	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_00F0	–	–	–	–
ID-Read	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	IA	0x0000_0000	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	–	–
Automatic page program	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	PA	In the following cycles, write addresses and data successively per page.		
	Data	0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
Automatic chip erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	–
Automatic block erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	BA	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	–
Automatic protect bit program	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	PBA
	Data	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A
Automatic protect bit erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550
	Data	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A

(2) Data single boot mode

Command		Bus cycle						
		1	2	3	4	5	6	7
Read	Address	0x3F80_0000	–	–	–	–	–	–
	Data	0x0000_00F0	–	–	–	–	–	–
Read/reset	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	–	–	–	–
	Data	0x0000_00AA	0x3F80_0055	0x3F80_00F0	–	–	–	–
ID-Read	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	IA	0x0000_0000	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	–	–
Automatic page program	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	PA	In the following cycles, write addresses and data successively per page.		
	Data	0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
Automatic chip erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	–
Automatic block erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	BA	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	–
Automatic protect bit program	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	PBA
	Data	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A
Automatic protect bit erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550
	Data	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A

22.2.7 Flowchart

22.2.7.1 Automatic Program

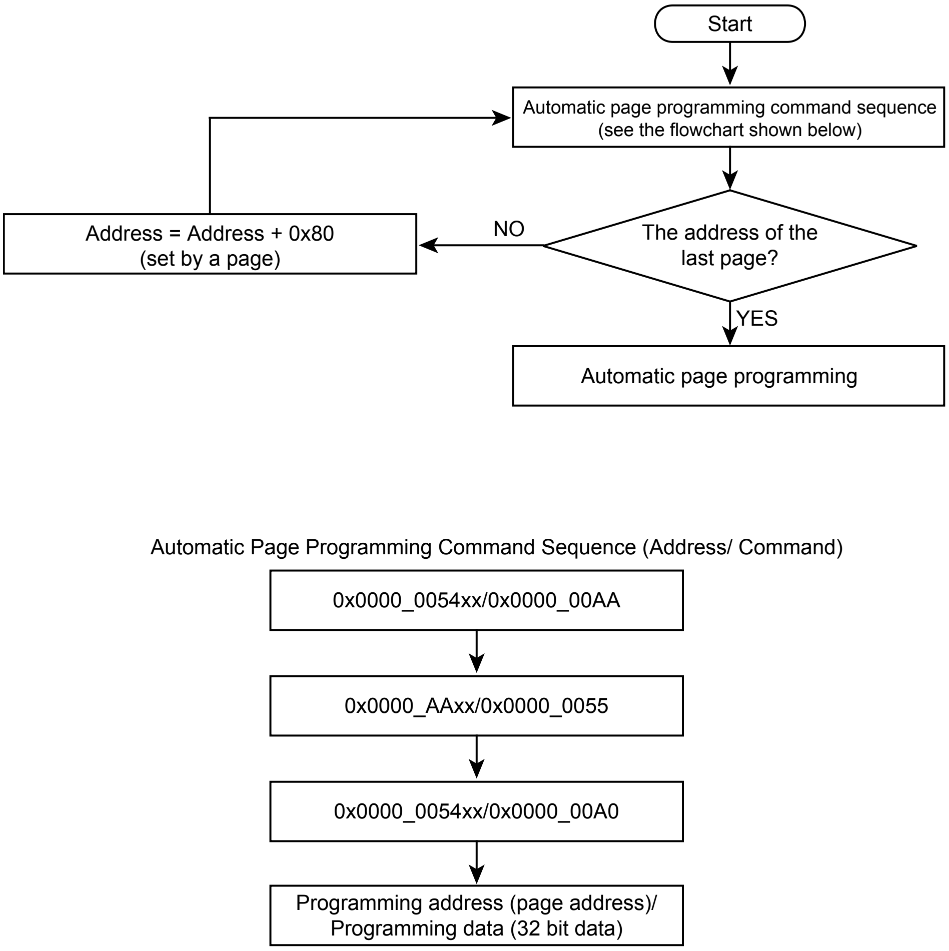

Start
Automatic page programming command sequence
(see the flowchart shown below)
The address of the last page?
NO
Address = Address + 0x80
(set by a page)
YES
Automatic page programming
Automatic Page Programming Command Sequence (Address/ Command)
0x0000_0054xx/0x0000_00AA
0x0000_AAxx/0x0000_0055
0x0000_0054xx/0x0000_00A0
Programming address (page address)/
Programming data (32 bit data)

Figure 22-6 Flowchart of automatic program

22.2.7.2 Automatic Erase

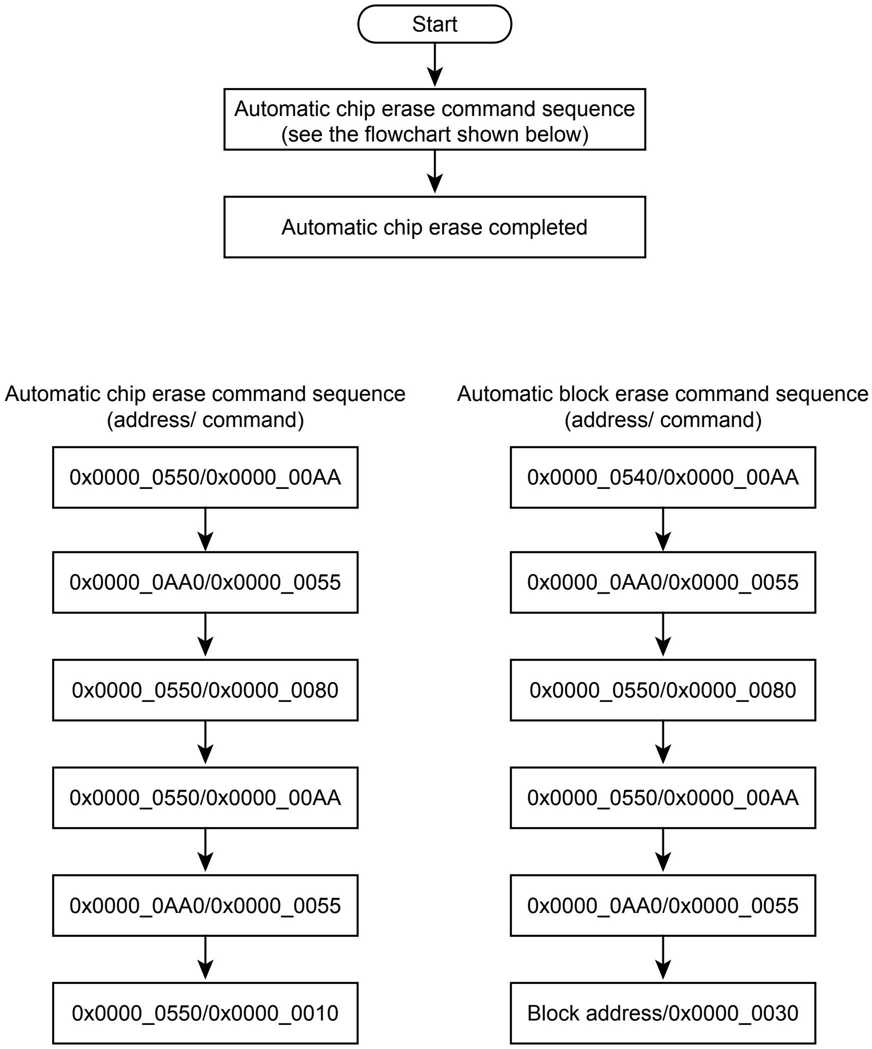


Figure 22-7 Flowchart of automatic erase

22.3 How to Reprogram Flash using Single Boot Mode

The single boot mode utilizes a program contained in built-in BOOT ROM for reprogramming Flash memory. In this mode, BOOT ROM is mapped to the area containing interrupt vector tables and Flash memory is mapped to another address area other than BOOT ROM area.

In the boot mode, Flash memory is reprogrammed using serial command/data transfer. With connecting serial channel (UART) of this device to the external host, a reprogramming program is copied from the external host to the built-in RAM. A reprogramming routine in the RAM is executed to reprogram Flash memory. For details of communication with host, follow the protocol described later.

Even in the single boot mode, do not generate interrupt/fault except reset to avoid abnormal program termination.

To secure the contents of Flash memory in the single chip mode (normal operation mode), once re-programming is complete, it is recommended to protect relevant flash blocks against accidental erasure during subsequent single chip operations.

22.3.1 Mode Setting

In order to execute the on-board programming, this device is booted-up in the single boot mode. Below setting is for the single boot mode setting.

$\overline{\text{BOOT}}$ = 0
$\overline{\text{RESET}}$ = 0 → 1

While $\overline{\text{BOOT}}$ pin is set to the above in advance, set $\overline{\text{RESET}}$ pin to "0". Then release $\overline{\text{RESET}}$ pin, the device will boot-up in the single boot mode.

22.3.2 Interface Specification

This section describes UART communication format in the single boot mode. The serial operation supports UART (asynchronous communication) modes. In order to execute the on-board programming, set the communication format of the programming controller as well.

- UART communication

  Communication channel: channel 0

  Serial transfer mode: UART (asynchronous), half-duplex, LSB first

  Data length: 8-bit

  Parity bit: None

  STOP bit: 1-bit

  Baud rate: Arbitrary baud rate

The boot program operates the clock/mode control block setting as an initial condition. For detail of the initial setting of the clock, refer to "Clock/Mode control".

As explained in the "22.3.5.1 Serial Operation Mode Determination", a baud rate is determined by the 16-bit timer (TMRB). When determining the baud rate, communication is executed by 1/16 of a desired baud rate. Therefore, the communication baud rate must be within the measurable range. The timer count clock operates at ΦT1 (fc/2).

Table 22-10 shows the pins used in the boot program. Other than these pins are not used by the boot program.

Table 22-10 Pin connection

Pin		Interface
		UART
Mode setting pin	$\overline{\text{BOOT}}$	o
Reset pin	$\overline{\text{RESET}}$	o
Communication pin	TXD0 (PE0)	o
	RXD0 (PE1)	o

o:used ×:unused

22.3.3 Restrictions on Internal Memories

Note that the single boot mode places restrictions on the built-in RAM and built-in flash memory as shown in Table 22-11.

Table 22-11 Restrictions on the memories in the single boot mode

Memory	Restrictions
Internal RAM	Boot program uses the memory as a work area through 0x2000_0000 to 0x2000_03FF. Store the program 0x2000_0400 through the end address of RAM. The start address of the program must be even address.
Internal flash memory	The following addresses are assigned for storing software ID information and passwords. Storing program in below addresses is not recommendable. 0x3F81_FFF0 to 0x3F81_FFFF

Note: If a password is erased data (0xFF), it is difficult to protect data secure due to an easy-to-guess password. Even if the single boot mode is not used, it is recommended to set a unique value as a password.

22.3.4 Operation Command

The boot program provides the following operation commands.

Table 22-12 Operation command data

Operation command data	Operation mode
0x10	RAM transfer
0x40	Flash memory chip erase and protect bit erase

22.3.4.1 RAM Transfer

The RAM transfer is to store data from the controller to the built-in RAM. When the transfer is complete normally, a user program starts. User program can use the memory address of 0x2000_0400 or later except 0x2000_0000 to 0x2000_03FF for the boot program. CPU will start execution from RAM store start address. The start address must be even address.

This RAM transfer function enables user-specific on-board programming control. In order to execute the on-board programming by a user program, use Flash memory command sequence explained in 22.2.6.

22.3.4.2 Flash Memory Chip Erase and Protect Bit Erase

Flash memory chip erase and protect bit erase commands erase the entire blocks of Flash memory and write/erase protects of all blocks regardless of write/erase protect or security status.

22.3.5 Common Operation regardless of Command

This section describes common operation under the boot program execution.

22.3.5.1 Serial Operation Mode Determination

When the controller communicates via UART, set the 1st byte to 0x86 at the desired baud rate. Figure 22-8shows waveforms .

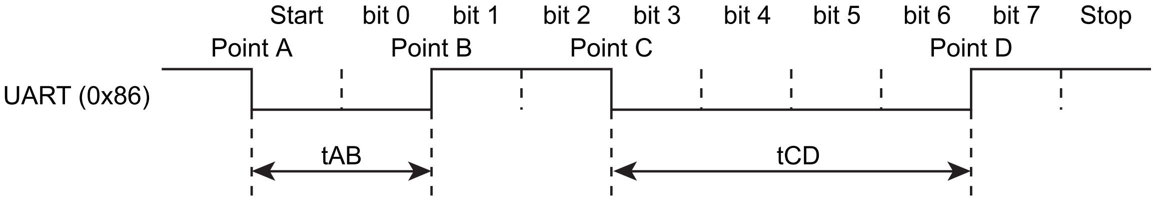


Figure 22-8 Serial operation mode determination data

Figure 22-9 shows a flowchart of boot program. Using 16-bit timer (TMRB) with the time of tAB, tAC and tAD, the 1st byte of serial operation mode determination data (0x86) after reset is provided. In Figure 22-9, the CPU monitors level of the receive pin, and obtains a timer value at the moment when the receive pin's level is changed. Consequently, the timer values of tAB, tAC and tAD have a margin of error. In addition, note that if the transfer goes at a high baud rate, the CPU may not be able to determine the level of receive pin.

The flowchart in Figure 22-10 shows the serial operation mode is determined that the time length of the receive pin is long or short. If the length is tAB ≤ tCD, the serial operation mode is determined as UART mode. The time of tAD is used whether the automatic baud rate setting is enable or not. Note that timer values of tAB, tAC and tAD have a margin of error. If the baud rate is high and operation frequency is low, each timer value becomes small. This may generates unexpected determination occurs. (To prevent this problem, re-set UART within the programming routine.)

For example, When UART mode is utilized, the controller should allow for a time-out period where the time is expected 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.

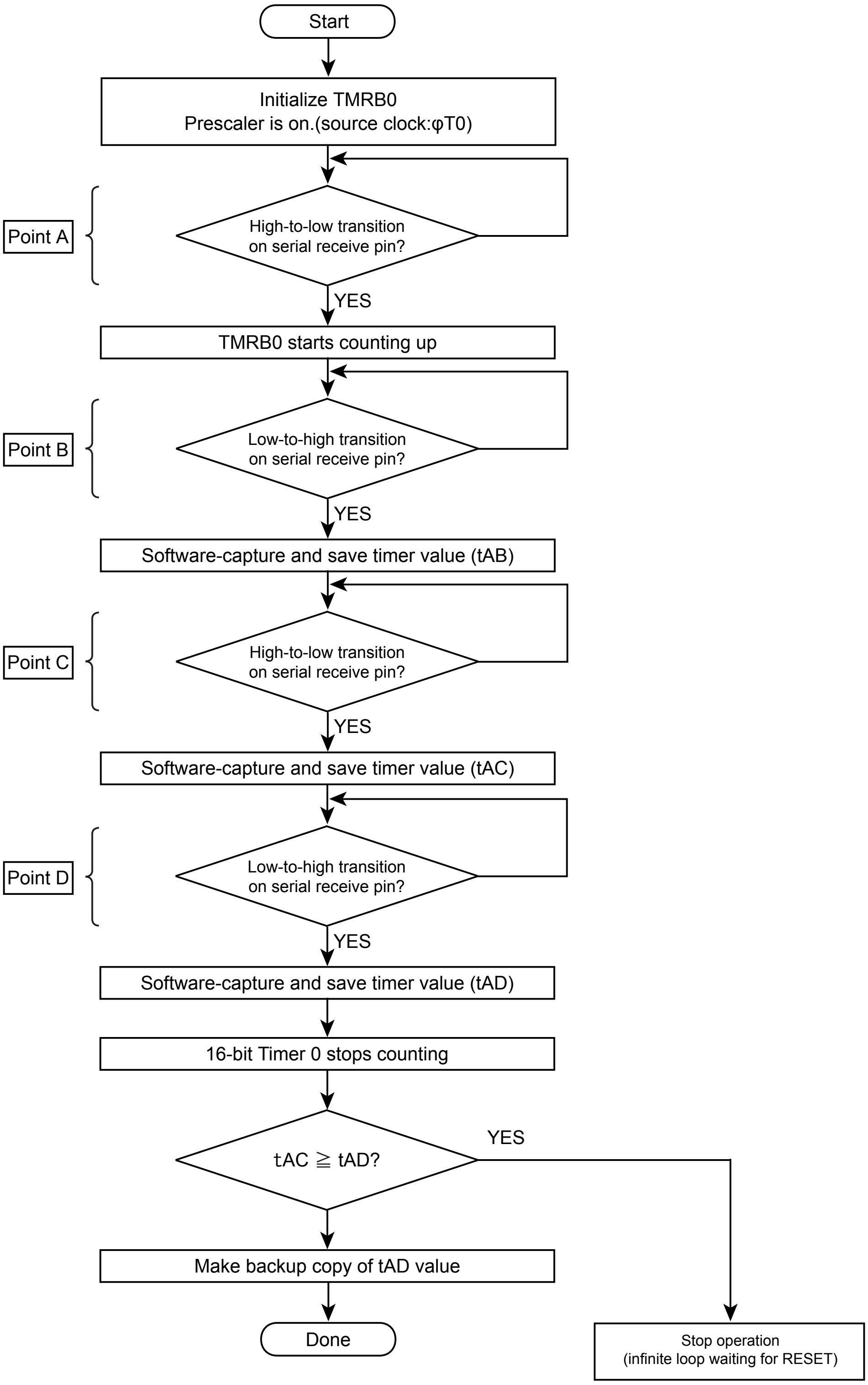


Figure 22-9 Serial operation mode receive flowchart

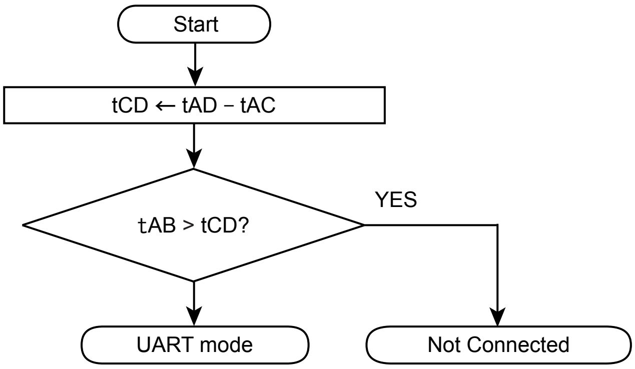


Figure 22-10 Serial operation mode determination flowchart

22.3.5.2 Acknowledge Response Data

The boot program represents processing states in specific codes and sends them to the controller. Table 22-13 to Table 22-16 show the values of acknowledge responses to each receive data.

In Table 22-14 to Table 22-16, the upper four bits of the acknowledge response are equal to those of the operation command data. The 3rd bit indicates a receive error. The 0th bit indicates an invalid operation command error, a checksum error or a password error. The 1st bit and 2nd bit are always "0".

Table 22-13 ACK response to the serial operation determination data

Transmit data	Description
0x86	Determined that UART communication is possible. (Note)

Note: When the serial operation is determined as UART, if the baud rate setting is determined as unacceptable, the boot program aborts without sending back any response.

Table 22-14 ACK response to the operation command data

Transmit data	Description
0x?8 (Note)	A receive error occurs in the operation command data
0x?1 (Note)	An undefined operation command data is received normally.
0x10	Determined as a RAM transfer command
0x40	Determined as a flash memory chip erase command

Note: The upper 4 bits of the ACK response data are the same as those of the previous command data.

Table 22-15 ACK response to the CHECK SUM data

Transmit data	Description
0xN8 (Note)	A receive error occurs.
0xN1 (Note)	A CHECK SUM or a password error occurs.
0xN0 (Note)	The CHECK SUM value is correct.

Note: The upper 4 bits of the ACK response data are the same as those of the operation command data.

Table 22-16 ACK response to Flash memory chip erase and protect bit erase operation

Transmit data	Description
0x54	Determined as a erase enable command
0x4F	Erase command is complete.
0x4C	Erase command is abnormally terminated.

Note: Even when an erase command is performed normally, a Negative acknowledge may be returned by ACK response. Check the FCSR<RDY_BSY> to make sure the command sequence end, and then hold for 200 μ s or more, after that reconfirm the erase status.

22.3.5.3 Password Determination

The boot program use the below area to determine whether a password is required or use as a password.

Area	Address
Password requirement determination	0x3F81_FFF0 (1byte)
Password area	0x3F81_FFF4 to 0x3F81_FFFF (12byte)

The RAM Transfer command performs a password verification regardless of necessity judging data. Flash memory chip erase or protect bit erase command performs a password verification only when necessity judging is determined as "required".

Password requirement setting	Data
Need password	Other than 0xFF
No password	0xFF

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.

(1) Password verification using RAM transfer command

If all these address locations contain the same bytes of data other than 0xFF, this condition is determined as a password area error as shown in Figure 22-11. In this case, the boot program returns an error acknowledge (0x11) in response to the 17th byte of checksum value regardless of the password verification.

The boot program verifies 5th byte to 16th byte of receive data (password data). A password error occurs if all 12 bytes do not match. If the password error is determined, an ACK response data to the 17th of CHECK SUM data is a password error.

The password verification is performed even if the security function is enabled.

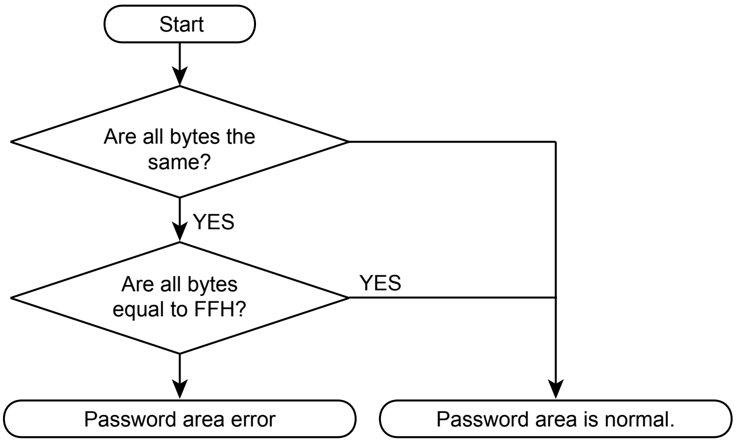


Figure 22-11 Password area check flowchart

(2) Password verification to Flash memory chip erase and protect bit erase command

When a password is enable in the erase password necessity determination area as shown in Figure 22-12 and the passwords are identical data, a password area error occurs. If a password area error is determined, an ACK response to the 17th byte of CHECK SUM sends 0x41 regardless of the password verification.

The boot program verifies 5th byte to 16th byte of receive data (password data). A password error occurs if all 12 bytes do not match. If the password error is determined, an ACK response data to the 17th of CHECK SUM data is a password error.

The password verification is performed even if the security function is enabled.

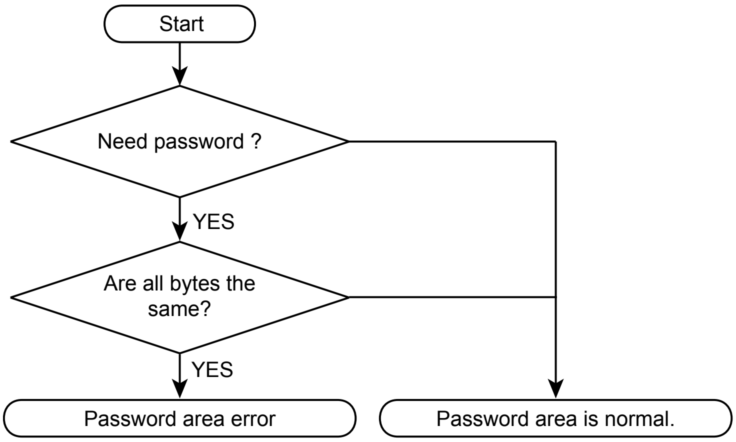


Figure 22-12 Password area check flowchart

22.3.5.4 CHECK SUM Calculation

The checksum is calculated by 8-bit addition to transmit data, dropping the carries, and taking the two's complement of the total sum. The controller must perform the same checksum operation in transmitting checksum bytes.

Example of CHECK SUM

To calculate the checksum for a series of 0xE5 and 0xF6, perform 8-bit addition.

0xE5 + 0xF6 = 0x1DB

Take the two's complement of the sum to the lower 8-bit, and that is a checksum value. So the boot program sends 0x25 to the controller.

0 − 0xDB = 0x25

22.3.6 Transfer Format at RAM Transfer

This section shows a RAM transfer command format. Transfer directions in the table are indicated as follows:

Transfer direction (C→T): Controller to TMPM381/383

Transfer direction (C←T): TMPM381/383 to Controller

Number of transfer bytes	Transfer direction	Transfer data	Description
1	C→T	Serial operation mode and baud rate setting	Sends data to determine the serial operation mode. For detail of mode determination, refer to "22.3.5.1 Serial Operation Mode Determination".
		[UART mode] 0x86	Sends 0x86. If UART mode is determined, the program determines whether a baud setting is possible. If not, the program stops and communication is shutdown.
2	C←T	ACK response to serial operation mode	The 2nd byte of transmit data is a ACK response data to the 1st byte that corresponds to the serial operation setting mode data. If the setting is possible, sets SIO/UART. A receive enable timing is set before transmit buffer is written to the data.
		[UART mode] Normal state: 0x86	If the setting is determined to be possible, sends 0x86. If not, the operation aborts without sending back any response. When the controller finished to send the 1st byte of data, requires a time-out time (5 seconds). If data (0x86) is not normally received within a time-out time, communication is not possible.
3	C→T	Operation command data (0x10)	Sends RAM transfer command data (0x10).
4	C←T	ACK response to operation command Normal state: 0x10 Abnormal state: 0xX1 Communication error: 0xX8	ACK response data to the operation command. First, checks if 3rd byte of receive data has errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command.) Then, if the 3rd byte of receive data corresponds to either operation command data in Table 22-12, receive data is echoed back. In the case of RAM transfer, 0x10 is echoed back and the transfer data branches to the RAM transfer service routine. If the data does not correspond to the command in Table 22-12, sends a ACK response data 0xX1 that means operation command errors, and waits for next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command data.)
5 to 16	C→T	Password data (12-byte) 0x3F81_FFF4 to 0x3F81_FFFF	Checks data in the password area. For detail of password area checking, refer to "22.3.5.3 Password Determination". Compares 5th to 16th byte of receive data with 0x3F81_FFF0 to 0x3F81_FFFF of data of Flash memory. If the data does not match the address, a password error flag is set.
17	C→T	5th to 16th byte of CHECK SUM values	Send 5th to 16th byte of CHECK SUM values. For detail of CHECK SUM calculation, refer to 22.3.5.4.

Number of transfer bytes	Transfer direction	Transfer data	Description
18	C←T	ACK response to CHECK SUM value Normal state: 0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 5th to 17th byte of receive data have errors.(UART mode only) If receive errors exist, sends a ACK response data 0x18 that means abnormal communications and waits for a next operation command (3rd byte). Then checks 17th byte of CHECK SUM data. If errors exist, sends 0x11 and waits for a next operation command (3rd byte). Finally, checks the result of password verification. If a password error exists, sends a ACK response data 0x11 that means a password error and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
19	C→T	RAM store start Address 31 to 24	Sends a start address of block transfer for RAM store. The 19th byte corresponds to 31st to 24th bit of address.The 22nd byte corresponds to 7th to 0th bit of address. Specify the address to the address 0x2000_0400 through the last address of RAM. The address must be even address.
20	C→T	RAM store start Address 23 to 16	
21	C→T	RAM store start Address 15 to 8	
22	C→T	RAM store start Address 7 to 0	
23	C→T	Number of RAM store bytes 15 to 8	Set the number of bytes to perform block transfer. The 23rd byte corresponds to the15th bit to 8th bit of transfer bytes. The 24th byte corresponds to 7th bit to 0th bit of transfer bytes. Specify the data to be stored in the address from 0x2000_0400 through the last address of RAM.
24	C→T	Number of RAM store bytes 7 to 0	
25	C→T	19th to 24th byte of CHECK SUM value	Send 19th byte to 24th byte of CHECK SUM values
26	C←T	ACK response to CHECK SUM value Normal state: 0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 19th byte to 25th byte of receive data have errors.(UART mode only) If receive errors exist, sends a ACK response data 0x18 that means abnormal communications and waits for a next operation command (3rd byte). Then checks 25th byte of CHECK SUM data. If errors exist, sends 0x11 and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
27 to m	C→T	RAM stored data	Sends same bytes of data specified in 23th bytes to 24 byte for RAM stored data.
m+1	C→T	27 to m byte of CHECK SUM value	Sends 27th byte to m byte of CHECK SUM value
m+2	C←T	ACK response to CHECK SUM value Normal state:0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 27th byte to m+1 byte of receive data have errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Then checks m+1 byte of CHECK SUM data, if errors exist, sends 0x11 and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
–	–	–	If m + 2nd byte of ACK response data is normal ACK response data, the transfer data branches to the address specified in 19th byte to 22 byte.

22.3.7 Transfer Format of Flash memory Chip Erase and Protect Bit Erase

This section shows a transfer format of Flash memory chip erase and protect bit erase commands. Transfer directions in the table are indicated as follows:

Transfer direction (C→T): Controller to TMPM381/383

Transfer direction (C←T): TMPM381/383 to Controller

Number of transfer bytes	Transfer direction	Transfer data	Description
1	C→T	Serial operation mode and baud rate setting	Sends data to determine the serial operation mode. For detail of mode determination, refer to"22.3.5.1 Serial Operation Mode Determination".
		[UART mode] 0x86	Sends 0x86. If UART mode is determined, checks if baud rate setting can be done. If not, operation stops communications.
2	C←T	ACK response to serial operation mode	The 2nd byte of transmit data is a ACK response data to the 1st byte that corresponds to the serial operation setting mode data. If the setting is possible, sets SIO/UART. A receive enable timing is set before transmit buffer is written to the data.
		[UART mode] Normal state: 0x86	If the setting is determined to be possible, sends 0x86. If not, the operation aborts without sending back any response. When the controller finished to send the 1st byte of data, requires a time-out time (5 seconds). If data (0x86) is not normally received within a time-out time, communication is not possible.
3	C→T	Operation command data (0x40)	Sends Flash memory chip erase and protect bit erase command data (0x40).
4	C←T	ACK response to the operation command Normal state: 0x40 Abnormal state: 0xX1 Communication error: 0xX8	ACK response data to the operation command. First, checks if 3rd byte of receive data has errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command data.) Note that in the I/O interface, receive error check is not performed. Then, if the 3rd byte of receive data corresponds to either operation command data in Table 22-12, receive data is echoed back. If the data does not correspond to the command in Table 22-12, sends a ACK response data 0xX1 that means operation command errors, and waits for next operation command. (3rd byte) Upper 4 bits of transmit data are undefined. (Upper 4 bits of immediate before operation command data are used.)
5 to 16	C→T	Password data (12-byte) 0x3F81_FFF4 to 0x3F81_FFFF	If password necessity is set to "none", this data is dummy data. If password necessity is set to "necessary", checks data in the password area. For a method of password area data checking, refer to "22.3.5.3 Password Determination". Compares 5th to 16th byte of receive data with 0x3F81_FFF0 to 0x3F81_FFFF of data of Flash memory in order. If the data does not match, a password error flag is set.
17	C→T	5th to 16th byte CHECK SUM value	Sends 5th byte to 16 byte of CHECK SUM value. For a method of CHECK SUM calculation, refer to"22.3.5.4 CHECK SUM Calculation".

Number of transfer bytes	Transfer direction	Transfer data	Description
18	C←T	ACK response to the CHECK SUM value Normal state: 0x40 Abnormal state: 0x41 Communication error: 0x48	If password necessity is set to "none", sends a normal ACK response data 0x40. If password necessity is set to "necessary", first checks if receive errors exist in the 5th byte to 17th byte receive data. (UART mode only) If a receive error exists, sends a ACK response data 0x48 that means abnormal communications and waits for next operation command. (3rd byte) Then checks 17th byte of CHECK SUM data. If error occurs, sends 0x41 and waits for a next operation command (3rd byte) Finally, checks the result of password verification. If a password error exists, sends a ACK response data 0x41that means a password error and waits for a next operation command (3rd byte) If all procedure normally ends, sends a normal ACK response data 0x40.
19	C→T	Erase enable command data (0x54)	Sends an enable command data (0x54).
20	C←T	ACK response to the erase enable command Normal state: 0x54 Abnormal state: 0xX1 Communication error: 0x58	First, checks if 19th byte of receive data has errors. If receive errors exist, sends a ACK response data (bit 3) 0x58 that means abnormal communication and waits for next operation command (3rd byte). Then, if 19th byte of receive data corresponds to the erase enable command, receive data is echoed back (normal ACK response data). In this case, 0x54 is echoed back and the transfer data branches into Flash memory chip erase process routine. If the data does not correspond to the erase enable command, sends a ACK response data (bit 0) 0xX1 and waits for next operation command. Upper 4 bits of transmit data are undefined. (Upper 4 bits of immediate before operation command data are used.)
21	C→T	ACK response to the erase command (note1) Normal state: 0x4F Abnormal state: 0x4C	If the operation is normally complete, the end code (0x4F) is returned. If erase error occurs, an error code (0x4C) is returned.
−	−	−	Waits for a next operation command.

Note 1: Even when an erase command is performed normally, a Negative acknowledge may be returned by ACK response. Check the FCSR<RDY_BSY> to make sure the command sequence end, and then hold for 200 μ s or more, after that reconfirm the erase status.

22.3.8 Boot Program Whole Flowchart

This section shows a boot program whole flowchart.

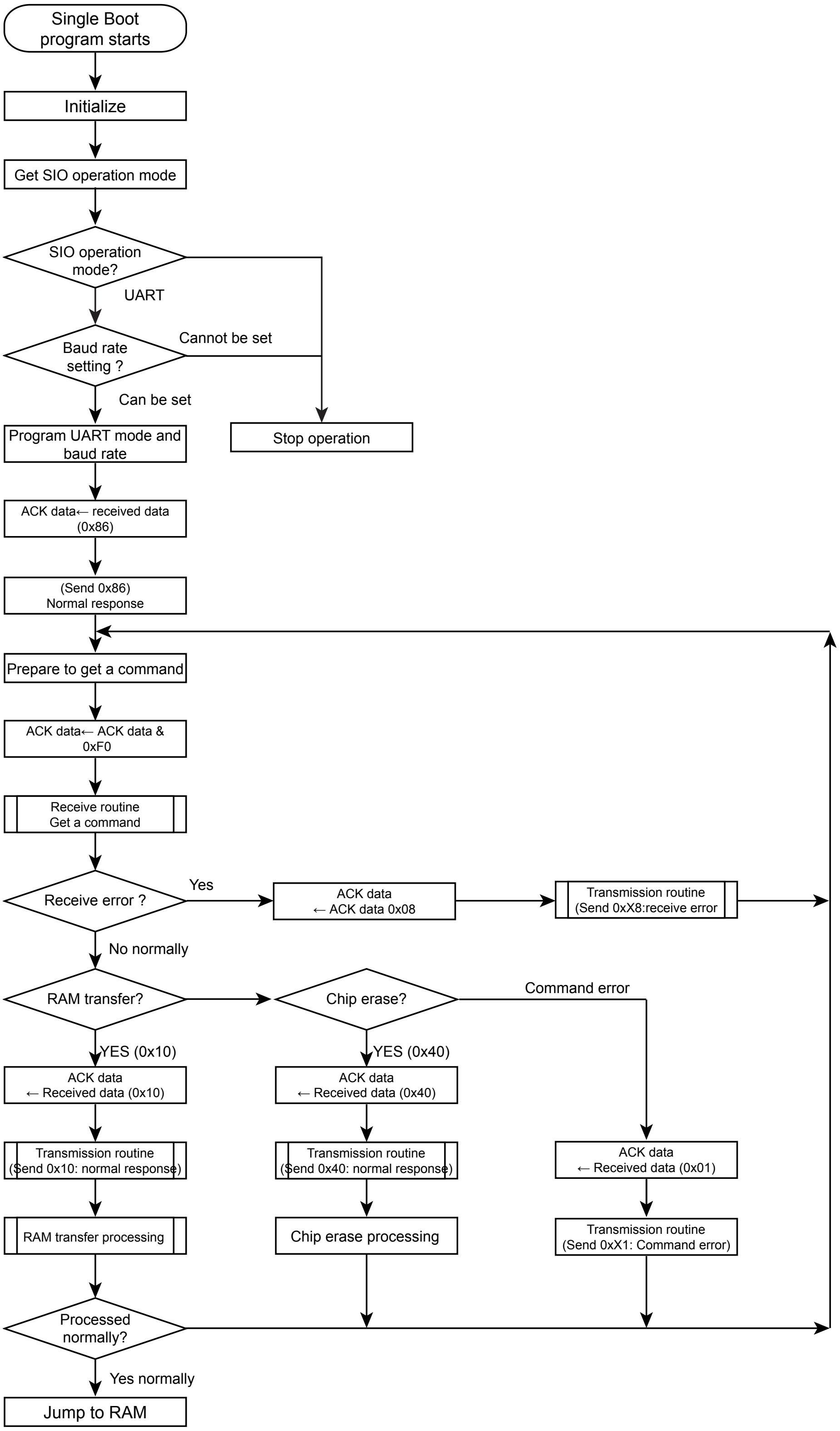

Single Boot program starts
Initialize
Get SIO operation mode
SIO operation mode?
UART
Baud rate setting ?
Cannot be set
Can be set
Program UART mode and baud rate
Stop operation
ACK data← received data (0x86)
(Send 0x86) Normal response
Prepare to get a command
ACK data← ACK data & 0xF0
Receive routine Get a command
Receive error ?
Yes
ACK data ← ACK data 0x08
Transmission routine (Send 0xX8:receive error
No normally
RAM transfer?
Chip erase?
Command error
YES (0x10)
YES (0x40)
ACK data ← Received data (0x10)
ACK data ← Received data (0x40)
Transmission routine (Send 0x10: normal response)
Transmission routine (Send 0x40: normal response)
ACK data ← Received data (0x01)
RAM transfer processing
Chip erase processing
Transmission routine (Send 0xX1: Command error)
Processed normally?
Yes normally
Jump to RAM

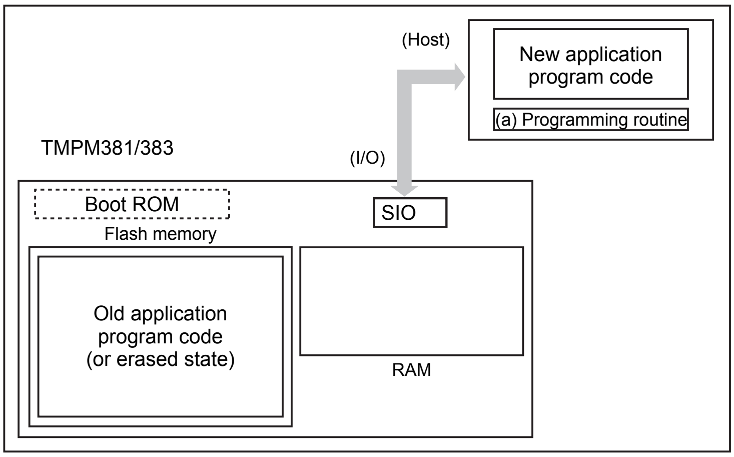
Figure 22-13 Boot program whole flowchart

22.3.9 Reprogramming Procedure of Flash using reprogramming algorithm in the on-chip BOOT ROM

This section describes the reprogramming procedure of the flash using reprogramming algorithm in the on-chip boot ROM.

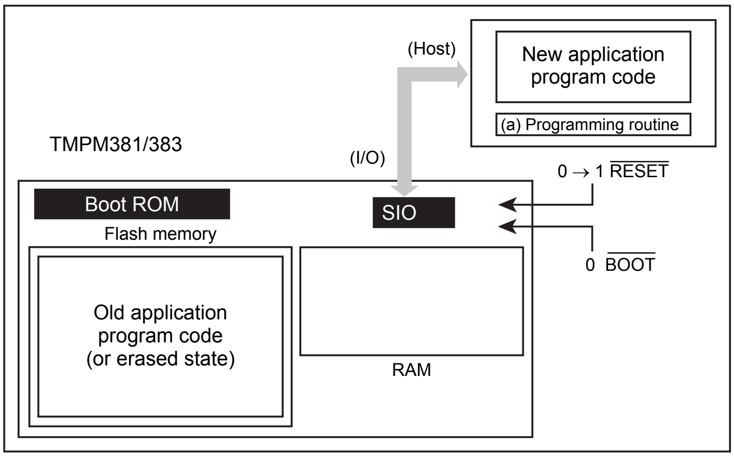
22.3.9.1 Step-1

The condition of Flash memory does not care whether a user program made of former versions has been written or erased. Since a programming routine and programming data are transferred via the SIO0/UART0/, the SIO0/UART0/ must be connected to an external host. A programming routine (a) is prepared on the host.


(Host)
New application program code
(a) Programming routine
TMPM381/383
(I/O)
Boot ROM
SIO
Flash memory
Old application program code (or erased state)
RAM

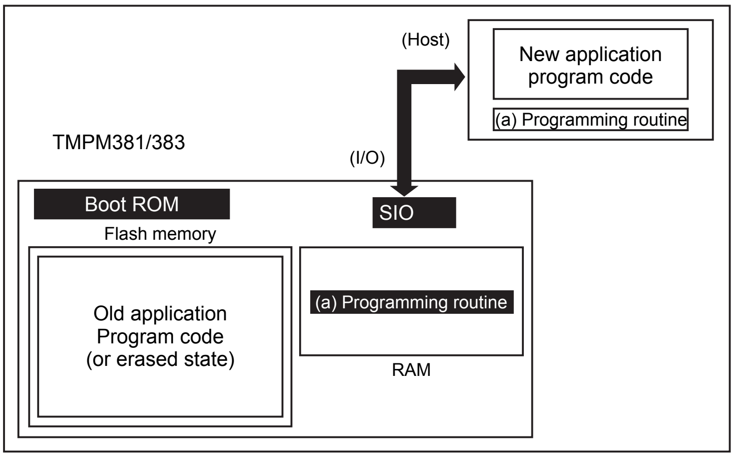
22.3.9.2 Step-2

Release the reset by pin condition setting in the boot mode and boot-up the BOOT ROM. According to the procedure of boot mode, transfer the programming routine (a) via SIO0/UART0/ from the source (host). A password verification with the password in the user application program is performed. (If Flash memory is erased, an erase data (0xFF) is dealt with a password.)



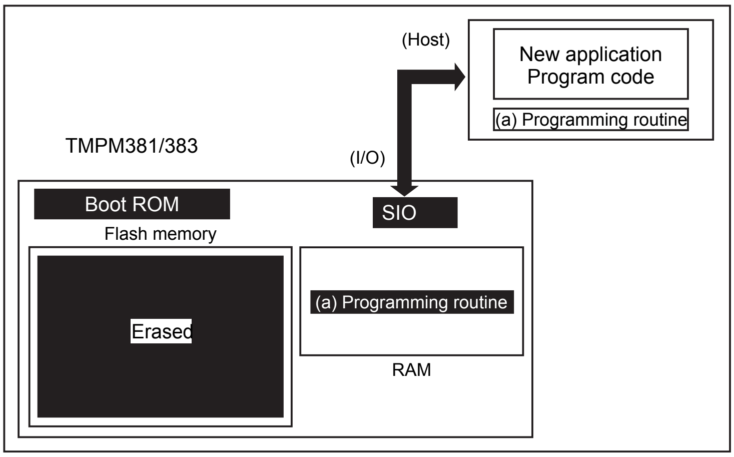
22.3.9.3 Step-3

If the password verification is complete, the boot program transfer a programming routine (a) from the host into the on-chip RAM. The programming routine must be stored in the range from 0x2000_0400 to the end address of RAM.



22.3.9.4 Step-4

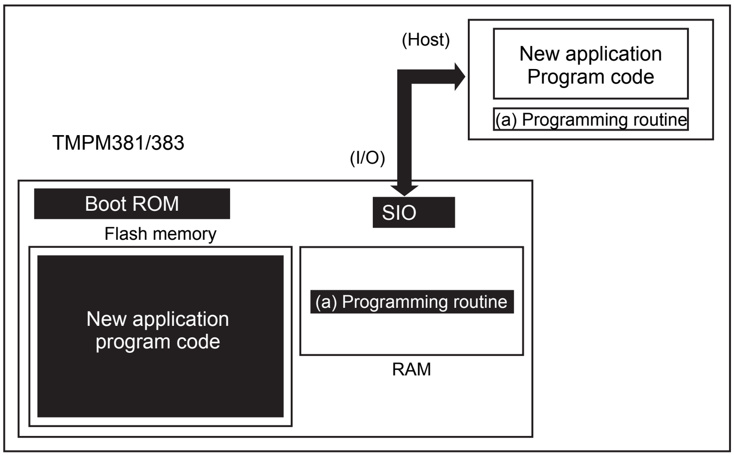
The boot program jumps to the programming routine (a) in the on-chip RAM to erase the flash block containing old application program codes. The Block Erase or Chip Erase command is be used.



22.3.9.5 Step-5

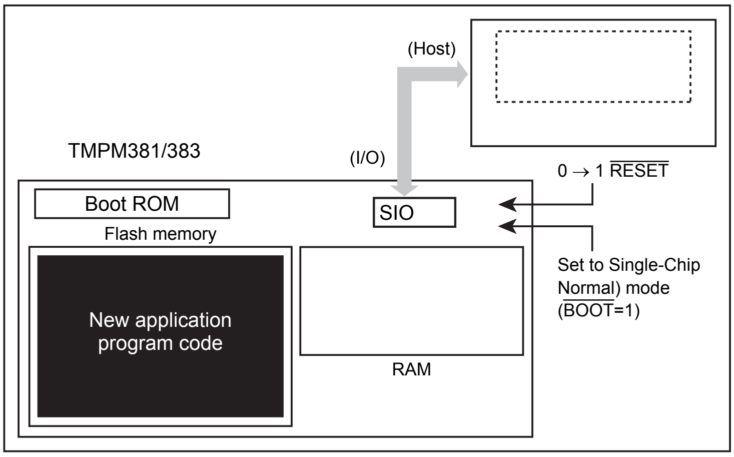
The boot program executes the programming routine (a) to download new application program codes from the host and programs it into the erased flash area. When the programming is complete, the writing or erase protection of that flash area in the user's program must be set.

In the example below, new program code comes from the same host via the same SIO0/UART0/ 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 a hardware board and programming routine to suit your particular needs.



22.3.9.6 Step-6

When programming of 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 device re-boots in the single-chip (Normal) mode to execute the new program.


(Host)
TMPM381/383
(I/O)
0 → 1 RESET
Boot ROM
SIO
Flash memory
Set to Single-Chip Normal) mode (BOOT=1)
New application program code
RAM

22.4 Programming in the User Boot Mode

A 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 user application is different from the serial I/O. It operates in the single chip mode; therefore, a switch from normal mode in which user application is activated in the use boot mode to the user boot mode for programming flash is required. Specifically, add a mode judgment routine to the reset service routine in the user application program.

The condition to switch the modes needs to be set according to the user's system setup condition. Also, a 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 the user boot mode. The data in built-in Flash memory cannot be read out during erase/reprogramming mode. Thus, reprogramming routine must be take place while it is stored in the area outside of Flash memory area. Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental reprogramming. Be sure not to generate interrupt/fault except reset to avoid abnormal termination during the user boot mode.

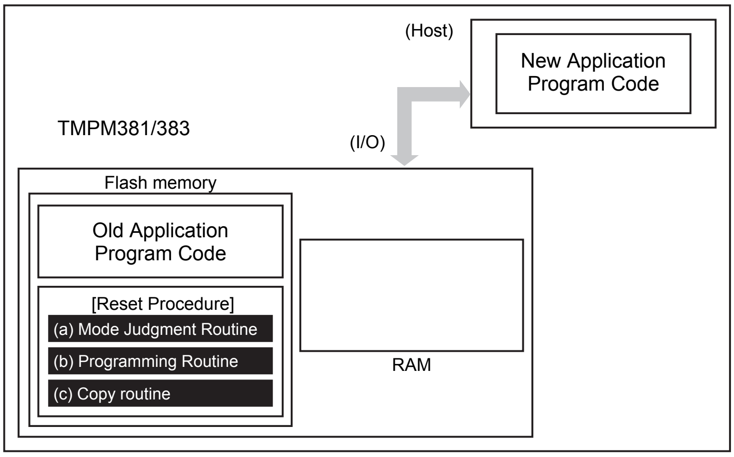
Taking examples from two cases such as the method that reprogramming routine stored in Flash memory (1-A) and transferred from the external device (1-B), the following section explains the procedure. For a detail of the program/erase to Flash memory, refer to "22.2 Detail of Flash Memory".

22.4.1 (1-A) Procedure that a Programming Routine Stored in Flash memory

22.4.1.1 Step-1

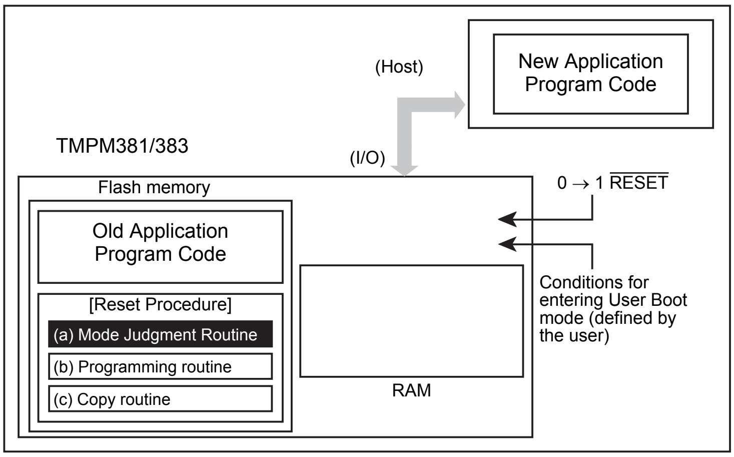
A user determines the conditions (e.g., pin status) to enter the user boot mode and the I/O bus to be used to transfer data. Then suitable circuit design and program are created. Before installing the device on a printed circuit board, write the following three program routines into an arbitrary flash block using programming equipment such as a flash writer.

(a) Mode determination routine:	A program to determine to switch to user boot mode or not
(b) Flash programming routine:	A program to download new program from the host controller and re-program Flash memory
(c) Copy routine:	A program to copy the data described in (a) to the built-in RAM or external memory device



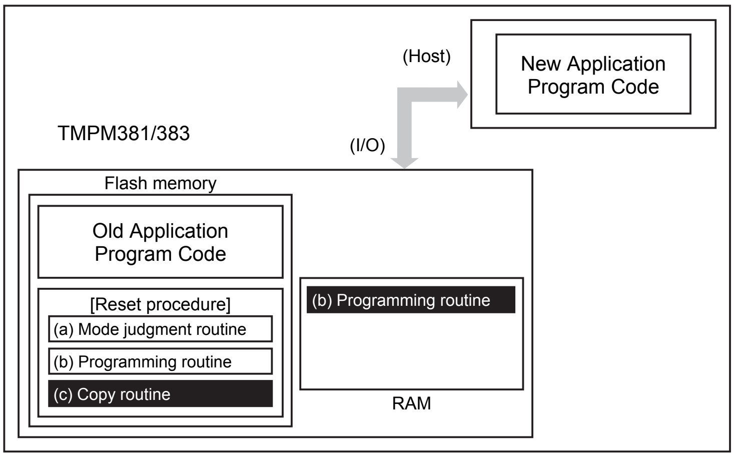
22.4.1.2 Step-2

This section explains the case that a programming routine storied in the reset routine. First, the reset routine determines to enter the user boot mode. If mode switching conditions are met, the device enters the user boot mode to reprogram data.


(Host)
New Application Program Code
TMPM381/383
(I/O)
Flash memory
0 → 1 $\overline{\text{RESET}}$
Old Application Program Code
[Reset Procedure]
(a) Mode Judgment Routine
(b) Programming routine
(c) Copy routine
RAM
Conditions for entering User Boot mode (defined by the user)

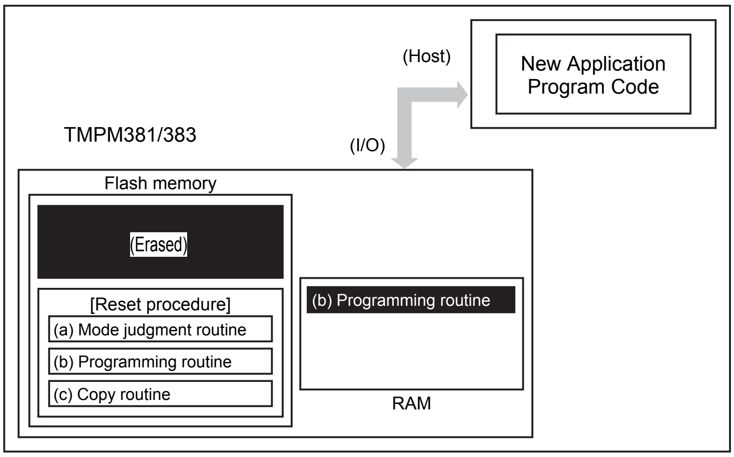
22.4.1.3 Step-3

Once the device enters the user boot mode, execute the copy routine (C) to download the flash programming routine (b) from the host controller to the built-in RAM.



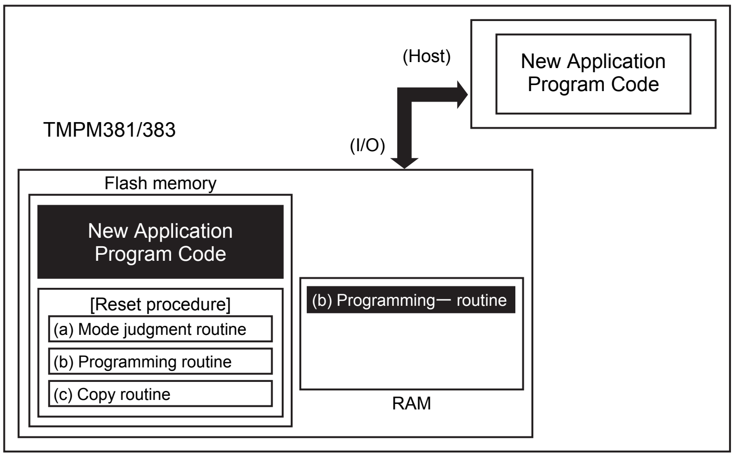
22.4.1.4 Step-4

Jump to the reprogramming routine in the built-in RAM to release the write/erase protection for the old application program, and to erase a flash in block unit.



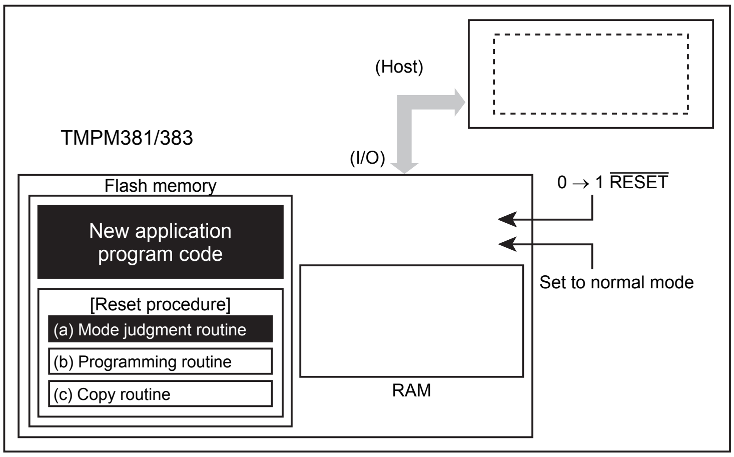
22.4.1.5 Step-5

Continue to execute the flash programming routine to download new program data from the host controller and program it into the erased flash block. When the programming is complete, the write/erase protection of that flash block in the user program area must be set.



22.4.1.6 Step-6

Set $\overline{\text{RESET}}$ to "0". Upon reset, Flash memory is set to the normal mode. After reset, the CPU will start along with the new application program.



22.4.2 (1-B) Procedure that a Programming Routine is transferred from External Host

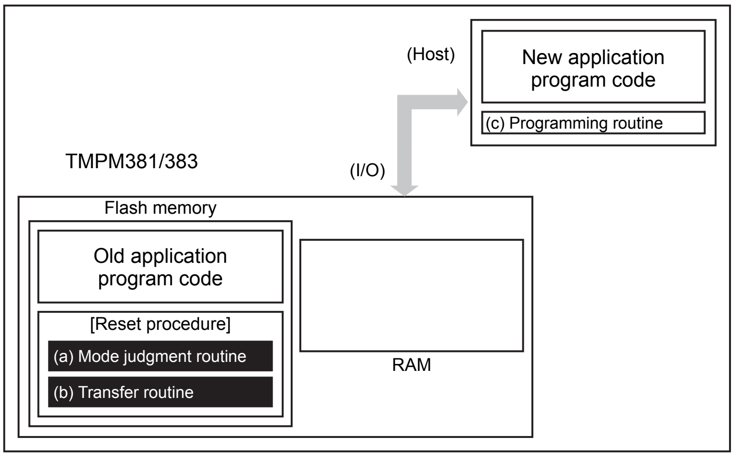
22.4.2.1 Step-1

A user determines the conditions (e.g., pin status) to enter the user boot mode and the I/O bus to be used to transfer data. Then suitable circuit design and program are created. Before installing the device on a printed circuit board, write the following two program routines into an arbitrary flash block using programming equipment such as a flash writer.

(a) Mode determination routine:	A program to determine to switch to reprogramming operation
(b) Transfer routine:	A program to obtain a reprogramming program from the external device.

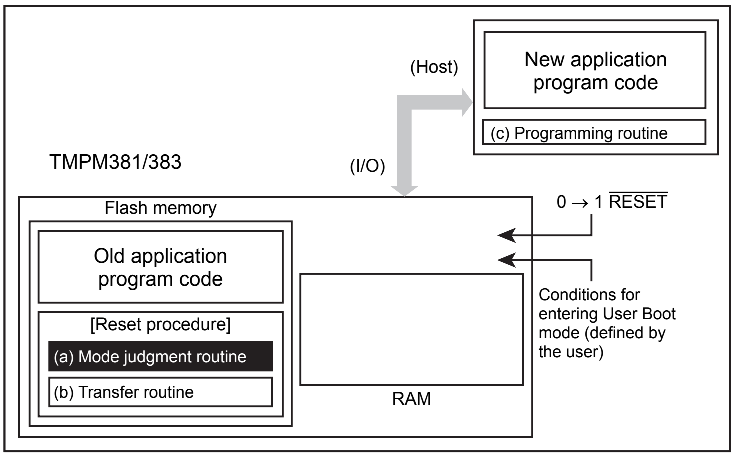
In addition, prepare a reprogramming routine shown below must be stored on the host controller.

(c) Reprogramming routine:	A program to reprogram data



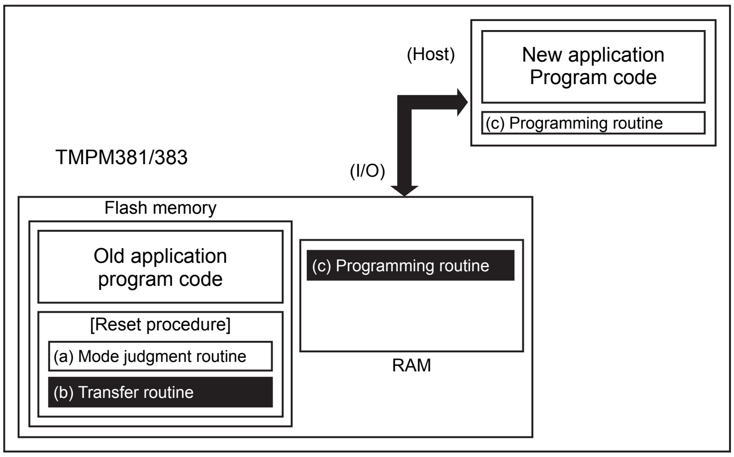
22.4.2.2 Step-2

This section explains the case that a programming routine storied in the reset routine. First, the reset routine determines to enter the user boot mode. If mode switching conditions are met, the device enters the user boot mode to reprogram data.


(Host)
New application program code
(c) Programming routine
TMPM381/383
(I/O)
Flash memory
0 → 1 $\overline{\text{RESET}}$
Old application program code
[Reset procedure]
(a) Mode judgment routine
(b) Transfer routine
RAM
Conditions for entering User Boot mode (defined by the user)

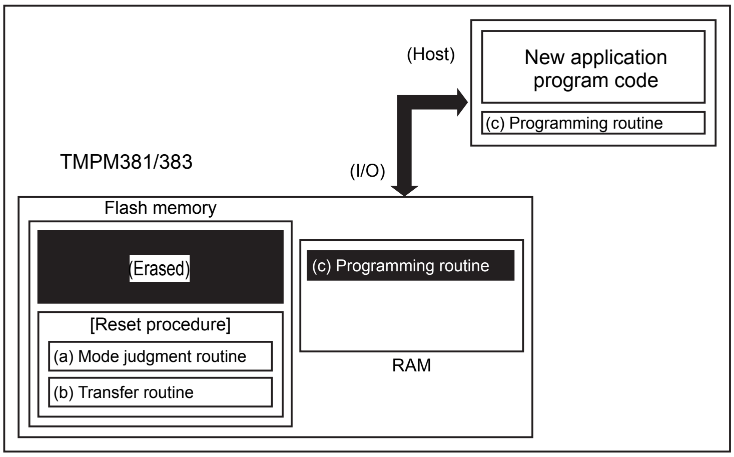
22.4.2.3 Step-3

Once the device enters the user boot mode, execute the transfer routine (b) to download the programming routine (c) from the host controller to the built-in RAM.



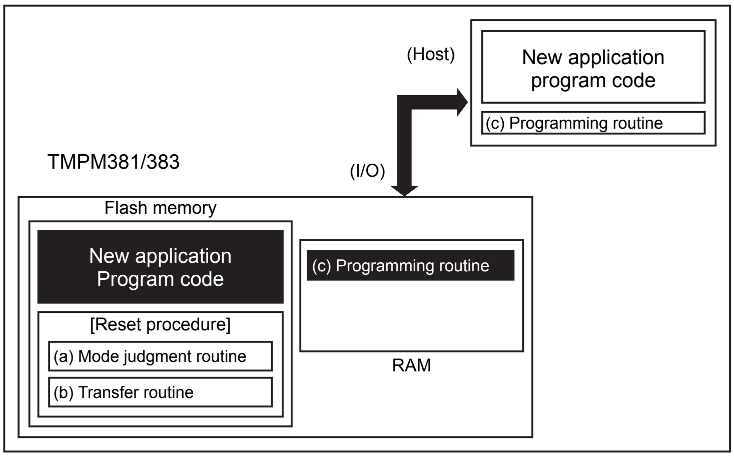
22.4.2.4 Step-4

Jump to the reprogramming routine in the built-in RAM to release the write/erase protection for the old application program, and to erase a flash in block unit.



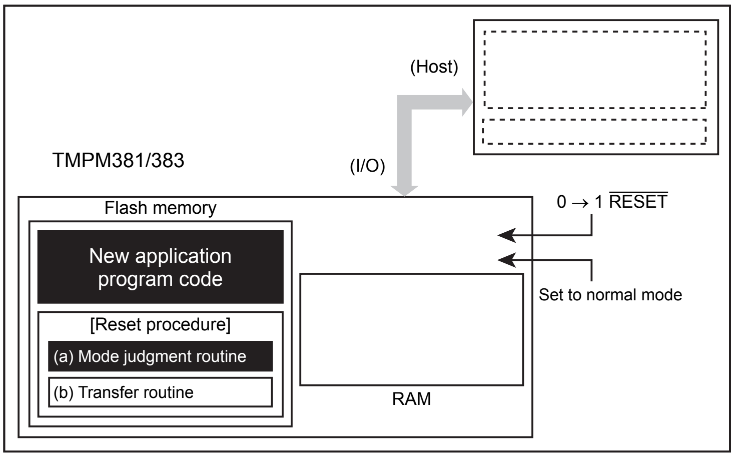
22.4.2.5 Step-5

Continue to execute the flash programming routine (c) to download new program data from the host controller and program it into the erased flash block. When the programming is complete, the write/erase protection of that flash block in the user program area must be set.



22.4.2.6 Step-6

Set $\overline{\text{RESET}}$ to "0". Upon reset, Flash memory is set to the normal mode. After reset, the CPU will start along with the new application program.



23. Debug Interface

23.1 Specification Overview

TMPM381/383 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[1:0], 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" .

23.2 SWJ-DP

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

23.3 ETM

ETM supports two data signal pins (TRACEDATA[1:0]), one clock signal pin (TRACECLK) and trace output from Serial Wire Viewer (SWV).

23.4 Pin Functions

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

The PB3 and PB4 pins are shared between the JTAG debug port function and the Serial Wire Debug Port function. The PB5 pin is shared between the JTAG debug port function and the SWV trace output function.

Table 23-1 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	PB3	Input	JTAG Test Mode Selection	I / O	Serial Wire Data Input/Output
TCK / SWCLK	PB4	Input	JTAG Test Check	Input	Serial Wire Clock
TDO / SWV	PB5	Output	JTAG Test Data Output	(Output)(Note)	(Serial Wire Viewer Output)
TDI	PB6	Input	JTAG Test Data Input	-	-
$\overline{\text{TRST}}$	PB7	Input	JTAG Test $\overline{\text{RESET}}$	-	-
TRACECLK	PB0	Output	TRACE Clock Output		
TRACEDATA0	PB1	Output	TRACE DATA Output0		
TRACEDATA1	PB2	Output	TRACE DATA Output1		

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

After reset, PB3, PB4, PB5, PB6 and PB7 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 PB3 and PB5 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 PB4 is configured as a debug function pin, it prevents a low power consumption mode from being fully effective. Configure PB4 to function as a general-purpose port if the debug function is not used.

Table 23-2 summarizes the debug interface pin and related port settings after reset.

Table 23-2 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)
PB3	TMS/SWDIO	1	1	1	1	0
PB4	TCK/SWCLK	1	1	0	0	1
PB5	TDO/SWV	1	0	1	0	0
PB6	TDI	1	1	0	1	0
PB7	$\overline{\text{TRST}}$	1	1	0	1	0
PB0	TRACECLK	0	0	0	0	0
PB1	TRACEDATA0	0	0	0	0	0
PB2	TRACEDATA1	0	0	0	0	0

－ : Don't care

23.5 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.

23.6 Connection with a Debug Tool

23.6.1 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.

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

When setting a debugging interface terminal to a general-purpose port by a user's program after reset release,after that the control from a debugging tool is impossible.

Please note that it is necessary to prepare for the structure which changes the general-purpose port to the debugging interface function by some kind of methods to connect a debugging tool again..

Table 23-3 Example Table of using debug interface pins

	Debug interface pins						
	$\overline{\text{TRST}}$	TDI	TDO / SWV	TCK / SWCLK	TMS / SWDIO	TRACE DATA[1:0]	TRACE CLK
JTAG+SW (After reset)	o	o	o	o	o	×	×
JTAG+SW (without TRST)	(Note)	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)

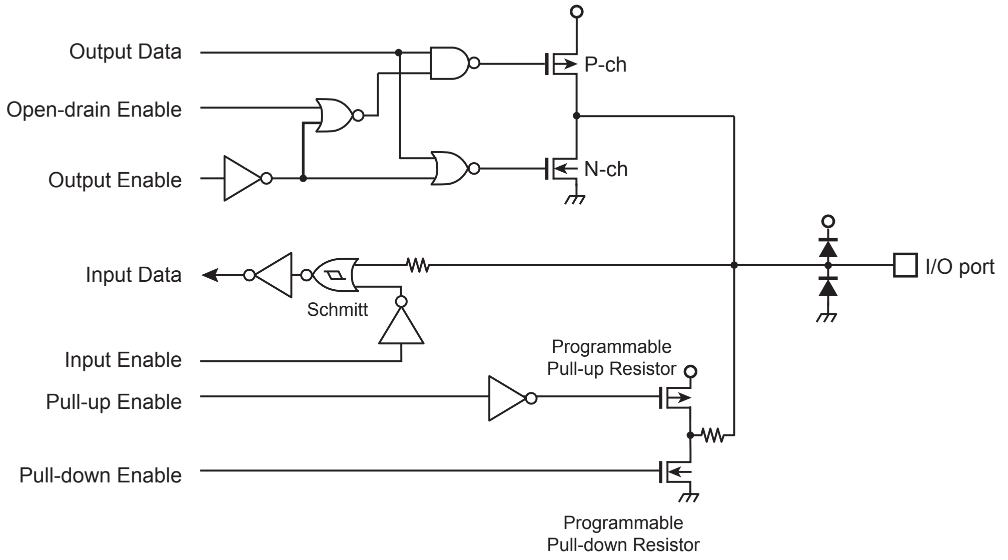
Note: For the treatment of the pin of which the $\overline{\text{TRST}}$ function is assigned, select the $\overline{\text{TRST}}$ function with the function register and set the pin to OPEN or "High level".

24. Port Section Equivalent Circuit Schematic

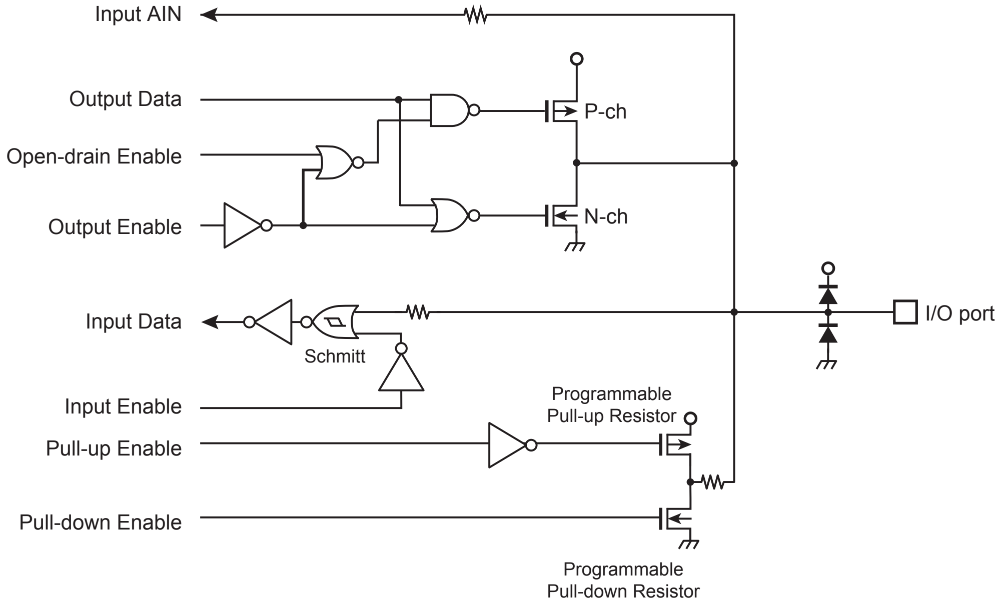
Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

The input protection resistance ranges from several tens of Ω to several hundred Ω. Feedback resistor and Damping resistor are shown with a typical value.

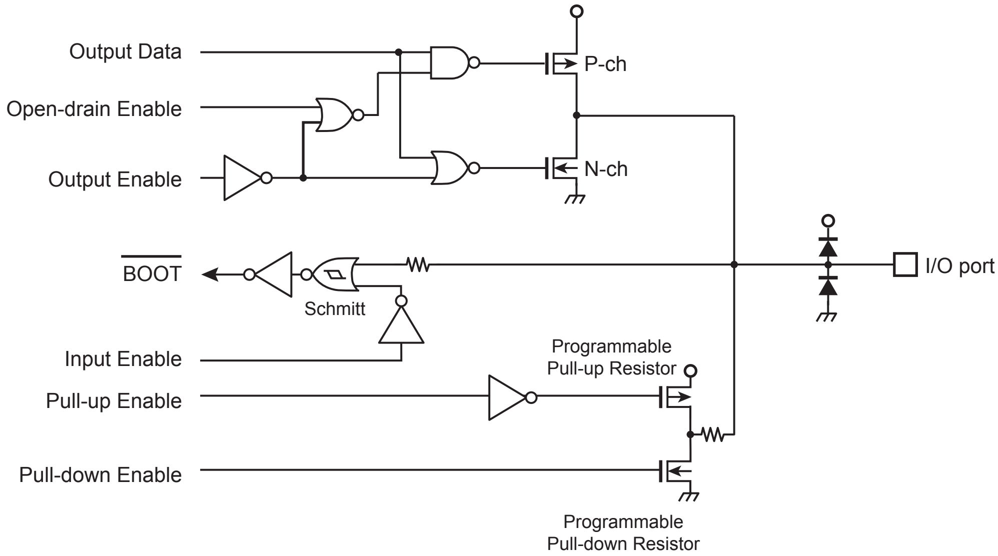
24.1 PA0 to 7, PB0 to 7, PC0 to 7, PD0 to 6, PE0 to 7, PF0 to 4, PG0 to 7, PL2, PN0 to 7



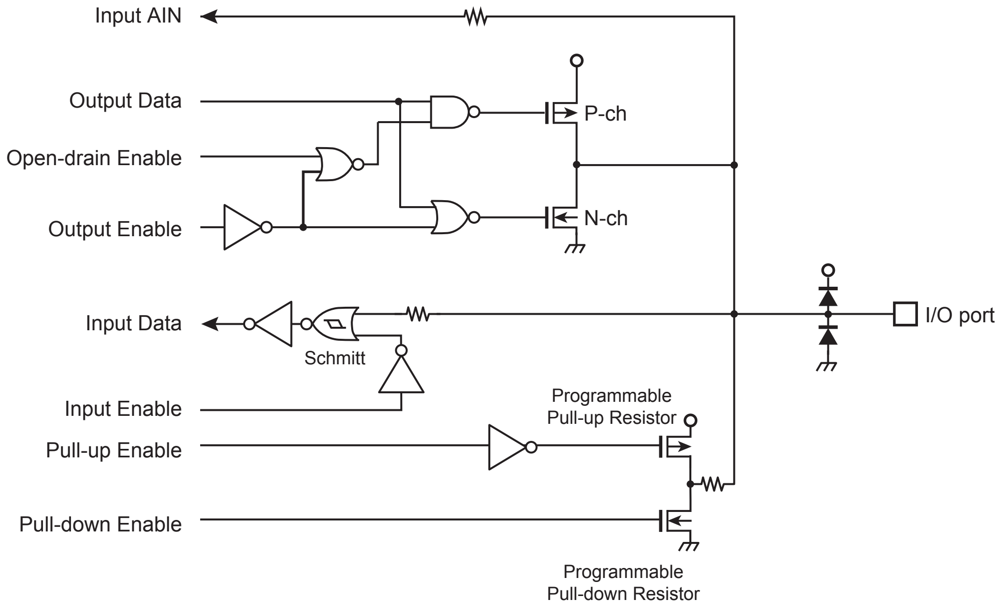
24.2 PH0 to 7, PI0 to 1, PJ0 to 7



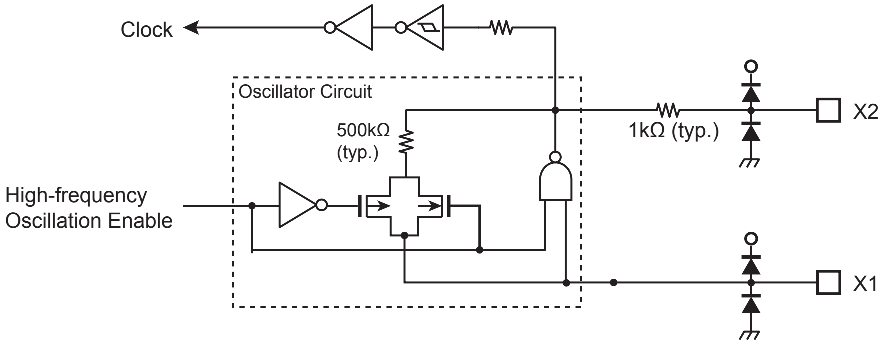
24.3 PL0



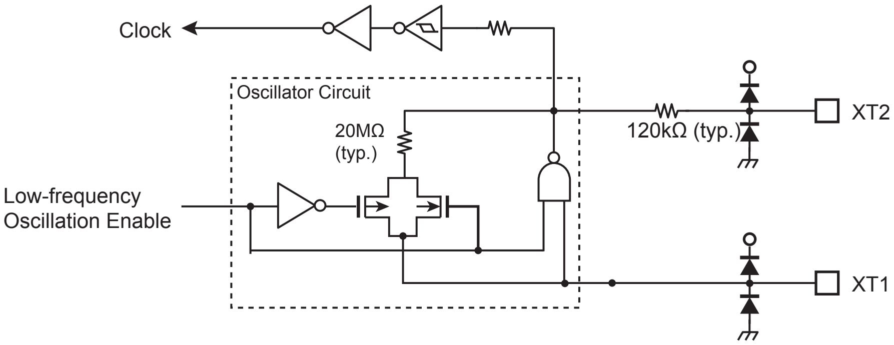
24.4 PM0 to 1, PP0 to 1



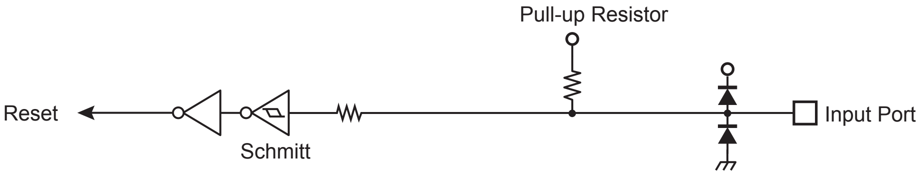
24.5 X1, X2



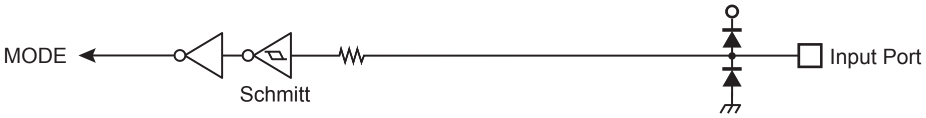
24.6 XT1, XT2



24.7 $\overline{\text{RESET}}$

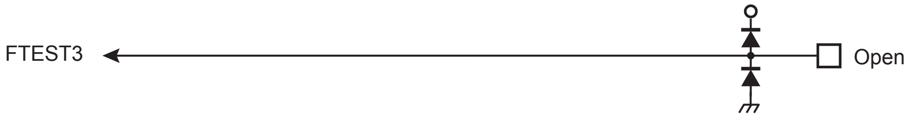


24.8 MODE



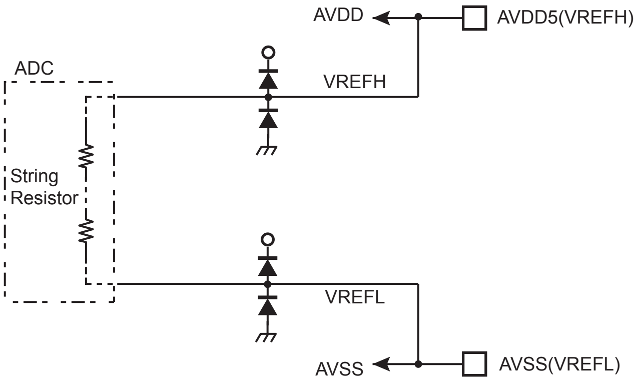
(Note)MODE pin is fixed to GND.

24.9 FTEST3



(Note)FTEST3 pin is fixed to Open.

24.10 VREFH, VREFL



25. Electrical Characteristics

25.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		DVDD5	−0.3 to 6.0	V
		RVDD5	−0.3 to 6.0	
		AVDD5	−0.3 to 6.0	
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}	−55 to 125	°C
Operating Tempera-ture	Except during and debug Flash W/E	T_{OPR}	−40 to 85	°C
	During Flash W/E and debug		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 ICblowup and/or burning.**

25.2 DC Electrical Characteristics (1/3)

DVDD5 = RVDD5 = AVDD5 = 3.9V to 5.5V , DVSS = AVSS = 0V, Ta = −40 to 85 °C

Parameter		Symbol	Condition	Min	Typ. (Note 1)	Max	Unit
Supply voltage (Note 2)		DVDD5 AVDD5 RVDD5	f_{OSC} = 8 to 10 MHz fsys = 1 to 40 MHz fs = 30 to 34 kHz	3.9	–	5.5	V
Capacitance for VOUT3 (Note 3)		C_{Out3}	DVDD5= 3.9 ~ 5.5V	–	4.7	–	μF
Capacitance for VOUT15 (Note 4)		C_{Out15}	DVDD5= 3.9 ~ 5.5V	–	4.7	–	μF
Low-level Input voltage	PORT A,B,C,D,E,F,G,L,M,N,P	V_{IL1}	3.9 V ≤ DVDD5 ≤ 5.5 V	−0.3	–	0.25 DVDD5	V
	PORT H,I,J	V_{IL2}	3.9 V ≤ AVDD5 ≤ 5.5 V	−0.3	–	0.25 DVDD5	
High-level Input voltage	PORT A,B,C,D,E,F,G,L,M,N,P	V_{IH1}	3.9 V ≤ DVDD5 ≤ 5.5 V	0.75 DVDD5	–	DVDD5 + 0.3	V
	PORT H,I,J	V_{IH2}	3.9 V ≤ AVDD5 ≤ 5.5 V	0.75 AVDD5	–	AVDD5 + 0.3	
Low-level Output voltage	PORT A,B,C,D,E,F,G,L,M,N,P	V_{OL1}	DVDD5 ≥ 3.9V I_{OL} = 1.6 mA	–	–	0.4	V
	PORT H,I,J	V_{OL2}	AVDD5 ≥ 3.9V I_{OL} = 1.6 mA	–	–	0.4	
High-level Output voltage	PORT A,B,C,D,E,F,G,L,M,N,P	V_{OH1}	DVDD5 ≥ 3.9V I_{OH} = −1.6 mA	DVDD5 − 0.4	–	–	V
	PORT H,I,J	V_{OH2}	AVDD5 ≥ 3.9V I_{OH} = −1.6 mA	AVDD5 − 0.4	–	–	
Input leakage current		I_{LI}	0.0 V ≤ V_{IN} ≤ DVDD5 0.0 V ≤ V_{IN} ≤ AVDD5	–	0.02	±5	μA
Output leakage current		I_{LO}	0.2 V ≤ V_{IN} ≤ DVDD5 -0.2 V 0.2 V ≤ V_{IN} ≤ AVDD5 -0.2 V	–	0.05	±10	
Pull-up resistor at Reset		R_{RST}	3.9 V ≤ DVDD5 ≤ 5.5 V	38.5	50	71.4	kΩ
Schmitt trigger input width		V_{TH}	3.9 V ≤ DVDD5 ≤ 5.5 V 3.9 V ≤ AVDD5 ≤ 5.5 V	0.3	0.6	–	V
Programmable pull-up/pull-down resistor		P_{KH}	3.9 V ≤ DVDD5 ≤ 5.5 V 3.9 V ≤ AVDD5 ≤ 5.5 V	38.5	50	71.4	kΩ
Pin capacitance (Except power supply pins)		C_{IO}	fc = 1 MHz	–	–	10	pF

Note 1: Ta = 25 °C, DVDD5= AVDD5 = RVDD5 = 5 V, unless otherwise noted.

Note 2: The same voltage must be supplied to DVDD5, AVDD5, and RVDD5.

Note 3: VOUT3 pin should be connected to GND via a capacitance.

Note 4: VOUT15 pin should be connected to GND via a capacitance.

25.3 DC Electrical Characteristics (2/3)

DVDD5 = RVDD5 = AVDD5 = 3.9V to 5.5V , DVSS = AVSS = 0V, Ta = −40 to 85 °C

Parameter	Symbol	Condition	Min	Typ. (Note 1)	Max	Unit
Low-level output current	I_{OL}	Per pin	−	−	2	mA
	ΣI_{OL1}	Per group 3.9 V ≤ DVDD5 ≤ 5.5 V GrL1 = <PA0-7/PE0-5/PG0-7> GrL2 = <PB0-7/PD0-6/PF0-4/PL0> GrL3 = <PC0-7/PM0-1/PP0-1> GrL4 = <PE6-7/PL2/PN0-7>	−	−	20	mA
	ΣI_{OL2}	Per group 3.9 V ≤ AVDD5 ≤ 5.5 V GrL5 = <PH0-7/PI0-1/PJ0-7>	−	−	9	mA
	ΣI_{OL}	Total, all ports	−	−	30	mA
High-level output current	I_{OH}	Per pin	−	−	-2	mA
	ΣI_{OH1}	Per group 3.9 V ≤ DVDD5 ≤ 5.5 V GrH1 = <PA0-7/PE0-3/PG0-7/PM0-1/PP0-1> GrH2 = <PB0-7/PC0-7/PD0-6/PF0-4/PL0> GrH3 = <PE4-7/PL2/PN0-7>	−	−	-20	mA
	ΣI_{OH2}	Per group 3.9 V ≤ AVDD5 ≤ 5.5 V GrH4 = <PH0-7/PI0-1/PJ0-7>	−	−	-9	mA
	ΣI_{OH}	Total, all ports	−	−	-30	mA

25.4 DC Electrical Characteristics (3/3)

DVDD5 = RVDD5 = AVDD5 = 3.9V to 5.5V , DVSS = AVSS = 0V, Ta = −40 to 85 °C

Parameter	Symbol	Condition	Min	Typ. (Note 1)	Max	Unit
NORMAL (Note 2)	I_{DD}	fsys = fc = 40 MHz (fosc = 10MHz)	−	22	27	mA
IDLE (Note 3)			−	16	20	mA
SLOW		fsys = fs = 32.768 kHz	−	400	1700	μA
SLEEP (Note 4)			−	120	1200	μA
STOP		−	−	100	1180	μA

Note 1: Ta = 25 °C, DVDD5 = RVDD5 = AVDD5 = 5 V, unless otherwise noted.

Note 2: Measurement condition of I_{DD} NORMAL :
Execution program: Dhrystone V2.1 (built-in FLASH operation)
All peripheral functions operate excluding A/DC.

Note 3: Measurement condition of I_{DD} IDLE:
CPU is stopped, all peripheral functions operate excluding A/DC.

Note 4: Measurement condition of I_{DD} SLEEP:
All peripheral functions stopped.
CPU is stopped, using RMC,RTC only.

25.5 12/10-bit AD Converter Electrical Characteristics

DVDD5 = RVDD5 = 4.5V to 5.5V , DVSS = AVSS = 0V, Ta = −40 to 85 °C

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage (+) (Note 1)	AVDD5	AVDD5 = V_{REFH}	DVDD5 − 0.2	−	DVDD5	V
Analog reference voltage	AVSS	AVSS = V_{REFL}	0	−	0	V
Analog input voltage	V_{AIN}	−	AVSS	−	AVDD5	V
Power supply current of analog reference voltage	I_{REF}	I_{REF} ON (During AD conversion)	−	7.5	10.0	mA
		I_{REF} ON (During AD stop)	−	3.5	5	mA
		I_{REF} OFF (During STOP MODE)	−	3	70	µA
INL error	−	12bit mode AIN resistance ≤ 600 Ω AIN load capacitance ≥ 0.1 µF Conversion time ≥ 1.85 µs	−	−	± 9	LSB (Note 2)
DNL error			−	−	+ 6 ~ −1	
Offset error			−	−	± 5	
Full-scale error			−	−	+ 8 ~ −2	
Total error			−	−	+ 12 ~ −8	
INL error	−	10bit mode AIN resistance ≤ 600 Ω AIN load capacitance ≥ 0.1µF Conversion time ≥ 1.70 µs	−	−	± 3	LSB (Note 3)
DNL error			−	−	± 2	
Offset error			−	−	± 3	
Full-scale error			−	−	± 3	
Total error			−	−	± 4	

Note 1: A/D when using separate power supply for the converter, you must keep this condition.

Note 2: 1LSB = (AVDD5 − AVSS)/4096 [V]

Note 3: 1LSB = (AVDD5 − AVSS)/1024 [V]

Note 4: The relevant pin for I_{REF} is AVDD5, so that the current flowing into AVDD5 is the power supply current AVDD5 + I_{REF} .

Note 5: Peripheral functions are disabled.

Note 6: In the range whose Power-line is 3.9 V ≤ DVDD5=RVDD5 < 4.5V, does not guarantee a A/D converter.

25.6 AC Electrical Characteristics

25.6.1 AC Measurement Condition

The AC characteristics data of this chapter is measured under the following conditions unless otherwise noted.

- Output levels: High = 0.8 × DVDD5, Low = 0.2 × DVDD5
- Input levels: Refer to low-level input voltage and high-level input voltage in "DC Electrical Characteristics".
- Load capacity: CL = 30pF

25.6.2 Serial Channel (SIO/UART)

25.6.2.1 I/O Interface Mode

In the table below, the letter x represents the SIO operation clock SCLK Clock Low width (input) 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

[Input]

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	4x	−	100	−	ns
SCLK Clock Low width (input)	t_{SCL}	4x	−	100	−	
SCLK cycle	t_{SCY}	t_{SCH} + t_{SCL}	−	200	−	
Valid Data Input ← SCLK rise or fall (Note 1)	t_{SRD}	30	−	30	−	
SCLK rise or fall → Input Data hold (Note 1)	t_{HSR}	x + 30	−	55	−	

[Output]

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	4x	−	120 (Note 3)	−	ns
SCLK Clock Low width (input)	t_{SCL}	4x	−	120 (Note 3)	−	
SCLK cycle	t_{SCY}	t_{SCH} + t_{SCL}	−	240	−	
Output Data ← SCLK rise or fall (Note 1)	t_{OSS}	t_{SCY}/2 − 3x− 45	−	0 (Note 2)	−	
SCLK rise or fall → Output Data hold (Note 1)	t_{OHS}	t_{SCY}/2	−	120	−	

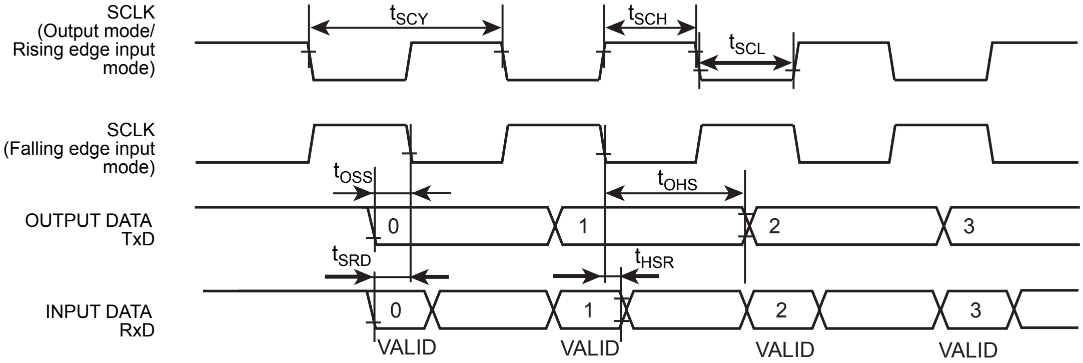
Note 1: SCLK rise/fall : SCLK rise mode uses the rise timing of SCLK. SCLK fall mode uses the fall timing of SCLK.

Note 2: Use the cycle of SCLK in a range where the calculation value keeps positive.

Note 3: The value indicates a minimum value that enables t_{OSS} to be zero or more.

(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 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	−	



25.6.3 Serial Bus Interface (I2C/SIO)

25.6.3.1 I2C Mode

In the table below, the letter x represents the I2C/SIO operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.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 clock low-width (Input) (Note 1)	t_{LOW}	−	−	4.7	−	1.3	−	μs
SCL clock high-width (Input) (Note 2)	t_{HIGH}	−	−	4.0	−	0.6	−	μs
Setup time for a repeated START condition	$t_{SU;\ STA}$	(Note 5)	−	4.7	−	0.6	−	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}	(Note 5)	−	4.7	−	1.3	−	μs

Note 1: SCL clock Low width (output): $(2^{n-1} + 58)/x$

Note 2: SCL clock High width (output): $(2^{n-1} + 14)/x$

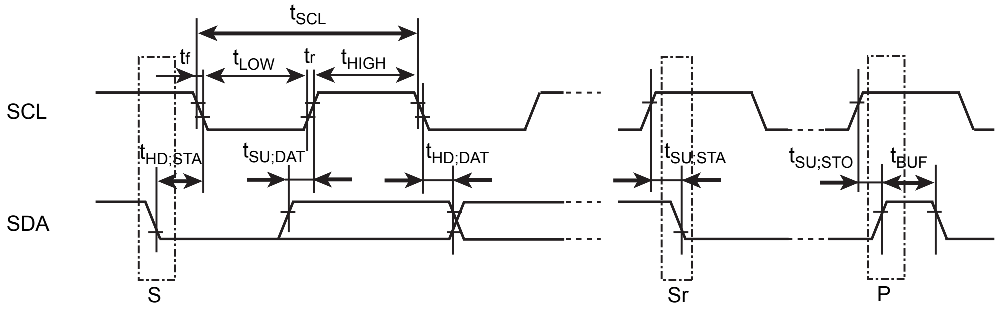
On I2C-bus specification, maximum Speed of Standard Mode/fast mode is 100kHz/400khz. Internal SCL Frequency setting should comply with fsys and Note1 & Note2 shown above.

Note 3: The output data hold time is equal to 4x 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: Re-start condition
P: Stop condition

25.6.3.2 Clock-Synchronous 8-bit SIO mode

In the table below, the letter x represents the I2C/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) SCK input mode (for an SCK signal with a 50% duty cycle)

[Input]

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}	8x	−	200	−	
Valid Data input ← SCK rise	t_{SRD}	30 − x	−	5	−	
SCK rise → Input Data hold	t_{HSR}	2x + 30	−	80	−	

[Output]

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCK Clock High width (input)	t_{SCH}	4x	−	120 (Note 2)	−	ns
SCK Clock Low width (input)	t_{SCL}	4x	−	120 (Note 2)	−	
SCK cycle	t_{SCY}	8x	−	240	−	
Output Data ← SCK rise	t_{OSS}	$t_{SCY}/2 - 3x - 45$	−	0 (Note 1)	−	
SCK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 + x$	−	145	−	

Note 1: Keep this value positive by adjusting SCK cycle.

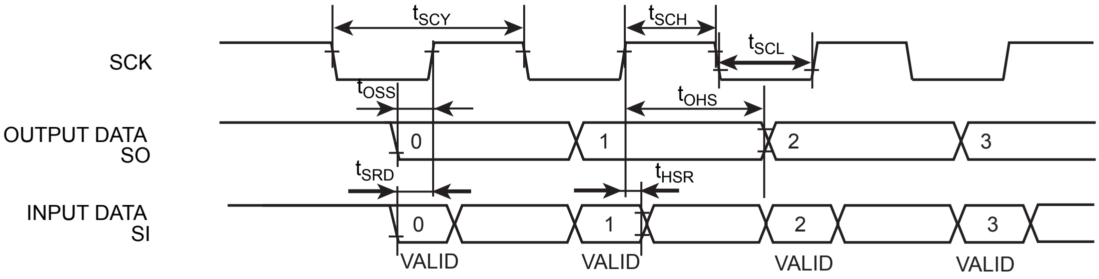
Note 2: The value indicates a minimum value that enables t_{OSS} to be zero or more.

(2) SCK output mode (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 (Note1)	−	400	−	ns
Output Data ← SCK rise	t_{OSS}	t_{SCY}/2 − 20 (Note2)	−	180	−	
SCK rise → Output Data hold	t_{OHS}	t_{SCY}/2 − 20	−	180	−	
Valid Data input ← SCK rise	t_{SRD}	x + 45	−	70	−	
SCK rise → Input Data hold	t_{HSR}	0	−	0	−	

Note 1: SCK cycle after automatic wait becomes 14x.

Note 2: t_{OSS} after automatic wait may be t_{SCY}/2−x−20.



25.6.4 Synchronous Serial Interface (SSP)

25.6.4.1 AC Measurement Condition

The letter "T" used in the equations in the table represents the period of the input clock (f_{PCLK}) into the internal prescaler.

- Output levels : High = 0.7 × DVDD5, Low = 0.3 × DVDD5
- Input levels: High = 0.9 × DVDD5, Low = 0.1 × DVDD5
- Load capacitance: CL=30pF
- Ta = −40 to 85°C

Note: The "Equation" column in the table shows the specifications under the conditions DVDD5 = 3.9V to 5.5V.

Parameter	Symbol	Equation		fsys=40MHz (m=4,n=12)	Unit
		Min	Max		
SPCLK cycle (master)	T_m	(m)T At least 100ns or more	−	100 (10MHz)	ns
SPCLK cycle (slave)	T_s	(n)T	−	300 (3.3MHz)	
SPCLK rise up time	t_r	−	15	15	
SPCLK fall down time	t_f	−	15	15	
Master mode: SPCLK low-level pulse width	t_{WLM}	(m)T/2 - 20.0	−	30	
Master mode: SPCLK high-level pulse width	t_{WHM}	(m)T/2 - 20.0	−	30	
Slave mode: SPCLK low-level pulse width	t_{WLS}	(n)T/2 - 10.0	−	140	
Slave mode: SPCLK high-level pulse width	t_{WHS}	(n)T/2 - 10.0	−	140	
Master mode: SPCLK rise/fall to output data valid	t_{ODSM}	−	15	15	
Master mode: SPCLK rise/fall to output data hold	t_{ODHM}	(m)T/2 - 15	−	35	
Master mode: SPCLK rise/fall to input data valid delay time	t_{IDSM}	35	−	35	
Master mode: SPCLK rise/fall to input data hold	t_{IDHM}	5	−	5	
Master mode: SPFSS valid to SPCLK rise/fall	t_{OFSM}	(m)T - 15	(m)T + 15	85 to 115	
Slave mode: SPCLK rise/fall to output data valid delay time	t_{ODSS}	−	(3T) + 35	110	
Slave mode: SPCLK rise/fall (Output data hold)	t_{ODHS}	(n)T/2 + (2T)	−	200	
Slave mode: SPCLK rise/fall to input data valid delay time	t_{IDSS}	10	−	10	
Slave mode: SPCLK rise/fall to input data hold	t_{IDHS}	(3T) + 15	−	90	
Slave mode: SPFSS valid to SPCLK rise/fall	t_{OFSS}	(n)T - 20	−	280	

Note: Baud rate clock is set under below condition:

- Master mode:

  m = (<CPSDVSR> × (1 + <SCR>)) = fsys/SPCLK

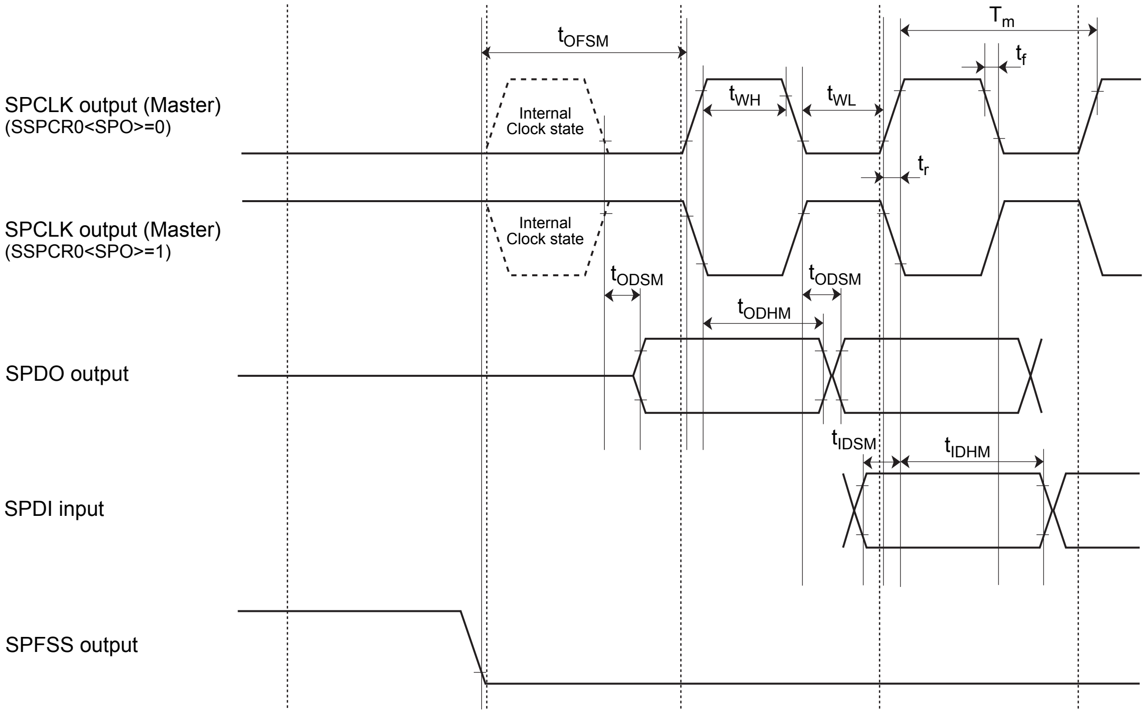
  <CPSDVR>is set only even number and "m" must set between the range of $65024 \geq m \geq 2$.
- Slave mode

  n = fsys/SPCLK ($65024 \geq n \geq 12$)

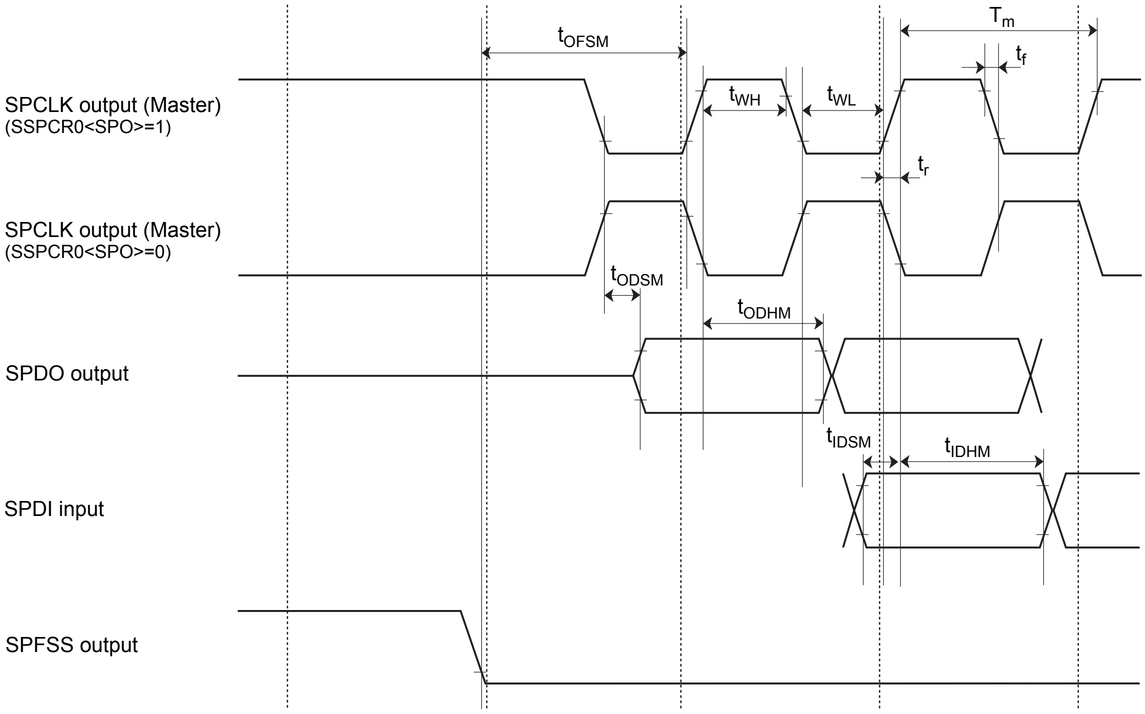
25.6.4.2 SSP SPI mode (Master)

- fsys ≥ 2 × SPxCLK (Maximum)
- fsys ≥ 65024 × SPxCLK (Minimum)

(1) Master SSPCR0<SPH>="0" (Data is latched on the first edge.)



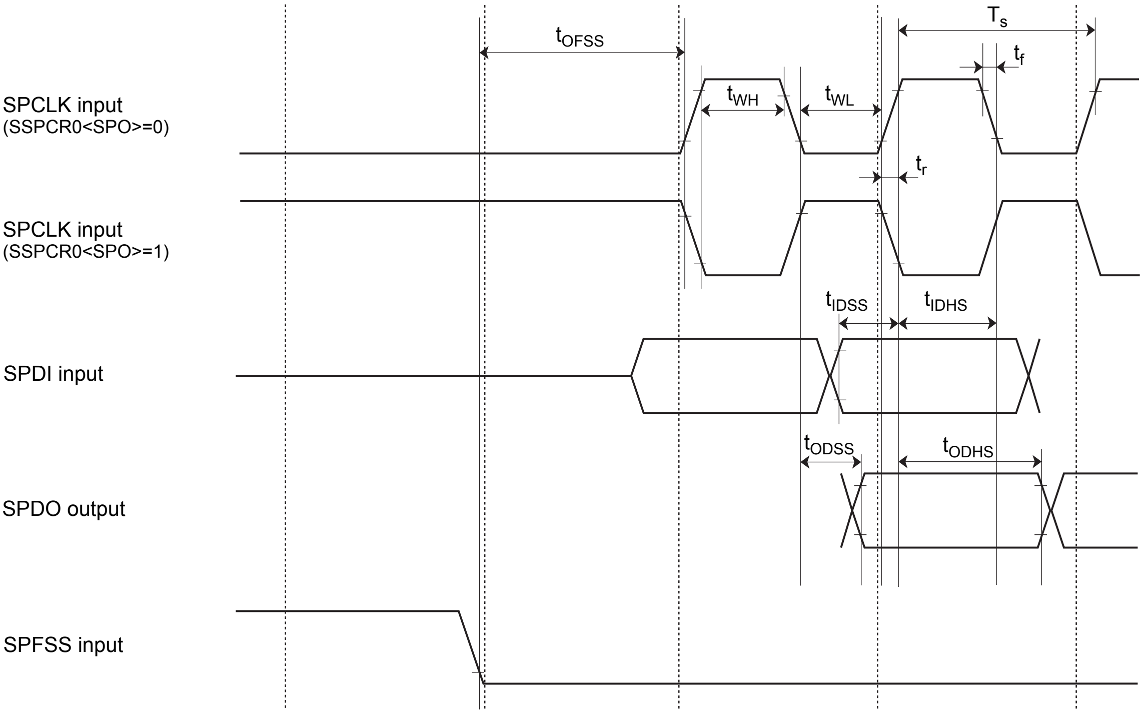
(2) Master SSPCR0<SPH>="1" (Data is latched on the second edge)



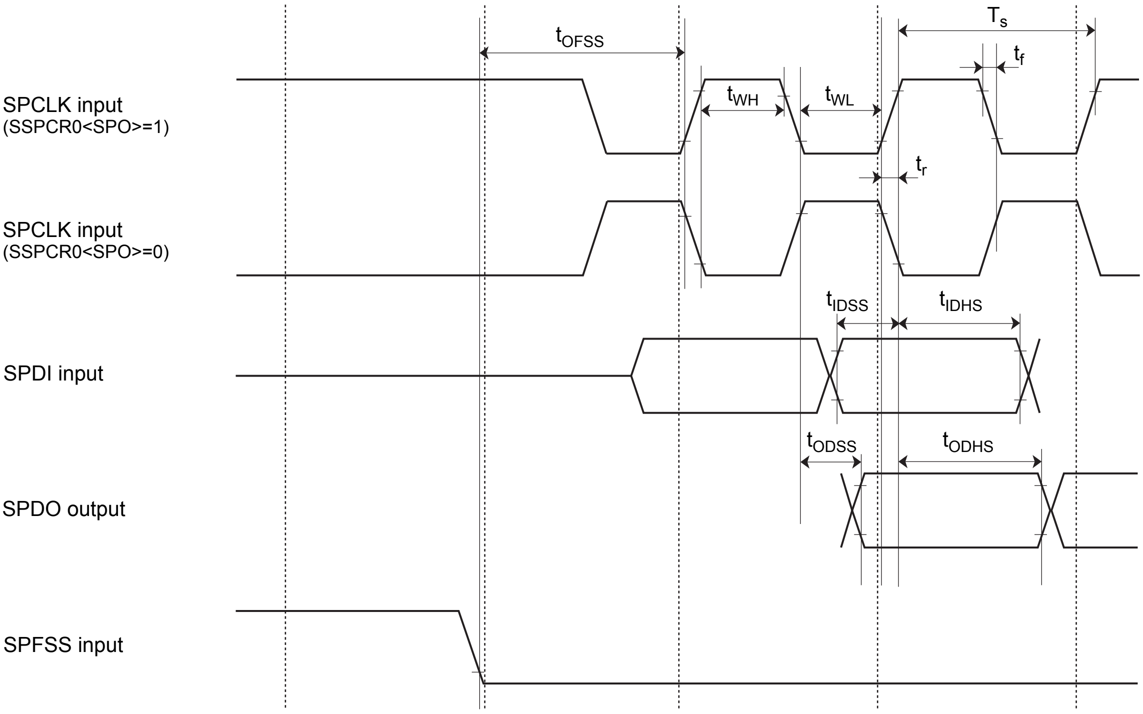
25.6.4.3 SSP SPI mode (Slave)

- fsys ≥ 12 × SPCLK (Maximum)
- fsys ≥ 65024 × SPCLK (Minimum)

(3) Slave SSPCR0<SPH>="0" (Data is latched on the first edge.)



(4) Slave SSPCR0<SPH>="1" (Data is latched on the second edge.)



25.6.5 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-level pulse width	t_{VCKL}	2x + 100	−	150	−	ns
Clock high-level pulse width	t_{VCKH}	2x + 100	−	150	−	ns

25.6.6 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-level pulse width	t_{CPL}	2x + 100	−	150	−	ns
High-level pulse width	t_{CPH}	2x + 100	−	150	−	ns

25.6.7 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 F low-level pulse width	t_{INTAL}	x + 100	−	125	−	ns
INT0 to F high level pulse width	t_{INTAH}	x + 100	−	125	−	ns

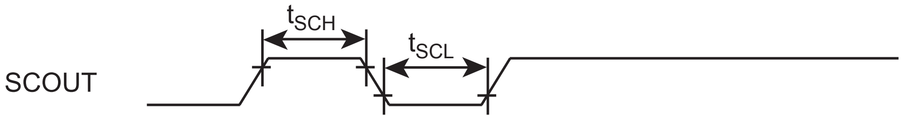
2. STOP release interrupts

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
INT0 to F low-level pulse width	t_{INTBL}	100	−	100	−	ns
INT0 to F high level pulse width	t_{INTBH}	100	−	100	−	ns

25.6.8 SCOUT pin AC Characteristics

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
High-level pulse width	t_{SCH}	0.5T − 5	−	7.5	−	ns
Low-level 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.



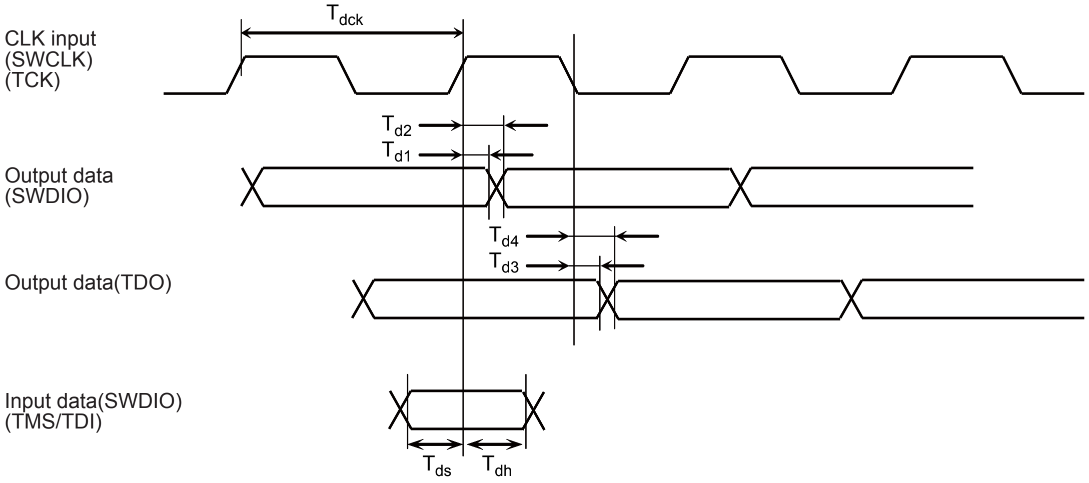
25.6.9 Debug Communication

25.6.9.1 SWD Interface

Parameter	Symbol	Min	Max	Unit
CLK cycle	T_{dck}	100	–	ns
CLK rise → Output data hold	T_{d1}	4	–	
CLK rise → Output data valid	T_{d2}	–	37	
Input data valid ← CLK rise	T_{ds}	20	–	
CLK rise → Input data hold	T_{dh}	15	–	

25.6.9.2 JTAG Interface

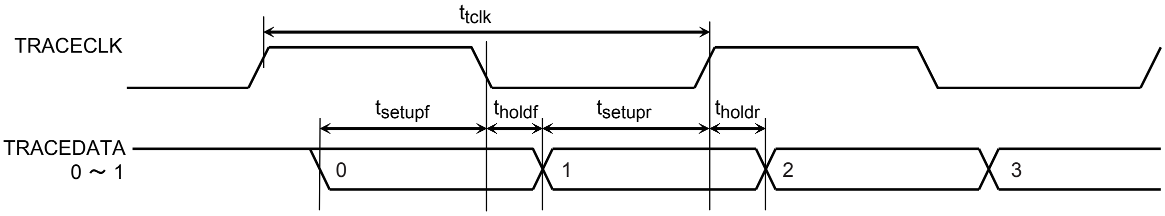
Parameter	Symbol	Min	Max	Unit
CLK cycle	T_{dck}	100	–	ns
CLK fall → Output data hold	T_{d3}	4	–	
CLK fall → Output data valid	T_{d4}	–	37	
Input data valid ← CLK rise	T_{ds}	20	–	
CLK rise → Input data hold	T_{dh}	15	–	



25.6.10 ETM Trace

- Output levels: High = 0.7 × DVDD5, Low = 0.3 × DVDD5
- Load capacitance: TRACECLK = 25pF, TRACEDATA = 20pF

Parameter	Symbol	Min	Max	Unit
TRACECLK cycle	t_{tclk}	50	−	ns
TRACEDATA valid ← TRACECLK rise	t_{setupr}	2	−	
TRACECLK rise → TRACEDATA hold	t_{holdr}	1	−	
TRACEDATA valid ← TRACECLK fall	t_{setupf}	2	−	
TRACECLK fall→ TRACEDATA hold	t_{holdf}	1	−	



25.6.11 On-chip Oscillator Characteristic

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Oscillation frequency	IHOSC	Ta = 25°C	−	9.0	−	MHz
Oscillation accuracy	−	Ta = -40 to 85°C	−15	−	+15	%

Note: Do not use an on-chip oscillator as a system clock (fsys) when high-accuracy oscillation frequency is required.

25.6.12 Flash Characteristic

Parameter	Condition	Min	Typ.	Max	Unit
Guaranteed number of Flash memory programming	Ta = 0 to 70°C	−	−	100	times

25.7 Recommended Oscillation Circuit

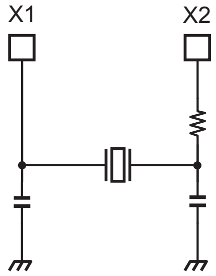


Figure 25-1 High-frequency oscillation connection

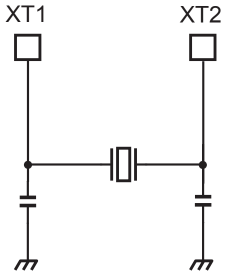


Figure 25-2 Low-frequency oscillation connection

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 substrate patterns, 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

25.7.1 Ceramic Oscillator

This product recommends the ceramic oscillator by Murata Manufacturing Co., Ltd.

Please refer to the following URL for details.

http://www.murata.co.jp

25.7.2 Crystal Oscillator

This product recommends the crystal oscillator by KYOCERA Crystal Device Corporation.

Please refer to the following URL for details.

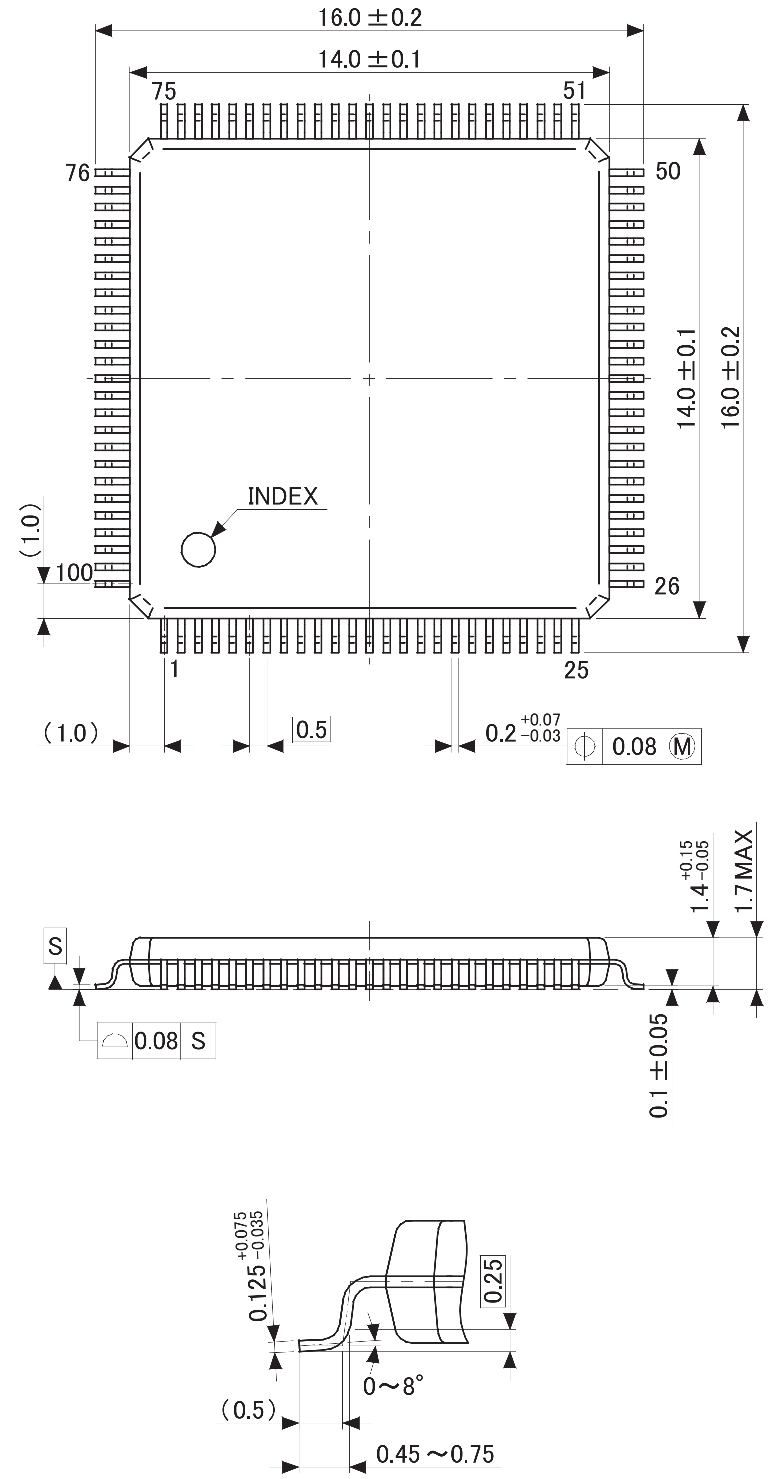
http://www.kyocera-crystal.co.jp

26. Package Dimensions

26.1 TMPM381FWFG

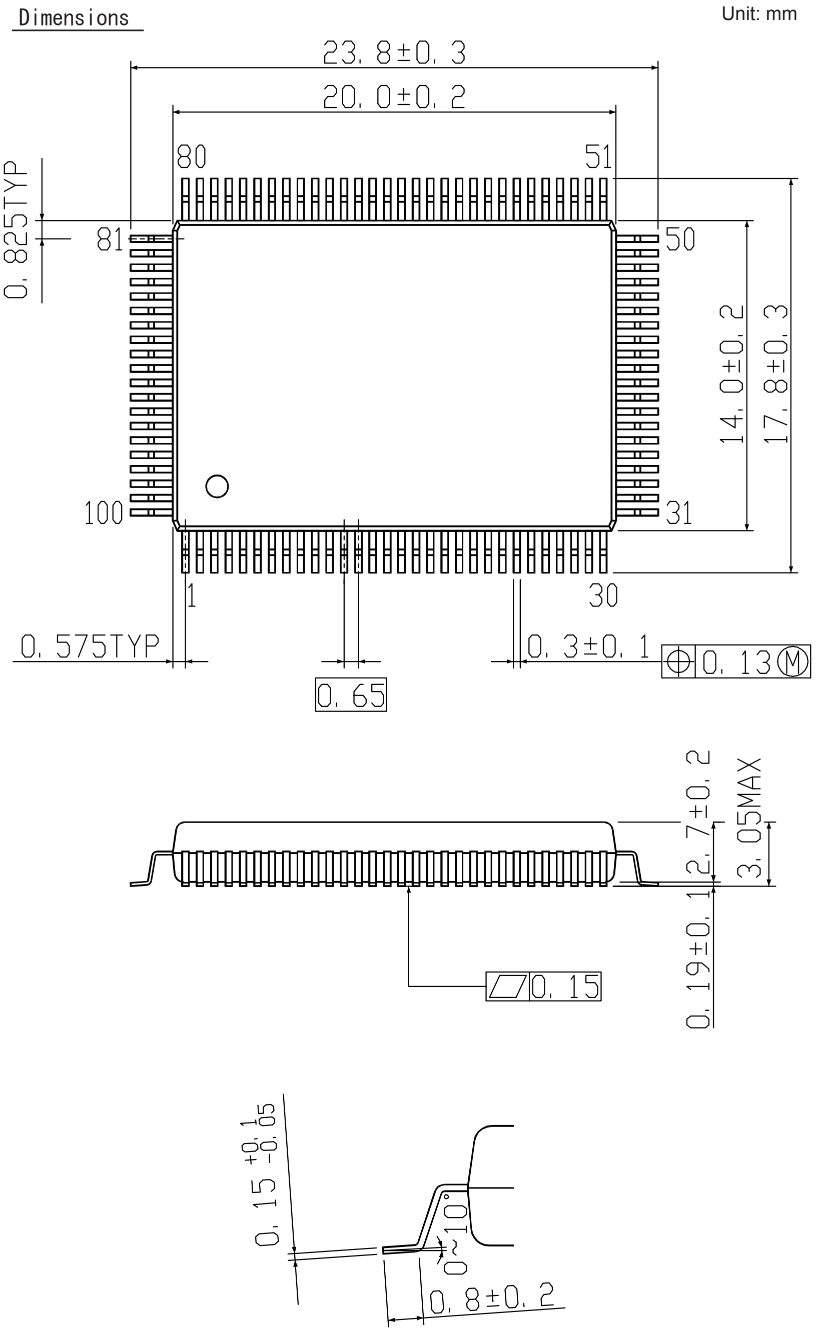
Type:LQFP100-P-1414-0.50H

Dimensions "Unit:mm"



26.2 TMPM381FWDFG

Type:QFP100-P-1420-0.65A

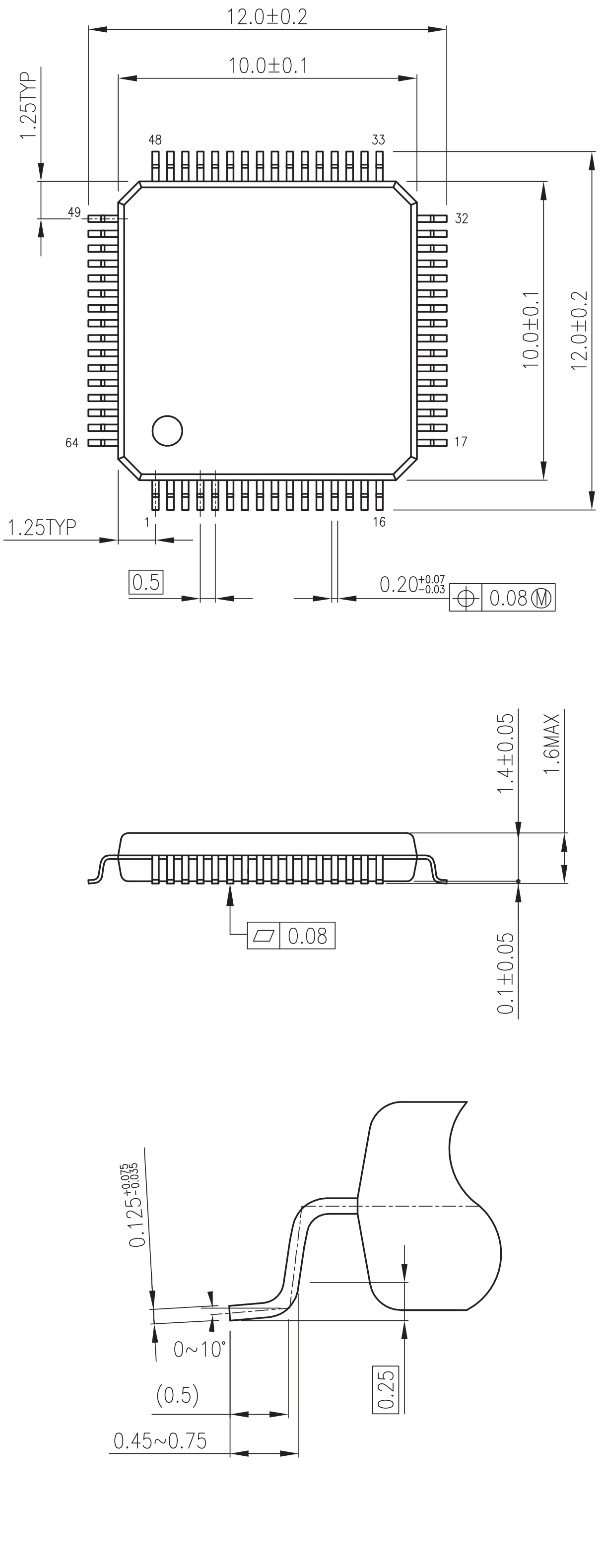


26.3 TMPM383FWUG/TMPM383FSUG

Type:LQFP64-P-1010-0.50E

Dimensions

"Unit:mm"

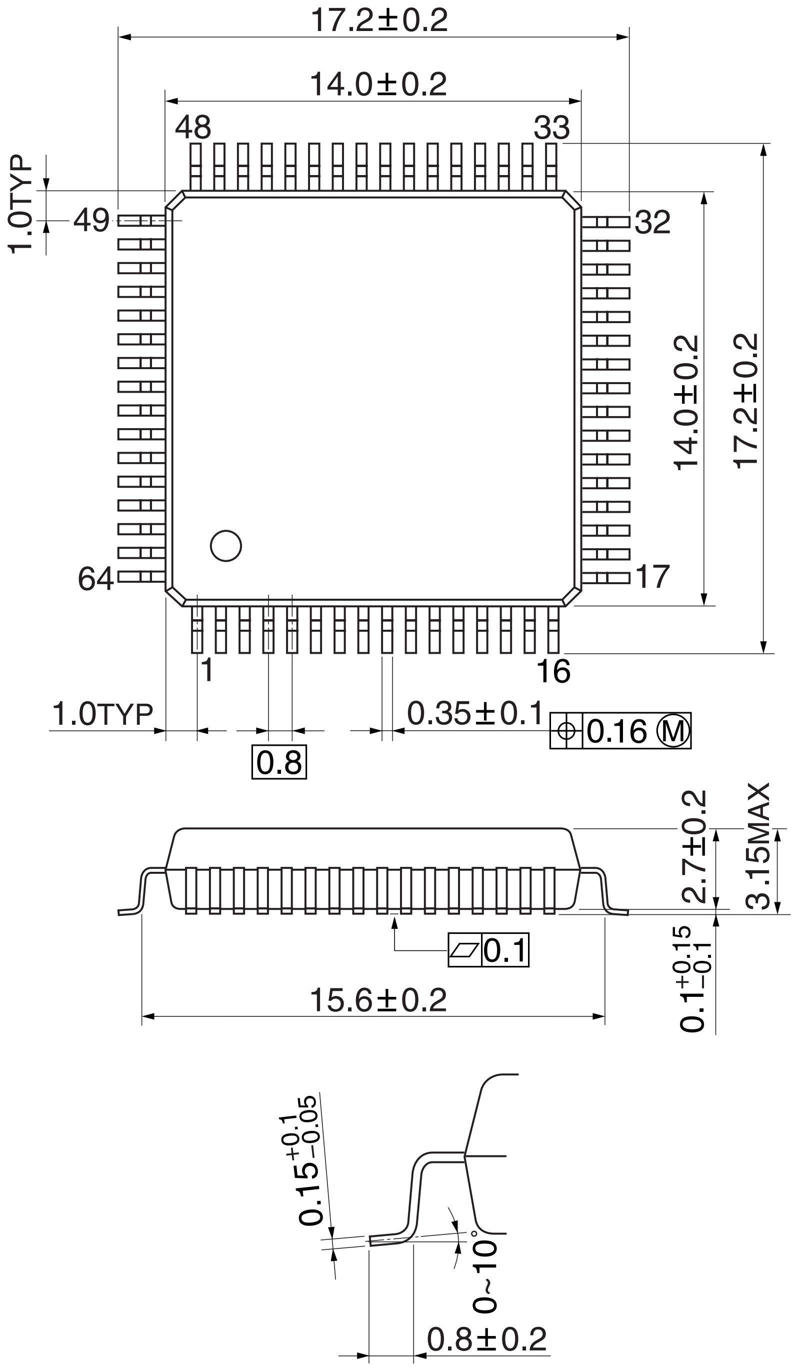


26.4 TMPM383FWEFG/TMPM383FSEFG

Type:QFP64-P-1414-0.80A

Dimensions

Unit: mm



RESTRICTIONS ON PRODUCT USE

- Do not design your products or systems based on the information on this document. Please contact your Toshiba sales representative for updated information before designing your products.
- 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 NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS 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 AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, 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. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative.
- 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 applicable export laws and regulations including, without limitation, 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.**