TOSHIBA

32 Bit RISC Microcontroller

TX03 Series

TMPM367FDFG

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

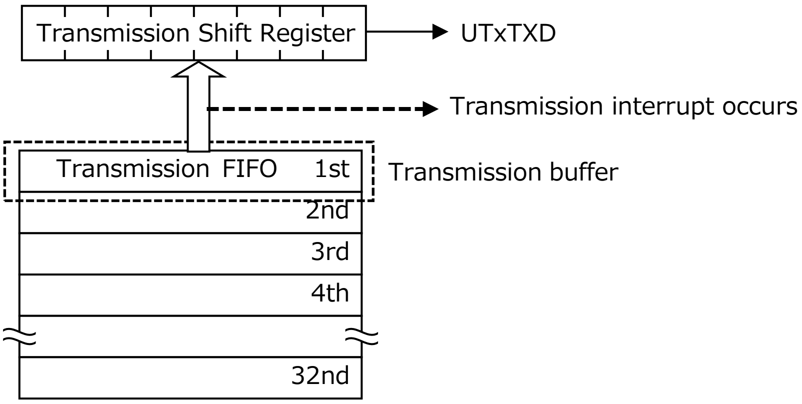
2. Details

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").

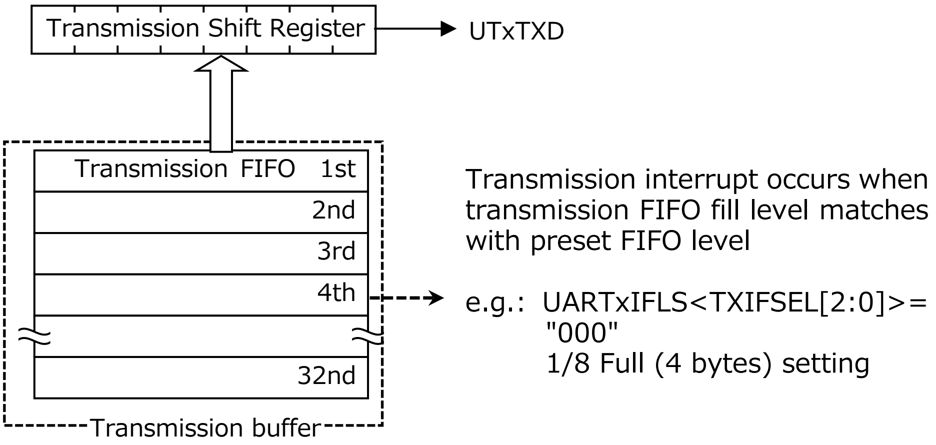
When the transmission is terminated, the final transmission data is transferred to the shift register, and the final transmission interrupt occurs when the transmission buffer becomes empty. If the next data is not written to the transmission buffer, the transmission interrupt can be intentionally cleared 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

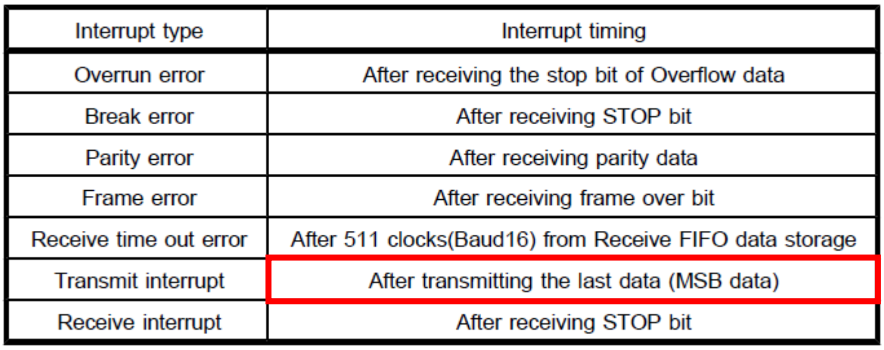
3.1.1. Applicable products and chapter of the description

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

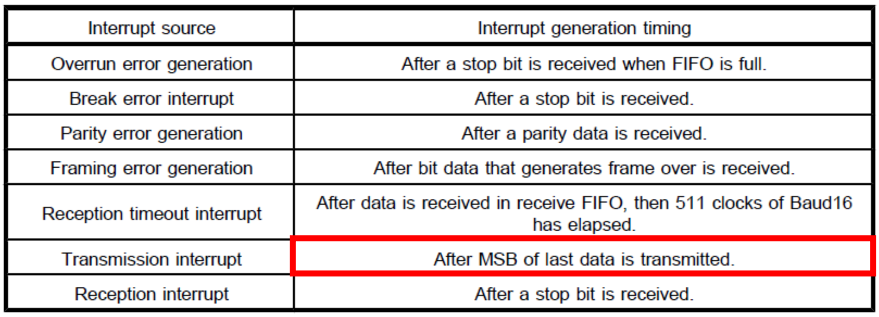
3.2. Description Type B(1)

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)

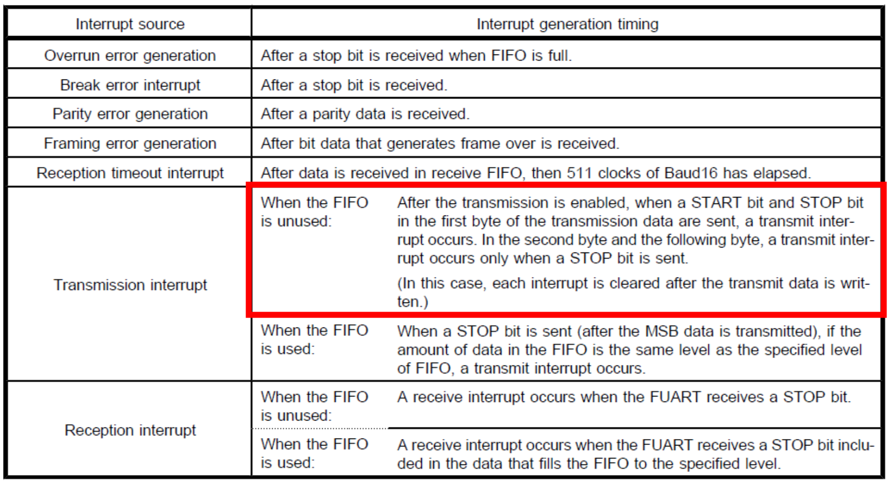
3.3.1. Applicable products and chapter of the description

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)

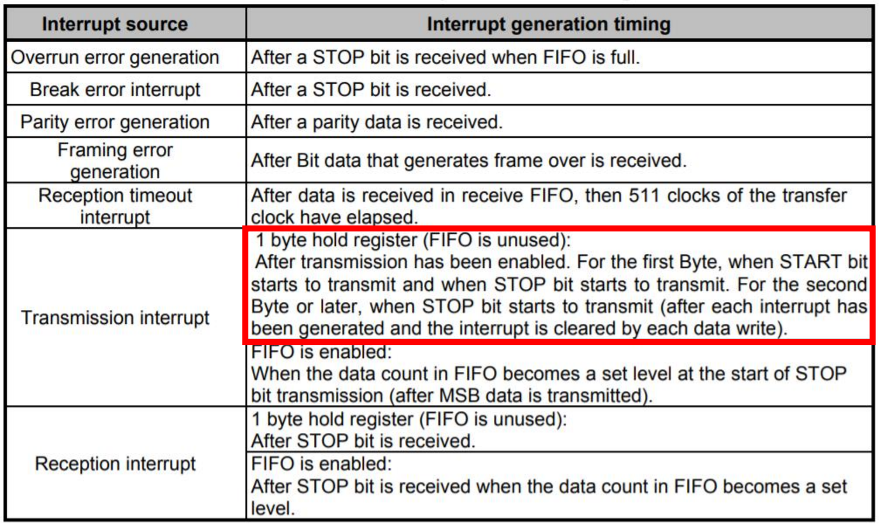
3.4.1. Applicable products and chapter of the description

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.

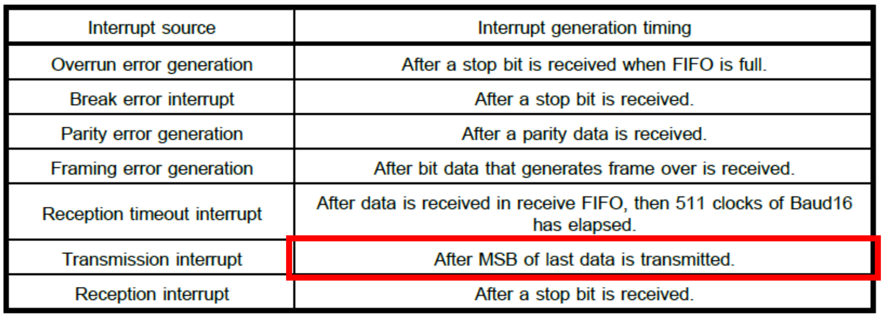
3.5. Description Type C

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

©TOSHIBA CORPORATION 2012-2023 All Rights Reserved

****
ARM, ARM Powered, AMBA, ADK, ARM9TDMI, TDMI, PrimeCell, RealView, Thumb, Cortex, Coresight, ARM9, ARM926EJ-S, Embedded Trace Macrocell, ETM, AHB, APB, and KEIL are registered trademarks or trademarks of ARM Limited in the EU and other countries.
****

Important Notices

Make sure to read read this chapter before using the product.

1 Serial bus interface

There are restrictions on the use of I2C bus mode when the multi-master function is used.

1.1 Description

When the multi-master function is used in I2C bus mode, if these masters start the communications simultaneously, the following phenomena may occur:

1. Communications may be locked up.
2. SCL pulse widths shorten; therefore these pulses may not satisfy I2C Specifications.

1.2 Condition

These phenomena occur only when the multi-master function is used in I2C bus mode. If a single master is used, these phenomena do not occur.

1.3 Workaround

There is no workaround for these phenomena. Perform recovery process by software.

1.4 How to Recover from These Phenomena

Perform recovery process by software.

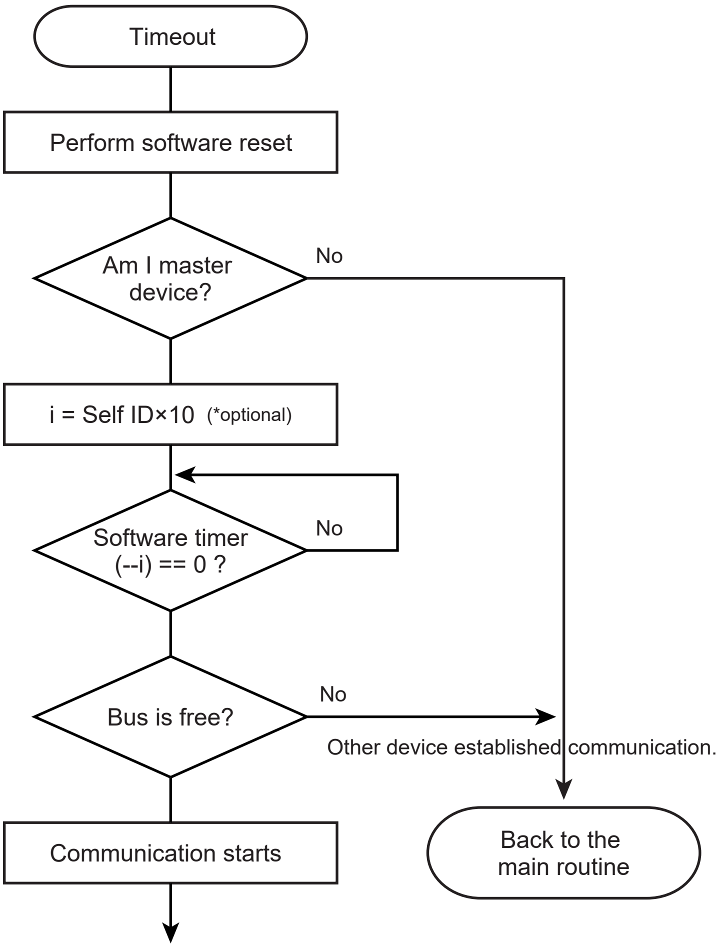
By using a timer, add timeout process to check whether communication is in a lock-up state.

An example of recovery process:

1. Start a timer count synchronously with start of the transmission.
2. If a serial interface interrupt (INTSBIx) does not occur in a certain period, the MCU determines the timeout.
3. If the MCU determines the timeout, communications may be locked up. Perform software reset on the serial bus interface circuit. This circuit is initialized to release communication from the lock up state.
4. Resend transmission data.

Mostly, Process 1 to 4 are enough to recovery; however if the multiple products are connected to the same bus line, add a delay time between each product's recovery process before Process 4 (resending data) is performed. This delay makes a time difference between each master; therefore bus collision can be avoided when the data is sent again.

Example: Recovery process after a timeout is detected.


Timeout
Perform software reset
Am I master device?
No
i = Self ID×10 (*optional)
Software timer (--i) == 0 ?
No
Bus is free?
No
Other device established communication.
Communication starts
Back to the main routine

2 Transitions to Low-power Consumption Mode and Generating a Non-maskable Interrupt

This chapter describes the precautions at which non-maskable interrupt (NMI) occurs when the MCU enters low-power consumption mode.

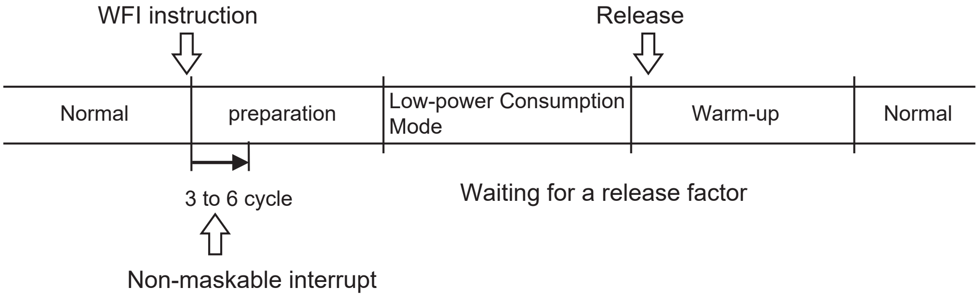
- STOP1
- STOP2

2.1 Description

When the WFI instruction is executed to enter the above-mentioned low-power consumption mode, if a non-maskable interrupt (NMI) occurs, the MCU may enter low-power consumption mode without the process for releasing low-power consumption mode.

Note 1: An NMI notice and flag setting to the CPU are normal; therefore the NMI process after low-power consumption mode is released can be performed.

Note 2: When the MCU has entered low-power consumption mode, factors other than an NMI are accepted; however an NMI may not be accepted.



2.2 Conditions

- The WFI instruction is executed to enter low-power consumption mode.
- A non-maskable interrupt occurs after WFI instruction execution and within 3 to 6 cycles.

2.3 Workaround

Do not use a non-maskable interrupt as a release factor for the above-mentioned low-power consumption modes.

To avoid generating a non-maskable interrupt, the following settings should be specified before the MCU enters the above-mentioned low-power consumption mode.

- NMI pin: This input pin must be fixed to "High".
- Watchdog timer: Stop the watchdog timer or set the reset function.
- Voltage detection circuit: Stop the voltage detection circuit or set the reset function.

3 Restrictions on the Use of the DMA function

There are restrictions on the use of the DMA function.

3.1 Description

When the synchronous serial interface (SSP) or the asynchronous serial communication circuit (UART) is used to transmit data or receive data, the following problems may occur:

- At transmission : A part of communication data may be lost (the FIFO buffer may overflow).
- At reception : Unnecessary data may be transferred (the FIFO buffer may underflow).

3.2 Conditions

The table below lists the peripheral functions that are connectable at DMA transfer. When the SSP <ch4 to 9> or UART <ch10 to 13> are used, if single-transfer is enabled (the connection channel of DMAxChnlUseburstSet is set to "0"), the problem may occur.

ch	Peripheral circuit	TMPM367FDFG TMPM367FDXBG TMPM368FDFG TMPM368FDXBG TMPM369FDFG TMPM369FDXBG	TMPM36BFYFG TMPM36BF10FG	ch	Peripheral circuit	TMPM367FDFG TMPM367FDXBG TMPM368FDFG TMPM368FDXBG TMPM369FDFG TMPM369FDXBG	TMPM36BFYFG TMPM36BF10FG
0	ADC conversion completion	–	o	16	SIO/UART1 reception	o	o
	ADC A conversion completion	o	–	17	SIO/UART1 transmission	o	o
1	ADC B conversion completion	o	–	18	SIO/UART2 reception	o	o
2	DAC0 conversion trigger	o	–	19	SIO/UART2 transmission	o	o
3	DAC1 conversion trigger	o	–	20	SIO/UART3 reception	o	o
4	SSP0 reception	o	o	21	SIO/UART3 transmission	o	o
5	SSP0 transmission	o	o	22	I2C/SIO0 transmission/reception	o	o
6	SSP1 reception	o	o	23	I2C/SIO1 transmission/reception	o	o
7	SSP1 transmission	o	o	24	I2C/SIO2 transmission/reception	o	o
8	SSP2 reception	o	o	25	TMRB0 compare match	o	o
9	SSP2 transmission	o	o	26	TMRB1 compare match	o	o
10	UART4 reception	o	o	27	TMRB2 compare match	o	o
11	UART4 transmission	o	o	28	TMRB3 compare match	o	o
12	UART5 reception	o	o	29	TMRB4 compare match	o	o
13	UART5 transmission	o	o	30	DMA request pin	o	o
14	SIO/UART0 reception	o	o	31	Software trigger	o	–
15	SIO/UART0 transmission	o	o				

o: Connectable peripheral circuit –: Not supported

3.3 Explanation

- At transmission

  Under the following circumstance, one additional DMA request may occur compared with the number of unused data space of the FIFO of the SSP or UART. Therefore, DMA transfer is executed even if the FIFO is full and the FIFO overflow occurs.

  - The number of data of the FIFO becomes lower than the watermark level while the DMA is transferring data to the FIFO.

- At reception

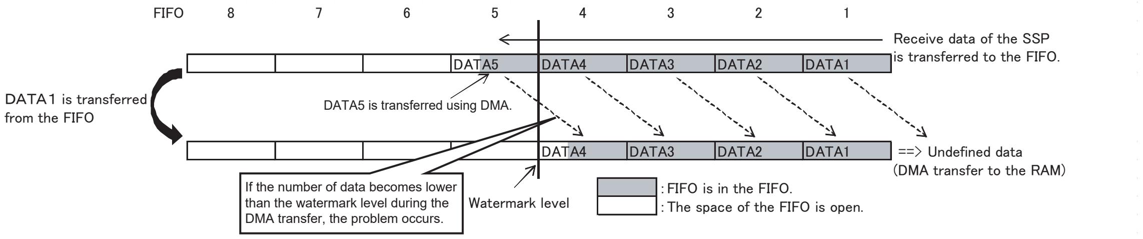
  Under the following circumstance, one additional DMA request may occur compared with the number of stored data. Therefore, invalid data is transferred from the FIFO of the SSP or UART and the FIFO underflow occurs.

  - The number of data of the FIFO becomes higher than the watermark level while the DMA is tranferring data from the FIFO of the SSP or UART.

Note: Wartermark level:

SSP	Fixed to 4
UART	Full level of UARTxIFLS<RXIFSEL[2:0]> or <TXIFSEL[2:0]>

Example of relationship between the condition of the FIFO and the watermark level (SSP transmission)


FIFO
8
7
6
5
4
3
2
1
Receive data of the SSP is transferred to the FIFO.
DATA5
DATA4
DATA3
DATA2
DATA1
DATA1 is transferred from the FIFO
DATA5 is transferred using DMA.
DATA4
DATA3
DATA2
DATA1
==> Undefined data (DMA transfer to the RAM)
If the number of data becomes lower than the watermark level during the DMA transfer, the problem occurs.
Watermark level
: FIFO is in the FIFO.
: The space of the FIFO is open.

3.4 Workaround

When the DMA function is used, set "1" to the corresponding channel of DMAxChnlUseburstSet (single-transfer is prohibited).

However, there are several conditions and restrictions.

Note: These conditions and restrictions vary depending on transmission or reception.

3.4.1 Transmission

When single-transfer is prohibited, arbitration setting is subject to a constraint. According to the number of transfers, select the appropriate method below:

a. When the number of transfers is a multiple of the watermark level of the FIFO.

Set the arbitration rates to the watermark level of the FIFO. After the number of specified transfers is complete, arbitration for the priorities between the peripheral functions that are connected to the DMA controller unit is issued. Therefore, the DMA transfer can be performed at high-speed.

Set the number of arbitration rates <R_power> for the control data to the watermark level of the FIFO.

b. When the number of transfers is not a multiple of the watermark level of the FIFO.

Set the arbitration rates to "after one transfer". This setting can be used in every case. After each transfer is complete, arbitration for the priorities between the peripheral functions that are connected to the DMA controller unit is issued. Therefore, the DMA transfer is performed at slower speed than the case of (a).

Specify "0000" as the arbitration rate setting <R_power> for the control data.

3.4.2 Reception

Disable or Enable single-transfer according to the number of transfers of control data <n_minus_1>.

a. When the number of transfers is a multiple of the watermark level.

This setting can be used when the number of transfers is a multiple of the watermark level. For example, the watermark level is n, "n Å~integer number " can be set as the number of transfers.

Set "1" (disable the single-transfer) to the corresponding channel of DMAxChnlUseburstSet.

Set the number of arbitration rates <R_power> for the control data to the watermark level of the FIFO.

b. When the number of transfers is less than the watermark level.

This setting can be used when the number of transfers is less than the watermark level.

Set "0" (enable the single-transfer) to the corresponding channel of DMAxChnlUseburstSet.

Specify "0000" as the arbitration rate <R_power> for the control data.

c. When the number of transfers is other than the above.

This setting can be used when the number of transfers is set to over the watermark level and the setting is not a multiple of the watermark level.

Use "Peripheral scatter-gather" as the transfer mode and combine the two tasks at DMA transfer.

For example, in the case of the number of transfers = (n × watermark) + m

Set Task A to the same as (a).

Disable single-transfer. Set the number of arbitration rates <R_power> for the control data to the watermark level of the FIFO. Set "watermark level × n" as the number of transfers.

Set Task B to the same as (b).

Enable single-transfer. Set "0000" to <R_power>. Set "m" as the number of transfers.

Example of setting

The peripheral circuit that performs DMA transfer.		SSO(reception)	
The number of DMA transfers to be set		15 times	
Watermark level		4 (Fixed to 4 for SSP communication)	
The DMA register setting		DMAxChnlUseburstSet<ch4>=1 : Disable single-transfer	
Channel control data setting (Alternative data)	Task A Use the same setting as (a).	<n_minus_1>=0x00B	The number of transfers 4 × 3 = 12 times
		<R_power>=0011	Arbitration is issued after four transfers are complete.
		<next_useburst>=0	Enables Task B to perform single-transfer
		<cycle_ctrl>=111	Peripheral scatter-gather mode
	Task B Use the same setting as (b).	<n_minus_1>=0x002	The number of transfers 15 - 12 = 3 times
		<R_power>=0000	Arbitration is issued after one transfer is complete.
		<cycle_ctrl>=001	The transfer ends in basic mode.

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

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

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

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

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

Base Address = 0x0000_0000

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

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

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

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

1.2.2 SAMCR(Control register)

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

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

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

R / W	READ WRITE
R	READ
W	WRITE

c. Data descriptopn

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

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

d. Register descriptoption

Registers are described as shown below.

- Register name <Bit Symbol>

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

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

- Register name [Bit]

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

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

Revision History

Date	Revision	Comment
2012/11/07	1	First Release
2013/10/11	2	Contents Revised
2022/05/25	3	Contents Revised
2022/09/30	4	Contents Revised
2023/07/21	5	Contents Revised

CMOS 32-Bit Microcontroller

TMPM367FDFG

The TMPM367FDFG is a 32-bit RISC microprocessor with an ARM Cortex™-M3 microprocessor core.

Product Name	ROM (FLASH)	RAM	Package
TMPM367FDFG	512 Kbyte	128 Kbyte	LQFP100

The outlines and features are as follows:

1.1 Features

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

   b. Both high performance and low power consumption have been achieved.
      - High performance
      - A 32-bit multiplication (32×32=32 bit) can be executed with one clock.
      - Division takes between 2 and 12 cycles depending on dividend and devisor
      - Low power consumption
      - Optimized design using a low power consumption library
      - Standby function that stops the operation of the micro controller core

   c. High-speed interrupt response suitable for real-time control
      - An interruptible long instruction.
      - Stack push automatically handled by hardware.

2. High-speed programming & low power consumption using Toshiba NANO FLASH™ technology
   - High-speed programming has an effect on mass production stage and developing device
   - Low power consumption design

3. On-chip program memory and data memory
   - On-chip FlashROM : 512K bytes
   - On-chip RAM : 128K bytes

4. μDMA controller (μDMAC) : 32 channels / 2 units

   Transfer object : On-chip memory, on-chip I/O and external memory

5. 16-bit timer (TMRB) : 8 channels
   - 16-bit interval timer mode

- 16-bit event counter mode
- 16-bit PPG output (4-phase synchronous output supported)
- Input capture function

6. Real-time clock (RTC) : 1 channel
    - Clock (hour, minute and second)
    - Calender (Month, week, date and leap year)
    - Operable regardless of operational modes (NORMAL/IDLE/STOP1/STOP2)

7. Watch-dog timer (WDT) : 1 channel

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

8. General-purpose serial interface (SIO/UART) : 4 channels

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

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

    Either I2C bus mode or synchronous mode can be selected.

10. Synchronous serial interface (SSP) : 3 channels

    Supports SPI/SSI/Microwire formats

    Channel 0/1 : 10MHz (Master), 3.3MHz (Slave) (@ fsys=80MHz)

    Channel 2 : 20MHz (Master), 6.6MHz (Slave) (@ fsys=80MHz)

11. UART : 2 channels

    Either 8-wired UART or IrDA 1.0 mode can be selected.

12. 12-bit AD converter (ADC) : 8 channels/ 2 units
    - Start by an internal timer trigger
    - Fixed channel/scan mode
    - Single/repeat mode
    - AD monitoring 2ch
    - Conversion speed 1.0 µs (@fsys = 80 MHz in normal mode)
    - High-speed conversion using interleave mode (conversion speed max. 0.5µs)

13. 10-bit DA converter (DAC) : 2 channels
    - VREFH cutting function (Power down mode)
    - Output current : 1mA
    - Settling time : 1µs
    - Signal generation function

14. USB2.0 full-speed device : 1 channel
    - Conforms to Universal Serial Bus Specification Rev2.0
    - End point : 8 channels
    - Control/Bulk/Interrupt/Isochronous mode
    - Full-speed 12Mbps (Low speed is not available.)

15. Remote control signal preprocessor (RMC) : 1 channel
    - Can receive up to 72 bits data at a time
    - Noise canceller
    - Reader code detection

16. Multi-purpose timer (MPT) : 4 channels
    - Motor control (PMD : 1 channel)
    - IGBT control
    - 16-bit timer

17. Encoder input function (ENC) : 1 channel

    Support incremental type encoder

18. LVD/POR function : 1 unit

19. Oscillation Frequency Detection (OFD) : 1 unit

20. External bus interface (EBIF) : 1 unit
    - Supports multiplex bus : 8-bit/16-bit width
    - Chip select/wait controller : 4 channels

21. Interrupt source
    - Internal: 95 factors. The order of precedence can be set over 7 levels (except the watchdog timer interrupt).
    - External: 14 factors. The order of precedence can be set over 7 levels.

22. Non maskable interrupt (NMI)

    NMI is generated by Watchdog timer, LVD and $\overline{\text{NMI}}$ pin.

23. Input/ output ports

    Input/output 59 pins (One 5V tolerant input included), output: 1 pin

24. Low power cunsumption mode

    IDLE, STOP1, STOP2

    IDLE : CPU stops

    STOP1/STOP2 : All circuits stop except RTC and remote control signal preprocessor

    (In STOP2 mode, partial circuit is shut-down.)

25. Clock generator
    - On-chip PLL (switchable multiplier among 3-, 4-, 5-, 6-, 8- and 10-fold)
    - Clock gear function : The high-speed clock can be divided into 1/1, 1/2, 1/4, 1/8 or 1/16.

26. Endian

    Little endian

27. Debug interface

JTAG/SWD/SWV/TRACE (DATA 4 bits)

28. Maximum operating frequency

    80MHz (External oscillator@8MHz/10MHz/16MHz or internal oscillator@10MHz)

29. Operating voltage range

    2.7V to 3.6V (When USB is not used)

    3.0V to 3.45V (When USB is used)

30. Temperature range
    - −40 to 85 degrees (except during Flash writing/erasing)
    - 0 to 70 degrees (during Flash writing/erasing)

31. Package

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

1.2 Block Diagram

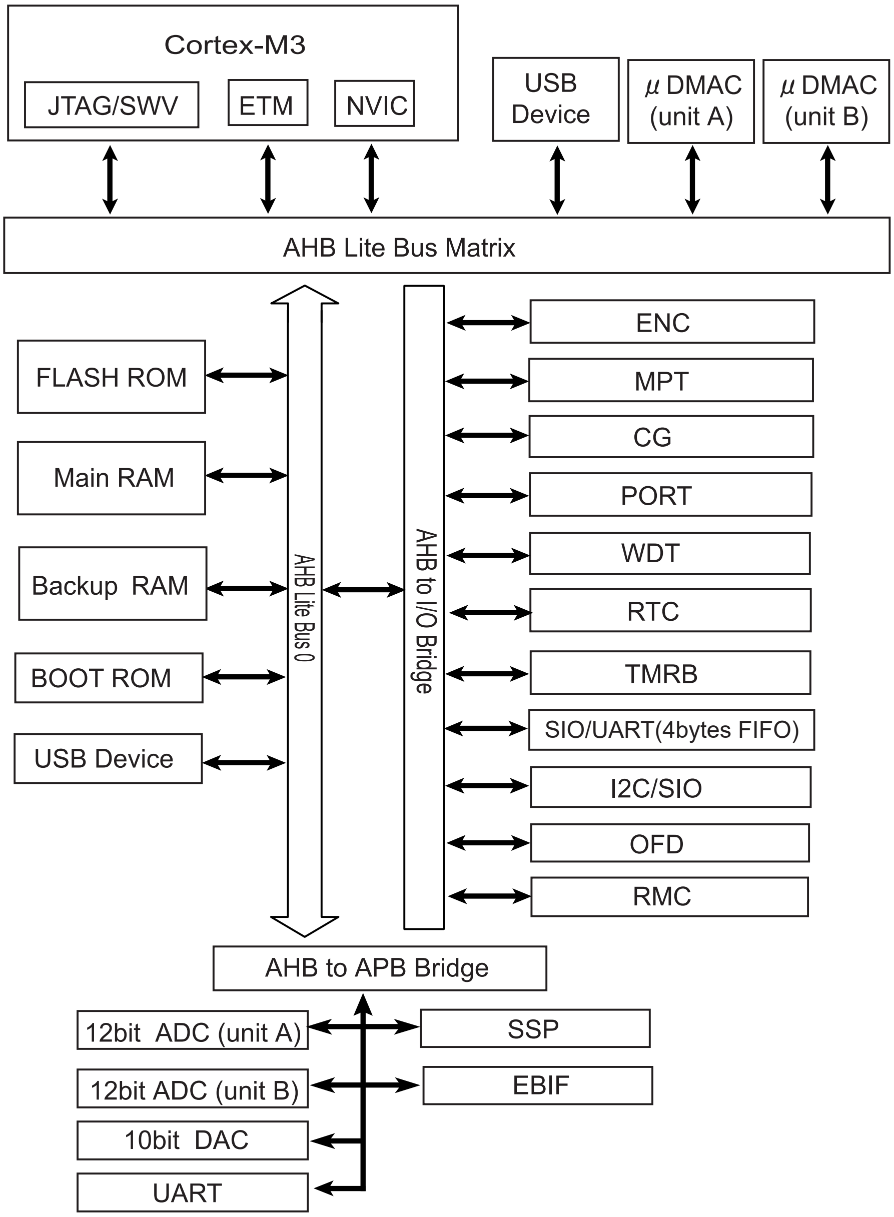

Cortex-M3
JTAG/SWV
ETM
NVIC
USB Device
μ DMAC (unit A)
μ DMAC (unit B)
AHB Lite Bus Matrix
FLASH ROM
Main RAM
Backup RAM
BOOT ROM
USB Device
AHB Lite Bus 0
AHB to I/O Bridge
ENC
MPT
CG
PORT
WDT
RTC
TMRB
SIO/UART(4bytes FIFO)
I2C/SIO
OFD
RMC
AHB to APB Bridge
12bit ADC (unit A)
12bit ADC (unit B)
10bit DAC
UART
SSP
EBIF

Figure 1-1 TMPM367FDFG Block Diagram

1.3 Pin Layout

Figure 1-2 shows the pin layout of TMPM367FDFG.

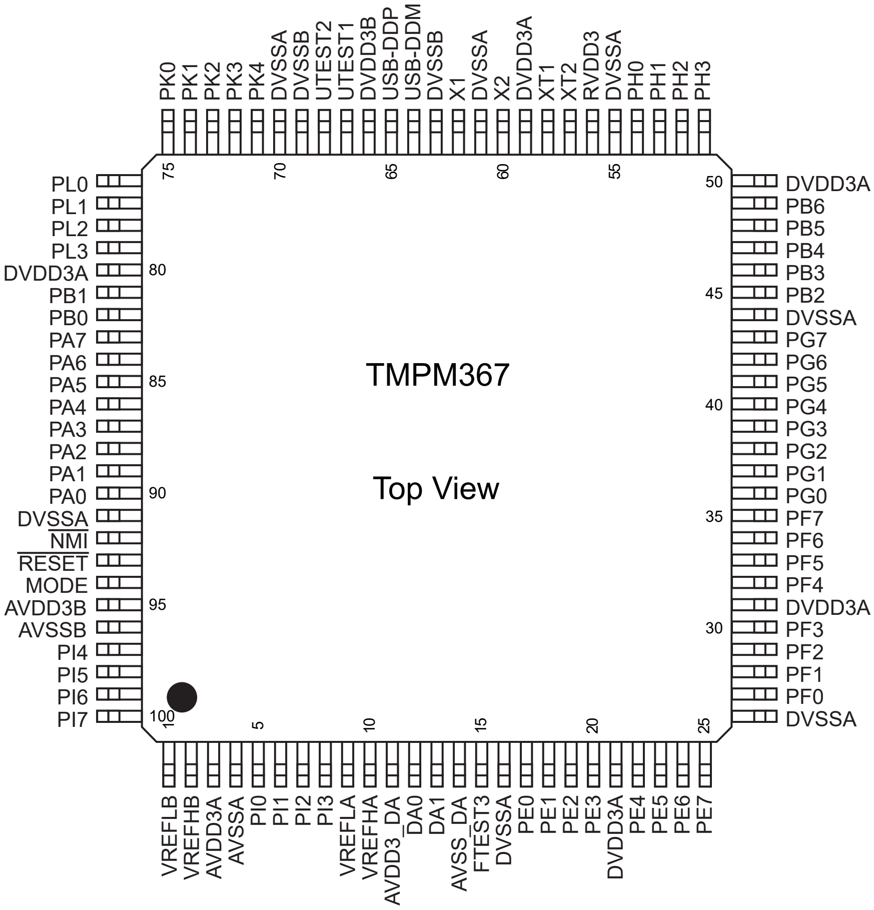

PK0
PK1
PK2
PK3
PK4
DVSSA
DVSSB
UTEST2
UTEST1
DVDD3B
USB-DDP
USB-DDM
DVSSB
X1
DVSSA
X2
DVDD3A
XT1
XT2
RVDD3
DVSSA
PH0
PH1
PH2
PH3
75
70
65
60
55
50
PL0
PL1
PL2
PL3
DVDD3A
PB1
PB0
PA7
PA6
PA5
PA4
PA3
PA2
PA1
PA0
DVSSA
NMI
RESET
MODE
AVDD3B
AVSSB
PI4
PI5
PI6
PI7
80
85
90
95
100
TMPM367
Top View
DVDD3A
PB6
PB5
PB4
PB3
PB2
DVSSA
PG7
PG6
PG5
PG4
PG3
PG2
PG1
PG0
PF7
PF6
PF5
PF4
DVDD3A
PF3
PF2
PF1
PF0
DVSSA
45
40
35
30
25
1
5
10
15
20
VREFLB
VREFHB
AVDD3A
AVSSA
PI0
PI1
PI2
PI3
VREFLA
VREFHA
AVDD3_DA
DA0
DA1
AVSS_DA
FTEST3
DVSSA
PE0
PE1
PE2
PE3
DVDD3A
PE4
PE5
PE6
PE7

Figure 1-2 Pin Layout

1.4 Pin names and Functions

The following table shows the input/output pin names and functions of TMPM367FDFG.

1.4.1 Sorted by Pin Number

Table 1-1 Pin Names and Functions (1/8)

Type	Pin No.	Pin Name	Input/ Output	Function
PS	1	VREFLB	-	Supplying the AD converter with a reference power supply. (note) VREFLB must be connected to power supply even if AD converter is not used.
PS	2	VREFHB	-	Supplying the AD converter with a reference power supply. (note) VREFHB must be connected to power supply even if AD converter is not used.
PS	3	AVDD3A	-	Supplying the AD converter with a power supply. (note) AVDD3A must be connected to power supply even if AD converter is not used.
PS	4	AVSSA	-	AD converter GND pin (note) AVSSA must be connected to GND even if the AD converter is not used.
Function	5	PI0 AINA0 INT9	Input/Output Input Input	Input/Output port Analog input External interrupt pin
Function	6	PI1 AINA1 INTA	Input/Output Input Input	Input/Output port Analog input External interrupt pin
Function	7	PI2 AINA2 INTB	Input/Output Input Input	Input/Output port Analog input External interrupt pin
Function	8	PI3 AINA3 INTC $\overline{\text{DMAREQ}}$	Input/Output Input Input Input	Input/Output port Analog input External interrupt pin DMA request pin
PS	9	VREFLA	-	Supplying the AD converter with a reference power supply. (note) VREFLA must be connected to power supply even if AD converter is not used.
PS	10	VREFHA	-	Supplying the AD converter with a reference power supply. (note) VREFHB must be connected to power supply even if AD converter is not used.
PS	11	AVDD3_DA	-	Supplying the DA converter with a power supply. (note) AVDD3_DA must be connected to power supply even if DA converter is not used.
Function	12	DA0	Output	DA converter analog output pin
Function	13	DA1	Output	DA converter analog output pin
PS	14	AVSS_DA	-	Supplying the DA converter GND pin (note) AVSS_DA must be connected to GND even if DA converter is not used.
TEST	15	FTEST3	-	TEST pin (note) TEST pin must be left OPEN.
PS	16	DVSSA	-	GND pin
Function	17	PE0 A16 INT4 TB0IN	Input/Output Output Input Input	Input/Output port (This port is used for single boot mode. Refer to "Flash" for more detail.) Address bus External interrupt pin Inputting the timer B capture trigger

Table 1-1 Pin Names and Functions (2/8)

Type	Pin No.	Pin Name	Input/ Output	Function
Function	18	PE1 RXD0 A17 INT5 TB1IN	Input/Output Input Output Input Input	Input/Output port (This port is used for single boot mode. Refer to "Flash" for more detail.) SIO receive pin Address bus External interrupt pin Inputting the timer B capture trigger
Function	19	PE2 TXD0 A18 TB1OUT	Input/Output Output Output OUTput	Input/Output port (This port is used for single boot mode. Refer to "Flash" for more detail.) SIO transmit pin Address bus Timer B output
Function	20	PE3 SCLK0 A19 $\overline{\text{CTS0}}$ TB0OUT	Input/Output Input/Output Output Input Output	Input/Output port (This port is used for single boot mode. Refer to "Flash" for more detail.) SIO clock pin Address bus Handshake pin Timer B output pin
PS	21	DVDD3A	-	Power supply pin
Function	22	PE4 SCLK1 A20 $\overline{\text{CTS1}}$ TB2OUT	Input/Output Input/Output Output Input Output	Input/Output port SIO clock pin Address bus Handshake pin Timer B output pin
Function	23	PE5 TXD1 A21	Input/Output Output Output	Input/Output port (This port is used for single boot mode. Refer to "Flash" for more detail.) SIO transmit pin Address bus
Function	24	PE6 RXD1 A22	Input/Output Input Output	Input/Output port SIO receive pin Address bus
Function	25	PE7 A23 INT6 TB2IN	Input/Output Output Input Input	Input/Output port Address bus External interrupt pin Inputting the timer B capture trigger
PS	26	DVSSA	-	GND pin
Function	27	PF0 AD0 $\overline{\text{CTS4}}$	Input/Output Input/Output Input	Input/Output port Address data bus pin Handshake pin
Function	28	PF1 AD1 TXD4 IROUT4	Input/Output Input/Output Output Output	Input/Output port Address data bus pin UART transmit pin IrDA1.0 transmit pin
Function	29	PF2 AD2 RXD4 IRIN4	Input/Output Input/Output Input Input	Input/Output port Address data bus pin UART receive pin IrDA1.0 receive pin

Table 1-1 Pin Names and Functions (3/8)

Type	Pin No.	Pin Name	Input/ Output	Function
Function	30	PF3 AD3 $\overline{\text{RTS4}}$	Input/Output Input/Output Output	Input/Output port Address data bus pin UART Modem control ($\overline{\text{RTS}}$)
PS	31	DVDD3A	-	Power supply pin
Function	32	PF4 AD4 INT0 DCD4	Input/Output Input/Output Input Input	Input/Output port Address data bus pin External interrupt pin Modem status (DCD)
Function	33	PF5 AD5 ENCZ RIN4 SCK1	Input/Output Input/Output Input Input Input/Output	Input/Output port Address data bus pin Z-phase input pin Modem status (RIN) SIO mode clock pin
Function	34	PF6 AD6 ENCB DSR4 SI1/SCL1	Input/Output Input/Output Input Input Input/Output	Input/Output port Address data bus pin B-phase input pin Modem status (DSR) SIO mode receive pin, I2C mode clock pin
Function	35	PF7 AD7 ENCA DTR4 SO1/SDA1	Input/Output Input/Output Input Output Input/Output	Input/Output port Address data bus pin A-phase input pin Modem control (DTR) SIO mode transmit pin, I2C mode transmit/receive pin
Function	36	PG0 AD8 MT0IN	Input/Output Input/Output Input	Input/Output port Address data bus pin Multi-purpose timer (IGBT mode) input pin
Function	37	PG1 AD9 $\overline{\text{EMG}}$ $\overline{\text{GEMG0}}$	Input/Output Input/Output Input Input	Input/Output port Address data bus pin Multi-purpose timer (PMD mode abnormal status detection input Multi-purpose timer (IGBT mode abnormal status detection input
Function	38	PG2 AD10 ZO MTOUT10 MTTB0IN	Input/Output Input/Output Output Output Input	Input/Output port Address data bus Multi-purpose timer (PMD mode) Z-phase output pin Multi-purpose timer (IGBT mode) output pin Multi-purpose timer (timer mode) input pin
Function	39	PG3 AD11 WO MTOUT00 MTTB0OUT	Input/Output Input/Output Output Output Output	Input/Output port Address data bus Multi-purpose timer (PMD mode) W-phase output pin Multi-purpose timer (IGBT mode) output pin Multi-purpose timer (timer mode) output pin

Table 1-1 Pin Names and Functions (4/8)

Type	Pin No.	Pin Name	Input/ Output	Function
Function	40	PG4 AD12 YO SP1CLK	Input/Output Input/Output Output Input/Output	Input/Output port Address data bus Multi-purpose timer (PMD mode) Y-phase output pin SSP clock pin
Function	41	PG5 AD13 VO SP1DO	Input/Output Input/Output Output Output	Input/Output port Address data bus Multi-purpose timer (PMD mode) V-phase output pin SSP data output pin
Function	42	PG6 AD14 XO SP1DI	Input/Output Input/Output Output Input	Input/Output port Address data bus Multi-purpose timer (PMD mode) X-phase output pin SSP data input pin
Function	43	PG7 AD15 UO SP1FSS	Input/Output Input/Output Output Input/Output	Input/Output port Address data bus Multi-purpose timer (PMD mode) U-phase output pin SSP frame/slave selection pin
PS	44	DVSSA	-	GND pin
Function	45	PB2 $\overline{\text{WR}}$ SP2CLK MTOUT03 MTTB3OUT	Input/Output Output Input/Output Output Output	Input/Output port Write strobe pin SSP clock pin Multi-purpose timer (IGBT mode) output pin Multi-purpose timer (timer mode) output pin
Function	46	PB3 $\overline{\text{RD}}$ SP2DO MTOUT13 MTTB3IN	Input/Output Output Output Output Input	Input/Output port Read strobe pin SSP data output pin Multi-purpose timer (IGBT mode) output pin Multi-purpose timer (timer mode) input pin
Function	47	PB4 $\overline{\text{CS0}}$ SP2DI $\overline{\text{GEMG3}}$ INT7	Input/Output Output Input Input Input	Input/Output port Chip select pin Output SSP data input pin Multi-purpose timer read strobe pin (IGBT mode) abnormal status detection input External interrupt pin
Function	48	PB5 ALE SP2FSS MT3IN INT1	Input/Output Output Input/Output Input Input	Input/Output port Address latch enable pin SSP frame/slave selection pin Multi-purpose timer (IGBT mode) input pin External interrupt pin
Function/ Control	49	PB6 $\overline{\text{BELL}}$ SCOUT TB3OUT $\overline{\text{BOOT}}$	Output Output Output Output Input	Output port Byte enable pin Internal clock output pin Timer B output pin Boot mode pin (This port is used for single boot mode. Refer to "Flash" for more detail.)
PS	50	DVDD3A	-	Power supply pin

Table 1-1 Pin Names and Functions (5/8)

Type	Pin No.	Pin Name	Input/ Output	Function
Function	51	PH3 $\overline{\text{CS3}}$ MTOUT02 MTTB2OUT	Input/Output Output Output Output	Input/Output port Chip select pin Multi-purpose timer (IGBT mode) output pin Multi-purpose timer (timer mode) output pin
Function	52	PH2 $\overline{\text{CS2}}$ MTOUT12 MTTB2IN SCK2	Input/Output Output Output Input Input/Output	Input/Output port Chip select pin Multi-purpose timer (IGBT mode) output pin Multi-purpose timer (timer mode) input pin SIO mode clock pin
Function	53	PH1 $\overline{\text{CS1}}$ TB4OUT $\overline{\text{GEMG2}}$ SI2/SCL2	Input/Output Output Output Input Input/Output	Input/Output port Chip select pin Timer B output pin Multi-purpose timer (IGBT mode) abnormal status detection input SIO mode receive pin, I2C mode clock pin
Function	54	PH0 $\overline{\text{BELH}}$ TB5OUT MT2IN SO2/SDA2	Input/Output Output Output Input Input/Output	Input/Output port Byte enable pin Timer B output pin Multi-purpose timer (IGBT mode) input pin SIO mode transmit pin, I2C mode transmit/receive pin
PS	55	DVSSA	-	GND pin
PS	56	RVDD3	-	Power supply pin
Clock	57	XT2	Output	Connect to a Low-speed oscillator
Clock	58	XT1	Input	Connect to a Low-speed oscillator
PS	59	DVDD3A	-	Power supply pin
Clock	60	X2	Output	Connect to a high-speed oscillator
PS	61	DVSSA	-	GND pin
Clock	62	X1	Input	Connect to a high-speed oscillator
PS	63	DVSSB	-	GND pin
Function	64	USB-DDM	Input/Output	USB data minus (device) (Note) Pulled-down USB-DDM when USB device is not used.
Function	65	USB-DDP	Input/Output	USB data plus (device) (Note) Pulled-down USB-DDP when USB device is not used.
PS	66	DVDD3B	-	Power supply pin
Function	67	UTEST1	-	- Pulled-down UTEST1.
Function	68	UTEST2	-	- Pulled-down UTEST2.
PS	69	DVSSB	-	GND pin

Table 1-1 Pin Names and Functions (6/8)

Type	Pin No.	Pin Name	Input/ Output	Function
PS	70	DVSSA	-	GND pin
Function	71	PK4 RXIN SP0CLK SCK0	Input/Output Input Input/Output Input/Output	Input/Output port Remote control input pin SSP clock pin SIO mode clock pin
Function	72	PK3 SP0DO SI0/SCL0	Input/Output Output Input/Output	Input/Output port SSP data output pin SIO mode receive pin, I2C mode clock pin
Function	73	PK2 USB_ECLK SP0DI SO0/SDA0	Input/Output Input Input Input/Output	Input/Output port USB clock input SSP data input pin SIO mode transmit pin, I2C mode transmit/receive pin
Function	74	PK1 SP0FSS INT8 TB6OUT	Input/Output Input/Output Input Output	Input/Output port SSP frame/slave selection pin External interrupt pin Timer B output pin
Function	75	PK0 USBDPON (INTD)	Input/Output Input (Input)	Input/Output port (5V tolerant input) (note) USB power-on detection (USB device) (External interrupt pin)
Function	76	PL0 INT2 MT1IN $\overline{\text{ADTRGA}}$	Input/Output Input Input Input	Input/Output port External interrupt pin Multi-purpose timer (IGBT mode) input pin External activation pin for AD converter
Function	77	PL1 $\overline{\text{GEMG1}}$ $\overline{\text{DATRG}}$ RXD2	Input/Output Input Input Input	Input/Output port Multi-purpose timer (IGBT mode) abnormal status detection input External activation pin for DA converter SIO receive pin
Function	78	PL2 MTOUT11 MTTB1IN TXD2	Input/Output Output Input Output	Input/Output port Multi-purpose timer (IGBT mode) output pin Multi-purpose timer (timer mode) input pin SIO transmit pin
Function	79	PL3 MTOUT01 MTTB1OUT SCLK2 $\overline{\text{CTS2}}$	Input/Output Output Output Input/Output Input	Input/Output port Multi-purpose timer (IGBT mode) output pin Multi-purpose timer (timer mode) output pin SIO clock pin Handshake pin
PS	80	DVDD3A	-	Power supply pin
Function/ Debug	81	PB1 TRACEDATA3 RXD3	Input/Output Output Input	Input/Output port Debug pin SIO receive pin

Table 1-1 Pin Names and Functions (7/8)

Type	Pin No.	Pin Name	Input/ Output	Function
Function/ Debug	82	PB0 TRACEDATA2 TXD3	Input/Output Output Output	Input/Output port Debug pin SIO transmit pin
Function/ Debug	83	PA7 TRACEDATA1 $\overline{\text{CTS5}}$ SCLK3 $\overline{\text{CTS3}}$ TB7OUT	Input/Output Output Input Input/Output Input Output	Input/Output port Debug pin Handshake pin SIO clock pin Handshake pin Timer B output pin
Function/ Debug	84	PA6 TRACEDATA0 TXD5 IROUT5	Input/Output Output Output Output	Input/Output port Debug pin UART transmit pin IrDA1.0 transmit pin
Function/ Debug	85	PA5 TRACECLK RXD5 IRIN5	Input/Output Output Input Input	Input/Output port Debug pin UART receive pin IrDA1.0 receive pin
Function/ Debug	86	PA4 $\overline{\text{TRST}}$ $\overline{\text{RTS5}}$	Input/Output Input Output	Input/Output port Debug pin Modem control($\overline{\text{RTS}}$)
Function/ Debug	87	PA3 TDI DCD5 INT3	Input/Output Input Input Input	Input/Output port Debug pin Modem status (DCD) External interrupt pin
Function/ Debug	88	PA2 TCK/SWCLK RIN5	Input/Output Input Input	Input/Output port Debug pin Modem status (RIN)
Function/ Debug	89	PA1 TMS/SWDIO DSR5	Input/Output Input./Output Input	Input/Output port Debug pin Modem status (DSR)
Function/ Debug	90	PA0 TDO/SWV DTR5	Input/Output Output Output	Input port Debug pin Modem control (DTR)
PS	91	DVSSA	-	GND pin
Function	92	$\overline{\text{NMI}}$	Input	Non-maskable interrupt
Function	93	$\overline{\text{RESET}}$	Input	Reset input pin
Control	94	MODE	Input	Mode pin (note) MODE must be connected to GND.
PS	95	AVDD3B	-	Supplying the AD converter with a power supply. (note) AVDD3 must be connected to power supply even if AD converter is not used.

Table 1-1 Pin Names and Functions (8/8)

Type	Pin No.	Pin Name	Input/ Output	Function
PS	96	AVSSB	-	AD converter GND pin (note) AVSSB must be connected to GND even if the AD converter is not used.
Function	97	PI4 AINB0	Input/Output Input	Input/Output port Analog input
Function	98	PI5 AINB1	Input/Output Input	Input/Output port Analog input
Function	99	PI6 AINB2	Input/Output Input	Input/Output port Analog input
Function	100	PI7 AINB3	Input/Output Input	Input/Output port Analog input

Note: Only when input is enabled, these pins tolerate 5V inputs. Note that these pins cannot be pulled up over the power supply voltage when using as open-drain output.

1.5 Power Supplies and Power Supplied Pins

Table 1-2 Power supplies and power supplied pins

Power supply	Voltage range	Pin No.	Power supplied pin
DVDD3A	2.7V to 3.6V 3.0 to 3.45V (when USB is used.)	21 31 50 59 80	PA0-7, PB0-6, PE0-7, PF0-7, PG0-7, PH0-3, PK0-4, PL0-3, X1, X2, XT1, XT2, $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, MODE
DVSSA	0V	16 26 44 55 61 70 91	
DVDD3B	2.7V to 3.6V 3.0 to 3.45V (when USB is used.)	66	USB-DDM, USB-DDP UTEST1, UTEST2
DVSSB	0V	63 69	
AVDD3A	2.7V to 3.6V 3.0 to 3.45V (when USB is used.)	3	PI0-3
AVSSA	0V	4	
AVDD3B	2.7V to 3.6V 3.0 to 3.45V (when USB is used.)	95	PI4-PI7
AVSSB	0V	96	
AVDD3_DA	2.7V to 3.6V 3.0 to 3.45V (when USB is used.)	11	DA0 DA1
AVSS_DA	0V	14	
RVDD3	2.7V to 3.6V 3.0 to 3.45V (when USB is used.)	56	-

2. Processor Core

The TX03 series has a high-performance 32-bit processor core (the ARM Cortex-M3 processor core). For information on the operations of this processor core, please refer to the "Cortex-M3 Technical Reference Manual" issued by ARM Limited.This chapter describes the functions unique to the TX03 series that are not explained in that document.

2.1 Information on the processor core

The following table shows the revision of the processor core in the TMPM367FDFG.

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
TMPM367FDFG	r2p1

2.2 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 TMPM367FDFG.

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

2.3 Exceptions/ Interruptions

Exceptions and interruptions are described in the following section.

2.3.1 Number of Interrupt Inputs

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

TMPM367FDFG has 109 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, 0x00 is read out.

2.3.2 Number of Priority Level Interrupt Bits

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

TMPM367FDFG 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.

2.3.3 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.

2.3.4 SYSRESETREQ

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

TMPM367FDFG provides the same operation when SYSRESETREQ signal are output.

2.3.5 LOCKUP

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

TMPM367FDFG does not use this signal. To return from LOCKUP status, it is necessary to use non-maskable interruput (NMI) or reset.

2.3.6 Auxiliary Fault Status register

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

However, TMPM367FDFG is not defined this function. If auxiliary fault status register is read, always "0x0000_0000" is read out.

2.4 Events

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

TMPM367FDFG does not use event output signals and event input signals. Please do not use SEV instruction and WFE instruction.

2.5 Power Management

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

TMPM367FDFG 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."

2.6 Exclusive access

In Cortex-M3 core, the DCode bus system supports exclusive access. However TMPM367FDFG does not use this function.

3. Memory Map

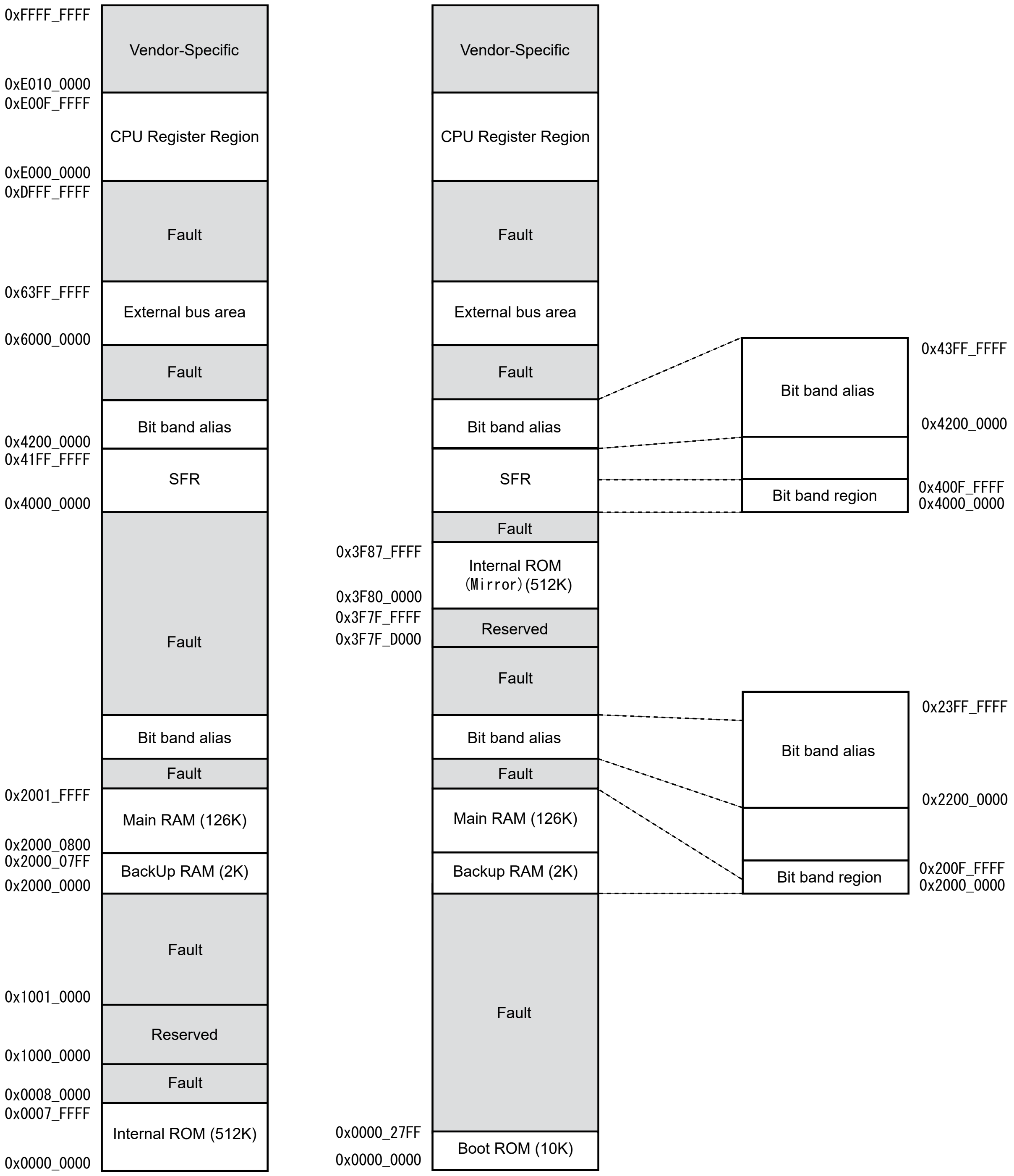
3.1 Memory Map

The memory maps for TMPM367FDFG are based on the ARM Cortex-M3 processor core memory map. The internal ROM, internal RAM and special function registers (SFR) of TMPM367FDFG 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 SRAM and SFR areas are all included in the bit-band region.

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

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

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


0xFFFF_FFFF
Vendor-Specific
0xE010_0000
0xE00F_FFFF
CPU Register Region
0xE000_0000
0xDFFF_FFFF
Fault
0x63FF_FFFF
External bus area
0x6000_0000
Fault
Bit band alias
0x4200_0000
0x41FF_FFFF
SFR
0x4000_0000
Fault
Bit band alias
Fault
0x2001_FFFF
Main RAM (126K)
0x2000_0800
0x2000_07FF
BackUp RAM (2K)
0x2000_0000
Fault
0x1001_0000
Reserved
0x1000_0000
Fault
0x0008_0000
0x0007_FFFF
Internal ROM (512K)
0x0000_0000
Vendor-Specific
CPU Register Region
Fault
External bus area
Fault
Bit band alias
SFR
Fault
0x3F87_FFFF
Internal ROM (Mirror)(512K)
0x3F80_0000
0x3F7F_FFFF
Reserved
0x3F7F_D000
Fault
Bit band alias
Fault
Main RAM (126K)
Backup RAM (2K)
Fault
0x0000_27FF
Boot ROM (10K)
0x0000_0000
Bit band alias
0x43FF_FFFF
0x4200_0000
Bit band region
0x400F_FFFF
0x4000_0000
Bit band alias
0x23FF_FFFF
0x2200_0000
Bit band region
0x200F_FFFF
0x2000_0000

Single chip mode Single boot mode

3.2 Bus Matrix

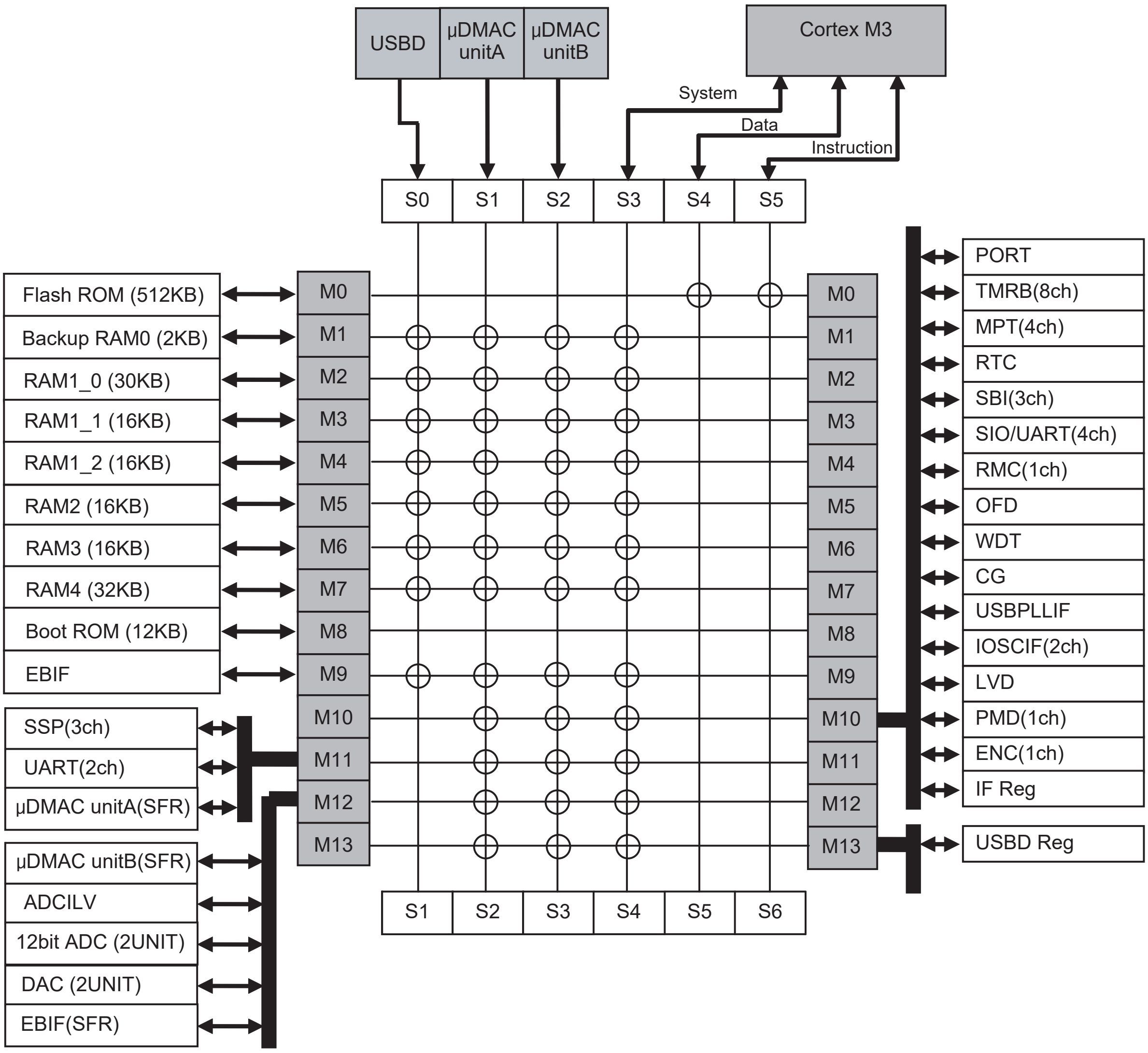
This MCU contains three bus masters such as a CPU core, a μDMA controller and a USBD controller .

Bus masters connect to slave ports (S0 to S5) of Bus Matrix. In the bus matrix, master ports (M0 to M13) connect to peripheral functions via connections described as (o) in the following figure.

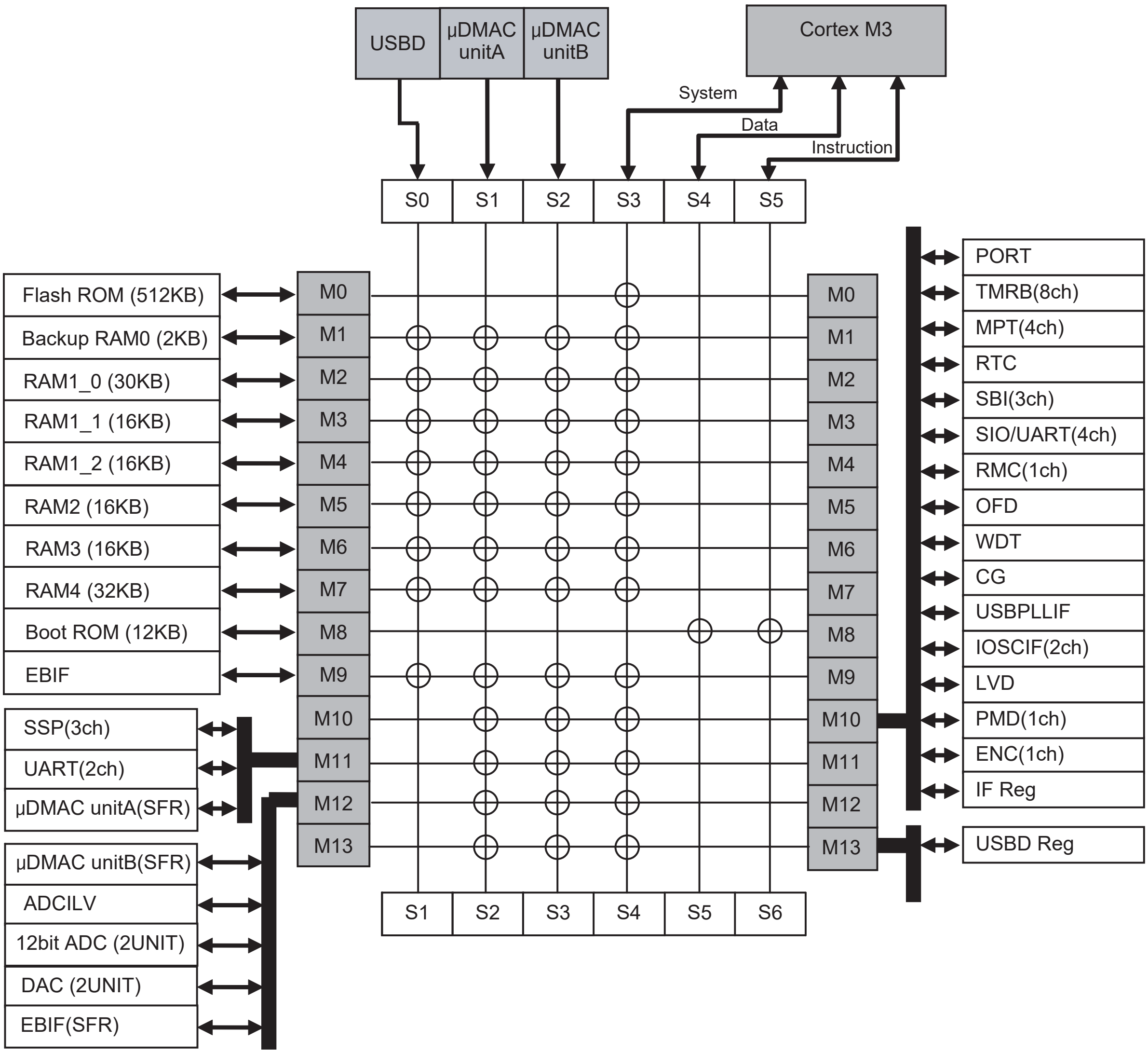
While multiple slaves are connected on the same bus master line in the Bus Matrix, if multiple slave accesses are generated at the same time, a priority is given to access from a master with the smallest slave number.

3.2.1 Structure

3.2.1.1 Single chip mode


USBD
μDMAC unitA
μDMAC unitB
Cortex M3
System
Data
Instruction
S0
S1
S2
S3
S4
S5
Flash ROM (512KB)
Backup RAM0 (2KB)
RAM1_0 (30KB)
RAM1_1 (16KB)
RAM1_2 (16KB)
RAM2 (16KB)
RAM3 (16KB)
RAM4 (32KB)
Boot ROM (12KB)
EBIF
SSP(3ch)
UART(2ch)
μDMAC unitA(SFR)
μDMAC unitB(SFR)
ADCILV
12bit ADC (2UNIT)
DAC (2UNIT)
EBIF(SFR)
M0
M1
M2
M3
M4
M5
M6
M7
M8
M9
M10
M11
M12
M13
S1
S2
S3
S4
S5
S6
PORT
TMRB(8ch)
MPT(4ch)
RTC
SBI(3ch)
SIO/UART(4ch)
RMC(1ch)
OFD
WDT
CG
USBPLLIF
IOSCIF(2ch)
LVD
PMD(1ch)
ENC(1ch)
IF Reg
USBD Reg

3.2.1.2 Single boot mode


USBD
µDMAC unitA
µDMAC unitB
Cortex M3
System
Data
Instruction
S0
S1
S2
S3
S4
S5
S6
M0
M1
M2
M3
M4
M5
M6
M7
M8
M9
M10
M11
M12
M13
Flash ROM (512KB)
Backup RAM0 (2KB)
RAM1_0 (30KB)
RAM1_1 (16KB)
RAM1_2 (16KB)
RAM2 (16KB)
RAM3 (16KB)
RAM4 (32KB)
Boot ROM (12KB)
EBIF
SSP(3ch)
UART(2ch)
µDMAC unitA(SFR)
µDMAC unitB(SFR)
ADCILV
12bit ADC (2UNIT)
DAC (2UNIT)
EBIF(SFR)
PORT
TMRB(8ch)
MPT(4ch)
RTC
SBI(3ch)
SIO/UART(4ch)
RMC(1ch)
OFD
WDT
CG
USBPLLIF
IOSCIF(2ch)
LVD
PMD(1ch)
ENC(1ch)
IF Reg
USBD Reg

3.2.2 Connection table

3.2.2.1 Code area / SRAM area

(1) Single chip mode

Start Address			USBD	µDMAC unitA	µDMAC unitB	Core S-Bus	Core D-Bus	Core I-Bus
			S1	S2	S3	S4	S5	S6
0x0000_0000	Flash ROM	M0	Fault	Fault	Fault	Fault	o	o
0x0008_0000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault
0x1000_0000	Reserved	-	Fault	Fault	Fault	Fault	Reserved	Reserved
0x1001_0000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault
0x2000_0000	Backup RAM (mirror)	M1	o	o	o	o	Fault	Fault
0x2000_0800	Main RAM1_0	M2	o	o	o	o	Fault	Fault
0x2000_8000	Main RAM1_1	M3	o	o	o	o	Fault	Fault
0x2000_C000	Main RAM1_2	M4	o	o	o	o	Fault	Fault
0x2001_0000	Main RAM2	M5	o	o	o	o	Fault	Fault
0x2001_4000	Main RAM3	M6	o	o	o	o	Fault	Fault
0x2001_8000	Main RAM4	M7	o	o	o	o	Fault	Fault
0x2002_0000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault
0x2200_0000	Bit band alias	-	Fault	Fault	Fault	o	Fault	Fault
0x2240_0000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault

(2) Single boot mode

Start Address			USBD	µDMAC unitA	µDMAC unitB	Core S-Bus	Core D-Bus	Core I-Bus
			S0	S1	S2	S3	S4	S5
0x0000_0000	Boot ROM	M8	Fault	Fault	Fault	Fault	o	o
0x0000_2800	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault
0x2000_0000	Backup RAM	M1	o	o	o	o	Fault	Fault
0x2000_0800	Main RAM1_0	M2	o	o	o	o	Fault	Fault
0x2000_8000	Main RAM1_1	M3	o	o	o	o	Fault	Fault
0x2000_C000	Main RAM1_2	M4	o	o	o	o	Fault	Fault
0x2001_0000	Main RAM2	M5	o	o	o	o	Fault	Fault
0x2001_4000	Main RAM3	M6	o	o	o	o	Fault	Fault
0x2001_8000	Main RAM4	M7	o	o	o	o	Fault	Fault
0x2002_0000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault
0x2000_0000	Bit band alias	-	Fault	Fault	Fault	o	Fault	Fault
0x2240_0000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault
0x3F7F_C000	Reserved	-	Fault	Fault	Fault	Reserved	Fault	Fault
0x3F7F_E800	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault
0x3F80_0000	Flash ROM (mirror)	-	Fault	Fault	Fault	o	Fault	Fault
0x3F88_0000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault

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

3.2.2.2 Peripheral area / External bus area

Start Address			USBD	μDMAC unitA	μDMAC unitB	Core S-Bus	Core D-Bus	Core I-Bus
			S0	S1	S2	S3	S4	S5
0x4004_0000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault
0x4000_5000	Reserved	-	Fault	Reserved	Reserved	Reserved	Fault	Fault
0x4000_6000	Reserved	-	Fault	Reserved	Reserved	Reserved	Fault	Fault
0x4000_7000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault
0x4000_8000	USBD	M13	Fault	o	o	o	Fault	Fault
0x4000_9000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault
0x4000_A000	Reserved	-	Fault	Reserved	Reserved	Reserved	Fault	Fault
0x4000_C000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault
0x4004_0000	SSP	M11	Fault	o	o	o	Fault	Fault
0x4004_8000	UART	M11	Fault	o	o	o	Fault	Fault
0x4004_C000	μDMAC unitA(SFR)	M11	Fault	o	o	o	Fault	Fault
0x4004_D000	μDMAC unitB(SFR)	M12	Fault	o	o	o	Fault	Fault
0x4005_0000	ADC	M12	Fault	o	o	o	Fault	Fault
0x4005_2000	ADCILV	M12	Fault	o	o	o	Fault	Fault
0x4005_4000	DAC	M12	Fault	o	o	o	Fault	Fault
0x4005_C000	EBIF(SFR)	M12	Fault	o	o	o	Fault	Fault
0x4005_D000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault
0x400C_0000	PORT	M10	Fault	o	o	o	Fault	Fault
0x400C_4000	TMRB		Fault	o	o	o	Fault	Fault
0x400C_7000	MPT		Fault	o	o	o	Fault	Fault
0x400C_C000	RTC		Fault	o	o	o	Fault	Fault
0x400E_0000	SBI		Fault	o	o	o	Fault	Fault
0x400E_1000	SIO/UART		Fault	o	o	o	Fault	Fault
0x400E_7000	RMC		Fault	o	o	o	Fault	Fault
0x400F_1000	OFD		Fault	o	o	o	Fault	Fault
0x400F_2000	WDT		Fault	o	o	o	Fault	Fault
0x400F_3000	CG		Fault	o	o	o	Fault	Fault
0x400F_3000	USBPLLIF		Fault	o	o	o	Fault	Fault
0x400F_3000	IDSCIF		Fault	o	o	o	Fault	Fault
0x400F_3200	TRMOSC		Fault	o	o	o	Fault	Fault
0x400F_4000	LVD		Fault	o	o	o	Fault	Fault
0x400F_6000	PMD		Fault	o	o	o	Fault	Fault
0x400F_6000	MPT		Fault	o	o	o	Fault	Fault
0x400F_7000	ENC		Fault	o	o	o	Fault	Fault
0x4010_0000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault
0x41FF_F000	FLASH	M10	Fault	o	o	o	Fault	Fault
0x4200_0000	Bit band alias	-	Fault	Fault	Fault	o	Fault	Fault
0x4400_0000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault
0x6000_0000	EBIF	M9	Fault	o	o	o	Fault	Fault
0x6400_0000	Fault	-	Fault	Fault	Fault	Fault	Fault	Fault

3.2.3 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
USB Device Controller (USBD)	ch1	0x4000_8000
Synchronous Serial Port (SSP)	ch0	0x4004_0000
	ch1	0x4004_1000
	ch2	0x4004_2000
Asynchronous Serial Channel (UART)	ch4	0x4004_8000
	ch5	0x4004_9000
μDMA Controller (μDMAC)	unitA	0x4004_C000
	unitB	0x4004_D000
Analog / Digital Converter (ADC)	unitA	0x4005_0000
	unitB	0x4005_1000
	ADCILV	0x4005_2000
Digital / Analog Converte r(DAC)	unitA	0x4005_4000
	unitB	0x4005_5000
External bus interface (EBIF)		0x4005_C000
Input / Output port	PORTA	0x400C_0000
	PORTB	0x400C_0100
	PORTC	0x400C_0200
	PORTE	0x400C_0400
	PORTF	0x400C_0500
	PORTG	0x400C_0600
	PORTH	0x400C_0700
	PORTI	0x400C_0800
	PORTJ	0x400C_0900
	PORTK	0x400C_0A00
	PORTL	0x400C_0B00
16-bit Timer / Event Counters (TMRB)	ch0	0x400C_4000
	ch1	0x400C_4100
	ch2	0x400C_4200
	ch3	0x400C_4300
	ch4	0x400C_4400
	ch5	0x400C_4500
	ch6	0x400C_4600
	ch7	0x400C_4700
16-bit Multi-Purpose Timer (MPT)	MPT0	0x400C_7000
	MPT1	0x400C_7100
	MPT2	0x400C_7200
	MPT3	0x400C_7300
	PMD0	0x400F_6000
Real Time Clock (RTC)		0x400C_C000
Serial Bus Interface (I2C/SIO)	ch0	0x400E_0000
	ch1	0x400E_0100
	ch2	0x400E_0200

Peripheral Function		Base Address
Serial Channel (SIO/UART)	ch0	0x400E_1000
	ch1	0x400E_1100
	ch2	0x400E_1200
	ch3	0x400E_1300
Remote control signal preprocessor (RMC)		0x400E_7000
Ocsillation Frequency Detector (OFD)		0x400F_1000
Watchdog Timer(WDT)		0x400F_2000
Clock/Mode control		0x400F_3000
Internal High-speed Oscillation Adjustment Function (TRMOSC)		0x400F_3200
Low Voltage Detection Circuit (LVD)		0x400F_4000
Encoder Input Circuit (ENC)	ch0	0x400F_7000
Flash Control		0x41FF_F000

4. Internal High-speed Oscillation Adjustment Function (TRMOSC)

TMPM367FDFG has the internal high-speed oscillation adjustment function.

Note: This adjustment function is not applicable to the reference clock for OFD.

4.1 Register Description

4.1.1 Register List

The control registers and its addresses are as follows:

Base Address = 0x400F_3200

Register name		Address(Base+)
Protect register	TRMOSCPRO	0x0000
Enable register	TRMOSCEN	0x0004
Initial trimming value monitoring register	TRMOSCINIT	0x0008
Trimming value setting register	TRMOSCSET	0x000C

4.1.2 TRMOSCPRO (Protect 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	PROTECT							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PROTECT	R/W	Writing register control 0xC1 : Enable Other than 0xC1 : Disable When "0xC1" is set, TRMOSCEN, TRMOSCINIT and TRMOSCSET are allowed to write.

4.1.3 TRMOSCEN (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	-	-	-	-	-	-	-	TRIMEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	TRIMEN	R/W	Trimming control 0 : Disable 1 : Enable When "1" is set, a trimming value of the internal oscillator is switched to a value read from TRIMOSCINIT to a value set in TRMOSCSET.

4.1.4 TRMOSCINIT (Initial trimming value monitoring 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	-	-	TRIMINITC					
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	TRIMINITF			
After reset	0	0	0	0	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-14	−	R	Read as "0".
13-8	TRIMINITC	R	Initial coarse trimming value Reads an initial coarse trimming value before shipping.
7-4	−	R	Read as "0".
3-0	TRIMINITF	R	Initial delicate trimming value Reads an initial delicate trimming value before shipping.

Note: For details about the specific setting and adjustment value of coarse trimming and delicate trimming, refer to Section"4.2.2 Adjustment range".

4.1.5 TRMOSCSET (Trimming value 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	-	-	TRIMSETC					
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	TRIMSETF			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-14	−	R	Read as "0".
13-8	TRIMSETC	RW	Coarse trimming value setting Sets a coarse trimming value.
7-4	−	R	Read as "0".
3-0	TRIMSETF	RW	Delicate trimming value setting Sets a delicate trimming value.

Note: For details about the specific setting and adjustment value of coarse trimming and delicate trimming, refer to Section"4.2.2 Adjustment range".

4.2 Operational Description

4.2.1 Adjustment

Oscillation is adjusted using coarse trimming values and delicate trimming values.

The value setting before shipping can be checked with TRMOSCINIT<TRIMINITC> or <TRIMINITF>.

When the value changing, set a new value to TRMOSCSET<TRIMSETC> or <TRIMSETF>. By setting "1" to TRMOSCEN<TRIMEN>, a setting value of the internal oscillator will be changed.

Note: After reset, writing to TRMOSCSET and TRMOSCEN is prohibited. When writing to these bits, TRMOSCPRO<PROTECT> must be set to "0xC1".

4.2.2 Adjustment range

In the coarse trimming, -57.6% to +55.8% adjustment by 1.8%-step is feasible. In the delicate trimming, -2.4% to +2.1% adjustment by 0.3%-step is feasible. Table 4-1 and Table 4-2 show a adjustment range.

Note: Each step value is assumed based on the typical condition. In the coarse trimming, it has ± 0.2% margin of error. In the delicate trimming, it has ± 0.1% margin of error.

Table 4-1 Adjustment range of coarse trimming

Coarse trimming	
<TRIMSETC>	Frequency change (typ.)
011111	+55.8%
.	.
000001	+1.8%
000000	±0%
111111	-1.8%
111110	-3.6%
.	.
100000	-57.6%

Table 4-2 Adjustment range delicate trimming

Delicate trimming	
<TRIMSETF>	Frequency change (typ.)
0111	+2.1%
.	.
0001	+0.3%
0000	±0
1111	-0.3%
1110	-0.6%
.	.
1000	-2.4%

4.2.3 Example of Internal High-speed Oscillation by using 16-bit timer/event counter (TMRB)

The internal oscillation adjustment function uses the pulse width measurement function of 16-bit timer/event counter (TMRB).

4.2.3.1 The way of input a pulse into TBxIN pin

First, choose an internal oscillator as a prescaler clock φT0 of TMRB.

Second, input a pulse from TBxIN. Third, capture an up-counter value at the rising edge of the pulse using the capture function.

Finally, determine the adjustment value using a difference between a frequency of TBxIN calculated with capture value and the actual frequency.

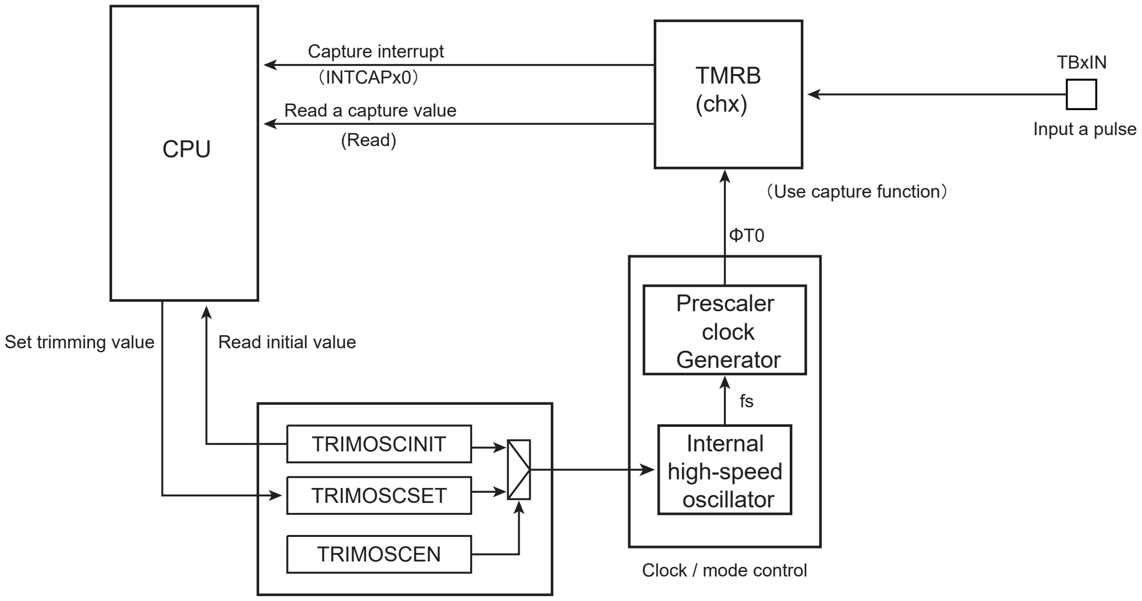

CPU
Capture interrupt
（INTCAPx0）
Read a capture value
(Read)
TMRB
(chx)
TBxIN
Input a pulse
（Use capture function）
ΦT0
Prescaler clock Generator
fs
Internal high-speed oscillator
Clock / mode control
Set trimming value
Read initial value
TRIMOSCINIT
TRIMOSCSET
TRIMOSCEN

Figure 4-1 The way of input a pulse into TBxIN pin

4.2.3.2 The way of input fs into TB5IN

In TMPM367FDFG, fs is input into TB5IN. A pulse is made by TMRB ch5 which uses fs.

TB5OUT of TMRB ch5 is connected with TMRB ch6 and ch7 internally.

Choose an internal oscillator as a prescaler clock φT0 of TMRB ch6 or ch7.

Capture an up-counter value at the rising edge of the TB5OUT using the capture function.

Determine the adjustment value using a difference between a frequency of TBxIN calculated with capture value and the actual frequency.

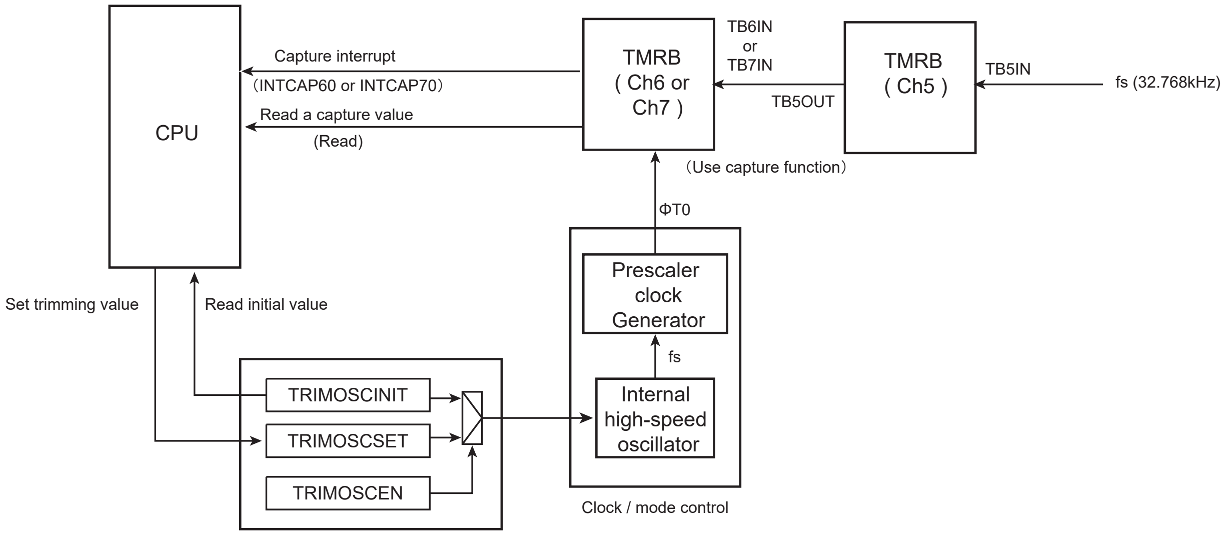

CPU
Capture interrupt
（INTCAP60 or INTCAP70）
Read a capture value
(Read)
TMRB
(Ch6 or Ch7)
TB6IN
or
TB7IN
TB5OUT
TMRB
(Ch5)
TB5IN
fs (32.768kHz)
（Use capture function）
ΦT0
Prescaler clock Generator
fs
Internal high-speed oscillator
Clock / mode control
Set trimming value
Read initial value
TRIMOSCINIT
TRIMOSCSET
TRIMOSCEN

Figure 4-2 The way of input fs into TB5IN

5. Clock/Mode control

5.1 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.

5.2 Registers

5.2.1 Register List

The following table shows the Clock/Mode control related registers and addresses.

Base Address = 0x400F_3000

Register name		Address (Base+)
System control register	CGSYSCR	0x0000
Oscillation control register	CGOSCCR	0x0004
Standby control register	CGSTBYCR	0x0008
PLL selection register	CGPLLSEL	0x000C
Reserved	-	0x0010
Reserved	-	0x0014
Reserved	-	0x0015
Clock stop register for peripheral	CGCKSTP	0x0018
Reserved	-	0x001C
Reserved	-	0x0038
Protect register	CGPROTECT	0x003C

Note: Access to the "Reserved" area is prohibited.

5.2.2 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
31-21	−	R	Read as "0".
20	FCSTOP	R/W	ADC clock 0: Active 1: Stop Enables to stop providing ADC clock. ADC clock is provided after reset. Confirming that ADC is stopped or finished in advance is required when setting "1"(stop). Enables to confirm that ADC is stopped or finished.
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	−	R/W	Write "0"
12	FPSEL	−	fperiph source clock 0: fgear 1: fc Specifies the source clock to fperiph. Selecting fc fixes fperiph regardless of the clock gear mode.
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 circuit.
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

5.2.3 CGOSCCR (Oscillation control register)

	31	30	29	28	27	26	25	24
bit symbol	WUPT							
After reset	1	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	WUPT				WUPSEL2	EHOSCSEL	OSCSEL	XEN2
After reset	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
bit symbol	WUPTL		-	-	-	XEN3	XTEN	XEN1
After reset	0	0	0	0	0	1	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	WUPT[11:0]	R/W	Warm-up counter setup value. Setup the 16-bit timer for warm-up timer of upper 12-bits counter value.
19	WUPSEL2	R/W	High-speed warm-up clock. 0: internal (f_{IHOSC}) 1: external (f_{EHOSC}) Selects warm-up counter by high-speed oscillator. The warm-up counter is counted up by the selected clock.
18	EHOSCSEL	R/W	External oscillator. 0: input external clock 1: external oscillator
17	OSCSEL	R/W	High-speed oscillator 0: internal high-speed oscillator 1: external high-speed oscillator
16	XEN2	R/W	Internal high-speed oscillator operation (for SYS) 0: Stop 1: Oscillation
15-14	WUPTL[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.
13-12	−	R/W	Write "0".
11	−	R	Read as "0".
10	XEN3	R/W	Internal high-speed oscillator operation (for OFD) 0: Stop 1: Oscillation
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	Select warm-up counter 0: High-speed 1: Low-speed
2	PLLON	R/W	PLL (multiplying circuit) operation (note 3) 0: Stop 1: Oscillation

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

Note 1: Refer to 5.6.9.1 about the Warm-up setup.

Note 2: When selecting external oscillator (input external clock), select <OSCSEL> after setting <EHOSCSEL>. (Do not select simultaneously)

Note 3: Refer to "5.3.5 Clock Multiplication Circuit (PLL)" about setting PLL.

Note 4: Returning from the STOP1/STOP2 mode, related bits <WUPSEL2>, <OSCSEL>, <XEN3>, <XEN2>, <XEN1>, <PLLON> of the register CGOSCCR and CGPLLSEL<PLLSEL> are initialized because of internal high-speed oscillator starts up.

Note 5: When using internal high-speed oscillator (IHOSC), do not use it as system clock which high accuracy assurance is required.

5.2.4 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	-	-	-	-	-	-	PTKEEP	DRVE
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	-	-	-	-	-	STBY		
After reset	0	0	0	0	0	0	1	1

Bit	Bit Symbol	Type	Function
31-20	−	R	Read as "0".
19-18	−	R/W	Write "0".
17	PTKEEP	R/W	Keeps I/O control signal in STOP2 mode 0: Control by port 1: Keep status when setting 0->1 (This register must be set before entering STOP2 mode)
16	DRVE	R/W	Pin status in STOP1 mode. 0: Inactive in STOP1 mode 1: Active in STOP1 mode
15-3	−	R	Read as "0".
2-0	STBY[2:0]	R/W	Low power consumption mode 000: Reserved 001: STOP1 010: Reserved 011: IDLE 100: Reserved 101: STOP2 110: Reserved 111: Reserved

Note: Access to the "Reserved" area is prohibited.

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-1	PLLSET[14:0]	R/W	PLL multiplying value (Do not use except below) 0x5917 : Input clock 8MHz, output clock 48MHz (6 multiplying) 0x59A6 : Input clock 8MHz, output clock 80MHz (10 multiplying) 0x729E : Input clock 10MHz, output clock 80MHz (8 multiplying) 0x5A0F : Input clock 12MHz, output clock 48MHz (4 multiplying) 0x6296 : Input clock 12MHz, output clock 72MHz (6 multiplying) 0x720B : Input clock 16MHz, output clock 48MHz (3 multiplying) 0x3A92 : Input clock 16MHz, output clock 80MHz (5 multiplying)
0	PLLSEL	R/W	Use of PLL 0: f_{osc} 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 1: Select PLL multiplying value which is shown in Table 5-2.

Note 2: Refer to "5.3.5 Clock Multiplication Circuit (PLL)" about setting PLL.

Note 3: Returning from the STOP1/STOP2 mode, related bits <PLLSEL>, CGOSCCR<WUPSEL2>, <OSCSEL>, <XEN2>, <XEN1>, and <PLLON> are initialized because of internal high-speed oscillator starts up.

5.2.6 CGCKSTP (Clock stop register for peripheral)

	31	30	29	28	27	26	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	-	-	-	-	-	-	-	USBDSTP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3	−	R/W	Write "1".
2	−	R/W	Write "1".
1	−	R/W	Write "1".
0	USBDSTP	R/W	USBD clock control 0 : operation 1 : clock stop Enable / disable USBD clock

Note: After reset, USBD clock are operated. Set register to "1" after confirming stop of a circuit operation.

5.2.7 CGPROTECT (Protect 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	CGPROTECT							
After reset	1	1	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as "0".
7-0	CGPROTECT	R/W	Register protection control 0xC1 : Register write enable Except 0xC1 : Register write disable Initial value is "0xC1" as writing enable to each register and when writing except "0xC1", each register except CGPROTECT register can not be written.

5.3 Clock control

5.3.1 Clock Type

Each clock is defined as follows:

fosc : Clock generated by internal oscillator. Clock input from the X1 and X2 pins.
f_{PLL} : Clock multiplied by PLL.
fc : Clock specified by CGPLLSEL<PLLSEL> (High-speed clock)
fgear : Clock specified by CGSYSCR<GEAR[2:0]>(Gear Clock)
fsys : Clock specified by CGCKSEL (System clock)
fperiph : Clock specified by CGSYSCR<FPSEL[2:0]>
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

5.3.2 Initial Values after Reset

Reset operation initializes the clock configuration as follows.

Internal high-speed oscillator : oscillating
External high-speed oscillator : stop
PLL (phase locked loop circuit) : stop
Gear clock : fc (no frequency dividing)
Internal high-speed oscillator for OFD : stop

Reset operation causes all the clock configurations to be the same as fosc.

fc = fosc
fsys = fosc
φT0 = fosc

5.3.3 Clock system Diagram

Figure 5-1 shows the clock system diagram.

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

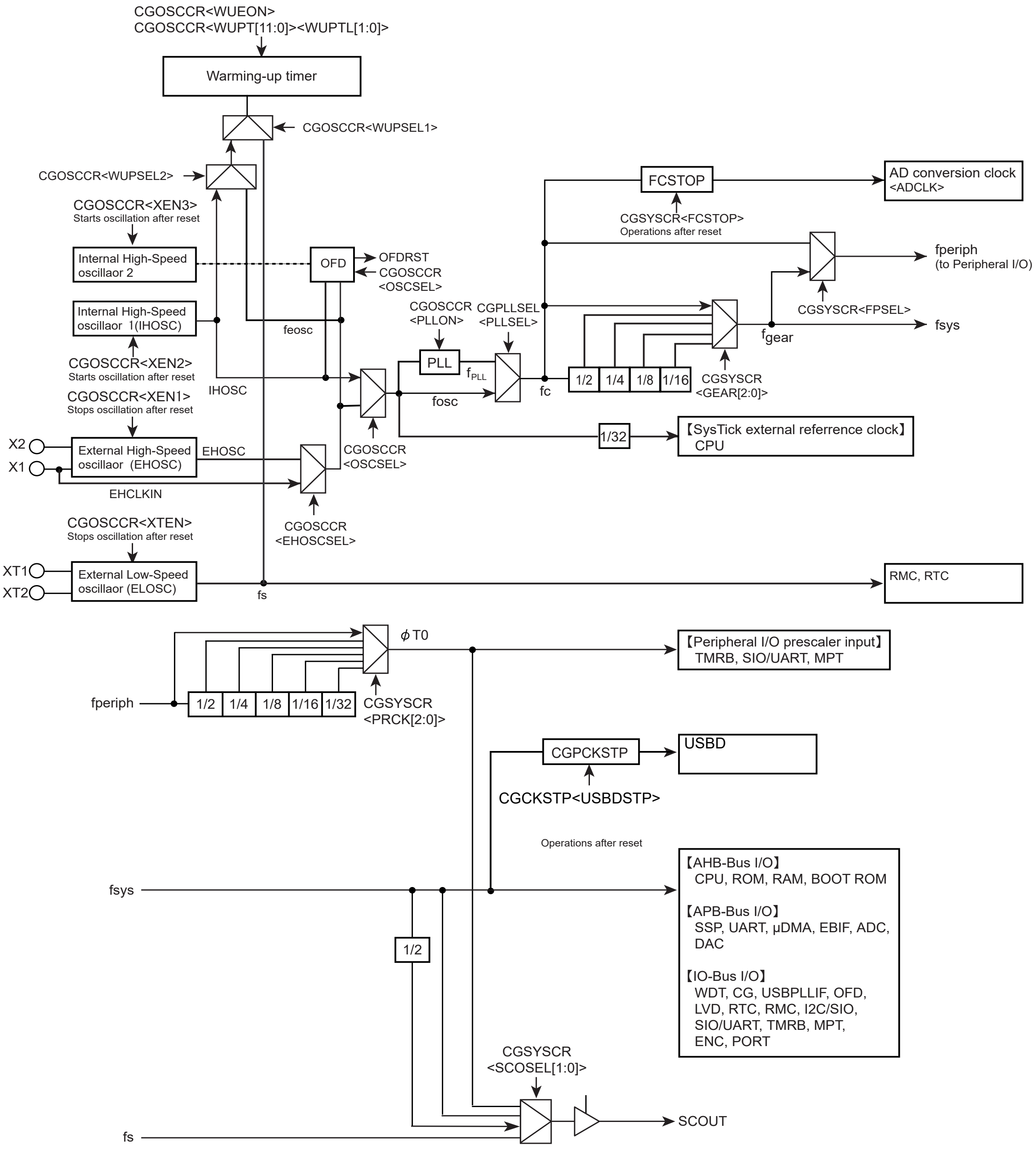

CGOSCCR<WUEON>
CGOSCCR<WUPT[11:0]><WUPTL[1:0]>
Warming-up timer
CGOSCCR<WUPSEL1>
CGOSCCR<WUPSEL2>
CGOSCCR<XEN3>
Starts oscillation after reset
Internal High-Speed oscillaor 2
OFD
OFDRST
CGOSCCR <OSCSEL>
Internal High-Speed oscillaor 1(IHOSC)
feosc
CGOSCCR<XEN2>
Starts oscillation after reset
IHOSC
CGOSCCR<XEN1>
Stops oscillation after reset
X2
X1
External High-Speed oscillaor (EHOSC)
EHOSC
EHCLKIN
CGOSCCR <OSCSEL>
CGOSCCR <EHOSCSEL>
CGOSCCR<XTEN>
Stops oscillation after reset
XT1
XT2
External Low-Speed oscillaor (ELOSC)
fs
RMC, RTC
CGOSCCR <PLLON>
PLL
CGPLLSEL <PLLSEL>
f_{PLL}
fosc
fc
FCSTOP
CGSYSCR<FCSTOP>
Operations after reset
AD conversion clock <ADCLK>
fperiph (to Peripheral I/O)
CGSYSCR<FPSEL>
fsys
f_{gear}
1/2
1/4
1/8
1/16
CGSYSCR <GEAR[2:0]>
1/32
【SysTick external referrence clock】 CPU
fperiph
1/2
1/4
1/8
1/16
1/32
CGSYSCR <PRCK[2:0]>
ϕT0
【Peripheral I/O prescaler input】 TMRB, SIO/UART, MPT
CGPCKSTP
CGCKSTP<USBDSTP>
USBD
Operations after reset
fsys
【AHB-Bus I/O】 CPU, ROM, RAM, BOOT ROM
【APB-Bus I/O】 SSP, UART, μDMA, EBIF, ADC, DAC
【IO-Bus I/O】 WDT, CG, USBPLLIF, OFD, LVD, RTC, RMC, I2C/SIO, SIO/UART, TMRB, MPT, ENC, PORT
1/2
CGSYSCR <SCOSEL[1:0]>
SCOUT
fs

Figure 5-1 Clock Block Diagram

5.3.4 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 STOP1 and STOP2. Refer to "5.6.8 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>.

2. Specify the warm-up counter value

   The warm-up time can be selected by setting the CGOSCCR<WUPT[11:0]><WUPTL[1:0]>. The value can be calculated by following formula with round lower 4 bit off, set to the bit of <WUPT[11:0]> for high-speed oscillation and set to the bit of <WUPT[11:0]><WUPTL[1:0]> for low-speed oscillation.

   Warm-up time equation and setup example are shown below.

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

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

$$\frac{\text{warm-up time to set}}{\text{input frequency cycle (s)}} = \frac{\text{5ms}}{\text{1/8MHz}} = \text{40000 cycle} = \text{0x9C40}$$

   Round lower 4 bit off, set 0x9C4 to CGOSCCR<WUPT[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<WUPT[11:0]> and <WUPTL[1:0]>, wait until this value is reflected, then transit to the low power consumption 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 5-1 The example of warm-up setting (When an internal high-speed oscillator is selected)

CGOSCCR<WUODR[11:0]> = "0x9C4" : Specify the warm-up time

↻ Read CGOSCCR<WUODR[11:0]> : Confirm warm-up time reflecting
Repeat until reading "0x9C4"

CGOSCCR<XEN2> = "1" : Internal high-speed oscillator(IHOSC) enable

CGOSCCR<WUEON> = "1" : Start the warm-up timer (WUP)

↻ CGOSCCR<WUEF> read : Wait until the state becomes "0" (warm-up is finished)

Note 1: The warm-up function is not necessary when using stable external clock.

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

Note 3: Setting warm-up count value to CGOSCCR<WUPT[11:0]><WUPTL[1:0]>, wait until this value is reflected, then transit to standby mode by executing a command "WFI".

Note 4: Returning from the STOP1/STOP2 mode, related bits <WUPSEL2>, <OSCSEL>, <XEN3>, <XEN2>, <XEN1>, <PLLON> of the register CGOSCCR and CGPLLSEL<PLLSEL> are initialized because of internal high-speed oscillator starts up.

5.3.5 Clock Multiplication Circuit (PLL)

This circuit outputs the f_{PLL} clock (80MHz max.) that is multiplied by 3, 4, 5, 6, 8 or 10 of the high-speed oscillator output clock (fosc : 8MHz to 16MHz). As a result, the input frequency to oscillator can be low frequency, and the internal clock be made high-speed.

5.3.5.1 How to configure the PLL function

The PLL is disabled after reset.

To enable the PLL, set CGPLLSEL<PLLSET> to multiplying value. And set <PLLON> to "1" after 100µs for initialize time of PLL. After 100µs for lock-up time elapses, set CGPLLSEL<PLLSEL> to "1", f_{PLL} which is multiplied by 3,4,5,6,8 or 10 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.

A Multiplying value can be selected from 3, 4, 5, 6, 8, or 10. The preset value of <PLLSET> is as follows.

Multiplying	External clock input [MHz]	<PLLSET>
6	8	0x5917
10	8	0x59A6
8	10	0x729E
4	12	0x5ADF
6	12	0x6296
3	16	0x720B
5	16	0x3A92

5.3.5.2 Change PLL multiplying

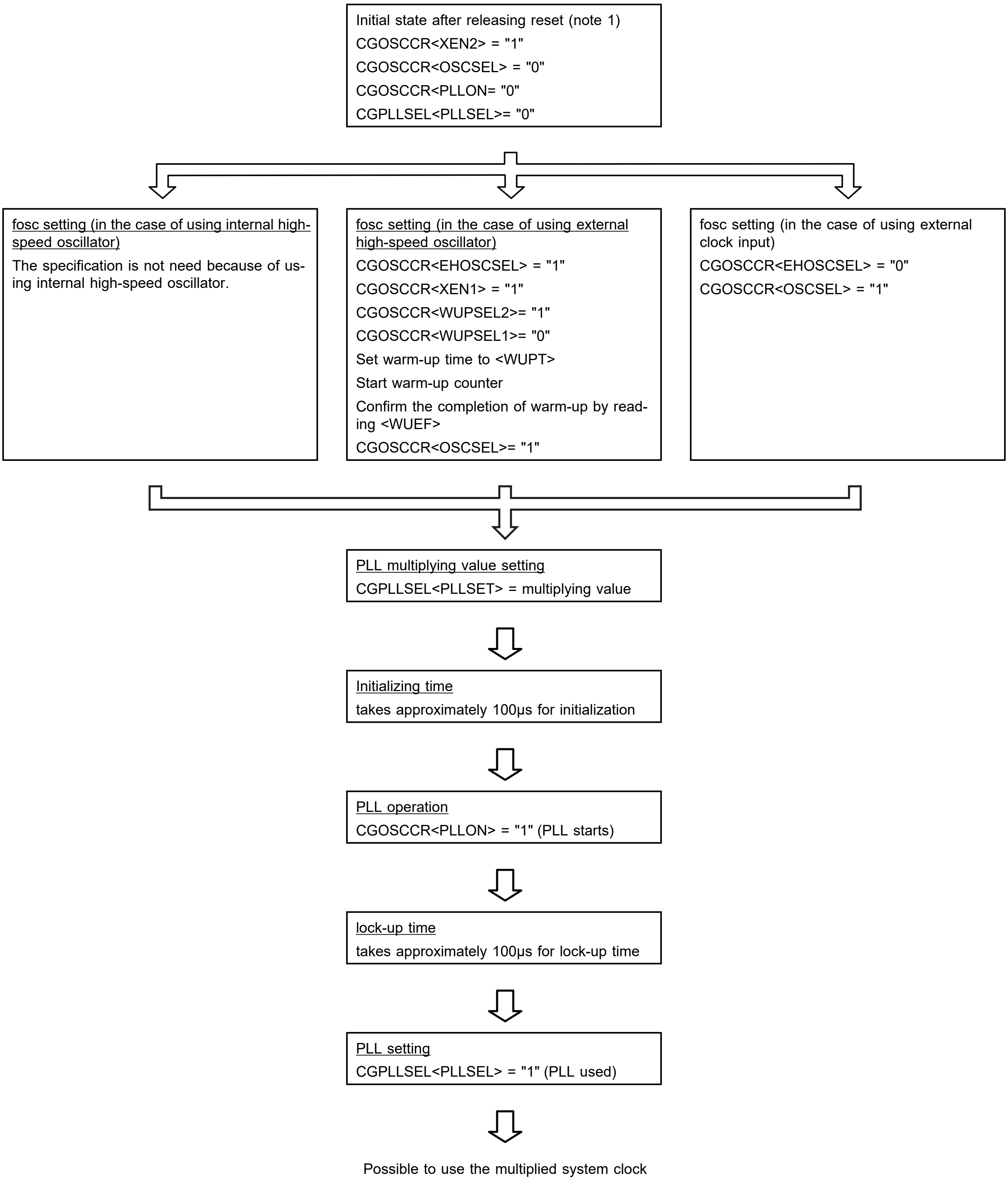
When number of multiplication is changed, firstly set "0" to CGPLLSEL<PLLSEL>. Secondly read CGPLLSEL<PLLSEL> to check the setting in which multiplication clock is not used (CGPLLSEL<PLLSEL>="0"). Thirdly, set "0" to <PLLON> to stop PLL.

Modify CGPLLSEL<PLLSEL> to multiplying value. And set <PLLON> to "1" after 100µs for initialize time of PLL. After 100µs for lock-up time elapses, set CGPLLSEL<PLLSEL> to "1".

5.3.5.3 The sequence of PLL setting

The sequence of PLL setting is shown below.

The sequence of PLL setting



Initial state after releasing reset (note 1)
CGOSCCR<XEN2> = "1"
CGOSCCR<OSCSEL> = "0"
CGOSCCR<PLLON= "0"
CGPLLSEL<PLLSEL>= "0"

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

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

fosc setting (in the case of using external clock input)
CGOSCCR<EHOSCSEL> = "0"
CGOSCCR<OSCSEL> = "1"

PLL multiplying value setting
CGPLLSEL<PLLSET> = multiplying value

Initializing time
takes approximately 100μs for initialization

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

lock-up time
takes approximately 100μs for lock-up time

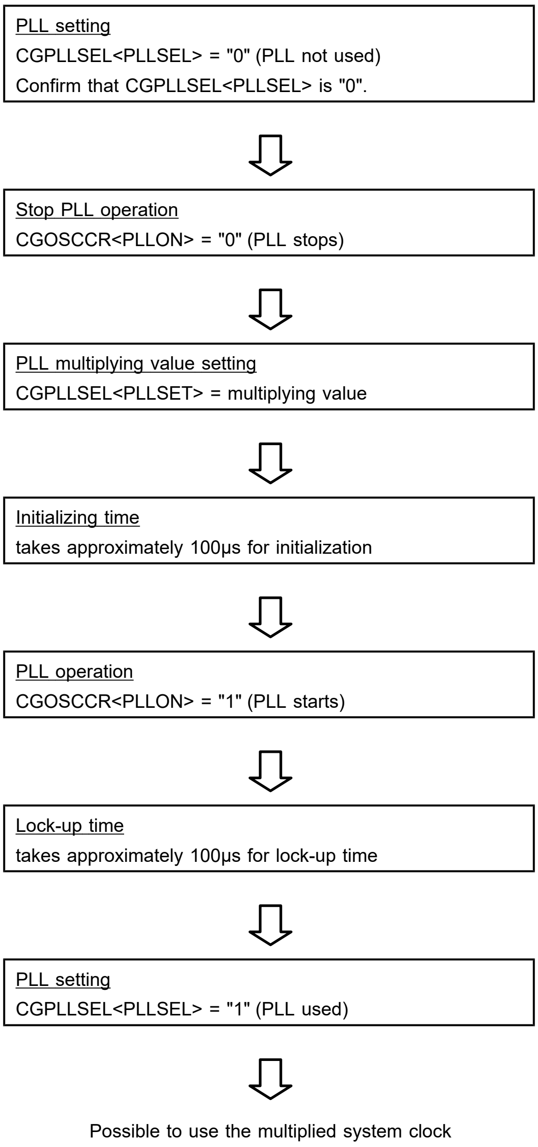
PLL setting
CGPLLSEL<PLLSEL> = "1" (PLL used)

Possible to use the multiplied system clock

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

5.3.5.4 Change PLL multiplying sequence

The sequence of PLL multiplying setting is shown below.



PLL setting
CGPLLSEL<PLLSEL> = "0" (PLL not used)
Confirm that CGPLLSEL<PLLSEL> is "0".

Stop PLL operation
CGOSCCR<PLLON> = "0" (PLL stops)

PLL multiplying value setting
CGPLLSEL<PLLSET> = multiplying value

Initializing time
takes approximately 100μs for initialization

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

Lock-up time
takes approximately 100μs for lock-up time

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

Possible to use the multiplied system clock

5.3.6 System clock

The internal high-speed oscillation clock and the external high-speed oscillation clocks which are an oscillator connecting or an inputting clock can be used as a source clock of the system clock.

Source clock		Frequency	Using PLL
Internal high-speed oscillation (IHOSC)		10MHz Note) (Target)	Not use, 3, 4, 5, 6, 8 or 10 multiplying
External high-speed oscillation	Oscillator (EHOSC)	8 to 16MHz	
	Input clock (EHCLKIN)	8 to 16MHz	

Note: The frequency of an internal high-speed oscillator must be adjusted by the internal high-speed oscillation adjustment function in order to keep fc equal or less then 80MHz when PLL is used.

The system clock can be divided by CGSYSCR<GEAR>. Although the setting can be changed while operating, the actual switching takes place after a slight delay.

Table 5-2 shows the example of the operation frequency by the setting of PLL and the clock gear.

Table 5-2 System clock (Unit : MHz, "-" : Reserved)

External oscillator (MHz)	External clock input (MHz)	PLL multiplying	Max. operation freq. (fc) (MHz)	USB PLL (48MHz) multiplying	ADC Max. operation freq. (fc/2) (MHz)	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
8	8	10	80	6	40	80	40	20	10	5	8	4	2	1	-
8	8	6	48	6	24	48	24	12	6	3	8	4	2	1	-
10	10	8	80	-	40	80	40	20	10	5	10	5	2.5	1.25	-
12	12	6	72	4	36	72	36	18	9	4.5	12	6	3	1.5	-
12	12	4	48	4	24	48	24	12	6	3	12	6	3	1.5	-
16	16	5	80	3	40	80	40	20	10	5	16	8	4	2	1
16	16	3	48	3	24	48	24	12	6	3	16	8	4	2	1

↑ initial value after reset

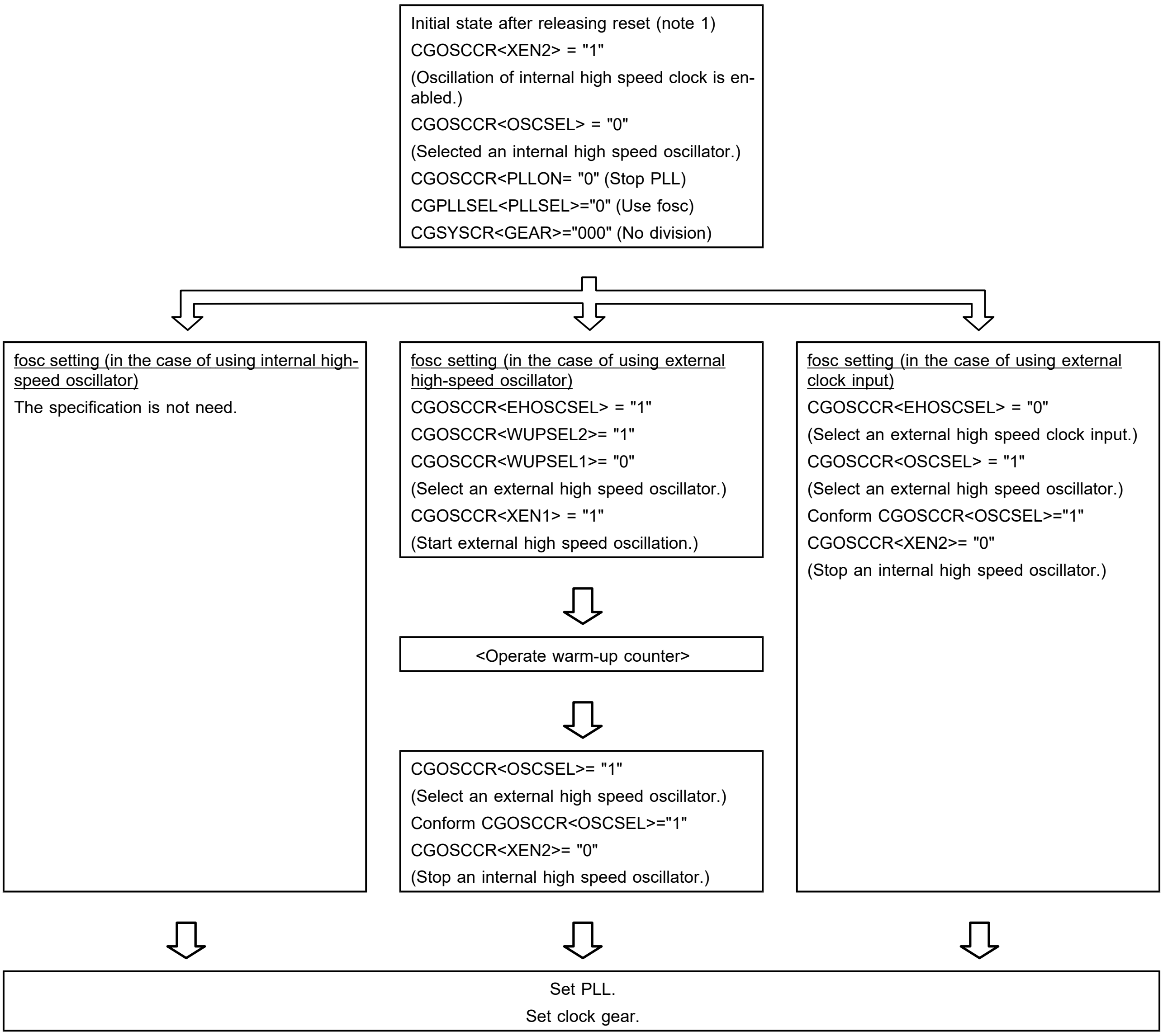
Note 1: The maximum operation frequency of an AD converter is 40 MHz. It is the value set to fc/2 by setup of ADCLK<ADCLK[2:0]>.

Note 2: Do not use 1/16 when SysTick is used.

5.3.6.1 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 (note 1)
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<EHOSCSEL> = "1"
CGOSCCR<WUPSEL2>= "1"
CGOSCCR<WUPSEL1>= "0"
(Select an 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.)
Conform CGOSCCR<OSCSEL>="1"
CGOSCCR<XEN2>= "0"
(Stop an internal high speed oscillator.)

fosc setting (in the case of using external clock input)
CGOSCCR<EHOSCSEL> = "0"
(Select an external high speed clock input.)
CGOSCCR<OSCSEL> = "1"
(Select an external high speed oscillator.)
Conform CGOSCCR<OSCSEL>="1"
CGOSCCR<XEN2>= "0"
(Stop an internal high speed oscillator.)

Set PLL.
Set clock gear.

5.3.7 Prescaler Clock Control

Peripheral I/O has a prescaler for dividing a clock. As the clock φT0 to be input to each prescaler, the "fperiph" clock specified in the CGSYSCR<FPSEL> can be divided according to the setting in the CGSYSCR<PRCK[2:0]>. After the controller is reset, fperiph/1 is selected as φT0.

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

5.3.8 System Clock Pin Output Function

TMPM367FDFG enables to output the system clock from a pin. The SCOUT pin can output low speed clock fs and the system clock fsys and fsys/2, and the prescaler input clock for peripheral I/O 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 shold be disabled when changing the clock gear.

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

Table 5-3 shows the pin status in each mode when the SCOUT pin is set to the SCOUT output.

Table 5-3 SCOUT Output Status in Each Mode

<table>
<tr><th rowspan="2">Mode
SCOUT selection
CGSYSCR</th><th rowspan="2">NORMAL</th><th colspan="2">Low power consumption mode</th></tr>
<tr><th>IDLE</th><th>STOP1/STOP2(Note)</th></tr>
<tr><td><SCOSEL[1:0]> = "00"</td><td colspan="3">Output the fs clock</td></tr>
<tr><td><SCOSEL[1:0]> = "01"</td><td colspan="2">Output the fsys/2 clock</td><td rowspan="3">Fixed to "0" or "1".</td></tr>
<tr><td><SCOSEL[1:0]> = "10"</td><td colspan="2">Output the fsys clock</td></tr>
<tr><td><SCOSEL[1:0]> = "11"</td><td>Output the φT0 clock</td><td></td></tr>
</table>

Note: When you change to the STOP2 mode, please set "1" as CGSTBYCR<PTKEEP> first and hold the state of a port.

5.4 Modes and Mode Transitions

5.4.1 Operation Mode Transitions

The IDLE and STOP1 modes can be used as the low power consumption mode that enables to reduce power consumption by halting processor core and some peripheral circuits operation.

And TMPM367FDFG has STOP2 mode that enables to reduce power consumption significantly by halting main voltage supply, retaining some peripheral circuits operations.

Figure 5-2 shows a mode transition diagram.

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

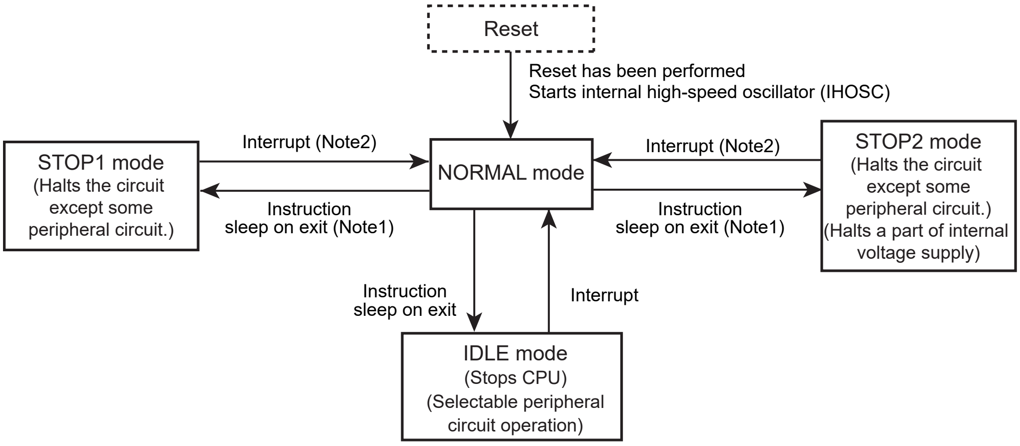

Reset
Reset has been performed
Starts internal high-speed oscillator (IHOSC)
STOP1 mode
(Halts the circuit except some peripheral circuit.)
Interrupt (Note2)
Instruction sleep on exit (Note1)
NORMAL mode
Interrupt (Note2)
Instruction sleep on exit (Note1)
STOP2 mode
(Halts the circuit except some peripheral circuit.)
(Halts a part of internal voltage supply)
Instruction sleep on exit
Interrupt
IDLE mode
(Stops CPU)
(Selectable peripheral circuit operation)

Figure 5-2 Mode Transition Diagram

Note 1: Returning from the STOP1/STOP2 mode, related bits <WUPSEL2>, <OSCSEL>, <XEN3>, <XEN2>, <XEN1>, <PLLON> of the register CGOSCCR and CGPLLSEL<PLLSEL> are initialized because of internal high-speed oscillator starts up.

Note 2: It branches to interrupt service routine of reset when returning from the STOP2 mode and it branches to interrupt service routine of interrupt factor when returning from the STOP1 mode.

Note 3: The warm-up is needed when returning from the STOP1/STOP2 mode.The warm-up time is needed to set in NORMAL mode, Regarding to warm-up time, refer to "5.6.9 Clock Operations in Mode Transition"

5.5 Operation mode

5.5.1 NORMAL mode

This mode is to operate the CPU core and the peripheral circuits by using the high-speed clock.

It is shifted to the NORMAL mode after reset.

5.6 Low Power Consumption Modes

The TMPM367FDFG has three low power consumption modes: IDLE, STOP1 and STOP2. To shift to the low power consumption mode, specify the mode in the system control register CGSTBYCR<STBY[2:0]> and execute the WFI (Wait For Interrupt) instruction. In this case, execute reset or generate the interrupt to release the mode. Releasing by the interrupt requires settings in advance. See the chapter "Exceptions" for details.

Note 1: The TMPM367FDFG does not offer any event for releasing the low power consumption mode. Transition to the low power consumption mode by executing the WFE (Wait For Event) instruction is prohibited.

Note 2: The TMPM367FDFG 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, STOP1, STOP2 mode are described as follows.

5.6.1 IDLE mode

Only the CPU is stopped in this mode. Each peripheral circuit has one bit in its control register for enabling or disabling operation in the IDLE mode. When the IDLE mode is entered, peripheral circuits for which operation in the IDLE mode is disabled stop operation and hold the state at that time.

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

- 16-bit timer/event counter (TMRB)
- 16-bit multi purpose timer (except PMD operation)
- Serial channel (SIO/UART)
- Serial bus interface (I2C/SIO)
- Analog Digital converter (ADC)
- Digital Analog converter (DAC)
- Watchdog timer (WDT)

Note 1: WDT should be stopped before entering IDLE mode.

Note 2: Please suspend the sauce clock to USBD before changing to IDLE mode. (USBPLLEN<USBDEN>="0").

5.6.2 STOP1 mode

Except some peripheral circuits, all the internal circuits including the internal oscillator are brought to a stop in STOP1 mode. When releasing STOP1 mode, an internal oscillator begins to operate and the operation mode changes to NORMAL mode.

The STOP1 mode enables to select the pin status by setting the CGSTBYCR<DRVE>. Table 5-4 shows the pin status in the STOP1 mode.

5.6.3 STOP2 mode

This mode halts voltage supply, retaining some peripheral circuits operation. This enables to reduce power consumption significantly compares to STOP1 mode.

After releasing the STOP2 mode, voltage is supplied to the halted voltage supply then an internal high-speed oscillator starts, and returns to NORMAL mode.

Before entering STOP2 mode, set CGSTBYCR<PTKEEP>="0"→"1" and keeps each port conditions. If internal voltage is halted, it can be held interface to the external IC, and STOP2 release source interrupt is available.

Note 1: Warming up is needed at the time of a return. It is necessary to set up a setup of warming up time in the mode (NORMAL mode) before going into STOP1 and the STOP2 mode. Please refer to 5.6.9.1 and 5.6.9.2 about warming up time.

Note 2: At the time of the return from STOP1 and the STOP2 mode, for internal high-speed oscillation starting Related bit CGPLLSEL<PLLSEL> and CGOSCCR<HWUPSEL>, <OSCSEL>, <XEN2>, <XEN1>, and <PLLON> are initialized, and CGOSCCR<WUDOR[11:0] > is not initialized.

Note 3: The STOP2 mode should secure the period for 50 µs or more from mode transition to release in order to perform internal electrical power source interception. If it cancels within a period, the internal electrical power source management cannot operate normally.

Table 5-4 Pin States in the STOP1/STOP2 mode

Function	Function name	I/O	STOP1		STOP2
			<DRVE> = 1	<DRVE> = 0	<PTKEEP> = 1
PORT	PAx to PLx	Input	Depend on PxIE[m]	Disable	keep state
		Output	Depend on PxCR[m]	Disable	keep state
Debug	$\overline{\text{TRST}}$, TCK, TMS, TDI, SWCLK, SWDI	Input	Depend on PxIE[m]		keep state
	TDO, SWDO, SWV, TRACECLK, TRACEDATA0/1/2/3	Output	Depend on PxCR[m] and enable when data is valid		keep state
Inrterrupt	INT0 to C	Input	Depend on PxIE[m]		keep state
SSP	SPxCLK, SPxFSS, SPxDO	Output	Depend on PxCR[m] and enable when data is valid	Disable	keep state
MPT(PMD mode)	UO, VO, WO, XO, YO, ZO	Output	Depend on PxCR[m] and enable when data is valid	Depend on PxCR[m] and enable when data is valid	keep state
MPT(IGBT mode)	MTOUTxx	Output	Depend on PxCR[m] and enable when data is valid	Depend on PxCR[m] and enable when data is valid	keep state
except above	Except above	Input	Depend on PxIE[m]	Disable	keep state
	Except above	Output	Depend on PxCR[m]	Disable	keep state

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

5.6.4 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 5-5 shows the mode setting in the <STBY[2:0]>.

Table 5-5 Low power consumption mode setting

Mode	CGSTBYCR <STBY[2:0]>
STOP1	001
IDLE	011
STOP2	101

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

5.6.5 Operational Status in Each Mode

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

Table 5-6 Operational Status in Each Mode

Block	NORMAL Internal high-speed oscillator use (IHOSC)	IDLE Internal high-speed oscillator use (IHOSC)	STOP1 (Note 1)	STOP2 (Note 1)
Processor core	o	−	−	×
DMAC	o	o	−	Δ×
I/O port	o	o	−(note 3)	−(note 4)
I/O port for USB	o	o	−	−
ADC	o	o	Δ	Δ×
DAC	o	o	Δ	−
USB Device	o	o	o	Δ×
SSP	o	o	Δ	Δ×
SIO/UART	o	o	Δ	Δ×
I2C/SIO	o	o	Δ	Δ×
WDT	o	o(note 6)	−	Δ×
TMRB	o	o	−	Δ×
MPT	o	o	−	Δ×
RMC	o	o	o	o
RTC	o	o	o	o
POR	o	o	o	o
LVD	o	o	o	o
External bus interface	o	o	−	×
CG	o	o	−	−
PLL	o	o	Δ	Δ×
OFD	o	o	Δ	Δ×
External high-speed oscillator (EHOSC)	o	o	Δ	−
External low-speed oscillator (ELOSC)	o	o	o	o
Internal high-speed oscillator 1 (IHOSC)	o	o	−	−
Internal high-speed oscillator 2	o	o	−	−
Backup RAM	o	o	o	o
Main RAM	o	o	o	×

o : Operation is available when in the target mode.

− : The clock to module stops automatically when transiting to the target mode.

Δ : Enables to select enabling or disabling module operation by software when in the target mode.

× :Voltage supply to module turns off automatically when transiting to the target mode.

Note 1: The peripheral function of "−", "Δ", and "×" is stopped, so please change in the STOP1/2 mode, before changing in the STOP1/2 mode. It is available to reduce leakage current by stopping reference voltage for AD converter or DA converter.

Note 2: After returning to NORMAL mode, the peripheral circuit which is cutted- down voltage supply in STOP2 mode should be initialized by software, Note 3: The status depends on the CGSTBYCR<DRVE>.

Note 4: The status depends on the CGSTBYCR<PTKEEP>. The state of port keeps the state when <PTKEEP> is set to "1".

Note 5: After reset or STOP1/STOP2 mode released, clock is provided from internal high-speed oscillator.

Note 6: Pay attention that the counter of watch dog timer function can not be cleared by CPU while in IDLE mode.

5.6.6 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 5-7.

Table 5-7 Release Source in Each Mode

Low power consumption mode			IDLE	STOP1	STOP2
Release source	Interrupt	INT0 to B (note 5)	o	o	o (note 4)
		INTC	o	×	×
		INTUSBDPON(INTD), INTUSBWKUP	o	o	×
		INTUSBD	o	×	×
		INTSSP0 to 2, INTSBI0 to 2	o	×	×
		INTRX0 to 3, INTTX0 to 3, INTUART0 to 1	o	×	×
		INTRTC, INTRMCRX	o	o	o
		INTTB0 to 7, INTCAP00 to 71	o	×	×
		INTMTTB00 to 31, INTMTCAP00 to 31, INTMTEMG0 to 3	o	×	×
		INTPMD, INTEMG, INTENC	o	×	×
		INTADA, INTADAHP, INTADAM0 to 1 (ADC unit A)	o	×	×
		INTADB, INTADBHP, INTADBM0 to 1 (ADC unit B)	o	×	×
		INTDMAAERR, INTDMABERR, µDMAC transfer request Note5)	o	×	×
	SysTick interrupt		o	×	×
	Non-Maskable Interrupt (INTWDT)		o	×	×
	Non-Maskable Interrupt ($\overline{\text{NMI}}$ pin)		o	o	o
	Non-Maskable Interrupt (INTLVD)		o	o	o
	RESET (WDT)		o	×	×
	RESET (POR)		o	o	o
	RESET (LVD)		o	o	o
	RESET (ODF)		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: Regarding to the warm-up time which is need to return from each mode, refer to "5.6.8 Warm-up".

Note 2: After STOP2 mode is released, reset operation initializes the internal supply voltage cut off peripheral circuit (Refer to Table 5-6). But back-up module is not initialized.

Note 3: For shifting to the low power consumption mode, set the CPU to prohibit all the interrupts other than the release source. If not, releasing may be executed by an unspecified interrupt.

Note 4: When releasing from IDLE,STOP1/2 mode by interrupting level mode, hold the level until the interrupt handling starts. If the level is changed before that, the correct interrupt handling cannot be started.

Note 5: TMPM367FDFG has following request factor.

INTDMAADA, INTDMAADB, INTDMADAA, INTDMADAB, INTDMASPR0, INTDMASPT0, INTDMASPR1, INTDMASPT1, INTDMASPR2, INTDMASPT2, INTDMAUTR0, INTDMAUTT0, INTDMAUTR1, INTDMAUTT1, INTDMARX0, INTDMATX0, INTDMARX1, INTDMATX1, INTDMARX2, INTDMATX2, INTDMARX3, INTDMATX3, INTDMASBI1, INTDMASBI2, INTDMATB, INTDMARQ

- 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 STOP1 and STOP2 modes, refer to "Exceptions".

- Release by Non-Maskable Interrupt (NMI)

  There are two kinds of NMI sources: WDT interrupt (INTWDT), LVD interrupt (INTLVD) and $\overline{\text{NMI}}$ pin. INTWDT can only be used in the IDLE mode.

  INTLVD and the $\overline{\text{NMI}}$ pin can be used to release all the lower power consumption modes.

- Release by reset

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

  IDLE mode can be released by reset from WDT or OFD.

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

- Release by SysTick interrupt

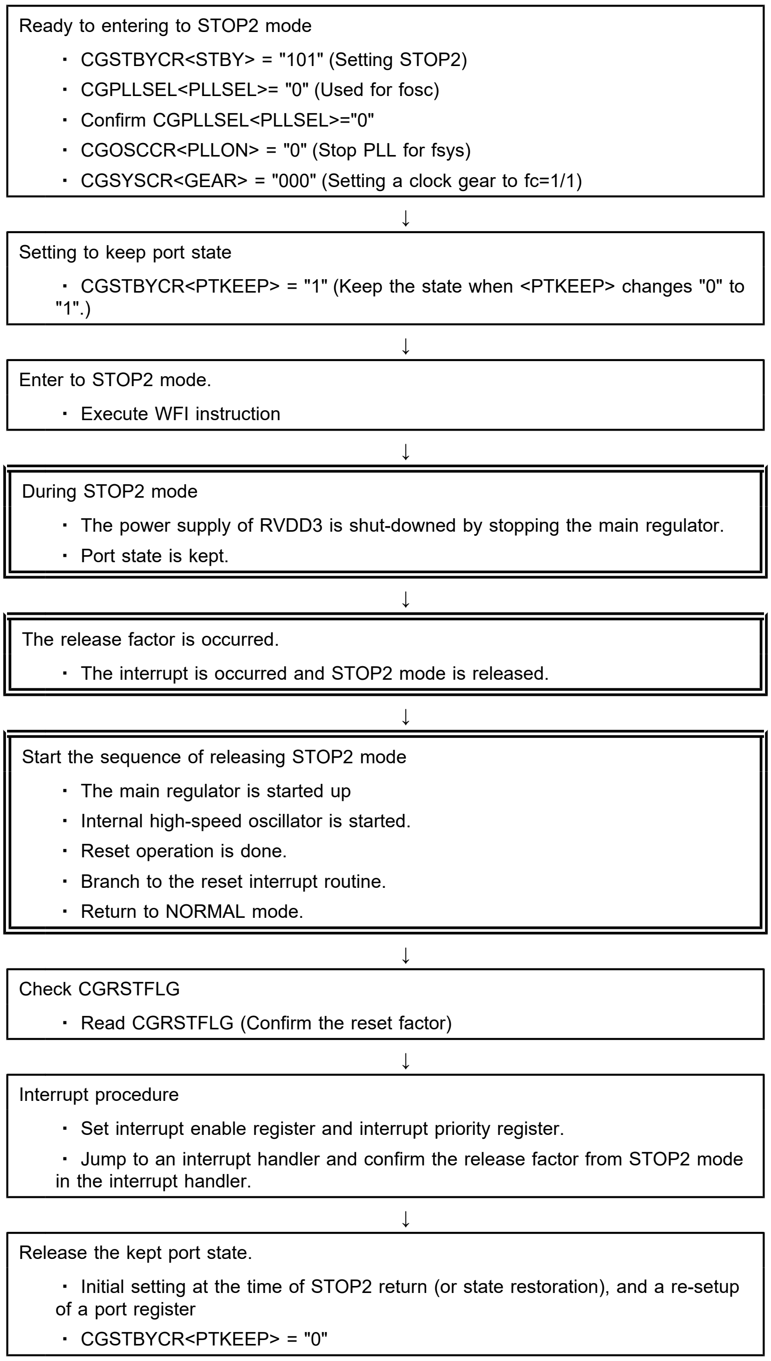
  SysTick interrupt can only be used in the IDLE mode.

Refer to "Exceptions" for details.

5.6.7 Flow chart of entering and releasing to or from STOP2 mode

The flow chart of entering and releasing to or from STOP2 mode is shown bellows.

☐ indicates hardware handling, ⧈ indicates software handling.



Ready to entering to STOP2 mode

- CGSTBYCR<STBY> = "101" (Setting STOP2)
- CGPLLSEL<PLLSEL>= "0" (Used for fosc)
- Confirm CGPLLSEL<PLLSEL>="0"
- CGOSCCR<PLLON> = "0" (Stop PLL for fsys)
- CGSYSCR<GEAR> = "000" (Setting a clock gear to fc=1/1)

↓

Setting to keep port state

- CGSTBYCR<PTKEEP> = "1" (Keep the state when <PTKEEP> changes "0" to "1".)

↓

Enter to STOP2 mode.

- Execute WFI instruction

↓

During STOP2 mode

- The power supply of RVDD3 is shut-downed by stopping the main regulator.
- Port state is kept.

↓

The release factor is occurred.

- The interrupt is occurred and STOP2 mode is released.

↓

Start the sequence of releasing STOP2 mode

- The main regulator is started up
- Internal high-speed oscillator is started.
- Reset operation is done.
- Branch to the reset interrupt routine.
- Return to NORMAL mode.

↓

Check CGRSTFLG

- Read CGRSTFLG (Confirm the reset factor)

↓

Interrupt procedure

- Set interrupt enable register and interrupt priority register.
- Jump to an interrupt handler and confirm the release factor from STOP2 mode in the interrupt handler.

↓

Release the kept port state.

- Initial setting at the time of STOP2 return (or state restoration), and a re-setup of a port register
- CGSTBYCR<PTKEEP> = "0"

5.6.8 Warm-up

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

In the mode transition from STOP1/STOP2 to the NORMAL, an internal high-speed oscillator is activated automatically. And an internal high-speed oscillator is selected as the source clock of warm-up counter, warm-up counter is activated automatically.

Then the system clock output is started after the elapse of warm-up time. It is necessary to set a warm-up time in the CGOSCCR<WUPT[11:0]> before executing the instruction to enter the STOP1/STOP2 mode. Regarding to warm-up time, refer to "5.6.9 Clock Operations in Mode Transition".

Note: Returning from the STOP1/STOP2 mode, related bits <WUPSEL2>, <XEN3>, <OSCSEL>, <XEN2>, <XEN1>, <PLLON> of the register CGOSCCR and CGPLLSEL<PLLSEL> are initialized because of internal high-speed oscillator starts up.

Table 5-8 shows whether the warm-up setting of each mode transition is required or not.

Table 5-8 Warm-up setting in mode transition

Mode transition	Warm-up setting
NORMAL → IDLE	Not required
NORMAL → STOP1	Not required
NORMAL → STOP2	Not required
IDLE → NORMAL	Not required
STOP1 → NORMAL	Auto-warm-up (Note)
STOP2 → NORMAL	Auto-warm-up (Note)

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

5.6.9 Clock Operations in Mode Transition

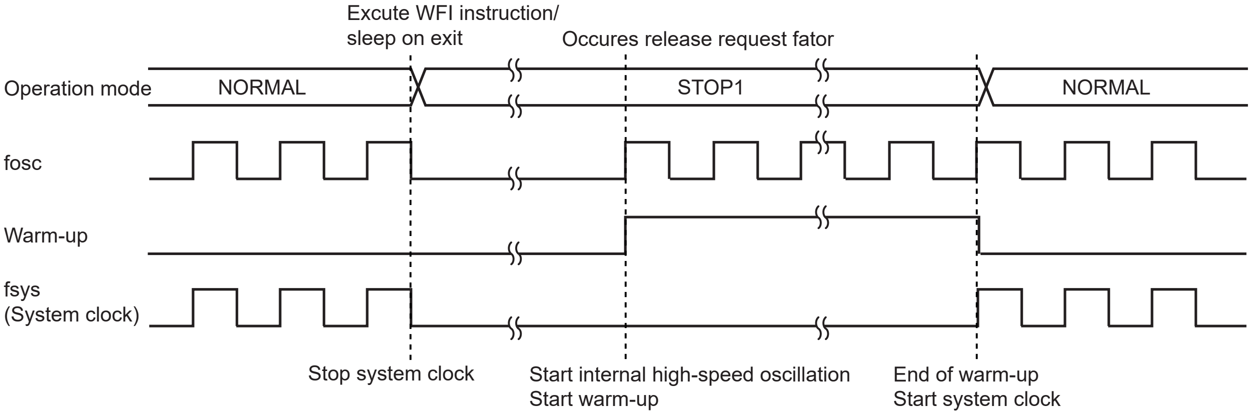
The clock operations in mode transition are described as follows.

5.6.9.1 Transition of operation modes: NORMAL → STOP1 → NORMAL

When returning to the NORMAL mode from the STOP1 mode, the warm-up is activated automatically.

It is necessary for the warm-up to be set CGOSCCR<WUPT[11:0]>=0x03f in this case as a stability time (equal or more than approximate 100μs) of internal circuits before entering the STOP1 mode.

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


Excute WFI instruction/
sleep on exit
Occures release request fator
Operation mode
NORMAL
STOP1
NORMAL
fosc
Warm-up
fsys
(System clock)
Stop system clock
Start internal high-speed oscillation
Start warm-up
End of warm-up
Start system clock

5.6.9.2 Transition of operation modes: NORMAL → STOP2 → NORMAL

When returning to the NORMAL mode from the STOP2 mode, the warm-up is activated automatically.

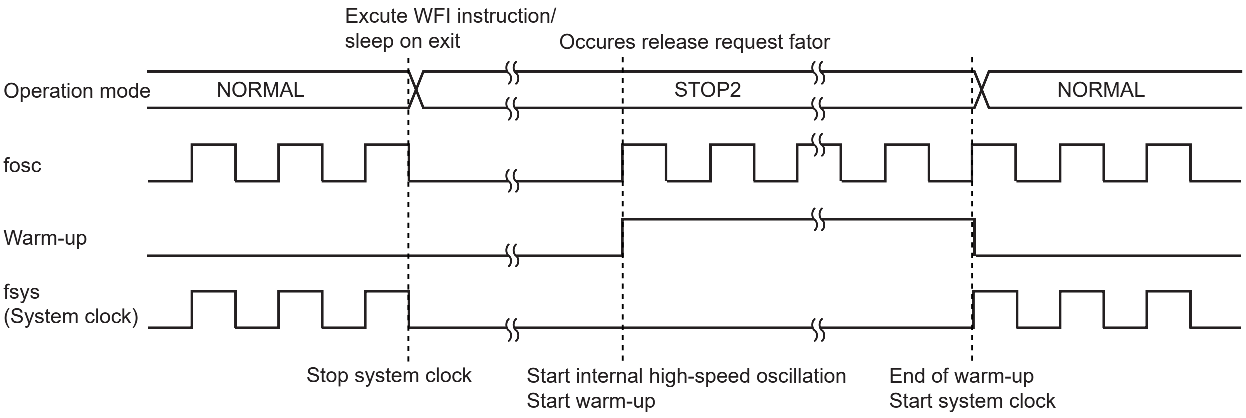
It is necessary for the warm-up to be set CGOSCCR<WUPT[11:0]>=0x271 in this case as a stability time (equal or more than approximate 1ms) of internal circuits before entering the STOP2 mode.

After STOP2 mode is released, reset operation initializes the internal supply voltage cut off peripheral circuits. But the peripheral circuits which are cut the connection with the internal supply voltage are not initialized.

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

Even returning to the NORMAL mode by without reset, it would be branched to interrupt service routine of reset.

Note: When releasing STOP2 by external interrupt pin, set <PTKEE> to "1" before entering STOP2 mode.


Excute WFI instruction/
sleep on exit
Occures release request fator
Operation mode
NORMAL
STOP2
NORMAL
fosc
Warm-up
fsys
(System clock)
Stop system clock
Start internal high-speed oscillation
Start warm-up
End of warm-up
Start system clock

6. Reset Operation

The following are sources of reset operation.

- Power-on-reset circuit (POR)
- Low Voltage Detection Circuit (LVD)
- 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 1: Once reset operation is done, internal RAM data is not assured.

6.1 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. TMPM367FDFG has a function to insert a stable time automatically.

6.1.1 Reset by power-on-reset 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 0.8ms internal reset signal is released.

Power-on-reset circuit operation is referred to Section of "Power-on-reset circuit (POR)".

Note: If power setup time is expected to exceed internal reset time, use a $\overline{\text{RESET}}$ pin.

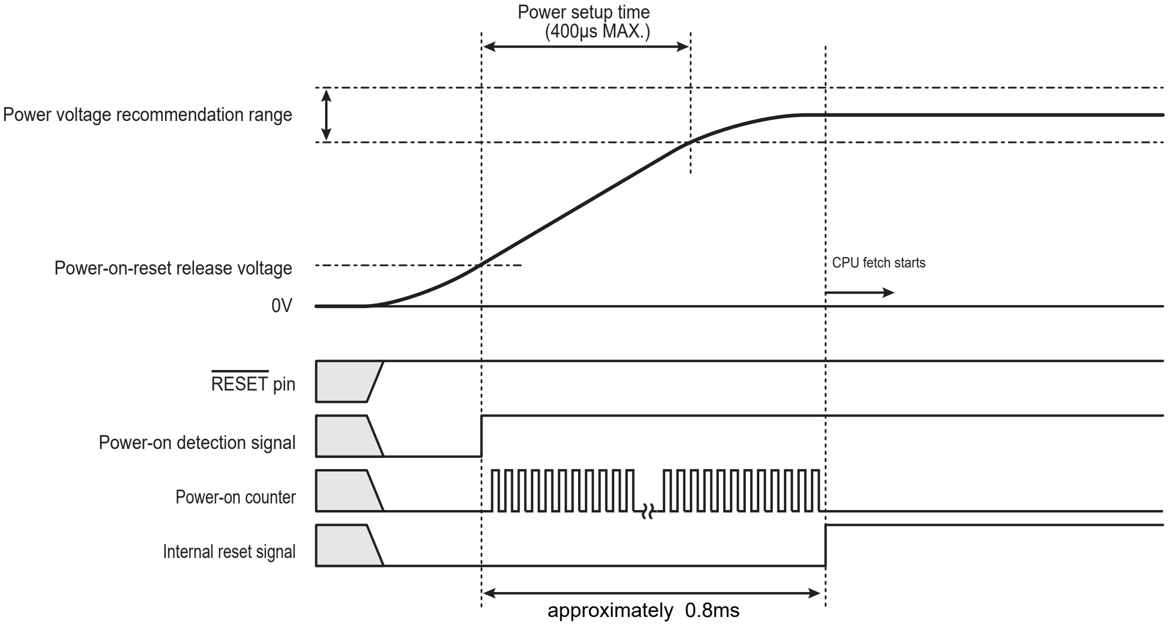

Power setup time
(400µs MAX.)
Power voltage recommendation range
Power-on-reset release voltage
0V
CPU fetch starts
RESET pin
Power-on detection signal
Power-on counter
Internal reset signal
approximately 0.8ms

Figure 6-1 Reset Operation by Power-on Circuit

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

Internal reset signal is released approximately 0.8ms after $\overline{\text{RESET}}$ pin becomes "High". However if $\overline{\text{RESET}}$ pin is set to "High" within 400μs after power-on reset signal becomes "High", the reset process will be the same as the power-on described in 6.1.1.

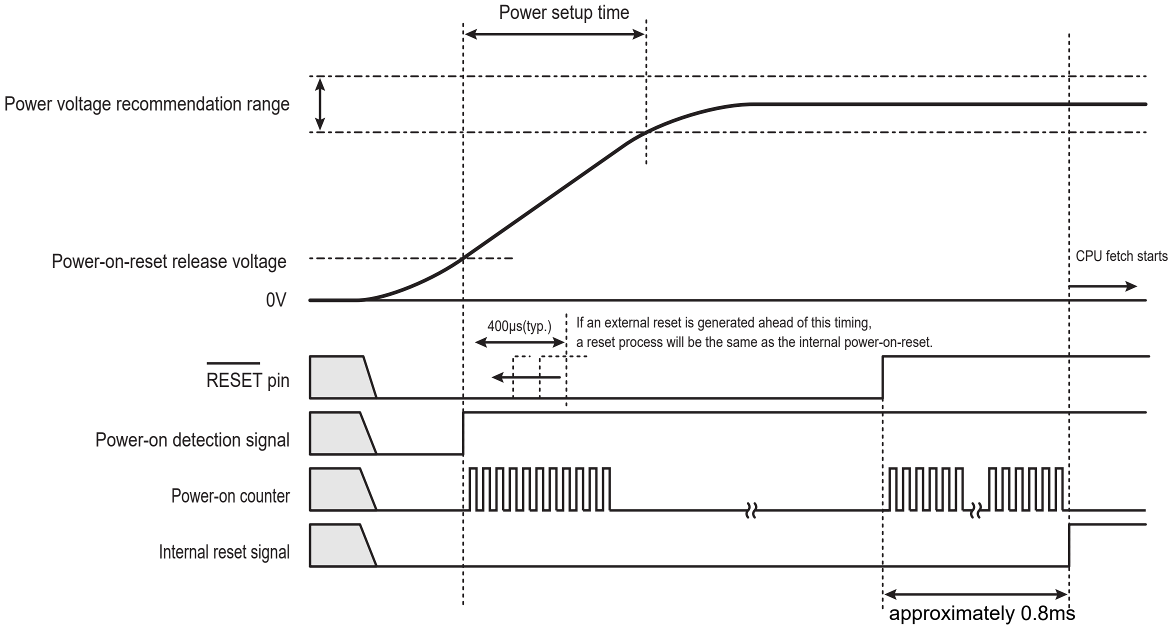

Power setup time
Power voltage recommendation range
Power-on-reset release voltage
0V
CPU fetch starts
400μs(typ.)
If an external reset is generated ahead of this timing,
a reset process will be the same as the internal power-on-reset.
RESET pin
Power-on detection signal
Power-on counter
Internal reset signal
approximately 0.8ms

Figure 6-2 Reset Operation by $\overline{\text{RESET}}$ pin

6.2 Warm-up

6.2.1 Reset Duration

To do reset TMPM367FDFG, 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 frequecy oscillator. Approximately 0.8ms after $\overline{\text{RESET}}$ pin becomes "High", internal reset will be released.

6.3 After reset

After reset, the control register of Cortex-M3 and the peripheral function control register (SFR) are 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.

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.

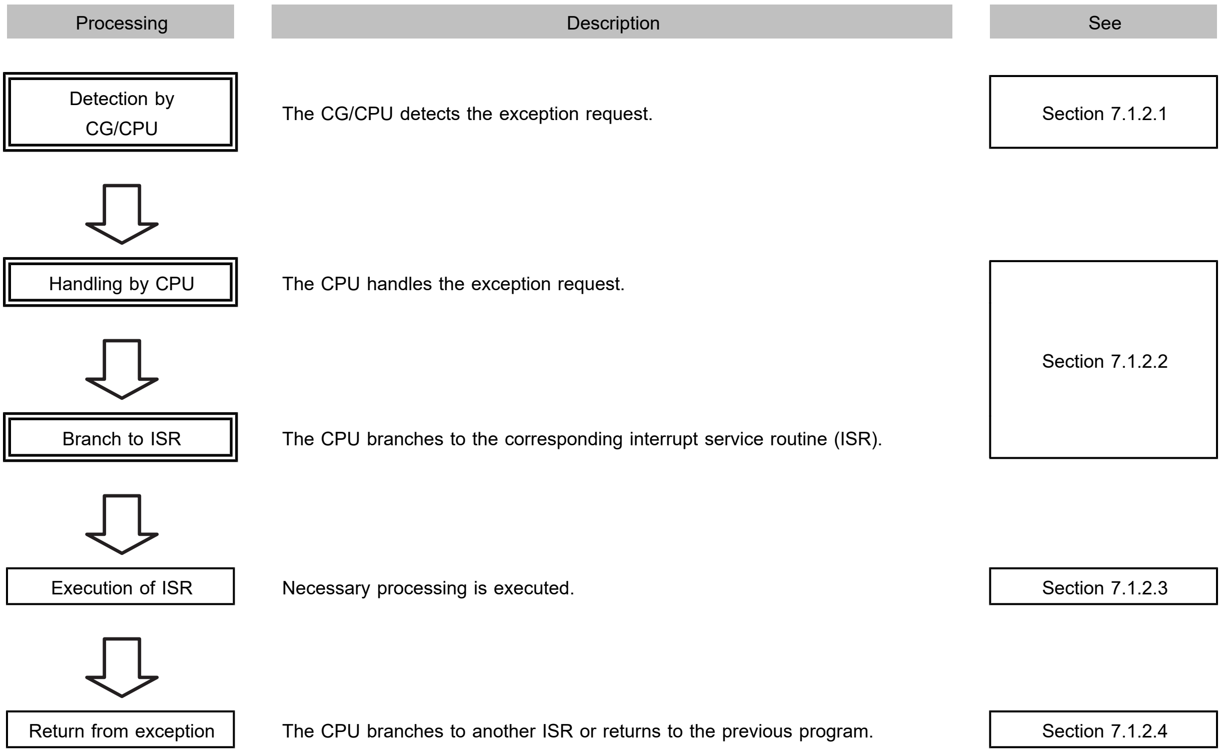
For detailed descriptions on each exception, refer to "Cortex-M3 Technical Reference Manual".

- 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, POR, LVD, OFD, WDT or SYSRETREQ
2	Non-Maskable Interrupt	−2	$\overline{\text{NMI}}$ pin, WDT or LVD
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 to 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
up to 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 sub-priorities
	Pre-emption field	Sub-priority 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)

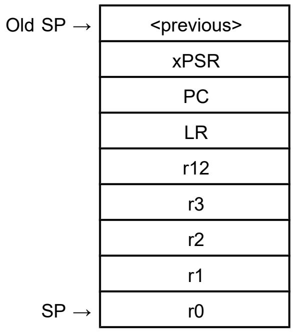
When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called "pre-emption".

(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 to 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 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 POR

  The power on reset (POR) has a reset generating feature. For details. see the chapter on the POR.

- Reset exception by LVD

  The low voltage detection circuit (LVD) has a reset generating feature. For details, see the chapter on the LVD.

- Reset exception by OFD

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

- 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.

7.3 Non-Maskable Interrupts (NMI)

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

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

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

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

- Non-maskable interrupt by WDT

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

- Non-maskable interrupt by LVD

  The LVD has a non-maskable interrupt generating feature. For details, see the chapter on the LVD.

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, fosc which is selected by CGOSCCR <OSCSEL> <EHOSCSEL> by 32 is used as external reference clock.**

7.5 Interrupts

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

The CPU is notified of interrupt requests by the interrupt signal from each interrupt source.

It sets priority on interrupts and handles an interrupt request with the highest priority.

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

7.5.1 Interrupt Sources

7.5.1.1 Interrupt Route

Figure 7-1 shows an interrupt request route.

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

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

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

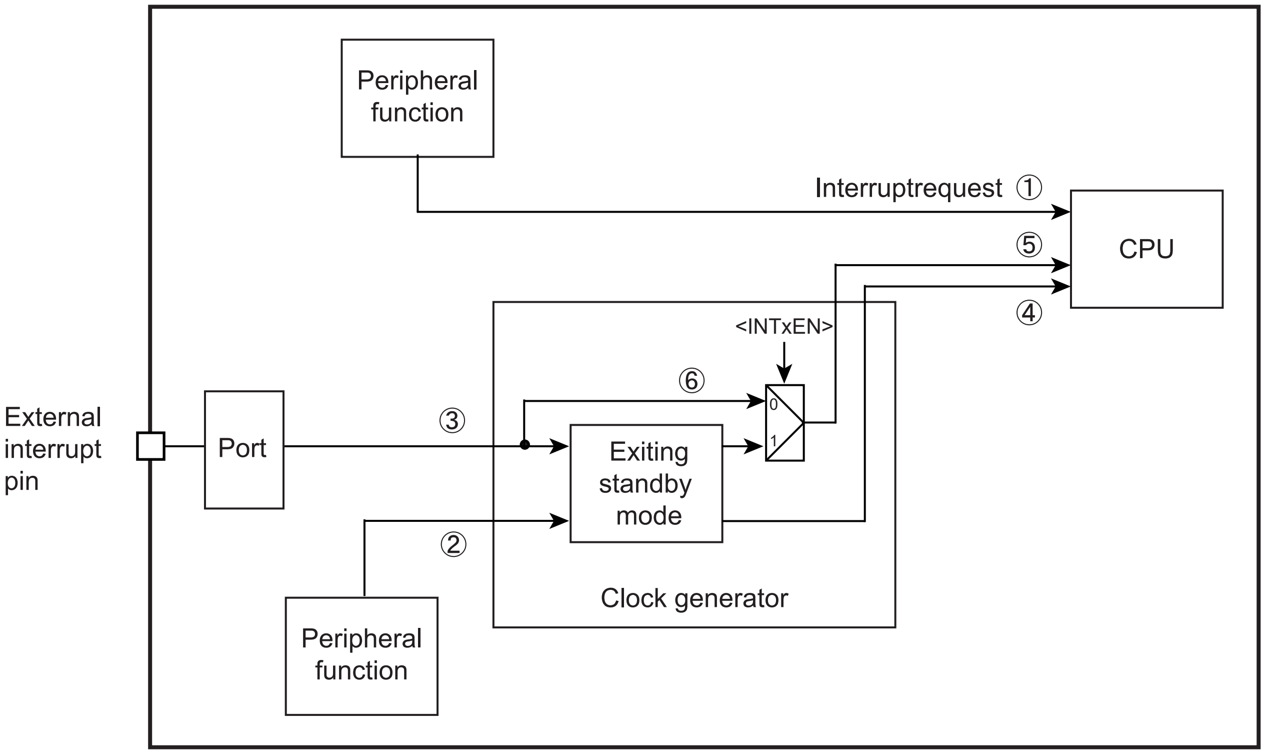

Peripheral function
Interruptrequest ①
CPU
⑤
④
<INTxEN>
⑥
0
1
External interrupt pin
Port
③
Exiting standby mode
②
Clock generator
Peripheral function

Figure 7-1 Interrupt Route

7.5.1.2 Generation

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

- From external pin

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

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

  See the chapter of each peripheral function for details.
- By setting Interrupt Set-Pending Register (forced pending)

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

7.5.1.3 Transmission

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

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

7.5.1.4 Precautions when using external interrupt pins

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

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

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

7.5.1.5 List of Interrupt Sources

Table 7-3 shows the list of interrupt sources.

Table 7-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
0	INT0	Interrupt pin 0	Selectable	CGIMCGA
1	INT1	Interrupt pin 1		
2	INT2	Interrupt pin 2		
3	INT3	Interrupt pin 3		
4	INT4	Interrupt pin 4		CGIMCGB
5	INT5	Interrupt pin 5		
6	INT6	Interrupt pin 6		
7	INT7	Interrupt pin 7		
8	INT8	Interrupt pin 8		CGIMCGC
9	INT9	Interrupt pin 9		
10	INTA	Interrupt pin A		
11	INTB	Interrupt pin B		
12	INTC	Interrupt pin C		
13	INTUSBDPON(INTD)	USB Device bus power (Interrupt pin D)	Selectable	CGIMCGD
14	Reserved	-		
15	Reserved	-		
16	INTRX0	Serial reception (channel 0)		
17	INTTX0	Serial transmission (channel 0)		
18	INTRX1	Serial reception (channel 1)		
19	INTTX1	Serial transmission (channel 1)		
20	INTRX2	Serial reception (channel 2)		
21	INTTX2	Serial transmission (channel 2)		
22	INTRX3	Serial reception (channel 3)		
23	INTTX3	Serial transmission (channel 3)		
24	INTUART0	UART interrupt (channel 0)		
25	INTUART1	UART interrupt (channel 1)		
26	INTSBI0	Serial bus interface (channel 0)		
27	INTSBI1	Serial bus interface (channel 1)		
28	INTSBI2	Serial bus interface (channel 2)		
29	INTSSP0	SPI serial interface (channel 0)		
30	INTSSP1	SPI serial interface (channel 1)		
31	INTSSP2	SPI serial interface (channel 2)		
32	Reserved	-		
33	INTUSBD	USB device interrupt		
34	INTUSBWKUP	USB device walk-up interrupt	Selectable	CGIMCGD
35	Reserved	-		
36	Reserved	-		
37	Reserved	-		
38	Reserved	-		
39	Reserved	-		
40	INTADAHP	Highest priority AD conversion complete interrupt (unit A)		
41	INTADAM0	AD conversion monitoring function interrupt 0 (unit A)		

Table 7-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
42	INTADAM1	AD conversion monitoring function interrupt 1 (unit A)		
43	INTADA	AD conversion completion interrupt (unit A)		
44	INTADBHP	Highest priority AD conversion complete interrupt (unit B)		
45	INTADBM0	AD conversion monitoring function interrupt 0 (unit B)		
46	INTADBM1	AD conversion monitoring function interrupt 1 (unit B)		
47	INTADB	AD conversion completion interrupt (unit B)		
48	INTEMG	PMD EMG interrupt		
49	INTPMD	PMD PWM interrupt		
50	INTENC	Encoder input interrupt		
51	Reserved	-		
52	Reserved	-		
53	Reserved	-		
54	INTMTEMG0	MPT EMG interrupt (channel 0)		
55	INTMTTB00	MPT compare match 0 / overflow, IGBT cycle interrupt (channel 0)		
56	INTMTTB01	MPT compare match 1, IGBT trigger interrupt (channel 0)		
57	INTMTCAP00	MPT input capture 0 (channel 0)		
58	INTMTCAP01	MPT input capture 1 (channel 0)		
59	INTMTEMG1	MPT EMG interrupt (channel 1)		
60	INTMTTB10	MPT compare match 0 / overflow, IGBT cycle interrupt (channel 1)		
61	INTMTTB11	MPT compare match 1, IGBT trigger interrupt (channel 1)		
62	INTMTCAP10	MPT input capture 0 (channel 1)		
63	INTMTCAP11	MPT input capture 1 (channel 1)		
64	INTMTEMG2	MPT EMG interrupt (channel 2)		
65	INTMTTB20	MPT compare match 0 / overflow, IGBT cycle interrupt (channel 2)		
66	INTMTTB21	MPT compare match 1, IGBT trigger interrupt (channel 2)		
67	INTMTCAP20	MPT input capture 0 (channel 2)		
68	INTMTCAP21	MPT input capture 1 (channel 2)		
69	INTMTEMG3	MPT EMG interrupt (channel 3)		
70	INTMTTB30	MPT compare match 0 / overflow, IGBT cycle interrupt (channel 3)		
71	INTMTTB31	MPT compare match 1, IGBT trigger interrupt (channel 3)		
72	INTMTCAP30	MPT input capture 0 (channel 3)		
73	INTMTCAP31	MPT input capture 1 (channel 3)		
74	INTRMCRX	remote control reception	Rising edge	CGIMCGD
75	INTTB0	16-bit TMRB compare match 0 / 1 / overflow (channel 0)		
76	INTCAP00	16-bit TMRB input capture 0 (channel 0)		

Table 7-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
77	INTCAP01	16-bit TMRB input capture 1 (channel 0)		
78	INTTB1	16-bit TMRB compare match 0 / 1 / overflow (channel 1)		
79	INTCAP10	16-bit TMRB input capture 0 (channel 1)		
80	INTCAP11	16-bit TMRB input capture 1 (channel 1)		
81	INTTB2	16-bit TMRB compare match 0 / 1 / overflow (channel 2)		
82	INTCAP20	16-bit TMRB input capture 0 (channel 2)		
83	INTCAP21	16-bit TMRB input capture 1 (channel 2)		
84	INTTB3	16-bit TMRB compare match 0 / 1 / overflow (channel 3)		
85	Reserved	-		
86	Reserved	-		
87	INTTB4	16-bit TMRB compare match 0 / 1 / overflow (channel 4)		
88	Reserved	-		
89	Reserved	-		
90	INTTB5	16-bit TMRB compare match 0 / 1 / overflow (channel 5)		
91	INTCAP50	16-bit TMRB input capture 0 (channel 5)		
92	INTCAP51	16-bit TMRB input capture 1 (channel 5)		
93	INTTB6	16-bit TMRB compare match 0 / 1 / overflow (channel 6)		
94	Reserved	-		
95	Reserved	-		
96	INTTB7	16-bit TMRB compare match 0 / 1 / overflow (channel 7)		
97	Reserved	-		
98	Reserved	-		
99	INTRTC	RTC	Falling edge	CGIMCGD
100	INTDMAADA	DMAC ADC conversion completion (unit A)		
101	INTDMAADB	DMAC ADC conversion completion (unit B)		
102	INTDMADAA	DMAC ADC conversion trigger (unit A)		
103	INTDMADAB	DMAC ADC conversion trigger (unit B)		
104	INTDMASPR0	DMAC SSP reception (channel 0) / DMAC I2C/SIO (channel 0)		
105	INTDMASPT0	DMAC SSP transmission (channel 0)		
106	INTDMASPR1	DMAC SSP reception (channel 1)		
107	INTDMASPT1	DMAC SSP transmission (channel 1)		
108	INTDMASPR2	DMAC SSP reception (channel 2)		
109	INTDMASPT2	DMAC SSP transmission (channel 2)		
110	INTDMAUTR0	DMAC UART reception (channel 0)		
111	INTDMAUTT0	DMAC UART transmission (channel 0)		
112	INTDMAUTR1	DMAC UART reception (channel 1)		
113	INTDMAUTT1	DMAC UART transmission (channel 1)		
114	INTDMARX0	DMAC SIO / UART reception (channel 0)		
115	INTDMATX0	DMAC SIO / UART transmission (channel 0)		

Table 7-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
116	INTDMARX1	DMAC SIO / UART reception (channel 1)		
117	INTDMATX1	DMAC SIO / UART transmission (channel 1)		
118	INTDMARX2	DMAC SIO / UART reception (channel 2)		
119	INTDMATX2	DMAC SIO / UART transmission (channel 2)		
120	INTDMARX3	DMAC SIO / UART reception (channel 3)		
121	INTDMATX3	DMAC SIO / UART transmission (channel 3)		
122	INTSBI1	DMAC I2C / SIO (channel 1)		
123	INTSBI2	DMAC I2C / SIO (channel 2)		
124	INTDMATB	DMAC TMRB compare match 0 / 1 / overflow (channel 0 to 4)		
125	INTDMARQ	DMAC request pin		
126	INTDMAAERR	DMAC transmission error interrupt (unit A)		
127	INTDMABERR	DMAC transmission error interrupt (unit B)		

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.

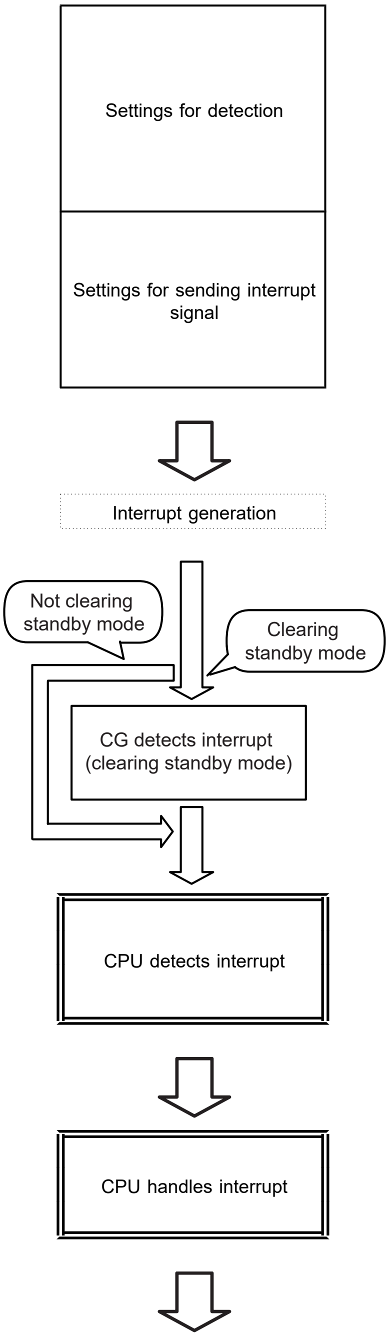
An interrupt request detected by the clock generator is notified to the CPU with a signal in "High" level.

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. Pre configuration (1) (Interrupt from external pin)
4. Pre configuration (2) (Interrupt from peripheral function)
5. Pre configuration (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) Pre configuration (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) Pre configuration (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) Pre configuration (3) (Interrupt Set-Pending Register)

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

NVIC register		
Interrupt Set-Pending [m]	←	"1"

Note: **m: corresponding bit**

(6) Configuring the clock generator

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

Before enabling an interrupt, clear the corresponding interrupt request already held. This can avoid unexpected interrupt.To clear corresponding interrupt request, write a value corresponding to the interrupt to be used to the CGICRCG register.See "7.6.3.5 CGICRCG(CG Interrupt Request Clear Register)" for each value.

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

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

Note: **n: register number / m: number assigned to interrupt source**

(7) Enabling interrupt by CPU

Enable the interrupt by the CPU as shown below.

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

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

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

At the end, PRIMASK register is zero cleared.

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

Note 1: **m : corresponding bit**

Note 2: **PRIMASK register cannot be modified by the user access level.**

7.5.2.3 Detection by Clock Generator

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

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

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

7.5.2.4 Detection by CPU

The CPU detects an interrupt request with the highest priority.

7.5.2.5 CPU processing

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

7.5.2.6 Interrupt Service Routine (ISR)

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

(1) Pushing during ISR

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

Push the contents of other registers if needed.

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

(2) Clearing an interrupt source

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

If an interrupt 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.

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.

7.6 Exception/Interrupt-Related Registers

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

7.6.1 Register List

NVIC registers Base Address = 0xE000_E000

Register name	Address
SysTick Control and Status Register	0x0010
SysTick Reload Value Register	0x0014
SysTick Current Value Register	0x0018
SysTick Calibration Value Register	0x001C
Interrupt Set-Enable Register 1	0x0100
Interrupt Set-Enable Register 2	0x0104
Interrupt Set-Enable Register 3	0x0108
Interrupt Set-Enable Register 4	0x010C
Interrupt Clear-Enable Register 1	0x0180
Interrupt Clear-Enable Register 2	0x0184
Interrupt Clear-Enable Register 3	0x0188
Interrupt Clear-Enable Register 4	0x018C
Interrupt Set-Pending Register 1	0x0200
Interrupt Set-Pending Register 2	0x0204
Interrupt Set-Pending Register 3	0x0208
Interrupt Set-Pending Register 4	0x020C
Interrupt Clear-Pending Register 1	0x0280
Interrupt Clear-Pending Register 2	0x0284
Interrupt Clear-Pending Register 3	0x0288
Interrupt Clear-Pending Register 4	0x028C
Interrupt Priority Register	0x0400 to 0x047F
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 = 0x400F_3000

Register name		Address
CG Interrupt Mode Control Register A	CGIMCGA	0x0040
CG Interrupt Mode Control Register B	CGIMCGB	0x0044
CG Interrupt Mode Control Register C	CGIMCGC	0x0048
CG Interrupt Mode Control Register D	CGIMCGD	0x004C
CG Interrupt Request Clear Register	CGICRCG	0x0060
Reset Flag Register	CGRSTFLG	0x0064
NMI Flag Register	CGNMIFLG	0x0068

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, fosc which is selected by CGOSCCR <OSCSEL> <EHOSCSEL> by 32 is used as external reference clock.

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

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 (0xC35) by external reference clock. (Note)

Note: **In the case of multishot, please use <TENMS>-1.**

7.6.2.5 Interrupt Set-Enable Register 1

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

Bit	Bit Symbol	Type	Function
31-16	SETENA	R/W	Interrupt number [31:16] [Write] 1: Enable [Read] 0: Disabled 1: Enabled Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.
15-14	-	R/W	Write as "0".
13-0	SETENA	R/W	Interrupt number [13:0] [Write] 1: Enable [Read] 0: Disabled 1: Enabled Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

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

7.6.2.6 Interrupt Set-Enable Register 2

	31	30	29	28	27	26	25	24
bit symbol	SETENA (Interrupt 63)	SETENA (Interrupt 62)	SETENA (Interrupt 61)	SETENA (Interrupt 60)	SETENA (Interrupt 59)	SETENA (Interrupt 58)	SETENA (Interrupt 57)	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 55)	SETENA (Interrupt 54)	-	-	-	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 45)	SETENA (Interrupt 44)	SETENA (Interrupt 43)	SETENA (Interrupt 42)	SETENA (Interrupt 41)	SETENA (Interrupt 40)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	SETENA (Interrupt 34)	SETENA (Interrupt 33)	SETENA (Interrupt 32)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-22	SETENA	R/W	Interrupt number [63:54] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.
21-19	-	R/W	Write as "0".
18-8	SETENA	R/W	Interrupt number [50:40] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.
7-3	-	R/W	Write as "0".
2-0	SETENA	R/W	Interrupt number [34:32] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

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

7.6.2.7 Interrupt Set-Enable Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	-	SETENA (Interrupt 93)	SETENA (Interrupt 92)	SETENA (Interrupt 91)	SETENA (Interrupt 90)	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	SETENA (Interrupt 87)	-	-	SETENA (Interrupt 84)	SETENA (Interrupt 83)	SETENA (Interrupt 82)	SETENA (Interrupt 81)	SETENA (Interrupt 80)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SETENA (Interrupt 79)	SETENA (Interrupt 78)	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)	SETENA (Interrupt 65)	SETENA (Interrupt 64)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-30	-	R/W	Write as "0".
29-26	SETENA	R/W	Interrupt number [93:90] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.
25-24	-	R/W	Write as "0".
23	SETENA	R/W	Interrupt number [87] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.
22-21	-	R/W	Write as "0".
20-0	SETENA	R/W	Interrupt number [84:64] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

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

7.6.2.8 Interrupt Set-Enable Register 4

	31	30	29	28	27	26	25	24
bit symbol	SETENA (Interrupt 127)	SETENA (Interrupt 126)	SETENA (Interrupt 125)	SETENA (Interrupt 124)	SETENA (Interrupt 123)	SETENA (Interrupt 122)	SETENA (Interrupt 121)	SETENA (Interrupt 120)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	SETENA (Interrupt 119)	SETENA (Interrupt 118)	SETENA (Interrupt 117)	SETENA (Interrupt 116)	SETENA (Interrupt 115)	SETENA (Interrupt 114)	SETENA (Interrupt 113)	SETENA (Interrupt112)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SETENA (Interrupt 111)	SETENA (Interrupt 110)	SETENA (Interrupt 109)	SETENA (Interrupt 108)	SETENA (Interrupt 107)	SETENA (Interrupt 106)	SETENA (Interrupt 105)	SETENA (Interrupt 104)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SETENA (Interrupt 103)	SETENA (Interrupt 102)	SETENA (Interrupt 101)	SETENA (Interrupt 100)	SETENA (Interrupt 99)	-	-	SETENA (Interrupt 96)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	SETENA	R/W	Interrupt number [127:99] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.
2-1	-	R/W	Write as "0".
0	SETENA	R/W	Interrupt number [96] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

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

7.6.2.9 Interrupt Clear-Enable Register 1

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

Bit	Bit Symbol	Type	Function
31-16	CLRENA	R/W	Interrupt number [31:16] [Write] 1: Disabled [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. It can be performed to enable interrupts and to check if interrupts are disabled. Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.
15-14	-	R/W	Write as "0".
13-0	CLRENA	R/W	Interrupt number [13:0] [Write] 1: Disabled [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. It can be performed to enable interrupts and to check if interrupts are disabled. Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

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

7.6.2.10 Interrupt Clear-Enable Register 2

	31	30	29	28	27	26	25	24
bit symbol	CLRENA (Interrupt 63)	CLRENA (Interrupt 62)	CLRENA (Interrupt 61)	CLRENA (Interrupt 60)	CLRENA (Interrupt 59)	CLRENA (Interrupt 58)	CLRENA (Interrupt 57)	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 55)	CLRENA (Interrupt 54)	-	-	-	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 45)	CLRENA (Interrupt 44)	CLRENA (Interrupt 43)	CLRENA (Interrupt 42)	CLRENA (Interrupt 41)	CLRENA (Interrupt 40)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	CLRENA (Interrupt 34)	CLRENA (Interrupt 33)	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-22	CLRENA	R/W	Interrupt number [63:54] [Write] 1: Disabled [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. It can be performed to enable interrupts and to check if interrupts are disabled. Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.
21-19	-	R/W	Write as "0".
18-8	CLRENA	R/W	Interrupt number [50:40] [Write] 1: Disabled [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. It can be performed to enable interrupts and to check if interrupts are disabled. Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.
7-3	-	R/W	Write as "0".
2-1	CLRENA	R/W	Interrupt number [34:33] [Write] 1: Disabled [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. It can be performed to enable interrupts and to check if interrupts are disabled. Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.
0	-	R/W	Write as "0".

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

7.6.2.11 Interrupt Clear-Enable Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	-	CLRENA (Interrupt 93)	CLRENA (Interrupt 92)	CLRENA (Interrupt 91)	CLRENA (Interrupt 90)	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	CLRENA (Interrupt 87)	-	-	CLRENA (Interrupt 84)	CLRENA (Interrupt 83)	CLRENA (Interrupt 82)	CLRENA (Interrupt 81)	CLRENA (Interrupt 80)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CLRENA (Interrupt 79)	CLRENA (Interrupt 78)	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)	CLRENA (Interrupt 65)	CLRENA (Interrupt 64)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-30	-	R/W	Write as "0".
29-26	CLRENA	R/W	Interrupt number [93:90] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. It can be performed to enable interrupts and to check if interrupts are disabled. Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.
25-24	-	R/W	Write as "0".
23	CLRENA	R/W	Interrupt number [87] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.
22-21	-	R/W	Write as "0".
20-0	CLRENA	R/W	Interrupt number [84:64] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

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

7.6.2.12 Interrupt Clear-Enable Register 4

	31	30	29	28	27	26	25	24
bit symbol	CLRENA (Interrupt 127)	CLRENA (Interrupt 126)	CLRENA (Interrupt 125)	CLRENA (Interrupt 124)	CLRENA (Interrupt 123)	CLRENA (Interrupt 122)	CLRENA (Interrupt 121)	CLRENA (Interrupt 120)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	CLRENA (Interrupt 119)	CLRENA (Interrupt 118)	CLRENA (Interrupt 117)	CLRENA (Interrupt 116)	CLRENA (Interrupt 115)	CLRENA (Interrupt 114)	CLRENA (Interrupt 113)	CLRENA (Interrupt112)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CLRENA (Interrupt 111)	CLRENA (Interrupt 110)	CLRENA (Interrupt 109)	CLRENA (Interrupt 108)	CLRENA (Interrupt 107)	CLRENA (Interrupt 106)	CLRENA (Interrupt 105)	CLRENA (Interrupt 104)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CLRENA (Interrupt 103)	CLRENA (Interrupt 102)	CLRENA (Interrupt 101)	CLRENA (Interrupt 100)	CLRENA (Interrupt 99)	-	-	CLRENA (Interrupt 96)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	CLRENA	R/W	Interrupt number [127:99] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. It can be performed to enable interrupts and to check if interrupts are disabled. Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.
2-1	-	R/W	Write as "0".
0	CLRENA	R/W	Interrupt number [96] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. It can be performed to enable interrupts and to check if interrupts are disabled. Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

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

7.6.2.13 Interrupt Set-Pending Register 1

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

Bit	Bit Symbol	Type	Function
31-16	CLRENA	R/W	Interrupt number [31:16] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts. Writing "1" to a corresponding bit in the Interrupt Clear-Pending Register clears the bit in this register.
15-14	-	R/W	Write as "0".
13-0	CLRENA	R/W	Interrupt number [13:0] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts. Writing "1" to a corresponding bit in the Interrupt Clear-Pending Register clears the bit in this register.

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

7.6.2.14 Interrupt Set-Pending Register 2

	31	30	29	28	27	26	25	24
bit symbol	SETPEND (Interrupt 63)	SETPEND (Interrupt 62)	SETPEND (Interrupt 61)	SETPEND (Interrupt 60)	SETPEND (Interrupt 59)	SETPEND (Interrupt 58)	SETPEND (Interrupt 57)	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 55)	SETPEND (Interrupt 54)	-	-	-	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 45)	SETPEND (Interrupt 44)	SETPEND (Interrupt 43)	SETPEND (Interrupt 42)	SETPEND (Interrupt 41)	SETPEND (Interrupt 40)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	SETPEND (Interrupt 34)	SETPEND (Interrupt 33)	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-22	SETPEND	R/W	Interrupt number [63:54] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts. Clear and Interrupt Set-Pending Register bit by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.
21-19	-	R/W	Write as "0".
18-8	SETPEND	R/W	Interrupt number [50:40] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts. Clear and Interrupt Set-Pending Register bit by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.
7-3	-	R/W	Write as "0".
2-1	SETPEND	R/W	Interrupt number [34:33] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts. Clear and Interrupt Set-Pending Register bit by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.
0	-	R/W	Write as "0".

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

7.6.2.15 Interrupt Set-Pending Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	-	SETPEND (Interrupt 93)	SETPEND (Interrupt 92)	SETPEND (Interrupt 91)	SETPEND (Interrupt 90)	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	SETPEND (Interrupt 87)	-	-	SETPEND (Interrupt 84)	SETPEND (Interrupt 83)	SETPEND (Interrupt 82)	SETPEND (Interrupt 81)	SETPEND (Interrupt 80)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	SETPEND (Interrupt 79)	SETPEND (Interrupt 78)	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)	SETPEND (Interrupt 65)	SETPEND (Interrupt 64)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-30	-	R/W	Write as "0".
29-26	SETPEND	R/W	Interrupt number [93:90] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts. Clear and Interrupt Set-Pending Register bit by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.
25-24	-	R/W	Write as "0".
23	SETPEND	R/W	Interrupt number [87] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts. Clear and Interrupt Set-Pending Register bit by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.
22-21	-	R/W	Write as "0".
20-0	SETPEND	R/W	Interrupt number [84:64] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts. Clear and Interrupt Set-Pending Register bit by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.

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

7.6.2.16 Interrupt Set-Pending Register 4

	31	30	29	28	27	26	25	24
bit symbol	SETPEND (Interrupt 127)	SETPEND (Interrupt 126)	SETPEND (Interrupt 125)	SETPEND (Interrupt 124)	SETPEND (Interrupt 123)	SETPEND (Interrupt 122)	SETPEND (Interrupt 121)	SETPEND (Interrupt 120)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	SETPEND (Interrupt 119)	SETPEND (Interrupt 118)	SETPEND (Interrupt 117)	SETPEND (Interrupt 116)	SETPEND (Interrupt 115)	SETPEND (Interrupt 114)	SETPEND (Interrupt 113)	SETPEND (Interrupt112)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	SETPEND (Interrupt 111)	SETPEND (Interrupt 110)	SETPEND (Interrupt 109)	SETPEND (Interrupt 108)	SETPEND (Interrupt 107)	SETPEND (Interrupt 106)	SETPEND (Interrupt 105)	SETPEND (Interrupt 104)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	SETPEND (Interrupt 103)	SETPEND (Interrupt 102)	SETPEND (Interrupt 101)	SETPEND (Interrupt 100)	SETPEND (Interrupt 99)	-	-	SETPEND (Interrupt 96)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-3	SETPEND	R/W	Interrupt number [127:99] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts. Clear and Interrupt Set-Pending Register bit by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.
2-1	-	R/W	Write as "0".
0	SETPEND	R/W	Interrupt number [96] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts. Clear and Interrupt Set-Pending Register bit by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.

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

7.6.2.17 Interrupt Clear-Pending Register 1

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

Bit	Bit Symbol	Type	Function
31-16	CLRPEND	R/W	Interrupt number [31:16] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts.
15-14	-	R/W	Write as "0".
13-0	CLRPEND	R/W	Interrupt number [13:0] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts.

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

7.6.2.18 Interrupt Clear-Pending Register 2

	31	30	29	28	27	26	25	24
bit symbol	CLRPEND (Interrupt 63)	CLRPEND (Interrupt 62)	CLRPEND (Interrupt 61)	CLRPEND (Interrupt 60)	CLRPEND (Interrupt 59)	CLRPEND (Interrupt 58)	CLRPEND (Interrupt 57)	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 55)	CLRPEND (Interrupt 54)	-	-	-	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 45)	CLRPEND (Interrupt 44)	CLRPEND (Interrupt 43)	CLRPEND (Interrupt 42)	CLRPEND (Interrupt 41)	CLRPEND (Interrupt 40)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	CLRPEND (Interrupt 34)	CLRPEND (Interrupt 33)	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-22	CLRPEND	R/W	Interrupt number [63:54] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts.
21-19	-	R/W	Write as "0".
18-8	CLRPEND	R/W	Interrupt number [50:40] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts.
7-3	-	R/W	Write as "0".
2-1	CLRPEND	R/W	Interrupt number [34:33] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts.
0	-	R/W	Write as "0".

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

7.6.2.19 Interrupt Clear-Pending Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	-	CLRPEND (Interrupt 93)	CLRPEND (Interrupt 92)	CLRPEND (Interrupt 91)	CLRPEND (Interrupt 90)	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	CLRPEND (Interrupt 87)	-	-	CLRPEND (Interrupt 84)	CLRPEND (Interrupt 83)	CLRPEND (Interrupt 82)	CLRPEND (Interrupt 81)	CLRPEND (Interrupt 80)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	CLRPEND (Interrupt 79)	CLRPEND (Interrupt 78)	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)	CLRPEND (Interrupt 65)	CLRPEND (Interrupt 64)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-30	-	R/W	Write as "0".
29-26	CLRPEND	R/W	Interrupt number [93:90] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts.
25-24	-	R/W	Write as "0".
23	CLRPEND	R/W	Interrupt number [87] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts.
22-21	-	R/W	Write as "0".
20-0	CLRPEND	R/W	Interrupt number [84:64] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts.

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

7.6.2.20 Interrupt Clear-Pending Register 4

	31	30	29	28	27	26	25	24
bit symbol	CLRPEND (Interrupt 127)	CLRPEND (Interrupt 126)	CLRPEND (Interrupt 125)	CLRPEND (Interrupt 124)	CLRPEND (Interrupt 123)	CLRPEND (Interrupt 122)	CLRPEND (Interrupt 121)	CLRPEND (Interrupt 120)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	CLRPEND (Interrupt 119)	CLRPEND (Interrupt 118)	CLRPEND (Interrupt 117)	CLRPEND (Interrupt 116)	CLRPEND (Interrupt 115)	CLRPEND (Interrupt 114)	CLRPEND (Interrupt 113)	CLRPEND (Interrupt112)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	CLRPEND (Interrupt 111)	CLRPEND (Interrupt 110)	CLRPEND (Interrupt 109)	CLRPEND (Interrupt 108)	CLRPEND (Interrupt 107)	CLRPEND (Interrupt 106)	CLRPEND (Interrupt 105)	CLRPEND (Interrupt 104)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	CLRPEND (Interrupt 103)	CLRPEND (Interrupt 102)	CLRPEND (Interrupt 101)	CLRPEND (Interrupt 100)	CLRPEND (Interrupt 99)	-	-	CLRPEND (Interrupt 96)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-3	CLRPEND	R/W	Interrupt number [127:99] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts.
2-1	-	R/W	Write as "0".
0	CLRPEND	R/W	Interrupt number [96] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts.

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

7.6.2.21 Interrupt Priority Register

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

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

	31 24	23 16	15 8	7 0
0xE000_E400	PRI_3	PRI_2	PRI_1	PRI_0
0xE000_E404	PRI_7	PRI_6	PRI_5	PRI_4
0xE000_E408	PRI_11	PRI_10	PRI_9	PRI_8
0xE000_E40C	Reserved	Reserved	PRI_13	PRI_12
0xE000_E410	PRI_19	PRI_18	PRI_17	PRI_16
0xE000_E414	PRI_23	PRI_22	PRI_21	PRI_20
0xE000_E418	PRI_27	PRI_26	PRI_25	PRI_24
0xE000_E41C	PRI_31	PRI_30	PRI_29	PRI_28
0xE000_E420	Reserved	PRI_34	PRI_33	Reserved
0xE000_E424	Reserved	Reserved	Reserved	Reserved
0xE000_E428	PRI_43	PRI_42	PRI_41	PRI_40
0xE000_E42C	PRI_47	PRI_46	PRI_45	PRI_44
0xE000_E430	Reserved	PRI_50	PRI_49	PRI_48
0xE000_E434	PRI_55	PRI_54	Reserved	Reserved
0xE000_E438	PRI_59	PRI_58	PRI_57	PRI_56
0xE000_E43C	PRI_63	PRI_62	PRI_61	PRI_60
0xE000_E440	PRI_67	PRI_66	PRI_65	PRI_64
0xE000_E444	PRI_71	PRI_70	PRI_69	PRI_68
0xE000_E448	PRI_75	PRI_74	PRI_73	PRI_72
0xE000_E44C	PRI_79	PRI_78	PRI_77	PRI_76
0xE000_E450	PRI_83	PRI_82	PRI_81	PRI_80
0xE000_E454	PRI_87	Reserved	Reserved	PRI_84
0xE000_E458	PRI_91	PRI_90	Reserved	Reserved
0xE000_E45C	Reserved	Reserved	PRI_93	PRI_92
0xE000_E460	PRI_99	Reserved	Reserved	PRI_96
0xE000_E464	PRI_103	PRI_102	PRI_101	PRI_100
0xE000_E468	PRI_107	PRI_106	PRI_105	PRI_104
0xE000_E46C	PRI_111	PRI_110	PRI_109	PRI_108
0xE000_E470	PRI_115	PRI_114	PRI_113	PRI_112
0xE000_E474	PRI_119	PRI_118	PRI_117	PRI_116
0xE000_E478	PRI_123	PRI_122	PRI_121	PRI_120
0xE000_E47C	PRI_127	PRI_126	PRI_125	PRI_124

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.

7.6.2.22 Vector Table Offset Register

	31	30	29	28	27	26	25	24
bit symbol	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
31-7	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.

Note:<TBLOFF[31:30]> should be set to "00".

7.6.2.23 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 sub-priority 001: six bits of pre-emption priority, two bits of sub-priority 010: five bits of pre-emption priority, three bits of sub-priority 011: four bits of pre-emption priority, four bits of sub-priority 100: three bits of pre-emption priority, five bits of sub-priority 101: two bits of pre-emption priority, six bits of sub-priority 110: one bit of pre-emption priority, seven bits of sub-priority 111: no pre-emption priority, eight bits of sub-priority 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 re initialize 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.**

7.6.2.24 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.

7.6.2.25 System Handler Control and State Register

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

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

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

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

7.6.3 Clock generator registers

7.6.3.1 CGIMCGA(CG Interrupt Mode Control Register A)

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

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

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

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

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

7.6.3.2 CGIMCGB(CG Interrupt Mode Control Register B)

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

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-28	EMCG7[2:0]	R/W	active level setting of INT7 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
27-26	EMST7[1:0]	R	active level of INT7 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
25	−	R	Reads as undefined.
24	INT7EN	R/W	INT7 clear input 0:Disable 1: Enable
23	−	R	Read as 0.
22-20	EMCG6[2:0]	R/W	active level setting of INT6 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
19-18	EMST6[1:0]	R	active level of INT6 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
17	−	R	Reads as undefined.
16	INT6EN	R/W	INT6 clear input 0:Disable 1: Enable
15	−	R	Read as 0.
14-12	EMCG5[2:0]	R/W	active level setting of INT5 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges

Bit	Bit Symbol	Type	Function
11-10	EMST5[1:0]	R	active level of INT5 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
9	−	R	Reads as undefined.
8	INT5EN	R/W	INT5 clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCG4[2:0]	R/W	active level setting of INT4 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
3-2	EMST4[1:0]	R	active level of INT4 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
1	−	R	Reads as undefined.
0	INT4EN	R/W	INT4 clear input 0:Disable 1: Enable

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

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

7.6.3.3 CGIMCGC(CG Interrupt Mode Control Register C)

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

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-28	EMCGB[2:0]	R/W	active level setting of INTB standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
27-26	EMSTB[1:0]	R	active level of INTB standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
25	−	R	Reads as undefined.
24	INTBEN	R/W	INTB clear input 0:Disable 1: Enable
23	−	R	Read as 0.
22-20	EMCGA[2:0]	R/W	active level setting of INTA standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
19-18	EMSTA[1:0]	R	active level of INTA standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
17	−	R	Reads as undefined.
16	INTAEN	R/W	INTA clear input 0:Disable 1: Enable
15	−	R	Read as 0.
14-12	EMCG9[2:0]	R/W	active level setting of INT9 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges

Bit	Bit Symbol	Type	Function
11-10	EMST9[1:0]	R	active level of INT9 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
9	−	R	Reads as undefined.
8	INT9EN	R/W	INT9 clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCG8[2:0]	R/W	active level setting of INT8 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
3-2	EMST8[1:0]	R	active level of INT8 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
1	−	R	Reads as undefined.
0	INT8EN	R/W	INT8 clear input 0:Disable 1: Enable

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

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

7.6.3.4 CGIMCGD(CG Interrupt Mode Control Register D)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCGRMCRX			EMSTRMCRX		-	INTRMC RXEN
After reset	0	0	1	0	0	0	undefined	0
	23	22	21	20	19	18	17	16
bit symbol	-	EMCGRTC			EMSTRTC		-	INTRTCEN
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	-	EMCGUSBWKUP			EMSTUSBWKUP		-	INTUSB WKUPEN
After reset	0	0	1	0	0	0	undefined	0

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-28	EMCGRMC RX[2:0]	R/W	active level setting of INTRMCRX standby clear request. (except below : setting prohibited) 011: Rising edge
27-26	EMST RMCRX[1:0]	R	active level of INTRMCRX standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
25	−	R	Reads as undefined.
24	INTRMC RXEN	R/W	INTRMCRX clear input 0:Disable 1: Enable
23	−	R	Read as 0.
22-20	EMCG RTC[2:0]	R/W	active level setting of INTRTC standby clear request. (except below: setting prohibited) 010: Falling edge
19-18	EMSTRTC[1:0]	R	active level of INTRTC standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
17	−	R	Reads as undefined.
16	INTRTCEN	R/W	INTRTC clear input 0:Disable 1: Enable
15	−	R	Read as 0.
14-12	EMCGD[2:0]	R/W	active level setting of INTD standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
11-10	EMSTD[1:0]	R	active level of INTD standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
9	−	R	Reads as undefined.

Bit	Bit Symbol	Type	Function
8	INTDEN	R/W	INTD clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCGUSB WKUP[2:0]	R/W	active level setting of INTUSBWKUP standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
3-2	EMSTUSB WKUP[1:0]	R	active level of INTUSBWKUP standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
1	−	R	Reads as undefined.
0	INTUSB WKUPEN	R/W	INTUSBWKUP clear input 0:Disable 1: Enable

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

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

7.6.3.5 CGICRCG(CG Interrupt Request Clear Register)

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

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4-0	ICRCG[4:0]	W	Clear interrupt requests. 0_0000: INT0 0_1000: INT8 0_0001: INT1 0_1001: INT9 0_0010: INT2 0_1010: INTA 0_0011: INT3 0_1011: INTB 0_0100: INT4 0_1100: INTUSBWKUP 0_0101: INT5 0_1101: INTD 0_0110: INT6 0_1110: INTRTC 0_0111: INT7 0_1111: INTRMCRX 1_0000 to 1_1111: Reserved. Read as 0.

7.6.3.6 CGNMIFLG(NMI Flag Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	NMIFLG3	NMIFLG2	NMIFLG1	NMIFLG0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as 0.
3	NMIFLG3	R	NMI source generation flag 0: not applicable 1: Only upper than the setting voltage when voltage decreasing.
2	NMIFLG2	R	NMI source generation flag 0: not applicable 1: Only lower than the setting voltage when voltage decreasing.
1	NMIFLG1	R	NMI source generation flag 0: not applicable 1: generated from $\overline{\text{NMI}}$ pin
0	NMIFLG0	R	NMI source generation flag 0: not applicable 1: generated from WDT

Note: **<NMIFLG[3:0]> are cleared to "0" when they are read.**

7.6.3.7 CGRSTFLG (Reset Flag Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After 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	-	LVDRSTF	OFDRSTF	DBGRSTF	STOP2RSTF	WDTRSTF	PINRSTF	PONRSTF
After power on reset	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	1

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	LVDRSTF	R/W	LVD reset flag 0: "0" is written 1: Reset by LVD
5	OFDRSTF	R/W	OFD reset flag 0: "0" is written 1: Reset by OFD
4	DBGRSTF	R/W	Debug reset flag (Note2) 0: "0" is written 1: Reset from SYSRESETREQ
3	STOP2RSTF	R/W	STOP2 reset flag 0: "0" is written 1: Reset flag by STOP2 mode release
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 by Power On Reset

Note 1: **The reset which is generated by application interrupt in NVIC of CPU and setting reset control register <SYSRESETREQ> is displayed.**

Note 2: **CGRSTFLG is not cleared automatically. Therefore, clear the corresponded bit to "0" to clear it.**

Note 3: **TMPM367FDFG has a built-in power on reset circuit. <PONRSTF> is set to "1" after a power supply is turned on. After other reset is occurred, the corresponded reset flag is set to "1".**

Note 4: **After a power supply is turned on, the flag excepted <PONRSTF> is invalid.**

8. µDMA Controller (µDMAC)

8.1 Function Overview

8.1.1 Function list

The main functions are shown below.

Table 8-1 µDMA outline

Functions	Features		Outline
Channels	64 channels (2 units)		Unit A: 32 channels, Unit B: 32 channels
Start trigger	Hardware	Burst (continuous transfer)	DMA requests from peripheral functions
		Single (single transfer)	
	Software		Configuring using DMAxChnlSwRequest register
Priority	Between units	Unit A > Unit B	Hardware fixation
	Between channels	ch0 (high priority) > … > ch31 (high priority) > ch0 (normal priority) > … > ch31 (normal priority)	High-priority can be configured by DMAxChnlPrioritySet register
Transfer data size	8/16/32bit		
Address	Transfer source address	Increment / fixed	Transfer source address and destination address can be set to increment or fixed.
	transfer destination address	Increment / fixed	
The number of transfer	1 to 1024		
Transfer type	Peripheral circuits (register) → memory Memory → peripheral circuits (register) Memory → memory		If you select memory to memory, hardware start for DMA start up is not supported.
Interrupt function	Transfer end interrupt Error interrupt		
Transfer mode	Basic mode Automatic request mode Ping-pong mode Memory scatter / gather mode Peripheral scatter / gather mode		

Note: 1 word = 32bit

8.1.2 DMA requests

The below table shows DMA requests. The compositions of the unit A and the unit B are the same.

Table 8-2 DMA request list

ch	Hardware request (The request from the peripheral function connected to DMA)			Software request (Request setting register)		The interrupt request which DMA outputs	
	Factor	Burst	Single	0x4004_C014 (Unit A)	0x4004_D014 (Unit B)	No	Factor
0	ADC A Conversion End	o	–	bit0	bit0	100	INTDMAADA
1	ADC B Conversion End	o	–	bit1	bit1	101	INTDMAADB
2	DAC0 conversion trigger	–	o	bit2	bit2	102	INTDMADAA
3	DAC1 conversion trigger	–	o	bit3	bit3	103	INTDMADAB
4	SSP0 reception	o	o	bit4	bit4	104	INTDMASPR0
5	SSP0 transmission	o	o	bit5	bit5	105	INTDMASPT0
6	SSP1 reception	o	o	bit6	bit6	106	INTDMASPR1
7	SSP1 transmission	o	o	bit7	bit7	107	INTDMASPT1
8	SSP2 reception	o	o	bit8	bit8	108	INTDMASPR2
9	SSP2 transmission	o	o	bit9	bit9	109	INTDMASPT2
10	UART4 reception	o	o	bit10	bit10	110	INTDMAUTR0
11	UART4 transmission	o	o	bit11	bit11	111	INTDMAUTT0
12	UART5 reception	o	o	bit12	bit12	112	INTDMAUTR1
13	UART5 transmission	o	o	bit13	bit13	113	INTDMAUTT1
14	SIO/UART0 reception	o	–	bit14	bit14	114	INTDMARX0
15	SIO/UART0 transmission	o	–	bit15	bit15	115	INTDMATX0
16	SIO/UART1 reception	o	–	bit16	bit16	116	INTDMARX1
17	SIO/UART1 transmission	o	–	bit17	bit17	117	INTDMATX1
18	SIO/UART2 reception	o	–	bit18	bit18	118	INTDMARX2
19	SIO/UART2 transmission	o	–	bit19	bit19	119	INTDMATX2
20	SIO/UART3 reception	o	–	bit20	bit20	120	INTDMARX3
21	SIO/UART3 transmission	o	–	bit21	bit21	121	INTDMATX3
22	I2C/SIO0 receive / transmit	o	–	bit22	bit22	104	INTDMASPR0
23	I2C/SIO1 receive / transmit	o	–	bit23	bit23	122	INTDMASBI1
24	I2C/SIO2 receive / transmit	o	–	bit24	bit24	123	INTDMASBI2
25	TMRB0 compare match	o	–	bit25	bit25	124	INTDMATB
26	TMRB1 compare match	o	–	bit26	bit26		
27	TMRB2 compare match	o	–	bit27	bit27		
28	TMRB3 compare match	o	–	bit28	bit28		
29	TMRB4 compare match	o	–	bit29	bit29		
30	DMA request pin	o	–	bit30	bit30	125	INTDMARQ
31	–	–	–	bit31	bit31	–	–
						126	INTDMAAERR
						127	INTDMABERR

Note 1: After setting the units A and B as DMAxCfg = 0x00000001, DMAxChnlReqMaskSet = 0xFFFFFFFF and DMAxChnlEnableSet = 0xFFFFFFFF, the channel of the unit to be used is set as mask release (it is an applicable bit of DMAxChnlReqMaskClr "1"). However, please do not cancel the same factor by the unit A and B both.

Note 2: Interruption corresponding to each request factor of the software request and the hardware request generated by register setup of a notes relevance bit is outputted. The interruption factor list is a name corresponding to a hardware request. (see the table of the interruption factor list of exceptions chapters)

8.2 Block Diagram

The µDMA controller contains the following blocks.

- APB block

  This block controls the access to the control register.
- AHB block

  This block controls the bus cycle of the DMA transfer.
- DMA control block

  This block controls the whole operation of the DMA.

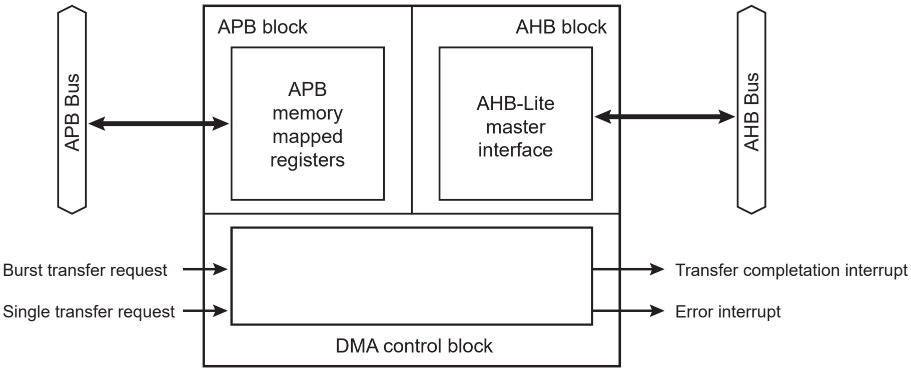

APB Bus
APB block
APB memory mapped registers
AHB block
AHB-Lite master interface
AHB Bus
Burst transfer request
Single transfer request
Transfer completation interrupt
Error interrupt
DMA control block

Figure 8-1 µDMA Block Diagram (common to the Unit A and B)

8.3 Descriptions of Registers

8.3.1 Register list

The following lists the control registers and addresses.

Unit x	Base Address
Unit A	0x4004_C000
Unit B	0x4004_D000

Register names		Address (Base+)
DMA status Register	DMAxStatus	0x0000
DMA configuration Register	DMAxCfg	0x0004
channel control data base pointer Register	DMAxCtrlBasePtr	0x0008
channel alternate control data base pointer Register	DMAxAltCtlBasePtr	0x000C
reserved	-	0x0010
channel software request Register	DMAxChnlSwRequest	0x0014
channel useburst set Register	DMAxChnlUseburstSet	0x0018
channel useburst clear Register	DMAxChnlUseburstClr	0x001C
channel request mask set Register	DMAxChnlReqMaskSet	0x0020
channel request mask clear Register	DMAxChnlReqMaskClr	0x0024
channel enable set Register	DMAxChnlEnableSet	0x0028
channel enable clear Register	DMAxChnlEnableClr	0x002C
channel primary-alternate set Register	DMAxChnlPriAltSet	0x0030
channel primary-alternate clear Register	DMAxChnlPriAltClr	0x0034
channel priority set Register	DMAxChnlPrioritySet	0x0038
channel priority clear Register	DMAxChnlPriorityClr	0x003C
reserved	-	0x0040 - 0x004B
Bus error clear Register	DMAxErrClr	0x004C
reserved	-	0x0050 - 0x0FFF

Note 1: Access the registers by using word (32 bit) reads and word writes.

Note 2: Access to the "Reserved" address is prohibited.

8.3.2 DMAxStatus (DMA Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	1	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	1	1	1	1	1
	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	-	-	-	-	-	-	-	master_ enable
After reset	Undefined	Undefined	Undefined	Undefined	0	0	0	0

Bit	Bit Symbol	Type	Functions
31-29	-	R	Read as zero.
28	-	R	Read as one.
27-21	-	R	Read as zero.
20-16	-	R	Read as one.
15-8	-	R	Read as zero.
7-4	-	R	Read as undefined value.
3-1	-	R	Read as zero.
0	master_enable	R	DMA operation 0: Disabled 1: Enabled

8.3.3 DMAxCfg (DMA Configuration 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	-	-	-	-	-	-	-	master_ enable
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-1	-	W	Write zero.
0	master_ enable	W	DMA operation 0: Disabled 1: Enabled

8.3.4 DMAxCtrlBasePtr (Channel control data base pointer Register)

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

Bit	Bit Symbol	Type	Function
31-10	ctrl_base_ptr	R/W	Primary database pointer Specifies the base address of the primary data.
9-0	-	R	Read as zero.

8.3.5 DMAxAltCtrlBasePtr (Channel alternate control data base pointer Register)

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

Bit	Bit Symbol	Type	Function
31-0	alt_ctrl_base_pt	R	Alternative data base pointer. Reads the base address of the alternative data.

8.3.6 DMAxChnlSwRequest(Channel software request Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_sw_request (ch31)	chnl_sw_request (ch30)	chnl_sw_request (ch29)	chnl_sw_request (ch28)	chnl_sw_request (ch27)	chnl_sw_request (ch26)	chnl_sw_request (ch25)	chnl_sw_request (ch24)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	chnl_sw_request (ch23)	chnl_sw_request (ch22)	chnl_sw_request (ch21)	chnl_sw_request (ch20)	chnl_sw_request (ch19)	chnl_sw_request (ch18)	chnl_sw_request (ch17)	chnl_sw_request (ch16)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	chnl_sw_request (ch15)	chnl_sw_request (ch14)	chnl_sw_request (ch13)	chnl_sw_request (ch12q)	chnl_sw_request (ch11)	chnl_sw_request (ch10)	chnl_sw_request (ch9)	chnl_sw_request (ch8)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	chnl_sw_request (ch7)	chnl_sw_request (ch6)	chnl_sw_request (ch5)	chnl_sw_request (ch4)	chnl_sw_request (ch3)	chnl_sw_request (ch2)	chnl_sw_request (ch1)	chnl_sw_request (ch0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-0	chnl_sw_request	W	DMA request 0: Requests a transfer 1: Does not request a transfer Specifies transfer requests to the each channel.

8.3.7 DMAxChnlUseburstSet(Channel useburst set Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_useburst _set (ch31)	chnl_useburst _set (ch30)	chnl_useburst _set (ch29)	chnl_useburst _set (ch28)	chnl_useburst _set (ch27)	chnl_useburst _set (ch26)	chnl_useburst _set (ch25)	chnl_useburst _set (ch24)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	chnl_useburst _set (ch23)	chnl_useburst _set (ch22)	chnl_useburst _set (ch21)	chnl_useburst _set (ch20)	chnl_useburst _set (ch19)	chnl_useburst _set (ch18)	chnl_useburst _set (ch17)	chnl_useburst _set (ch16)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	chnl_useburst _set (ch15)	chnl_useburst _set (ch14)	chnl_useburst _set (ch13)	chnl_useburst _set (ch12)	chnl_useburst _set (ch11)	chnl_useburst _set (ch10)	chnl_useburst _set (ch9)	chnl_useburst _set (ch8)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	chnl_useburst _set (ch7)	chnl_useburst _set (ch6)	chnl_useburst _set (ch5)	chnl_useburst _set (ch4)	chnl_useburst _set (ch3)	chnl_useburst _set (ch2)	chnl_useburst _set (ch1)	chnl_useburst _set (ch0)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	chnl_useburst_ set	R/W	Single transfer disabled [Write] 1: Single transfer is disabled. [Read] 0: Single transfer enabled 1: Single transfer disabled. Each bit corresponds to the channels in specified numbers. Writing "1" disables the single transfer to the corresponding channel, and only burst transfer request becomes valid. Writing "0" is invalid. Set the DMAxChnlUseburstClr register in order to cancel the disabled single transfer. The enable / disable state of the corresponding bit can be confirmed by reading the bit. Bits are automatically set in the following condition: ・ This bit is zero cleared, if the number of remaining transfer is less than 2^R times at the end of second 2^R time transfer from the end ("R" is specified by the channel_cfg<R_power> of the control data). ・ If the channel_cfg<next_useburst> of the control data is set to "1" in the peripheral scatter / gather mode, this bit is set to "1" when the DMA transfer of the alternative data ends.

Note: Do not set this bit to "1" if you do not use the burst transfer request in the condition that the number of transfer is less than 2^R times.

8.3.8 DMAxChnlUseburstClr(Channel useburst clear Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_useburst _clr (ch31)	chnl_useburst _clr (ch30)	chnl_useburst _clr (ch29)	chnl_useburst _clr (ch28)	chnl_useburst _clr (ch27)	chnl_useburst _clr (ch26)	chnl_useburst _clr (ch25)	chnl_useburst _clr (ch24)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	chnl_useburst _clr (ch23)	chnl_useburst _clr (ch22)	chnl_useburst _clr (ch21)	chnl_useburst _clr (ch20)	chnl_useburst _clr (ch19)	chnl_useburst _clr (ch18)	chnl_useburst _clr (ch17)	chnl_useburst _clr (ch16)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	chnl_useburst _clr (ch15)	chnl_useburst _clr (ch14)	chnl_useburst _clr (ch13)	chnl_useburst _clr (ch12)	chnl_useburst _clr (ch11)	chnl_useburst _clr (ch10)	chnl_useburst _clr (ch9)	chnl_useburst _clr (ch8)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	chnl_useburst _clr (ch7)	chnl_useburst _clr (ch6)	chnl_useburst _clr (ch5)	chnl_useburst _clr (ch4)	chnl_useburst _clr (ch3)	chnl_useburst _clr (ch2)	chnl_useburst _clr (ch1)	chnl_useburst _clr (ch0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-0	chnl_useburst_ clr	W	Single transfer enabled. 1: Enables single transfer Each bit corresponds to the channels in specified numbers. Writing "1" enables the single transfer to the corresponding channel. Writing "0" is invalid. To disable or confirm signal transfer, configure the DMAxChnlUseburstSet register.

8.3.9 DMAxChnlReqMaskSet(Channel request mask set Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_req_mask_set (ch31)	chnl_req_mask_set (ch30)	chnl_req_mask_set (ch29)	chnl_req_mask_set (ch28)	chnl_req_mask_set (ch27)	chnl_req_mask_set (ch26)	chnl_req_mask_set (ch25)	chnl_req_mask_set (ch24)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	chnl_req_mask_set (ch23)	chnl_req_mask_set (ch22)	chnl_req_mask_set (ch21)	chnl_req_mask_set (ch20)	chnl_req_mask_set (ch19)	chnl_req_mask_set (ch18)	chnl_req_mask_set (ch17)	chnl_req_mask_set (ch16)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	chnl_req_mask_set (ch15)	chnl_req_mask_set (ch14)	chnl_req_mask_set (ch13)	chnl_req_mask_set (ch12)	chnl_req_mask_set (ch11)	chnl_req_mask_set (ch10)	chnl_req_mask_set (ch9)	chnl_req_mask_set (ch8)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	chnl_req_mask_set (ch7)	chnl_req_mask_set (ch6)	chnl_req_mask_set (ch5)	chnl_req_mask_set (ch4)	chnl_req_mask_set (ch3)	chnl_req_mask_set (ch2)	chnl_req_mask_set (ch1)	chnl_req_mask_set (ch0)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	chnl_req_mask_set	R/W	DMA request mask [Write] 1: Mask a DMA request [Read] 0: DMA request valid 1: DMA request invalid Each bit corresponds to the channels in specified numbers. Writing "1" disables the single transfer to the corresponding channel. Writing "0" is invalid. To disable mask, configure the DMAxChnlReqMaskClr register. The status of DMA request mask valid / invalid can be confirmed by reading the bit.

8.3.10 DMAxChnlReqMaskClr(Channel request mask clear Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_req_mask_clr (ch31)	chnl_req_mask_clr (ch30)	chnl_req_mask_clr (ch29)	chnl_req_mask_clr (ch28)	chnl_req_mask_clr (ch27)	chnl_req_mask_clr (ch26)	chnl_req_mask_clr (ch25)	chnl_req_mask_clr (ch24)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	chnl_req_mask_clr (ch23)	chnl_req_mask_clr (ch22)	chnl_req_mask_clr (ch21)	chnl_req_mask_clr (ch20)	chnl_req_mask_clr (ch19)	chnl_req_mask_clr (ch18)	chnl_req_mask_clr (ch17)	chnl_req_mask_clr (ch16)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	chnl_req_mask_clr (ch15)	chnl_req_mask_clr (ch14)	chnl_req_mask_clr (ch13)	chnl_req_mask_clr (ch12)	chnl_req_mask_clr (ch11)	chnl_req_mask_clr (ch10)	chnl_req_mask_clr (ch9)	chnl_req_mask_clr (ch8)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	chnl_req_mask_clr (ch7)	chnl_req_mask_clr (ch6)	chnl_req_mask_clr (ch5)	chnl_req_mask_clr (ch4)	chnl_req_mask_clr (ch3)	chnl_req_mask_clr (ch2)	chnl_req_mask_clr (ch1)	chnl_req_mask_clr (ch0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-0	chnl_req_mask_clr	W	DMA request mask clear 1: Clears the corresponding channel of the DMA request mask. Each bit corresponds to the channels in specified numbers. DMA request mask of the corresponding channel is cleared by writing "1". Writing "0" is invalid. Configure the DMAxChnlReqMaskSet register to enable and to confirm the function.

8.3.11 DMAxChnlEnableSet(Channel enable set Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_enable_ set (ch31)	chnl_enable_ set (ch30)	chnl_enable_ set (ch29)	chnl_enable_ set (ch28)	chnl_enable_ set (ch27)	chnl_enable_ set (ch26)	chnl_enable_ set (ch25)	chnl_enable_ set (ch24)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	chnl_enable_ set (ch23)	chnl_enable_ set (ch22)	chnl_enable_ set (ch21)	chnl_enable_ set (ch20)	chnl_enable_ set (ch19)	chnl_enable_ set (ch18)	chnl_enable_ set (ch17)	chnl_enable_ set (ch16)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	chnl_enable_ set (ch15)	chnl_enable_ set (ch14)	chnl_enable_ set (ch13)	chnl_enable_ set (ch12)	chnl_enable_ set (ch11)	chnl_enable_ set (ch10)	chnl_enable_ set (ch9)	chnl_enable_ set (ch8)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	chnl_enable_ set (ch7)	chnl_enable_ set (ch6)	chnl_enable_ set (ch5)	chnl_enable_ set (ch4)	chnl_enable_ set (ch3)	chnl_enable_ set (ch2)	chnl_enable_ set (ch1)	chnl_enable_ set (ch0)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	chnl_enable_ set	R/W	DMA operation [Write] 1: Enable the corresponding channels. [Read] 0: Corresponding bits are invalid. 1: Corresponding bits are valid. Each bit corresponds to the channels in specified numbers. Writing "1" enables the corresponding channels. Writing "0" is invalid. To disable the setting, configure the DMAxChnlEnableClr register. The status of valid / invalid if the corresponding channels can be confirmed by reading. In the following conditions, the functions automatically becomes invalid. ・DMA cycle end ・If the channel_cfg<cycle_ctrl> reads the control data of "000". ・A bus error occurs.

8.3.12 DMAxChnlEnableClr(Channel enable clear Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_enable_clr (ch31)	chnl_enable_clr (ch30)	chnl_enable_clr (ch29)	chnl_enable_clr (ch28)	chnl_enable_clr (ch27)	chnl_enable_clr (ch26)	chnl_enable_clr (ch25)	chnl_enable_clr (ch24)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	chnl_enable_clr (ch23)	chnl_enable_clr (ch22)	chnl_enable_clr (ch21)	chnl_enable_clr (ch20)	chnl_enable_clr (ch19)	chnl_enable_clr (ch18)	chnl_enable_clr (ch17)	chnl_enable_clr (ch16)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	chnl_enable_clr (ch15)	chnl_enable_clr (ch14)	chnl_enable_clr (ch13)	chnl_enable_clr (ch12)	chnl_enable_clr (ch11)	chnl_enable_clr (ch10)	chnl_enable_clr (ch9)	chnl_enable_clr (ch8)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	chnl_enable_clr (ch7)	chnl_enable_clr (ch6)	chnl_enable_clr (ch5)	chnl_enable_clr (ch4)	chnl_enable_clr (ch3)	chnl_enable_clr (ch2)	chnl_enable_clr (ch1)	chnl_enable_clr (ch0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-0	chnl_enable_clr	W	DMA disabled 1: Disables the corresponding channels. Each bit corresponds to the channels in specified numbers. Writing "1" disables the corresponding channels. Writing "0" is invalid. Configure the DMAxChnlEnableSet register in order to enable and to confirm the function. In the following conditions, the functions automatically becomes invalid. ・ DMA cycle end ・ The channel_cfg<cycle_ctrl> reads the control data of "000". ・ A bus error occurs.

8.3.13 DMAxChnlPriAltSet(Channel primary-alternate set Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_pri_alt_ set (ch31)	chnl_pri_alt_ set (ch30)	chnl_pri_alt_ set (ch29)	chnl_pri_alt_ set (ch28)	chnl_pri_alt_ set (ch27)	chnl_pri_alt_ set (ch26)	chnl_pri_alt_ set (ch25)	chnl_pri_alt_ set (ch24)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	chnl_pri_alt_ set (ch23)	chnl_pri_alt_ set (ch22)	chnl_pri_alt_ set (ch21)	chnl_pri_alt_ set (ch20)	chnl_pri_alt_ set (ch19)	chnl_pri_alt_ set (ch18)	chnl_pri_alt_ set (ch17)	chnl_pri_alt_ set (ch16)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	chnl_pri_alt_ set (ch15)	chnl_pri_alt_ set (ch14)	chnl_pri_alt_ set (ch13)	chnl_pri_alt_ set (ch12)	chnl_pri_alt_ set (ch11)	chnl_pri_alt_ set (ch10)	chnl_pri_alt_ set (ch9)	chnl_pri_alt_ set (ch8)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	chnl_pri_alt_ set (ch7)	chnl_pri_alt_ set (ch6)	chnl_pri_alt_ set (ch5)	chnl_pri_alt_ set (ch4)	chnl_pri_alt_ set (ch3)	chnl_pri_alt_ set (ch2)	chnl_pri_alt_ set (ch1)	chnl_pri_alt_ set (ch0)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	chnl_pri_alt_ set	R/W	Selects primary data or alternative data [Write] 1: Uses alternative data [Read] 0: Primary data 1: Alternative data Each bit corresponds to the channels in specified numbers. Writing "1" specifies the data of the corresponding channel "alternative". Writing "0" is invalid. Configure the DMAxChnlEnableClr register to disable. Data of the corresponding channel can be confirmed, primary or alternative, by reading the bit. In the following states, the settings are automatically changed. ・ The primary data transfer is completed in ping-pong mode, memory scatter / gather mode or peripheral scatter / gather mode. ・ Data transfer of the alternative data is completed in the ping-pong mode, memory scatter / gather mode or peripheral scatter / gather mode.

8.3.14 DMAxChnlPriAltClr(Channel primary-alternate clear Register)

	31	30	29	28	27	26	25	24
bit symbol	chn_pri_alt_ clr (ch31)	chn_pri_alt_ clr (ch30)	chn_pri_alt_ clr (ch29)	chn_pri_alt_ clr (ch28)	chn_pri_alt_ clr (ch27)	chn_pri_alt_ clr (ch26)	chn_pri_alt_ clr (ch25)	chn_pri_alt_ clr (ch24)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	chn_pri_alt_ clr (ch23)	chn_pri_alt_ clr (ch22)	chn_pri_alt_ clr (ch21)	chn_pri_alt_ clr (ch20)	chn_pri_alt_ clr (ch19)	chn_pri_alt_ clr (ch18)	chn_pri_alt_ clr (ch17)	chn_pri_alt_ clr (ch16)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	chn_pri_alt_ clr (ch15)	chn_pri_alt_ clr (ch14)	chn_pri_alt_ clr (ch13)	chn_pri_alt_ clr (ch12)	chn_pri_alt_ clr (ch11)	chn_pri_alt_ clr (ch10)	chn_pri_alt_ clr (ch9)	chn_pri_alt_ clr (ch8)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	chn_pri_alt_ clr (ch7)	chn_pri_alt_ clr (ch6)	chn_pri_alt_ clr (ch5)	chn_pri_alt_ clr (ch4)	chn_pri_alt_ clr (ch3)	chn_pri_alt_ clr (ch2)	chn_pri_alt_ clr (ch1)	chn_pri_alt_ clr (ch0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-0	chnl_pri_alt_clr	W	Clears the alternative data setting 1: Uses the primary data Each bit corresponds to the channels in specified numbers. Writing "1" sets the data of the corresponding channel to the primary data. Setting "0" is invalid. Configure the DMAxChnlPriAltSet register to set the primary data or to confirm the settings. In the following conditions, settings are automatically changed. ・ The primary data transfer in the memory scatter / gather mode or peripheral scatter / gather mode. ・ The primary data transfer in the ping-pong mode ends. ・ The alternative transfer in the ping-pong mode, memory scatter / gather mode or peripheral scatter / gather mode ends.

8.3.15 DMAxChnlPrioritySet(Channel priority set Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_priority_ set (ch31)	chnl_priority_ set (ch30)	chnl_priority_ set (ch29)	chnl_priority_ set (ch28)	chnl_priority_ set (ch27)	chnl_priority_ set (ch26)	chnl_priority_ set (ch25)	chnl_priority_ set (ch24)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	chnl_priority_ set (ch23)	chnl_priority_ set (ch22)	chnl_priority_ set (ch21)	chnl_priority_ set (ch20)	chnl_priority_ set (ch19)	chnl_priority_ set (ch18)	chnl_priority_ set (ch17)	chnl_priority_ set (ch16)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	chnl_priority_ set (ch15)	chnl_priority_ set (ch14)	chnl_priority_ set (ch13)	chnl_priority_ set (ch12)	chnl_priority_ set (ch11)	chnl_priority_ set (ch10)	chnl_priority_ set (ch9)	chnl_priority_ set (ch8)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	chnl_priority_ set (ch7)	chnl_priority_ set (ch6)	chnl_priority_ set (ch5)	chnl_priority_ set (ch4)	chnl_priority_ set (ch3)	chnl_priority_ set (ch2)	chnl_priority_ set (ch1)	chnl_priority_ set (ch0)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	chnl_priority_ set	R/W	Priority settings [Write] 1: Sets the high-priority [Read] 0: Normal priority 1: High priority Each bit corresponds to the channels in specified numbers. Writing "1" sets the priority of the corresponding channel high. Writing "0" is invalid.To change the priority again to the normal, configure the DMAxChnlPriorityClr register. the priority of the corresponding channel, high-priority or normal priority, can be confirmed by reading the bit.

8.3.16 DMAxChnlPriorityClr(Channel priority clear Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_priority_ clr (ch31)	chnl_priority_ clr (ch30)	chnl_priority_ clr (ch29)	chnl_priority_ clr (ch28)	chnl_priority_ clr (ch27)	chnl_priority_ clr (ch26)	chnl_priority_ clr (ch25)	chnl_priority_ clr (ch24)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	chnl_priority_ clr (ch23)	chnl_priority_ clr (ch22)	chnl_priority_ clr (ch21)	chnl_priority_ clr (ch20)	chnl_priority_ clr (ch19)	chnl_priority_ clr (ch18)	chnl_priority_ clr (ch17)	chnl_priority_ clr (ch16)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	chnl_priority_ clr (ch15)	chnl_priority_ clr (ch14)	chnl_priority_ clr (ch13)	chnl_priority_ clr (ch12)	chnl_priority_ clr (ch11)	chnl_priority_ clr (ch10)	chnl_priority_ clr (ch9)	chnl_priority_ clr (ch8)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	chnl_priority_ clr (ch7)	chnl_priority_ clr (ch6)	chnl_priority_ clr (ch5)	chnl_priority_ clr (ch4)	chnl_priority_ clr (ch3)	chnl_priority_ clr (ch2)	chnl_priority_ clr (ch1)	chnl_priority_ clr (ch0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-0	chnl_priority_ clr	W	Clears the high-priority setting. [Write] 1:Sets normal priority setting Each bit corresponds to the channels in specified numbers. Writing "1" changes the priority of the corresponding channel to normal priority. Writing "0" is invalid. Configure the DMAxChnlPrioritySet register to set the high-priority and to confirm the setting.

8.3.17 DMAxErrClr(Bus 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	-	-	-	-	-	-	-	err_clr
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as zero.
0	err_clr	W	Bus error [Write] 1: Clears a bus error [Read] 0: No bus error 1: The state of a bus error The bus error occurrence can be confirmed by reading the bit. Writing "1" clears a bus error. Writing "0" is invalid.

8.4 Operation

This DMA is controlled by the channel control data, which locates on the memory. A channel of the each data is four words and allocated in the contiguous areas same as the number of channels.

There are two kinds of channel control data - primary data and alternative data. According to the operation mode, one of them is selected by setting the register or both data is used.

8.4.1 Channel control data memory map

Figure 8-2 shows the memory map of the channel control data.

Since the channel control data uses a 1KB area, the start address [9:0] of the channel control data must be set to 0x000.

Set the start address of the primary data to the DMAxCtrlBasePtr and the start address of the alternative data to the DMAxAltCtrlBasePtr.

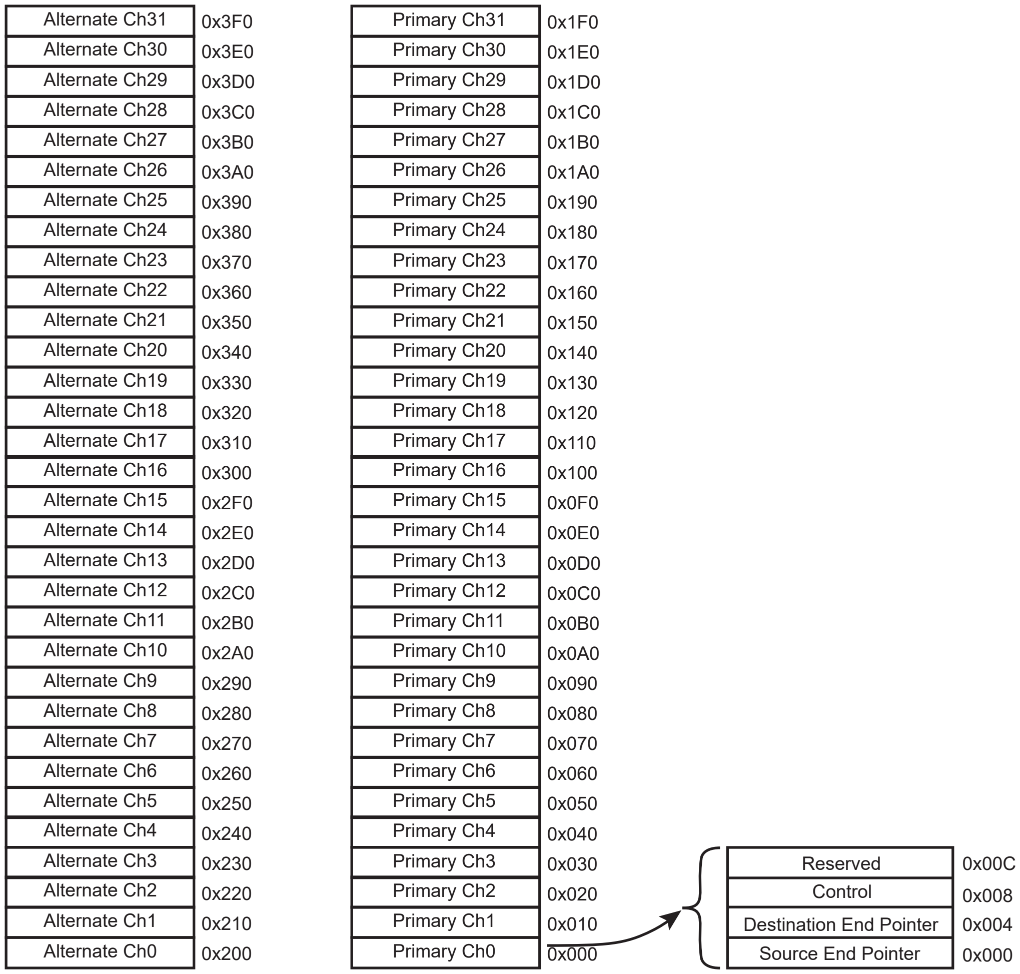

Alternate Ch31 0x3F0
Alternate Ch30 0x3E0
Alternate Ch29 0x3D0
Alternate Ch28 0x3C0
Alternate Ch27 0x3B0
Alternate Ch26 0x3A0
Alternate Ch25 0x390
Alternate Ch24 0x380
Alternate Ch23 0x370
Alternate Ch22 0x360
Alternate Ch21 0x350
Alternate Ch20 0x340
Alternate Ch19 0x330
Alternate Ch18 0x320
Alternate Ch17 0x310
Alternate Ch16 0x300
Alternate Ch15 0x2F0
Alternate Ch14 0x2E0
Alternate Ch13 0x2D0
Alternate Ch12 0x2C0
Alternate Ch11 0x2B0
Alternate Ch10 0x2A0
Alternate Ch9 0x290
Alternate Ch8 0x280
Alternate Ch7 0x270
Alternate Ch6 0x260
Alternate Ch5 0x250
Alternate Ch4 0x240
Alternate Ch3 0x230
Alternate Ch2 0x220
Alternate Ch1 0x210
Alternate Ch0 0x200
Primary Ch31 0x1F0
Primary Ch30 0x1E0
Primary Ch29 0x1D0
Primary Ch28 0x1C0
Primary Ch27 0x1B0
Primary Ch26 0x1A0
Primary Ch25 0x190
Primary Ch24 0x180
Primary Ch23 0x170
Primary Ch22 0x160
Primary Ch21 0x150
Primary Ch20 0x140
Primary Ch19 0x130
Primary Ch18 0x120
Primary Ch17 0x110
Primary Ch16 0x100
Primary Ch15 0x0F0
Primary Ch14 0x0E0
Primary Ch13 0x0D0
Primary Ch12 0x0C0
Primary Ch11 0x0B0
Primary Ch10 0x0A0
Primary Ch9 0x090
Primary Ch8 0x080
Primary Ch7 0x070
Primary Ch6 0x060
Primary Ch5 0x050
Primary Ch4 0x040
Primary Ch3 0x030
Primary Ch2 0x020
Primary Ch1 0x010
Primary Ch0 0x000
Reserved 0x00C
Control 0x008
Destination End Pointer 0x004
Source End Pointer 0x000

Figure 8-2 Memory Map of the Control Data

Figure 8-2 shows the memory map of which all 32 channels can be used. Necessary areas are determined by the number of usable channels. Table 8-3 shows the relationship between the number of channels and ad-dresses.

Table 8-3 Address bit setting of channel control

Channel	Address							Settable base address
	[9]	[8]	[7]	[6]	[5]	[4]	[3:0]	
0	-	-	-	-	-	A	Channel control data setting	0xXXXX_XX00, 0xXXXX_XX20, 0xXXXX_XX40, 0xXXXX_XX60, 0xXXXX_XX80, 0xXXXX_XXA0, 0xXXXX_XXC0, 0xXXXX_XXE0
0 to 1	-	-	-	-	A	C[0]		0xXXXX_XX00, 0xXXXX_XX40, 0xXXXX_XX80, 0xXXXX_XXC0
0 to 3	-	-	-	A	C[1:0]			0xXXXX_XX00, 0xXXXX_XX80
0 to 7	-	-	A	C[2:0]				0xXXXX_X000, 0xXXXX_X100, 0xXXXX_X200, 0xXXXX_X300, 0xXXXX_X400, 0xXXXX_X500, 0xXXXX_X600, 0xXXXX_X700, 0xXXXX_X800, 0xXXXX_X900, 0xXXXX_XA00, 0xXXXX_XB00, 0xXXXX_XC00, 0xXXXX_XD00, 0xXXXX_XE00, 0xXXXX_XF00
0 to 15	-	A	C[3:0]					0xXXXX_X000, 0xXXXX_X200, 0xXXXX_X400, 0xXXXX_X600, 0xXXXX_X800, 0xXXXX_XA00, 0xXXXX_XC00, 0xXXXX_XE00
0 to 31	A	C[4:0]						0xXXXX_X000, 0xXXXX_X400, 0xXXXX_X800, 0xXXXX_XC00

A: Primary/alternative setting (0:primary, 1:alternative)
C[x:0]: Channel number setting

8.4.2 Channel control data structure

The channel control data contains the three kinds of data shown below.

- The final address of the transfer source address
- The final address of the transfer destination address
- Control data

Each of these data is described below.

8.4.2.1 Final address of the transfer source data

Specifies the final address of the data to be transferred. The DMA calculates the start address of the source address using this data.

31 … 0
src_data_end_ptr

bit	bit symbol	Function
[31:0]	src_data_end_ptr	The final address of source transfer data

8.4.2.2 Final address of the transfer destination address

Specifies the final address of the destination address. The DMA calculates the start address of the destination address of the transfer destination address.

31 … 0
dst_data_end_ptr

bit	bit symbol	Function
[31:0]	dst_data_end_ptr	The final address of the transfer destination address

8.4.2.3 Control data settings

31 30	29 28	27 26	25 24	23 … 18	17 … 14	13 … 4	3	2 … 0
dst_inc	dst_size	src_inc	src_size	-	R_power	n_minus_1	next_useburst	cycle_ctrl

bit	bit symbol	Function
[31:30]	dst_inc	Increment the transfer destination address (note 2) 00: 1 byte 01: 2 bytes 10: 4 bytes 11: Do not increment
[29:28]	dst_size	Data size of transfer destination (note1) 00: 1 byte 01: 2 bytes 10: 4 bytes 11: Reserved
[27:26]	src_inc	Increment the transfer source address (note 2) 00: 1 byte 01: 2 bytes 10: 4 bytes 11: Do not increment
[25:24]	src_size	Data size of transfer source (note 1) 00: 1 byte 01: 2 bytes 10: 4 bytes 11: Reserved
[23:18]	-	Set "000000".
[17:14]	R_power	Arbitration 0000: After one transfer 0001: After two transfers 0010: After four transfers 0011: After eight transfers 0100: After 16 transfers 0101: After 32 transfers 0110: After 64 transfers 0111: After 128 transfers 1000: After 256 transfers 1001: After 512 transfers 1010 - 1111: Do not arbitration After the specified numbers of transfers, an existence of a transfer request is checked. If there is a high-priority request, the control is switched to high-priority channel.
[13:4]	n_minus_1	The number of transfers 0x000: Once 0x001: Twice 0x002: Three times : 0x3FF: 1024 times
[3]	next_useburst	Changes the setting of single transfer 0: Do not change the value of <chnl_useburst_set>. 1: Sets <chnl_useburst_set> to "1". Specifies whether to set "1" to the <chnl_useburst_set> bit at the end of the DMA transfer using alternative data in the peripheral scatter/ gather mode. Note) This bit <chnl_useburst_set> is zero cleared, if the number of remaining transfer is less than 2^R times at the end of second 2^Rtime transfer from the end ("R" is specified by the <R_power>). Setting this bit to "1" sets "1" to the <chnl_userburst_set>.
[2:0]	cycle_ctrl	Operation mode 000: Invalid. DMA stops the operation. 001: Basic mode 010: Automatic request mode 011: Ping-pong mode 100: Memory scatter / gather mode (primary data) 101: Memory scatter / gather mode (alternative data) 110: Peripheral memory scatter / gather mode (primary data) 111: Peripheral memory scatter / gather mode (alternative data)

Note 1: The setting value of <dst_size> must be the same as <src_size>.

Note 2: According to the settings of <dst_size> and <src_size>, the settings of <dst_inc> and <src_inc> are limited as shown below.

<src_inc>/<dst_inc>	<src_size>/<dst_size>		
	00 (1byte)	01 (2byte)	10 (4byte)
00(1byte)	o	-	-
01(2byte)	o	o	-
10(4byte)	o	o	o
No increments	o	o	o

8.4.3 Operation modes

This section describes the operation modes configured by channel_cfg<cycle_ctrl> of the channel control data.

8.4.3.1 Invalid setting

The DMA sets the operation mode invalid after the end of transfer. This operation prevents a transfer from being performed again. Also, the operation completes if an invalid data is read either in ping-pong mode, memory scatter / gather mode or peripheral scatter / gather mode.

8.4.3.2 Basic mode

A transfer starts by receiving a transfer request.

An arbitration is performed for every transfer configured by <R_power>. If a higher-priority request exits, the DMA switches a channel. If a transfer request for the operating channel is received, the transfer is continued.

After performing transfers for the number of times specified by <n_minus_1>, a transfer end interrupt is generated.

8.4.3.3 Automatic request mode

A single transfer request stops the transfer in this mode.

A transfer request starts a transfer.

For every transfers configured by <R_power>, a channel is switched if a higher-priority request is received. If not, the transfer is continued.

After performing transfers for the number of times specified by <n_minus_1>, a transfer end interrupt is generated.

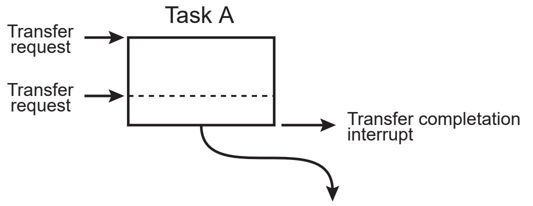
8.4.3.4 Ping-pong mode

In ping-pong mode, continuos DMA transfer is performed by switching primary data and alternative data. Reading a data that invalid ("000") is set to <cycle_ctrl> or setting the channel invalid stops the transfer operation and generates a transfer end interrupt.

Preparation:

Prepare primary data and alternative data, and set "1" to the bits of the channels corresponding to both DMAxdma_cfg<master_enable> and DMAxchnl_enable_set.

Task A: Primary data
<cycle_ctrl[2:0]> = "011"
(ping-pong mode)
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> =
"00_0000_0101"
(6 times)


Task A
Transfer request
Transfer request
Transfer completation interrupt

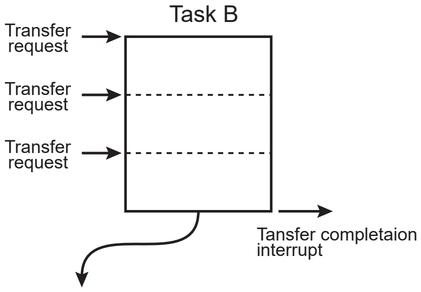
Receiving a transfer request, DMA performs a transfer four times and performs arbitration.

If there is no other high-priority requests, DMA performs remaining transfers twice toward a request for a transfer to the corresponding channels.

DMA generates a transfer end interrupt request and performs an arbitration.

After completing the Task A, a primary data for the Task C can be set.

Task B: Alternative data
<cycle_ctrl[2:0]> = "011"
<R_powe[3:0]r> = "0010"
(4 times)
<n_minus_1[9:0]> =
"00_0000_1011"
(12 times)


Task B
Transfer request
Transfer request
Transfer request
Tansfer completaion interrupt

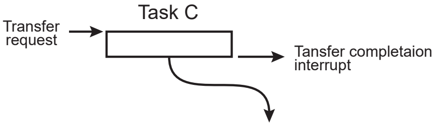
Receiving a transfer request, DMA performs a transfer four times and performs arbitration.

If there is no other high-priority requests, DMA performs transfers twice toward a request for a transfer to the corresponding channels.

DMA generates a transfer end interrupt request and performs an arbitration.

After completing the Task B, an alternative data for the Task D can be set.

Task C: Primary data
<cycle_ctrl[2:0]> = "011"
<R_power[3:0]> = "0001"
(2 times)
<n_minus_1[9:0]> =
"00_0000_0001"
(2 times)

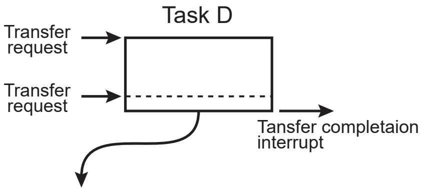

Task C
Transfer request
Tansfer completaion interrupt

Receiving a transfer request, DMA performs a transfer twice and performs arbitration.

DMA generates a transfer end interrupt request and performs an arbitration.

After completing the Task C, an alternative data for the Task E can be set.

Task D: Alternative data
<cycle_ctrl[2:0]> = "011"
<R_powe[3:0]r> = "0010"
(4 times)
<n_minus_1[9:0]> =
"00_0000_0100"
(5 times)

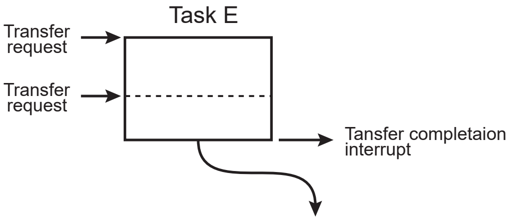

Task D
Transfer request
Transfer request
Tansfer completaion interrupt

Receiving a transfer request, DMA performs a transfer four times and performs arbitration.

If there is no other high-priority requests, DMA performs a transfer once toward a request for a transfer to the corresponding channels.

DMA generates a transfer end interrupt request and performs an arbitration.

Task E: Primary data
<cycle_ctrl[2:0]> = "011"
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> =
"00_0000_0110"
(7 times)

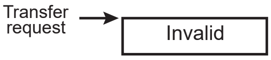

Task E
Transfer request
Transfer request
Tansfer completaion interrupt

Receiving a transfer request, DMA performs a transfer four times and performs arbitration.

If there is no other high-priority requests, DMA performs transfers three times toward a request for a transfer to the corresponding channels.

DMA generates a transfer end interrupt request and performs an arbitration.

Final: Alternative data
<cycle_ctrl[2:0]> = "000"
(invalid)


Transfer request
Invalid

Even receiving a transfer request, the operation stops because <cycle_ctrl> is set to invalid.

(The operation can be also stopped by setting the <cycle_ctrl> of the Task E to normal mode "001".)

8.4.3.5 Memory scatter / gather mode

In memory scatter / gather mode, primary data is used in order to transfer data for alternative data.

Receiving a transfer request, the DMA transfers four alternative data using primary data. If there is no new requests, it starts data transferring using alternative data. Then, it keeps transferring alternative data using primary data and transfer using alternative data, until either invalid setting ("000") of the <cycle_ctrl> or setting data of the basic mode ("001") is read. A new transfer request is not required during this period. After the transfer operation, an interrupt is generated.

The settings of the channel_cfg of primary data must be configured as shown below.

Table 8-4 Setting values of Memory scatter / gather mode (Primary data)

bit	bit symbol	Setting values	Description
[31:30]	dst_inc	10	4-byte increment is specified for transfer destination address.
[29:28]	dst_size	10	4 bytes are specified as transfer destination address.
[27:26]	src_inc	10	4-byte increment is specified for transfer source address.
[25:24]	src_size	10	4 bytes are specified as transfer source address.
[17:14]	R_power	0010	4 is specified as arbitration cycle.
[13:4]	n_minus_1	N	The number of alternative task to be prepared ×4 is specified.
[3]	next_useburst	0	"0" is specified in memory scatter / gather mode.
[2:0]	cycle_ctrl	100	Memory scatter / gather mode (Primary data) is specified. note)

Note: If transfers in the number of transfers set to the <n_minus_1> completes, invalid data "000" is automatically set.

Preparation:

Prepare primary data. Set "100" to <cycle_ctrl[2:0]> and set four task data 4 × 4 = 16 as the number of transfer <n_minus_1[9:0]>.

Set alternative data for the task A,B,C and D to the memory location which is set to the <src_data_end_ptr>.

Set "1" to bits of channels corresponding to DMAxdma_cfg<master_enable> and DMAxchnl_enable_set.

Copy A: Primary data

<cycle_ctrl[2:0]> = "100"

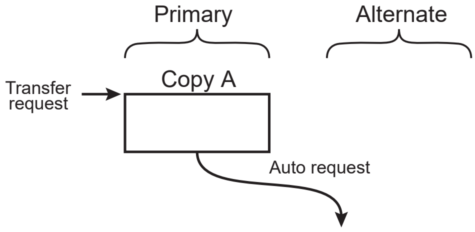
(Memory scatter / gather mode)

<R_power[3:0]> = "0010"

(4 times)

<n_minus_1[9:0]> = "00_0000_1111"

(16 times)


Primary
Alternate
Transfer request
Copy A
Auto request

Receiving a transfer request, DMA performs a transfer for alternative data of the task A for four times.

After completing the transfer, a transfer request is automatically generated and arbitration starts.

Task A: Alternative data

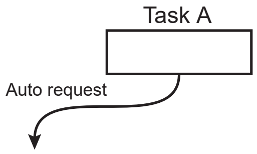
<cycle_ctrl[2:0]> = "100"

<R_power[3:0]> = "0010"

(4 times)

<n_minus_1[9:0]> = "00_0000_0010"

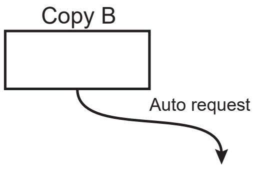
(3 times)


Task A
Auto request

DMA performs the task A.

After completing the transfer, a transfer request is automatically generated and arbitration starts.

Copy B: Primary data


Copy B
Auto request

DMA performs transfers for alternative data of the task B for four times.

After completing the transfer, a transfer request is automatically generated and arbitration starts.

Task B: Alternative data

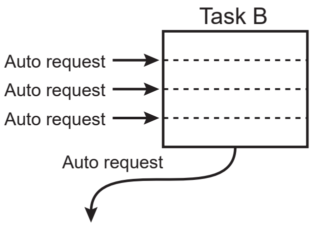
<cycle_ctrl[2:0]> = "100"

<R_power[3:0]> = "0001"

(2 times)

<n_minus_1[9:0]> = "00_0000_0111"

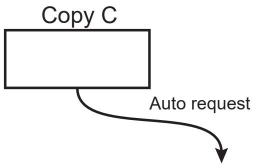
(8 times)


Task B
Auto request
Auto request
Auto request
Auto request

DMA performs the task B.

After completing the transfer, a transfer request is automatically generated and arbitration starts.

Copy C: Primary data


Copy C
Auto request

DMA performs transfers for alternative data of the task C for four times.

After completing the transfer, a transfer request is automatically generated and arbitration starts.

Task C: Alternative data

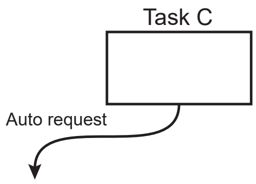
<cycle_ctrl[2:0]> = "100"

<R_power[3:0]> = "0011"

(8 times)

<n_minus_1[9:0]> = "00_0000_0100"

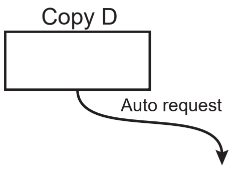
(5 times)


Task C
Auto request

DMA performs the task C.

After completing the transfer, a transfer request is automatically generated and arbitration starts.

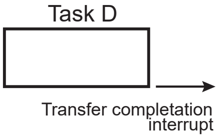
Copy D: Primary data


Copy D
Auto request

DMA performs transfers for alternative data of the task D for four times. DMA also sets "000" to <cycle_ctrl> of the primary data in order to set the next primary data invalid.

A transfer request is automatically generated and arbitration starts.

Task D: Alternative data
<cycle_ctrl[2:0]> = "001"
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> = "00_0000_0011"
(4 times)



DMA performs the task D.

Since <cycle_ctrl> is set to the basic mode "001", DMA generates a transfer end interrupt request after the end of the transfer, and completes the operation.

8.4.3.6 Peripheral scatter / gather mode

Primary data is used in order to transfer data for alternative data in the peripheral scatter / gather mode.

Receiving a transfer request, the DMA transfers four alternative data using primary data, and then starts transfer using alternative data.

After that, if a transfer request is generated, it starts alternative data transferring using primary data. Then, it keeps transferring alternative data using primary data and transfer using alternative data, until either invalid setting ("000") of the <cycle_ctrl> or setting data of the basic mode ("001") is read. A new transfer request is not required during this period. After the transfer operation, an interrupt is generated.

The settings of the channel_cfg of primary data must be configured as shown below.

Table 8-5 Fixed values in peripheral scatter / gather mode (Primary data)

bit	bit symbol	Setting value	Description
[31:30]	dst_inc	10	4-byte increment is specified for transfer destination address.
[29:28]	dst_size	10	4 bytes are specified as transfer destination address.
[27:26]	src_inc	10	4-byte increment is specified for transfer source address.
[25:24]	src_size	10	4 bytes are specified as transfer source address.
[17:14]	R_power	0010	
[13:4]	n_minus_1	N	The number of alternative task to be prepared ×4 is specified.
[2:0]	cycle_ctrl	110	Specify peripheral scatter / gather mode (Primary data).

Note: If transfers in the number of transfers set to the <n_minus_1> completes, invalid data "000" is automatically set.

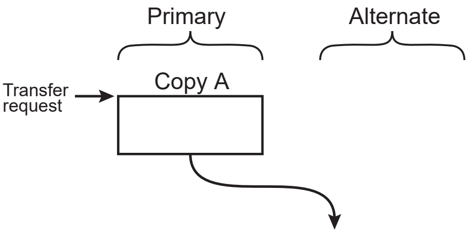
Preparation:

Prepare primary data. Set "110" to <cycle_ctrl[2:0]> and 4 × 4 = 16 for four tasks to the number of transfers <n_minus_1[9:0]>.

Set alternative data for the task A,B,C and D to the memory location which is set to the <src_data_end_ptr>.

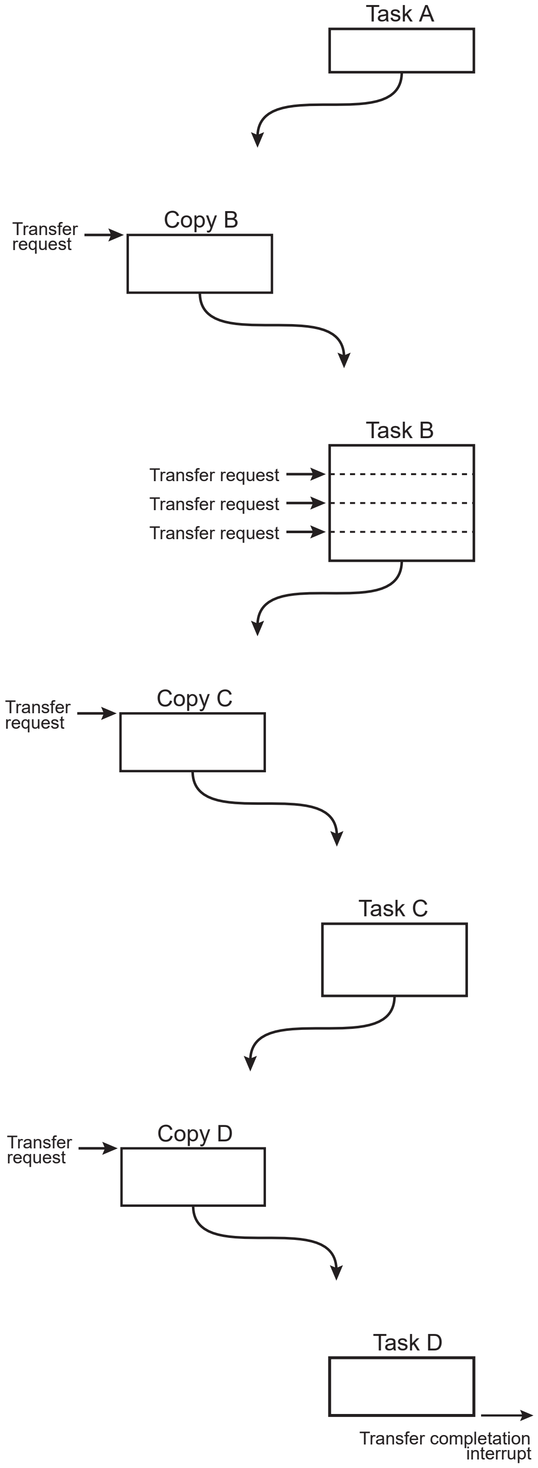
Set "1" to bits of channels corresponding to DMAxdma_cfg<master_enable> and DMAxchnl_enable_set.

Copy A: Primary data
<cycle_ctr[2:0]l> = "110"
(Peripheral scatter / gather)
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> = "00_0000_1111"
(16 times)



Receiving a transfer request, DMA performs transfers for alternative data of the task A for four times.

After completing the transfer, operation automatically moves on to the task A.



Task A: Alternative data

<cycle_ctrl[2:0]> = "111"

<R_power[3:0]> = "0010"

(4 times)

<n_minus_1[9:0]> = "00_0000_0010"

(3 times)

DMA performs the task A.

After completing the transfer, if a transfer request is sent from peripheral function and if it is high-priority request, the next operation starts.

Copy B: Primary data

DMA performs transfers for alternative data of the task B for four times.

After completing the transfer, processing of the task B automatically starts.

Task B: Alternative data

<cycle_ctrl[2:0]> = "111"

<R_power[3:0]> = "0001"

(2 times)

<n_minus_1[9:0]> = "00_0000_0111"

(8 times)

DMA performs the task B. Since an arbitration occurs every 2^R times of transfers, three times of transfer is required at least to complete the task B.

After completing the transfer, if a transfer request is sent from peripheral function and if it is high-priority request, the next operation starts

Copy C: Primary data

DMA performs transfers for alternative data of the Task C for four times.

After completing the transfer, operation automatically moves on to the task C.

Task C: Alternative data

<cycle_ctrl[2:0]> = "111"

<R_power[3:0]> = "0011"

(8 times)

<n_minus_1[9:0]> = "00_0000_0100"

(5 times)

DMA performs the task C.

After completing the transfer, if a transfer request is sent from peripheral function and if it is high-priority request, the next operation starts.

Copy D: Primary data

DMA performs transfers for alternative data of the Task D for four times. Also, DMA performs transfers for alternative data for four times. Sets "000" to <cycle_ctrl> of the primary data and makes the next primary data invalid.

The operation automatically moves on to the task D.

Task D: Alternative data

<cycle_ctrl[2:0]> = "001"

<R_power[3:0]> = "0010"

(4 times)

<n_minus_1[9:0]> = "00_0000_0011"

(4 times)

DMA performs the task D.

Since <cycle_ctrl> is set to the basic mode "001", DMA generates a transfer end interrupt request after the end of the transfer, and completes the operation.

9. Input / OUtput port

9.1 Port Function

9.1.1 Function List

TMPM367FDFG has 60 ports. These ports are also used as input/output pins for built-in peripheral functions.

Table 9-1 show lists of port functions.

Table 9-1 List of Port Function

Port	Pin name	Input / Output	Programmable Pull-up Pull-down	Schmitt input	Noise filter	Programmable Open-drain	Function pin
Port A							
	PA0	I/O	Pull-up Pull-down	o	−	o	TDO/SWV/DTR5
	PA1	I/O	Pull-up Pull-down	o	−	o	TMS/SWDIO/DSR5
	PA2	I/O	Pull-up Pull-down	o	−	o	TCK/SWCLK/RIN5
	PA3	I/O	Pull-up Pull-down	o	o(INT3 only)	o	TDI/DCD5/INT3
	PA4	I/O	Pull-up Pull-down	o	o	o	$\overline{TRST}$/$\overline{RTS5}$
	PA5	I/O	Pull-up Pull-down	o	−	o	TRACECLK/RXD5/IRIN5
	PA6	I/O	Pull-up Pull-down	o	−	o	TRACEDATA0/TXD5/IROUT5
	PA7	I/O	Pull-up Pull-down	o	−	o	TRACEDATA1/$\overline{CTS5}$/ SCLK3$\overline{CTS3}$/TB7OUT
Port B							
	PB0	I/O	Pull-up Pull-down	o	−	o	TRACEDATA2/TXD3
	PB1	I/O	Pull-up Pull-down	o	−	o	TRACEDATA3/RXD3
	PB2	I/O	Pull-up Pull-down	o	−	o	$\overline{WR}$/SP2CLK/MTOUT03/ MTTB3OUT
	PB3	I/O	Pull-up Pull-down	o	−	o	$\overline{RD}$/SP2DO/MTOUT13/MTTB3IN
	PB4	I/O	Pull-up Pull-down	−	o(INT7 only)	o	$\overline{CS0}$/SP2DI/$\overline{GEMG3}$/INT7
	PB5	I/O	Pull-up Pull-down	o	o(INT1 only)	o	ALE/SP2FSS/MT3IN/INT1
	PB6	Output	Pull-up Pull-down	o	−	o	$\overline{BELL}$/SCOUT/TB3OUT
Port E							
	PE0	I/O	Pull-up Pull-down	o	o(INT4 only)	o	A16/INT4/TB0IN
	PE1	I/O	Pull-up Pull-down	o	o(INT5 only)	o	RXD0/A17/INT5/TB1IN
	PE2	I/O	Pull-up Pull-down	o	−	o	TXD0/A18/TB1OUT
	PE3	I/O	Pull-up Pull-down	o	−	o	SCLK0/A19/$\overline{CTS0}$/TB0OUT
	PE4	I/O	Pull-up Pull-down	o	−	o	SCLK1/A20/$\overline{CTS1}$/TB2OUT
	PE5	I/O	Pull-up	o	−	o	TXD1/A21
	PE6	I/O	Pull-up Pull-down	o	−	o	RXD1/A6/A22
	PE7	I/O	Pull-up Pull-down	o	o(INT6 only)	o	A23/INT6/TB2IN
Port F							

Table 9-1 List of Port Function

Port	Pin name	Input / Output	Programmable Pull-up Pull-down	Schmitt input	Noise filter	Programmable Open-drain	Function pin
	PF0	I/O	Pull-up Pull-down	o	−	o	AD0/$\overline{\text{CTS4}}$
	PF1	I/O	Pull-up Pull-down	o	−	o	AD1/TXD4/IROUT4
	PF2	I/O	Pull-up Pull-down	o	−	o	AD2/RXD4/IRIN4
	PF3	I/O	Pull-up Pull-down	o	−	o	AD3/$\overline{\text{RTS4}}$
	PF4	I/O	Pull-up Pull-down	o	o(INT0 only)	o	AD4/INT0/DCD4
	PF5	I/O	Pull-up Pull-down	o	−	o	AD5/ENCZ/RIN4/SCK1
	PF6	I/O	Pull-up Pull-down	o	−	o	AD6/ENCB/DSR4/SI1/SCL1
	PF7	I/O	Pull-up Pull-down	o	−	o	AD7/ENCA/DTR4/SO1/SDA1
Port G							
	PG0	I/O	Pull-up Pull-down	o	−	o	AD8/MT0IN
	PG1	I/O	Pull-up Pull-down	o	−	o	AD9/$\overline{\text{EMG}}$/$\overline{\text{GEMG0}}$
	PG2	I/O	Pull-up Pull-down	o	−	o	AD10/ZO/MTOUT10/MTTB0IN
	PG3	I/O	Pull-up Pull-down	o	−	o	AD11/WO/MTOUT00/MTTB0OUT
	PG4	I/O	Pull-up Pull-down	o	−	o	AD12/YO/SP1CLK
	PG5	I/O	Pull-up Pull-down	o	−	o	AD13/VO/SP1DO
	PG6	I/O	Pull-up Pull-down	o	−	o	D14/AD14/XO/SP1DI
	PG7	I/O	Pull-up Pull-down	o	−	o	AD15/UO/SP1FSS
Port H							
	PH0	I/O	Pull-up Pull-down	o	−	o	$\overline{\text{BELH}}$/TB5OUT/MT2IN/ SO2/SDA2
	PH1	I/O	Pull-up Pull-down	o	−	o	$\overline{\text{CS1}}$/TB4OUT/$\overline{\text{GEMG2}}$/ SI2/SCL2
	PH2	I/O	Pull-up Pull-down	o	−	o	$\overline{\text{CS2}}$/MTOUT12/MTTB2IN/SCK2
	PH3	I/O	Pull-up Pull-down	o	−	o	$\overline{\text{CS3}}$/MTOUT02/MTTB2OUT/ EM_RPAUSE
Port I							
	PI0	I/O	Pull-up Pull-down	o	o(INT9 only)	o	INT9/AINA0
	PI1	I/O	Pull-up Pull-down	o	o(INTA only)	o	INTA/AINA1
	PI2	I/O	Pull-up Pull-down	o	o(INTB only)	o	INTB/AINA2

Table 9-1 List of Port Function

Port	Pin name	Input / Output	Programmable Pull-up Pull-down	Schmitt input	Noise filter	Programmable Open-drain	Function pin
	PI3	I/O	Pull-up Pull-down	o	o(INTC only)	o	INTC/$\overline{\text{DMAREQ}}$/AINA3
	PI4	I/O	Pull-up Pull-down	o	−	o	AINB0
	PI5	I/O	Pull-up Pull-down	o	−	o	AINB1
	PI6	I/O	Pull-up Pull-down	o	−	o	AINB2
	PI7	I/O	Pull-up Pull-down	o	−	o	AINB3
Port K							
	PK0	I/O	Pull-up Pull-down	o	o(INTD only)	o	USBDPON(INTD)
	PK1	I/O	Pull-up Pull-down	o	o(INT8 only)	o	SP0FSS/INT8/TB6OUT
	PK2	I/O	Pull-up Pull-down	−	−	o	USB_ECLK/SP0DI/ SO0/SDA0
	PK3	I/O	Pull-up Pull-down	o	−	o	SP0DO/ SI0/SCL0
	PK4	I/O	Pull-up Pull-down	o	−	o	RXIN/SP0CLK/SCK0
Port L							
	PL0	I/O	Pull-up Pull-down	o	o(INT2 only)	o	INT2/MT1IN/$\overline{\text{ADTRGA}}$
	PL1	I/O	Pull-up Pull-down	o	−	o	$\overline{\text{GEMG1}}$/$\overline{\text{DATRG}}$/RXD2
	PL2	I/O	Pull-up Pull-down	o	−	o	MTOUT11/MTTB1IN/TXD2
	PL3	I/O	Pull-up Pull-down	o	−	o	MTOUT01/MTTB1OUT/SCLK2/ $\overline{\text{CTS2}}$

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

9.1.2 Port Register General Description

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

- PxDATA: Port x data register

  This register reads/writes port data.

- PxCR: Port x output control register

  This register controls outputs.

  To enable/disable input with PxIE register.

- PxFRn: Port x function register n

  This register sets the functions.

  The assigned function can be enabled by setting "1".

- PxOD: Port x open-drain control register

  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: Port x pull-up control register

  This register controls programmable pull-ups.

- PxPDN: Port x pull-down control register

  This register controls programmable pull-downs.

- PxIE: Port x input control register

  This register controls inputs.

  To prevent through-current, the initial state is disabled to input.

9.1.3 Port Status during STOP mode

The CGSTBYCR<DRVE> of the clock/mode control part controls inputs/outputs during STOP1 mode. The CGSTBYCR<PTKEEP> controls inputs/outputs during STOP2 mode as well.

While PxIE or PxCR is enabled, if <DRVE>="1" is set or <PTKEEP>="0" → "1" is set, inputs or outputs is disabled even during STOP1/STOP2 mode. When <DRVE> is set to "0", inputs or outputs is disabled during STOP1 mode except partial ports even PxIE or PxCR is enabled. When transferring the normal mode to the STOP2 mode, <PTKEEP> bit must be set to "0" → "1" to sustain the port status.

Table 9-2 shows pin status in the STOP mode

Table 9-2 Pin status in the STOP mode

Function setting	Function	Input/Output	STOP1 mode		STOP2 mode
			<DRVE> = 1	<DRVE> = 0	<PTKEEP> = 1
Port	PAx to PLx	Input	Setting with PxIE[m]	Disabled	Status is sustained.
		Output	Setting with PxCR[m	Disabled	Status is sustained.
Debug function	$\overline{\text{TRST}}$, TCKI, TMS, TDI, SWCLK, SWDI	Input	Setting with PxIE[m]		Status is sustained.
	TDO, SWDO, SWV, TRACECLK, TRACEDATA0/1/2/3	Output	Setting with PxCR[m] and it is enabled when data is valid		Status is sustained.
External interrupt	INT0 to C	Input	Setting with PxIE[m]		Status is sustained.
SSP	SPxCLK, SPxFSS, SPxDO	Output	Setting with PxCR[m] and it is enabled when data is valid	Disabled	Status is sustained.
MPT (PMDmode)	UO, VO, WO, XO, YO, ZO	Output	Setting with PxCR[m] and it is enabled when data is valid	Setting with PxCR[m] and it is enabled when data is valid.	Status is sustained.
MPT (IGBTmode)	MTOUTxx	Output	Setting with PxCR[m] and it is enabled when data is valid.	Setting with PxCR[m] and it is enabled when data is valid.	Status is sustained.
Other than the above function	Other than the above function	Input	Setting with PxIE[m]	Disabled	Status is sustained.
	Other than the above function	Output	Setting with PxCR[m	Disabled	Status is sustained.

Note: In the above table, "x" indicates a specified port number; "m" indicates a specified bit; and "n" indicates function register numbers.

9.1.4 Precaution on exiting STOP1 / STOP2 mode using interrupts

When interrupt input is used to exit STOP1/STOP2, set functions using the function register and set inputs using the control register. In this case, interrupts can be input even CGSTBYCR<DRVE> in the clock mode control part is set to the setting where pins are not driven during STOP mode.

When ports are used as input ports, set the input control register.

9.1.5 Setting an external interrupt pin

Interrupts are enabled to input in the following two conditions while the control register is enabled; where CGSTBYCR<DRVE> bit is set to "1" in the STOP1/STOP2 mode, or where input is enabled by PxIE in the NORMAL/IDLE mode. Both conditions are regardless of function register settings. Do not enable unused interrupts when interrupts are set.

9.2 Function Details in Each Ports

This chapter describes details of registers in each port.

9.2.1 Port A (PA0 to PA7)

9.2.1.1 List of Port A register

Base Address = 0x400C_0000

register name		Address (Base+)
Port A data register	PADATA	0x0000
Port A output control register	PACR	0x0004
Port A function register 1	PAFR1	0x0008
Port A function register 2	PAFR2	0x000C
Port A function register 3	PAFR3	0x0010
Port A function register 4	PAFR4	0x0014
Port A function register 5	PAFR5	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port A open-drain control register	PAOD	0x0028
Port A pull-up control register	PAPUP	0x002C
Port A pull-down control register	PAPDN	0x0030
Reserved	-	0x0034
Port A input control register	PAIE	0x0038

Note 1: Do not access the addresses described as "Reserved".

Note 2: If PA1 and PA0 are set to TMS/SWDIO and TDO/SWV respectively, outputs are kept as valid regardless of CGSTBYCR<DRVE>/<PTKEEP> setting during STOP1/STOP2 mode.

9.2.1.2 PADATA (Port A data register)

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

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

9.2.1.3 PACR (Port A output control register)

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

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

9.2.1.4 PAFR1 (Port A function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PA7F1	PA6F1	PA5F1	PA4F1	PA3F1	PA2F1	PA1F1	PA0F1
After reset	0	0	0	1	1	1	1	1

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PA7F1	R/W	0: PORT 1: TRACEDATA1
6	PA6F1	R/W	0: PORT 1: TRACEDATA0
5	PA5F1	R/W	0: PORT 1: TRACECLK
4	PA4F1	R/W	0: PORT 1: $\overline{\text{TRST}}$
3	PA3F1	R/W	0: PORT 1: TDI
2	PA2F1	R/W	0: PORT 1: TCK/SWCLK
1	PA1F1	R/W	0: PORT 1: TMS/SWDIO
0	PA0F1	R/W	0: PORT 1: TDO/SWV

9.2.1.5 PAFR2 (Port A function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PA7F2	PA6F2	PA5F2	PA4F2	PA3F2	PA2F2	PA1F2	PA0F2
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PA7F2	R/W	0: PORT 1: $\overline{\text{CTS5}}$
6	PA6F2	R/W	0: PORT 1: TXD5
5	PA5F2	R/W	0: PORT 1: RXD5
4	PA4F2	R/W	0: PORT 1: $\overline{\text{RTS5}}$
3	PA3F2	R/W	0: PORT 1: DCD5
2	PA2F2	R/W	0: PORT 1: RIN5
1	PA1F2	R/W	0: PORT 1: DSR5
0	PA0F2	R/W	0: PORT 1: DTR5

9.2.1.6 PAFR3 (Port A function 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	PA7F3	PA6F3	PA5F3	-	PA3F3	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PA7F3	R/W	0: PORT 1: SCLK3
6	PA6F3	R/W	0: PORT 1: IROUT5
5	PA5F3	R/W	0: PORT 1: IRIN5
4	−	R	Read as "0".
3	PA3F3	R/W	0: PORT 1: INT3
2-0	−	R	Read as "0".

9.2.1.7 PAFR4 (Port A function 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	PA7F4	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PA7F4	R/W	0: PORT 1: $\overline{\text{CTS3}}$
6-0	−	R	Read as "0".

9.2.1.8 PAFR5 (Port A function register 5)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PA7F5	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PA7F5	R/W	0: PORT 1: TB7OUT
6-0	−	R	Read as "0".

9.2.1.9 PAOD (Port A open-drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PA7OD	PA6OD	PA5OD	PA4OD	PA3OD	PA2OD	PA1OD	PA0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PA7OD-PA0OD	R/W	0: Push-pull output 1: Open-drain output

9.2.1.10 PAPUP (Port A pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PA7UP	PA6UP	PA5UP	PA4UP	PA3UP	PA2UP	PA1UP	PA0UP
After reset	0	0	0	1	1	0	1	0

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

9.2.1.11 PAPDN (Port A pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PA7DN	PA6DN	PA5DN	PA4DN	PA3DN	PA2DN	PA1DN	PA0DN
After reset	0	0	0	0	0	1	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PI7DN-PA0DN	R/W	Pull-down 0: Disable 1: Enable

9.2.1.12 PAIE (Port A input control register)

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

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

9.2.2 Port B (PB0 to PB6)

9.2.2.1 List of Port B register

Base Address = 0x400C_0100

register name		Address (Base+)
Port B data register	PBDATA	0x0000
Port B output control register	PBCR	0x0004
Port B function register 1	PBFR1	0x0008
Port B function register 2	PBFR2	0x000C
Port B function register 3	PBFR3	0x0010
Port B function register 4	PBFR4	0x0014
Reserved	-	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port B open-drain control register	PBOD	0x0028
Port B pull-up control register	PBPUP	0x002C
Port B pull-down control register	PBPDN	0x0030
Reserved	-	0x0034
Port B input controlregister	PBIE	0x0038

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

9.2.2.2 PBDATA (Port B data register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-0	PB6-PB0	R/W	Port B data register

9.2.2.3 PBCR (Port B output control register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-0	PB6C-PB0C	R/W	Output 0: Disable 1: Enable

9.2.2.4 PBFR1 (Port B function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PB6F1	PB5F1	PB4F1	PB3F1	PB2F1	PB1F1	PB0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6	PB6F1	R/W	0: PORT 1: $\overline{\text{BELL}}$
5	PB5F1	R/W	0: PORT 1: ALE
4	PB4F1	R/W	0: PORT 1: $\overline{\text{CS0}}$
3	PB3F1	R/W	0: PORT 1: $\overline{\text{RD}}$
2	PB2F1	R/W	0: PORT 1: $\overline{\text{WR}}$
1	PB1F1	R/W	0: PORT 1: TRACEDATA3
0	PB0F1	R/W	0: PORT 1: TRACEDATA2

9.2.2.5 PBFR2 (Port B function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PB6F2	PB5F2	PB4F2	PB3F2	PB2F2	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6	PB6F2	R/W	0: PORT 1: SCOUT
5	PB5F2	R/W	0: PORT 1: SP2FSS
4	PB4F2	R/W	0: PORT 1: SP2DI
3	PB3F2	R/W	0: PORT 1: SP2DO
2	PB2F2	R/W	0: PORT 1: SP2CLK
1-0	−	R	Read as "0".

9.2.2.6 PBFR3 (Port B function 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	-	-	PB5F3	PB4F3	PB3F3	PB2F3	PB1F3	PB0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as "0".
5	PB5F3	R/W	0: PORT 1: MT3IN
4	PB4F3	R/W	0: PORT 1: $\overline{\text{GEMG3}}$
3	PB3F3	R/W	0: PORT 1: MTOUT13
2	PB2F3	R/W	0: PORT 1: MTOUT03
1	PB1F3	R/W	0: PORT 1: RXD3
0	PB0F3	R/W	0: PORT 1: TXD3

9.2.2.7 PBFR4 (Port B function 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	-	PB6F4	PB5F4	PB4F4	PB3F4	PB2F4	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6	PB6F4	R/W	0: PORT 1: TB3OUT
5	PB5F4	R/W	0: PORT 1: INT1
4	PB4F4	R/W	0: PORT 1: INT7
3	PB3F4	R/W	0: PORT 1: MTTB3IN
2	PB2F4	R/W	0: PORT 1: MTTB3OUT
1-0	−	R	Read as "0".

9.2.2.8 PBOD (Port B open-drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PB6OD	PB5OD	PB4OD	PB3OD	PB2OD	PB1OD	PB0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-0	PB6OD-PB0OD	R/W	0: Push-pull output 1: Open-drain output

9.2.2.9 PBPUP (Port B pull-up control register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-0	PB6UP-PB0UP	R/W	Pull-up 0: Disable 1: Enable

9.2.2.10 PBPDN (Port B pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PB6DN	PB5DN	PB4DN	PB3DN	PB2DN	PB1DN	PB0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-0	PB6DN-PB0DN	R/W	Pull-down 0: Disable 1: Enable

9.2.2.11 PBIE (Port B input control register)

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

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as "0".
5-0	PB5IE-PB0IE	R/W	Input 0: Disable 1: Enable

9.2.3 Port E (PE0 to PE7)

9.2.3.1 List of Port E register

Base Address = 0x400C_0400

register name		Address (Base+)
Port E data register	PEDATA	0x0000
Port E output control register	PECR	0x0004
Port E function register 1	PEFR1	0x0008
Reserved	-	0x000C
Port E function register 3	PEFR3	0x0010
Port E function register 4	PEFR4	0x0014
Port E function register 5	PEFR5	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port E open-drain control register	PEOD	0x0028
Port E pull-up control register	PEPUP	0x002C
Port E pull-down control register	PEPDN	0x0030
Reserved	-	0x0034
Port E input controlregister	PEIE	0x0038

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

9.2.3.2 PEDATA (Port E data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PE7-PE0	R/W	Port E data register

9.2.3.3 PECR (Port E output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PE7C	PE6C	PE5C	PE4C	PE3C	PE2C	PE1C	PE0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PE7C-PE0C	R/W	Output 0: Disable 1: Enable

9.2.3.4 PEFR1 (Port E function register 1)

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

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

9.2.3.5 PEFR3 (Port E function 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	PE7F3	PE6F3	PE5F3	PE4F3	PE3F3	PE2F3	PE1F3	PE0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PE7F3	R/W	0: PORT 1: A23
6	PE6F3	R/W	0: PORT 1: A22
5	PE5F3	R/W	0: PORT 1: A21
4	PE4F3	R/W	0: PORT 1: A20
3	PE3F3	R/W	0: PORT 1: A19
2	PE2F3	R/W	0: PORT 1: A18
1	PE1F3	R/W	0: PORT 1: A17
0	PE0F3	R/W	0: PORT 1: A16

9.2.3.6 PEFR4 (Port E function 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	PE7F4	-	-	PE4F4	PE3F4	-	PE1F4	PE0F4
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PE7F4	R/W	0: PORT 1: INT6
6-5	-	R	Read as "0".
4	PE4F4	R/W	0: PORT 1: $\overline{\text{CTS1}}$
3	PE3F4	R/W	0: PORT 1: $\overline{\text{CTS0}}$
2	-	R	Read as "0".
1	PE1F4	R/W	0: PORT 1: INT5
0	PE0F4	R/W	0: PORT 1: INT4

9.2.3.7 PEFR5 (Port E function register 5)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PE7F5	-	-	PE4F5	PE3F5	PE2F5	PE1F5	PE0F5
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PE7F5	R/W	0: PORT 1: TB2IN
6-5	-	R	Read as "0".
4	PE4F5	R/W	0: PORT 1: TB2OUT
3	PE3F5	R/W	0: PORT 1: TB0OUT
2	PE2F5	R/W	0: PORT 1: TB1OUT
1	PE1F5	R/W	0: PORT 1: TB1IN
0	PE0F5	R/W	0: PORT 1: TB0IN

9.2.3.8 PEOD (Port E open-drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PE7OD	PE6OD	PE5OD	PE4OD	PE3OD	PE2OD	PE1OD	PE0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PE7OD-PE0OD	R/W	0: Push-pull output 1: Open-drain output

9.2.3.9 PEPUP (Port E pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PE7UP	PE6UP	PE5UP	PE4UP	PE3UP	PE2UP	PE1UP	PE0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PE7UP-PE0UP	R/W	Pull-up 0: Disable 1: Enable

9.2.3.10 PEPDN (Port E pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PE7DN	PE6DN	PE5DN	PE4DN	PE3DN	PE2DN	PE1DN	PE0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PE7DN-PE0DN	R/W	Pull-down 0: Disable 1: Enable

9.2.3.11 PEIE (Port E input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PE7IE	PE6IE	PE5IE	PE4IE	PE3IE	PE2IE	PE1IE	PE0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PE7IE-PE0IE	R/W	Input 0: Disable 1: Enable

9.2.4 Port F (PF0 to PF7)

9.2.4.1 List of Port F register

Base Address = 0x400C_0500

register name		Address (Base+)
Port F data register	PFDATA	0x0000
Port F output control register	PFCR	0x0004
Port F function register 1	PFFR1	0x0008
Port F function register 2	PFFR2	0x000C
Port F function register 3	PFFR3	0x0010
Port F function register 4	PFFR4	0x0014
Reserved	-	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port F open-drain control register	PFOD	0x0028
Port F pull-up control register	PFPUP	0x002C
Port F pull-down control register	PFPDN	0x0030
Reserved	-	0x0034
Port F input controlregister	PFIE	0x0038

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

9.2.4.2 PFDATA (Port F data register)

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

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

9.2.4.3 PFCR (Port F output control register)

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

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

9.2.4.4 PFFR1 (Port F function register 1)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PF7F1	R/W	0: PORT 1: AD7
6	PF6F1	R/W	0: PORT 1: AD6
5	PF5F1	R/W	0: PORT 1: AD5
4	PF4F1	R/W	0: PORT 1: AD4
3	PF3F1	R/W	0: PORT 1: AD3
2	PF2F1	R/W	0: PORT 1: AD2
1	PF1F1	R/W	0: PORT 1: AD1
0	PF0F1	R/W	0: PORT 1: AD0

9.2.4.5 PFFR2 (Port F function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PF7F2	PF6F2	PF5F2	PF4F2	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PF7F2	R/W	0: PORT 1: ENCA
6	PF6F2	R/W	0: PORT 1: ENCB
5	PF5F2	R/W	0: PORT 1: ENCZ
4	PF4F2	R/W	0: PORT 1: INT0
3-0	−	R	Read as "0".

9.2.4.6 PFFR3 (Port F function 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	PF7F3	PF6F3	PF5F3	PF4F3	PF3F3	PF2F3	PF1F3	PF0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PF7F3	R/W	0: PORT 1: DTR4
6	PF6F3	R/W	0: PORT 1: DSR4
5	PF5F3	R/W	0: PORT 1: RIN4
4	PF4F3	R/W	0: PORT 1: DCD4
3	PF3F3	R/W	0: PORT 1: $\overline{\text{RTS4}}$
2	PF2F3	R/W	0: PORT 1: RXD4
1	PF1F3	R/W	0: PORT 1: TXD4
0	PF0F3	R/W	0: PORT 1: $\overline{\text{CTS4}}$

9.2.4.7 PFFR4 (Port F function 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	PF7F4	PF6F4	PF5F4	-	-	PF1F4	PF0F4	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PF7F4	R/W	0: PORT 1: SO1/SDA1
6	PF6F4	R/W	0: PORT 1: SI1/SCL1
5	PF5F4	R/W	0: PORT 1: SCK1
4-3	-	R	Read as "0".
2	PF2F4	R/W	0: PORT 1: IRIN4
1	PF1F4	R/W	0: PORT 1: IROUT4
0	−	R	Read as "0".

9.2.4.8 PFOD (Port F open-drain control register)

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

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

9.2.4.9 PFPUP (Port F pull-up control register)

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

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

9.2.4.10 PFPDN (Port F pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PF7DN	PF6DN	PF5DN	PF4DN	PF3DN	PF2DN	PF1DN	PF0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PF7DN-PF0DN	R/W	Pull-down 0: Disable 1: Enable

9.2.4.11 PFIE (Port F input control register)

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

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

9.2.5 Port G (PG0 to PG7)

9.2.5.1 List of Port G register

Base Address = 0x400C_0600

register name		Address (Base+)
Port G data register	PGDATA	0x0000
Port G output control register	PGCR	0x0004
Port G function register 1	PGFR1	0x0008
Port G function register 2	PGFR2	0x000C
Port G function register 3	PGFR3	0x0010
Port G function register 4	PGFR4	0x0014
Reserved	-	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port G open-drain control register	PGOD	0x0028
Port G pull-up control register	PGPUP	0x002C
Port G pull-down control register	PGPDN	0x0030
Reserved	-	0x0034
Port G input controlregister	PGIE	0x0038

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

9.2.5.2 PGDATA (Port G data register)

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

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

9.2.5.3 PGCR (Port G output control register)

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

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

9.2.5.4 PGFR1 (Port G function register 1)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PG7F1	R/W	0: PORT 1: AD15
6	PG6F1	R/W	0: PORT 1: AD14
5	PG5F1	R/W	0: PORT 1: AD13
4	PG4F1	R/W	0: PORT 1: AD12
3	PG3F1	R/W	0: PORT 1: AD11
2	PG2F1	R/W	0: PORT 1: AD10
1	PG1F1	R/W	0: PORT 1: AD9
0	PG0F1	R/W	0: PORT 1: AD8

9.2.5.5 PGFR2 (Port G function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7F2	PG6F2	PG5F2	PG4F2	PG3F2	PG2F2	PG1F2	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PG7F2	R/W	0: PORT 1: UO
6	PG6F2	R/W	0: PORT 1: XO
5	PG5F2	R/W	0: PORT 1: VO
4	PG4F2	R/W	0: PORT 1: YO
3	PG3F2	R/W	0: PORT 1: WO
2	PG2F2	R/W	0: PORT 1: ZO
1	PG1F2	R/W	0: PORT 1: $\overline{\text{EMG}}$
0	−	R	Read as "0".

9.2.5.6 PGFR3 (Port G function 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	PG7F3	PG6F3	PG5F3	PG4F3	PG3F3	PG2F3	PG1F3	PG0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PG7F3	R/W	0: PORT 1: SP1FSS
6	PG6F3	R/W	0: PORT 1: SP1DI
5	PG5F3	R/W	0: PORT 1: SP1DO
4	PG4F3	R/W	0: PORT 1: SP1CLK
3	PG3F3	R/W	0: PORT 1: MTOUT00
2	PG2F3	R/W	0: PORT 1: MTOUT10
1	PG1F3	R/W	0: PORT 1: $\overline{\text{GEMG0}}$
0	PG0F3	R/W	0: PORT 1: MT0IN

9.2.5.7 PGFR4 (Port G function 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	-	-	-	-	PG3F4	PG2F4	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3	PG3F4	R/W	0: PORT 1: MTTB0OUT
2	PG2F4	R/W	0: PORT 1: MTTB0IN
1-0	−	R	Read as "0".

9.2.5.8 PGOD (Port G open-drain control register)

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

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

9.2.5.9 PGPUP (Port G pull-up control register)

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

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

9.2.5.10 PGPDN (Port G pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7DN	PG6DN	PG5DN	PG4DN	PG3DN	PG2DN	PG1DN	PG0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PG7DN-PG0DN	R/W	Pull-down 0: Disable 1: Enable

9.2.5.11 PGIE (Port G input control register)

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

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

9.2.6 Port H (PH0 to PH3)

9.2.6.1 List of Port H register

Base Address = 0x400C_0700

register name		Address (Base+)
Port H data resister	PHDATA	0x0000
Port H output control resister	PHCR	0x0004
Port H function resister 1	PHFR1	0x0008
Port H function resister 2	PHFR2	0x000C
Port H function resister 3	PHFR3	0x0010
Port H function resister 4	PHFR4	0x0014
Port H function resister 5	PHFR5	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port H open-drain control resister	PHOD	0x0028
Port H pull-up control resister	PHPUP	0x002C
Port H pull-down control resister	PHPDN	0x0030
Reserved	-	0x0034
Port H input controlresister	PHIE	0x0038

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

9.2.6.2 PHDATA (Port H data register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	−	R/W	Write as "0".
3-0	PH3-PH0	R/W	Port H data register

9.2.6.3 PHCR (Port H output control register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	−	R/W	Write as "0".
3-0	PH3C-PH0C	R/W	Output 0: Disable 1: Enable

9.2.6.4 PHFR1 (Port H function register 1)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-5	−	R/W	Write as "0".
4	-	R	Read as "0".
3	PH3F1	R/W	0: PORT 1: $\overline{\text{CS3}}$
2	PH2F1	R/W	0: PORT 1: $\overline{\text{CS2}}$
1	PH1F1	R/W	0: PORT 1: $\overline{\text{CS1}}$
0	PH0F1	R/W	0: PORT 1: $\overline{\text{BELH}}$

9.2.6.5 PHFR2 (Port H function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PH1F2	PH0F2
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 as "0".
6	-	R	Read as "0".
5	−	R/W	Write as "0".
4	-	R	Read as "0".
3-2	−	R/W	Write as "0".
1	PH1F2	R/W	0: PORT 1: TB4OUT
0	PH0F2	R/W	0: PORT 1: TB5OUT

9.2.6.6 PHFR3 (Port H function 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	-	-	-	-	PH3F3	PH2F3	PH1F3	PH0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3	PH3F3	R/W	0: PORT 1: MTOUT02
2	PH2F3	R/W	0: PORT 1: MTOUT12
1	PH1F3	R/W	0: PORT 1: $\overline{\text{GEMG2}}$
0	PH0F3	R/W	0: PORT 1: MT2IN

9.2.6.7 PHFR4 (Port H function 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	-	-	-	-	PH3F4	PH2F4	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
3	PH3F4	R/W	0: PORT 1: MTTB2OUT
2	PH2F4	R/W	0: PORT 1: MTTB2IN
1-0	−	R	Read as "0".

9.2.6.8 PHFR5 (Port H function register 5)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PH2F5	PH1F5	PH0F5
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4-3	−	R/W	Write as "0".
2	PH2F5	R/W	0: PORT 1: SCK2
1	PH1F5	R/W	0: PORT 1: SI2/SCL2
0	PH0F5	R/W	0: PORT 1: SO2/SDA2

9.2.6.9 PHOD (Port H open-drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PH3OD	PH2OD	PH1OD	PH0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	−	R/W	Write as "0".
3-0	PH3OD-PH0OD	R/W	0: Push-pull output 1: Open-drain output

9.2.6.10 PHPUP (Port H pull-up control register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	−	R/W	Write as "0".
3-0	PH3UP-PH0UP	R/W	Pull-up 0: Disable 1: Enable

9.2.6.11 PHPDN (Port H pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PH3DN	PH2DN	PH1DN	PH0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	−	R/W	Write as "0".
3-0	PH3DN-PH0DN	R/W	Pull-down 0: Disable 1: Enable

9.2.6.12 PHIE (Port H input control register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	−	R/W	Write as "0".
3-0	PH3IE-PH0IE	R/W	Input 0: Disable 1: Enable

9.2.7 Port I (PI0 to PI7)

9.2.7.1 List of Port I register

Base Address = 0x400C_0800

register name		Address (Base+)
Port I data resister	PIDATA	0x0000
Port I output control resister	PICR	0x0004
Port I function resister 1	PIFR1	0x0008
Port I function resister 2	PIFR2	0x000C
Reserved	-	0x0010
Reserved	-	0x0014
Reserved	-	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port B open-drain control resister	PIOD	0x0028
Port B pull-up control resister	PIPUP	0x002C
Port B pull-down control resister	PIPDN	0x0030
Reserved	-	0x0034
Port B input controlresister	PIIE	0x0038

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

9.2.7.2 PIDATA (Port I data register)

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

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

9.2.7.3 PICR (Port I output control register)

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

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

9.2.7.4 PIFR1 (Port I function register 1)

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

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3	PI3F1	R/W	0: PORT 1: INTC
2	PI2F1	R/W	0: PORT 1: INTB
1	PI1F1	R/W	0: PORT 1: INTA
0	PI0F1	R/W	0: PORT 1: INT9

9.2.7.5 PIFR2 (Port I function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PI3F2	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3	PI3F2	R/W	0: PORT 1: $\overline{\text{DMAREQ}}$
2-0	−	R	Read as "0".

9.2.7.6 PIOD (Port I open-drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PI7OD	PI6OD	PI5OD	PI4OD	PI3OD	PI2OD	PI1OD	PI0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PI7OD-PI0OD	R/W	0: Push-pull output 1: Open-drain output

9.2.7.7 PIPUP (Port I pull-up control register)

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

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

9.2.7.8 PIPDN (Port I pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PI7DN	PI6DN	PI5DN	PI4DN	PI3DN	PI2DN	PI1DN	PI0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PI7DN-PI0DN	R/W	Pull-down 0: Disable 1: Enable

9.2.7.9 PIIE (Port I input control register)

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

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

9.2.8 Port K (PK0 to PK4)

Only when input is enabled, PK0 tolerates 5V input. Note that this pin cannot be pulled up over the power supply voltage when using as open-drain output.

9.2.8.1 List of Port K regidter

Base Address = 0x400C_0A00

register name		Address (Base+)
Port K data resister	KIDATA	0x0000
Port K output control resister	PKCR	0x0004
Port K function resister 1	PKFR1	0x0008
Port K function resister 2	PKFR2	0x000C
Port K function resister 3	PKFR3	0x0010
Port K function resister 4	PKFR4	0x0014
Reserved	-	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port K open-drain control resister	PKOD	0x0028
Port K pull-up control resister	PKPUP	0x002C
Port K pull-down control resister	PKPDN	0x0030
Reserved	-	0x0034
Port K input controlresister	PKIE	0x0038

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

9.2.8.2 PKDATA (Port K data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PK4	PK3	PK2	PK1	PK0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4-0	PK4-PK0	R/W	Port K data register

9.2.8.3 PKCR (Port K output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PK4C	PK3C	PK2C	PK1C	PK0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4-0	PK4C-PK0C	R/W	Output 0: Disable 1: Enable

9.2.8.4 PKFR1 (Port K function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PK4F1	-	PK2F1	-	PK0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4	PK4F1	R/W	0: PORT 1: RXIN
3	−	R/W	Write as "0".
2	PK2F1	R/W	0: PORT 1: USB_ECLK
1	−	R/W	Write as "0".
0	PK0F1	R/W	0: PORT 1: USBDPON (INTD)

Note: PK0 is used as USBDPON input or output port when USB device controller is used. It is used as INTD input or input / output port when USB device controller is not used.

9.2.8.5 PKFR2 (Port K function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PK4F2	PK3F2	PK2F2	PK1F2	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4	PK4F2	R/W	0: PORT 1: SP0CLK
3	PK3F2	R/W	0: PORT 1: SP0DO
2	PK2F2	R/W	0: PORT 1: SP0DI
1	PK1F2	R/W	0: PORT 1: SP0FSS
0	−	R	Read as "0".

9.2.8.6 PKFR3 (Port K function 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	-	-	-	PK4F3	PK3F3	PK2F3	PK1F3	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4	PK4F3	R/W	0: PORT 1: SCK0
3	PK3F3	R/W	0: PORT 1: SI0/SCL0
2	PK2F3	R/W	0: PORT 1: SO0/SDA0
1	PK1F3	R/W	0: PORT 1: INT8
0	−	R	Read as "0".

9.2.8.7 PKFR4 (Port K function 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	-	-	-	-	-	-	PK1F4	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	PK1F4	R/W	0: PORT 1: TB6OUT
0	−	R	Read as "0".

9.2.8.8 PKOD (Port K open-drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PK4OD	PK3OD	PK2OD	PK1OD	PK0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4-0	PK4OD-PK0OD	R/W	0: Push-pull output 1: Open-drain output

Note: Only when input is enabled, PK0 tolerates 5V input. Note that this pin cannot be pulled up over the power supply voltage when using as open-drain output.

9.2.8.9 PKPUP (Port K pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PK4UP	PK3UP	PK2UP	PK1UP	PK0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4-0	PK4UP-PK0UP	R/W	Pull-up 0: Disable 1: Enable

9.2.8.10 PKPDN (Port K pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PK4DN	PK3DN	PK2DN	PK1DN	PK0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4-0	PK4DN-PK0DN	R/W	Pull-down 0: Disable 1: Enable

9.2.8.11 PKIE (Port K input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PK4IE	PK3IE	PK2IE	PK1IE	PK0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4-0	PK4IE-PK0IE	R/W	Input 0: Disable 1: Enable

Note: PK0 is used as USBDPON input or output port when USB device controller is used. It is used as INTD input or input / output port when USB device controller is not used.

9.2.9 Port L (PL0 to PL3)

9.2.9.1 List of Port L register

Base Address = 0x400C_0B00

register name		Address (Base+)
Port L data resister	LIDATA	0x0000
Port L output control resister	PLCR	0x0004
Reserved	-	0x0008
Port L function resister 2	PLFR2	0x000C
Port L function resister 3	PLFR3	0x0010
Port L function resister 4	PLFR4	0x0014
Port L function resister 5	PLFR5	0x0018
Port L function resister 6	PLFR6	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port L open-drain control resister	PLOD	0x0028
Port L pull-up control resister	PLPUP	0x002C
Port L pull-down control resister	PLPDN	0x0030
Reserved	-	0x0034
Port L input controlresister	PLIE	0x0038

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

9.2.9.2 PLDATA (Port L data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PL3	PL2	PL1	PL0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	−	R/W	Write as "0".
3-0	PL3-PL0	R/W	Port L data register

9.2.9.3 PLCR (Port L output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PL3C	PL2C	PL1C	PL0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	−	R/W	Write as "0".
3-0	PL3C-PL0C	R/W	Output 0: Disable 1: Enable

9.2.9.4 PLFR2 (Port L function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	PL0F2
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-1	−	R/W	Write as "0".
0	PL0F2	R/W	0: PORT 1: INT2

9.2.9.5 PLFR3 (Port L function 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	-	-	-	-	PL3F3	PL2F3	PL1F3	PL0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3	PL3F3	R/W	0: PORT 1: MTOUT01
2	PL2F3	R/W	0: PORT 1: MTOUT11
1	PL1F3	R/W	0: PORT 1: $\overline{\text{GEMG1}}$
0	PL0F3	R/W	0: PORT 1: MT1IN

9.2.9.6 PLFR4 (Port L function 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	-	-	-	-	PL3F4	PL2F4	PL1F4	PL0F4
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3	PL3F4	R/W	0: PORT 1: MTTB1OUT
2	PL2F4	R/W	0: PORT 1: MTTB1IN
1	PL1F4	R/W	0: PORT 1: $\overline{\text{DATRG}}$
0	PL0F4	R/W	0: PORT 1: $\overline{\text{ADTRGA}}$

9.2.9.7 PLFR5 (Port L function register 5)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PL3F5	PL2F5	PL1F5	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3	PL3F5	R/W	0: PORT 1: SCLK2
2	PL2F5	R/W	0: PORT 1: TXD2
1	PL1F5	R/W	0: PORT 1: RXD2
0	−	R	Read as "0".

9.2.9.8 PLFR6 (Port L function 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	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PL3F6	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3	PL3F6	R/W	0: PORT 1: $\overline{\text{CTS2}}$
2-0	−	R	Read as "0".

9.2.9.9 PLOD (Port L open-drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PL3OD	PL2OD	PL1OD	PL0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	−	R/W	Write as "0".
3-0	PL3OD-PL0OD	R/W	0: Push-pull output 1: Open-drain output

9.2.9.10 PLPUP (Port L pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PL3UP	PL2UP	PL1UP	PL0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	−	R/W	Write as "0".
3-0	PL3UP-PL0UP	R/W	Pull-up 0: Disable 1: Enable

9.2.9.11 PLPDN (Port L pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PL3DN	PL2DN	PL1DN	PL0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	−	R/W	Write as "0".
3-0	PL3DN-PL0DN	R/W	Pull-down 0: Disable 1: Enable

9.2.9.12 PLIE (Port L input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PL3IE	PL2IE	PL1IE	PL0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	−	R/W	Write as "0".
3-0	PL3IE-PL0IE	R/W	Input 0: Disable 1: Enable

9.3 Block Diagram of Port

9.3.1 Port Types

The ports are classified as shown below. Please refer to the following pages for the block diagrams of each port type. Dot lines in the figure indicate the part of the equivalent circuit described in the "Block diagrams of ports".

Table 9-3 Function Lists

Type	GP Port	Function	Analog	Pull-up	Pull-down	Programmable open-drain	Note
FT1	I/O	I/O	−	R	R	o	-
FT2	I/O	I/O	−	R	R	o	Function output triggered by enable signal
FT3	I/O	I/O	−	R	R	o	Function output triggered by enable signal
FT4	I/O	Input(int)	−	R	R	o	with Noise filter
FT5	I/O	Input	o	R	R	o	Analog input
FT6	Output	Output	−	EnR	R	o	$\overline{\text{BOOT}}$ input enabled during reset
FT7	I/O	I/O	−	R	R	o	Function output triggered by enable signal
FT8	I/O	I/O	−	R	R	o	-
FT9	I/O	I/O	−	R	R	o	-
FT10	I/O	I/O	−	R	R	o	Function output triggered by enable signal

int: Interrupt input
−: Not exit
o: Exit
R: Force disable during reset.
NoR: Unaffected by reset.
EnR: Force enable during reset.

9.3.2 Type FT1

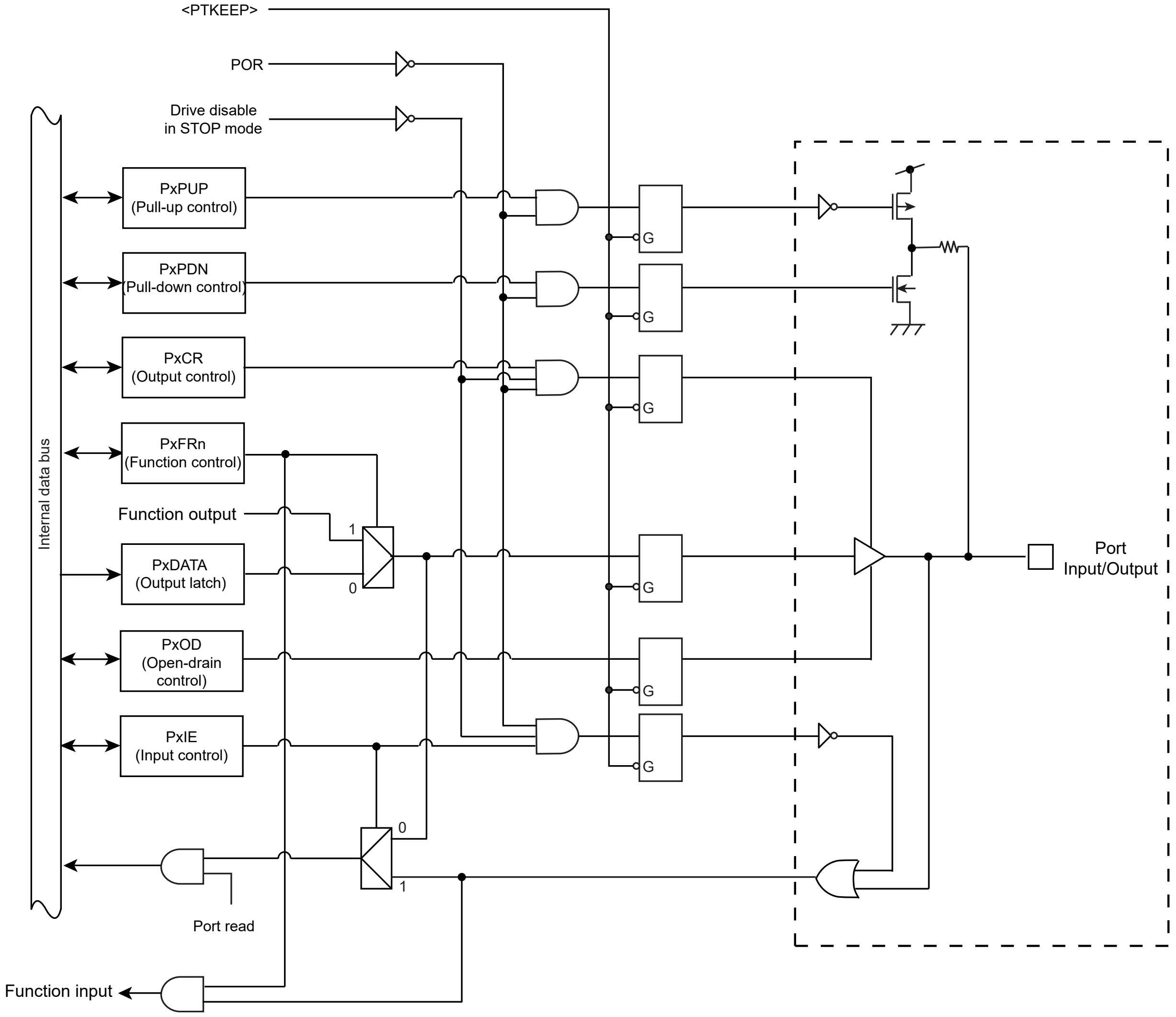

<PTKEEP>
POR
Drive disable
in STOP mode
Internal data bus
PxPUP
(Pull-up control)
PxPDN
(Pull-down control)
PxCR
(Output control)
PxFRn
(Function control)
Function output
PxDATA
(Output latch)
PxOD
(Open-drain
control)
PxIE
(Input control)
Port read
Function input
G
1
0
Port
Input/Output

Figure 9-1 Port Type FT1

9.3.3 Type FT2

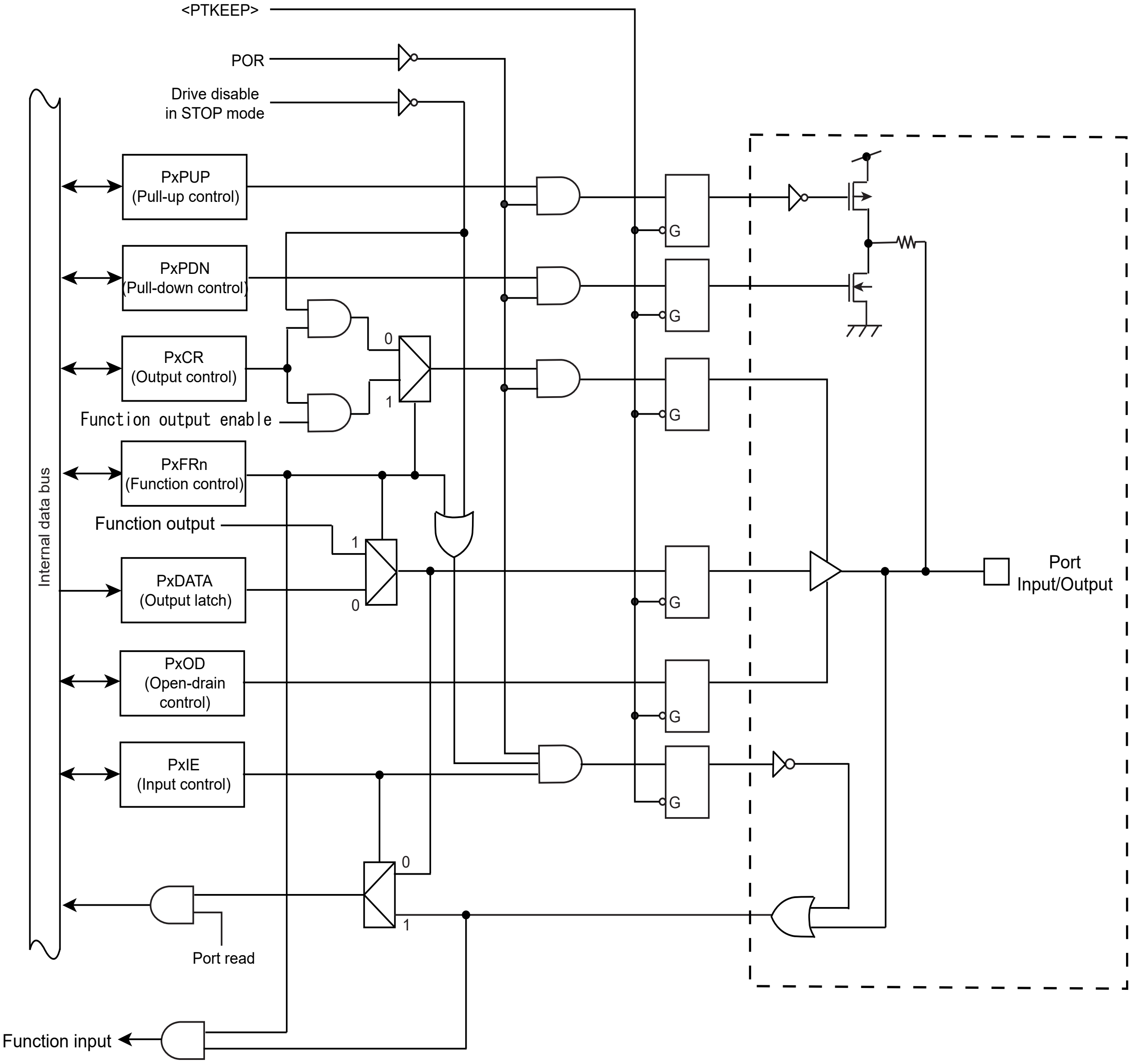

<PTKEEP>
POR
Drive disable in STOP mode
PxPUP (Pull-up control)
PxPDN (Pull-down control)
PxCR (Output control)
Function output enable
PxFRn (Function control)
Function output
PxDATA (Output latch)
PxOD (Open-drain control)
PxIE (Input control)
Port read
Function input
Internal data bus
G
0
1
Port Input/Output

Figure 9-2 Port Type FT2

Note: $\overline{\text{TRST}}$ has noise filter(30ns Typ.).

9.3.4 Type FT3

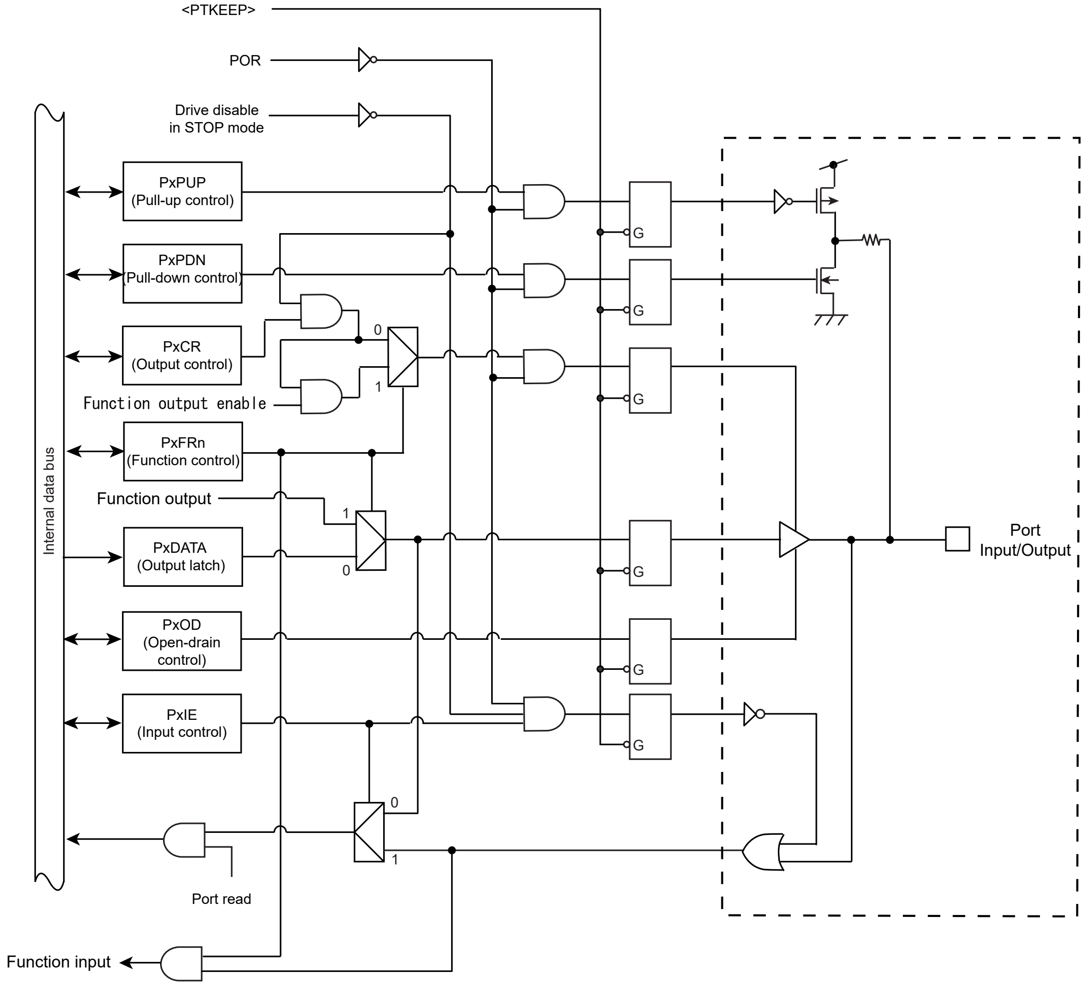

<PTKEEP>
POR
Drive disable
in STOP mode
PxPUP
(Pull-up control)
PxPDN
(Pull-down control)
PxCR
(Output control)
Function output enable
PxFRn
(Function control)
Function output
PxDATA
(Output latch)
PxOD
(Open-drain
control)
PxIE
(Input control)
Internal data bus
Port read
Function input
G
0
1
Port
Input/Output

Figure 9-3 Port Type FT3

9.3.5 Type FT4

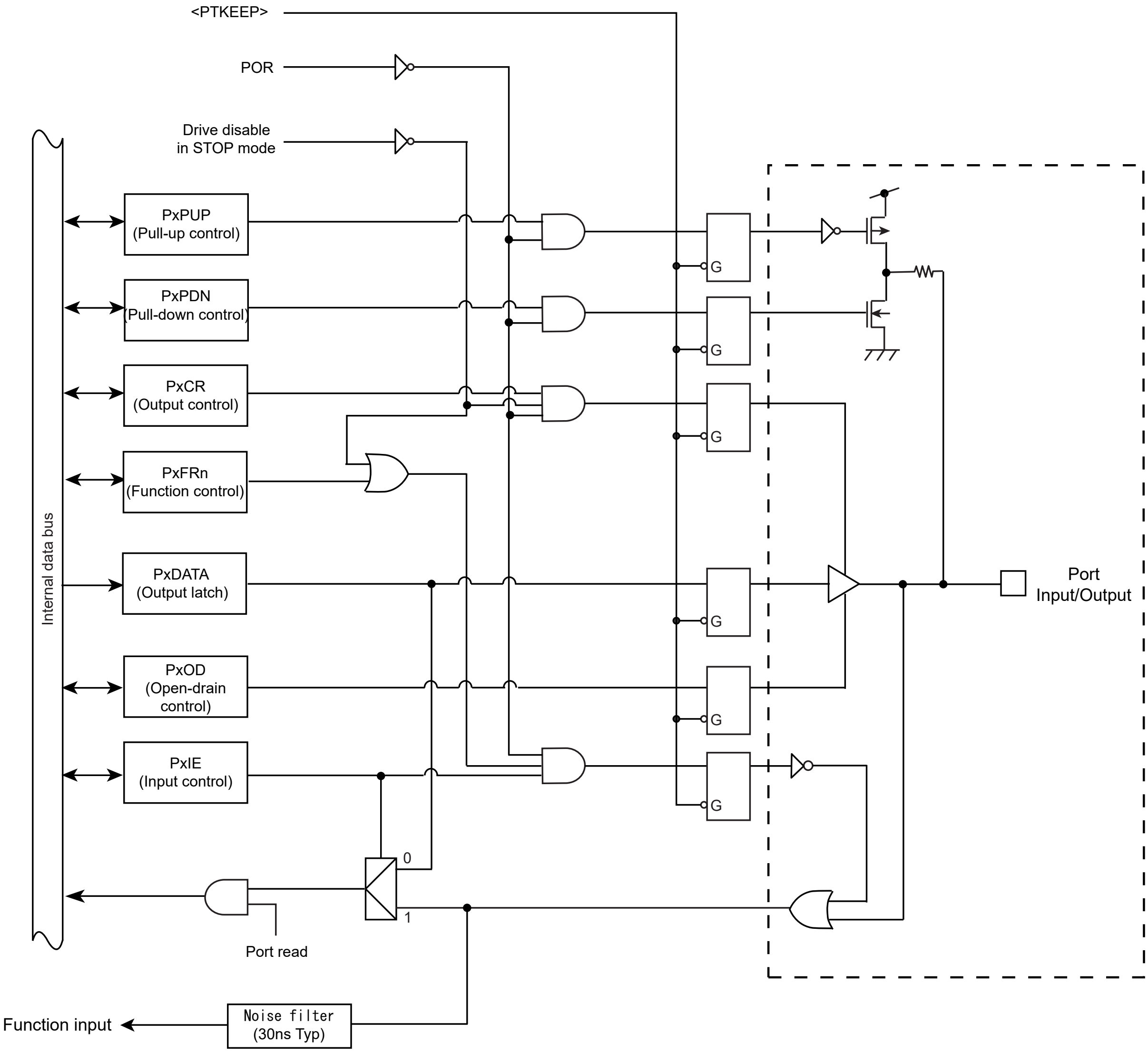

<PTKEEP>
POR
Drive disable in STOP mode
Internal data bus
PxPUP (Pull-up control)
PxPDN (Pull-down control)
PxCR (Output control)
PxFRn (Function control)
PxDATA (Output latch)
PxOD (Open-drain control)
PxIE (Input control)
G
G
G
G
G
G
0
1
Port read
Port Input/Output
Function input
Noise filter (30ns Typ)

Figure 9-4 Port Type FT4

9.3.6 Type FT5

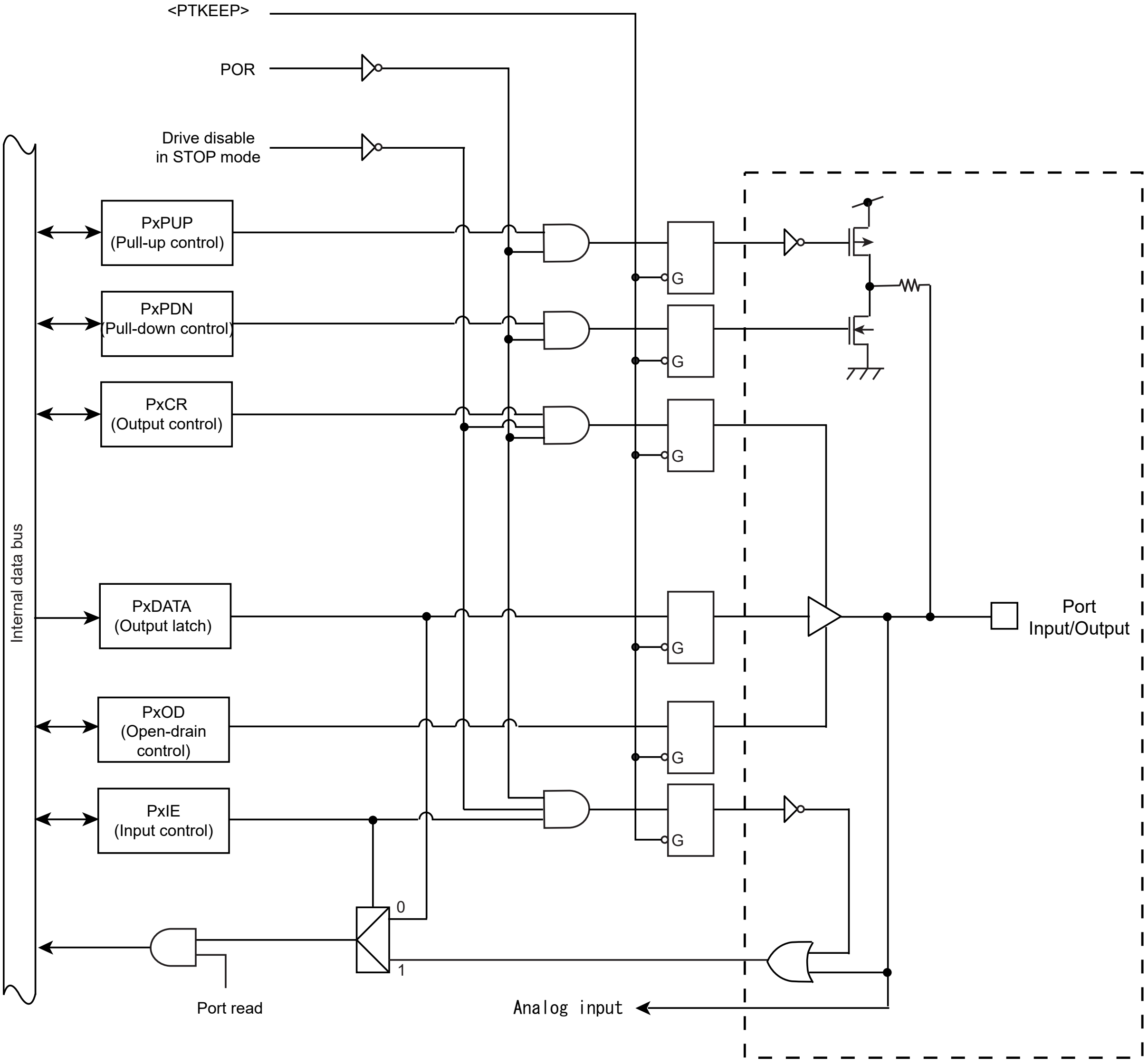

<PTKEEP>
POR
Drive disable
in STOP mode
PxPUP
(Pull-up control)
PxPDN
(Pull-down control)
PxCR
(Output control)
Internal data bus
PxDATA
(Output latch)
PxOD
(Open-drain
control)
PxIE
(Input control)
G
G
G
G
G
G
0
1
Port read
Analog input
Port
Input/Output

Figure 9-5 Port Type FT5

9.3.7 Type FT6

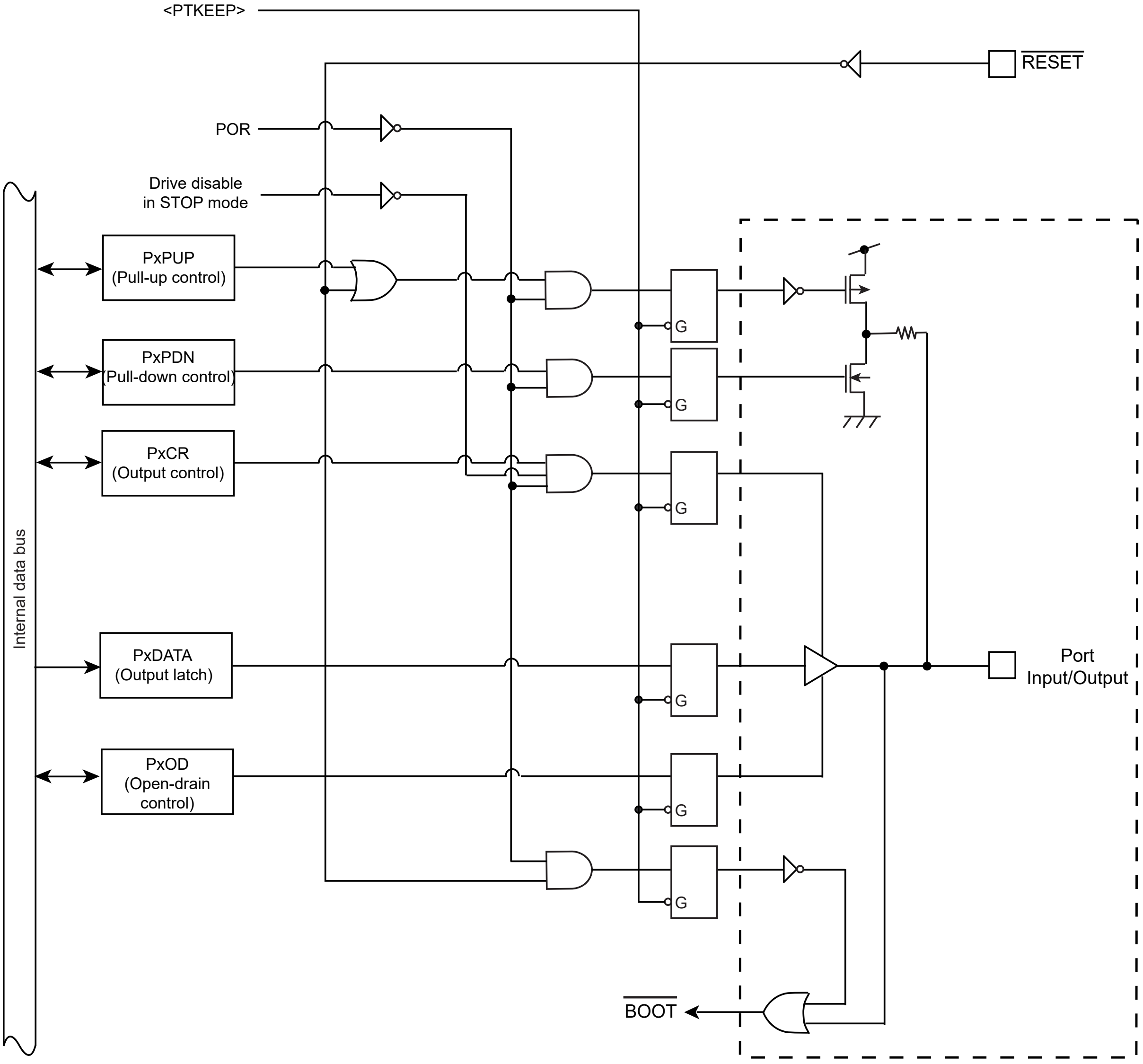

<PTKEEP>
RESET
POR
Drive disable in STOP mode
PxPUP (Pull-up control)
PxPDN (Pull-down control)
PxCR (Output control)
Internal data bus
PxDATA (Output latch)
PxOD (Open-drain control)
G
Port Input/Output
BOOT

Figure 9-6 Port Type FT6

9.3.8 Type FT7

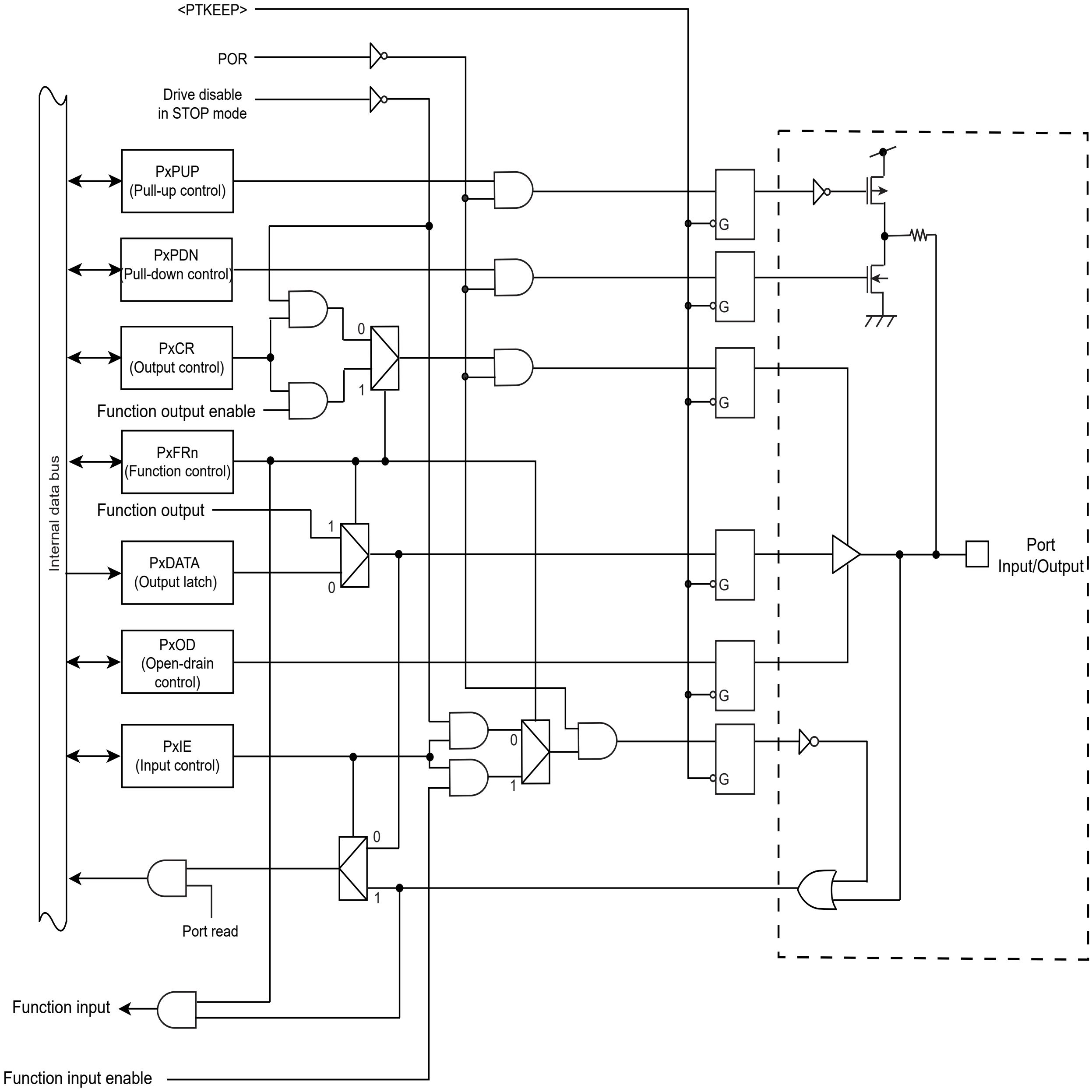

<PTKEEP>
POR
Drive disable
in STOP mode
Internal data bus
PxPUP
(Pull-up control)
PxPDN
(Pull-down control)
PxCR
(Output control)
Function output enable
PxFRn
(Function control)
Function output
PxDATA
(Output latch)
PxOD
(Open-drain
control)
PxIE
(Input control)
Port read
Function input
Function input enable
G
0
1
Port
Input/Output

Figure 9-7 Port Type FT7

9.3.9 Type FT8

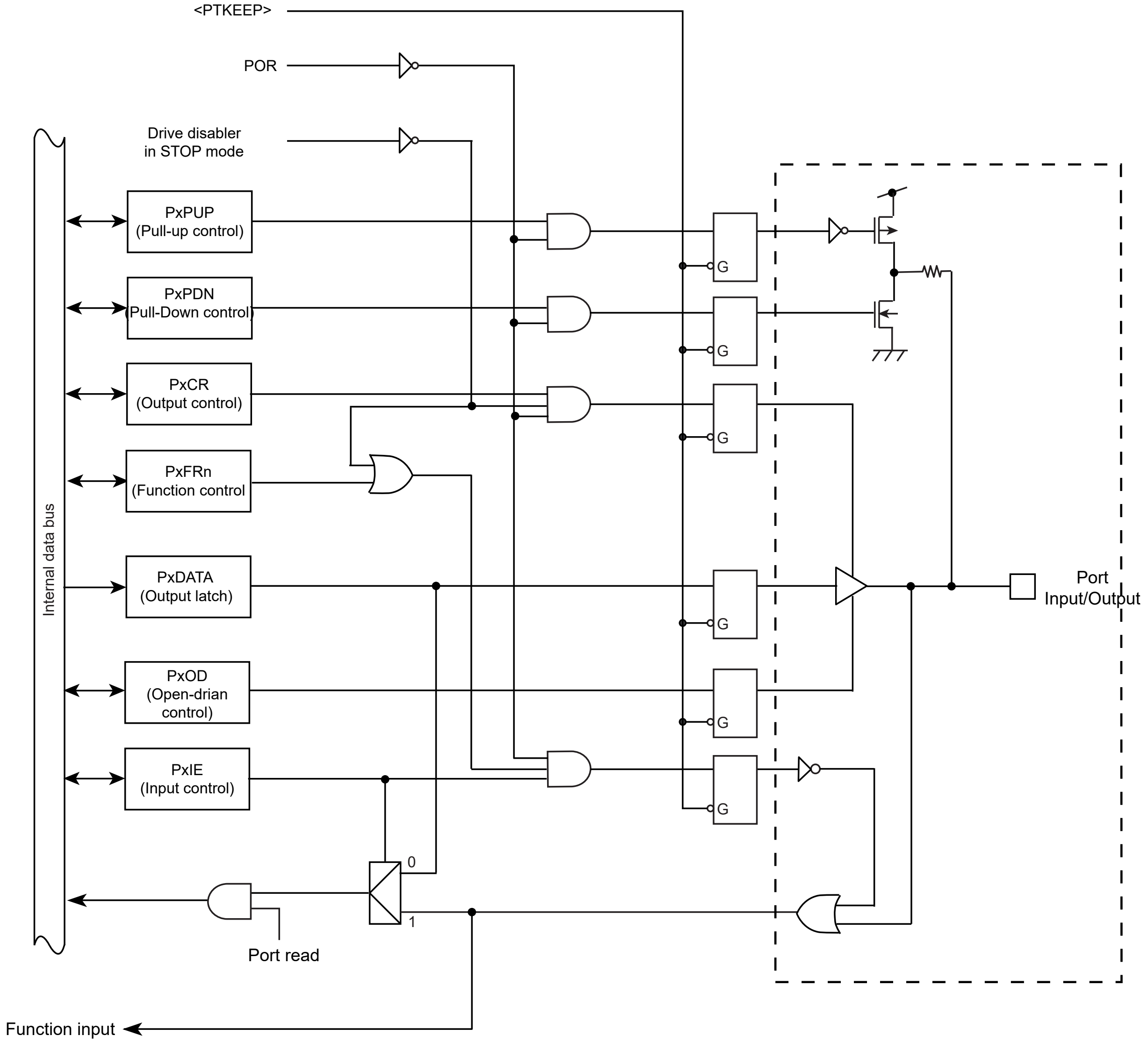

<PTKEEP>
POR
Drive disabler
in STOP mode
Internal data bus
PxPUP
(Pull-up control)
PxPDN
(Pull-Down control)
PxCR
(Output control)
PxFRn
(Function control
PxDATA
(Output latch)
PxOD
(Open-drian
control)
PxIE
(Input control)
G
0
1
Port read
Port
Input/Output
Function input

Figure 9-8 Port Type FT8

9.3.10 Type FT9

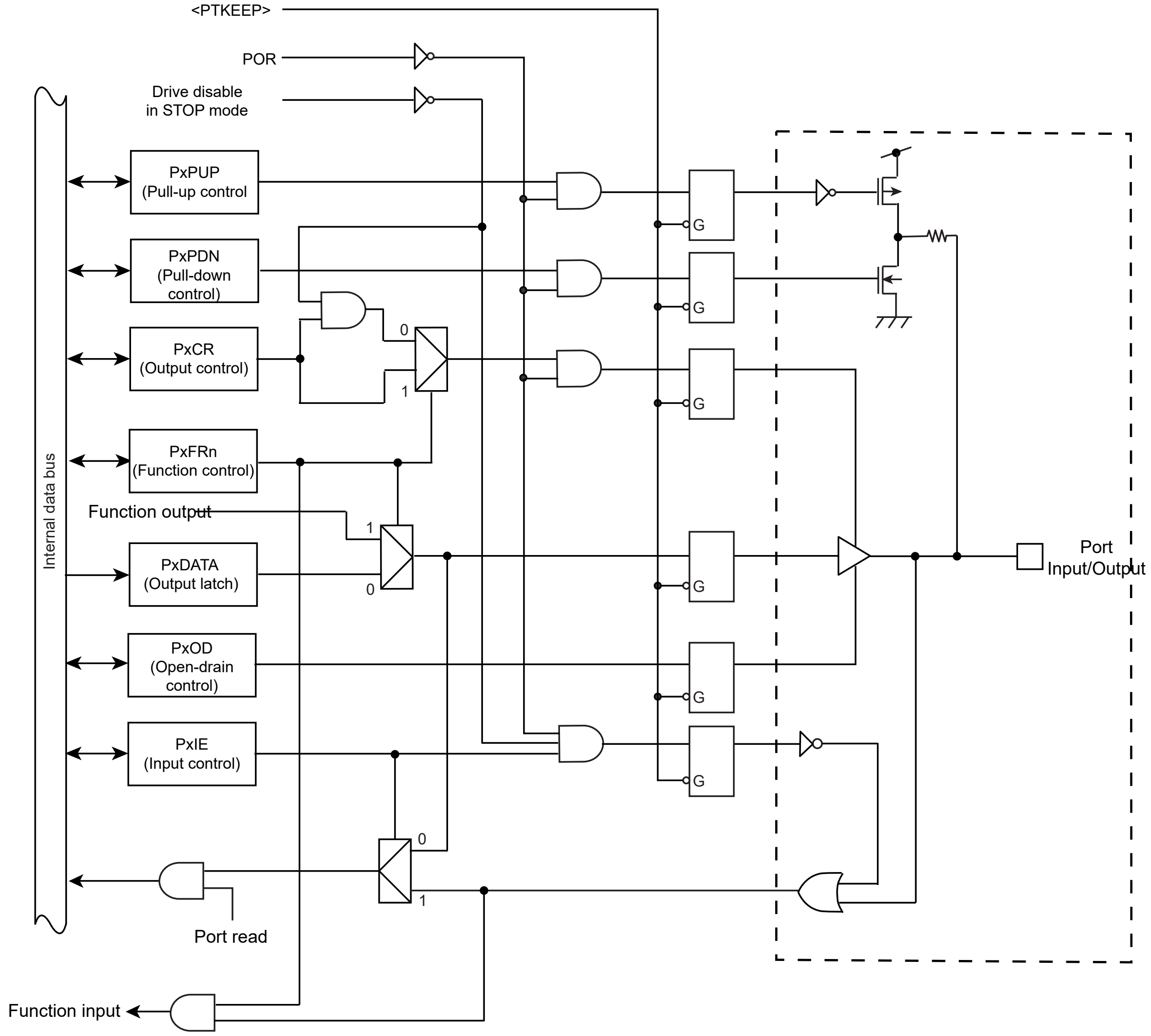

<PTKEEP>
POR
Drive disable
in STOP mode
Internal data bus
PxPUP
(Pull-up control
PxPDN
(Pull-down
control)
PxCR
(Output control)
PxFRn
(Function control)
Function output
PxDATA
(Output latch)
PxOD
(Open-drain
control)
PxIE
(Input control)
Port read
Function input
0
1
G
Port
Input/Output

Figure 9-9 Port Type FT9

9.3.11 Type FT10

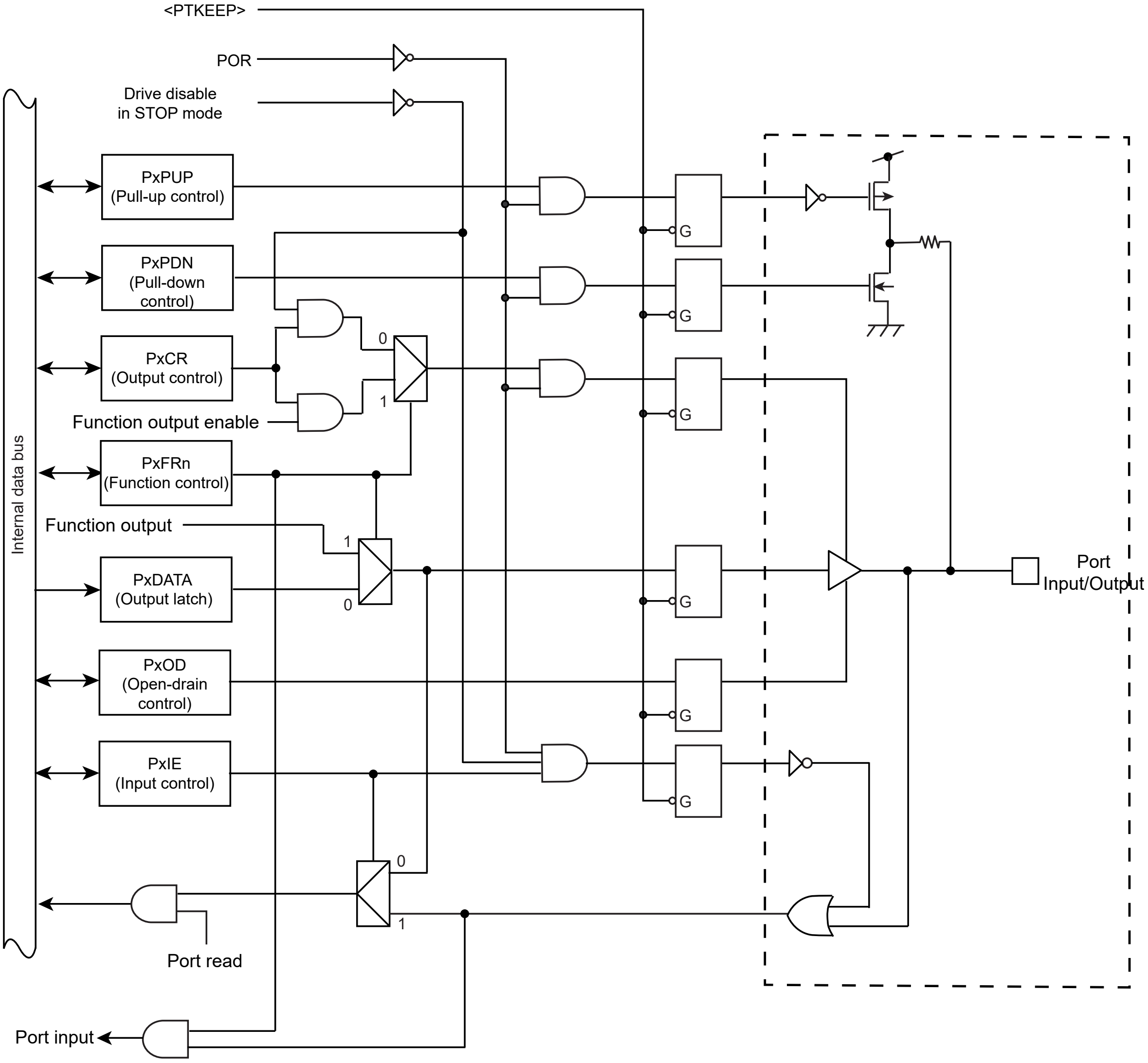

<PTKEEP>
POR
Drive disable
in STOP mode
PxPUP
(Pull-up control)
PxPDN
(Pull-down
control)
PxCR
(Output control)
Function output enable
PxFRn
(Function control)
Function output
PxDATA
(Output latch)
PxOD
(Open-drain
control)
PxIE
(Input control)
Internal data bus
Port read
Port input
G
0
1
Port
Input/Output

Figure 9-10 Port Type FT10

9.4 Apendex (List of Port Setting)

The following tables show port register settings in each pin.

The register is set or cleared for a perpheral function.

- Setting of using PE4 as SCLKx (Output)
    - "1" appeared below PECR means setting <PE4C> to "1".
    - "PE4F1" appeared below PEFRn means setting <PE4F1> to "1".
    - "x" appeared below PEOD, PEPUP and PEPDN means setting voluntary.
    - "0" appeared below PEIE means setting <PE4IE> to "0".

Pin name	Port type	Function	After reset	Px CR	Px FRn	Px OD	Px PUP	Px PDN	Px IE
PE4	FT2	SCLKx (Output)		"1"	PE4 F1	x	x	x	0

9.4.1 The Setting of I/O DEdicated Port

When an I/O dedicated port is used, set its registers as follows:

Pin name	Port type	Function	After reset	Px CR	Px FRn	Px OD	Px PUP	Px PDN
Pxn	-	Input port		0	x	x	x	1
		Output port		1	x	x	x	0

9.4.2 The Setting of Input Dedicated Port

When an input dedicated port is used, set its registers as follows:

Pin name	Port type	Function	After reset	Px CR	Px FRn	Px OD	Px PUP	Px PDN
Pxn	-	Input port		0	x	x	x	1

9.4.3 The Setting of Output Dedicated Port

When an output dedicated port is used, set its registers as follows:

Pin name	Port type	Function	After reset	Px CR	Px FRn	Px OD	Px PUP	Px PDN
Pxn	-	Output port (Hi-Z output)		0	0	x	x	x
		Output port		1	0	x	x	x

9.4.4 Setting of peripheral’s I/O port

This section describes the settings in case that ports are used for peripheral functions.

Some pins are specified as a certain function after reset. In this case, the symbol "o" is described in the "After reset" Column of the following tables.

9.4.4.1 Port A Setting

Table 9-4 Port Setting List (Port A)

Pin name	Port Type	Function	After reset	PA CR	PA FRx	PA OD	PA PUP	PA PDN	PA IE
PA0	FT2	TDO (Output)/ SWV (Output)	o	1	PA0 F1	0	0	0	0
	FT1	DTR5 (Output)		1	PA0 F2	x	x	x	0
PA1	FT2	TMS (Input)/ SWDIO (I/O)	o	1	PA1 F1	0	1	0	1
	FT1	DSR5 (Input)		0	PA1 F2	x	x	x	1
PA2	FT2	TCK (Input)/ SWCLK (Input)	o	0	PA2 F1	0	0	1	1
	FT1	RIN5 (Input)		0	PA2 F2	x	x	x	1
PA3	FT2	TDI (Input)	o	0	PA3 F1	0	1	0	1
	FT1	DCD5 (Input)		0	PA3 F2	x	x	x	1
	FT4	INT3 (Input)		0	PA3 F3	x	x	x	1
PA4	FT2	$\overline{\text{TRST}}$(Input)	o	0	PA4 F1	0	1	0	1
	FT1	$\overline{\text{RTS5}}$ (Output)		1	PA4 F2	x	x	x	0
PA5	FT9	TRACECLK (Output)		1	PA5 F1	0	0	0	0
	FT1	RXD5 (Input)		0	PA5 F2	x	x	x	1
	FT1	IRIN5 (Input)		0	PA5 F3	x	x	x	1
PA6	FT9	TRACEDATA0 (Output)		1	PA6 F1	0	0	0	0
	FT1	TXD5 (Output)		1	PA6 F2	x	x	x	0
	FT1	IROUT5 (Output)		1	PA6 F3	x	x	x	0
PA7	FT9	TRACEDATA1 (Output)		1	PA7 F1	0	0	0	0
	FT1	$\overline{\text{CTS5}}$ (Input)		0	PA7 F2	x	x	x	1
	FT1	SCLK3 (Input)		0	PA7 F3	x	x	x	1
	FT1	SCLK3 (Output)		1	PA7 F3	x	x	x	0
	FT1	$\overline{\text{CTS3}}$ (Input)		0	PA7 F4	x	x	x	1
	FT1	TB7OUT (Output)		1	PA7 F5	x	x	x	0

9.4.4.2 Port B Setting

Table 9-5 Port Setting List (Port B)

Pin name	Port Type	Function	After reset	PB CR	PB FRx	PB OD	PB PUP	PB PDN	PB IE
PB0	FT9	TRACEDATA2 (Output)		1	PB0 F1	0	0	0	0
	FT1	TXD3 (Output)		1	PB0 F3	x	x	x	0
PB1	FT9	TRACEDATA3 (Output)		1	PB1 F1	0	0	0	0
	FT1	RXD3 (Input)		0	PB1 F3	x	x	x	1
PB2	FT9	$\overline{WR}$ (Output)		1	PB2 F1	x	x	x	0
	FT3	SP2CLK (Input)		0	PB2 F2	x	x	x	1
		SP2CLK (Output)		1	PB2 F2	x	x	x	0
	FT2	MTOUT03 (Output)		1	PB2 F3	x	x	x	0
	FT1	MTTB3OUT (Output)		1	PB2 F4	x	x	x	0
PB3	FT9	$\overline{RD}$ (Output)		1	PB3 F1	x	x	x	0
	FT3	SP2DO (Output)		1	PB3 F2	x	x	x	0
	FT2	MTOUT13 (Output)		1	PB3 F3	x	x	x	0
	FT1	MTTB3IN (Input)		0	PB3 F4	x	x	x	1
PB4	FT9	$\overline{CS0}$ (Output)		1	PB4 F1	x	x	x	0
	FT3	SP2DI (Input)		0	PB4 F2	x	x	x	1
	FT1	$\overline{GEMG3}$ (Input)		0	PB4 F3	x	x	x	1
	FT4	INT7 (Input)		0	PB4 F4	x	x	x	1
PB5	FT9	ALE (Output)		1	PB5 F1	x	x	x	0
	FT3	SP2FSS (Input)		0	PB5 F2	x	x	x	1
		SP2FSS (Output)		1	PB5 F2	x	x	x	0
	FT1	MT3IN (Input)		0	PB5 F3	x	x	x	1
	FT4	INT1 (Input)		0	PB5 F4	x	x	x	1

Table 9-5 Port Setting List (Port B)

Pin name	Port Type	Function	After reset	PB CR	PB FRx	PB OD	PB PUP	PB PDN	PB IE
PB6	FT9	$\overline{\text{BELL}}$ (Output)		1	PB6 F1	x	x	x	0
	FT1	SCOUT (Output)		1	PB6 F2	x	x	x	0
	FT1	TB3OUT (Output)		1	PB6 F4	x	x	x	0

Note: The input and pulled-up of PB6 are enabled when $\overline{\text{RESET}}$ pin is low. PB6 is used as $\overline{\text{BOOT}}$ pin.

9.4.4.3 Port E Setting

Table 9-6 Port Setting List (Port E)

Pin name	Port Type	Function	After reset	PE CR	PE FRx	PE OD	PE PUP	PE PDN	PE IE
PE0	FT9	A16 (Output)		1	PE0 F3	x	x	x	0
	FT4	INT4 (Input)		0	PE0 F4	x	x	x	1
	FT1	TB0IN (Input)		0	PE0 F5	x	x	x	1
PE1	FT1	RXD0 (Input)		0	PE1 F1	x	x	x	1
	FT9	A17 (Output)		1	PE1 F3	x	x	x	0
	FT4	INT5 (Input)		0	PE1 F4	x	x	x	1
	FT1	TB1IN (Input)		0	PE1 F5	x	x	x	1
PE2	FT1	TXD0 (Output)		1	PE2 F1	x	x	x	0
	FT9	A18 (Output)		1	PE2 F3	x	x	x	0
	FT1	TB1OUT (Output)		1	PE2 F5	x	x	x	0
PE3	FT1	SCLK0 (Input)		0	PE3 F1	x	x	x	1
	FT1	SCLK0 (Output)		1	PE3 F1	x	x	x	0
	FT9	A19 (Output)		1	PE3 F3	x	x	x	0
	FT1	$\overline{\text{CTS0}}$ (Input)		0	PE3 F4	x	x	x	1
	FT1	TB0OUT (Output)		1	PE3 F5	x	x	x	0
PE4	FT1	SCLK1 (Input)		0	PE4 F1	x	x	x	1
	FT1	SCLK1 (Output)		1	PE4 F1	x	x	x	0
	FT9	A20 (Output)		1	PE4 F3	x	x	x	0
	FT1	$\overline{\text{CTS1}}$ (Input)		0	PE4 F4	x	x	x	1
	FT1	TB2OUT (Output)		1	PE4 F5	x	x	x	0
PE5	FT1	TXD1 (Output)		1	PE5 F1	x	x	x	0
	FT9	A21 (Output)		1	PE5 F3	x	x	x	0
PE6	FT1	RXD1 (Input)		0	PE6 F1	x	x	x	1
	FT9	A22 (Output)		1	PE6 F3	x	x	x	0

Table 9-6 Port Setting List (Port E)

Pin name	Port Type	Function	After reset	PE CR	PE FRx	PE OD	PE PUP	PE PDN	PE IE
PE7	FT9	A23 (Output)		1	PE7 F3	x	x	x	0
	FT4	INT6 (Input)		0	PE7 F4	x	x	x	1
	FT1	TB2IN (Input)		0	PE7 F5	x	x	x	1

9.4.4.4 Port F Setting

Table 9-7 Port Setting List (Port F)

Pin name	Port Type	Function	After reset	PF CR	PF FRx	PF OD	PF PUP	PF PDN	PF IE
PF0	FT7	AD0 (I/O)		1	PF0 F1	x	x	x	1
	FT1	$\overline{\text{CTS4}}$ (Input)		0	PF0 F3	x	x	x	1
PF1	FT7	AD1 (I/O)		1	PF1 F1	x	x	x	1
	FT1	TXD4 (Output)		1	PF1 F3	x	x	x	0
		IROUT4 (Output)		1	PF1 F4	x	x	x	0
PF2	FT7	AD2 (I/O)		1	PF2 F1	x	x	x	1
	FT1	RXD4 (Input)		0	PF2 F3	x	x	x	1
		IRIN4 (Input)		0	PF2 F4	x	x	x	1
PF3	FT7	AD3 (I/O)		1	PF3 F1	x	x	x	1
	FT1	$\overline{\text{RTS4}}$ (Output)		1	PF3 F3	x	x	x	0
PF4	FT7	AD4 (I/O)		1	PF4 F1	x	x	x	1
	FT4	INT0 (Input)		0	PF4 F2	x	x	x	1
	FT1	DCD4 (Input)		0	PF4 F3	x	x	x	1
PF5	FT7	AD5 (I/O)		1	PF5 F1	x	x	x	1
	FT1	ENCZ (Input)		0	PF5 F2	x	x	x	1
		RIN4 (Input)		0	PF5 F3	x	x	x	1
		SCK1 (Input)		0	PF5 F4	x	x	x	1
		SCK1 (Output)		1	PF5 F4	x	x	x	0
PF6	FT7	AD6 (I/O)		1	PF6 F1	x	x	x	1
	FT1	ENCB (Input)		0	PF6 F2	x	x	x	1
		DSR4 (Input)		0	PF6 F3	x	x	x	1
		SI1 (Input)		0	PF6 F4	x	x	x	1
		SCL1 (I/O)		1	PF6 F4	1	x	x	1

Table 9-7 Port Setting List (Port F)

Pin name	Port Type	Function	After reset	PF CR	PF FRx	PF OD	PF PUP	PF PDN	PF IE
PF7	FT7	AD7 (I/O)		1	PF7 F11	x	x	x	1
	FT1	ENCA (Input)		0	PF7 F2	x	x	x	1
		DTR4 (Output)		1	PF7 F3	x	x	x	0
		SO1 (Input)		0	PF7 F4	x	x	x	1
		SDA1 (I/O)		1	PF7 F4	1	x	x	1

9.4.4.5 Port G Setting

Table 9-8 Port Setting List (Port G)

Pin name	Port Type	Function	After reset	PG CR	PG FRx	PG OD	PG PUP	PG PDN	PG IE
PG0	FT7	AD8 (I/O)		1	PG0 F1	x	x	x	1
	FT1	MT0IN (Input)		0	PG0 F3	x	x	x	1
PG1	FT7	AD9 (I/O)		1	PG1 F1	x	x	x	1
	FT1	$\overline{\text{EMG}}$ (Input)		0	PG1 F2	x	x	x	1
		$\overline{\text{GEMG0}}$ (Input)		0	PG1 F3	x	x	x	1
PG2	FT7	AD10 (I/O)		1	PG2 F1	x	x	x	1
	FT2	ZO (Output)		1	PG2 F2	x	x	x	0
	FT2	MTOUT10 (Output)		1	PG2 F3	x	x	x	0
	FT1	MTTB0IN (Input)		0	PG2 F4	x	x	x	1
PG3	FT7	AD11 (I/O)		1	PG3 F1	x	x	x	1
	FT2	WO (Output)		1	PG3 F2	x	x	x	0
	FT2	MTOUT00 (Output)		1	PG3 F3	x	x	x	0
	FT1	MTTBOUT (Output)		1	PG3 F4	x	x	x	0
PG4	FT7	AD12 (I/O)		1	PG4 F1	x	x	x	1
	FT2	YO (Output)		1	PG4 F2	x	x	x	0
	FT3	SP1CLK (Input)		0	PG4 F3	x	x	x	1
	FT1	SP1CLK (Output)		1	PG4 F3	x	x	x	0
PG5	FT7	AD13 (I/O)		1	PG5 F1	x	x	x	1
	FT2	VO (Output)		1	PG5 F2	x	x	x	0
	FT3	SP1DO (Output)		1	PG5 F3	x	x	x	0
PG6	FT7	AD14 (I/O)		1	PG6 F1	x	x	x	1
	FT2	XO (Output)		1	PG6 F2	x	x	x	0
	FT3	SP1DI (Input)		0	PG6 F3	x	x	x	1

Table 9-8 Port Setting List (Port G)

Pin name	Port Type	Function	After reset	PG CR	PG FRx	PG OD	PG PUP	PG PDN	PG IE
PG7	FT7	AD15 (I/O)		1	PG7 F1	x	x	x	1
	FT2	UO (Output)		1	PG7 F2	x	x	x	0
	FT3	SP1FSS (Input)		0	PG7 F3	x	x	x	1
		SP1FSS (Output)		1	PG7 F3	x	x	x	0

9.4.4.6 Port H Setting

Table 9-9 Port Setting List (Port H)

Pin name	Port Type	Function	After reset	PH CR	PH FRx	PH OD	PH PUP	PH PDN	PH IE
PH0	FT9	$\overline{\text{BELH}}$ (Output)		1	PH0 F1	x	x	x	0
	FT1	TB5OUT (Output)		1	PH0 F2	x	x	x	0
		MT2IN (Input)		0	PH0 F3	x	x	x	1
		SO2 (Output)		1	PH0 F5	x	x	x	0
		SDA2 (I/O)		1	PH0 F5	1	x	x	1
PH1	FT9	$\overline{\text{CS1}}$ (Output)		1	PH1 F1	x	x	x	0
	FT1	TB4OUT (Output)		1	PH1 F2	x	x	x	0
		$\overline{\text{GEMG2}}$ (Input)		0	PH1 F3	x	x	x	1
		SI2 (Input)		0	PH1 F5	x	x	x	1
		SCL2 (I/O)		1	PH1 F5	1	x	x	1
PH2	FT9	$\overline{\text{CS2}}$ (Output)		1	PH2 F1	x	x	x	0
	FT2	MTOUT12 (Output)		1	PH2 F3	x	x	x	0
	FT1	MTTB2IN (Input)		0	PH2 F4	x	x	x	1
	FT1	SCK2 (Input)		0	PH2 F5	x	x	x	1
	FT1	SCK2 (Output)		1	PH2 F5	x	x	x	0
PH3	FT9	$\overline{\text{CS3}}$ (Output)		1	PH3 F1	x	x	x	0
	FT2	MTOUT02 (Output)		1	PH3 F3	x	x	x	0
	FT1	MTTB2OUT (Output)		1	PH3 F4	x	x	x	0

9.4.4.7 Port I Setting

Table 9-10 Port Setting List (Port I)

Pin name	Port Type	Function	After reset	PI CR	PI FRx	PI OD	PI PUP	PI PDN	PI IE
PI0	FT4	INT9 (Input)		0	PI0 F1	x	x	x	1
	FT5	AINA0	o	0	0	0	0	0	0
PI1	FT4	INTA (Input)		0	PI1 F1	x	x	x	1
	FT5	AINA1	o	0	0	0	0	0	0
PI2	FT4	INTB (Input)		0	PI2 F1	x	x	x	1
	FT5	AINA2	o	0	0	0	0	0	0
PI3	FT4	INTC (Input)		0	PI3 F1	x	x	x	1
	FT1	$\overline{\text{DMAREQ}}$ (Input)		0	PI3 F2	x	x	x	1
	FT5	AINA3	o	0	0	0	0	0	0
PI4	FT5	AINB0	o	0	0	0	0	0	0
PI5	FT5	AINB1	o	0	0	0	0	0	0
PI6	FT5	AINB2	o	0	0	0	0	0	0
PI7	FT5	AINB3	o	0	0	0	0	0	0

9.4.4.8 Port K Setting

Table 9-11 Port Setting List (Port K)

Pin name	Port Type	Function	After reset	PK CR	PK FRx	PK OD	PK PUP	PK PKN	PKn name
PK0	FT4	INTD (Input)		0	PK0F1	x	x	x	1
		USBDPON (Input)		0	PK0F1	x	x	x	1
PK1	FT3	SP0FSS (Input)		0	PK1F2	x	x	x	1
		SP0FSS (Output)		1	PK1F2	x	x	x	0
	FT4	INT8 (Input)		0	PK1F3	x	x	x	1
	FT1	TB6OUT (Output)		1	PK1F4	x	x	x	0
PK2	FT1	USB_ECLK (Input)		0	PK2F1	x	x	x	1
	FT3	SP0DI (Input)		0	PK2F2	x	x	x	1
	FT1	SO0 (Input)		0	PK2F3	x	x	x	1
		SDA0 (I/O)		1	PK2F4	1	x	x	1
PK3	FT3	SP0DO (Output)		1	PK3F2	x	x	x	0
	FT1	SI0 (Input)		0	PK3F3	x	x	x	1
		SCL0 (I/O)		1	PK3F3	1	x	x	1
PK4	FT1	RXIN (Input)		0	PK4F1	x	x	x	1
	FT3	SP0CLK (Input)		0	PK4F2	x	x	x	1
		SP0CLK (Output)		1	PK4F2	x	x	x	0
	FT1	SCK0 (Input)		0	PK4F3	x	x	x	1
		SCK0 (Output)		1	PK4F3	x	x	x	0

Note: PK0 is used as USBPON input or output port when USB device controller is used. PK0 is used as I/O port or INTD input when USB device controller is not used.

9.4.4.9 Port L Setting

Table 9-12 Port Setting List (Port L)

Pin name	Port Type	Function	After reset	PL CR	PL FRx	PL OD	PL PUP	PL PDN	PL IE
PL0	FT4	INT2 (Input)		0	PL0 F2	x	x	x	1
	FT1	MT1IN (Input)		0	PL0 F3	x	x	x	1
		$\overline{\text{ADTRGA}}$ (Input)		0	PL0 F4	x	x	x	1
PL1	FT1	$\overline{\text{GEMG1}}$ (Input)		0	PL1 F3	x	x	x	1
		$\overline{\text{DATRG}}$ (Input)		0	PL1 F4	x	x	x	1
		RXD2 (Input)		0	PL1 F5	x	x	x	1
PL2	FT2	MTOUT11 (Output)		1	PL2 F3	x	x	x	0
	FT1	MTTB1IN (Input)		0	PL2 F4	x	x	x	1
	FT1	TXD2 (Output)		1	PL2 F5	x	x	x	0
PL3	FT2	MTOUT01 (Output)		1	PL3 F3	x	x	x	0
	FT1	MTTB1OUT (Output)		1	PL3 F4	x	x	x	0
	FT1	SCLK2 (Input)		0	PL3 F5	x	x	x	1
	FT1	SCLK2 (Output)		1	PL3 F5	x	x	x	0
	FT1	$\overline{\text{CTS2}}$ (Input)		0	PL3 F6	x	x	x	1

10. External bus interface (EBIF)

10.1 Overview

The TMPM367FDFG has a built-in external bus interface function to connect to external memory, I/Os, etc. This interface consists of an external bus interface circuit (EBIF), a chip selector (CS) and a wait controller.

The chip selector and wait controller designate mapping addresses in a 4-block address space and also control wait states and data bus widths (8- or 16-bit) in these and other external address spaces.

The external bus interface circuit (EBIF) controls the timing of external buses based on the chip selector and wait controller settings.

Their features are given in the following.

Table 10-1 Features of External bus interface

features	
Memory supports	Asynchronous memory (NOR Flash memory, SRAM, Peripheral I/O and e.t.c.) Selectable multiplex bus mode.
Data bus width	Either an 8- or 16-bit width can be set for each channel.
Chip select	4 channels ($\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$)
Address access spaces	Supports up to 64MB memory spaces CS0: 0x6000_0000 to 0x63FF_FFFF (Max. 16MB for each CS)
Internal wait function	This function can be enabled for each channel. A wait of up to 15 cycles can be automatically inserted.
ALE wait function	This function can be enabled for each channel. An ALE high pulse of up to 4 cycles can be automatically inserted.
Setup cycle insertion function	This function can be enabled for each channel. A $\overline{RD}$ or $\overline{WR}$ setup cycle can be automatically inserted. (tAC cycle expanded)
Recovery (Hold) cycle insertion function	When an external bus is selected, a dummy cycle of up to 8 clocks can be inserted and this dummy cycle can be specified for each channel. Address/Data hold cycle can be inserted for $\overline{CS}$, $\overline{RD}$ and $\overline{WR}$. (tCAR, tRAE cycles expanded)
Bus expansion function	The internal wait, the ALE wait, the Setup wait and the Recovery cycle can be expanded double or quadruple.
Control pins	Multiplex bus mode: AD[15:0], A[23:16], $\overline{RD}$, $\overline{WR}$, $\overline{BELL}$, $\overline{BELH}$, $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$, ALE

10.2 Address and Data pin setting

The TMPM367FDFG can be set to multiplexed bus mode. Setting the bit <EXBSEL> of EXBMOD register to "0" as the multiplexed bus mode. Port pins E to G which are to be connected to external devices (memory), are used as address buses, data buses and address/data buses. The below table shows these.

Table 10-2 Bus Mode, Address and Data Pins

PORT	Multiplex Mode EXBMOD<EXBSEL> = "0"
Port E(PE0 to PE7)	A16 to A23
Port F(PF0 to PF7)	AD0 to AD7
Port G(PG0 to PG7)	AD8 to AD15

Each port is put into input mode after a reset. To access an external device, set the address and data bus functions by using the port control register (PxCR) and the port function register (PxFRm), and set the input enable register (PxIE).

When the access changing from the external area to internal area, the address buses are kept the previously external area address output and the data buses will be high impedance.

10.3 Registers

10.3.1 Registers List

Address and names of EBIF control registers are shown below.

Base Address = 0x4005_C000

Register name		Address (Base+)
External Bus Mode Control Register	EXBMOD	0x0000
Reserved	-	0x0004 to 0x000C
External Bus Area and Start Address Configuration Register 0	EXBAS0	0x0010
External Bus Area and Start Address Configuration Register 1	EXBAS1	0x0014
External Bus Area and Start Address Configuration Register 2	EXBAS2	0x0018
External Bus Area and Start Address Configuration Register 3	EXBAS3	0x001C
Reserved	-	0x0020 to 0x003C
External Bus Chip Select Control Register 0	EXBCS0	0x0040
External Bus Chip Select Control Register 1	EXBCS1	0x0044
External Bus Chip Select Control Register 2	EXBCS2	0x0048
External Bus Chip Select Control Register 3	EXBCS3	0x004C
Reserved	-	0x0050 to 0x0FFC

Note 1: Access the registers by using word reads and word writes.

Note 2: Access to the "Reserved" area is prohibited.

10.3.2 EXBMOD (External Bus Mode 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	-	-	-	-	-	EXBWAIT		EXBSEL
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2-1	EXBWAIT[1:0]	R/W	Bus cycle extension 00: None 01: Double 10: Quadruple 11: Prohibited These bits are used to set the setup, wait and recovery of the bus cycle to be double or quadruple. For example, if a Read setup cycle is set as two cycles by setting <EXBWAIT>="00" (no extension), the two cycles can be quadruplicated by changing the bit setting to <EXBWAIT>="01" (double). It also can be octuplicated by setting the bits to <EXBWAIT>="10" (quadruple). The extended cycle is configured by setting Read/Write setup, chip select/Read/Write recovery, ALE/internal wait cycle and <EXBWAIT> (double or quadruple).
0	EXBSEL	R/W	Write as "0".

10.3.3 EXBAS0 to 3 (External Bus Area and Start Address Configuration Register)

	31	30	29	28	27	26	25	24
bit symbol	SA31	SA30	SA29	SA28	SA27	SA26	SA25	SA24
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16
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	EXAR							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	SA31-SA16	R/W	Chip select Start address (Note) The A[31:16] is specified as start address.
15-8	−	R	Read as 0.
7-0	EXAR[7:0]	R/W	Chip select Address space size The size of address space can be specified nine kind of setting from 64Kbyte up to 16Mbyte. 0000_0000: 16 Mbyte 0000_0001: 8 Mbyte 0000_0010: 4 Mbyte 0000_0011: 2 Mbyte 0000_0100: 1 Mbyte 0000_0101: 512 Kbyte 0000_0110: 256 Kbyte 0000_0111: 128 Kbyte 0000_1000: 64 Kbyte Others: Prohibited

Note: If same address space is specified between $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$ and $\overline{CS3}$, the chip selector will be given priority shown below

High priority $\overline{CS0}$ > $\overline{CS1}$ > $\overline{CS2}$ > $\overline{CS3}$ Low priority

Table 10-3 The address specification

The address area size of a chip select	SA																-	EXAR							
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 to 8	7	6	5	4	3	2	1	0
16Mbyte	0	1	1	0	0	0	x	x	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0
8Mbyte	0	1	1	0	0	0	x	x	x	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	1
4Mbyte	0	1	1	0	0	0	x	x	x	x	0	0	0	0	0	0	-	0	0	0	0	0	0	1	0
2Mbyte	0	1	1	0	0	0	x	x	x	x	x	0	0	0	0	0	-	0	0	0	0	0	0	1	1
1Mbyte	0	1	1	0	0	0	x	x	x	x	x	x	0	0	0	0	-	0	0	0	0	0	1	0	0
512Kbyte	0	1	1	0	0	0	x	x	x	x	x	x	x	0	0	0	-	0	0	0	0	0	1	0	1
256Kbyte	0	1	1	0	0	0	x	x	x	x	x	x	x	x	0	0	-	0	0	0	0	0	1	1	0
128Kbyte	0	1	1	0	0	0	x	x	x	x	x	x	x	x	x	0	-	0	0	0	0	0	1	1	1
64Kbyte	0	1	1	0	0	0	x	x	x	x	x	x	x	x	x	x	-	0	0	0	0	1	0	0	0

10.3.4 EXBCS0 to 3 (External Bus Chip Select Control Register)

	31	30	29	28	27	26	25	24
bit symbol	CSR		WRR			RDR		
After reset	0	1	0	0	1	0	0	1
	23	22	21	20	19	18	17	16
bit symbol	-	-	ALEW		WRS		RDS	
After reset	0	0	0	1	0	1	0	1
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	CSIW				
After reset	0	0	0	0	0	0	1	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	CSW		CSW0
After reset	0	0	0	0	0	0	1	0

Bit	Bit Symbol	Type	Function
31-30	CSR[1:0]	R/W	Chip select ($\overline{\text{CSx}}$) Recovery cycle 00: None 01: 1 cycle 10: 2 cycles 11: 4 cycles
29-27	WRR[2:0]	R/W	Write ($\overline{\text{WR}}$) Recovery cycl 000: None 010: 2 cycles 100: 4 cycles 110: 6 cycles 001: 1 cycle 011: 3 cycles, 101: 5 cycles 111: 8 cycles
26-24	RDR[2:0]	R/W	Read ($\overline{\text{RD}}$) Recovery cycle 000: None 010: 2 cycles 100: 4 cycles 110: 6 cycles 001: 1 cycle 011: 3 cycles, 101: 5 cycles 111: 8 cycles
23-22	−	R	Read as 0.
21-20	ALEW[1:0]	R/W	ALE wait cycle for multiplex bus Read ($\overline{\text{RD}}$) Recovery cycle 00: None 01: 1 cycles 10: 2 cycles 11: 4 cycles
19-18	WRS[1:0]	R/W	Write ($\overline{\text{WR}}$) Setup cycle 00: None 01: 1 cycles 10: 2 cycles 11: 4 cycles
17-16	RDS[1:0	R/W	Read ($\overline{\text{RD}}$) Setup cycle 00: None 01: 1 cycles 10: 2 cycles 11: 4 cycles
15-13	−	R	Read as 0.
12-8	CSIW[4:0]	R/W	Internal Wait (Automatically insertion) 0000: 0 wait 0100: 4 waits 1000: 8 waits 1100: 12 waits 0001: 1 wait 0101: 5 waits 1001: 9 waits 1101: 13 waits 0010: 2 waits 0110: 6 waits 1010: 10 waits 1110: 14 waits 0011: 3 waits 0111: 7 waits 1011: 11 waits 1111: 15 waits
7-4	−	R	Read as 0.
3	−	R/W	Always write to "0"
2-1	CSW[2:1]	R/W	Data bus width 00: 8-bit 01: 16-bit Others: Prohibited
0	CSW0	R/W	CS Enable 0: Disable 1: Enable

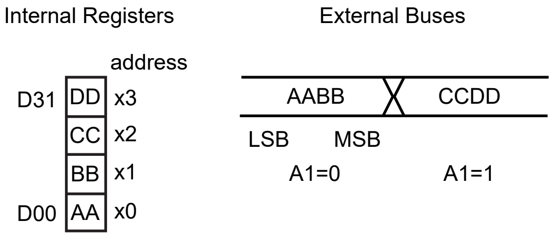
10.4 Data Format

Internal registers and external bus interfaces of the TMPM367FDFG are configured as described below.

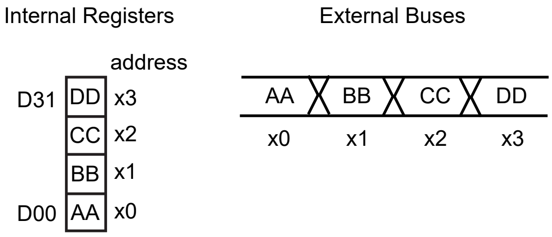
10.4.1 Little-endian mode

10.4.1.1 Word access

- 16-bit bus width

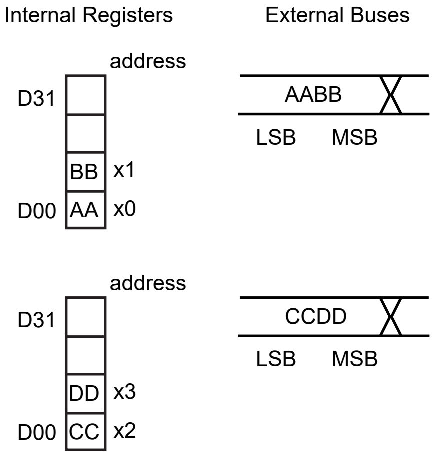

Internal Registers
External Buses
address
D31
DD
x3
CC
x2
BB
x1
D00
AA
x0
AABB
CCDD
LSB
MSB
A1=0
A1=1

- 8-bit bus width

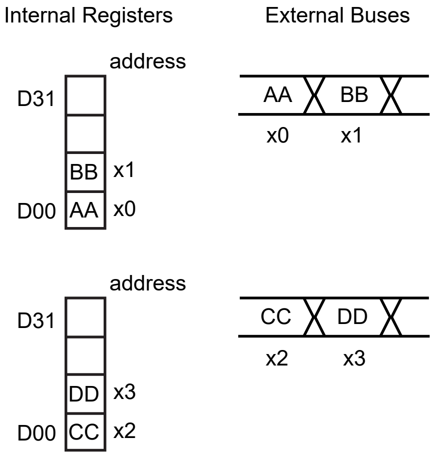

Internal Registers
External Buses
address
D31
DD
x3
CC
x2
BB
x1
D00
AA
x0
AA
BB
CC
DD
x0
x1
x2
x3

10.4.1.2 Half word access

- 16-bit bus width

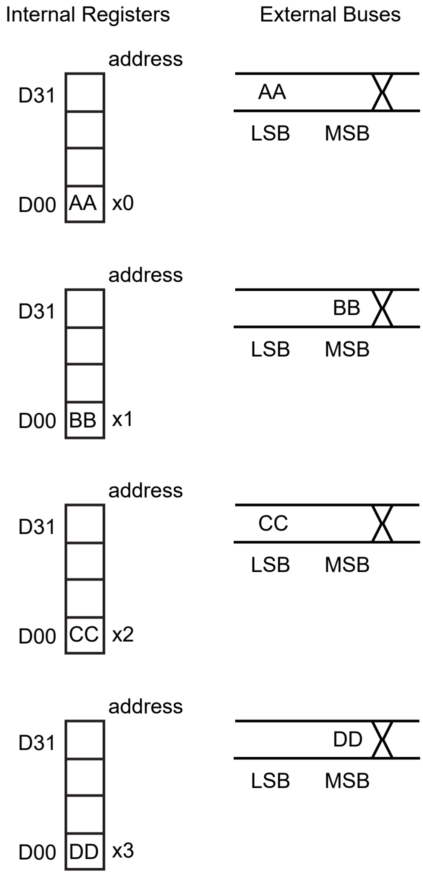

Internal Registers
External Buses
address
D31
BB
x1
D00
AA
x0
AABB
LSB
MSB
address
D31
DD
x3
D00
CC
x2
CCDD
LSB
MSB

- 8-bit bus width

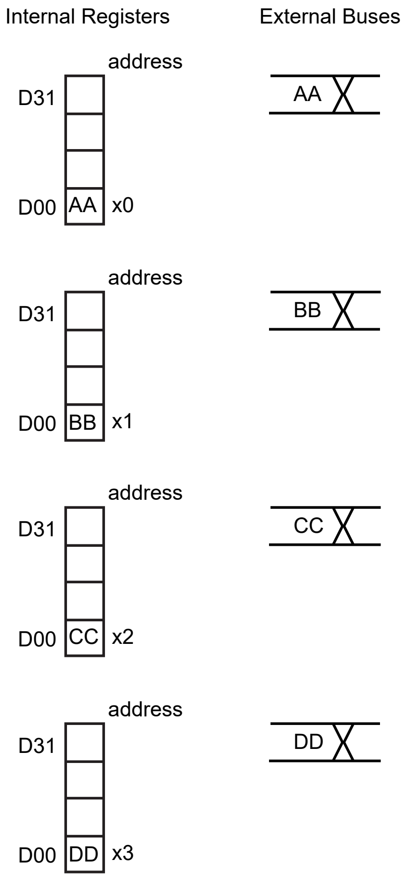

Internal Registers
External Buses
address
D31
BB
x1
D00
AA
x0
AA
BB
x0
x1
address
D31
DD
x3
D00
CC
x2
CC
DD
x2
x3

10.4.1.3 Byte access

- 16-bit bus width


Internal Registers
External Buses
address
D31
D00
AA
x0
AA
LSB
MSB
address
D31
D00
BB
x1
BB
LSB
MSB
address
D31
D00
CC
x2
CC
LSB
MSB
address
D31
D00
DD
x3
DD
LSB
MSB

- 8-bit bus width


Internal Registers
External Buses
address
D31
D00
AA
x0
AA
address
D31
D00
BB
x1
BB
address
D31
D00
CC
x2
CC
address
D31
D00
DD
x3
DD

10.5 External Bus Operations (Multiplexed Bus Mode)

This section describes various bus timing values. The timing diagram shown below assumes that the address buses are A23 through A16 and that the data buses are AD15 through AD0.

10.5.1 Basic bus operation

The external bus cycle of the TMPM367FDFG basically consists of four clock pulses. The basic clock of an external bus cycle is the same as the internal system clock.Figure 10-1 shows read bus timing and Figure 10-2 shows write bus timing. If internal areas are accessed, address buses remain unchanged and the ALE does not output latch pulse as shown in these figures.

Additionally, address/data buses are in a state of high impedance and control signals such as $\overline{\text{RD}}$ and $\overline{\text{WR}}$ do not become active.

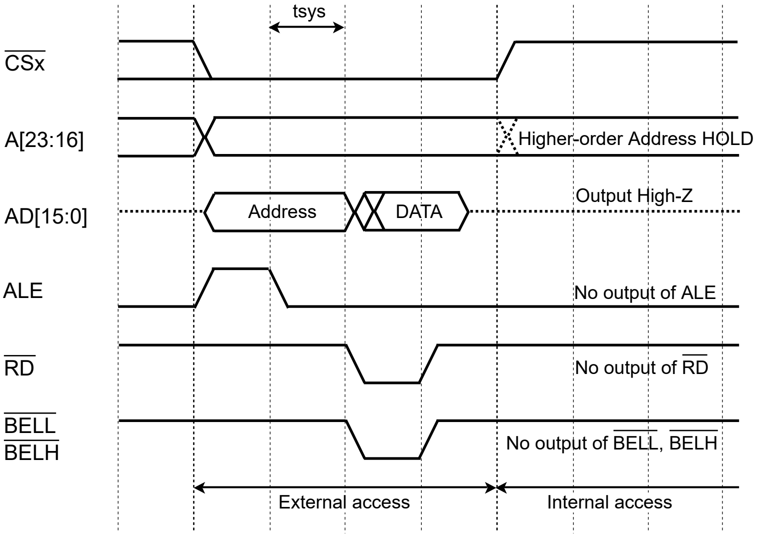

tsys
$\overline{\text{CSx}}$
A[23:16]
Higher-order Address HOLD
AD[15:0]
Address
DATA
Output High-Z
ALE
No output of ALE
$\overline{\text{RD}}$
No output of $\overline{\text{RD}}$
$\overline{\text{BELL}}$
$\overline{\text{BELH}}$
No output of $\overline{\text{BELL}}$, $\overline{\text{BELH}}$
External access
Internal access

Figure 10-1 Read Operation Timing

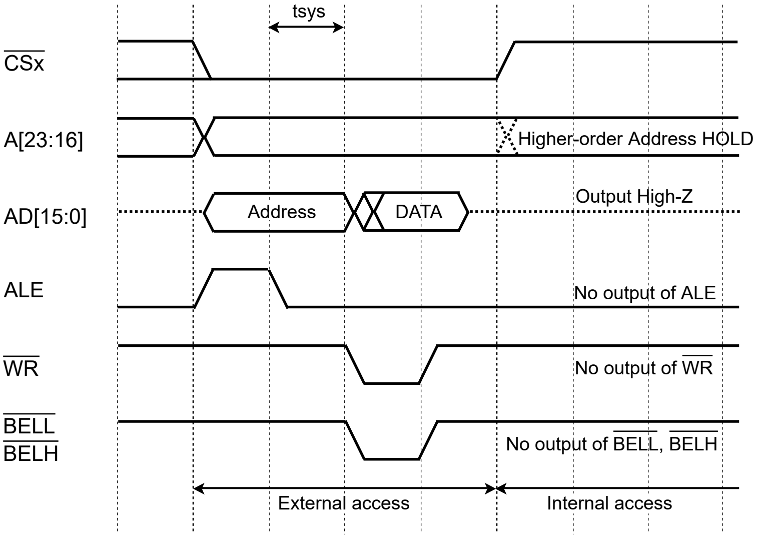

tsys
$\overline{\text{CSx}}$
A[23:16]
Higher-order Address HOLD
AD[15:0]
Address
DATA
Output High-Z
ALE
No output of ALE
$\overline{\text{WR}}$
No output of $\overline{\text{WR}}$
$\overline{\text{BELL}}$
$\overline{\text{BELH}}$
No output of $\overline{\text{BELL}}$, $\overline{\text{BELH}}$
External access
Internal access

Figure 10-2 Write Operation Timing

10.5.2 Wait timing

A wait cycle can be inserted for each channel by using the chip selector (CS) and wait controller.

The following wait can be inserted.

- A wait of up to 15 clocks can be automatically inserted.

The setting of the number of waits to be automatically inserted and the setting can be made using the chip select control registers, EXBCSx<CSIW[4:0]>.

Figure 10-3 through Figure 10-4 show the timing diagrams in which waits have been inserted.

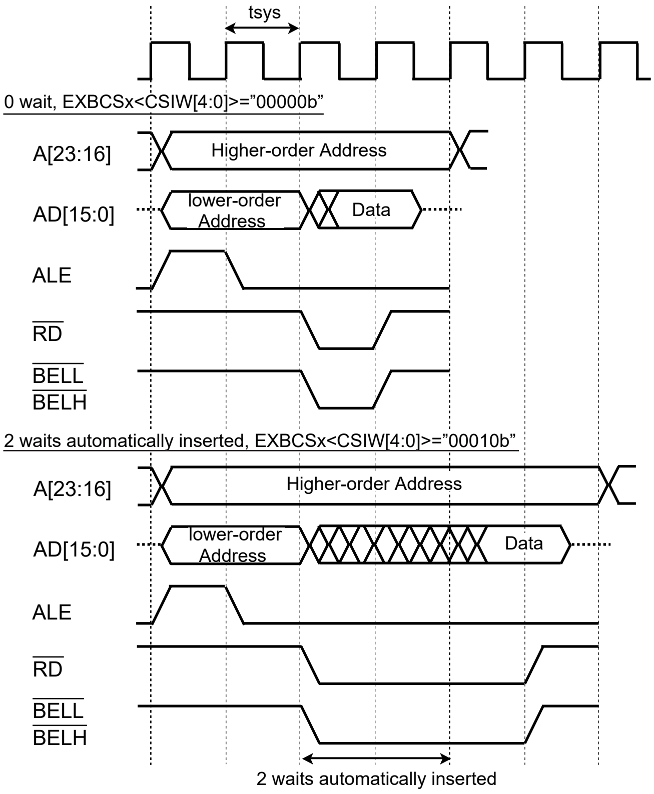

tsys
0 wait, EXBCSx<CSIW[4:0]>="00000b"
A[23:16]
Higher-order Address
AD[15:0]
lower-order Address
Data
ALE
$\overline{RD}$
$\overline{BELL}$
$\overline{BELH}$
2 waits automatically inserted, EXBCSx<CSIW[4:0]>="00010b"
A[23:16]
Higher-order Address
AD[15:0]
lower-order Address
Data
ALE
$\overline{RD}$
$\overline{BELL}$
$\overline{BELH}$
2 waits automatically inserted

Figure 10-3 Read Operation Timing

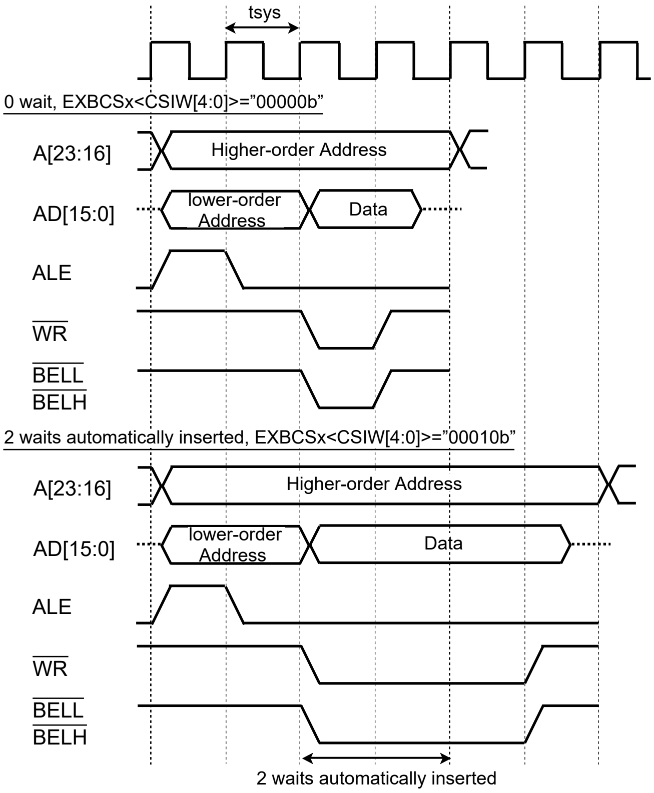

tsys
0 wait, EXBCSx<CSIW[4:0]>="00000b"
A[23:16]
Higher-order Address
AD[15:0]
lower-order Address
Data
ALE
WR
BELL
BELH
2 waits automatically inserted, EXBCSx<CSIW[4:0]>="00010b"
A[23:16]
Higher-order Address
AD[15:0]
lower-order Address
Data
ALE
WR
BELL
BELH
2 waits automatically inserted

Figure 10-4 Write Operation Timing

10.5.3 Time that it takes before ALE is asserted

One of system clocks of 1, 2 or 4 can be selected as the time that it takes before ALE is asserted. The setting can be made using the chip select control registers, EXBCSx<ALEW[1:0]>. The default is asserted the RD or WR signal from the address is generated after 2 clocks.

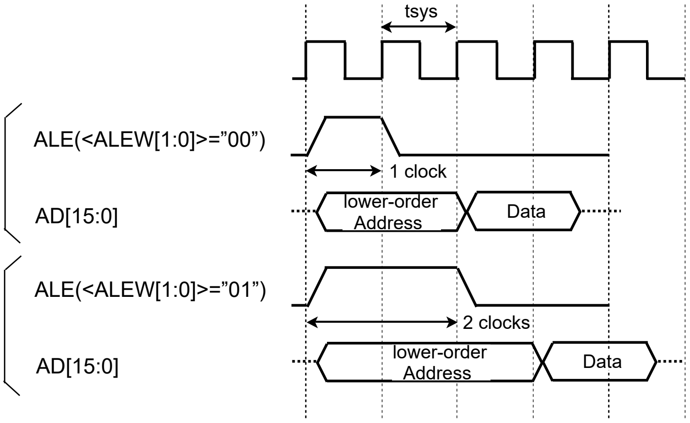

tsys
ALE(<ALEW[1:0]>="00")
1 clock
AD[15:0]
lower-order Address
Data
ALE(<ALEW[1:0]>="01")
2 clocks
AD[15:0]
lower-order Address
Data

Figure 10-5 Time that it takes before ALE is asserted

Figure 10-6 shows the timing when the ALE is 1 clock or 2 clocks.

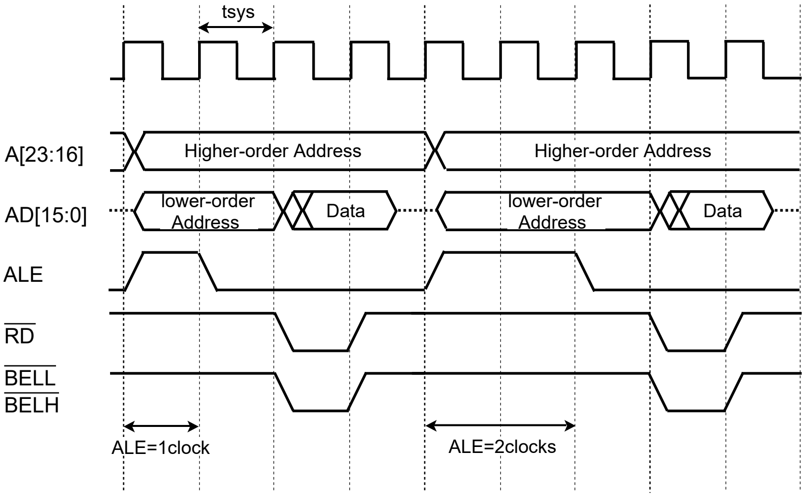

tsys
A[23:16]
Higher-order Address
Higher-order Address
AD[15:0]
lower-order Address
Data
lower-order Address
Data
ALE
RD
BELL
BELH
ALE=1clock
ALE=2clocks

Figure 10-6 Read Operation Timing (When the ALE is 1 Clock or 2 Clocks)

10.5.4 Read and Write Recovery Time

If access to external areas occurs consecutively, a dummy cycle can be inserted for recovery time.

A dummy cycle can be inserted in both a read and a write cycle. The dummy cycle insertion setting can be made in the chip select control registers, EXBCSx<WRR[2:0]> (write recovery cycle) and <RDR[2:0]> (read recovery cycle). As for the number of dummy cycles, none, one to six or eight system clocks (internal) can be specified for each channel. Figure 10-7 shows the timing of recovery time insertion.

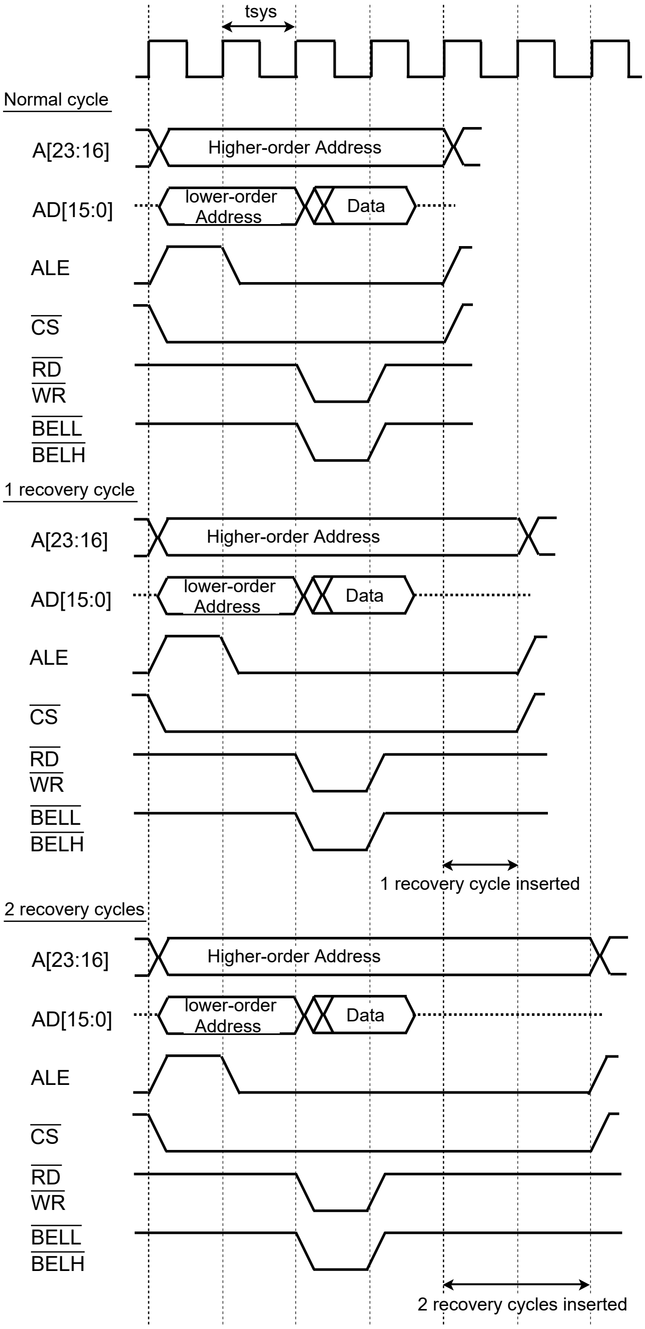

tsys
Normal cycle
A[23:16]
Higher-order Address
AD[15:0]
lower-order Address
Data
ALE
CS
RD
WR
BELL
BELH
1 recovery cycle
A[23:16]
Higher-order Address
AD[15:0]
lower-order Address
Data
ALE
CS
RD
WR
BELL
BELH
1 recovery cycle inserted
2 recovery cycles
A[23:16]
Higher-order Address
AD[15:0]
lower-order Address
Data
ALE
CS
RD
WR
BELL
BELH
2 recovery cycles inserted

Figure 10-7 Timing of Recovery Time Insertion

10.5.5 Chip select recovery time

If access to external areas occurs consecutively, a dummy cycle can be inserted for recovery time.

The dummy cycle insertion setting can be made in the chip select control registers, EXBCSx<CSR[1:0]>. As for the number of dummy cycles, none, one, two and four system clocks (internal) can be specified for each channel. Figure 10-8 shows the timing of recovery time insertion.

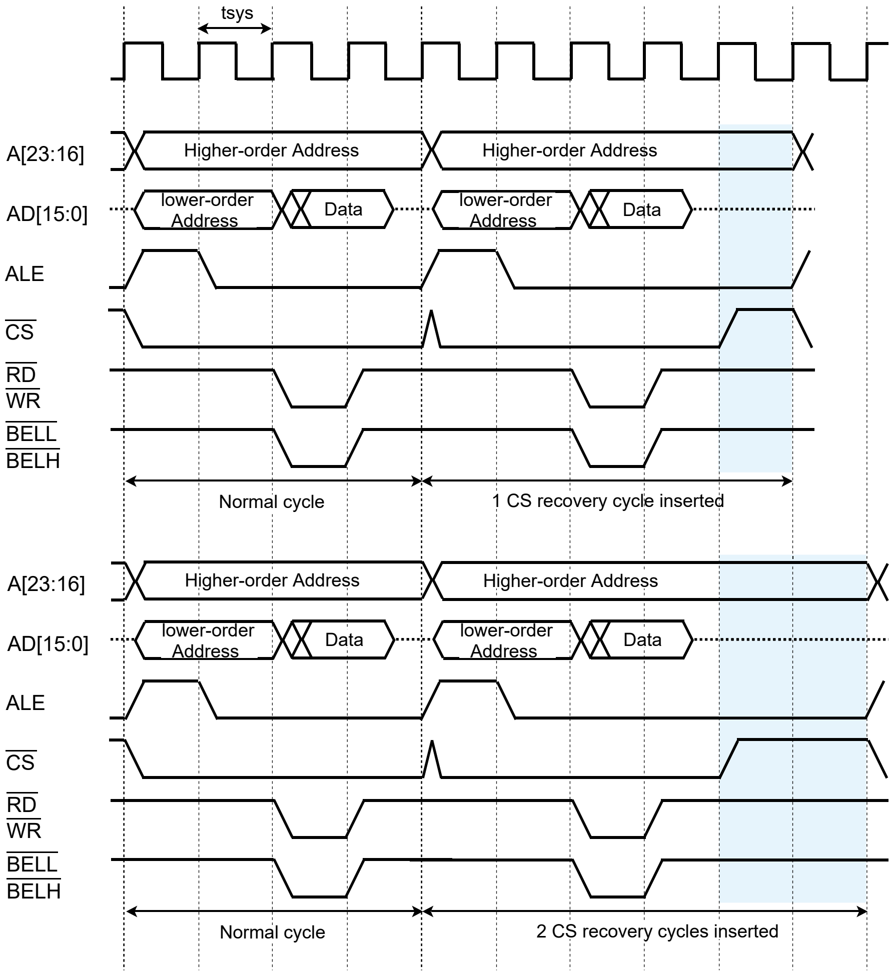

tsys
A[23:16]
Higher-order Address
Higher-order Address
AD[15:0]
lower-order Address
Data
lower-order Address
Data
ALE
CS
RD
WR
BELL
BELH
Normal cycle
1 CS recovery cycle inserted
A[23:16]
Higher-order Address
Higher-order Address
AD[15:0]
lower-order Address
Data
lower-order Address
Data
ALE
CS
RD
WR
BELL
BELH
Normal cycle
2 CS recovery cycles inserted

Figure 10-8 Timing of CS Recovery Time Insertion (ALE width: 1 clock)

10.5.6 Read and Write setup cycle

A read and a write setup cycle can be inserted for each channel by using the chip selector (CS) and wait controller.

The following can be inserted.

- A read and a write setup cycle of up to 4 clocks can be automatically inserted.

The setting of the number of setup cycles to be automatically inserted and the setting can be made using the chip select control registers, EXBCSx<WRS[1:0]> and <RDS[1:0]>.

Figure 10-9 shows the timing diagrams in which the read or write setup cycle have been inserted.

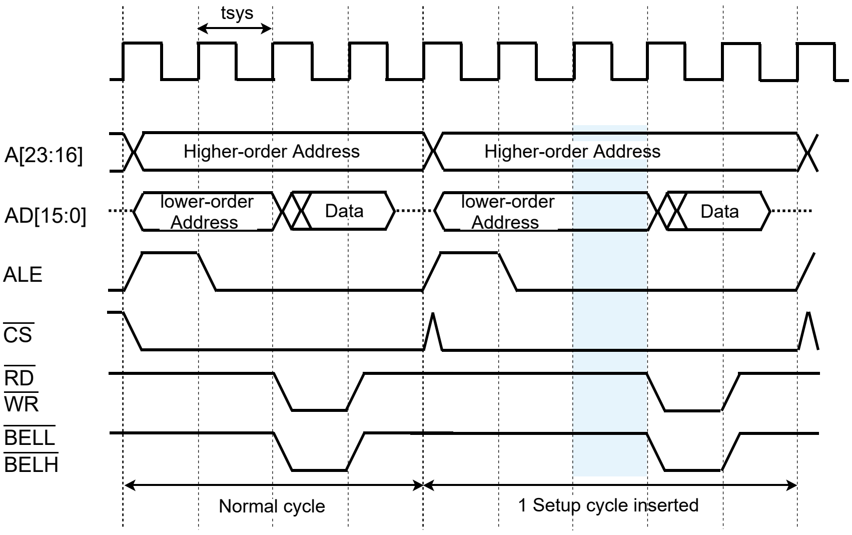

tsys
A[23:16]
Higher-order Address
Higher-order Address
AD[15:0]
lower-order Address
Data
lower-order Address
Data
ALE
$\overline{CS}$
$\overline{RD}$
$\overline{WR}$
$\overline{BELL}$
$\overline{BELH}$
Normal cycle
1 Setup cycle inserted

Figure 10-9 Timing of Read or Write Setup Time Insertion

10.6 Connection example for external memory

10.6.1 Connection Example for external 16-bit SRAM and NOR-Flash (Multiplex bus)

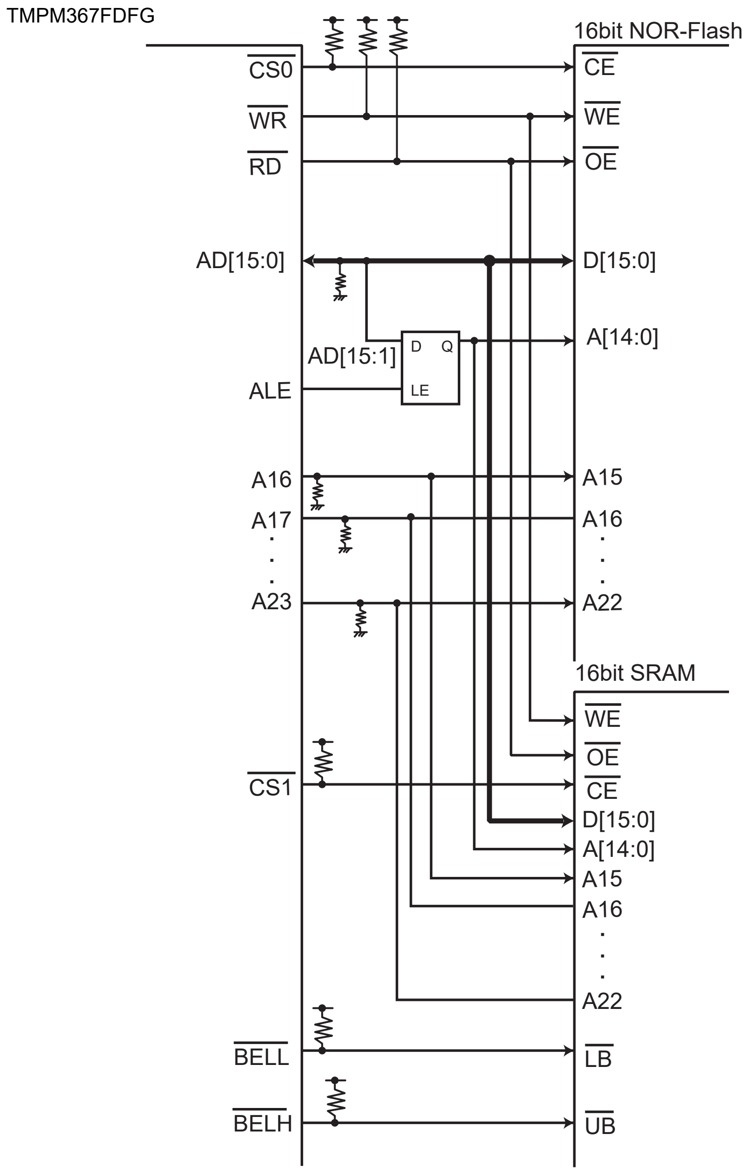

TMPM367FDFG
16bit NOR-Flash
$\overline{CS0}$
$\overline{WR}$
$\overline{RD}$
AD[15:0]
AD[15:1]
ALE
A16
A17
A23
$\overline{CS1}$
$\overline{BELL}$
$\overline{BELH}$
D
Q
LE
$\overline{CE}$
$\overline{WE}$
$\overline{OE}$
D[15:0]
A[14:0]
A15
A16
A22
16bit SRAM
$\overline{WE}$
$\overline{OE}$
$\overline{CE}$
D[15:0]
A[14:0]
A15
A16
A22
$\overline{LB}$
$\overline{UB}$

Figure 10-10 Connection Example for external 16-bit SRAM and NOR-Flash (Multiplex bus)

11. 16-bit Timer / Event Counters (TMRB)

11.1 Outline

TMRB operate in the following four operation modes:

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

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

- 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.

11.2 Differences in the Specifications

TMPM367FDFG contains 8-channel of TMRB.

Each channel functions independently and the channels operate in the same way except for the differences in their specification as shown in Table 11-1.

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

Table 11-1 Differences in the Specifications of TMRB Modules

Specification	External pins		Trigger function between timers		Interrupt		Internal Connects		
Channel	Timer Flip-Flop output pins	External clock / capture trigger input pins	Capture trigger	Synchronous start trigger channel	Capture interrupt	TMRB interrupt	Start AD conversion Start DAC conversion	Timer Flip-Flop Connect with SIO/ UART, RMC (TXTRG : Transfer clock)	μDMA request (DMATMRB compare match0/1 overflow (channel0 to 4))
TMRB0	TB0OUT	TB0IN	TB7OUT	–	INTCAP00 INTCAP01	INTTB0	-	-	INTTB0
TMRB1	TB1OUT	TB1IN	TB7OUT	TB0PRUN TB0RUN	INTCAP10 INTCAP11	INTTB1	-	RMC	INTTB1
TMRB2	TB2OUT	TB2IN	TB7OUT	TB0PRUN TB0RUN	INTCAP20 INTCAP21	INTTB2	INTTB21	-	INTTB2
TMRB3	TB3OUT	-	TB2OUT	TB0PRUN TB0RUN	-	INTTB3	INTTB31	-	INTTB3
TMRB4	TB4OUT	-	TB2OUT	–	-	INTTB4	INTTB41	SIO0 SIO1	INTTB4
TMRB5	TB5OUT	(TB5IN) (note)	TB2OUT	TB4PRUN TB4RUN	INTCAP50 INTCAP51	INTTB5	INTTB51	-	-
TMRB6	TB6OUT	-	TB5OUT	TB4PRUN TB4RUN	-	INTTB6	INTTB61	-	-
TMRB7	TB7OUT	-	TB5OUT	TB4PRUN TB4RUN	-	INTTB7	INTTB71	SIO2 SIO3	-

Note: fs is connected to T5BIN in TMPM367FDFG. For more detail, refer to internal High-speed Oscillation Adjustment function.

11.3 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.

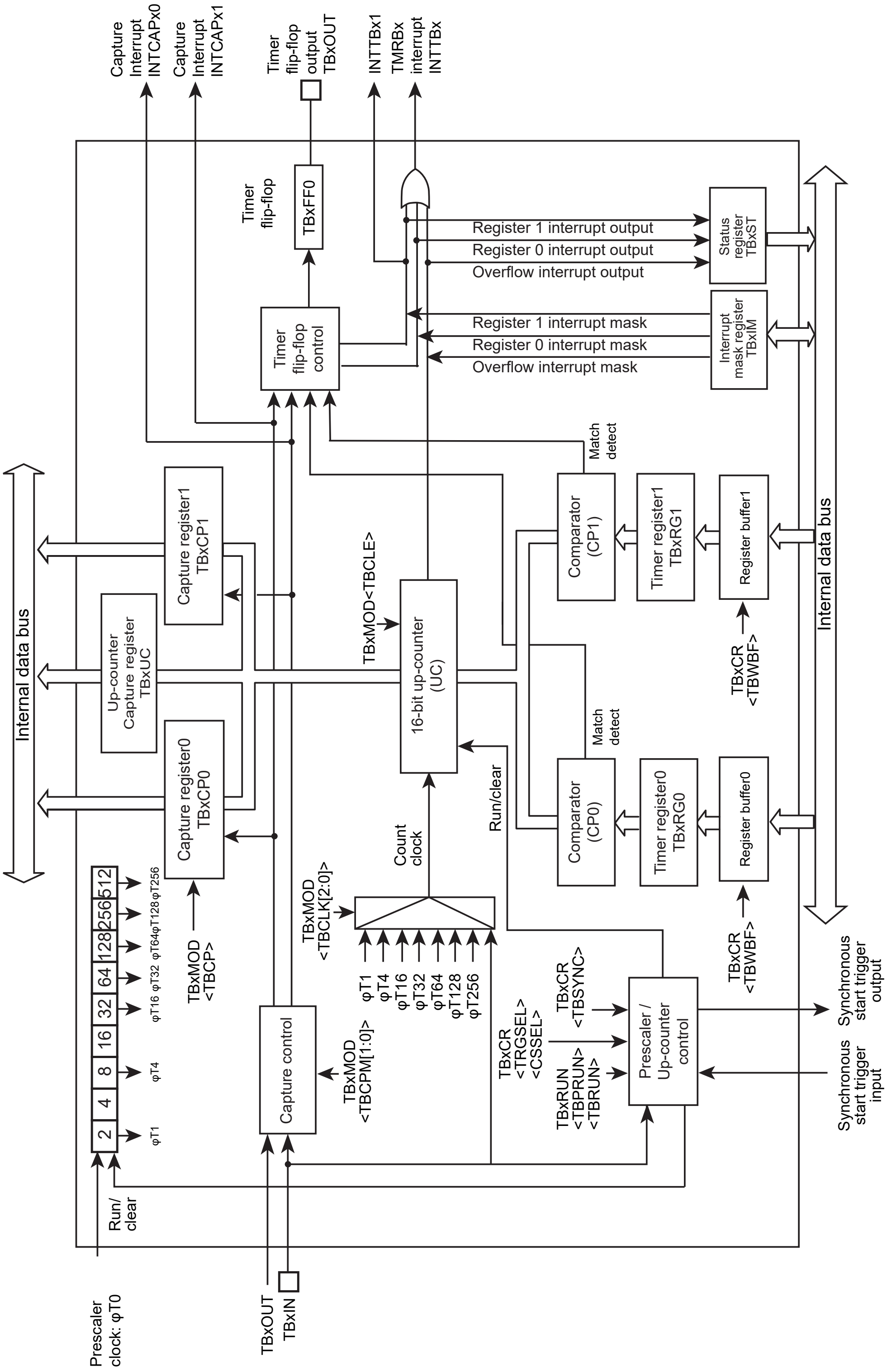

Capture Interrupt INTCAPx0
Capture Interrupt INTCAPx1
Timer flip-flop output TBxOUT
INTTBx1
TMRBx interrupt INTTBx
Timer flip-flop
TBxFF0
Timer flip-flop control
Register 1 interrupt output
Register 0 interrupt output
Overflow interrupt output
Status register TBxST
Register 1 interrupt mask
Register 0 interrupt mask
Overflow interrupt mask
Interrupt mask register TBxIM
Match detect
Internal data bus
Capture register1 TBxCP1
Up-counter Capture register TBxUC
Capture register0 TBxCP0
TBxMOD<TBCLE>
16-bit up-counter (UC)
Comparator (CP1)
Timer register1 TBxRG1
Register buffer1
TBxCR <TBWBF>
Internal data bus
Match detect
Comparator (CP0)
Timer register0 TBxRG0
Register buffer0
Run/clear
Count clock
TBxMOD <TBCLK[2:0]>
2
4
8
16
32
64
128
256
512
φT1
φT4
φT16
φT32
φT64
φT128
φT256
TBxMOD <TBCP>
φT1 φT4 φT16 φT32 φT64 φT128 φT256
TBxMOD <TBCPM[1:0]>
Capture control
TBxCR <TRGSEL> <CSSEL>
TBxCR <TBSYNC>
TBxRUN <TBPRUN> <TBRUN>
Prescaler / Up-counter control
TBxCR <TBWBF>
Synchronous start trigger output
Synchronous start trigger input
Run/ clear
Prescaler clock: φT0
TBxOUT
TBxIN

Figure 11-1 TMRBx Block Diagram

11.4 Registers

11.4.1 Register list according to channel

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

Channel x	**Base Address**
Channel 0	0x400C_4000
Channel 1	0x400C_4100
Channel 2	0x400C_4200
Channel 3	0x400C_4300
Channel 4	0x400C_4400
Channel 5	0x400C_4500
Channel 6	0x400C_4600
Channel 7	0x400C_4700

Register name **(x=0 to F)**		**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

Note: During timer operation, timer control register, timer mode register and timer Flip-Flop control register should not be modified. After stopping timer operation, they should be modified.

11.4.2 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	TBEN	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".

11.4.3 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.

11.4.4 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	TBINSEL	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	TBINSEL	R/W	Select external input 0: TBxIN 1: Reserved This bit should be written to "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

11.4.5 TBxMOD (Mode register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	−	R/W	Write as "0".
6	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".
5-4	TBCPM[1:0]	R/W	Capture timing 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. (x=7, n=0, 1, 2), (x=2, n=3, 4, 5), (x=5, n=6, 7), (TMRB0 to 2: TB7OUT, TMRB3 to 5: TB2OUT, TMRB6 to 7: TB5OUT)
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. 000: TBxIN pin input 001: φT1 010: φT4 011: φT16 100: φT32 101: φT64 110: φT128 111: φT256

Note:<TBCPM[1:0]> in TBxMOD (x=2, 5, 7) should not be set "11"

11.4.6 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".

11.4.7 TBxST (Status register)

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

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

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

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

Note 3: Even if TBxIM setting is done, TBxST is set.

11.4.8 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.

11.4.9 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.

11.4.10 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.

11.4.11 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.

11.4.12 TBxCP0 (Capture register 0)

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

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

11.4.13 TBxCP1 (Capture register 1)

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

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

11.5 Description of Operations for Each Circuit

The channels operate in the same way, except for the differences in their specifications as shown in Table 11-1.

11.5.1 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> 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 11-2 Prescaler Output Clock Resolutions (fc = 80MHz)

Select peripheral clock CGSYSCR <FPSEL>	Select gear clock CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
0 (fgear)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.025 μs)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)
		001 (fperiph/2)	fc/2^{2} (0.05 μs)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)
		010 (fperiph/4)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		011 (fperiph/8)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		100 (fperiph/16)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		101 (fperiph/32)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^{2} (0.05 μs)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)
		001 (fperiph/2)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		010 (fperiph/4)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		011 (fperiph/8)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		100 (fperiph/16)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
		101 (fperiph/32)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		001 (fperiph/2)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		010 (fperiph/4)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		011 (fperiph/8)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
		100 (fperiph/16)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)
		101 (fperiph/32)	fc/2^{8} (3.2μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
	110 (fc/8)	000 (fperiph/1)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		001 (fperiph/2)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		010 (fperiph/4)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
		011 (fperiph/8)	fc/2^{7} (1.6μs)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)
		100 (fperiph/16)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
		101 (fperiph/32)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)	fc/2^{13} (102.4 μs)
	111 (fc/16)	000 (fperiph/1)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		001 (fperiph/2)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
		010 (fperiph/4)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)
		011 (fperiph/8)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
		100 (fperiph/16)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)	fc/2^{13} (102.4 μs)
		101 (fperiph/32)	fc/2^{10} (12.8μs)	fc/2^{12} (51.2 μs)	fc/2^{14} (204.8 μs)

Table 11-2 Prescaler Output Clock Resolutions (fc = 80MHz)

Select peripheral clock CGSYSCR <FPSEL>	Select gear clock CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
1 (fc)	000 (fc)	000 (fperiph/1)	$fc/2^{1}$ (0.025 μs)	$fc/2^{3}$ (0.1 μs)	$fc/2^{5}$ (0.4 μs)
		001 (fperiph/2)	$fc/2^{2}$ (0.05 μs)	$fc/2^{4}$ (0.2 μs)	$fc/2^{6}$ (0.8 μs)
		010 (fperiph/4)	$fc/2^{3}$ (0.1 μs)	$fc/2^{5}$ (0.4 μs)	$fc/2^{7}$ (1.6 μs)
		011 (fperiph/8)	$fc/2^{4}$ (0.2 μs)	$fc/2^{6}$ (0.8 μs)	$fc/2^{8}$ (3.2 μs)
		100 (fperiph/16)	$fc/2^{5}$ (0.4 μs)	$fc/2^{7}$ (1.6 μs)	$fc/2^{9}$ (6.4 μs)
		101 (fperiph/32)	$fc/2^{6}$ (0.8 μs)	$fc/2^{8}$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
	100 (fc/2)	000 (fperiph/1)	–	$fc/2^{3}$ (0.1 μs)	$fc/2^{5}$ (0.4 μs)
		001 (fperiph/2)	$fc/2^{2}$ (0.05 μs)	$fc/2^{4}$ (0.2 μs)	$fc/2^{6}$ (0.8 μs)
		010 (fperiph/4)	$fc/2^{3}$ (0.1 μs)	$fc/2^{5}$ (0.4 μs)	$fc/2^{7}$ (1.6 μs)
		011 (fperiph/8)	$fc/2^{4}$ (0.2 μs)	$fc/2^{6}$ (0.8 μs)	$fc/2^{8}$ (3.2 μs)
		100 (fperiph/16)	$fc/2^{5}$ (0.4 μs)	$fc/2^{7}$ (1.6 μs)	$fc/2^{9}$ (6.4 μs)
		101 (fperiph/32)	$fc/2^{6}$ (0.8 μs)	$fc/2^{8}$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
	101 (fc/4)	000 (fperiph/1)	–	$fc/2^{3}$ (0.1 μs)	$fc/2^{5}$ (0.4 μs)
		001 (fperiph/2)	–	$fc/2^{4}$ (0.2 μs)	$fc/2^{6}$ (0.8 μs)
		010 (fperiph/4)	$fc/2^{3}$ (0.1 μs)	$fc/2^{5}$ (0.4 μs)	$fc/2^{7}$ (1.6 μs)
		011 (fperiph/8)	$fc/2^{4}$ (0.2 μs)	$fc/2^{6}$ (0.8 μs)	$fc/2^{8}$ (3.2 μs)
		100 (fperiph/16)	$fc/2^{5}$ (0.4 μs)	$fc/2^{7}$ (1.6 μs)	$fc/2^{9}$ (6.4 μs)
		101 (fperiph/32)	$fc/2^{6}$ (0.8 μs)	$fc/2^{8}$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
	110 (fc/8)	000 (fperiph/1)	–	–	$fc/2^{5}$ (0.4 μs)
		001 (fperiph/2)	–	$fc/2^{4}$ (0.2 μs)	$fc/2^{6}$ (0.8 μs)
		010 (fperiph/4)	–	$fc/2^{5}$ (0.4 μs)	$fc/2^{7}$ (1.6 μs)
		011 (fperiph/8)	$fc/2^{4}$ (0.2 μs)	$fc/2^{6}$ (0.8 μs)	$fc/2^{8}$ (3.2 μs)
		100 (fperiph/16)	$fc/2^{5}$ (0.4 μs)	$fc/2^{7}$ (1.6 μs)	$fc/2^{9}$ (6.4 μs)
		101 (fperiph/32)	$fc/2^{6}$ (0.8 μs)	$fc/2^{8}$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
	111 (fc/16)	000 (fperiph/1)	–	–	$fc/2^{5}$ (0.4 μs)
		001 (fperiph/2)	–	–	$fc/2^{6}$ (0.8 μs)
		010 (fperiph/4)	–	$fc/2^{5}$ (0.4 μs)	$fc/2^{7}$ (1.6 μs)
		011 (fperiph/8)	–	$fc/2^{6}$ (0.8 μs)	$fc/2^{8}$ (3.2 μs)
		100 (fperiph/16)	$fc/2^{5}$ (0.4 μs)	$fc/2^{7}$ (1.6 μs)	$fc/2^{9}$ (6.4 μs)
		101 (fperiph/32)	$fc/2^{6}$ (0.8 μs)	$fc/2^{8}$ (3.2 μs)	$fc/2^{10}$ (12.8 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 11-3 Prescaler Output Clock Resolutions (fc = 80MHz)

Select peripheral clock CGSYSCR <FPSEL0>	Select gear clock CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function			
			φT32	φT64	φT128	φT256
0 (fgear)	000 (fc)	000 (fperiph/1)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)
		001 (fperiph/2)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)
		010 (fperiph/4)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)
		011 (fperiph/8)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)
		100 (fperiph/16)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)
		101 (fperiph/32)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)
		001 (fperiph/2)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)
		010 (fperiph/4)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)
		011 (fperiph/8)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)
		100 (fperiph/16)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)
		101 (fperiph/32)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)	fc/2^{15} (409.6 μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)
		001 (fperiph/2)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)
		010 (fperiph/4)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)
		011 (fperiph/8)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)
		100 (fperiph/16)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)	fc/2^{15} (409.6 μs)
		101 (fperiph/32)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)	fc/2^{15} (409.6 μs)	fc/2^{16} (819.2 μs)
	110 (fc/8)	001 (fperiph/2)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)
		010 (fperiph/4)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)
		011 (fperiph/8)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)
		100 (fperiph/16)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)	fc/2^{15} (409.6 μs)
		101 (fperiph/32)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)	fc/2^{15} (409.6 μs)	fc/2^{16} (819.2 μs)
		101 (fperiph/32)	fc/2^{14} (204.8 μs)	fc/2^{15} (409.6 μs)	fc/2^{16} (819.2 μs)	fc/2^{17} (1638.4 μs)
	111 (fc/16)	001 (fperiph/2)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)
		010 (fperiph/4)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)
		011 (fperiph/8)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)	fc/2^{15} (409.6 μs)
		100 (fperiph/16)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)	fc/2^{15} (409.6 μs)	fc/2^{16} (819.2 μs)
		101 (fperiph/32)	fc/2^{14} (204.8 μs)	fc/2^{15} (409.6 μs)	fc/2^{16} (819.2 μs)	fc/2^{17} (1638.4 μs)
		101 (fperiph/32)	fc/2^{15} (405.6 μs)	fc/2^{16} (819.2 μs)	fc/2^{16} (1638.4 μs)	fc/2^{17} (3276.8 μs)

Table 11-3 Prescaler Output Clock Resolutions (fc = 80MHz)

Select peripheral clock CGSYSCR <FPSEL0>	Select gear clock CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function			
			φT32	φT64	φT128	φT256
1 (fc)	000 (fc)	000 (fperiph/1)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)
		001 (fperiph/2)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)
		010 (fperiph/4)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)
		011 (fperiph/8)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)
		100 (fperiph/16)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)
		101 (fperiph/32)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)
		001 (fperiph/2)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)
		010 (fperiph/4)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)
		011 (fperiph/8)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)
		100 (fperiph/16)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)
		101 (fperiph/32)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)
		001 (fperiph/2)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)
		010 (fperiph/4)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)
		011 (fperiph/8)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)
		100 (fperiph/16)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)
		101 (fperiph/32)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)
	110 (fc/8)	000 (fperiph/1)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)
		001 (fperiph/2)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)
		010 (fperiph/4)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)
		011 (fperiph/8)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)
		100 (fperiph/16)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)
		101 (fperiph/32)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 μs)
	111 (fc/16)	000 (fperiph/1)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)
		001 (fperiph/2)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)
		010 (fperiph/4)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)
		011 (fperiph/8)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)
		100 (fperiph/16)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)
		101 (fperiph/32)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)	fc/2^{13} (102.4 μs)	fc/2^{14} (204.8 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.

11.5.2 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, φT16, φT32, φT64, φT128 and φT256 - 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.

11.5.3 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.

11.5.4 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>.

11.5.5 Capture registers (TBxCP0, TBxCP1)

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

11.5.6 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.

11.5.7 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.

11.5.8 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.

11.5.9 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.

11.6 Description of Operations for Each Mode

11.6.1 16-bit interval Timer Mode

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

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits INTTBx interrupt by setting corresponding bit to "1".
TBxFFCR	←	X	X	0	0	0	0	1	1	Disable to TBxFF0 reverse trigger
TBxMOD	←	X	1	0	0	1	*	*	*	Changes to prescaler output clock as input clock. Specifies capture function to disable.
							(*** = 001 to 111)			
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a time interval. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note: X; Don’t care −; No change

11.6.2 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

11.6.3 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

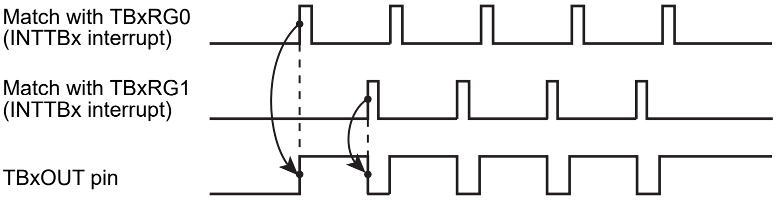

Match with TBxRG0
(INTTBx interrupt)
Match with TBxRG1
(INTTBx interrupt)
TBxOUT pin

Figure 11-2 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.

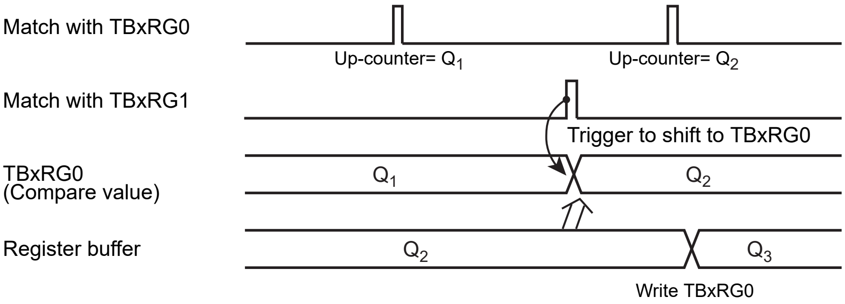

Match with TBxRG0
Up-counter= Q1
Up-counter= Q2
Match with TBxRG1
Trigger to shift to TBxRG0
TBxRG0
(Compare value)
Q1
Q2
Register buffer
Q2
Q3
Write TBxRG0

Figure 11-3 Register Buffer Operation

The block diagram of this mode is shown below.

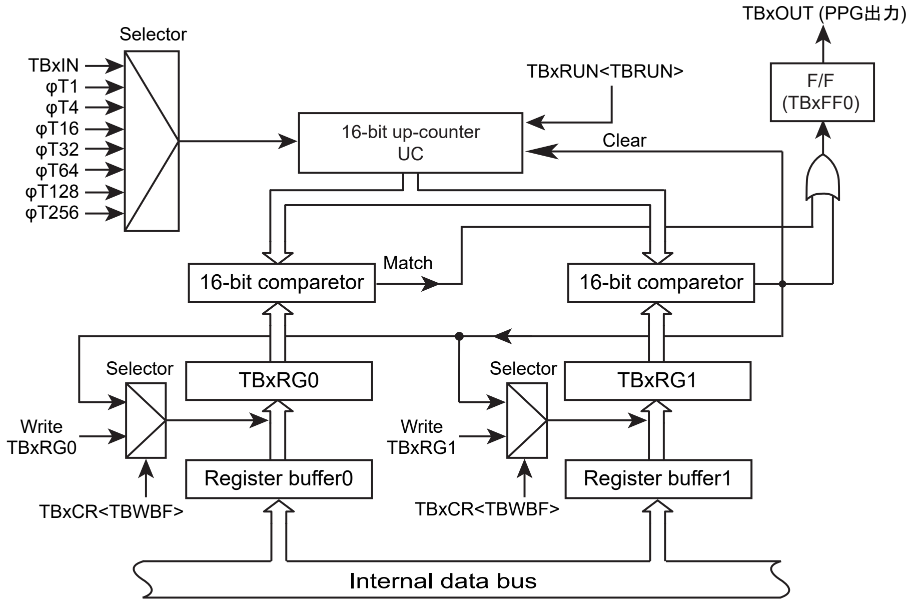

TBxOUT (PPG出力)
Selector
TBxIN
φT1
φT4
φT16
φT32
φT64
φT128
φT256
TBxRUN<TBRUN>
F/F
(TBxFF0)
16-bit up-counter
UC
Clear
Match
16-bit comparator
16-bit comparator
Selector
TBxRG0
Selector
TBxRG1
Write
TBxRG0
Write
TBxRG1
Register buffer0
Register buffer1
TBxCR<TBWBF>
TBxCR<TBWBF>
Internal data bus

Figure 11-4 Block Diagram of 16-bit PPG Mode

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

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
TBxCR	←	0	0	−	X	−	X	X	X	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	1	0	0	1	*	*	*	Designates the prescaler output clock as the input clock, and disables the capture function.
						(*** = 001 to 111)				
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

11.6.4 Timer synchronous mode

This mode enables the timers to start synchronously.

If the mode is used with PPG output, the output can be applied to drive a motor.

TMRB is consisted of pairs of 4-channel TMRB. If one channel starts, remaining 3 channels can be start synchronously. In the TMPM367FDFG, 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 1: 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.

Note 2: <TBSYNC> bit of TMRB0 and TMRB4 which are the trigger of timer start is always clear to "0".

11.6.5 External trigger count start mode

A timer can be started by an external signal when using external trigger count start mode.

Select count start by TBxCR<CSSEL>.

- <CSSEL>="0": A timer is operated with each timing.
- <CSSEL>="1": A timer is started by an external signal.

The edge of trigger is selected by TBxCR<TRGSEL>.

- <TRGSEL>="0": Select a rising edge of TBxIN input.
- <TRGSEL>="1": Select a falling edge of TBxIN input.

In case of setting timer synchronous mode, it has higher priority than this mode.

11.7 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

11.7.1 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 11-5 One-shot Pulse Output (With Delay)".

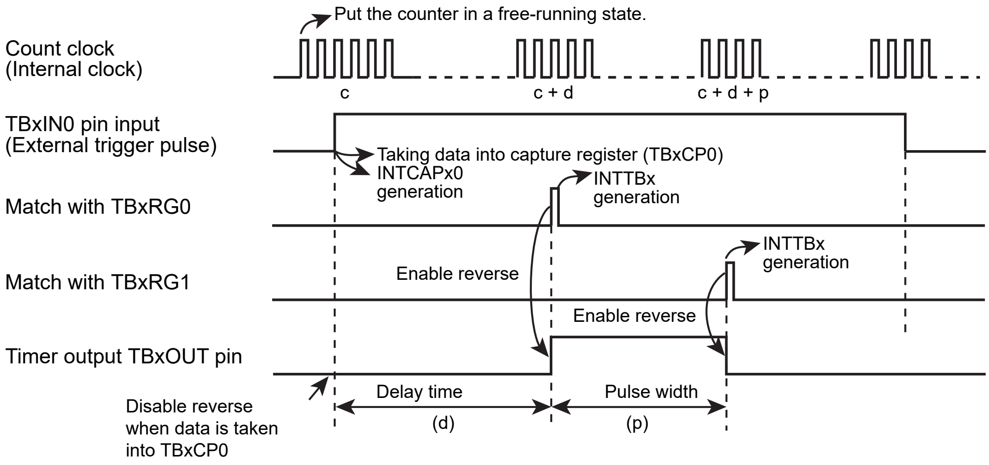

Put the counter in a free-running state.
Count clock
(Internal clock)
c
c + d
c + d + p
TBxIN0 pin input
(External trigger pulse)
Taking data into capture register (TBxCP0)
INTCAPx0
generation
INTTBx
generation
Match with TBxRG0
INTTBx
generation
Enable reverse
Match with TBxRG1
Enable reverse
Timer output TBxOUT pin
Disable reverse
when data is taken
into TBxCP0
Delay time
(d)
Pulse width
(p)

Figure 11-5 One-shot Pulse Output (With Delay)

The followings show the settings in the case that 2 ms width one-shot pulse is output after 3ms by triggering 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. (16bit) (TBxCAP0 + 3ms/φT1)
	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	Sets count value.(16bit) (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.

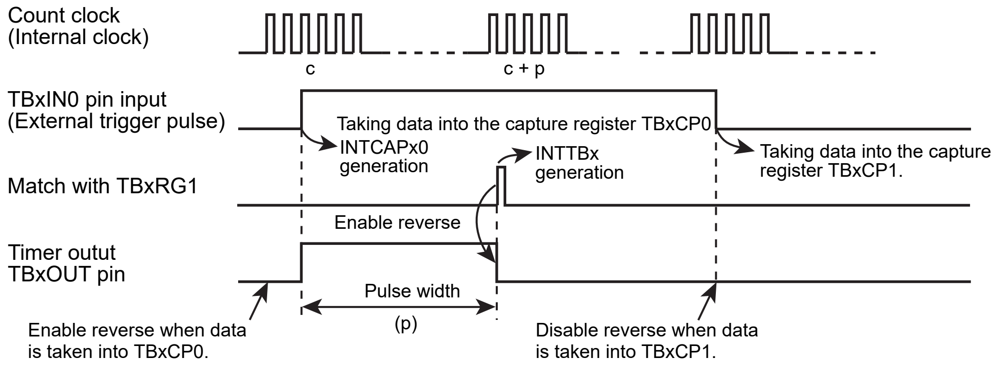

Count clock
(Internal clock)
c
c + p
TBxIN0 pin input
(External trigger pulse)
Taking data into the capture register TBxCP0
INTCAPx0
generation
INTTBx
generation
Taking data into the capture
register TBxCP1.
Match with TBxRG1
Enable reverse
Timer outut
TBxOUT pin
Pulse width
(p)
Enable reverse when data
is taken into TBxCP0.
Disable reverse when data
is taken into TBxCP1.

Figure 11-6 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

11.7.2 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 TMRB1 and TMRB0. TB0OUT of the 16-bit timer TMRB0 is used to specify the measurement time.

TMRB1 count clock selects TB1IN input and performs count operation by using external clock input. If TB1MOD<TBCPM[1:0]> is set "11", TMRB1 count clock takes the counter value into the TB1CP0 at the rising edge of TB0OUT and takes the counter value into TB1CP1 at the falling edge of TB0OUT.

This setting allows a count value of the 16-bit up-counter UC to be taken into the capture register (TB1CP0) upon rising of a timer flip-flop output (TB0OUT) of the 16-bit timer (TMRB0), and an UC counter value to be taken into the capture register (TB1CP1) upon falling of TB0OUT of the 16-bit timer (TMRB0).

A frequency is then obtained from the difference between TB1CP0 and TB1CP1 based on the measurement, by generating the INTTB0 16-bit timer interrupt.

For example, if the difference between TB1CP0 and TB1CP1 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).

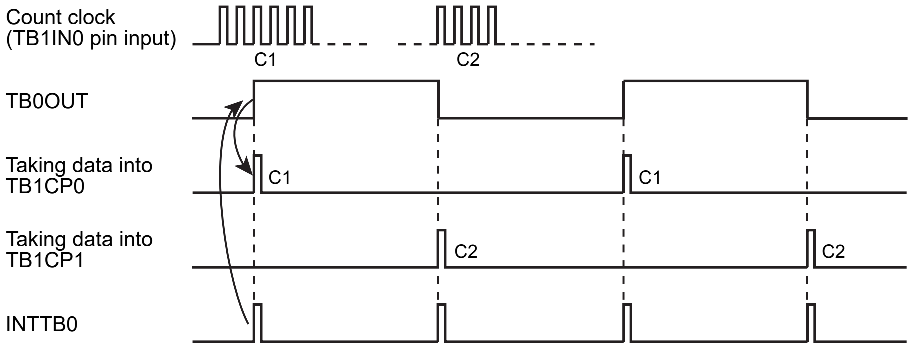


Figure 11-7 Frequency Measurement

11.7.3 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 TBxIN0 pin.

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

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

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

The "Low" level width of an external pulse can also be measured. In such cases, the difference between C2 generated the first time and C1 generated the second time is initially obtained by performing the second stage of INTCAPx0 interrupt processing as shown in "Figure 11-8 Pulse Width Measurement" and this difference is multiplied by the cycle of the prescaler output clock to obtain the "Low" level width.

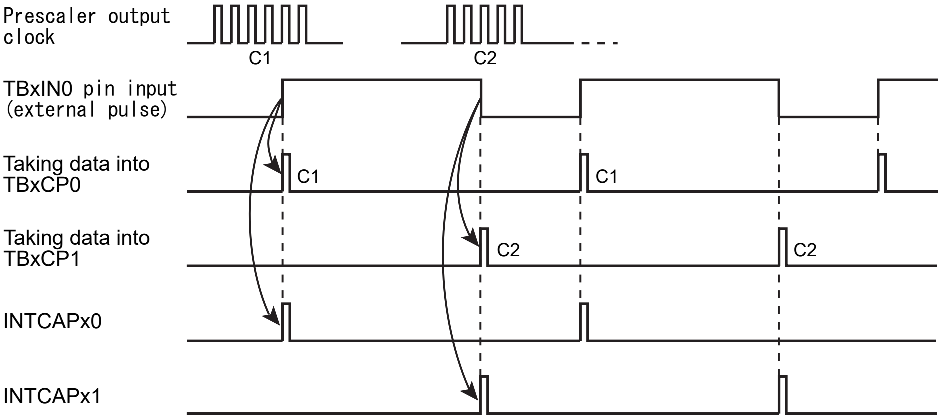

Prescaler output clock
C1
C2
TBxIN0 pin input (external pulse)
Taking data into TBxCP0
C1
C1
Taking data into TBxCP1
C2
C2
INTCAPx0
INTCAPx1

Figure 11-8 Pulse Width Measurement

12. Serial Channel (SIO/UART)

12.1 Overview

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

Their features are described as follows.

- Transfer Clock
  - Generate the transfer clock by dividing the peripheral clock (φT0) frequency into 1/2, 1/8, 1/32, 1/128.
  - The prescaler output clock frequency can be divided by each of the numbers from 1 to 16.
  - The prescaler output frequency can be divided by each of the numbers from 1, N+m/16 (N=2 to 15, m=1 to 15), and 16. (only UART mode)
  - The system clock is usable. (only UART mode)
- Double buffer / FIFO

  The double buffer function and the FIFO buffers (total of transmit and receive) can be used up to 4bytes.
- I/O Interface mode
  - Transfer Mode : the half duplex (transmit / receive) and the full duplex
  - Clock : Output (fixed rising edge) / Input (selectable rising / falling edge)
  - A time interval can be set within a range where continuous transmission is performed.
- UART Mode
  - Data length : 7, 8, 9 bits
  - Add parity bit (to be against 9 bits data length)
  - Serial links to use wake-up function
  - Handshaking function with $\overline{\text{CTS}}$ pin

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

12.2 Difference in the Specification of SIO / UART Modules

TMPM367FDFG has 4 channels.

Each channel function is not depended. The pins and interrupts for each channel are assigned as follows.

Table 12-1 Differences for each channels of SIO / UART Modules

	Pin name			Interrupt		Timer for serial clock	DMA
	TXD	RXD	$\overline{\text{CTS}}$ / SCLK	Receive interrupt	Transmit interrupt		
channel 0	PE2	PE1	PE3	INTRX0	INTTX0	TB4OUT	support
channel 1	PE5	PE6	PE4	INTRX1	INTTX1	TB4OUT	support
channel 2	PL2	PL1	PL3	INTRX2	INTTX2	TB7OUT	support
channel 3	PB0	PB1	PA7	INTRX3	INTTX3	TB7OUT	support

12.3 Configuration

Figure 12-1 shows SIO / UART block diagram.

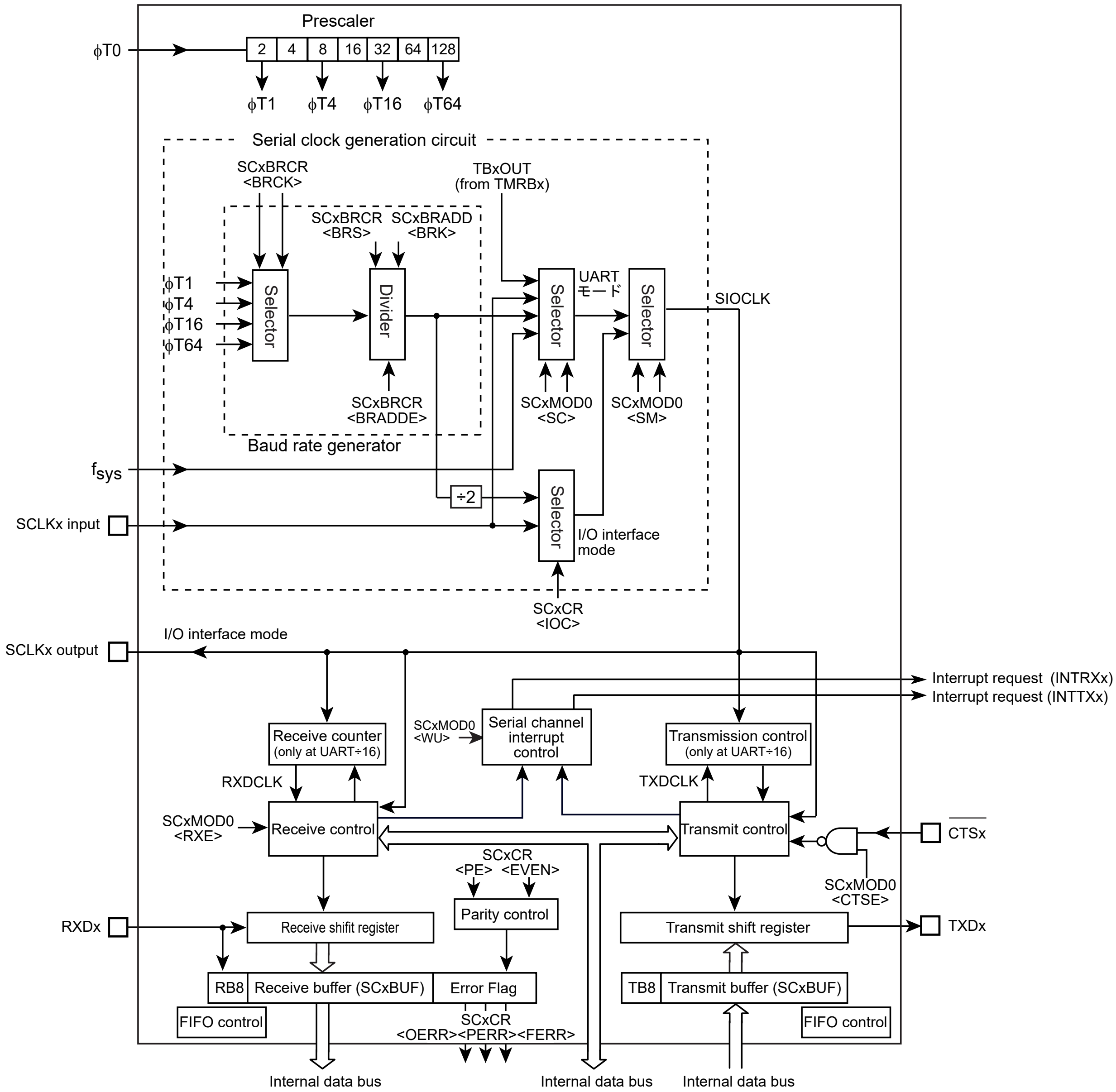

Prescaler
ϕT0
2
4
8
16
32
64
128
ϕT1
ϕT4
ϕT16
ϕT64
Serial clock generation circuit
SCxBRCR <BRCK>
TBxOUT (from TMRBx)
SCxBRCR <BRS>
SCxBRADD <BRK>
ϕT1
ϕT4
ϕT16
ϕT64
Selector
Divider
Selector
UART モード
Selector
SIOCLK
SCxBRCR <BRADDE>
SCxMOD0 <SC>
SCxMOD0 <SM>
Baud rate generator
fsys
÷2
SCLKx input
Selector
I/O interface mode
SCxCR <IOC>
SCLKx output
I/O interface mode
Interrupt request (INTRXx)
Interrupt request (INTTXx)
Receive counter (only at UART÷16)
SCxMOD0 <WU>
Serial channel interrupt control
Transmission control (only at UART÷16)
RXDCLK
TXDCLK
SCxMOD0 <RXE>
Receive control
Transmit control
CTSx
SCxCR <PE> <EVEN>
SCxMOD0 <CTSE>
RXDx
Receive shift register
Parity control
Transmit shift register
TXDx
RB8
Receive buffer (SCxBUF)
Error Flag
TB8
Transmit buffer (SCxBUF)
FIFO control
SCxCR <OERR><PERR><FERR>
FIFO control
Internal data bus
Internal data bus
Internal data bus

Figure 12-1 SIO / UART Block Diagram

12.4 Registers Description

12.4.1 Registers List in Each Channel

The below table shows registers and addresses for each register.

Channel x	Base Address
Channel0	0x400E_1000
Channel1	0x400E_1100
Channel2	0x400E_1200
Channel3	0x400E_1300

Register name (x=0 to 3)		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

12.4.2 SCxEN (Enable Register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	SIOE	R/W	SIO/UART operation 0: disable 1: Operation Specifies the SIO/UART operation. To use the SIO/UART, set <SIOE> = "1". When the operation is disabled, no clock is supplied to the other registers in the SIO/UART module. This can reduce the power consumption. If the SIO/UART operation is executed and then disabled, the settings will be maintained in each register except for SCxTFC<TIL>.

Note: **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.**

12.4.3 SCxBUF (Buffer Register)

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

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

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

12.4.4 SCxCR (Control Register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	RB8	R	Receive data bit 8 (for UART mode) 9th bit of the received data in the 9 bits UART mode.
6	EVEN	R/W	Parity (for UART mode) 0: Odd 1: Even Selects even or odd parity. "0" :odd parity, "1" : even parity The parity bit can be used only in the 7-bit or 8-bit UART mode.
5	PE	R/W	Adding parity (for UART mode) 0: Disabled 1: Enabled Controls enabling / disabling parity The parity bit can be used only in the 7-bit or 8-bit UART mode.
4	OERR	R	Overrun error flag (Note) 0: Normal operation 1: Error
3	PERR	R	Parity / Underrun error flag (Note) 0: Normal operation 1: Error
2	FERR	R	Framing error flag (Note) 0: Normal operation 1: Error
1	SCLKS	R/W	Selecting input clock edge (for I/O Interface mode) Set to "0" in the clock output mode. 0:Data in the transmit buffer is sent to TXDx pin one bit at a time on the falling edge of SCLKx. Data from RXDx pin is received in the receive buffer one bit at a time on the rising edge of SCLKx. In this case, the SCLKx starts from high level. 1:Data in the transmit buffer is sent to TXDx pin one bit at a time on the rising edge of SCLKx. Data from RXDx pin is received in the receive buffer one bit at a time on the falling edge of SCLKx. In this case, the SCLKx starts from low level.
0	IOC	R/W	Selecting clock (for I/O Interface) 0: Baud rate generator 1: SCLK pin input

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

12.4.5 SCxMOD0 (Mode Control Register 0)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TB8	R/W	Transmit data bit 8 (For UART mode) Writes the 9th bit of transmit data in the 9 bits UART mode.
6	CTSE	R/W	Handshake function control (For 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 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 set to be enabled, Interrupt occurs only when <RB8> = "1" at 9-bit in the UART mode.
3-2	SM[1:0]	R/W	Specifies transfer mode. 00: I/O interface mode 01: 7-bit length UART mode 10: 8-bit length UART mode 11: 9-bit length UART mode
1-0	SC[1:0]	R/W	Serial transfer clock (For UART mode) 00: Timer TBxOUT Refer to Table 12-1. 01: Baud rate generator 10: Internal clock fsys 11: External clock (SCLK input) (As for the I/O interface mode, the serial transfer clock can be set in the control register (SCxCR).

Note 1: **Specify all register first and then enable the <RXE> bit.**

Note 2: **Do not clear SCxMOD0<RXE> when data is being received.**

12.4.6 SCxMOD1 (Mode Control Register 1)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	I2SC	R/W	IDLE 0: Stop 1: Operate Specifies the IDLE mode operation.
6-5	FDPX[1:0]	R/W	Transfer mode setting 00: Transfer prohibited 01: Half duplex (Receive) 10: Half duplex (Transmit) 11: Full duplex Configures the transfer mode in the I/O interface mode. Also configures the FIFO if it is enabled. In the UART mode, it is used only to specify the FIFO configuration.
4	TXE	R/W	Transmit control (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: 1SCLK 010: 2SCLK 011: 4SCLK 100: 8SCLK 101: 16SCLK 110: 32SCLK 111: 64SCLK This parameter is valid only for the I/O interface mode when SCLK pin output is selected. In other modes, this function has no meaning. Specifies the interval time of continuous transmission when double buffering or FIFO is enabled in the I/O interface mode.
0	−	R/W	Write to "0".

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

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

Note 3: 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.

12.4.7 SCxMOD2 (Mode Control Register 2)

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

<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".
Writing data again to the double buffers sets this bit 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" while reading this bit changes it 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><TXRUN></th><th><TBEMP></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 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 (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> setting.</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 I/O interface mode.
In the UART mode, set this bit to LSB first.</td></tr>
<tr><td>2</td><td>WBUF</td><td>R/W</td><td>Double buffer
0: Disabled
1: Enabled
This parameter enables or disables the transmit / receive double buffers to transmit (in both SCLK output / input modes) and receive (in SCLK output mode) data in the I/O interface mode and to transmit data in the UART mode.
When receiving data in the I/O interface mode (SCLK input) and UART mode, double buffering is enabled in both case that "0" or "1" is set to <WBUF> bit.</td></tr>
</table>

Bit	Bit Symbol	Type	Function
1-0	SWRST[1:0]	R/W	Software reset Overwriting "01" in place of "10" generates a software reset. When this software reset is executed, the following bits are initialized and the transmit/receive circuit and the FIFO become initial state (see Note1 and Note2).

Register	Bit
SCxMOD0	<RXE>
SCxMOD1	<TXE>
SCxMOD2	<TBEMP>, <RBFLL>, <TXRUN>
SCxCR	<OERR>, <PERR>, <FERR>

Note 1: **While data transmission is in progress, any software reset operation must be executed twice in succession.**

Note 2: **A software reset requires 2 clocks-duration at the time between the end of recognition and the start of execution of software reset instruction.**

12.4.8 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 to "0".
6	BRADDE	R/W	N + (16 − K)/16 divider function (For UART mode) 0: Disabled 1: Enabled This division function can only be used in the UART mode.
5-4	BRCK[1:0]	R/W	Select input clock to the baud rate generator. 00: φT1 01: φT4 10: φT16 11: φT64
3-0	BRS[3:0]	R/W	Division ratio "N" (note1)(note2) 0000: 16 0001: 1 0010: 2 : 1111: 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.**

12.4.9 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 "N + (16 − K)/16" division (For UART mode) 0000: Prohibited 0001: K = 1 0010: K = 2 : 1111: K = 15

Table 12-2 lists the setting of baud rate generator division ratio.

Table 12-2 Setting division ratio

	<BRADDE> = "0"	<BRADDE> = "1" (Note1) (Only UART mode)
<BRS>	Specify "N" (Note2) (Note3)	
<BRK>	No setting required	Specify "K" (Note4)
Division ratio	Divide by N	$N + \frac{(16 - K)}{16}$ division

Note 1: **To use the "N + (16 - K)/16" division function, be sure to set <BRADDE> to "1" after setting the K value to <BRK>. The "N + (16 - K)/16" division function can only be used in the UART mode.**

Note 2: **Specifying "K = 0" is prohibited.**

12.4.10 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
When Receive FIFO is enabled, 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 SCxRFC <RIL[1:0]></td></tr>
<tr><td>3</td><td>TFIE</td><td>R/W</td><td>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>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 SCxMOD0<RXE> / SCxMOD1<TXE>
0: None
1: Auto disabled
Controls automatic disabling of transmission and reception.
Setting "1" enables to operate as follows
<table>
<tr><td>Half duplex Receive</td><td>When receive shift register, the receive buffer and the Receive FIFO are filled, SCxMOD0<RXE> is automatically set to "0" to inhibit further reception.</td></tr>
<tr><td>Half duplex Transmit</td><td>When the Transmit FIFO, the transmit buffer and the transmit shift register is 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/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>Enables FIFO
0: Disabled
1: Enabled
Enabled bit for FIFO. (note2)
If <CNFG> is set to "1", the SCxMOD1 <FDPX[1:0]> setting automatically configures FIFO as follows:
<table>
<tr><td>Half duplex Receive</td><td>Receive FIFO 4 bytes</td></tr>
<tr><td>Half duplex Transmit</td><td>Transmit FIFO 4 bytes</td></tr>
<tr><td>Full duplex</td><td>Receive FIFO 2 bytes + Transmit FIFO 2 bytes</td></tr>
</table></td></tr>
</table>

Note 1: **Regarding Transmit FIFO, the maximum number of bytes (Refer to <CNFG>) being configured is always available. The available number of bytes is the bytes already written to the Transmit FIFO.**

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

12.4.11 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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	RFCS	W	Receive FIFO clear (Note1) 1: Clear When SCxRFC<RFCS> is set to "1", the receive FIFO is cleared and SCxRST<RLVL> is "000". And also the read pointer is initialized.
6	RFIS	R/W	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]>
5-2	−	R	Read as "0".
1-0	RIL[1:0]	R/W	Receive FIFO fill level to generate receive interrupts

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

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

12.4.12 SCxTFC (Transmit FIFO Configuration Register) (Note2)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TFCS	W	Transmit FIFO clear (Note1) 1: Clear When SCxTST<TFCS> is set to "1", the transmit FIFO is cleared and SCxTST<TLVL> is "000". And also the write pointer is initialized.
6	TFIS	R/W	Select 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]>
5-2	−	R	Read as "0".
1-0	TIL[1:0]	R/W	Transmit FIFO fill level which transmit interrupt is occurred.

	Half duplex	Full duplex
00	Empty	Empty
01	1 byte	1 byte
10	2 bytes	Empty
11	3 bytes	1 byte

Note 1: **To use Transmit/Receive FIFO buffer, Transmit/Receive FIFO must be cleared after setting the SIO/UART 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.

12.4.13 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 register.**

12.4.14 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 register.**

12.5 Operation in Each Mode

Table 12-3 shows the modes and data format.

Table 12-3 Mode and Data format

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

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

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

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

Parity bits can be added in the mode 1 and mode 2. The mode 3 has a wake-up 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.

12.6 Data Format

12.6.1 Data Format List

Figure 12-2 shows data format.

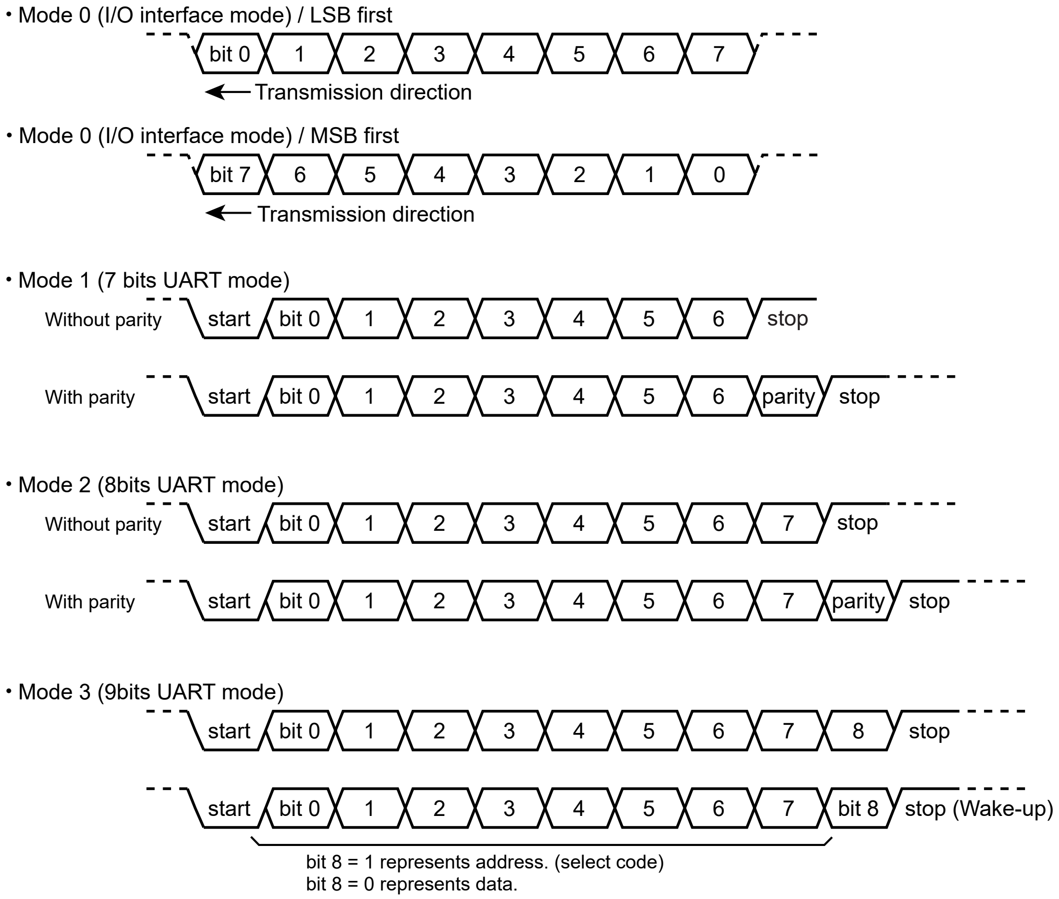

• Mode 0 (I/O interface mode) / LSB first
bit 0
1
2
3
4
5
6
7
Transmission direction
• Mode 0 (I/O interface mode) / MSB first
bit 7
6
5
4
3
2
1
0
Transmission direction
• Mode 1 (7 bits UART mode)
Without parity
start
bit 0
1
2
3
4
5
6
stop
With parity
start
bit 0
1
2
3
4
5
6
parity
stop
• Mode 2 (8bits 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
• Mode 3 (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 12-2 Data Format

12.6.2 Parity Control

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

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

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

12.6.2.1 Transmission

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

After data transmission is complete, the parity bit will be stored in SCxBUF<TB7> in the 7-bit UART mode SCxMOD0<TB8> in the 8-bit UART mode.

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

12.6.2.2 Receiving Data

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

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

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

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

12.6.3 STOP Bit Length

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

12.7 Clock Control

12.7.1 Prescaler

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

Use the CGSYSCR register in the clock / mode control block to select the input clock φT0 of the prescaler.

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

The below tables show the resolution of the input clock to the baud rate generator.

Table 12-4 Clock resolution to the Baud Rate Generator fc = 80 MHz

Peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
0 (fgear)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.03 µs)	fc/2^{3} (0.1 µs)	fc/2^{5} (0.4 µs)	fc/2^{7} (1.6 µs)
		001 (fperiph/2)	fc/2^{2} (0.05 µs)	fc/2^{4} (0.2 µs)	fc/2^{6} (0.8 µs)	fc/2^{8} (3.2 µs)
		010 (fperiph/4)	fc/2^{3} (0.1 µs)	fc/2^{5} (0.4 µs)	fc/2^{7} (1.6 µs)	fc/2^{9} (6.4 µs)
		011 (fperiph/8)	fc/2^{4} (0.2 µs)	fc/2^{6} (0.8 µs)	fc/2^{8} (3.2 µs)	fc/2^{10} (12.8 µs)
		100 (fperiph/16)	fc/2^{5} (0.4 µs)	fc/2^{7} (1.6 µs)	fc/2^{9} (6.4 µs)	fc/2^{11} (25.6 µs)
		101 (fperiph/32)	fc/2^{6} (0.8 µs)	fc/2^{8} (3.2 µs)	fc/2^{10} (12.8 µs)	fc/2^{12} (51.2 µs)
	100 (fc/2)	000 (fperiph/1)	fc/2^{2} (0.05 µs)	fc/2^{4} (0.2 µs)	fc/2^{6} (0.8 µs)	fc/2^{8} (3.2 µs)
		001 (fperiph/2)	fc/2^{3} (0.1 µs)	fc/2^{5} (0.4 µs)	fc/2^{7} (1.6 µs)	fc/2^{9} (6.4 µs)
		010 (fperiph/4)	fc/2^{4} (0.2 µs)	fc/2^{6} (0.8 µs)	fc/2^{8} (3.2 µs)	fc/2^{10} (12.8 µs)
		011 (fperiph/8)	fc/2^{5} (0.4 µs)	fc/2^{7} (1.6 µs)	fc/2^{9} (6.4 µs)	fc/2^{11} (25.6 µs)
		100 (fperiph/16)	fc/2^{6} (0.8 µs)	fc/2^{8} (3.2 µs)	fc/2^{10} (12.8 µs)	fc/2^{12} (51.2 µs)
		101 (fperiph/32)	fc/2^{7} (1.6 µs)	fc/2^{9} (6.4 µs)	fc/2^{11} (25.6 µs)	fc/2^{13} (102 µs)
	101 (fc/4)	000 (fperiph/1)	fc/2^{3} (0.1 µs)	fc/2^{5} (0.4 µs)	fc/2^{7} (1.6 µs)	fc/2^{9} (6.4 µs)
		001 (fperiph/2)	fc/2^{4} (0.2 µs)	fc/2^{6} (0.8 µs)	fc/2^{8} (3.2 µs)	fc/2^{10} (12.8 µs)
		010 (fperiph/4)	fc/2^{5} (0.4 µs)	fc/2^{7} (1.6 µs)	fc/2^{9} (6.4 µs)	fc/2^{11} (25.6 µs)
		011 (fperiph/8)	fc/2^{6} (0.8 µs)	fc/2^{8} (3.2 µs)	fc/2^{10} (12.8 µs)	fc/2^{12} (51.2 µs)
		100 (fperiph/16)	fc/2^{7} (1.6 µs)	fc/2^{9} (6.4 µs)	fc/2^{11} (25.6 µs)	fc/2^{13} (102 µs)
		101 (fperiph/32)	fc/2^{8} (3.2 µs)	fc/2^{10} (12.8 µs)	fc/2^{12} (51.2µs)	fc/2^{14} (205 µs)
	110 (fc/8)	000 (fperiph/1)	fc/2^{4} (0.2 µs)	fc/2^{6} (0.8 µs)	fc/2^{8} (3.2 µs)	fc/2^{10} (12.8 µs)
		001 (fperiph/2)	fc/2^{5} (0.4 µs)	fc/2^{7} (1.6 µs)	fc/2^{9} (6.4 µs)	fc/2^{11} (25.6 µs)
		010 (fperiph/4)	fc/2^{6} (0.8 µs)	fc/2^{8} (3.2 µs)	fc/2^{10} (12.8 µs)	fc/2^{12} (51.2 µs)
		011 (fperiph/8)	fc/2^{7} (1.6 µs)	fc/2^{9} (6.4 µs)	fc/2^{11} (25.6 µs)	fc/2^{13} (102 µs)
		100 (fperiph/16)	fc/2^{8} (3.2 µs)	fc/2^{10} (12.8 µs)	fc/2^{12} (51.2µs)	fc/2^{14} (205 µs)
		101 (fperiph/32)	fc/2^{9} (6.4µs)	fc/2^{11} (25.6 µs)	fc/2^{13} (102 µs)	fc/2^{15} (410 µs)
	111(fc/16)	000 (fperiph/1)	fc/2^{5} (0.4 µs)	fc/2^{7} (1.6 µs)	fc/2^{9} (6.4 µs)	fc/2^{11} (25.6 µs)
		001 (fperiph/2)	fc/2^{6} (0.8 µs)	fc/2^{8} (3.2 µs)	fc/2^{10} (12.8 µs)	fc/2^{12} (51.2 µs)
		010 (fperiph/4)	fc/2^{7} (1.6 µs)	fc/2^{9} (6.4 µs)	fc/2^{11} (25.6 µs)	fc/2^{13} (102 µs)
		011 (fperiph/8)	fc/2^{8} (3.2 µs)	fc/2^{10} (12.8 µs)	fc/2^{12} (51.2µs)	fc/2^{14} (205 µs)
		100 (fperiph/16)	fc/2^{9} (6.4µs)	fc/2^{11} (25.6 µs)	fc/2^{13} (102 µs)	fc/2^{15} (410 µs)
		101 (fperiph/32)	fc/2^{10} (12.8µs)	fc/2^{12} (51.2 µs)	fc/2^{14} (204 µs)	fc/2^{16} (820 µs)

Table 12-4 Clock resolution to the Baud Rate Generator fc = 80 MHz

Peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
1 (fc)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.03 μs)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		001 (fperiph/2)	fc/2^{2} (0.05 μs)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		010 (fperiph/4)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		011 (fperiph/8)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
		100 (fperiph/16)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)
		101 (fperiph/32)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
	100 (fc/2)	000 (fperiph/1)	–	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		001 (fperiph/2)	fc/2^{2} (0.05 μs)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		010 (fperiph/4)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		011 (fperiph/8)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
		100 (fperiph/16)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)
		101 (fperiph/32)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
	101 (fc/4)	000 (fperiph/1)	–	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		001 (fperiph/2)	–	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		010 (fperiph/4)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		011 (fperiph/8)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
		100 (fperiph/16)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)
		101 (fperiph/32)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
	110 (fc/8)	000 (fperiph/1)	–	–	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		001 (fperiph/2)	–	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		010 (fperiph/4)	–	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		011 (fperiph/8)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
		100 (fperiph/16)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)
		101 (fperiph/32)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
	111 (fc/16)	000 (fperiph/1)	–	–	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		001 (fperiph/2)	–	–	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		010 (fperiph/4)	–	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		011 (fperiph/8)	–	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
		100 (fperiph/16)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)
		101 (fperiph/32)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)

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

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

Note 3: **The "−" indicates that the setting is prohibited in the above table.**

Table 12-5 Clock resolution to the Baud Rate Generator fc = 48 MHz

Peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
0 (fgear)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.0417 μs)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)
		001 (fperiph/2)	fc/2^{2} (0.0833 μs)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)
		010 (fperiph/4)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		011 (fperiph/8)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		100 (fperiph/16)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^{2} (0.0833 μs)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)
		001 (fperiph/2)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		010 (fperiph/4)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		011 (fperiph/8)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		100 (fperiph/16)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
		101 (fperiph/32)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)	fc/2^{13} (171 μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		001 (fperiph/2)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		010 (fperiph/4)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		011 (fperiph/8)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
		100 (fperiph/16)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)	fc/2^{13} (171 μs)
		101 (fperiph/32)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3μs)	fc/2^{14} (341 μs)
	110 (fc/8)	000 (fperiph/1)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		001 (fperiph/2)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		010 (fperiph/4)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
		011 (fperiph/8)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)	fc/2^{13} (171 μs)
		100 (fperiph/16)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3μs)	fc/2^{14} (341 μs)
		101 (fperiph/32)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)	fc/2^{13} (171 μs)	fc/2^{15} (683 μs)
	111 (fc/16)	000 (fperiph/1)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		001 (fperiph/2)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
		010 (fperiph/4)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)	fc/2^{13} (171 μs)
		011 (fperiph/8)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3μs)	fc/2^{14} (341 μs)
		100 (fperiph/16)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)	fc/2^{13} (171 μs)	fc/2^{15} (683 μs)
		101 (fperiph/32)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)	fc/2^{14} (341 μs)	fc/2^{16} (1365 μs)

Table 12-5 Clock resolution to the Baud Rate Generator fc = 48 MHz

Peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
1 (fc)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.0417 μs)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)
		001 (fperiph/2)	fc/2^{2} (0.0833 μs)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)
		010 (fperiph/4)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		011 (fperiph/8)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		100 (fperiph/16)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
	100 (fc/2)	000 (fperiph/1)	−	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)
		001 (fperiph/2)	fc/2^{2} (0.0833 μs)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)
		010 (fperiph/4)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		011 (fperiph/8)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		100 (fperiph/16)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
	101 (fc/4)	000 (fperiph/1)	−	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)
		001 (fperiph/2)	−	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)
		010 (fperiph/4)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		011 (fperiph/8)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		100 (fperiph/16)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
	110 (fc/8)	000 (fperiph/1)	−	−	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)
		001 (fperiph/2)	−	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)
		010 (fperiph/4)	−	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		011 (fperiph/8)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		100 (fperiph/16)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
	111 (fc/16)	000 (fperiph/1)	−	−	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)
		001 (fperiph/2)	−	−	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)
		010 (fperiph/4)	−	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		011 (fperiph/8)	−	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		100 (fperiph/16)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)

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

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

Note 3: **The "−" indicates that the setting is prohibited in the above table.**

12.7.2 Serial Clock Generation Circuit

The serial clock circuit is a block to generate transmit and receive clocks (SIOCLK) and consists of the circuits in which clocks can be selected by the settings of the baud rates generator and modes.

12.7.2.1 Baud Rate Generator

The baud rate generator generates transmit and receive clocks to determine the serial channel transfer rate.

(1) Buad Rate Generator input clock

The input clock of the baud rate generator is selected from the prescaler outputs divided by 2, 8, 32 and 128.

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

(2) Baud Rate Generator output clock

The frequency division ratio of the output clock in the baud rate generator is set by SCxBRCR and SCxBRADD.

The following frequency divide ratios can be used; 1/N frequency division in the I/O interface mode, either 1/N or N + (16-K)/16 in the UART mode.

The table below shows the frequency division ratio which can be selected.

Mode	Divide Function Setting SCxBRCR<BRADDE>	Divide by N SCxBRCR<BRS>	Divide by K SCxBRADD<BRK>
I/O interface mode	Divide by N	1 to 16 (Note)	-
UART mode	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.**

12.7.2.2 Clock Selection Circuit

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

Modes can be specified by setting the SCxMOD0<SM>.

The input clock in I/O interface mode is selected by setting SCxCR.

The clock in UART mode is selected by setting SCxMOD0<SC>.

(1) Transfer Clock in I/O interface mode

Table 12-6 shows clock selection in I/O interface mode.

Table 12-6 Clock selection in I/O interface Mode

Mode SCxMOD0<SM>	Input / Output selection SCxCR<IOC>	Clock edge selection SCxCR<SCLKS>	Clock of use
I/O interface mode	SCLK output	Set to "0" (Fixed to the rising edge)	Divided by 2 of the baud rate generator output
	SCLK input	Rising edge	SCLK input rising edge
		Falling edge	SCLK input falling edge

To get the highest baud rate, the baud rate generator must be set as below.

Note: **When deciding clock settings, make sure that AC electrical character is satisfied.**

- Clock / mode control block settings
  - fc = 80MHz
  - fgear = 80MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
  - φT0=80MHz (CGSYSCR<PRCK[2:0:]>="000" : 1 division ratio)
- SIO/UART settings (if double buffer is used)
  - Clock (SCxBRCR<BRCK[1:0]> = "00" : φT1 selected) = 40MHz
  - Divided clock frequency (SCxBRCR<BRS[3:0]> = "0001":1 division ratio) = 40MHz

  1 division ratio can be selected if double buffer is used. In this case, baud rate is 20Mbps because 40MHz is divided by 2.
- SIO/UART settings (if double buffer is not used)
  - Clock (SCxBRCR<BRCK[1:0]> = "00":φT1 selected) = 40MHz
  - Divided clock frequency (SCxBRCR<BRS[3:0]> = "0010" : 2 division ratio) = 20MHz

  2 division ratio is the highest if double buffer is not used. In this case, baud rate is 10Mbps because 20MHz is divided by 2.

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

- If double buffer is used

  - SCLK cycle > 6/fsys

  The highest buad rate is less than 80 ÷ 6 = 13.33 Mbps.
- If double buffer is not used

  - SCLK cycle > 8/fsys

  The highest baud rate is less than 80 ÷ 8 = 10 Mbps.

(2) Transfer clock in the UART mode

Table 12-7 shows the clock selection in the UART mode. In the UART mode, selected clock is divided by 16 in the receive counter or the transmit counter before use.

Table 12-7 Clock selection in UART Mode

Mode SCxMOD0<SM>	Clock selection SCxMOD0<SC>
UART Mode	Timer output
	Baud rate generator
	fsys
	SCLK input

The examples of baud rate in each clock settings.

- If baud rate generator is used.
  - fc = 80MHz
  - fgear = 80MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
  - φT0 = 80MHz (CGSYSCR<PRCK[2:0]> = "000" : 1 division ratio)
  - Clock = φT1 = 40MHz (SCxBRCR<BRCK[1:0]> = "00" : φT1 selected)

  The highest baud rate is 2.5MHz because 40MHz is divided by 16.

  Table 12-8 shows examples of baud rate when the baud rate generator is used with the following clock settings.

  - fc = 9.8304MHz
  - fgear = 9.8304MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
  - φT0 = 4.9152MHz (CGSYSCR<PRCK[2:0]> = "001" : 2 division ratio)

Table 12-8 Example of UART Mode Baud Rate (Using the Baud Rate Generator)

fc [MHz]	Division ratio N (SCxBRCR<BRS[3:0]>)	φT1 (fc/4)	φT4 (fc/16)	φT16 (fc/64)	φT64 (fc/256)
9.830400	2	76.800	19.200	4.800	1.200
	4	38.400	9.600	2.400	0.600
	8	19.200	4.800	1.200	0.300
	16	9.600	2.400	0.600	0.150

Unit : kbps

- If the SCLK input is used

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

  - SCLK cycle > 2/fsys

  The highest baud rate must be less than 80 ÷ 2 ÷ 16 = 2.5 Mbps.

- If fsys is used

  Since the highest value of fsys is 80MHz, the highest baud rate is 80 ÷ 16 = 5Mbps.

- If timer output is used

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

  Baud rate can be obtained by using the following formula.

Baud rate calculation

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by CGSYSCR<PRCK[1:0]>}}{\underbrace{(\text{TBxRG1} \times 2)}_{\text{One clock cycle is a period that the timer flip-flop is inverted twice.}} \times \underbrace{2}_{\text{In the case the timer prescaler clock fT1(2 divition ratio) is selected}} \times 16}$$

Table 12-9 shows the examples of baud rates when the timer output is used with the following clock settings.

- fc = 32MHz / 9.8304MHz / 8MHz
- fgear = 32MHz / 9.8304MHz / 8MHz (CGSYSCR<GEAR[2:0]> = "000" :fc selected)
- φT0 = 16MHz / 4.9152MHz / 4MHz (CGSYSCR<PRCK[2:0]> = "001" :2 division)
- Timer count clock= 4MHz / 1.2287MHz / 1MHz (TBxMOD<TBCLK[1:0]> = "01" :φT1 selected)

Table 12-9 Example of UART Mode Baud Rate (Using the Timer Output)

TBxRG1 setting	fc		
	32MHz	9.8304MHz	8MHz
0x0001	250	76.8	62.5
0x0002	125	38.4	31.25
0x0003	-	25.6	-
0x0004	62.5	19.2	15.625
0x0005	50	15.36	12.5
0x0006	-	12.8	-
0x0008	31.25	9.6	-
0x000A	25	7.68	6.25
0x0010	15.625	4.8	-
0x0014	12.5	3.84	3.125

Unit : kbps

12.8 Transmit / Receive Buffer and FIFO

12.8.1 Configuration

Figure 12-3 shows the configuration of transmit buffer, receive buffer and FIFO.

Appropriate settings are required for using buffer and FIFO. The configuration may be predefined depending on the mode.

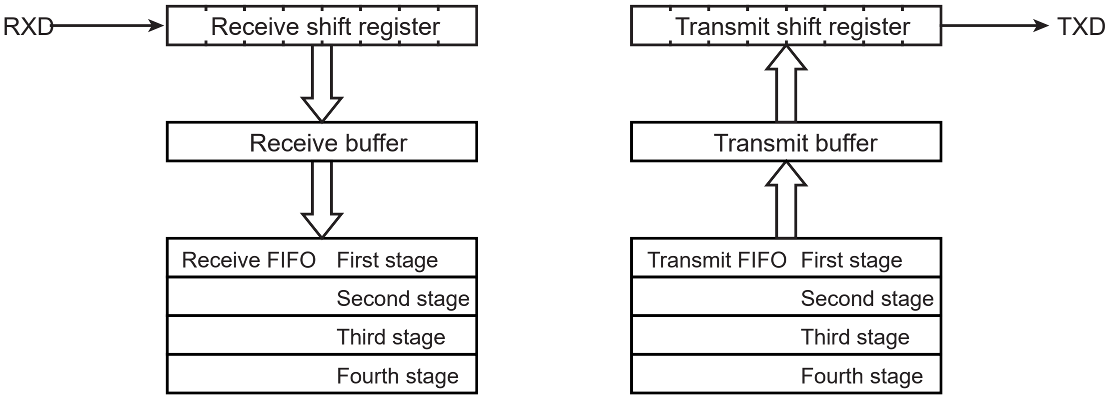

RXD
Receive shift register
Receive buffer
Receive FIFO
First stage
Second stage
Third stage
Fourth stage
Transmit shift register
TXD
Transmit buffer
Transmit FIFO
First stage
Second stage
Third stage
Fourth stage

Figure 12-3 The Configuration of Buffer and FIFO

12.8.2 Transmit / Receive Buffer

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

In the case of using a receive buffer, if SCLK input is set to generate clock output in the I/O interface mode or the UART mode is selected, it's double buffered despite the <WBUF> settings. In other modes, it's according to the <WBUF> settings.

Table 12-10 shows correlation between modes and buffers.

Table 12-10 Mode and buffer Composition

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

12.8.3 FIFO

In addition to the double buffer function above described, 4-byte FIFO can be used.

To enable FIFO, enable the double buffer by setting SCxMOD2<WBUF> to "1" and SCxFCNF<CNFG> to "1". The FIFO 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/UART transfer mode (half duplex/ full duplex) and enabling FIFO (SCxFCNF<CNFG> = "1").**

Table 12-11 shows correction between modes and FIFO.

Table 12-11 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

12.9 Status Flag

The SCxMOD2 register has two types of flag. This bit is significant only when the double buffer is enabled.

<RBFLL> is a flag to show that the receive buffer is full. When one frame of data is received and the data is moved from the receive shift register to the receive buffers, this bit changes to "1" while reading this bit changes it to "0".

<TBEMP> shows that the transmit buffers are empty. When data in the transmit buffers is moved to the transmit shift register, this bit is set to "1" When data is set to the transmit buffers, the bit is cleared to "0".

12.10 Error Flag

Three error flags are provided in the SCxCR register. The meaning of the flags is changed depending on the modes. The table below shows the meaning in each mode.

These flags are cleared to "0" after reading the SCxCR register.

Mode	Flag		
	<OERR>	<PERR>	<FERR>
UART mode	Overrun error	Parity error	Framing error
I/O interface mode (SCLK input)	Overrun error	Underrun error (When using double buffer or FIFO)	Fixed to "0"
		Fixed to "0" (When a double buffer and FIFO unused)	
I/O interface mode (SCLK output)	Undefined	Undefined	Fixed to "0"

12.10.1 OERR Flag

In both UART and I/O interface modes, this bit is set to "1" when an error is generated by completing the reception of the next frame of receive data before the receive buffer has been read. If the receive FIFO is enabled, the received data is automatically moved to the receive FIFO and no overrun error will be generated until the receive FIFO is full (or until the usable bytes are fully occupied).

In the I/O interface with SCLK output mode, the SCLK output stops upon setting the flag.

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

12.10.2 PERR Flag

This flag indicates a parity error in the UART mode and an under-run error in the I/O interface mode.

In the UART mode, <PERR> is set to "1" when the parity generated from the received data is different from the parity received.

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

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

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

Note: **To switch the I/O interface SCLK output mode to other modes, read the SCxCR register and clear the underrun flag.**

12.10.3 FERR Flag

A framing error is generated if the corresponding stop bit is determined to be "0" by sampling the bit at around the center. Regardless of the stop bit length settings in the SCxMOD2<SBLEN>register, the stop bit status is determined by only 1.

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

12.11 Receive

12.11.1 Receive Counter

The receive counter is a 4-bit binary counter and is up-counted by SIOCLK.

In the UART mode, sixteen SIOCLK clock pulses are used in receiving a single data bit and the data symbol is sampled at the seventh, eighth, and ninth pulses. From these three samples, majority logic is applied to decide the received data.

12.11.2 Receive Control Unit

12.11.2.1 I/O interface mode

In the SCLK output mode with SCxCR <IOC> set to "0", the RXD pin is sampled on the rising edge of the shift clock outputted to the SCLK pin.

In the SCLK input mode with SCxCR <IOC> set to "1", the serial receive data RXD pin is sampled on the rising or falling edge of SCLK input signal depending on the SCxCR <SCLKS> setting.

12.11.2.2 UART Mode

The receive control unit has a start bit detection circuit, which is used to initiate receive operation when a normal start bit is detected.

12.11.3 Receive Operation

12.11.3.1 Receive Buffer

The received data is stored by 1 bit in the receive shift register. When a complete set of bits has been stored, the interrupt INTRXx is generated.

When the double buffer is enabled, the data is moved to the receive buffer (SCxBUF) and the receive buffer full flag (SCxMOD2<RBFLL>) is set to "1". The receive buffer full flag is "0" cleared by reading the receive buffer. The receive buffer full flag does not have no meaning for the single buffer.

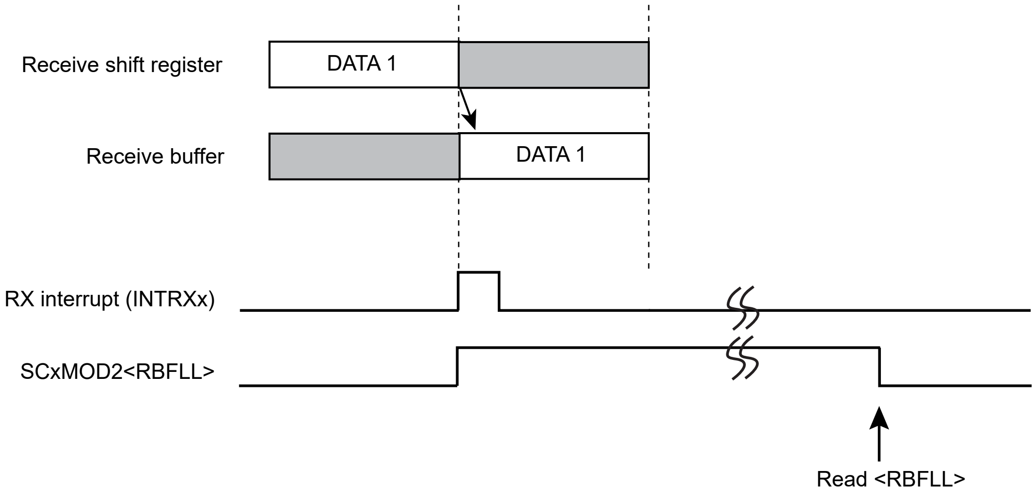


Figure 12-4 Receive Buffer Operation

12.11.3.2 Receive FIFO Operation

When FIFO is enabled, the received data is moved from receive buffer to receive FIFO and the receive buffer full flag is cleared immediately. An interrupt will be generated according to the SCxRFC<RIL> setting.

Note: **When the data with parity bit are received in UART mode by using the FIFO, the parity error flag is shown the occurring the parity error in the received data.**

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

SCxMOD1[6:5] =01	:Transfer mode is set to half duplex mode
SCxFCNF[4:0] = 10111	:Automatically inhibits continuous reception after reaching the fill level. : The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.
SCxRFC[1:0] = 00	:The fill level of FIFO in which generated receive interrupt is set to 4-byte
SCxRFC[7:6] = 11	:Clears receive FIFO and sets the condition of interrupt generation.

After setting of the above FIFO configuration, the data reception is started by writing "1" to the SCxMOD0<RXE>. When the data is stored all in the receive shift register, receive buffer and receive FIFO, SCxMOD0<RXE> is automatically cleared and the receive operations finished.

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

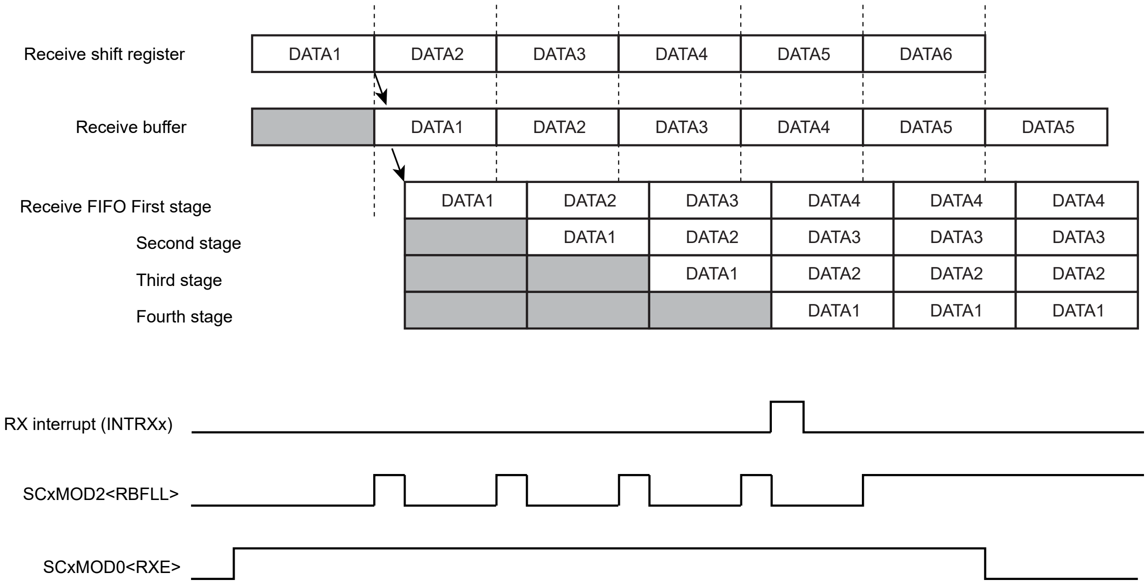

Receive shift register
DATA1
DATA2
DATA3
DATA4
DATA5
DATA6
Receive buffer
DATA1
DATA2
DATA3
DATA4
DATA5
DATA5
Receive FIFO First stage
Second stage
Third stage
Fourth stage
DATA1
DATA2
DATA3
DATA4
DATA4
DATA4
DATA1
DATA2
DATA3
DATA3
DATA3
DATA1
DATA2
DATA2
DATA2
DATA1
DATA1
DATA1
RX interrupt (INTRXx)
SCxMOD2<RBFLL>
SCxMOD0<RXE>

Figure 12-5 Receive FIFO Operation

12.11.3.3 I/O interface mode with SCLK output

In the I/O interface mode and SCLK output setting, SCLK output stops when all received data is stored in the receive buffer and FIFO. Thus, in this mode, the overrun error flag has no meaning.

The timing of SCLK output stop and re-output depends on receive buffer and FIFO.

(1) Case of single buffer

Stop SCLK output after receiving a data. In this mode, I/O interface mode can transfer each data with the transfer device by hand-shake.

When the data in a buffer is read, SCLK output restarts.

(2) Case of double buffer

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

When the data is read, SCLK output restarts.

(3) Case of FIFO

Stop SCLK output after receiving the data into a shift register, received buffer and FIFO.

When one byte data is read, the data in the received buffer is transferred into FIFO and the data in the receive shift register is transferred into the received buffer and SCLK output restarts.

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

12.11.3.4 Read Received Data

In spite of enabling or disabling FIFO, read the received data from the receive buffer (SCxBUF).

When receive FIFO is disabled, the buffer full flag SCxMOD2<RBFLL> is cleared to "0" by this reading. In the case of the next data can be received in the receive shift register before reading a data from the receive buffer. The parity bit to be added in the 8-bit UART mode as well as the most significant bit in the 9-bit UART mode will be stored in SCxCR<RB8>.

When the receive FIFO is available, the 9-bit UART mode is prohibited because up to 8-bit data can be stored in FIFO. In the 8-bit UART mode, the parity bit is lost but parity error is determined and the result is stored in SCxCR<PERR>.

12.11.3.5 Wake-up Function

In the 9-bit UART mode, the slave controller can be operated in the wake-up mode by setting the wake-up function SCxMOD0 <WU> to "1". In this case, the interrupt INTRXx will be generated only when SCxCR<RB8> is set to "1".

12.11.3.6 Overrun Error

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

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

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

12.12 Transmission

12.12.1 Transmission Counter

The transmit counter is a 4-bit binary counter and is counted by SIOCLK as in the case of the receive counter.

In UART mode, it generates a transmit clock (TXDCLK) on every 16th clock pulse.

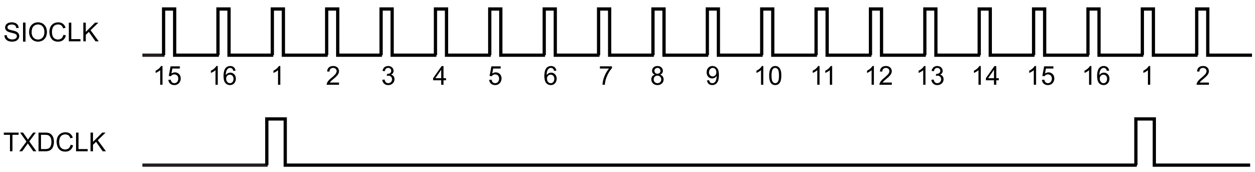


Figure 12-6 Generation of Transmission Clock in UART Mode

12.12.2 Transmission Control

12.12.2.1 I/O interface Mode

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

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

12.12.2.2 UART Mode

When the transmit data is written in the transmit buffer, data transmission is initiated on the rising edge of the next TXDCLK and the transmit shift clock signal is also generated.

12.12.3 Transmit Operation

12.12.3.1 Operation of Transmission 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".

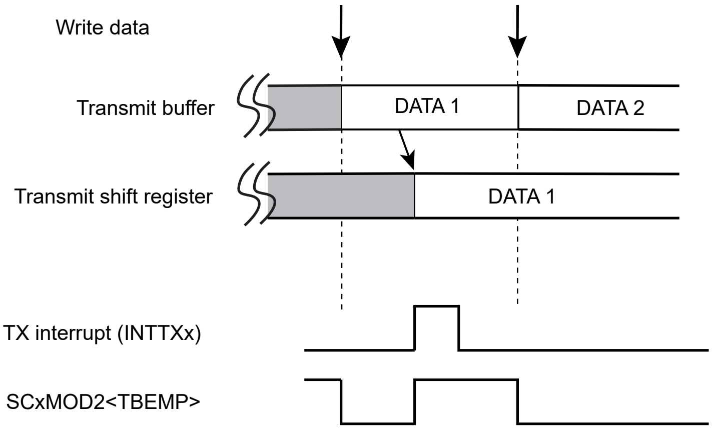


Figure 12-7 Operation of Transmission Buffer (Double buffer is enabled)

12.12.3.2 Transmit FIFO Operation

When FIFO is enabled, the maximum 5-byte data can be stored using the transmit buffer and FIFO. Once transmission is enabled, data is transferred to the transmit shift register from the transmit buffer and start transmission. If data exists in the FIFO, the data is moved to the transmit buffer immediately, and the <TBEMP> flag is cleared to "0".

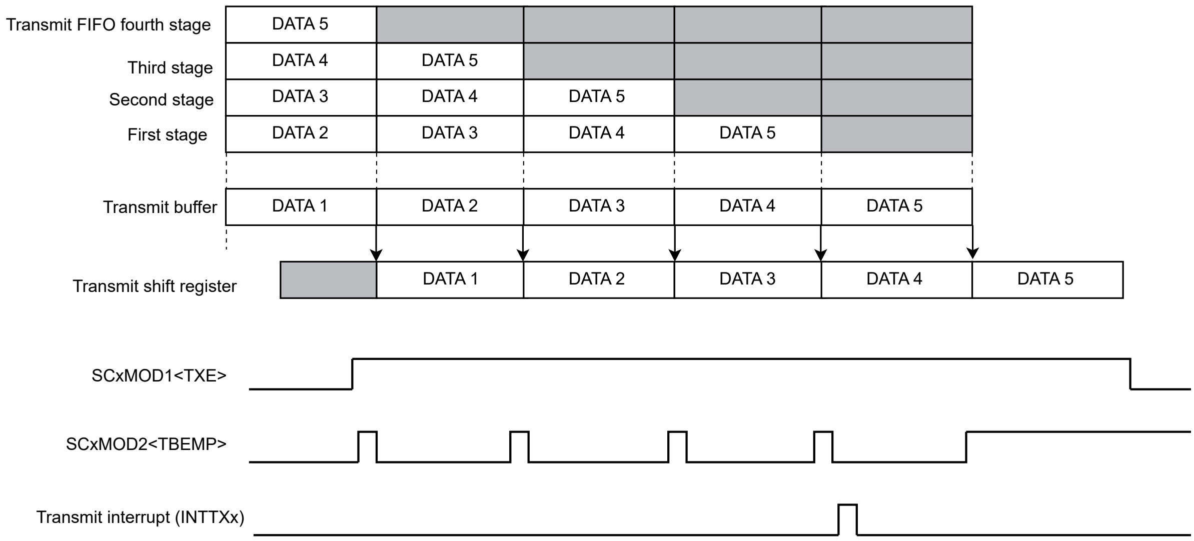
Note: **To use Transmit FIFO buffer, Transmit FIFO must be cleared after setting the SIO/UART transfer mode (half duplex/ full duplex) and enabling FIFO (SCxFCNF<CNFG>="1").**

Settings and operations to transmit 4-byte data stream by setting the transfer mode to half duplex are shown as below.

SCxMOD1[6:5] =10	:Transfer mode is set to half duplex.
SCxFCNF[4:0] = 11011	:Transmission is automatically disabled if FIFO becomes empty. :The number of bytes to be used in the receive FIFO is the same as the interrupt :generation fill level.
SCxTFC[1:0] = 00	:Sets the interrupt generation fill level to "0".
SCxTFC[7:6] = 11	:Clears receive FIFO and sets the condition of interrupt generation.
SCxFCNF[0] = 1	:Enable FIFO

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

Once above settings are configured, if the transmission is not set as auto disabled, the transmission should lasts writing transmit data.


Transmit FIFO fourth stage
Third stage
Second stage
First stage
DATA 5
DATA 4
DATA 5
DATA 3
DATA 4
DATA 5
DATA 2
DATA 3
DATA 4
DATA 5
Transmit buffer
DATA 1
DATA 2
DATA 3
DATA 4
DATA 5
Transmit shift register
DATA 1
DATA 2
DATA 3
DATA 4
DATA 5
SCxMOD1<TXE>
SCxMOD2<TBEMP>
Transmit interrupt (INTTXx)

12.12.3.3 I/O interface Mode/Transmission by SCLK Output

If SCLK is set to generate clock in the I/O interface mode, the SCLK output automatically stops when all data transmission is completed and underrun error will not occur.

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

(1) Single Buffer

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

(2) Double Buffer

The SCLK output stops upon completion of data transmission of the transmit shift register and the transmit buffer. The SCLK output resumes when the next data is written in the buffer.

(3) FIFO

The transmission of all data stored in the transmit shift register, transmit buffer and FIFO is completed, the SCLK output stops. The next data is written, SCLK output resumes.

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

12.12.3.4 Underrun Error

If the transmit FIFO is disabled in the I/O interface mode SCLK input mode and if no data is set in transmit buffer before the next frame clock input, which occurs upon completion of data transmission from transmit shift register, an under-run error occurs and SCxCR<PERR> is set to "1".

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

Note: **Before switching the I/O interface mode SCLK output mode to other modes, read the SCxCR register and clear the underrun flag.**

12.13 Handshake Function

The function of the handshake is to enable frame-by-frame data transmission by using the CTS (Clear to send) pin and to prevent overrun errors. This function can be enabled or disabled by SCxMOD0<CTSE>.

When the $\overline{\text{CTS}}$ pin is set to "High" level, the current data transmission can be completed but the next data transmission is suspended until $\overline{\text{CTS}}$ pin returns to the "Low" level. However in this case, the INTTXx interrupt is generated in the normal timing, the next transmit data is written in the transmit buffer, and it waits until it is ready to transmit data.

Note 1: **If the $\overline{\text{CTS}}$ signal is set to "High" during transmission, the next data transmission is suspended after the current transmission is completed (Point "a" in Figure 12-9).**

Note 2: **Data transmission starts on the first falling edge of the TXDCLK clock after $\overline{\text{CTS}}$ is set to "Low" (Point "b" in Figure 12-9).**

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.

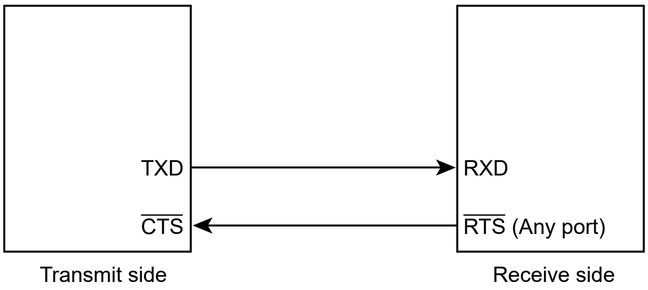

TXD
$\overline{\text{CTS}}$
RXD
$\overline{\text{RTS}}$ (Any port)
Transmit side
Receive side

Figure 12-8 Handshake Function

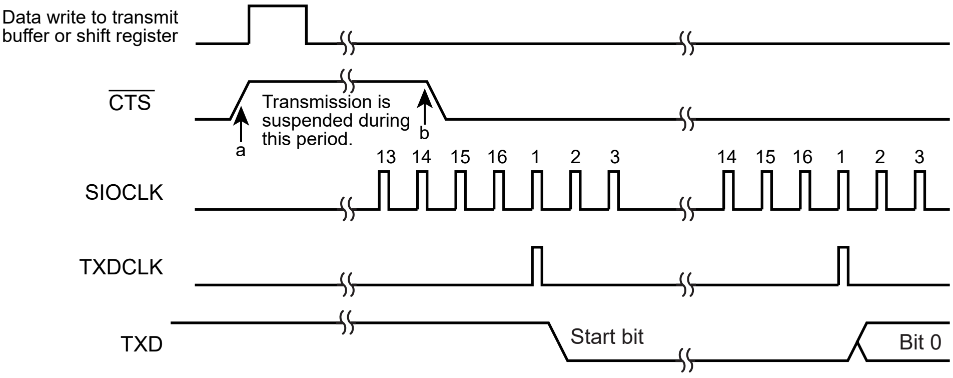

Data write to transmit buffer or shift register
$\overline{\text{CTS}}$
Transmission is suspended during this period.
a
b
13 14 15 16 1 2 3
14 15 16 1 2 3
SIOCLK
TXDCLK
TXD
Start bit
Bit 0

Figure 12-9 $\overline{\text{CTS}}$ Signal timing

12.14 Interrupt / Error Generation Timing

12.14.1 RX Interrupt

Figure 12-10 shows the data flow of receive operation and the route of read.

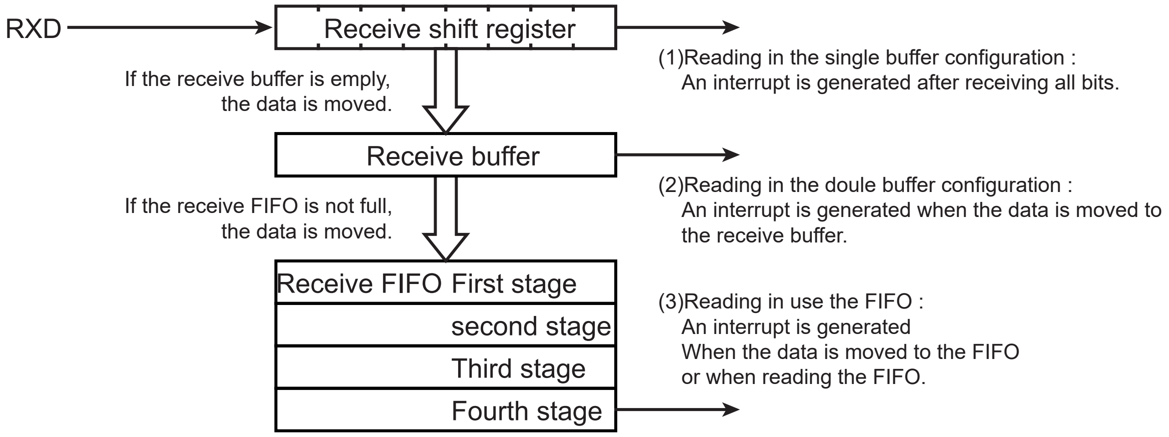

RXD
Receive shift register
If the receive buffer is emply, the data is moved.
(1)Reading in the single buffer configuration :
An interrupt is generated after receiving all bits.
Receive buffer
If the receive FIFO is not full, the data is moved.
(2)Reading in the doule buffer configuration :
An interrupt 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 12-10 Receive Buffer / FIFO Configuration Diagram

12.14.1.1 Single Buffer / Double Buffer

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

Table 12-12 Transmit interrupt condition with single buffer and double buffers

Buffer Configuration	UART modes	I/O interface modes
Single Buffer	−	Immediately after the rising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	Around the center of the first stop bit	Immediately after the rising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.) On data transfer from the shift register to the buffer by reading buffer.

Note: **Interrupts are not generated when an overrun error occurs.**

12.14.1.2 FIFO

When the FIFO is used, a receive interrupt occurs depending on the timing described in Table 12-13 and the condition specified with SCxRFC<RFIS>.

Table 12-13 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 received data is transferred from receive buffer to receive FIFO
"1"	When FIFO fill level (SCxRST<RLVL[2:0]>) ≥ Receive FIFO fill level to generate receive interrupt <RIL[1:0]>	・When received data is transferred from receive buffer to receive FIFO ・When received data is read from receive FIFO

12.14.2 Transmit interrupt

Figure 12-11 shows the data flow of transmit operation and the route of write.

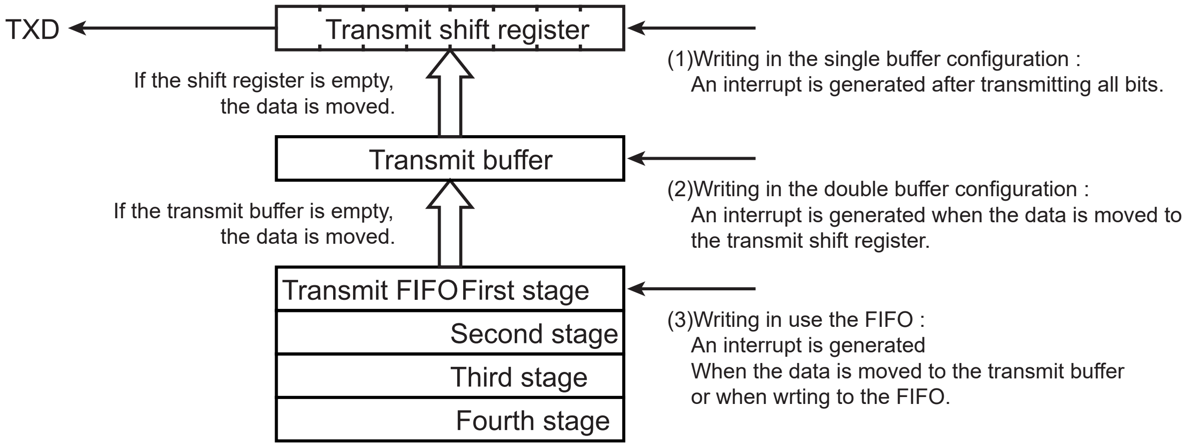

TXD
Transmit shift register
If the shift register is empty,
the data is moved.
(1)Writing in the single buffer configuration :
An interrupt is generated after transmitting all bits.
Transmit buffer
If the transmit buffer is empty,
the data is moved.
(2)Writing in the double buffer configuration :
An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

Figure 12-11 Transmit Buffer / FIFO Configuration Diagram

12.14.2.1 Single 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 12-14 Receive interrupt condition with single buffer and double buffers

Buffer Configuration	UART modes	I/O interface modes
Single Buffer	Just before the stop bit is sent	Immediately after the rising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	When a data is moved from the transmit buffet to the transmit shift register. In case of transmit shift register is empty, transmit interrupt is generated not depend on SCxMOD1<TXE> because a data written to transfer buffer is moved from transmit buffer to transmit shift register.	

12.14.2.2 FIFO

When the FIFO is used, a transmit interrupt occurs depending on the timing described in Table 12-15 and the condition specified with SCxTFC<TFIS>.

Table 12-15 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
"1"	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

12.14.3 Error Generation

12.14.3.1 UART Mode

Modes	9 bits	7 bits 8 bits 7 bits + parity 8bits + parity
Framing error Overrun error	Around the center of stop bit	
Parity Error	−	Around center of parity bit

12.14.3.2 I/O Interface Mode

Overrun error	Immediately after the rising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Underrun error	Immediately after the rising or falling edge of the next SCLK. (Rising or falling is determined according to SCxCR<SCLKS> setting.)

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

12.15 DMA Transfer

DMA transfer can be started at the timing of interrupt request.

TMPM367FDFG can be started at the half duplex mode (Transmit and Receive interrupt request) and full duplex mode (Transfer / Receive).

Note: In case using DMA transfer by transmit or receive interrupt request, enabled DMA and set transmit and receive registers after generating software reset by SCxMOD2<SWRST>.

12.15.1 Single and double buffer are used

The DMA transfer is started by generating interrupt request at the timing in 12.14.1.1 and 12.14.2.1.

12.15.2 FIFO is used

The DMA transfer is started by generating interrupt request at the timing in 12.14.1.2 and 12.14.2.2.

In the transmit mode, the fill level of interrupt generation, the minimum number of the written data into transmit FIFO before starting transmit and the maximum number of the DMA transfer data are shown as below.

Table 12-16 the fill level of interrupt generation, the minimum number of the written data into transmit FIFO before starting transmit and the maximum number of the DMA transfer data

SCxTFC <TIL[1:0]>	Half Duplex		Full duplex	
	The minimum number of the written data into transmit FIFO before starting transmit	The maximum number of the DMA transfer data	The minimum number of the written data into transmit FIFO before starting transmit	The maximum number of the DMA transfer data
00	3	4 bytes	3	2 bytes
01	4	3 bytes	4	1 byte
10	5	2 bytes	3	2 bytes
11	6	1 byte	4	1 byte

Note: In case of transmit shift register is empty, transmit interrupt is generated not depend on SCxMOD1<TXE> because a data written to transfer buffer is moved from transmit buffer to transmit shift register. The DMA transfer must be not started by this interrupt. Regarding to DMA transfer in detail, refer to DMA section.

In the receive mode, the fill level of interrupt generation and the maximum number of the DMA transfer data are shown as below.

Table 12-17 the fill level of interrupt generation, the minimum number of the written data into transmit FIFO before starting transmit and the maximum number of the DMA transfer data

SCxTFC <RIL[1:0]>	The maximum number of the DMA transfer data (Half duplex)	The maximum number of the DMA transfer data (Full duplex)
00	4 bytes	2 bytes
01	3 bytes	1 byte
10	2 bytes	2 bytes
11	1 byte	1 byte

12.16 Software Reset

Software reset is generated by writing SCxMOD2<SWRST[1:0]> as "10" followed by "01". As a result, SCxMOD0<RXE>, SCxMOD1<TXE>, SCxMOD2<TBEMP><RBFLL><TXRUN>, SCxCR<OERR><PERR><FERR> are initialized. And the receive circuit, the transmit circuit and the FIFO become initial state. Other states are held.

12.17 Operation in Each Mode

12.17.1 I/O Interface Mode

The I/O interface mode can be selected by setting the mode control register (SCxMOD<SM[1:0]>) to "00".

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

The following operational descriptions are for the case use of FIFO is disabled. For details of FIFO operation, refer to the previous sections describing receive/transmit FIFO functions.

12.17.1.1 Transmitting Data

(1) SCLK Output Mode

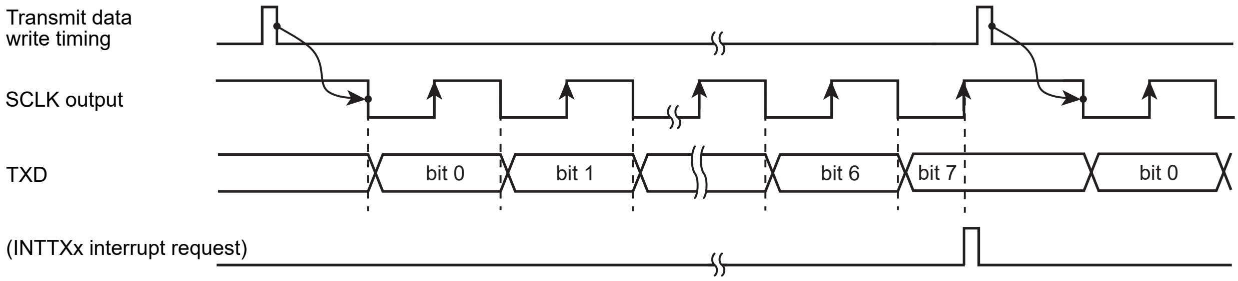
- If the transmit double buffer is disabled (SCxMOD2<WBUF> = "0")

  Data is output from the TXD pin and the clock is output from the SCLK pin each time the CPU writes data to the transmit buffer. When all data is output, an interrupt (INTTXx) is generated.

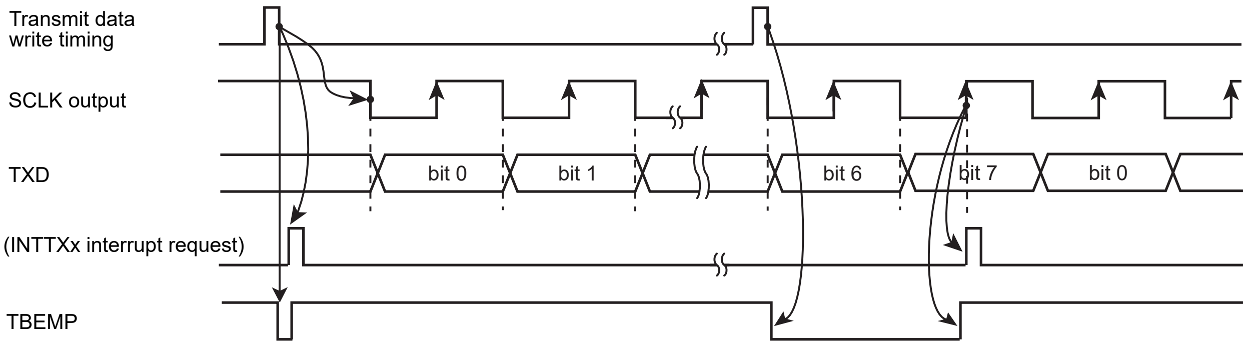
- If the transmit double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data is moved from the transmit buffer to the transmit shift register when the CPU writes data to the transmit buffer while data transmission is halted or when data transmission from the transmit buffer (shift register) is completed. Simultaneously, the transmit buffer empty flag SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

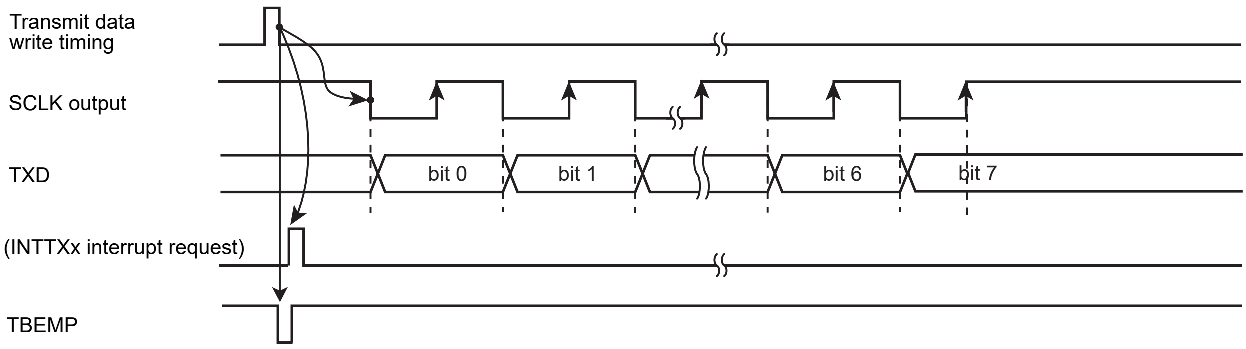
  When data is moved from the transmit buffer to the transmit shift register, if the transmit buffer has no data to be moved to the transmit shift register, INTTXx interrupt is not generated and the SCLK output stops.


Transmit data write timing
SCLK output
TXD
bit 0
bit 1
bit 6
bit 7
bit 0
(INTTXx interrupt request)

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


Transmit data write timing
SCLK output
TXD
bit 0
bit 1
bit 6
bit 7
bit 0
(INTTXx interrupt request)
TBEMP

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


Transmit data write timing
SCLK output
TXD
bit 0
bit 1
bit 6
bit 7
(INTTXx interrupt request)
TBEMP

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

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

(2) SCLK Input Mode

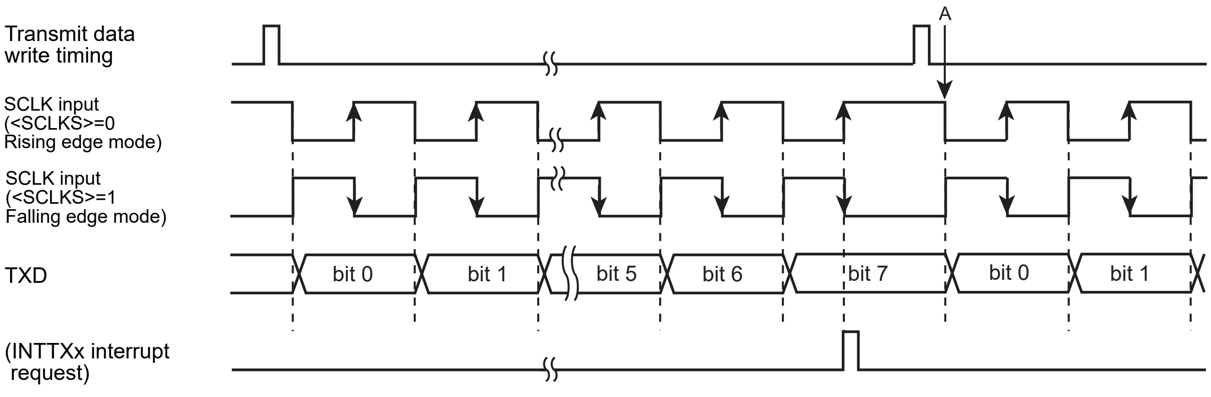
- If double buffering is disabled (SCxMOD2<WBUF> = "0")

  If the SCLK is input in the condition where data is written in the transmit buffer, 8-bit data is outputted from the TXD pin. When all data is output, an interrupt INTTXx is generated. The next transmit data must be written before the timing point "A" as shown in Figure 12-13.

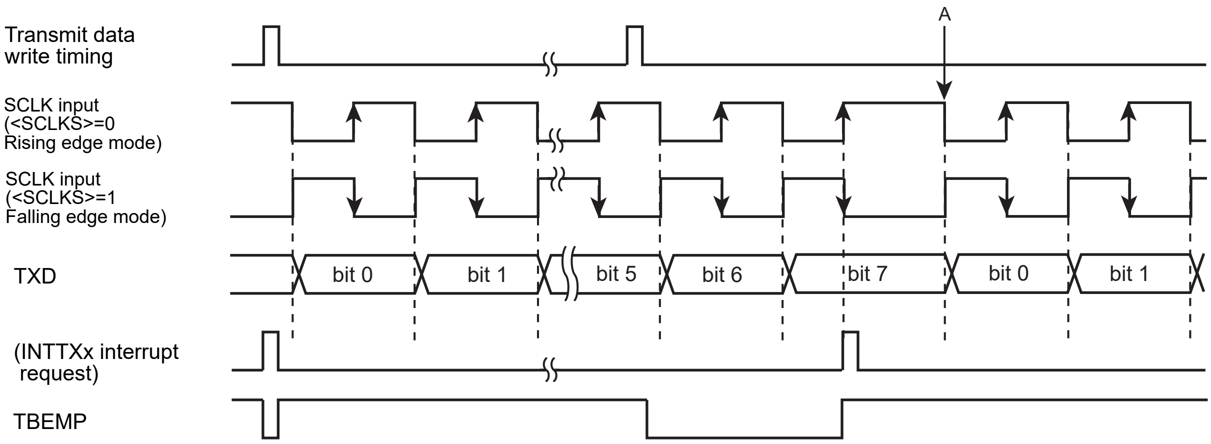
- If double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data is moved from the transmit buffer to the transmit shift register when the CPU writes data to the transmit buffer before the SCLK input becomes active or when data transmission from the transmit shift register is completed. Simultaneously, the transmit buffer empty flag SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

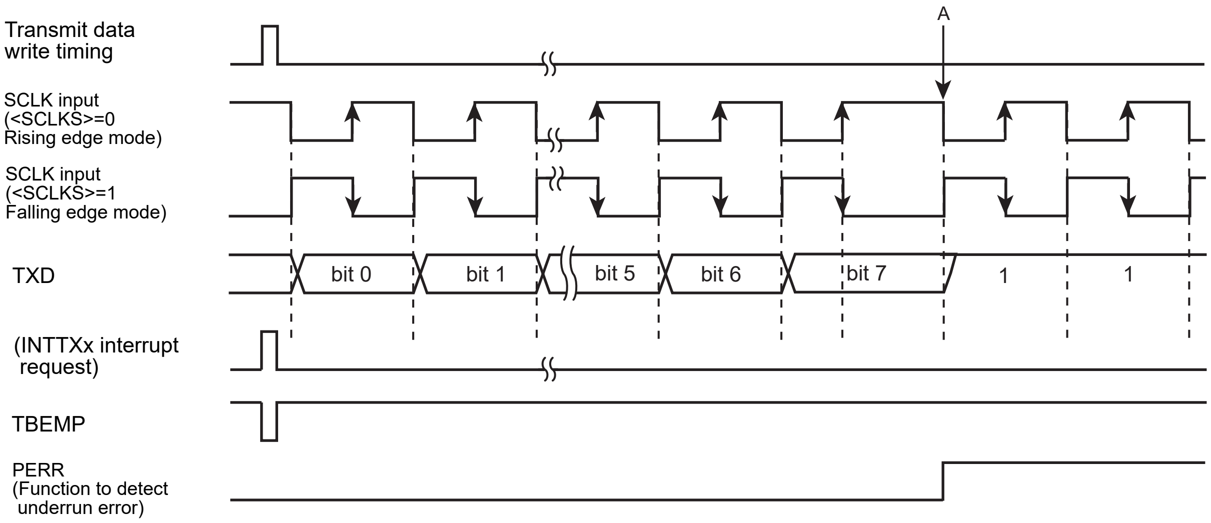
  If the SCLK input becomes active while no data is in the transmit buffer, although the internal bit counter is started, an under-run error occurs and 8-bit dummy data (0xFF) is sent.


A
Transmit data write timing
SCLK input (<SCLKS>=0 Rising edge mode)
SCLK input (<SCLKS>=1 Falling edge mode)
TXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
(INTTXx interrupt request)

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


A
Transmit data write timing
SCLK input (<SCLKS>=0 Rising edge mode)
SCLK input (<SCLKS>=1 Falling edge mode)
TXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
(INTTXx interrupt request)
TBEMP

<WBUF> = "1" (if double beffering is enabled and there is data in beffer2)


A
Transmit data write timing
SCLK input (<SCLKS>=0 Rising edge mode)
SCLK input (<SCLKS>=1 Falling edge mode)
TXD
bit 0
bit 1
bit 5
bit 6
bit 7
1
1
(INTTXx interrupt request)
TBEMP
PERR (Function to detect underrun error)

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

Figure 12-13 Transmit Operation in the I/O Interface Mode (SCLK Input Mode)

12.17.1.2 Receive

(1) SCLK Output Mode

The SCLK output can be started by setting the receive enable bit SCxMOD0<RXE> to "1".

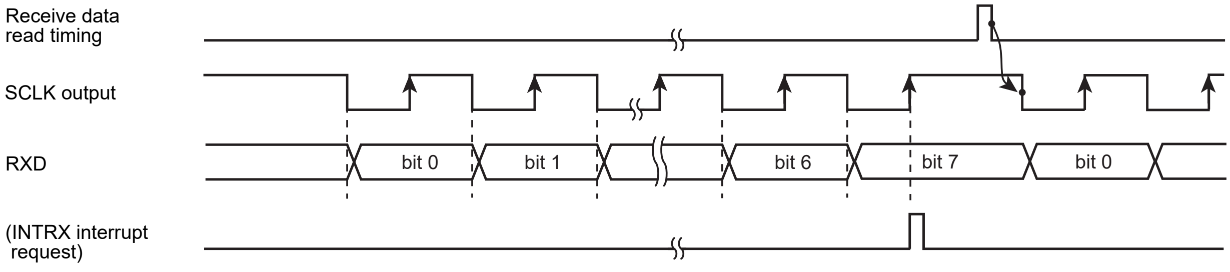
- If double buffer is disabled (SCxMOD2<WBUF> = "0")

  A clock pulse is outputted from the SCLK pin and the next data is stored into the shift register each time the CPU reads received data. When all the 8 bits are received, the INTRXx interrupt is generated.

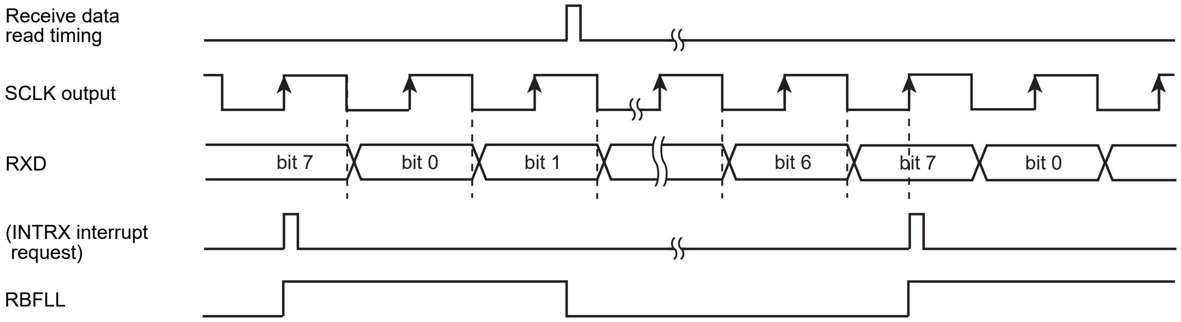
- If double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data stored in the shift register is moved to the receive buffer and the receive buffer can receive the next frame. A data is moved from the shift register to the receive buffer, the receive buffer full flag SCxMOD2<RBFLL> is set to "1" and the INTRXx is generated.

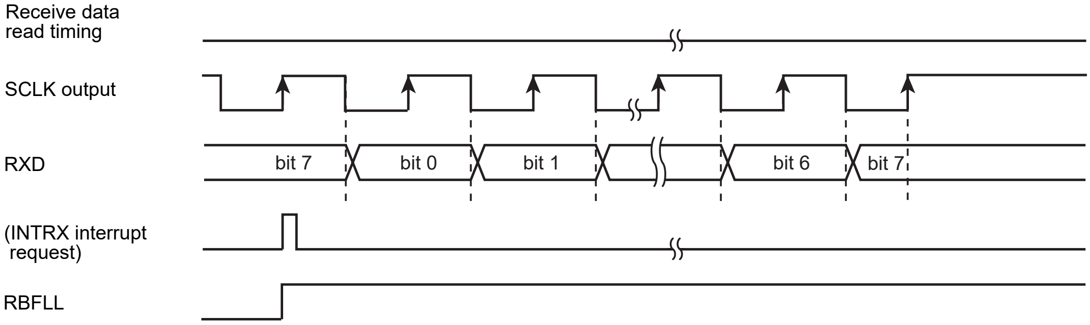
  While data is in the receive buffer, if the data cannot be read from the receive buffer before completing reception of the next 8 bits, the INTRXx interrupt is not generated and the SCLK output stops. In this state, reading data from the receive buffer allows data in the shift register to move to the receive buffer and thus the INTRXx interrupt is generated and data reception resumes.


Receive data read timing
SCLK output
RXD
bit 0
bit 1
bit 6
bit 7
bit 0
(INTRX interrupt request)

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


Receive data read timing
SCLK output
RXD
bit 7
bit 0
bit 1
bit 6
bit 7
bit 0
(INTRX interrupt request)
RBFLL

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


Receive data read timing
SCLK output
RXD
bit 7
bit 0
bit 1
bit 6
bit 7
(INTRX interrupt request)
RBFLL

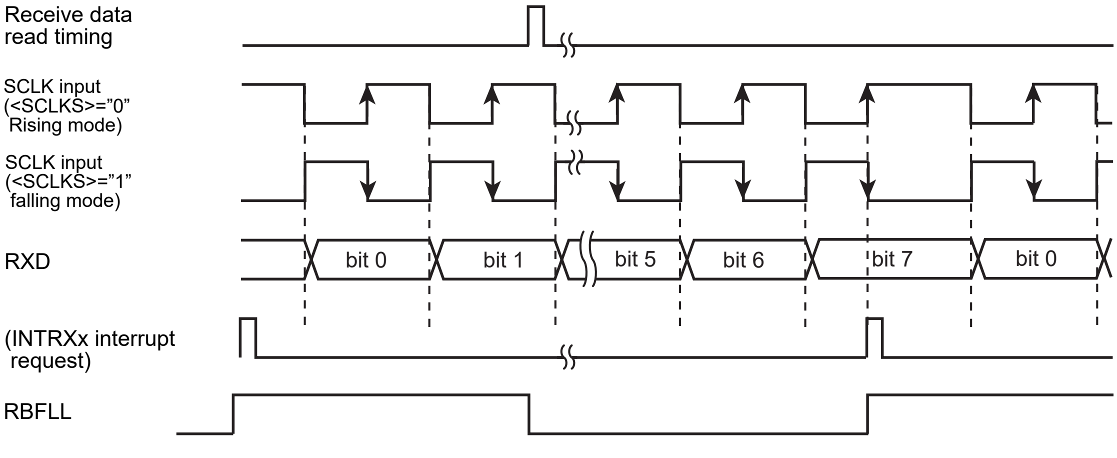
<WBUF> = "1" (if double beffering is enabled and data con not be read from beffer)

Figure 12-14 Receive Operation in the I/O Interface Mode (SCLK Output Mode)

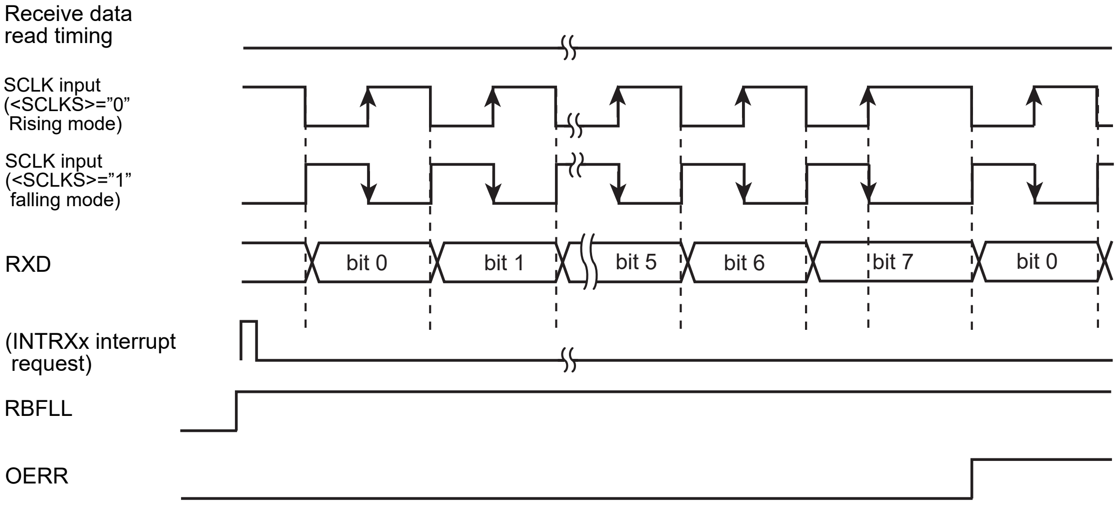
(2) SCLK Input Mode

In the SCLK input mode, receiving double buffering is always enabled, the received frame can be moved to the receive buffer from the shift register, and the receive buffer can receive the next frame successively.

The INTRXx receive interrupt is generated each time received data is moved to the receive buffer.


Receive data read timing
SCLK input (<SCLKS>="0" Rising mode)
SCLK input (<SCLKS>="1" falling mode)
RXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
(INTRXx interrupt request)
RBFLL

If data is read from buffer


Receive data read timing
SCLK input (<SCLKS>="0" Rising mode)
SCLK input (<SCLKS>="1" falling mode)
RXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
(INTRXx interrupt request)
RBFLL
OERR

If data can not be read from buffer

Figure 12-15 Receive Operation in the I/O Interface Mode (SCLK Input Mode)

12.17.1.3 Transmit and Receive (Full duplex)

(1) SCLK Output Mode

- If SCxMOD2<WBUF> is set to "0" and the double buffers are disabled

  SCLK is outputted when the CPU writes data to the transmit buffer.

  Subsequently, 8 bits of data are shifted into receive shift refister and the INTRXx receive interrupt is generated. Concurrently, 8 bits of data written to the transmit buffer are outputted from the TXD pin, the INTTXx transmit interrupt is generated when transmission of all data bits has been completed. Then, the SCLK output stops.

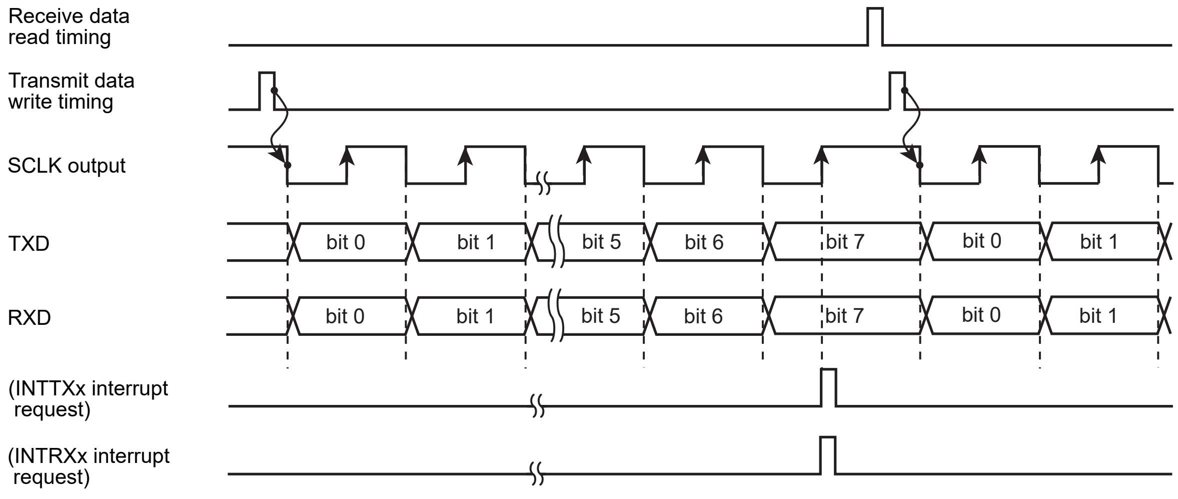
  The next round of data transmission and reception starts when the data is read from the receive buffer and the next transmit data is written to the transmit buffer by the CPU. The order of reading the receive buffer and writing to the transmit buffer can be freely determined. Data transmission is resumed only when both conditions are satisfied.

- If SCxMOD2<WBUF> is set to "1" and the double buffers are enabled

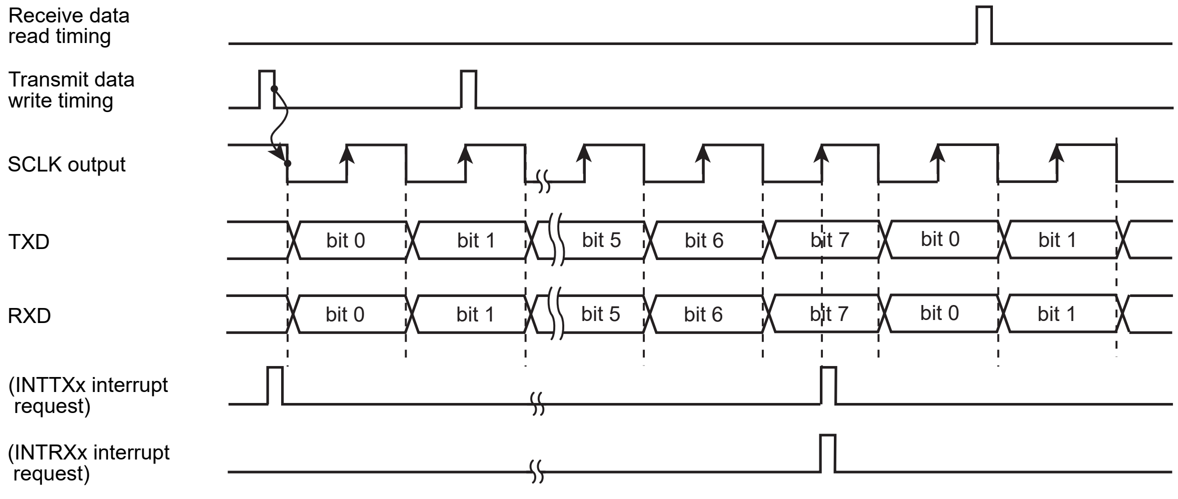
  SCLK is outputted when the CPU writes data to the transmit buffer.

  8 bits of data are shifted into the receive shift register, moved to the receive buffer, and the INTRXx interrupt is generated. While 8 bits of data is received, 8 bits of transmit data is outputted from the TXD pin. When all data bits are sent out, the INTTXx interrupt is generated and the next data is moved from the transmit buffer to the transmit shift register.

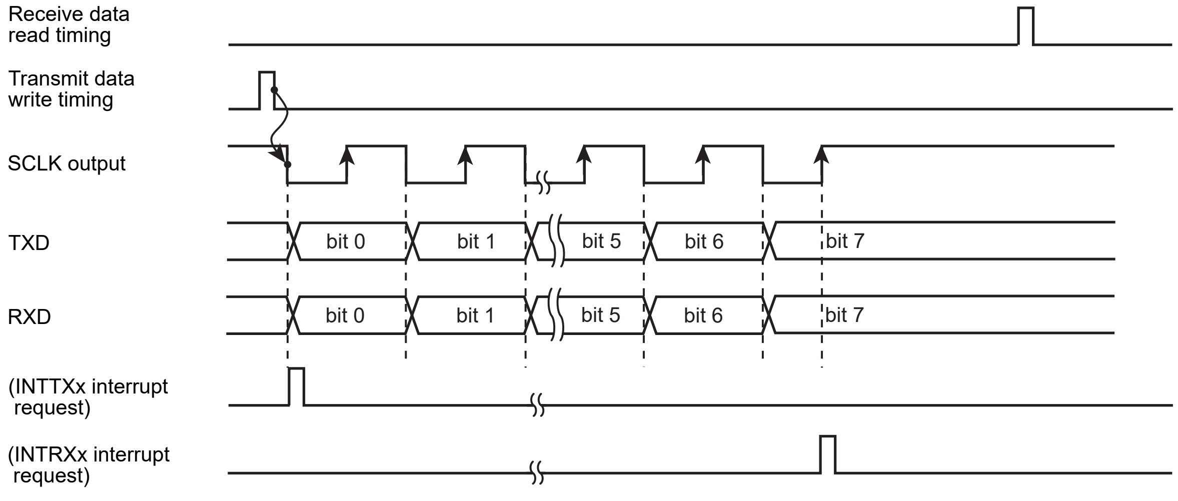
  If the transmit buffer has no data to be moved to the transmit buffer (SCxMOD2<TBEMP> = 1) or when the receive buffer is full (SCxMOD2<RBFLL> = 1), the SCLK output is stopped. When both conditions, receive data is read and transmit data is written, are satisfied, the SCLK output is resumed and the next round of data transmission and reception is started.


Receive data read timing
Transmit data write timing
SCLK output
TXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
RXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
(INTTXx interrupt request)
(INTRXx interrupt request)

<WBUF> = "0" (if double beffering is disabled)


Receive data read timing
Transmit data write timing
SCLK output
TXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
RXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
(INTTXx interrupt request)
(INTRXx interrupt request)

<WBUF> = "1" (if double buffering is enabled)


Receive data read timing
Transmit data write timing
SCLK output
TXD
bit 0
bit 1
bit 5
bit 6
bit 7
RXD
bit 0
bit 1
bit 5
bit 6
bit 7
(INTTXx interrupt request)
(INTRXx interrupt request)

<WBUF> = "1" (if double buffering is enabled)

Figure 12-16 Transmit / Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

- If SCxMOD2<WBUF> is set to "0" and the transmit double buffer is disabled

  When receiving data, double buffer is always enabled regardless of the SCxMOD2 <WBUF> settings.

  8-bit data written in the transmit buffer is outputted from the TXD pin and 8 bit of data is shifted into the receive buffer when the SCLK input becomes active.The INTTXx interrupt is generated upon completion of data transmission. The INTRXx interrupt is generated when the data is moved from receive shift register to receive buffer after completion of data reception.

  Note that transmit data must be written into the transmit buffer before the SCLK input for the next frame (data must be written before the point A in Figure 10-17). Data must be read before completing reception of the next frame data.

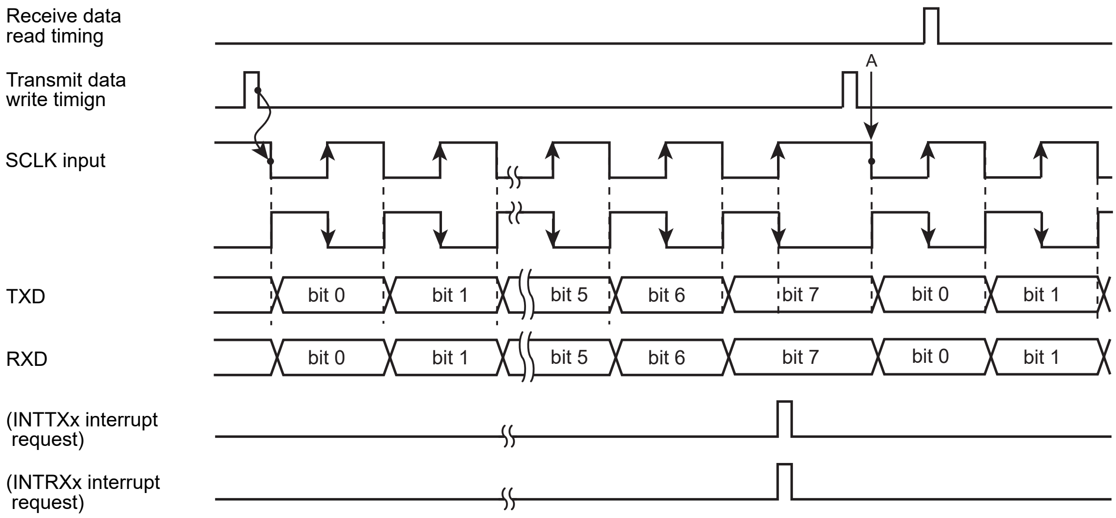
- If SCxMOD2<WBUF> is set to "1" and the double buffer is enabled.

  The interrupt INTTXx is generated at the timing when the transmit buffer data is moved to the transmit shift register after completing data transmission from the transmit shift register. At the same time, data received is shifted to the shift register, it is moved to the receive buffer, and the INTRXx interrupt is generated.

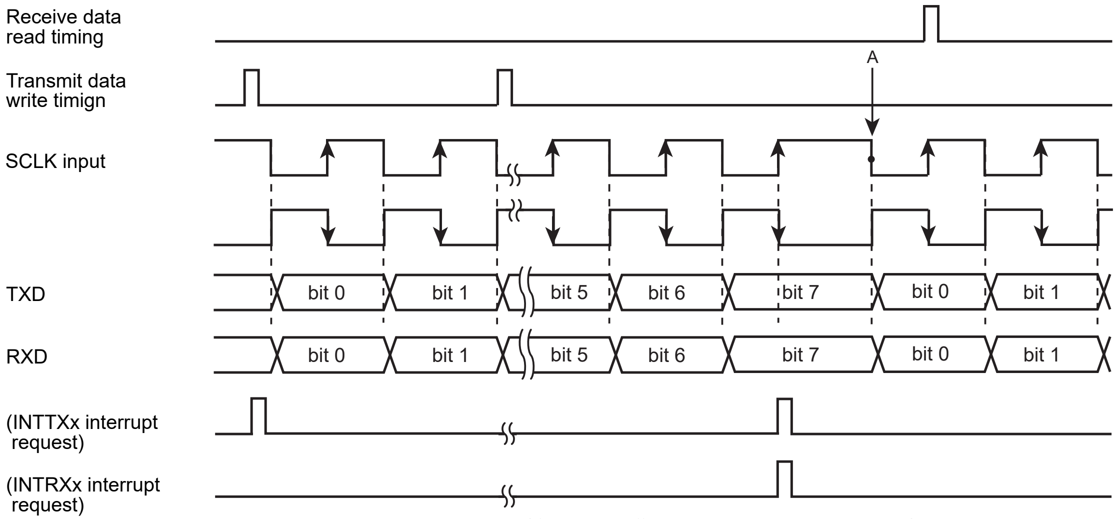
  Note that transmit data must be written into the transmit buffer before the SCLK input for the next frame (data must be written before the point A in Figure 12-17). Data must be read before completing reception of the next frame data.

  Upon the SCLK input for the next frame, transmission from transmit shift register (in which data has been moved from transmit buffer) is started while receive data is shifted into receive shift register simultaneously.

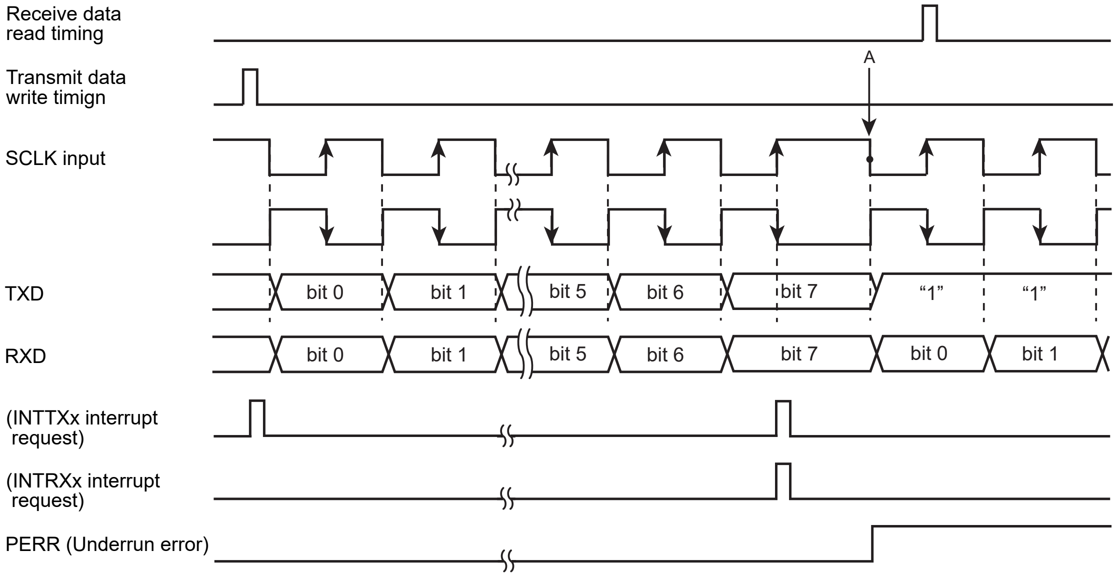
  If data in receive buffer has not been read when the last bit of the frame is received, an overrun error occurs. Similarly, if there is no data written to transmit buffer when SCLK for the next frame is input, an under-run error occurs and the dummy data (0xff) is output.


Receive data read timing
Transmit data write timign
A
SCLK input
TXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
RXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
(INTTXx interrupt request)
(INTRXx interrupt request)

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


Receive data read timing
Transmit data write timign
A
SCLK input
TXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
RXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
(INTTXx interrupt request)
(INTRXx interrupt request)

<WBUF> = "1" (if double buffering is enabled with no errors)


Receive data read timing
Transmit data write timign
A
SCLK input
TXD
bit 0
bit 1
bit 5
bit 6
bit 7
"1"
"1"
RXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
(INTTXx interrupt request)
(INTRXx interrupt request)
PERR (Underrun error)

<WBUF> = "1" (if double buffering is enabled with error generation)

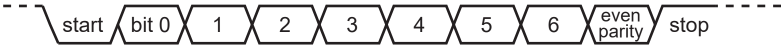
Figure 12-17 Transmit / Receive Operation in the I/O Interface Mode (SCLK Input Mode)

12.17.2 7-bit UART Mode

The 7-bit UART mode can be selected by setting the mode control register (SCxMOD<SM[1:0]>) to "01".

In this mode, parity bits can be added to the transmit data stream; the serial mode control register (SCxCR<PE>) controls the parity enable/disable setting. When <PE> is set to "1" (enable), either even or odd parity may be selected using the SCxCR<EVEN> bit. The length of the stop bit can be specified using SCxMOD2<SBLEN>.

The following table shows the control register settings for transmitting in the following data format.



← Transmission direction (Transmission rate of 2400 bps @fc = 9.8304 MHz)

Clocking condition	System clock :	High-speed (fc)
	High-speed clock gear :	x1 (fc)
	Prescaler clock :	fperiph/2 (fperiph = fsys)

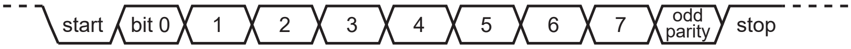
		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	-	0	0	1	0	1	Set 7-bit UART mode
SCxCR	←	x	1	1	x	x	x	0	0	Even parity enabled
SCxBRCR	←	0	0	1	0	0	1	0	0	Set 2400bps
SCxBUF	←	*	*	*	*	*	*	*	*	Set transmit data

x : don’t care - : no change

12.17.3 8-bit UART Mode

The 8-bit UART mode can be selected by setting SCxMOD0<SM[1:0]> to "10". In this mode, parity bits can be added and parity enable/disable is controlled using SCxCR<PE>. If <PE> = "1" (enabled), either even or odd parity can be selected using SCxCR<EVEN>.

The control register settings for receiving data in the following format are as follows :



← Transmission direction (Transmission rate of 9600 bps @fc = 9.8304 MHz)

Clocking condition	System clock :	High-speed (fc)
	High-speed clock gear :	x1 (fc)
	Prescaler clock :	fperiph/2 (fperiph = fsys)

		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	0	0	1	0	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

12.17.4 9-bit UART Mode

The 9-bit UART mode can be selected by setting SCxMOD0<SM[1:0]> to "11". In this mode, parity bits must be disabled (SCxCR<PE> = "0").

The most significant bit (9th bit) is written to SCxMOD0<TB8> for transmitting data. The received data is stored in SCxCR<RB8>.

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>.

12.17.4.1 Wake-up Function

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting the wake-up function control bit SCxMOD0<WU> to "1".

In this case, the interrupt INTRXx will be generated only when SCxCR<RB8> is set to "1".

Note: **The TXD pin of the slave controller must be set to the open drain output mode using the PxOD register.**

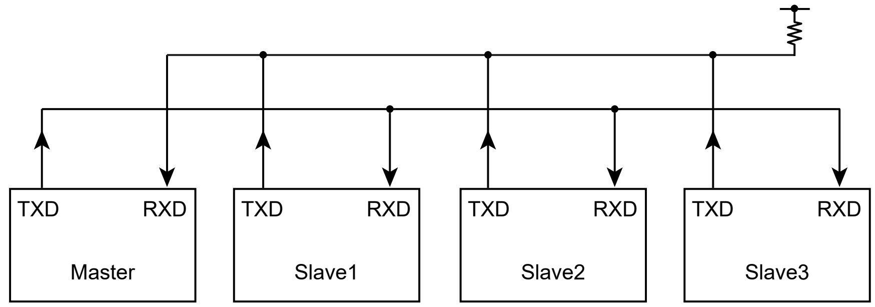


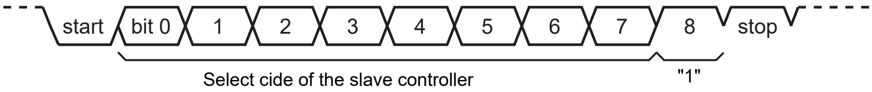
Figure 12-18 Serial Links to Use Wake-up Function

12.17.4.2 Protocol

1. Select the 9-bit UART mode for the master and slave controllers.

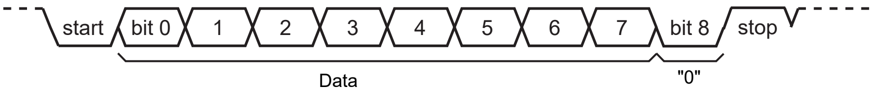
2. Set SCxMOD0<WU> to "1" for the slave controllers to make them ready to receive data.

3. The master controller is to transmit a single frame of data that includes the slave controller select code (8 bits). In this, the most significant bit (bit 8) <TB8> must be set to "1".



4. Each slave controller receives the above data frame; if the code received matches with the controller’s own select code, it clears the <WU> bit to "0".

5. The master controller transmits data to the designated slave controller (the controller of which SCxMOD<WU> bit is cleared to "0"). In this, the most significant bit (bit 8) <TB8> must be set to "0".



6. The slave controllers with the <WU> bit set to "1" ignore the receive data because the most significant bit (bit 8) <RB8> is set to "0" and thus no interrupt (INTRXx) is generated.Also, the slave controller with the <WU> bit set to "0" can transmit data to the master controller to inform that the data has been successfully received.

13. Asynchronous Serial Channel (UART)

13.1 Overview

This device has the Asynchronous serial channel (UART) with Modem control.

Their features are given in the following.

- Transmit FIFO

  8-bit width/ 32 location deep.
- Receive FIFO

  12-bit width/ 32 location deep.
- Transmit /Receive data format
  - DATA bits: 5,6,7,8 bits can be selected.
  - PARITY : use / no use
  - STOP bit : 1bit / 2 bits
- FIFO ON/OFF
  - ON (FIFO mode)/
  - OFF (characters mode)
- Interrupt
  - Combined interrupt factors are output to interrupt controller.
  - The permission of each interrupt factor is programmable.
- Baud rate generator

  Generates a common transmit and receive internal clock form UART internal reference clock input.

  Supports baud rates of up to 2.95Mbps at fsys = 48MHz.
- DMA
- IrDA 1.0 Function
  - Max data rate : 115.2 kbps (half-duplex).
  - support low power mode
- Control pins

  TXDx (IROUTx)

  RXDx (IRINx)

  $\overline{\text{CTSx}}$

  RINx

  $\overline{\text{RTSx}}$

  DCDx

  DSRx

  DTRx
- Hardware flow control

  RTS support

  CTS support

(1) UART transmit / receive data format.

Transmit / receive data format			
START	DATA (LSB → MSB)	PARITY	STOP

(2)Receive FIFO data format

	Receive data (LSB → MSB)								Framing error flag	Parity error flag	Break error flag	Overrun error flag
Bit Number	0	1	2	3	4	5	6	7				
Receive 8-bit data	1	1	1	1	1	1	1	1				
Receive 7-bi data	1	1	1	1	1	1	1	0				
Receive 6-bit data	1	1	1	1	1	1	0	0				
Receive 5-bit data	1	1	1	1	1	0	0	0				

13.2 Configuration

Figure 13-1 shows UART block diagram.

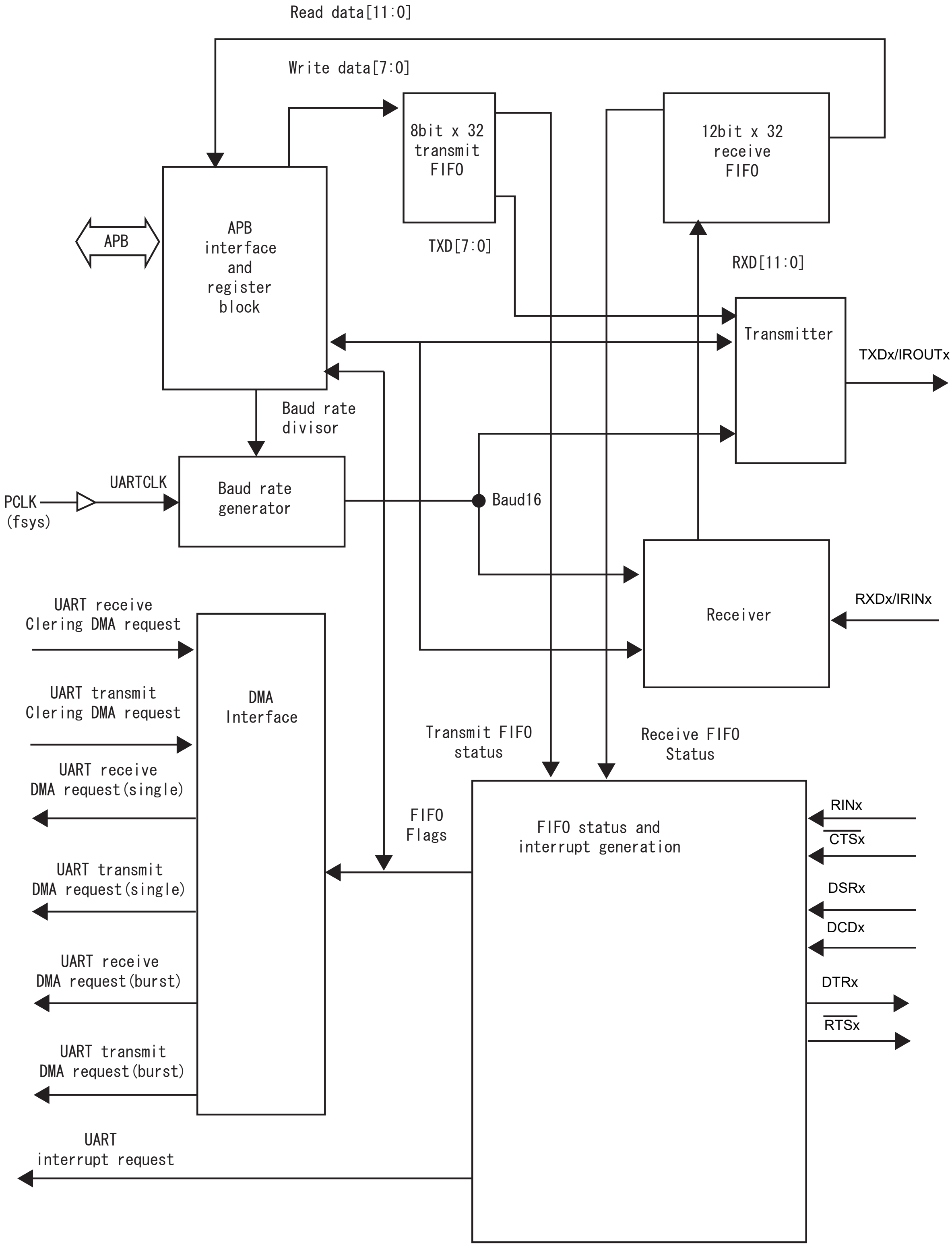

Read data[11:0]
Write data[7:0]
8bit x 32 transmit FIFO
12bit x 32 receive FIFO
APB
APB interface and register block
TXD[7:0]
RXD[11:0]
Transmitter
TXDx/IROUTx
Baud rate divisor
UARTCLK
PCLK (fsys)
Baud rate generator
Baud16
Receiver
RXDx/IRINx
UART receive Clering DMA request
UART transmit Clering DMA request
UART receive DMA request(single)
UART transmit DMA request(single)
UART receive DMA request(burst)
UART transmit DMA request(burst)
DMA Interface
Transmit FIFO status
Receive FIFO Status
FIFO Flags
FIFO status and interrupt generation
RINx
CTSx
DSRx
DCDx
DTRx
RTSx
UART interrupt request

Figure 13-1 UART Channel Block Diagram

13.3 Registers Description

13.3.1 Registers List

The channel registers and addresses are shown below.

Channel x	Base Address
Channel4	0x4004_8000
Channel5	0x4004_9000

Register name (x= 4, 5)		Address (Base+)
UART Data register	UARTxDR	0x0000
UART Receive status register	UARTxRSR	0x0004
UART Error clear register	UARTxECR	0x0004
Reserved	-	0x0008 to 0x0017
UART Flag register	UARTxFR	0x0018
Reserved	-	0x001C
UART IrDA low-power counter	UARTxILPR	0x0020
UART Integer baud rate register	UARTxIBDR	0x0024
UART Fractional baud rate register	UARTxFBDR	0x0028
UART Line control register	UARTxLCR_H	0x002C
UART Control register	UARTxCR	0x0030
UART Interrupt FIFO level select register	UARTxIFLS	0x0034
UART Interrupt mask set/clear register	UARTxIMSC	0x0038
UART Raw interrupt status register	UARTxRIS	0x003C
UART Masked interrupt status register	UARTxMIS	0x0040
UART Interrupt clear register	UARTxICR	0x0044
UART DMA control register	UARTxDMACR	0x0048
Reserved	-	0x004C to 0x0FFF

Note: **You must disable the UART before any of the control registers are reprogrammed. When the UART is disabled in the middle of transmit or receive operation, it stops after the transmission of the current character is completed.**

13.3.2 UARTxDR (UART 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 This bit is set to 1 if data is received and the receive FIFO is already full. In this case, the received data is not stored in the FIFO and is discarded. The bit is cleared to 0 once an empty space is made in the FIFO and a new data can be written to it.
10	BE	R	Break error This bit is set to 1 if a break condition was detected, indicating that the receive data input (defined as start, data parity, and stop bits) was held Low for a period longer than a full-word transmission time. In FIFO mode, this error is occurred by the highest character in FIFO. Break is occurred only when one zero character is loaded in FIFO. The next character is enabled when received data makes 1 (marking state) and the next effective start bit is received.
9	PE	R	Parity error When this bit is set to 1, it indicates that the parity of the received data does not match the parity defined by bits 2 and 7 of the UARTxLCR_H register. In FIFO mode, this error is occurred by the highest character in FIFO.
8	FE	R	Framing error When this bit is set to 1, it indicates that the received data did not have a valid stop bit (a valid stop bit is 1). In FIFO mode, this error is occurred by the highest character in FIFO.
7-0	DATA[7:0]	R/W	Read : Receive data Write : Transmit data

13.3.3 UARTxRSR (UART Receive status Register)

UARTxRSR and UARTxECR are mapped to same address.

These functions differ in read and write operations.

	31	30	29	28	27	26	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 This bit is set to 1 if data is received and the FIFO is already full. This bit is cleared 0 by writing a data to UARTxECR. When FIFO is already full, the received data is not stored in the FIFO. Therefore the contents of FIFO are valid and only the contents of shift register is over written. In this case, CPU must be read out a data from FIFO to make empty FIFO.
2	BE	R	Break error This bit is set to 1 if a break condition was detected, indicating that the receive data input (defined as start, data bit, data parity, and stop bits) was held Low for longer than a full-word transmission time. This bit is cleared 0 by writing a data to UARTxECR. In FIFO mode, this error is occurred by the highest character in FIFO. Break is occurred only when one zero character is loaded in FIFO. The next character is enabled when received data makes 1 (marking state) and the next effective start bit is received.
1	PE	R	Parity error When this bit is set to 1, it indicates that the parity of the received data does not match the parity defined by bits 2 and 7 of the UARTxLCR_H register. This bit is cleared 0 by writing a data to UARTxECR. In FIFO mode, this error is occurred by the highest character in FIFO.
0	FE	R	Framing error When this bit is set to 1, it indicates that the received data did not have a valid stop bit (a valid stop bit is 1). This bit is cleared 0 by writing a data to UARTxECR. In FIFO mode, this error is occurred by the highest character in FIFO.

13.3.4 UARTxECR (UART 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	A write to this register clears framing, parity, break, and overrun errors. The data value has no significance. The address of this register is the same as that of the UARTxRSR register.
2	BE	W	
1	PE	W	
0	FE	W	

Note 1: The UARTxRSR/UARTxECR register is the receive status register/error clear register. Receive status can also be read from UARTxRSR. If the status is read from this register, the status information for break, framing and parity corresponds to the data read from UARTxDR prior to reading UARTxRSR. The status information for overrun is set immediately when an overrun condition occurs. A write to UARTxECR clears the framing, parity, break and overrun errors. All the bits are cleared to 0 on reset.

Note 2: The receive data must be read first from UARTxDR before the error status associated with that data is read from UARTxRSR. This read sequence cannot be reversed because the status register UARTxRSR is updated only when the data is read from the data register UARTxDR. The status information can also be read directly from the UARTxDR register.

13.3.5 UARTxFR (UART Flag register)

The <TXFE>, <RXFF>, <TXFF>, and <RXFE> bits differ depending on the state of the <FEN> of the UARTxLCR_H 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	-	-	-	-	-	-	-	RI
After reset	0	0	0	0	0	0	0	Undefined
	7	6	5	4	3	2	1	0
bit symbol	TXFE	RXFF	TXFF	RXFE	BUSY	DCD	DSR	CTS
After reset	0	0	0	0	0	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-9	−	R	Read as undefined.
8	RI	R	RIng indicator flag. 1: Modem status input = 0
7	TXFE	R	UARTxLCR_H<FEN>=1 (Transmit FIFO empty flag) 0: Not empty 1: Empty
			UARTxLCR_H<FEN>=0 (Transmit hold register empty flag) 0: Not empty 1: Empty
6	RXFF	R	UARTxLCR_H<FEN>=1 (Receive FIFO full flag) 0: Not full 1: Full
			UARTxLCR_H<FEN>=0 (Receive hold register full flag) 0: Not full 1: Full
5	TXFF	R	UARTxLCR_H<FEN>=1 (Transmit FIFO full flag) 0: Not full 1: Full
			UARTxLCR_H<FEN>=0 (Transmit hold register full flag) 0: Not full 1: Full
4	RXFE	R	UARTxLCR_H<FEN>=1 (Receive FIFO empty flag) 0: Not empty 1: Empty
			UARTxLCR_H<FEN>=0 (Receive hold register empty flag) 0: Not empty 1: Empty
3	BUSY	R	BUSY flag 0: The UART has stopped transmitting data 1: The UART is transmitting data. (BUSY)
2	DCD	R	Data carrier detect (DCD) flag 0: DCDx pin is "High" 1: DCDx pin is "Low"

Bit	Bit Symbol	Type	Function
1	DSR	R	Data set ready (DSR) flag 0: DSRx pin is "High" 1: DSRx pin is "Low"
0	CTS	R	Clear To Send (CTS) flag 0: $\overline{\text{CTSx}}$ pin is "High" 1: $\overline{\text{CTSx}}$ pin is "Low"

1. Transmit FIFO

   The transmit FIFO is an 8-bit wide, 32-location deep FIFO memory buffer. CPU data written across the APB interface is stored in the FIFO until it is read out by the transmit logic. The transmit FIFO can be disabled to act like a one-byte holding register.

2. Receive FIFO

   The receive FIFO is a 12-bit wide, 32-location deep, FIFO memory buffer. Received data and corresponding error bits are stored in the receive FIFO by the receive logic until they are read out by the CPU across the APB interface. The receive FIFO can be disabled to act like a one-byte holding register.

13.3.6 UARTxILPR(UART IrDA low-power counter 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	ILPDVSR							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	ILPDVSR[7:0]	R/W	Low-power divisor <ILPDVSR> = ($f_{UARTCLK}$ / $f_{IrLPBaud16}$). The UARTxILPR register is the IrDA low-power counter register. This is an 8-bit read/write register that stores the low-power counter divisor value used to generate the IrLPBaud16 signal by dividing down of UARTCLK. All the bits are cleared to 0 when reset

Note 1: **Set this register before the UARTxCR<SIRLP> is set to 1.**

Note 2: **0x00 setting is prohibited.**

13.3.7 UARTxIBDR (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 undefined.
15-0	BAUD DIVINT [15:0]	R/W	Integer part of buad rate divisor. (0x0002 to 0xFFFF) This register, when put together with the fractional baud rate divisor described next, provides the baud rate divisor BAUDDIV.

Note 1: **To update the contents of UARTxIBRD internally, the write to UARTxLCR_H must always be executed last. For details, refer to the description of UARTxLCR_H.**

Note 2: **Set this register before the UARTxCR<UARTEN> is set to 1.**

Note 3: **0x0000, 0x0001 cannot be set.**

Note 4: **The value of the worst case baud rate divisor of the set value due to the baud rate shift (total error) lbetween the transmitter side and the receiver side is as shown in the table below. (8 bit data + Parity / 9 bit data)**

Total error	BAUDDIVINT(Lower limit)
2.0% or less	0x0002
2.8% or less	0x0003
3.3% or less	0x0004

13.3.8 UARTxFBDR(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	BAUDDIV FRAC [5:0]	R/W	Fractional part of baud rate divisor. 0x01 to 0x3F. The baud rate divisor is calculated as follows: Baud rate divisor BAUDDIV = ($f_{UARTCLK}$)/ (16 × baud rate) $f_{UARTCLK}$ is the frequency of UARTCLK. The BAUDDIV is comprised of the integer value (BAUD DIVINT) and the fractional value (BAUD DIVFRAC).

Note 1: To update the contents of UARTxFBRD internally, the write to UARTxLCR_H must always be executed last. For details, refer to the description of UARTxLCR_H.

Note 2: Set this register before the UARTxCR<UARTEN> is set to 1.

Note 3: The minimum baud rate divisor is 1 and the maximum baud rate divisor is 65535. Therefore, The integral part of baud rate divisor can not be set 0. And the fractional part of baud rate divisor must be set to 0 when the integral part of baud rate divisor is 65535.

Example: Calculating the divisor value

When the required baud rate is 230400 and $f_{UARTCLK}$ = 4 MHz:

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

Therefore, BRDI = 1 and BRDF = 0.085

Fractional part is $((0.085 \times 64) + 0.5) = 5.94$.

The integer part of this, m=0x5, should be set as the fractional baud rate divisor value.

Generated baud rate divisor = $1 + 5/64 = 1.078$

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

Error = $(231911 - 230400)/ 230400 \times 100 = 0.656\%$

The maximum error using a 6-bit UARTxFBRD register = $1/64 \times 100 = 1.56\%$

This error occurs when m = 1, and it is cumulative over 64 clock ticks.

13.3.9 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	Stick parity select : refer to Table 13-1 for the truth table. When <SPS>, <EPS> and <PEN> of the UARTxLCR_H register are set, the parity bit is transmitted and checked as a 0. When <SPS> and <PEN> are set and <EPS> is 0, the parity bit is transmitted and checked as a 1. When this bit is cleared, the stick parity is disabled. Refer toTable 13-1 for the truth table of <SPS>, <EPS>, and <PEN> bits.
6-5	WLEN[1:0]	R/W	Word length : 00: 5bits 01: 6bits 10: 7bits 11: 8bits This bit indicates the number of data bits transmitted or received in a frame.
4	FEN	R/W	FIFO control : 0: character mode 1: FIFO mode When this bit is set to 1, transmit and receive FIFO buffers are enabled (FIFO mode). When this bit is cleared to 0, the FIFOs are disabled (character mode) and they become 1-byte deep holding registers.
3	STP2	R/W	Stop bit select : 0: 1bit 1: 2 bits When this bit is set to 1, two stop bits are transmitted at the end of a frame. The receive logic does not check for the second stop bit being received.
2	EPS	R/W	Even parity select: 0: Odd 1: Even When this bit is set to 1, even parity generation and checking are performed during transmission and reception. This function checks whether the number of 1s contained in the data bits and parity bit is even. When this bit is cleared to 0, odd parity check is performed to check whether the number of 1s is odd. This bit has no effect when parity is disabled by Parity Enable bit <PEN> being cleared to 0. Refer to Table 13-1 for the truth table.
1	PEN	R/W	Parity control: 0:Disable 1: Enable When this bit is set to 1, parity check and generation are enabled. Otherwise, parity is disabled and no parity bit is added to data frames. Refer to Table 13-1 for the truth table of <SPS>, <EPS>, and <PEN> bits.
0	BRK	R/W	Send break : 0: No effect 1: Send break When this bit is set to 1, the TXDx output remains LOW after the current character is transmitted. For generation of the transmit break condition, this bit must be asserted while at least one frame is or longer being transmitted. Even when the break condition is generated, the contents of the transmit FIFO are not affected.

Note: **When you set UARTxLCR_H, UARTxIBRD and UARTxFBRD, UARTxLCR_H must be set at the end. When you update only UARTxIBRD or UARTxFBRD, UARTxLCR_H register must be set again.**

Table 13-1 is the truth table of the <SPS>, <EPS> and <PEN> bits of the UARTxLCR_H register.

Table 13-1 Truth table of UARTxLCR_H <SPS>, <EPS> and <PEN>

Parity enable <PEN>	Even parity select <EPS>	Stick parity select <SPS>	Parity bit(transmitted or checked)
0	×	×	No transmitted or checked
1	1	0	Even parity
1	0	0	Odd parity
1	0	1	1
1	1	1	0

13.3.10 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	CTSEN	RTSEN	-	-	RTS	DTR	RXE	TXE
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	SIRLP	SIREN	UARTEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15	CTSEBN	R/W	CTS hardware flow control enable : 0:Disable 1: Enable When this bit is set to 1, CTS hardware flow control is enabled. Data is transmitted only after the $\overline{\text{CTSx}}$ signal has been asserted.
14	RTSEN	R/W	RTS hardware flow control enable : 0: Disable 1: Enable When this bit is set to 1, RTS hardware flow control is enabled. Data is transmitted only when there is an empty space in the receive FIFO.
13-12	-	R	Read as undefined.
11	RTS	R/W	Complement of the UART Request To Send (RTS) modem status output : 0: Modem status output is 1 1: Modem status output is 0 . This bit is the inverting UART Request To Send (RST) modem status output signal. When this bit is set to 1, the output is 0.
10	DTR	R/W	Complement of the UART Data Set Ready (DTS) modem status output : 0: Modem status output is 1. 1: Modem status output is 0. This bit is the inverting UART Data Transmit Ready (DTR) modem status output signal. When this bit is set to 1, the output is 0.
9	RXE	R/W	UART receive enable : 0: Disable 1: Enable When this bit is set to 1, the receive circuit of the UART is enabled. Data reception occurs for either UART function or SIR function according to the setting of <SIREN>. When the UART is disabled in the middle of receive operation, it completes current reception and the subsequent receptions are disabled.
8	TXE	R/W	UART transmit enable : 0: Disable 1: Enable When this bit is set to 1, the transmit circuit of the UART is enabled. Data transmission occurs for either UART function or SIR function according to the setting of <SIREN>. When the UART is disabled in the middle of transmit operation, it completes the current transmission before stopping.
7	Reserved	R/W	Write as zero.
6-3	Reserved	-	Read as undefined.

Bit	Bit Symbol	Type	Function
2	SIRLP	R/W	IrDA encoding mode select for transmitting 0 bit : 0 : 0 bit are transmitted as an active high pulse of 3/16th of the bit period. 1 : 0 bit are transmitted with a pulse width that is 3 times the period of the IrLPBaud16 input signal. <SIRLP> selects IrDA encoding mode. When this bit is cleared to 0, 0 bits of the IrDA transmission data are transmitted as an active high pulse (IROUT) with a width of 3/16th of the bit period. When this bit is set to 1, 0 bits of the IrDA transmission data are transmitted with a pulse width that is 3 times the period of the IrLPBaud16 input signal. Setting this bit can reduce power consumption but might decrease transmission distances.
1	SIREN	R/W	SIR enable : 0 : Disable 1 : Enable When this bit is set to 1, the IrDA circuit is enabled. To use the UART, the <UARTEN> must be set to 1. When the IrDA circuit is enabled, the IROUT and IRIN pins are enabled. The TXD pin remains in the marking state (set to 1). Signal transitions on the RXD pin or modem status input have no effect. When IrDA circuit is disabled, IROUT remains cleared to 0 (no light pulse is generated) and the IRIN pin has no effect.
0	UARTEN	R/W	UART enable : 0 : Disable 1 : Enable When this bit is set to 1, the UART is enabled. Data transmission and reception occur for either UART function or SIR function according to the setting of <SIREN>. When the UART is disabled in the middle of transmit or receive operation, it completes current transmission or reception before stopping.

13.3.11 UARTxIFLS (UART interrupt FIFO level select 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 undefined .
5-3	RXIFLSEL[2:0]	R/W	Receive interrupt FIFO level select (1 word = 12 bits): The trigger points for the receive interrupt are as follows 000: Receive FIFO becomes ≥ 1/8 full 001: Receive FIFO becomes ≥ 1/4 full 010: Receive FIFO becomes ≥ 1/2 full 011: Receive FIFO becomes ≥ 3/4 full 100: Receive FIFO becomes ≥ 7/8 full 101to 111: Reserved
2-0	TXIFSEL[2:0]	R/W	Transmit FIFO level select (1 word = 8 bits) : The trigger points for the transmit interrupt are as follows: 000: Transmit FIFO becomes ≤ 1/8 full 001: Transmit FIFO becomes ≤ 1/4 full 010: Transmit FIFO becomes ≤ 1/2 full 011: Transmit FIFO becomes ≤ 3/4 full 100: Transmit FIFO becomes ≤ 7/8 full 101 to 111: Reserved

The UARTxIFLS register is the interrupt FIFO level select register. This register is used to define the FIFO level at which UARTTXINTR and UARTRXINTR are generated.

The interrupts are generated based on a transition through a level rather than based on the level. For example, an interrupt is generated at a point when the third word has been stored in the receive FIFO which contained two words.

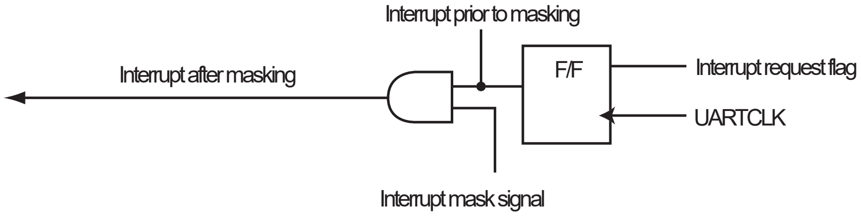
13.3.12 UARTxIMSC (UART Interrupt mask set/clear Register)

	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	-	-	-	-	-	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	DSRMIM	DCDMIM	CTSMIM	RIMIM
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-11	−	R	Read as undefined.
10	OEIM	R/W	Overrun error interrupt mask : 0: Clear the mask 1: Set the mask
9	BEIM	R/W	Break error interrupt mask : 0: Clear the mask 1: Set the mask
8	PEIM	R/W	Parity error interrupt mask : 0: Clear the mask 1: Set the mask
7	FEIM	R/W	Framing error interrupt mask : 0: Clear the mask 1: Set the mask
6	RTIM	R/W	Receive time out interrupt mask : 0: Clear the mask 1: Set the mask
5	TXIM	R/W	Transmit FIFO interrupt mask : 0: Clear the mask 1: Set the mask
4	RXIM	R/W	Receive FIFO interrupt mask : 0: Clear the mask 1: Set the mask
3	DSRMIM	R/W	DSR modem interrupt mask : 0: Clear the mask 1: Set the mask
2	DCDMIM	R/W	DCD modem interrupt mask : 0: Clear the mask 1: Set the mask
1	CTSMIM	R/W	CTS modem interrupt mask : 0: Clear the mask 1: Set the mask
0	RIMIM	R/W	RIN modem interrupt mask : 0: Clear the mask 1: Set the mask.

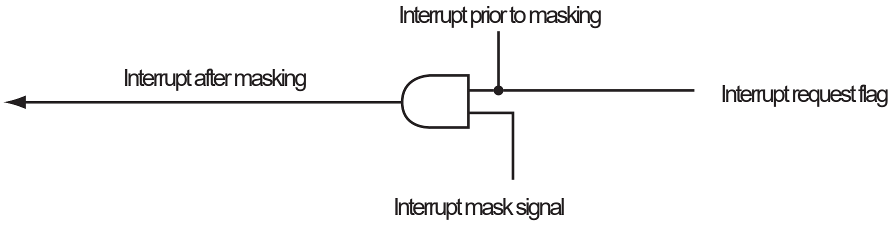
・UART interrupt generation block diagrams.

1. Block diagram of the break error <BE>, parity error <PE> and framing error <PE> flags


Interrupt prior to masking
Interrupt after masking
F/F
Interrupt request flag
UARTCLK
Interrupt mask signal

・The interrupt request flag state changes in real time and is retained in the F/F. Each flag can be cleared by a write to the corresponding bit in the interrupt clear register.

2. Block diagram of the overrun error <OE> flag.


Interrupt prior to masking
Interrupt after masking
Interrupt request flag
Interrupt mask signal

・The interrupt request flag state by the overrun error <OE> flag changes in real time and its state is not retained . And the <OE> flag is cleared by a read of receive FIFO.

13.3.13 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	DSRRMIS	DCDRMIS	CTSRMIS	RIRMIS
After reset	0	0	0	0	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-11	−	R	Read as undefined.
10	OERIS	R	Overrun error raw interrupt status : 0: Interrupt not requested 1: Interrupt requested.
9	BERIS	R	Break error raw interrupt status : 0: Interrupt not requested 1: Interrupt requested.
8	PERIS	R	Parity error raw interrupt status : 0: Interrupt not requested 1: Interrupt requested.
7	FERIS	R	Framing error raw interrupt status : 0: Interrupt not requested 1: Interrupt requested.
6	RTRIS	R	Receive time out raw interrupt status : 0: Interrupt not requested 1: Interrupt requested.
5	TXRIS	R	Transmit raw interrupt status : 0: Interrupt not requested 1: Interrupt requested.
4	RXRIS	R	Receive raw interrupt status : 0: Interrupt not requested 1: Interrupt requested.
3	DSRRMIS	R	DSR modem raw interrupt status : 0: Interrupt not requested 1: Interrupt requested.
2	DCDRMIS	R	DCD modem raw interrupt status : 0: Interrupt not requested 1: Interrupt requested.
1	CTSRMIS	R	CTS modem raw interrupt status : 0: Interrupt not requested 1: Interrupt requested.
0	RIRMIS	R	RIN modem raw interrupt status : 0: Interrupt not requested 1: Interrupt requested.

Note: **All the bits, except the modem raw status interrupt bits (bits 3 to 0), are cleared to 0 when reset. The modem status bit are undefined after reset.**

13.3.14 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	DSRMMIS	DCDMMIS	CTSMMIS	RIMMIS
After reset	0	0	0	0	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-11	−	R	Read as undefined.
10	OEMIS	R	Overrun error masked interrupt status: 0: Interrupt not requested. 1: Interrupt requested.
9	BEMIS	R	Break error masked interrupt status: 0: Interrupt not requested. 1: Interrupt requested.
8	PEMIS	R	Parity error masked interrupt status: 0: Interrupt not requested. 1: Interrupt requested.
7	FEMIS	R	Framing error masked interrupt status: 0: Interrupt not requested. 1: Interrupt requested.
6	RTMIS	R	Receive time out masked interrupt status: 0: Interrupt not requested. 1: Interrupt requested.
5	TXMIS	R	Transmit masked interrupt status: 0: Interrupt not requested. 1: Interrupt requested.
4	RXMIS	R	Receive masked interrupt status: 0: Interrupt not requested. 1: Interrupt requested.
3	DSRMMIS	R	DSR modem masked interrupt status: 0: Interrupt not requested. 1: Interrupt requested.
2	DCDMMIS	R	DCD modem masked interrupt status: 0: Interrupt not requested. 1: Interrupt requested.
1	CTSMMIS	R	CTS modem masked interrupt status: 0: Interrupt not requested. 1: Interrupt requested.
0	RIMMIS	R	RIN modem masked interrupt status: 0: Interrupt not requested. 1: Interrupt requested.

Note: **All the bits, except the modem masked status interrupt bits (bits 3 to 0), are cleared to 0 when reset. The modem status bits are undefined after reset.**

13.3.15 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	DSRMIC	DCDMIC	CTSMIC	RIMIC
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-11	−	R	Write 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	Received 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	DSRMIC	W	DSR modem interrupt clear : 0: Invalid 1: Clear
2	DCDMIC	W	DCD modem interrupt clear : 0: Invalid 1: Clear
1	CTSMIC	W	CTS modem interrupt clear : 0: Invalid 1: Clear
0	RIMIC	W	RIN modem interrupt clear : 0: Invalid 1: Clear

Note: The UARTxICR register is a write-only interrupt clear register. When a bit of this register is set to 1, the associated interrupt is cleared. A write of 0 to any bit of this register is invalid.

13.3.16 UARTxDMACR (UART DMA 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	-	-	-	-	-	DMAONERR	TXDMAE	RXDMAE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as undefined.
2	DMAONERR	R/W	DMA on error : 0: Not available 1: Available When this bit is set to 1, the DMA receive request output, UARTRXDMASREQ or UARTRXDMABREQ, is disabled on assertion of a UART error interrupt.
1	TXDMAE	R/W	Transmit FIFO DMA enable : 0: Disable 1: Enable
0	RXDMAE	R/W	Receive FIFO DMA enable : 0: Disable 1: Enable

Note 1: For example, if 19 characters have to be received and the watermark level is programmed to be four, then the DMA controller transfers four bursts of four characters and three single transfers to complete the stream.

Note 2: The bus width must be set to 8-bits, if you transfer the data of transmit/ receive FIFO by using DMAC.

13.4 Operation Description

13.4.1 Baud rate generator

The baud rate generator contains the internal Baud16 clock circuit which controls the timing of UART transmit and receive, and the internal IrLPBaud16 circuit which generates the pulse width of the IrDA encoded transmit bit stream when in low-power mode.

13.4.2 Transmit FIFO

The transmit FIFO is an 8-bit wide, 32-location deep, FIFO memory buffer. CPU data written across the APB interface is stored in the FIFO until it is read out by the transmit logic. You can disable the transmit FIFO to act like a one-byte holding register.

13.4.3 Receive FIFO

The receive FIFO is a 12-bit wide, 32 locations deep, FIFO memory buffer. Received data and corresponding error bits are stored in the receive FIFO by the receive logic until they are read out by the CPU across the APB interface. The receive FIFO can be disabled to act like a one-byte holding register.

13.4.4 Transmit logic

The transmit logic performs parallel-to-serial conversion on the data read from the transmit FIFO. Control logic outputs the serial bit stream beginning with a start bit, data bits with the Least Significant Bit (LSB) first, followed by the parity bit, and then the stop bits according to the programmed configuration in control registers.

13.4.5 Receive logic

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

13.4.6 Interrupt generation logic

UART outputs a maskable combined interrupt for every interrupt sources.

13.4.7 Interrupt timing

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

Note: STOP bit in above table means the last STOP bit. (The number of STOP bit can be selected among 1 or 2.)

13.4.8 UART interrupt block

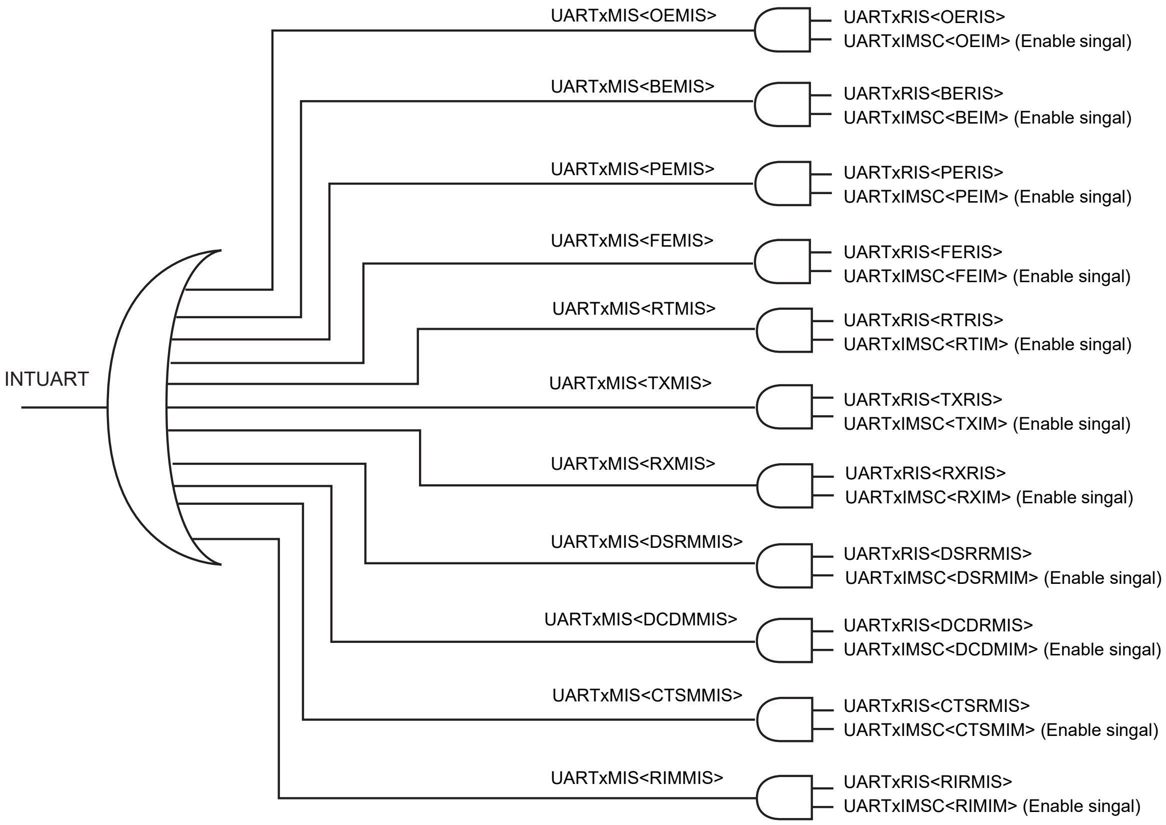

UARTxMIS<OEMIS>
UARTxRIS<OERIS>
UARTxIMSC<OEIM> (Enable singal)
UARTxMIS<BEMIS>
UARTxRIS<BERIS>
UARTxIMSC<BEIM> (Enable singal)
UARTxMIS<PEMIS>
UARTxRIS<PERIS>
UARTxIMSC<PEIM> (Enable singal)
UARTxMIS<FEMIS>
UARTxRIS<FERIS>
UARTxIMSC<FEIM> (Enable singal)
UARTxMIS<RTMIS>
UARTxRIS<RTRIS>
UARTxIMSC<RTIM> (Enable singal)
INTUART
UARTxMIS<TXMIS>
UARTxRIS<TXRIS>
UARTxIMSC<TXIM> (Enable singal)
UARTxMIS<RXMIS>
UARTxRIS<RXRIS>
UARTxIMSC<RXIM> (Enable singal)
UARTxMIS<DSRMMIS>
UARTxRIS<DSRRMIS>
UARTxIMSC<DSRMIM> (Enable singal)
UARTxMIS<DCDMMIS>
UARTxRIS<DCDRMIS>
UARTxIMSC<DCDMIM> (Enable singal)
UARTxMIS<CTSMMIS>
UARTxRIS<CTSRMIS>
UARTxIMSC<CTSMIM> (Enable singal)
UARTxMIS<RIMMIS>
UARTxRIS<RIRMIS>
UARTxIMSC<RIMIM> (Enable singal)

Figure 13-2 UART interrupt block

13.4.9 DMA interface

The UART support DMA controller.

13.4.10 IrDA circuit description

The IrDA is comprised of:

- IrDA SIR transmit encoder
- IrDA SIR receive decoder

Note: **The transmit encoder output (IROUT) has the opposite polarity to the receive decoder input (IRIN). Please refer to Figure 13-4**

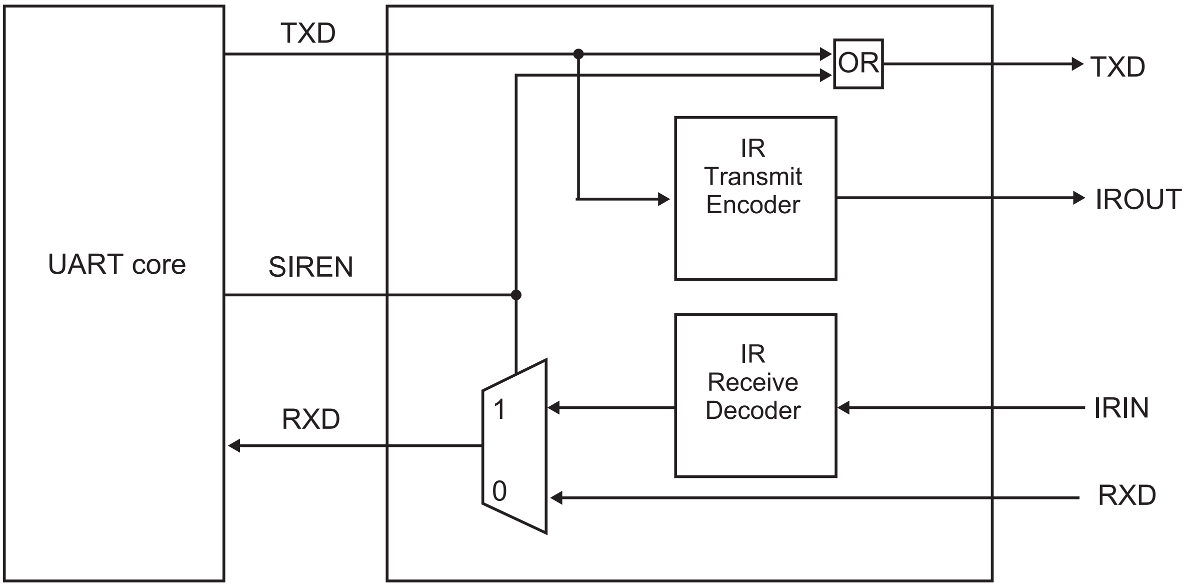

UART core
TXD
SIREN
RXD
OR
TXD
IR
Transmit
Encoder
IROUT
IR
Receive
Decoder
IRIN
RXD
1
0

Figure 13-3 IrDA Circuit Block Diagram

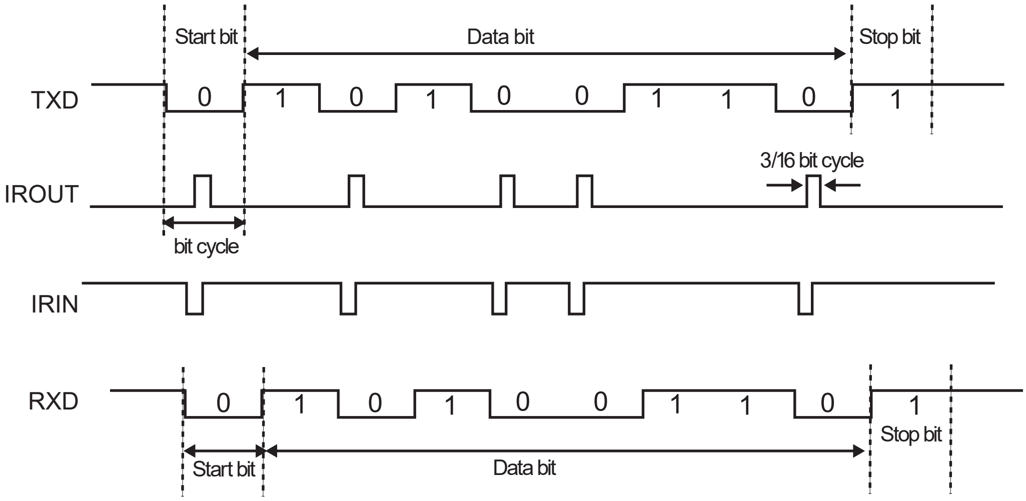

Start bit
Data bit
Stop bit
TXD
0
1
0
1
0
0
1
1
0
1
IROUT
3/16 bit cycle
bit cycle
IRIN
RXD
0
1
0
1
0
0
1
1
0
1
Stop bit
Start bit
Data bit

Figure 13-4 IrDA Data Modulation

13.4.11 Hardware flow control

The hardware flow control feature is fully selectable, and enables you to control the serial data flow by using the $\overline{\text{RTSx}}$ output and $\overline{\text{CTSx}}$ input signals.

Figure 13-5 shows how the two devices can communicate with each other using hardware flow control.

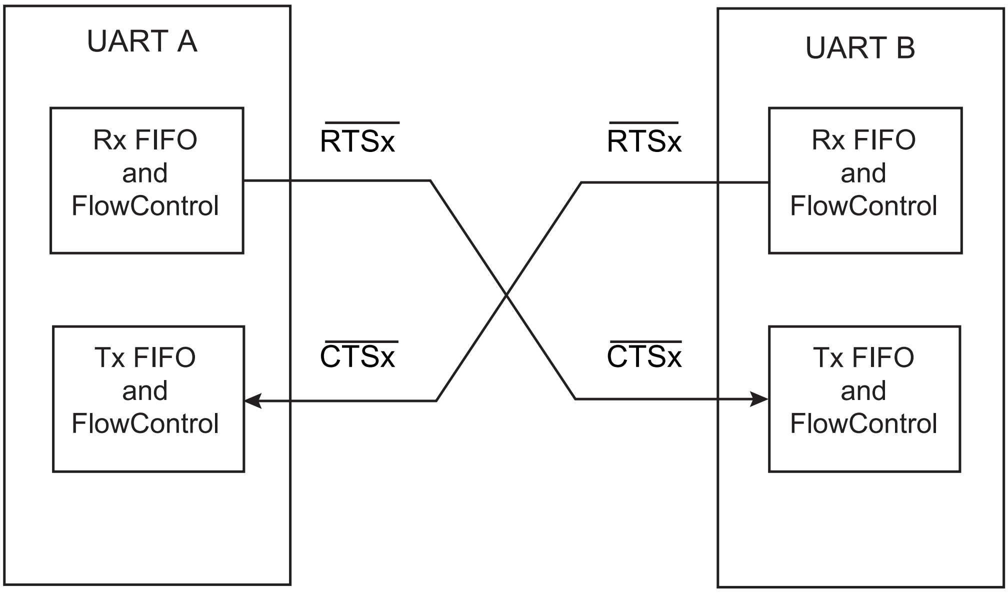

UART A
Rx FIFO and FlowControl
Tx FIFO and FlowControl
$\overline{\text{RTSx}}$
$\overline{\text{CTSx}}$
$\overline{\text{RTSx}}$
$\overline{\text{CTSx}}$
UART B
Rx FIFO and FlowControl
Tx FIFO and FlowControl

Figure 13-5 Hardware Flow Control

1. RTS flow control

   The RTS flow control logic is linked to the programmable receive FIFO watermark levels. When RTS flow control is enabled, the $\overline{\text{RTSx}}$ is asserted until the receive FIFO is filled up to the watermark level.

   When the amount of data stored in the receive FIFO exceeds watermark level, the $\overline{\text{RTSx}}$ signal is deasserted, indicating that there is no more room to receive.

   The $\overline{\text{RTSx}}$ signal is reasserted when data has been read out of the receive FIFO and it is filled to less than the watermark level.

   Even if RTS flow control is disabled, communication can be enabled.

2. CTS flow control

   If CTS flow control is enabled, then the transmitter checks the $\overline{\text{CTSx}}$ signal before transmitting. If the $\overline{\text{CTSx}}$ signal is asserted, it transmits the byte, otherwise transmission does not occur.

   The data transmission continues while $\overline{\text{CTSx}}$ is asserted and the transmit FIFO is not empty. If the transmit FIFO is empty, no data is transmitted even when the $\overline{\text{CTSx}}$ signal is asserted.

   If the $\overline{\text{CTSx}}$ signal is deasserted while CTS flow control is enabled, the current data transmission is completed before stopping.

   Even if CTS flow control is disabled, communication can be enabled.

UARTxCR		RTSx	Description
<CTSEN>	<RTSEN>		
1	1	0 (note)	Both RTS and CTS flow control enabled.
1	0	1	Only CTS flow control enabled.
0	1	0 (note)	Only RTS flow control enabled.
0	0	1	Both RTS and CTS flow control disabled.

Note: During in the <RTSEN>=1(Enable), the $\overline{\text{RTSx}}$ is set to 0(Enable) until the receive FIFO is filled up to the watermark level.

14. Serial Bus Interface (I2C/SIO)

The TMPM367FDFG contains 3 Serial Bus Interface (I2C/SIO) channels, in which the following two operating modes are included:

- I2C bus mode (with multi-master capability)
- Clock-synchronous 8-bit SIO mode

In the I2C bus mode, the I2C/SIO is connected to external devices via SCL and SDA.

In the clock-synchronous 8-bit SIO mode, the I2C/SIO is connected to external devices via SCK, SI and SO.

The following table shows the programming required to put the I2C/SIO in each operating mode.

Table 14-1 Port settings for using serial bus interface

channel	Operating mode	pin	Port Function Register	Port Output Control Register	Port Input Control Register	Port Open Drain Output Control Register
SBI0	I2C bus mode	SCL0 :PK3 SDA0 :PK2	PKFR3[3:2] = 11	PKCR[3:2] = 11	PKIE[3:2] = 11	PKOD[3:2] = 11
	SIO mode	SCK0 :PK4 SI0 :PK3 SO0 :PK2	PKFR3[4:2] = 111	PKCR[4:2] = 101(SCK0 output) PKCR[4:2] = 001(SCK0 input)	PKIE[4:2] = 010(SCK0 output) PKIE[4:2] = 110(SCK0 input)	PKOD[4:2] = xxx
SBI1	I2C bus mode	SCL1 :PF6 SDA1 :PF7	PFFR4[6:5] = 11	PFCR[6:5] = 11	PFIE[6:5] = 11	PFOD[6:5] = 11
	SIO mode	SCK1 :PF5 SI1 :PF6 SO1 :PF7	PFFR4[7:5] = 111	PFCR[7:5] = 101(SCK0 output) PFCR[7:5] = 100(SCK0 input)	PFIE[7:5] = 010(SCK0 output) PFIE[7:5] = 011(SCK0 input)	PFOD[7:5] = xxx
SBI2	I2C bus mode	SCL2 :PH1 SDA2 :PH0	PHFR5[1:0] = 11	PHCR[1:0] = 11	PHIE[1:0] = 11	PHOD[1:0] = 11
	SIO mode	SCK2 :PH2 SI2 :PH1 SO2 :PH0	PHFR5[2:0] = 111	PHCR[2:0] = 101(SCK0 output) PHCR[2:0] = 001(SCK0 input)	PHIE[2:0] = 010(SCK0 output) PHIE[2:0] = 110(SCK0 input)	PHOD[2:0] = xxx

Note:x: Don't care

14.1 Configuration

The configuration is shown in Figure 14-1.

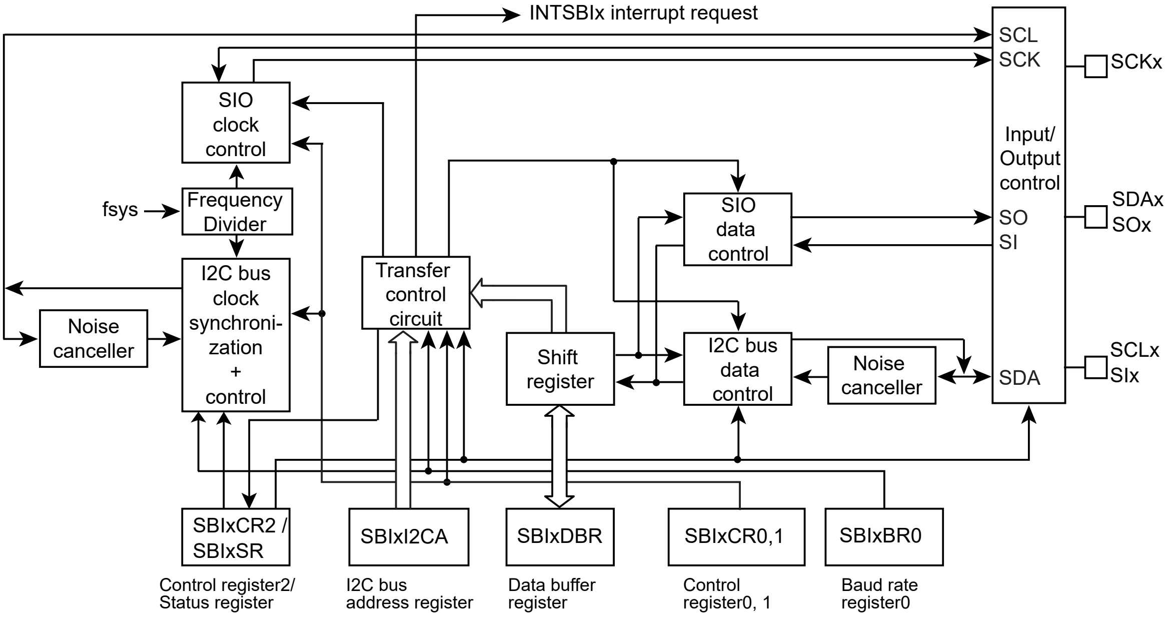

INTSBIx interrupt request
SCL
SCK
SCKx
Input/ Output control
SO
SI
SDAx
SOx
SDA
SCLx
SIx
SIO clock control
fsys
Frequency Divider
I2C bus clock synchroni- zation + control
Noise canceller
Transfer control circuit
Shift register
SIO data control
I2C bus data control
Noise canceller
SBIxCR2 / SBIxSR
SBIxI2CA
SBIxDBR
SBIxCR0,1
SBIxBR0
Control register2/ Status register
I2C bus address register
Data buffer register
Control register0, 1
Baud rate register0

Figure 14-1 (I2C/SIO) Block Interface

14.2 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 "14.4 Control Registers in the I2C Bus Mode" and "14.7 Control register of SIO mode".

14.2.1 Registers for each channel

The tables below show the registers and register addresses for each channel.

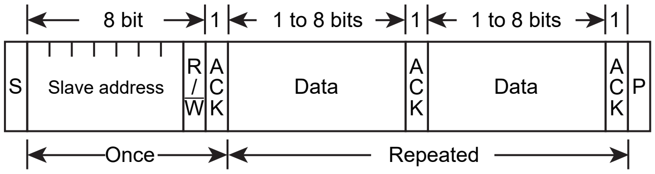
Channel x	Base Address
Channel0	0x400E_0000
Channel1	0x400E_0100
Channel2	0x400E_0200

Register name(x=0,1,2,)		Address(Base+)
Control register 0	SBIxCR0	0x0000
Control register 1	SBIxCR1	0x0004
Data buffer register	SBIxDBR	0x0008
I2C bus address register	SBIxI2CAR	0x000C
Control register 2	SBIxCR2 (writing)	0x0010
Status register	SBIxSR (reading)	
Baud rate register 0	SBIxBR0	0x0014

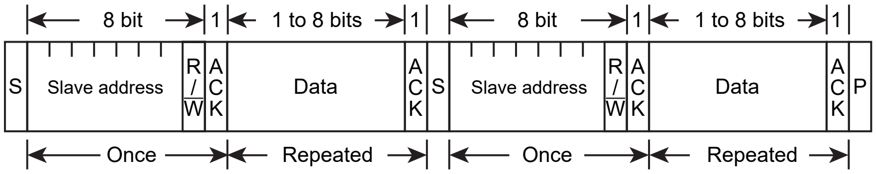
14.3 I2C Bus Mode Data Format

Figure 14-2 shows the data formats used in the I2C bus mode.

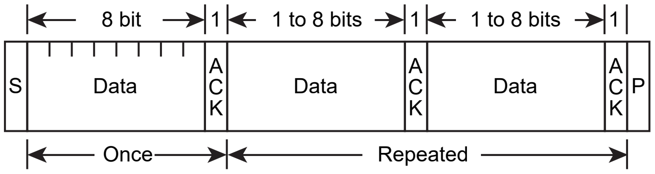
(a) Addressing format


8 bit
1
1 to 8 bits
1
1 to 8 bits
1
S
Slave address
R/W
ACK
Data
ACK
Data
ACK
P
Once
Repeated

(b) Addressing format (with repeated start condition)


8 bit
1
1 to 8 bits
1
8 bit
1
1 to 8 bits
1
S
Slave address
R/W
ACK
Data
ACK
S
Slave address
R/W
ACK
Data
ACK
P
Once
Repeated
Once
Repeated

(c) Free data format (master-transmitter to slave-receiver)


8 bit
1
1 to 8 bits
1
1 to 8 bits
1
S
Data
ACK
Data
ACK
Data
ACK
P
Once
Repeated

Note) S : Start condition
$R/\overline{W}$: Direction bit
ACK : Acknowledge bit
P : Stop condition

Figure 14-2 I2C Bus Mode Data Formats

14.4 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.

14.4.1 SBIxCR0(Control register 0)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SBIEN	R/W	Serial bus interface operation 0:Disable 1:Enable To use the serial bus interface, enable this bit first. For the first time in case of setting to enable, the relevant SBI registers can be read or written. Since all clocks except SBIxCR0 stop if this bit is disabled, power consumption can be reduced by disabling this bit. If this bit is disabled after it's been enabled once, the settings of each register are retained.
6-0	-	R	Read as 0.

Note: To use the serial bus interface, enable this bit first.

14.4.2 SBIxCR1(Control register 1)

<table>
<tr><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td>bit symbol</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>After reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><th></th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td>bit symbol</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>After reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><th></th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td>bit symbol</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>After reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><th></th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>bit symbol</td><td colspan="3">BC</td><td>ACK</td><td>-</td><td>SCK2</td><td>SCK1</td><td>SCK0 /
SWRMON</td></tr>
<tr><td>After reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1(Note3)</td></tr>
</table>

<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"><BC></th><th colspan="2">When <ACK> = 0</th><th colspan="2">When <ACK> = 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>769kHz</td><td rowspan="8">System Clock: fsys
(= 80MHz)
Clock gear : fc/1
Frequency = $\frac{\text{fsys}}{2^n + 72}$ [Hz]</td></tr>
<tr><td>001</td><td>n = 6</td><td>588kHz</td></tr>
<tr><td>010</td><td>n = 7</td><td>400 kHz</td></tr>
<tr><td>011</td><td>n = 8</td><td>244 kHz</td></tr>
<tr><td>100</td><td>n = 9</td><td>137 kHz</td></tr>
<tr><td>101</td><td>n = 10</td><td>73 kHz</td></tr>
<tr><td>110</td><td>n = 11</td><td>38 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 "14.5.1 Serial Clock".**

Note 3: **After a reset, the <SCK[0]/SWRMON> bit is read as "1". However, if the SIO mode is selected at the SBIxCR2 register, the initial value of the <SCK[0]> bit is "0".**

Note 4: **The initial value for selecting a frequency is <SCK[2:0]>=000 and is independent of the read initial value.**

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 bumber of bits per transfer should be set equal or more than 2 before generation of STOP condition.**

14.4.3 SBIxCR2(Control register 2)

This register serves as SBIxSR register by reading it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	W	Select master/slave 0: Slave mode 1: Master mode
6	TRX	W	Select transmit/ receive 0: Receive 1: Transmit
5	BB	W	Start/stop condition generation 0: Stop condition generated 1: Start condition generated
4	PIN	W	Clear INTSBIx interrupt request 0: − 1: Clear interrupt request
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note) 00: Port mode (Disables a serial bus interface output) 01: SIO mode 10: I2C bus mode 11: Reserved
1-0	SWRST[1:0]	W	Software reset generation Write "10" followed by "01" to generate a reset. When writing, set <SBIM[1:0]> to "10"; I2Cbus mode.

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.**

14.4.4 SBIxSR (Status Register)

This register serves as SBIxCR2 by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	R	Master/slave selection monitor 0: Slave mode 1: Master mode
6	TRX	R	Transmit/receive selection monitor 0: Receive 1: Transmit
5	BB	R	I2C bus state monitor 0: Free 1: Busy
4	PIN	R	INTSBIx interrupt request monitor 0:Interrupt request generated 1: Interrupt request cleared
3	AL	R	Arbitration lost detection 0: − 1:Detected
2	AAS	R	Slave address match detection 0: − 1: Detected (This bit is set when the general call is detected as well.)
1	ADO	R	General call detection 0: − 1:Detected
0	LRB	R	Last received bit monitor 0:Last received bit "0" 1:Last received bit "1"

14.4.5 SBIxBR0(Serial bus interface baud rate register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
After reset	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	-	R	Read as 1.
6	I2SBI	R/W	Operation at the IDLE mode 0: Stop 1: Operate
5-1	-	R	Read as 1.
0	-	R/W	Be sure to write "0".

14.4.6 SBIxDBR (Serial bus interface data buffer register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DB[7:0]	R (Receive)/ W (Transmit)	Receive data / Transmit data

Note 1: **The transmission data must be written in to the register from the MSB (bit 7). The received data is stored in the LSB.**

Note 2: **Since SBIxBDR 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.**

14.4.7 SBIxI2CAR (I2Cbus address register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-1	SA[6:0]	R/W	Set the slave address when the SBI acts as a slave device.
0	ALS	R/W	Specify address recognition mode. 0: Recognize its slave address. 1: Do not recognize its slave address (free-data format).

Note 1: **Please set the bit 0 <ALS> of I2C bus address register SBIxI2CAR to "0", except when you use a free data format. It operates as a free data format when setting it to "1". Selecting the master fixes to transmission. Selecting the slave fixes to reception.**

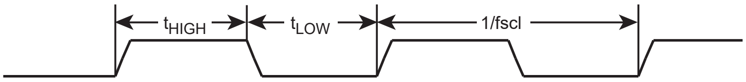
Note 2: **Do not set SBIxI2CAR to "0x00" in slave mode. (If SBIxI2CAR is set to "0x00", it’s recognized that the slave address matches the START byte ("0x01") of the I2C standard received in slave mode.)**

14.5 Control in the I2C Bus Mode

14.5.1 Serial Clock

14.5.1.1 Clock source

SBIxCR1<SCK[2:0]> specifies the maximum frequency of the serial clock to be output from the SCL pin in the master mode.


tHIGH
tLOW
1/fscl

$$t_{LOW} = 2^{n-1}/fsys + 58/fsys$$
$$t_{HIGH} = 2^{n-1}/fsys + 14/fsys$$
$$fscl = 1/(t_{LOW} + t_{HIGH}) = \frac{fsys}{2^n + 72}$$

SBIxCR1<SCK[2:0]>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Figure 14-3 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.**

14.5.1.2 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.

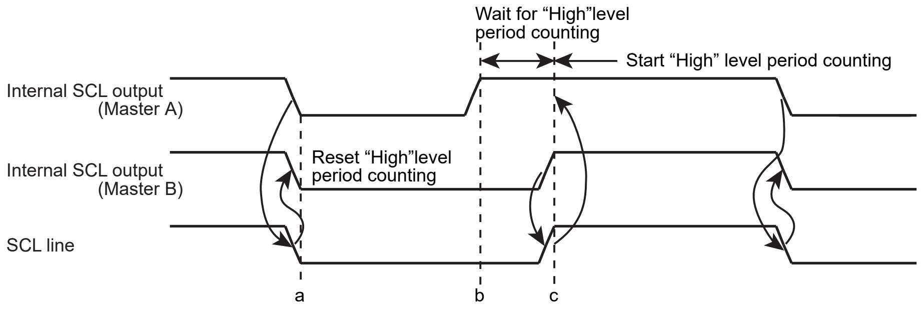

Wait for "High"level period counting
Start "High" level period counting
Internal SCL output (Master A)
Internal SCL output (Master B)
Reset "High"level period counting
SCL line
a
b
c

Figure 14-4 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.

14.5.2 Setting the Acknowledgement Mode

Setting SBIxCR1<ACK> to "1" selects the acknowledge mode.When operating as a master, the SBI adds one clock for acknowledgment signal. In slave mode, the clock for acknowledgement signals is counted. In transmitter mode, the SBI releases the SDAx pin during clock cycle to receive acknowledgement signals from the receiver. In receiver mode, the SBI pulls the SDAx pin to the "Low" level during the clock cycle and generates acknowledgement signals. Also in slave mode, if a general-call address is received, the SBI pulls the SDAx pin to the "Low" level during the clock cycle and generates acknowledgement signals.

By setting <ACK> to "0", the non-acknowledgment mode is activated. When operating as a master, the SBI does not generate clock for acknowledgement signals. In slave mode, the clock for acknowledgement signals is counted.

14.5.3 Setting the Number of Bits per Transfer

SBIxCR1<BC[2:0]> specifies the number of bits of the next data to be transmitted or received.

Under the start condition, <BC[2:0]> is set to "000", causing a slave address and the direction bit to be transferred in a packet of eight bits. At other times, <BC[2:0]> keeps a previously programmed value.

14.5.4 Slave Addressing and Address Recognition Mode

Setting "0" to SBIxI2CAR<ALS> and a slave address in SBIxI2CAR<SA[6:0]> sets addressing format, and then the SBI recognizes a slave address transmitted by the master device and receives data in the addressing format.

If <ALS> is set to "1", the SBI does not recognize a slave address and receives data in the free data format. In the case of free data format, a slave address and a direction bit are not recognized; they are recognized as data immediately after generation of the start condition.

14.5.5 Operating mode

The setting of SBIxCR2<SBIM[1:0]> controls the operating mode. To operate in I2C mode, ensure that the serial bus interface pins are at "High" level before setting <SBIM[1:0]> to "10". Also, ensure that the bus is free before switching the operating mode to the port mode.

14.5.6 Configuring the SBI as a Transmitter or a Receiver

Setting SBIxCR2<TRX> to "1" configures the SBI as a transmitter. Setting <TRX> to "0" configures the SBI as a receiver.

At the slave mode:

- when data is transmitted in the addressing format.
- when the received slave address matches the value specified at SBIxI2CAR.
- when a general-call address is received; i.e., the eight bits following the start condition are all zeros.

If the value of the direction bit ($R/\overline{W}$) is "1", <TRX> is set to "1" by the hardware. If the bit is "0", <TRX> is set to "0".

As a master device, the SBI receives acknowledgement from a slave device. If the direction bit of "1" is transmitted, <TRX> is set to "0" by the hardware. If the direction bit is "0", <TRX> changes to "1". If the SBI does not receive acknowledgement, <TRX> retains the previous value.

<TRX> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

If SBI is used in free data format, <TRX> is not changed by the hardware.

14.5.7 Configuring the SBI as a Master or a Slave

Setting SBIxCR2<MST> to "1" configures the SBI to operate as a master device.

Setting <MST> to "0" configures the SBI as a slave device. <MST> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

14.5.8 Generating Start and Stop Conditions

When SBIxSR<BB> is "0", writing "1" to SBIxCR2<MST, TRX, BB, PIN> causes the SBI to start a sequence for generating the start condition and to output the slave address and the direction bit prospectively written in the data buffer register. <ACK> must be set to "1" in advance.

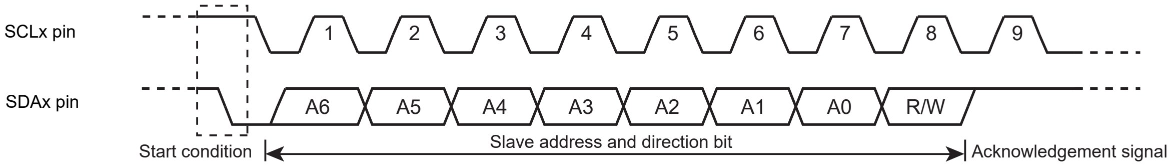


Figure 14-5 Generating the Start Condition and a Slave Address

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.

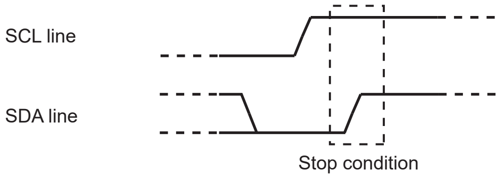


Figure 14-6 Generating the Stop Condition

SBIxSR<BB> can be read to check the bus state. <BB> is set to "1" when the start condition is detected on the bus (the bus is busy), and cleared to "0" when the stop condition is detected (the bus is free).

14.5.9 Interrupt Service Request and Release

In master mode, a serial bus interface request (INTSBIx) is generated when the transfer of the number of clock cycles set by <BC> and <ACK> is completed.

In slave mode, INTSBIx is generated under the following conditions.

- After output of the acknowledge signal which is generated when the received slave address matches the slave address set to SBIxI2CAR<SA[6:0]>.
- After the acknowledge signal is generated when a general-call address is received.
- When the slave address matches or a data transfer is completed after receiving a general-call address.

In the address recognition mode (<ALS> = "0"), INTSBIx is generated when the received slave address matches the values specified at SBIxI2CAR or when a general-call (eight bits data following the start condition is all "0") is received.

When an interrupt request (INTSBIx) is generated, SBIxCR2<PIN> is cleared to "0". While <PIN> is cleared to "0", the SBI pulls the SCL line to the "Low" level.

<PIN> is set to "1" when data is written to or read from SBIxDBR. It takes a period of t_{LOW} for the SCL line to be released after <PIN> is set to "1". When the program writes "1" to <PIN>, it is set to "1". However, writing "0" does not clear this bit to "0".

Note:When arbitration occurs while a slave address and direction bit are transferred in the master mode, <PIN> is cleared to "0" and INTSBIx occurs. This does not relate to whether a slave address matches <SA>.

14.5.10 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.

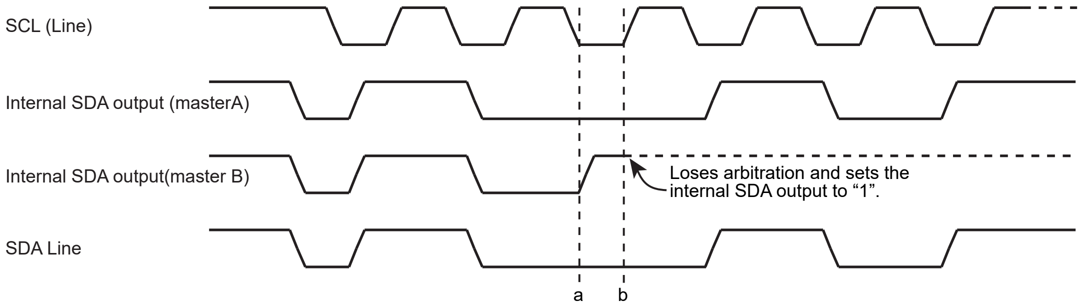

SCL (Line)
Internal SDA output (masterA)
Internal SDA output(master B)
SDA Line
Loses arbitration and sets the internal SDA output to “1”.
a
b

Figure 14-7 Lost Arbitration

A master compares the SDA bus line level and the internal SDA output level at the rising of the SCL line. If there is a difference between these two values, Arbitration Lost occurs and SBIxSR<AL> is set to "1".

When an arbitration lost occures, 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 after<AL> is set to "1".
If the master device which sends a slave address and direction bit generaters 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 INTSBIx occurs.

<AL> is cleared to "0" when data is written to or read from SBIxDBR or data is written to SBIxCR2.

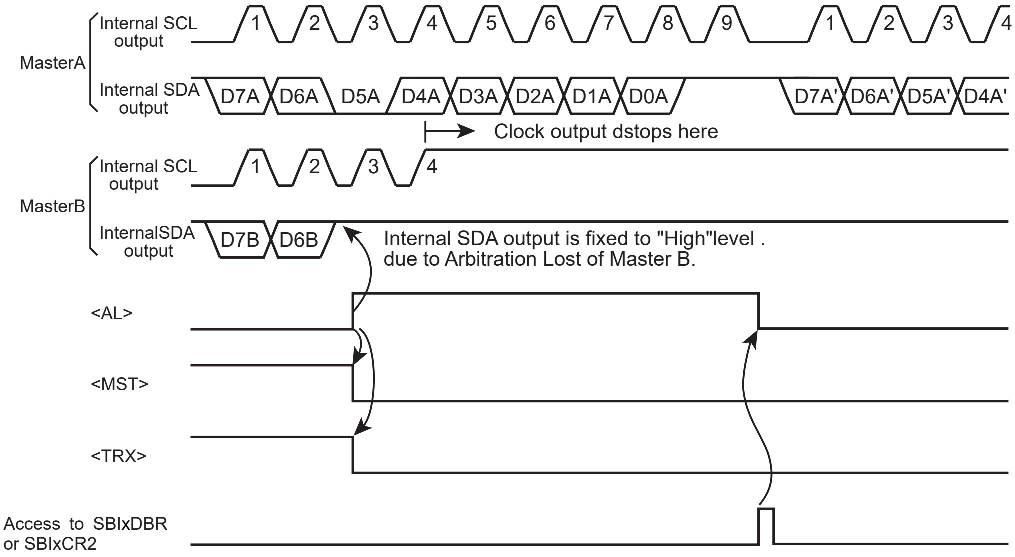

MasterA
Internal SCL output
1
2
3
4
5
6
7
8
9
1
2
3
4
Internal SDA output
D7A
D6A
D5A
D4A
D3A
D2A
D1A
D0A
D7A'
D6A'
D5A'
D4A'
Clock output dstops here
MasterB
Internal SCL output
1
2
3
4
InternalSDA output
D7B
D6B
Internal SDA output is fixed to "High"level .
due to Arbitration Lost of Master B.
<AL>
<MST>
<TRX>
Access to SBIxDBR
or SBIxCR2

Figure 14-8 Example of Master B Lost Arbitration (D7A = D7B, D6A = D6B)

14.5.11 Slave Address Match Detection Monitor

When the SBI operates as a slave device in the address recognition mode (SBIxI2CAR<ALS>="0"), SBIxSR<AAS> is set to "1" on receiving the general-call address or the slave address that matches the value specified at SBIxI2CAR.

When <ALS> is "1", <AAS> is set to "1" when the first data word has been received. <AAS> is cleared to "0" when data is written to or read from SBIxDBR.

14.5.12 General-call Detection Monitor

When the SBI operates as a slave device, SBIxSR<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.

14.5.13 Last Received Bit Monitor

SBIxSR<LRB> is set to the SDA line value that was read at the rising of the SCL line.

In the acknowledgment mode, reading SBIxSR<LRB> immediately after generation of the INTSBIx interrupt request causes ACK signal to be read.

14.5.14 Data Buffer Register (SBIxDBR)

Reading or writing SBIxDBR initiates reading received data or writing transmitted data.

When the SBI is acting as a master, setting a slave address and a direction bit to this register generates the start condition.

14.5.15 Baud Rate Register (SBIxBR0)

The SBIxBR0<I2SBI> register determines if the SBI operates or not when it enters the IDLE mode.

This register must be programmed before executing an instruction to switch to the standby mode.

14.5.16 Software Reset

If the serial bus interface circuit locks up due to external noise, it can be initialized by using a software reset.

Writing "10" followed by "01" to SBIxCR2<SWRST[1:0]> generates a reset signal that initializes the serial bus interface circuit. When writing, set <SBIM[1:0]> to "10"; I2Cbus 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.

14.6 Data Transfer Procedure in the I2C Bus ModeI2C

14.6.1 Device Initialization

Firstly, set SBIxCR1<ACK><SCK[2:0]>. Set "1" to <ACK> to specify the acknowledgement mode. Set "000" to SBIxCR1<BC[2:0]>.

Secondly, set<SA[6:0]>(a slave address) and <ALS> to SBIxI2CAR.(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 SBIxCR2<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	
SBIxCR1	←	0	0	0	1	0	X	X	X	Specifies ACK and SCL clock.
SBIxI2CAR	←	X	X	X	X	X	X	X	X	Specifies a slave address and an address recognition mode.
SBIxCR2	←	0	0	0	1	1	0	0	0	Configures the SBI as a slave receiver.

Note: X; Don’t care

14.6.2 Generating the Start Condition and a Slave Address

14.6.2.1 Master mode

In the master mode, the following steps are required to generate the start condition and a slave address.

First, ensure that the bus is free (<BB> = "0"). Then, write "1" to SBIxCR1<ACK> to select the acknowledgment mode. Write to SBIxDBR a slave address and a direction bit to be transmitted.

When <BB> = "0", writing "1111" to SBIxCR2<MST, TRX, BB, PIN> generates the start condition on the bus. Following the start condition, the SBI generates nine clocks from the SCL pin. The SBI outputs the slave address and the direction bit specified at SBIxDBR with the first eight clocks, and releases the SDA line in the ninth clock to receive an acknowledgment signal from the slave device.

The INTSBIx interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0". In the master mode, the SBI holds the SCL line at the "Low" level while <PIN> is = "0".<TRX> changes its value according to the transmitted direction bit at generation of the INTSBIx interrupt request, provided that an acknowledgment signal has been returned from the slave device.

Note: To output salve address, check with software that the bus is free before writing to SBIxDBR. If this rule is not followed, data being output on the bus may get ruined.

Settings in main routine

		7	6	5	4	3	2	1	0	
Reg.	←	SBIxSR								
Reg.	←	Reg. e 0x20								
if Reg.	≠	0x00								Ensures that the bus is free.
Then										
SBIxCR1	←	X	X	X	1	0	X	X	X	Selects the acknowledgement mode.
SBIxDBR	←	X	X	X	X	X	X	X	X	Specifies the desired slave address and direction.
SBIxCR2	←	1	1	1	1	1	0	0	0	Generates the start condition.

Example of INTSBI0 interrupt routine

Clears the interrupt request.

Processing

End of interrupt

14.6.2.2 Slave mode

In the slave mode, the SBI receives the start condition and a slave address.

After receiving the start condition from the master device, the SBI receives a slave address and a direction bit from the master device during the first eight clocks on the SCL line.

If the received address matches its slave address specified at SBIxI2CAR or is equal to the general-call address, the SBI pulls the SDA line to the "Low" level during the ninth clock and outputs an acknowledgment signal.

The INTSBIx interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0". In the slave mode, the SBI holds the SCL line at the "Low" level while <PIN> is "0".

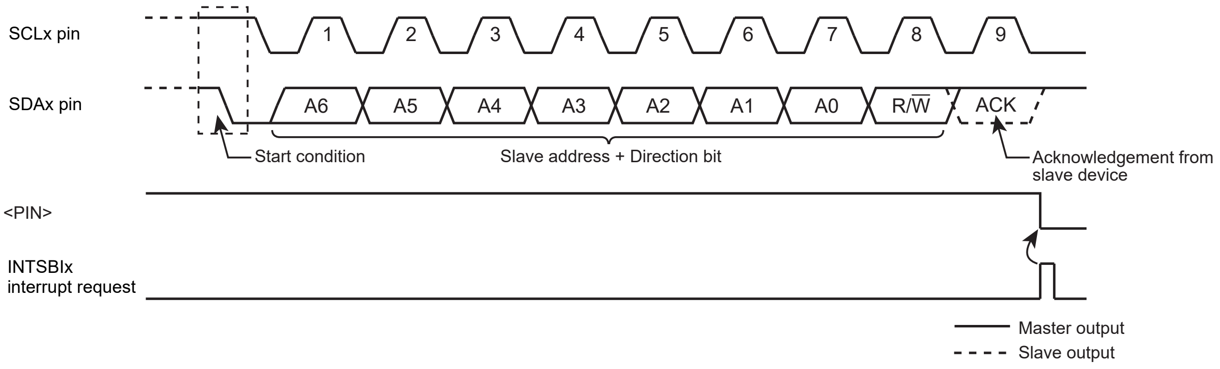

SCLx pin
1
2
3
4
5
6
7
8
9
SDAx pin
A6
A5
A4
A3
A2
A1
A0
R/W̄
ACK
Start condition
Slave address + Direction bit
Acknowledgement from slave device
<PIN>
INTSBIx interrupt request
Master output
Slave output

Figure 14-9 Generation of the Start Condition and a Slave Address

14.6.3 Transferring a Data Word

At the end of a data word transfer, the INTSBIx interrupt is generated to test <MST> to determine whether the SBI is in the master or slave mode.

14.6.3.1 Master mode (<MST> = "1")

Test <TRX> to determine whether the SBI is configured as a transmitter or a receiver.

(1) Transmitter mode (<TRX> = "1")

Test <LRB>. If <LRB> is "1", that means the receiver requires no further data.

The master then generates the stop condition as described later to stop transmission.

If <LRB> is "0", that means the receiver requires further data.If the next data to be transmitted has eight bits, the data is written into SBIxDBR. If the data has different length, <BC[2:0]> and <ACK> are programmed and the transmit data is written into SBIxDBR.Writing the data makes <PIN> to "1", causing the SCL pin to generate a serial clock for transferring a next data word, and the SDA pin to transfer the data word.

After the transfer is completed, the INTSBIx interrupt request is generated, <PIN> is cleared to "0", and the SCL pin is pulled to the "Low" level.

To transmit more data words, test <LRB> again and repeat the above procedure.

INTSBIx interrupt

```
if MST = 0
Then go to the slave-mode processing.
if TRX = 0
Then go to the receiver-mode processing.
if LRB = 0
Then go to processing for generating the stop condition.
SBIxCR1   ←  X  X  X  X  0  X  X  X    Specifies the number of bits to be transmitted and
                                        specify whether ACK is required.
SBIxDBR   ←  X  X  X  X  X  X  X  X    Writes the transmit data.
End of interrupt processing.
```

Note: **X; Don't care**

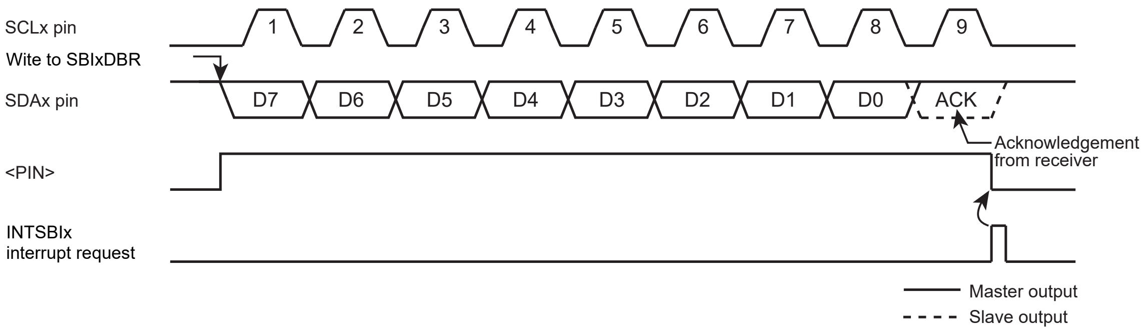

SCLx pin
Wite to SBIxDBR
SDAx pin
<PIN>
INTSBIx interrupt request
1
2
3
4
5
6
7
8
9
D7
D6
D5
D4
D3
D2
D1
D0
ACK
Acknowledgement from receiver
Master output
Slave output

Figure 14-10 <BC[2:0]>= "000",<ACK>= "1" (Transmitter Mode)

(2) Receiver mode (<TRX> = "0")

If the next data to be transmitted has eight bits, the transmit data is written into SBIxDBR.

If the data has different length, <BC[2:0]> and <ACK> are programmed and the received data is read from SBIxDBR to release the SCL line. (The data read immediately after transmission of a slave address is undefined.)On reading the data, <PIN> is set to "1", and the serial clock is output to the SCL pin to transfer the next data word.In the last bit, when the acknowledgment signal becomes the "Low" level, "0" is output to the SDA pin.

After that, the INTSBIx interrupt request is generated, and <PIN> is cleared to "0", pulling the SCL pin to the "Low" level.Each time the received data is read from SBIxDBR, one-word transfer clock and an acknowledgement signal are output.

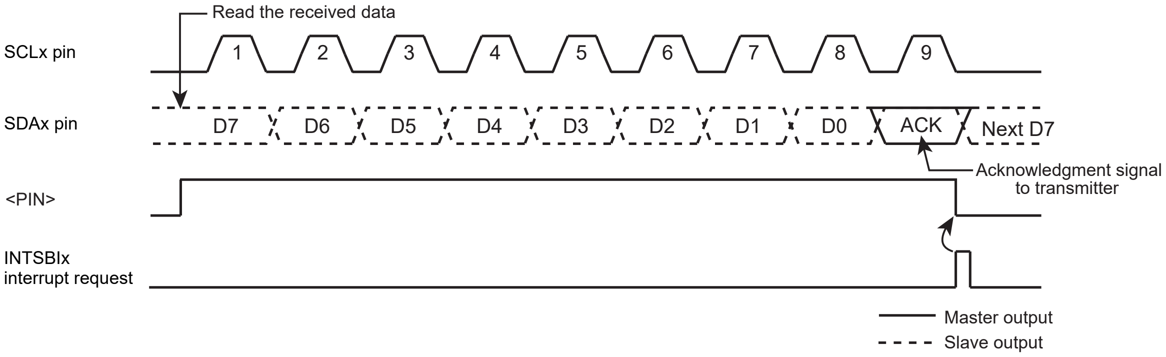

Read the received data
SCLx pin
SDAx pin
<PIN>
INTSBIx interrupt request
1
2
3
4
5
6
7
8
9
D7
D6
D5
D4
D3
D2
D1
D0
ACK
Next D7
Acknowledgment signal to transmitter
Master output
Slave output

Figure 14-11 <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.

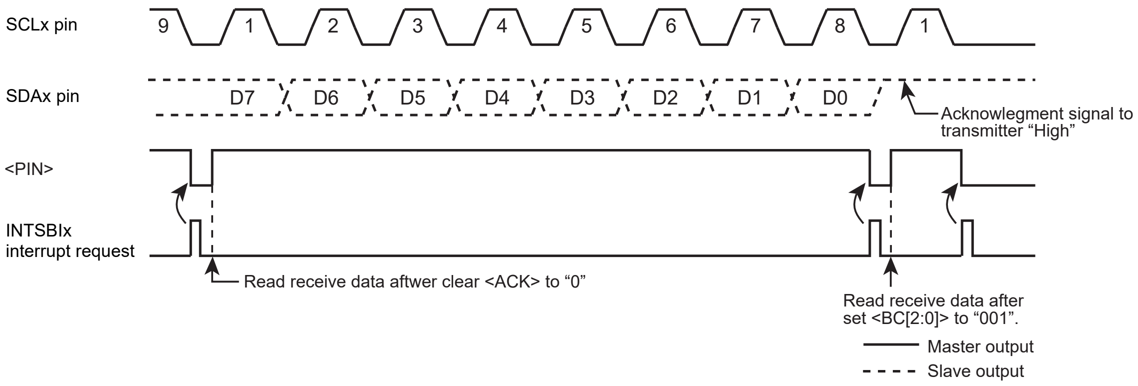

SCLx pin
9
1
2
3
4
5
6
7
8
1
SDAx pin
D7
D6
D5
D4
D3
D2
D1
D0
Acknowlegment signal to transmitter “High”
<PIN>
INTSBIx interrupt request
Read receive data aftwer clear <ACK> to “0”
Read receive data after set <BC[2:0]> to “001”.
Master output
Slave output

Figure 14-12 Terminating Data Transmission in the Master Receiver Mode

Example: When receiving N data word

INTSBIx interrupt (after data transmission)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	X	X	X	X	0	X	X	X	Sets the number of bits of data to be received and specify whether ACK is required.
Reg.	←	SBIxDBR								Reads dummy data.
End of interrupt										

INTSBIx interrupt (first to (N-2)th data reception)

		7	6	5	4	3	2	1	0	
Reg.	←	SBIxDBR								Reads the first to (N-2)th data words.
End of interrupt										

INTSBIx interrupt ((N-1)th data reception)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	X	X	X	0	0	X	X	X	Disables generation of acknowledgement clock.
Reg.	←	SBIxDBR								Reads the (N-1)th data word.
End of interrupt										

INTSBIx interrupt (Nth data reception)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	0	1	0	0	X	X	X	Disables generation of acknowledgement clock.
Reg.	←	SBIxDBR								Reads the Nth data word.
End of interrupt										

INTSBIx interrupt (after completing data reception)

Processing to generate the stop condition.	Terminates the data transmission.
End of interrupt	

Note: X; Don’t care

14.6.3.2 Slave mode (<MST> = "0")

In the slave mode, the SBI generates the INTSBIx interrupt request on four occasions:

1) when the SBI has received any slave address from the master.

2) when the SBI has received a general-call address.

3) when the received slave address matches its address.

4) when a data transfer has been completed in response to a general-call.

Also, if the SBI detects Arbitration Lost in the master mode, it switches to the slave mode.

Upon the completion of data word transfer in which Arbitration Lost is detected, the INTSBIx interrupt request is generated, <PIN> is cleared to "0", and the SCL pin is pulled to the "Low" level.

When data is written to or read from SBIxDBR or when <PIN> is set to "1", the SCLx pin is released after a period of t_{LOW}.

In the slave mode, the normal slave mode processing or the processing as a result of Arbitration Lost is carried out.

SBIxSR<AL>, <TRX>, <AAS> and <ADO> are tested to determine the processing required.

"Table 14-2 Processing in Slave Mode"shows the slave mode states and required processing.

Example: When the received slave address matches the SBI's own address and the direction bit is "1" in the slave receiver mode.

INTSBIx interrupt

```
if TRX = 0
Then go to other processing.
if AL = 0
Then go to other processing.
if AAS = 0
Then go to other processing.
SBIxCR1    ←  X  X  X  1  0  X  X  X    Sets the number of bits to be transmitted.
SBIxDBR    ←  X  X  X  X  X  X  X  X    Sets the transmit data.
```

Note: X; Don’t care

Table 14-2 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 SBIxDBR.
	0	1	0	In the slave receiver mode, the SBI received a slave address with the direction bit "1" transmitted by the master.	
		0	0	In the slave transmitter mode, the SBI has completed a transmission of one data word.	Test LRB. If it has been set to "1", that means the receiver does not require further data. Set <PIN> to 1 and reset <TRX> to 0 to release the bus. If <LRB> has been reset to "0", that means the receiver requires further data. Set the number of bits in the data word to <BC[2:0]> and write the transmit data to the SBIxDBR.
0	1	1	1/0	Arbitration Lost is detected while a slave address is being transmitted, and the SBI receives either a slave address with the direction bit "0" or a general-call address transmitted by another master.	Read the SBIxDBR (a dummy read) to set <PIN> to 1, or write "1" to <PIN>.
		0	0	Arbitration Lost is detected while a slave address or a data word is being transmitted, and the transfer is terminated.	
	0	1	1/0	In the slave receiver mode, the SBI received either a slave address with the direction bit "0" or a general-call address transmitted by the master.	
		0	1/0	In the slave receiver mode, the SBI has completed a reception of a data word.	Set the number of bits in the data word to <BC [2:0].> and read the received data from SBIxDBR.

14.6.4 Generating the Stop Condition

When SBIxSR<BB> is "1", writing "1" to SBIxCR2<MST, TRX, PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus.

Do not alter the contents of <MST, TRX, BB, PIN> until the stop condition appears on the bus.

If another device is holding down the SCL bus line, the SBI waits until the SCL line is released.

After that, the SDA pin goes "High", causing the stop condition to be generated.

		7	6	5	4	3	2	1	0	
SBIxCR2	←	1	1	0	1	1	0	0	0	Generates the stop condition.

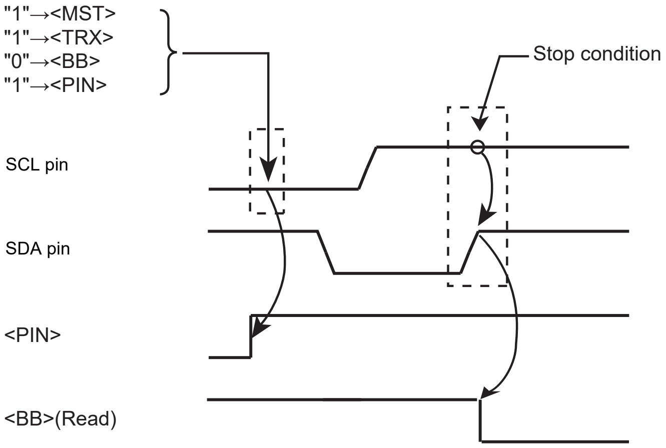


Figure 14-13 Generating the Stop Condition

14.6.5 Restart Procedure

Restart is used when a master device changes the data transfer direction without terminating the transfer to a slave device.The procedure of generating a restart in the master mode is described below.

First, write SBIxCR2<MST, TRX, BB> to "0" and write "1" to <PIN> to release the bus. At this time, the SDAx pin is held at the "High" level and the SCLx pin is released. Because no stop condition is generated on the bus, other devices recognize that the bus is busy.

Then, test SBIxSR<BB> and wait until it becomes "0" to ensure that the SCLx pin is released.

Next, test <LRB> and wait until it becomes "1" to ensure that no other device is pulling the SCLx bus line to the "Low" level.

Once the bus is determined to be free by following the above procedures, follow the procedures described in "14.6.2 Generating the Start Condition and a Slave Address"to generate the start condition.

To satisfy the setup time of restart, at least 4.7μs wait period (in the standard mode) must be created by the software after the bus is determined to be free.

Note 1: **Do not write <MST> to "0" when it is "0". (Restart cannot be initiated.)**

Note 2: **When the master device is acting as a receiver, data transmission from the slave device which serves as a transmitter must be completed before generating a restart. To complete data transfer, slave device must receive a "High" level acknowledge signal. For this reason, <LBR> before generating a restart becomes "1", the rising edge of the SCL line is not detected even <LBR>=**

"1" is confirmed by following the restart procedure. To check the status of the SCL line, read the port.

```
                        7   6   5   4   3   2   1   0
┌→
│  SBIxCR2        ←    0   0   0   1   1   0   0   0      Releases the bus.
└─
   if SBIxSR<BB> ≠ 0                                     Checks that the SCL pin is released.
┌→ Then
└─ if SBIxSR<LRB> ≠ 1                                    Checks that no other device is pulling the SCL pin to the "Low".
   Then
   4.7 μs Wait
   SBIxCR1        ←    X   X   X   1   0   X   X   X      Selects the acknowledgment mode.
   SBIxDBR        ←    X   X   X   X   X   X   X   X      Sets the desired slave address and direction.
   SBIxCR2        ←    1   1   1   1   1   0   0   0      Generates the start condition.
```

Note: X; Don't care

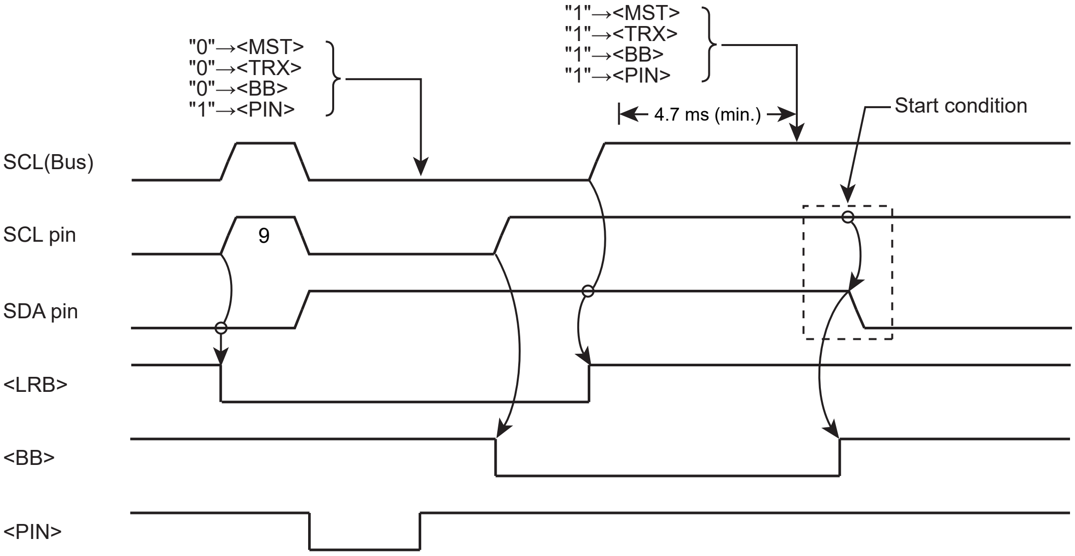


Figure 14-14 Timing Chart of Generating a Restart

14.7 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.

14.7.1 SBIxCR0(control register 0)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SBIEN	R/W	Serial bus interface operation. 0:Disable 1: Enable Enable this bit before using the serial bus interface. If this bit is disabled, power consumption can be reduced because all clocks except SBIxCR0 stop. If the serial bus interface operation is enabled and then disabled, the settings will be maintained in each register.
6-0	-	R	Read as 0.

14.7.2 SBIxCR1(Control register 1)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SIOS	R/W	Transfer Start/Stop 0: Stop 1: Start
6	SIOINH	R/W	Transfer 0: Continue 1: Forced termination
5-4	SIOM[1:0]	R/W	Select transfer mode 00: Transmit mode 01: Reserved 10:Transmit/receive mode 11:Receive mode
3	-	R	Read as 1.
2-0	SCK[2:0]	R/W	On writing <SCK[2:0]>: Select serial clock frequency. (Note 1) 000 n = 3 5 MHz 001 n = 4 2.5 MHz 010 n = 5 1.25 Hz 011 n = 6 625 kHz 100 n = 7 313 kHz 101 n = 8 156 kHz 110 n = 9 78 kHz System clock: fsys (=80MHz) Clock gear: fc/1 Frequency = $\frac{fsys/2}{2^n}$ [Hz] 111 − External clock

Note 1: **After a reset, the <SCK[0]> bit is read as "1". However, if the SIO mode is selected at the SBIxCR2 register, the initial value is read as "0". In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state. The descriptions of the SBIxCR2 register and the SBIxSR register are the same.**

Note 2: **Set <SIOS> to "0" and <SIOINH> to "1" before programming the transfer mode and the serial clock.**

14.7.3 SBIxDBR (Data buffer register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DB[7:0]	R	Receive data
		W	Transmit data

Note 1: **The transmission data must be written in to the register from the MSB (bit 7). The received data is stored in the LSB.**

Note 2: **Since SBIxDBR 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.**

14.7.4 SBIxCR2(Control register 2)

This register serves as SBIxSR register by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-4	-	R	Read as 1. (Note 1)
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note 2) 00: Port mode 01: SIO mode 10: I2Cbus mode 11: Reserved
1-0	-	R	Read as 1. (Note 1)

Note 1: **In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state.**

Note 2: **Make sure that modes are not changed during a communication session.**

14.7.5 SBIxSR (Status Register)

This register serves as SBIxCR2 by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-4	-	R	Read as 1.(Note 1)
3	SIOF	R	Serial transfer status monitor. 0: Completed 1: In progress
2	SEF	R	Shift operation status monitor 0: Completed. 1: In progress
1-0	-	R	Read as 1. (Note 1)

Note: **In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state.**

14.7.6 SBIxBR0 (Baud rate register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
After reset	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	-	R	Read as 1.
6	I2SBI	R/W	Operation in IDLE mode. 0: Stop 1: Operate
5-1	-	R	Read as 1.
0	-	R/W	Make sure to write "0".

14.8 Control in SIO mode

14.8.1 Serial Clock

14.8.1.1 Clock source

Internal or external clocks can be selected by programming SBIxCR1<SCK[2:0]>.

(1) Internal clocks

In the internal clock mode, one of the seven frequencies can be selected as a serial clock, which is output to the outside through the SCKx pin.

At the beginning of a transfer, the SCKx pin output becomes the "High" level.

If the program cannot keep up with this serial clock rate in writing the transmit data or reading the received data, the SBI automatically enters a wait period. During this period, the serial clock is stopped automatically and the next shift operation is suspended until the processing is completed.

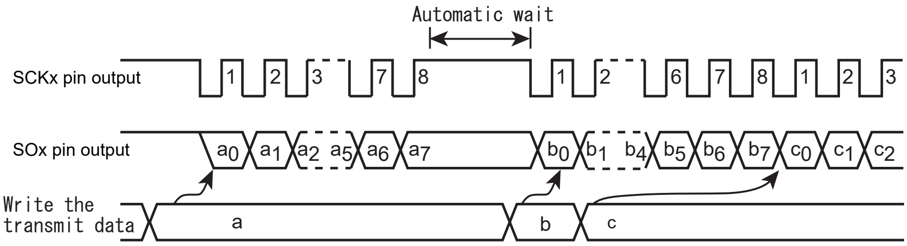

Automatic wait
SCKx pin output
1 2 3 7 8 1 2 6 7 8 1 2 3
SOx pin output
a0 a1 a2 a5 a6 a7 b0 b1 b4 b5 b6 b7 c0 c1 c2
Write the transmit data
a
b
c

Figure 14-15 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.

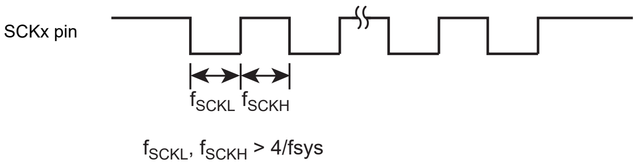

SCKx pin
fSCKL fSCKH
fSCKL, fSCKH > 4/fsys

Figure 14-16 Maximum Transfer Frequency of External Clock Input

14.8.1.2 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).

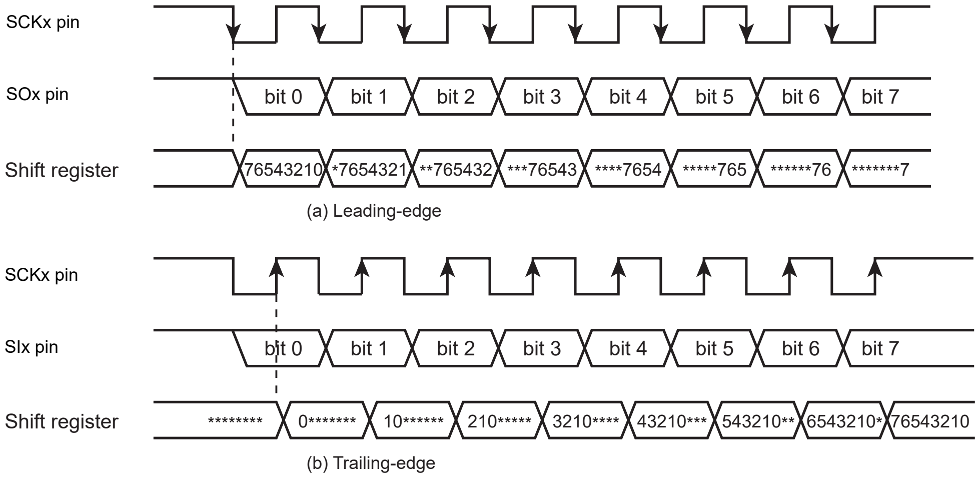

SCKx pin
SOx pin
bit 0
bit 1
bit 2
bit 3
bit 4
bit 5
bit 6
bit 7
Shift register
76543210
*7654321
**765432
***76543
****7654
*****765
******76
*******7
(a) Leading-edge
SCKx pin
SIx pin
bit 0
bit 1
bit 2
bit 3
bit 4
bit 5
bit 6
bit 7
Shift register
****
0*******
10******
210*****
3210****
43210***
543210**
6543210*
76543210
(b) Trailing-edge

Figure 14-17 Shift Edge

14.8.2 Transfer Modes

The transmit mode, the receive mode or the transmit/receive mode can be selected by programming SBIxCR1<SIOM[1:0]>.

14.8.2.1 8-bit transmit mode

Set the control register to the transmit mode and write the transmit data to SBIxDBR.

After writing the transmit data, writing "1" to SBIxCR1<SIOS> starts the transmission. The transmit data is moved from SBIxDBR to a shift register and output to the SO pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the transmit data is transferred to the shift register, SBIxDBR becomes empty, and the INTSBIx (buffer-empty) interrupt is generated, requesting the next transmit data.

In the internal clock mode, the serial clock will be stopped and automatically enter the wait state, if next data is not loaded after the 8-bit data has been fully transmitted. The wait state will be cleared when SBIxDBR is loaded with the next transmit data.

In the external clock mode, SBIxDBR must be loaded with data before the next data shift operation is started. Therefore, the data transfer rate varies depending on the maximum latency between when the interrupt request is generated and when SBIxDBR is loaded with data in the interrupt service program.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting SBIxSR<SIOF> to "1" to the falling edge of SCK.

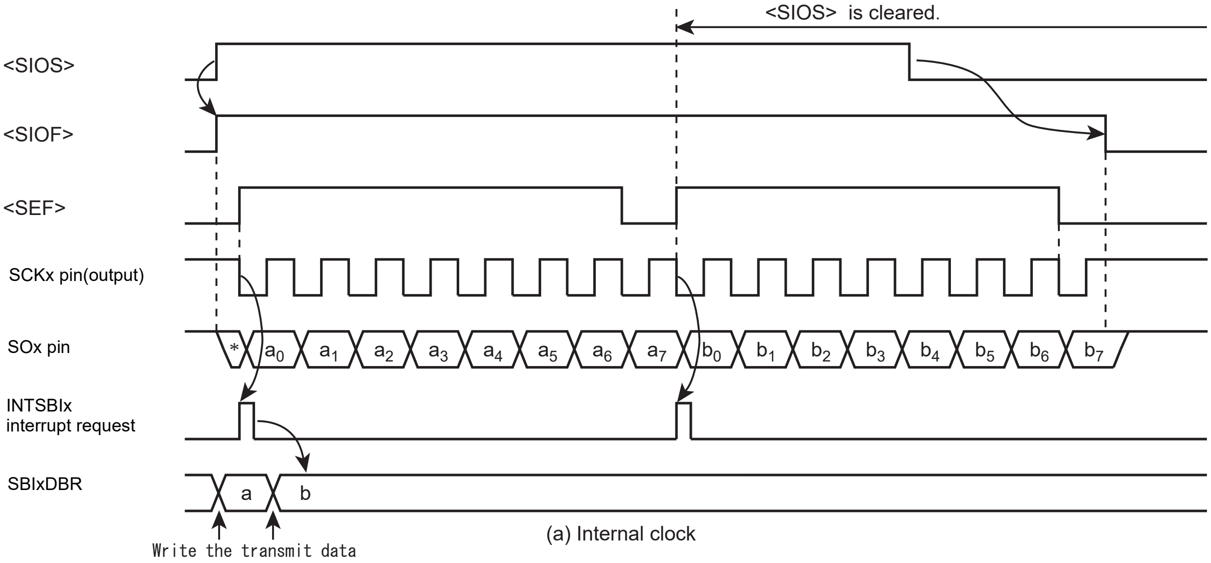
Transmission can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, remaining data is output before transmission ends. The program checks SBIxSR<SIOF> to determine whether transmission has come to an end. <SIOF> is cleared to "0" at the end of transmission. If <SIOINH> is set to "1", the transmission is aborted immediately and <SIOF> is cleared to "0".

When in the external clock mode, <SIOS> must be cleared to "0" before next data shifting. If <SIOS> does not be cleared to "0" before next data shifting, SBI output dummy data and stopped.

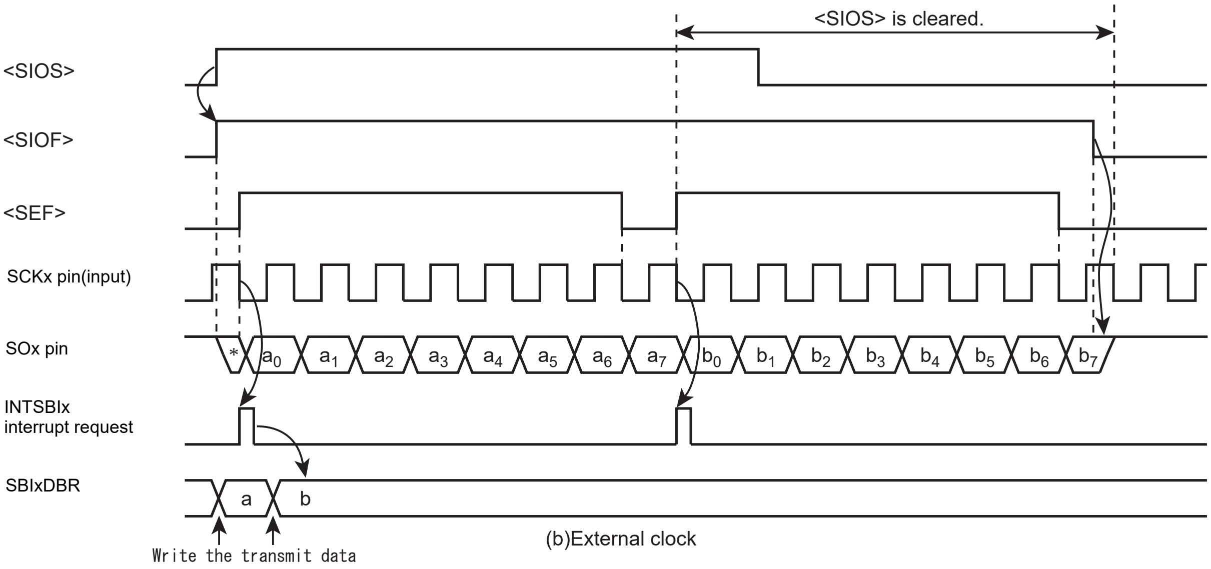
		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	0	0	0	X	X	X	Selects the transmit mode.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBIxCR1	←	1	0	0	0	0	X	X	X	Starts transmission.

INTSBIx interrupt

SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.


<SIOS> is cleared.
<SIOS>
<SIOF>
<SEF>
SCKx pin(output)
SOx pin
*
a0 a1 a2 a3 a4 a5 a6 a7
b0 b1 b2 b3 b4 b5 b6 b7
INTSBIx interrupt request
SBIxDBR
a
b
Write the transmit data

(a) Internal clock


<SIOS> is cleared.
<SIOS>
<SIOF>
<SEF>
SCKx pin(input)
SOx pin
*
a0 a1 a2 a3 a4 a5 a6 a7
b0 b1 b2 b3 b4 b5 b6 b7
INTSBIx interrupt request
SBIxDBR
a
b
Write the transmit data

(b)External clock

Figure 14-18 Transmit Mode

Example: Example of programming (external clock) to terminate transmission by <SIO>

		7	6	5	4	3	2	1	0	
if SBIxSR<SIOF> ≠ 0										Recognizes the completion of the transmission.
Then										
if SCK ≠ 1										Recognizes "1" is set to the SCK pin by monitoring the port.
Then										
SBIxCR1	←	0	0	0	0	0	1	1	1	Completes the transmission by setting <SIOS> = 0.

14.8.2.2 8-bit receive mode

Set the control register to the receive mode. Then writing "1" to SBIxCR1<SIOS> enables reception.Data is taken into the shift register from the SI pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIxDBR and the INTSBIx (buffer-full) interrupt request is generated to request reading the received data. The interrupt service program then reads the received data from SBIxDBR.

In the internal clock mode, the serial clock will be stopped and automatically be in the wait state until the received data is read from SBIxDBR.

In the external clock mode, shift operations are executed in synchronization with the external clock. The maximum data transfer rate varies, depending on the maximum latency between generating the interrupt request and reading the received data

Reception can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, reception continues until all the bits of received data are written to SBIxDBR. The program checks SBIxSR<SIOF> to determine whether reception has come to an end.<SIOF> is cleared to "0" at the end of reception. After confirming the completion of the reception, last received data is read. If <SIOINH> is set to "1", the reception is aborted immediately and <SIOF> is cleared to "0". (The received data becomes invalid, and there is no need to read it out.)

Note: **The contents of SBIxDBR will not be retained after the transfer mode is changed. The ongoing reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.**

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	1	1	0	X	X	X	Selects the receive mode.
SBIxCR1	←	1	0	1	1	0	X	X	X	Starts reception.

INTSBIx interrupt

Reg.	←	SBIxDBR	Reads the received data.

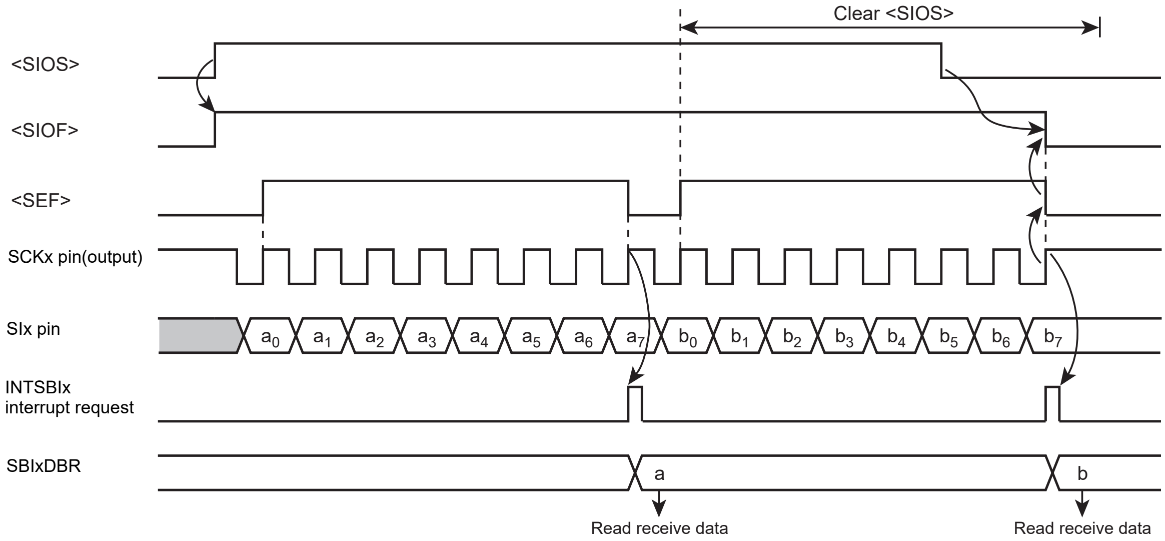


Figure 14-19 Receive Mode (Example: Internal Clock)

14.8.2.3 8-bit transmit/receive mode

Set the control register to the transfer/receive mode. Then writing the transmit data to SBIxDBR and setting SBIxCR1<SIOS> to "1" enables transmission and reception.The transmit data is output through the SOx pin at the falling of the serial clock, and the received data is taken in through the SI pin at the rising of the serial clock, with the least-significant bit (LSB) first. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIxDBR and the INTSBIx interrupt request is generated.The interrupt service program reads the received data from the data buffer register and writes the next transmit data. Because SBIxDBR is shared between transmit and receive operations, the received data must be read before the next transmit data is written.

In the internal clock operation, the serial clock will be automatically in the wait state until the received data is read and the next transmit data is written.

In the external clock mode, shift operations are executed in synchronization with the external serial clock. Therefore, the received data must be read and the next transmit data must be written before the next shift operation is started.The maximum data transfer rate for the external clock operation varies depending on the maximum latency between when the interrupt request is generated and when the transmit data is written.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting <SIOF> to "1" to the falling edge of SCK.

Transmission and reception can be terminated by clearing <SIOS> to "0" or setting SBIxCR1<SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, transmission and reception continue until the received data is fully transferred to SBIxDBR. The program checks SBIxSR<SIOF> to determine whether transmission and reception have come to an end. <SIOF> is cleared to "0" at the end of transmission and reception.If <SIOINH> is set to "1", the transmission and reception is aborted immediately and <SIOF> is cleared to "0".

Note: **The contents of SBIxDBR will not be retained after the transfer mode is changed. The ongoing transmission and reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.**

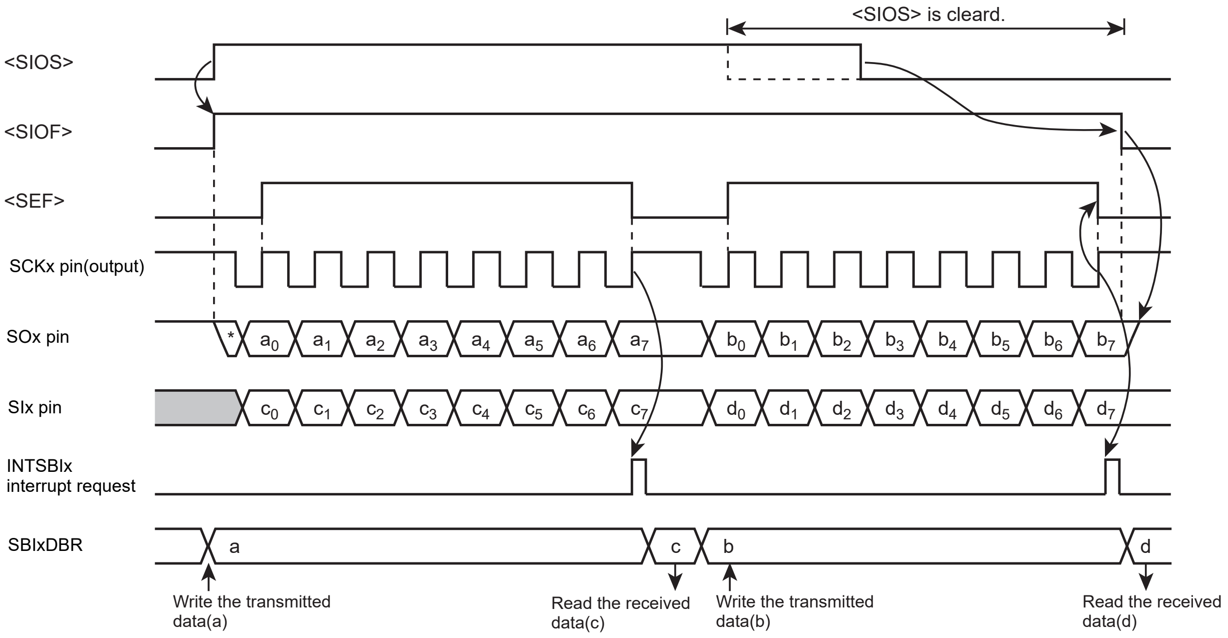

<SIOS> is cleard.
<SIOS>
<SIOF>
<SEF>
SCKx pin(output)
SOx pin
*
a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7
b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7
SIx pin
c_0 c_1 c_2 c_3 c_4 c_5 c_6 c_7
d_0 d_1 d_2 d_3 d_4 d_5 d_6 d_7
INTSBIx interrupt request
SBIxDBR
a
c
b
d
Write the transmitted data(a)
Read the received data(c)
Write the transmitted data(b)
Read the received data(d)

Figure 14-20 Transmit/Receive Mode (Example: Internal Clock)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	1	0	0	X	X	X	Selects the transmit mode.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBIxCR1	←	1	0	1	0	0	X	X	X	Starts reception/transmission.

INTSBIx interrupt

Reg.	←	SBIxDBR								Reads the received data.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

14.8.2.4 Data retention time of the last bit at the end of transmission

Under the condition SBIxCR1<SIOS>= "0", the last bit of the transmitted data retains the data of SCK rising edge as shown below. Transmit mode and transmit/receive mode are the same.

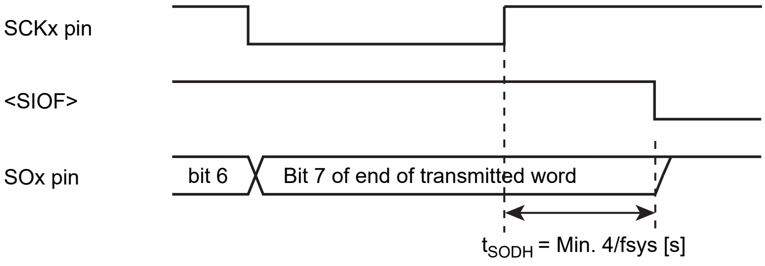

SCKx pin
<SIOF>
SOx pin
bit 6
Bit 7 of end of transmitted word
t_{SODH} = Min. 4/fsys [s]

Figure 14-21 Data retention time of the last bit at the end of transmission

15. Synchronous Serial Port (SSP)

15.1 Overview

This LSI contains the SSP (Synchronous Serial Port) with 3 channels. These channels have 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 to fsys/65024
	In slave mode	fsys/12 to fsys/65024
DMA		Supported
Internal test function		The internal loopback test mode is available.
Control pin (x = 0 to 2)		SPxCLK,SPxFSS,SPxDO,SPxDI

15.2 Block Diagram

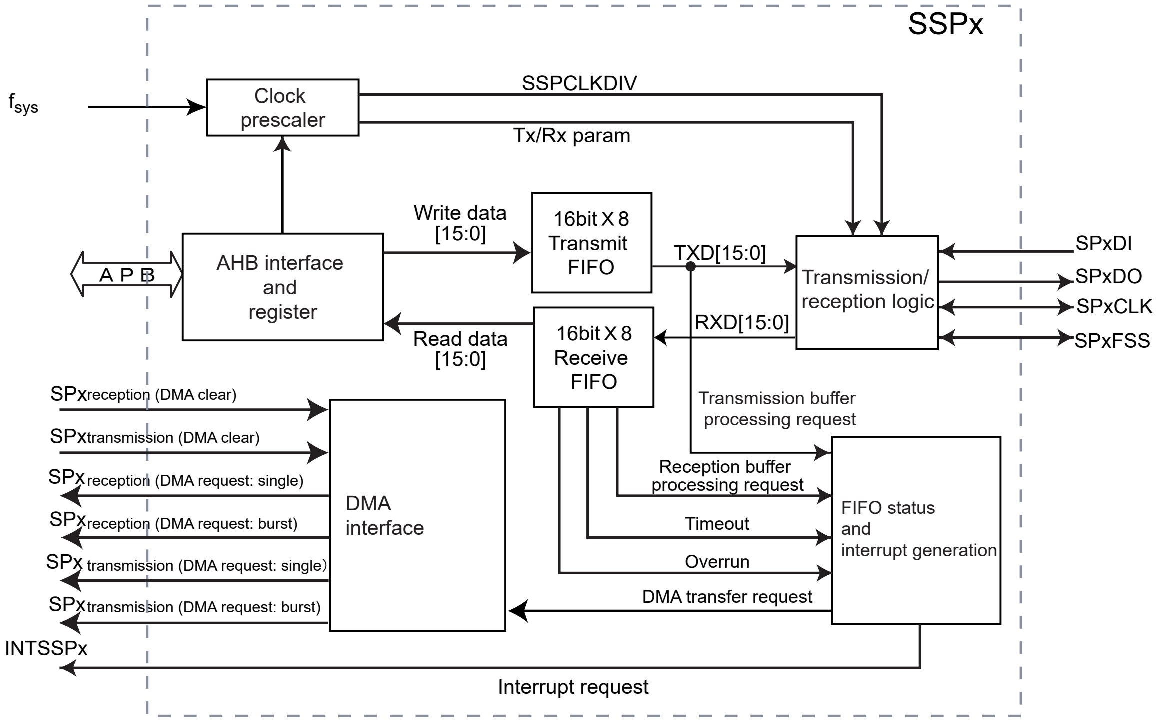

SSPx
f_{sys}
Clock prescaler
SSPCLKDIV
Tx/Rx param
Write data [15:0]
16bit X 8 Transmit FIFO
TXD[15:0]
APB
AHB interface and register
Transmission/ reception logic
SPxDI
SPxDO
SPxCLK
SPxFSS
Read data [15:0]
16bit X 8 Receive FIFO
RXD[15:0]
SPxreception (DMA clear)
SPxtransmission (DMA clear)
SPx reception (DMA request: single)
SPxreception (DMA request: burst)
SPx transmission (DMA request: single)
SPxtransmission (DMA request: burst)
DMA interface
Transmission buffer processing request
Reception buffer processing request
Timeout
Overrun
DMA transfer request
FIFO status and interrupt generation
INTSSPx
Interrupt request

Figure 15-1 SSP block diagram

15.3 Register

15.3.1 Register List

Channel x	Base Address
Channel0	0x4004_0000
Channel1	0x4004_1000
Channel2	0x4004_2000

Register Name(x=0~2)		Address (Base+)
Control register 0	SSPxCR0	0x0000
Control register 1	SSPxCR1	0x0004
Receive FIFO (read) and transmit FIFO (write) data register	SSPxDR	0x0008
Status register	SSPxSR	0x000C
Clock prescale register	SSPxCPSR	0x0010
Interrupt enable/disable register	SSPxIMSC	0x0014
Pre-enable interrupt status register	SSPxRIS	0x0018
Post-enable interrupt status register	SSPxMIS	0x001C
Interrupt clear register	SSPxICR	0x0020
DMA control register	SSPxDMACR	0x0024
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.

15.3.2 SSPxCR0(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 SSPxCPSR register, and <SCR> takes a value between 0 to 255.</td></tr>
<tr><td>7</td><td>SPH</td><td>R/W</td><td>SPxCLK 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>SPxCLK polarity:
0:SPxCLK is in Low state.
1:SPxCLK 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 SSPxCR0<SCR[7:0]> = 0x00 , SSPxCPSR<CPSDVSR[7:0]> = 0x02, when slave mode is selected.

15.3.3 SSPxCR1(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 SPxDO 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

15.3.4 SSPxDR(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.

15.3.5 SSPxSR(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.

15.3.6 SSPxCPSR (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 SSPxCR0<SCR[7:0]> = 0x00 , SSPxCPSR<CPSDVSR[7:0]> = 0x02, when slave mode is selected.

15.3.7 SSPxIMSC (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.

15.3.8 SSPxRIS (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

15.3.9 SSPxMIS (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

15.3.10 SSPxICR (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

15.3.11 SSPxDMACR (DMA control 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	-	-	-	-	-	-	TXDMAE	RXDMAE
After Reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Function
31-2	−	W	Write as "0".
1	TXDMAE	R/W	Transmit FIFO DMA control: 0:Disable 1:Enable
0	RXDMAE	R/W	Transmit FIFO DMA control: 0:Disable 1:Enable

15.4 Overview of SSP

This LSI contains the SSP with 3channels.

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 SPxDO and received via SPxDI.

The SSP contains a programmable prescaler to generate the serial output clock SPxCLK from the input clock fsys. The operation mode, frame format, and data size of the SSP are programmed in the control registers SSPxCR0 and SSPxCR1.

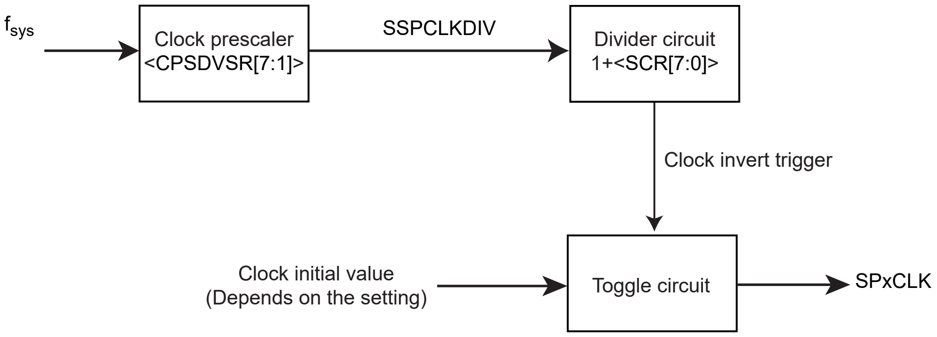
15.4.1 Clock prescaler

When configured as a master, a clock prescaler comprising two free-running serially linked counters is used to provide the serial output clock SPxCLK.

You can program the clock prescaler through the SSPxCPSR register, to divide fsys by a factor of 2 to 254 in steps of two. Because the least significant bit of the SSPxCPSR 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 SSPxCR0 register, to give the master output clock SPxCLK.

Bitrate = fsys / (<CPSDVSR>×(1+<SCR>))



15.4.2 Transmit FIFO

This is a 16-bit wide, 8-layered transmit FIFO buffer, which is shared in master and slave modes.

15.4.3 Receive FIFO

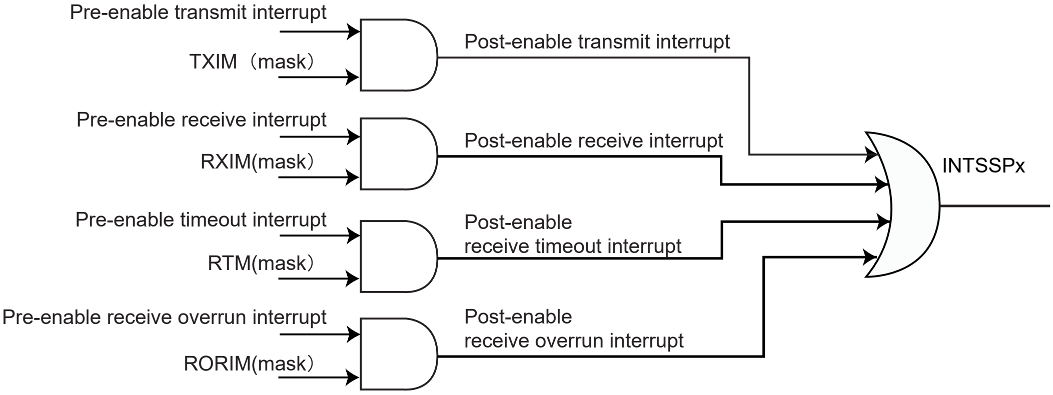
This is a 16-bit wide 8-layered receive FIFO buffer, which is shared in master and slave modes.

15.4.4 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 INTSSPx 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
INTSSPx

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 (SSPxCR1 <SSE> = "0").

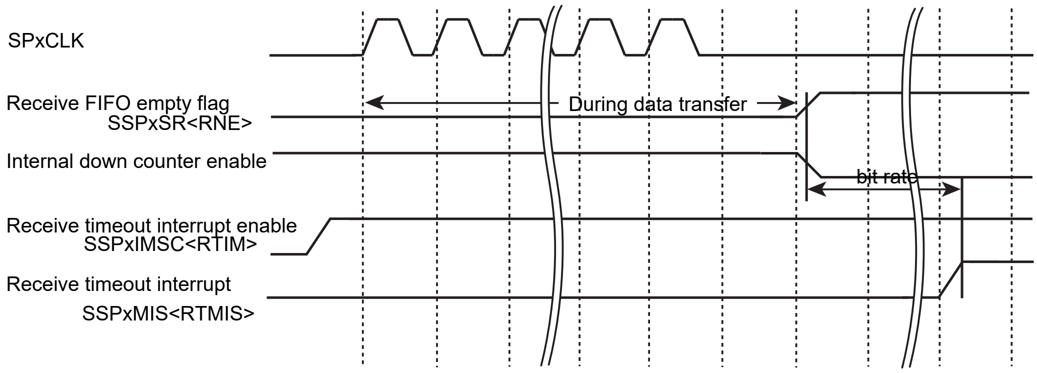
The first transmitted data can be written in the FIFO by using this interrupt.

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.


SPxCLK
Receive FIFO empty flag
SSPxSR<RNE>
Internal down counter enable
Receive timeout interrupt enable
SSPxIMSC<RTIM>
Receive timeout interrupt
SSPxMIS<RTMIS>
During data transfer
bit rate

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 SSPxICR<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.

15.4.5 DMA interface

The DMA operation of the SSP is controlled through SSPxDMACR register.

When there are more data than the watermark level (half of the FIFO) in the receive FIFO, the receive DMA request is asserted.

When the amount of data left in the transmit FIFO is less than the watermark level (half of the FIFO), the transmit DMA request is asserted.

To clear the transmit/receive DMA request, an input pin for the transmit/receive DMA request clear signals, which are asserted by the DMA controller, is provided.

Set the DMA burst length to four words.

Note: For the remaining three words, the SSP does not assert the burst request.

Each request signal remains asserted until the relevant DMA clear signal is asserted. After the request clear signal is deasserted, a request signal can become active again, depending on the conditions described above. All request signals are deasserted if the SSP is disabled or the DMA enable signal is cleared.

The following table shows the trigger points for DMABREQ, for both the transmit and receive FIFOs.

	Burst length	
Watermark level	Transmit (number of empty locations)	Receive (number of filled locations)
1/2	4	4

15.5 SSP operation

15.5.1 Initial setting for SSP

Settings for the SSP communication protocol must be made with the SSP disabled.

Control registers SSPxCR0 and SSPxCR1 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 SSPxCPSR and SSPxCR0 <SCR>.

This SSP supports the following protocols:

- SPI
- SSI
- Microwire

15.5.2 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 SPxFSS 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.

15.5.3 Clock ratios

When setting a frequency for fsys , the following conditions must be met.

- In master mode

  f_{SPxCLK} (maximum) → f_{sys} /2

  f_{SPxCLK} (minimum) → f_{sys} /(254×256)
- In slave mode

  f_{SPxCLK} (maximum) → f_{sys} /12

  f_{SPxCLK} (minimum) → f_{sys} /(254×256)

15.6 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 (SPxCLK)

  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 (SPxFSS)

  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 SPxCLK and the input data is received at its falling edge.

Refer to Section "15.6.1" to "15.6.3" for details of each frame format.

15.6.1 SSI frame format

In this mode, the SSP is in idle state, SPxCLK and SPxFSS are forcedly set to "Low", and the transmit data line SPxDO becomes Hi-Z. When data is written in the transmit FIFO, the master outputs "High" pulses of 1 SPxCLK to the SPxFSS 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 SPxDO pin at the next rising edge of SPxCLK.

Likewise, the received data will be input starting from the MSB to the SPxDI pin at the falling edge of SPxCLK. The received data will be transferred from the serial shift register into the receive FIFO at the rising edge of SPxCLK after its LSB data is latched.

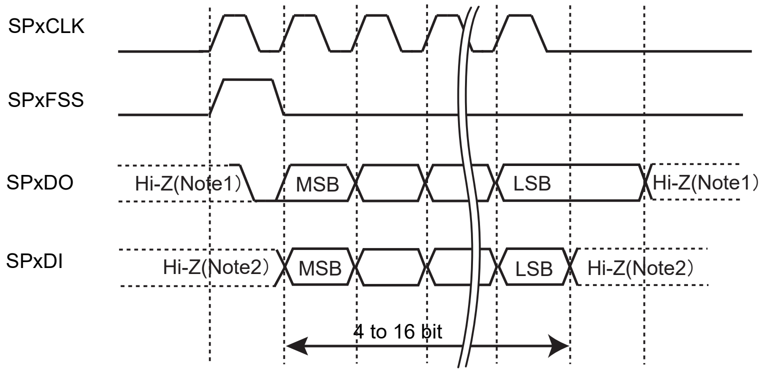

SPxCLK
SPxFSS
SPxDO
Hi-Z(Note1)
MSB
LSB
Hi-Z(Note1)
SPxDI
Hi-Z(Note2)
MSB
LSB
Hi-Z(Note2)
4 to 16 bit

Figure 15-2 SSI frame format (transmission/reception during single transfer)

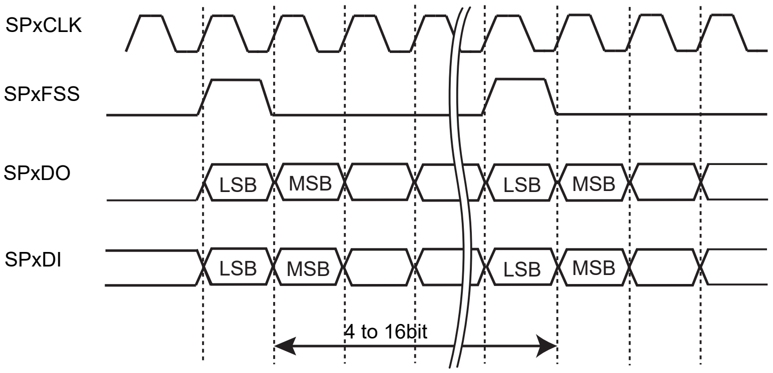

SPxCLK
SPxFSS
SPxDO
LSB
MSB
LSB
MSB
SPxDI
LSB
MSB
LSB
MSB
4 to 16bit

Figure 15-3 SSI frame format (transmission/reception during continuous transfer)

Note 1: When transmission is disable, SPxDO 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: SPxDI 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.

15.6.2 SPI frame format

The SPI interface has 4 lines. SPxFSS is used for slave selection. One of the main features of the SPI format is that the <SPO> and <SPH> bits in the SSPxCR0 register can be used to set the SPxCLK operation timing.

SSPxCR0 <SPO> is used to set the level at which SPxCLK in idle state is held.

SSPxCR0 <SPH> is used to select the clock edge at which data is latched.

	SSPxCR0<SPO>	SSPxCR0<SPH>
0	"Low" state	Capture data at the 1st clock edge.
1	"High" state	Capture data at the 2nd clock edge.

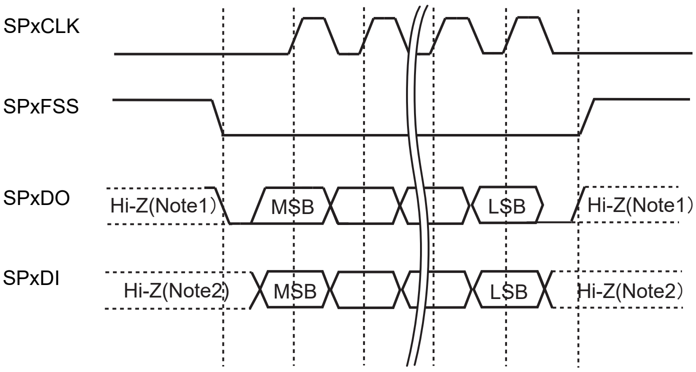

SPxCLK
SPxFSS
SPxDO
SPxDI
Hi-Z(Note1)
MSB
LSB
Hi-Z(Note1)
Hi-Z(Note2)
MSB
LSB
Hi-Z(Note2)

Figure 15-4 SPI frame format (single transfer, <SPO>="0" & <SPH>="0")

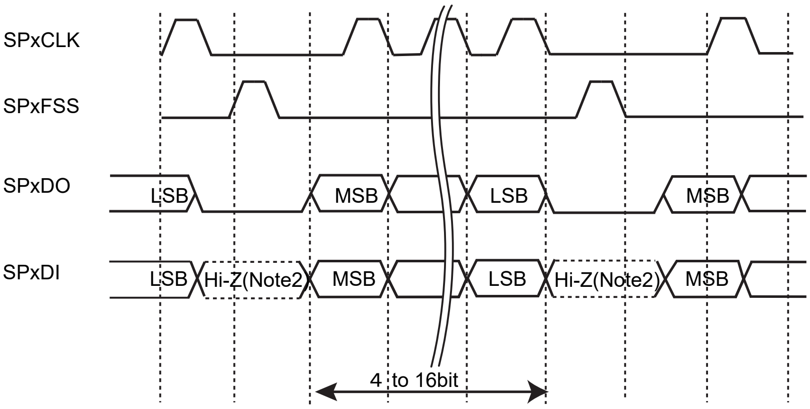

SPxCLK
SPxFSS
SPxDO
SPxDI
LSB
MSB
LSB
MSB
LSB
Hi-Z(Note2)
MSB
LSB
Hi-Z(Note2)
MSB
4 to 16bit

Figure 15-5 SPI frame format (continuous transfer,<SPO>="0" & <SPH>="0")

Note 1: When transmission is disable, SPxDO 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: SPxDI 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 SPxCLK signal is set to "Low".
- SPxFSS is set to "High".
- The transmit data line SPxDO is set to "Low".

If the SSP is enabled and valid data exists in the transmit FIFO, the SPxFSS master signal driven by "Low" notifies of the start of transmission. This enables the slave data in the SPxDI input line of the master.

When a half of the SPxCLK period has passed, valid master data is transferred to the SPxDO pin. Both the master data and slave data are now set. When another half of SPxCLK has passed, the SPxCLK master clock pin becomes "High". After that, the data is captured at the rising edge of the SPxCLK signal and transmitted at its falling edge.

In the single transfer, the SPxFSS line will return to the idle "High" state when all the bits of that data word have been transferred, and then one cycle of SPxCLK has passed after the last bit was captured.

However, for continuous transfer, the SPxFSS 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 SPxFSS pin of the slave device between individual data transfers. When the continuous transfer is completed, the SPxFSS pin will return to the idle state when one cycle of SPxCLK has passed after the last bit is captured.

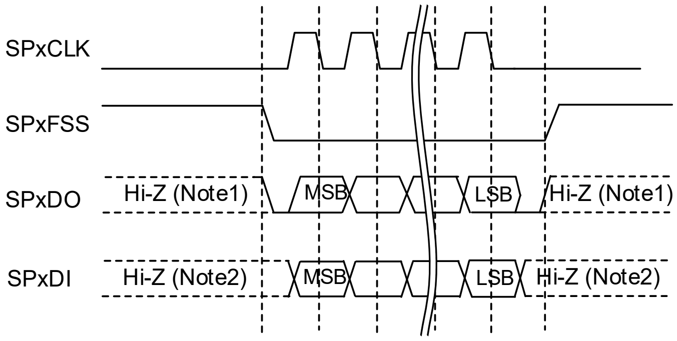


Figure 15-6 SPI frame format (Single & continuous transfer,<SPO>="0" & <SPH>="1")

Figure 15-6 show the SPI frame format with <SPO>=0 & <SPH>=1,which is covers both single and continuous transfer.

Note 1: When transmission is disable, SPxDO 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: SPxDI 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.

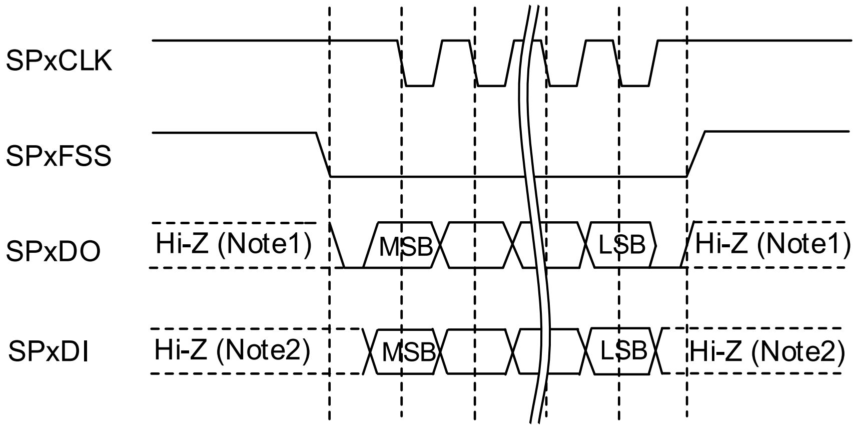

SPxCLK
SPxFSS
SPxDO
SPxDI
Hi-Z (Note1)
MSB
LSB
Hi-Z (Note1)
Hi-Z (Note2)
MSB
LSB
Hi-Z (Note2)

Figure 15-7 SPI frame format (single transfer,<SPO>="1" & <SPH>="0")

Figure 15-7 shows the SPI frame format with <SPO>=1 &<SPH>=0, which is for single transmission signal sequence.

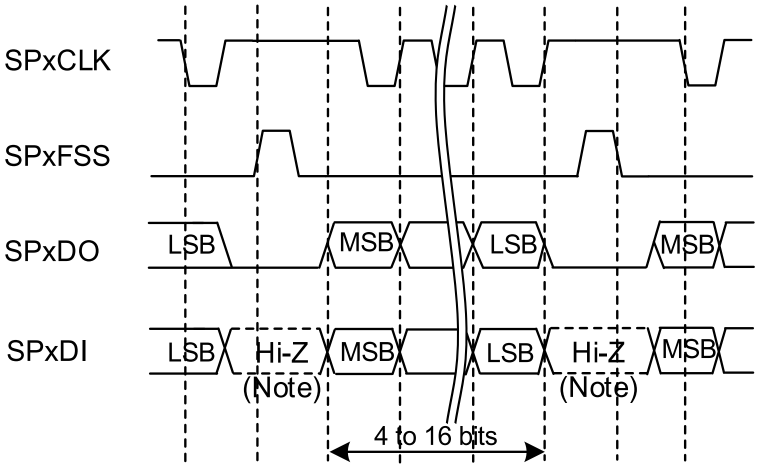

SPxCLK
SPxFSS
SPxDO
SPxDI
LSB
MSB
LSB
MSB
LSB
Hi-Z (Note)
MSB
LSB
Hi-Z (Note)
MSB
4 to 16 bits

Figure 15-8 SPI frame format (continuous transfer,<SPO>="1" & <SPH>="0")

Figure 15-8 shows the SPI frame format with <SPO>=1 &<SPH>=0, which is for continuous transmission signal sequence.

Note 1: When transmission is disable, SPxDO 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: SPxDI 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.

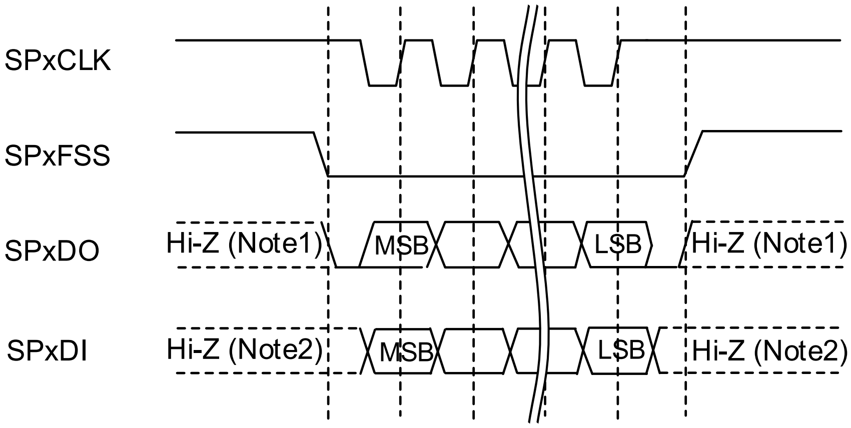

SPxCLK
SPxFSS
SPxDO
Hi-Z (Note1)
MSB
LSB
Hi-Z (Note1)
SPxDI
Hi-Z (Note2)
MSB
LSB
Hi-Z (Note2)

Figure 15-9 SPI frame format (Single & continuous transfer,<SPO>="1" & <SPH>="1")

Figure 15-9 show the SPI frame format with <SPO>=1 & <SPH>=1,which is covers both single and continuous transfer.

Note 1: When transmission is disable, SPxDO 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: SPxDI 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.

15.6.3 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.

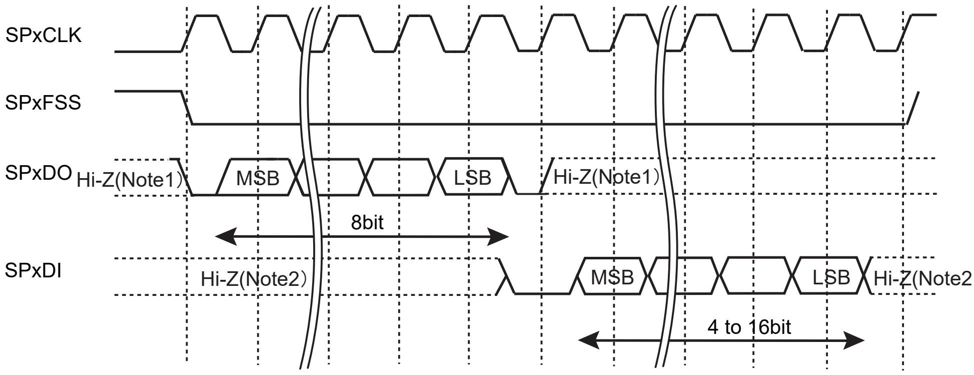


Figure 15-10 Microwire frame format (single transfer)

Note 1: When transmission is disabled, SPxDO 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: SPxDI 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 SPxCLK signal is set to "Low".
- SPxFSS is set to "High".
- The transmit data line SPxDO is set to "Low".

A transmission is triggered by writing a control byte to the transmit FIFO. The falling edge of SPxFSS 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 SPxDO pin.

SPxFSS remains "Low" and the SPxDI 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 SPxCLK.

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 SPxDI line on the falling edge of SPxCLK.

The SSP in turn latches each bit on the rising edge of SPxCLK. At the end of the frame, for single transfers, the SPxFSS 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 SPxCLK after the LSB has been latched by the receive shifter, or when the SPxFSS pin goes "High".

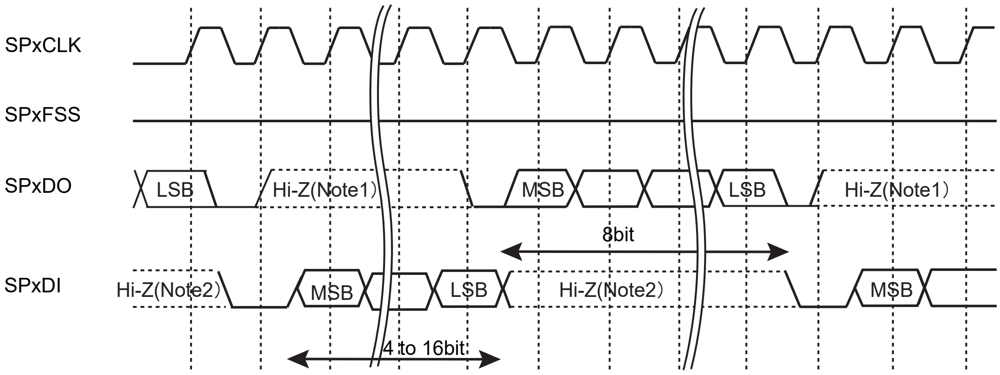


Figure 15-11 Microwire frame format (continuous transfer)

Note 1: When transmission is disabled, SPxDO 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: SPxDI 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 SPxFSS 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 SPxCLK, 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.

16. USB Clock Control (USBPLLIF)

16.1 Features

The USBPLLIF block enables to set to enable/disable PLL for USB device, to stop/operate a 48MHz clock for USB device and to select a clock for USB among PLL or USB_ECLK input.

When the USB function is not used, power consumption can be reduced by stopping each selection register.

16.2 Registers

16.2.1 Register List

The following table shows the PLL-related registers for USB and addresses.

Base Address = 0x400F_3100

Register name		Address (Base+)
PLL control register for USB	USBPLLCR	0x0000
PLL enable register for USB	USBPLLEN	0x0004
PLL select register for USB	USBPLLSEL	0x0008

16.2.2 USBPLLCR(PLL system control register for USB)

	31	30	29	28	27	26	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	-	-	-	-	-	-	-	USBPLLON
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	R	Reads as "0".
0	USBPLLON	R/W	Chooses USBPLL (multiplier circuit) operation 0: Stop 1: Oscillation After reset, the bit status is stop and is required to set.

16.2.3 USBPLLEN (PLL enable register for USB)

	31	30	29	28	27	26	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	-	-	-	-	-	-	-	USBDEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Reads as "0".
1	−	RW	Writes as "0".
0	USBDEN	RW	Feeds 48MHz clock for USB device 0: Stop 1: Operation After reset, the bit status is stop and is required to set.

16.2.4 USBPLLSEL (PLL select register for USB)

	31	30	29	28	27	26	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	USBPLLSET							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	USBPLLSET							USBPLLSEL
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Reads as "0".
15-1	USBPLLSET [14:0]	R/W	Chooses a value of PLL multiplier for USB (Other than the below values are prohibited.) When USBCLK is selected using 0x0000:<USBPLLSEL> 0x5917:Input clock 8MHz, output clock 48MHz (6 times) 0x5A0F:Input clock 12MHz, output clock 48MHz (4 times) 0x720B:Input clock 16MHz, output clock 48MHz (3 times)
0	USBPLLSEL	R/W	Chooses a clock for USB device 0: USB_ECLK input 1: fusbpll After reset, an external USB clock is set to " USB_ECLK input ". If PLL clock for USB is used, set to "1".

Note 1: When a multiplying value of PLL is set, set USBPLLCR<USBPLLON> = "0" (PLL stops).

Note 2: After a multiplying value of PLL is set, PLL requires a time 100μs or more as initial stable time. Maintain the condition of USBPLLCR<USBPLLON>="0" (PLL stops) during the initial stable time.

Note 3: When returning from STOP1/2 mode, <USBPLLSEL> and <USBPLLON> are initialized.

Note 4: When an internal high-speed oscillator (IHOSC) is used as the system clock, do not use PLL for USB.

16.3 USB Clock Control

16.3.1 USB Clock Type

The following table shows a type of USB clock.

EHCLKIN : External high-speed oscillation clock from X1 (clock inputs)
EHOSC : External high-speed oscillation clock fromX1 and X2 (oscillator connected)
feosc : Selected clock with CGOSCCR<EHOSCSEL>
fusbpll : Multiplied clock by PLL for USB
fusbclk : Multiplied clock by PLL for USB or USB_ECLK input
USBDCLK : 48MHz clock for USB device

16.3.2 Initial Values after Reset

Reset operation initializes the clock configuration as follows.

External high-speed oscillator : Stop
PLL for USB (multiplier) : Stop

16.3.3 USB System Diagram

Figure 16-1 shows a USB system diagram.

Among clocks inputting to the selector, an input with arrows described in the figure is chosen as the initial state after reset.

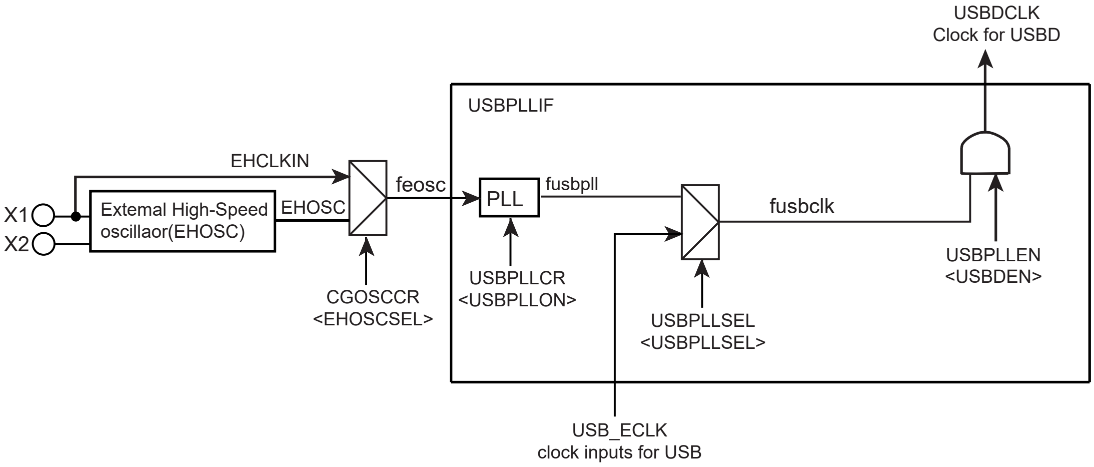

USBDCLK
Clock for USBD
USBPLLIF
EHCLKIN
feosc
fusbpll
PLL
fusbclk
X1
X2
Extemal High-Speed oscillaor(EHOSC)
EHOSC
CGOSCCR <EHOSCSEL>
USBPLLCR <USBPLLON>
USBPLLSEL <USBPLLSEL>
USBPLLEN <USBDEN>
USB_ECLK
clock inputs for USB

Figure 16-1 USB system diagram

16.3.4 Clock Multiplier circuit for USB (PLL for USB)

16.3.4.1 How to use

The PLL for USB is disabled after reset. To use the PLL, set as follows.

Set a multiplied value of USBPLLSEL<USBPLLSET> when USBPLLCR<PLLON> is "0" and wait for equal or more than approximate 100μs as the USB PLL stability time.

After this, set "1" to USBPLLCR<USBPLLON>.

After it takes equal or more than approximate 100μs as the lockup time, set "1" to USBPLLEN<USBDEN>. This generates a clock multiplied by feosc for USB clock.

Stable time is required until the PLL operation becomes stable using the warming-up function.

Note 1: When PLL starts operation, stable time is required approximately 100μs.
Note 2: For warming-up time setting, refer to Section "Clock / Mode control".

16.3.5 External Clock Input for USB

16.3.5.1 How to use

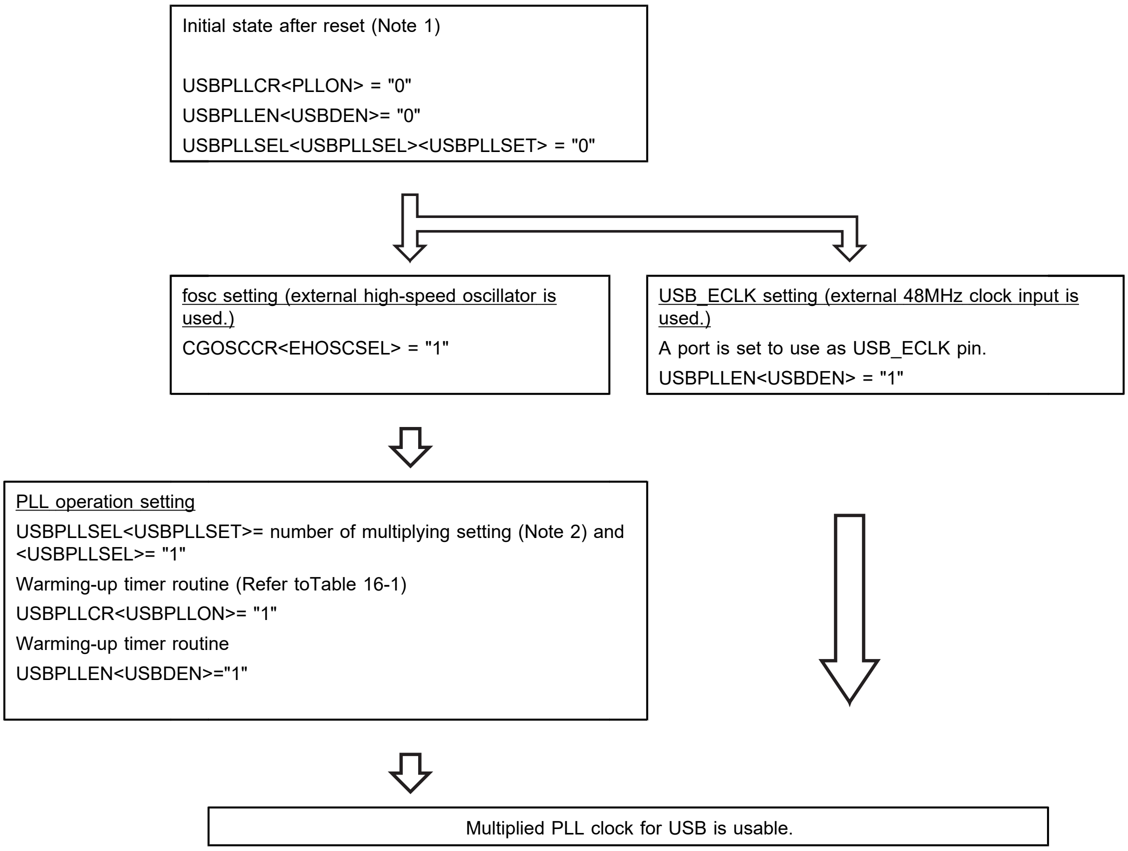
After reset, USB_ECLK pin is used as port. After setting the register of port to use port as USB_ECLK pin, setting "1" to USBPLLEN<USBDEN>, feeding clock for device is enabled.

Note: When an external clock for USB is used, PLL for USB is prohibited.

16.3.6 A Clock setting sequence for USB

The following flowchart shows a clock setting sequence for USB after reset.

Clock for USB recognition sequence



Initial state after reset (Note 1)

USBPLLCR<PLLON> = "0"
USBPLLEN<USBDEN>= "0"
USBPLLSEL<USBPLLSEL><USBPLLSET> = "0"

fosc setting (external high-speed oscillator is used.)
CGOSCCR<EHOSCSEL> = "1"

USB_ECLK setting (external 48MHz clock input is used.)
A port is set to use as USB_ECLK pin.
USBPLLEN<USBDEN> = "1"

PLL operation setting
USBPLLSEL<USBPLLSET>= number of multiplying setting (Note 2) and <USBPLLSEL>= "1"
Warming-up timer routine (Refer toTable 16-1)
USBPLLCR<USBPLLON>= "1"
Warming-up timer routine
USBPLLEN<USBDEN>="1"

Multiplied PLL clock for USB is usable.

Note 1: High-speed oscillator and supply voltage must be stable.

Note 2: When the number of PLL multiplier is changed, stable time is required 100μs or more as initial stable time. During this period, maintains the condition of USBPLLCR<USBPLLON>= "0"(PLL stops).

Table 16-1 Warming-up function setting

CGOSCCR<WUPT[11:0]> = "xxxx"	: Warming-up time setting
↻ Reads CGOSCCR<WUPT[11:0]>	: Check if warming-up time is reflected. Repeat until "xxxx" is read.
CGOSCCR<XEN2> = "1"	: High-speed oscillator (fosc) is enabled.
CGOSCCR<WUEON> = "1"	: Start warming-up timer (WUP)
Reads CGOSCCR<WUEF>	: Waits until "0" is read. (WUP is finished)

17. USB Device Controller (USBD)

This section describes about USB device controller.

An endpoint is described as EP in this section.

17.1 Outline

1. Conforming to Universal Serial Bus Specification Rev.2.0.
2. Supports Full-Speed (Low-Speed is not supported).
3. USB protocol processing
4. Detects SOF/USB_RESET/SUSPEND/RESUME.
5. Generates and checks packet IDs.
6. Checks CRC5. Generates and checks CRC16.
7. Supports 4 transfer modes (Control/Interrupt /Bulk/ Isochronous).
8. Supports 8 EPs.

Table 17-1 Endpoints

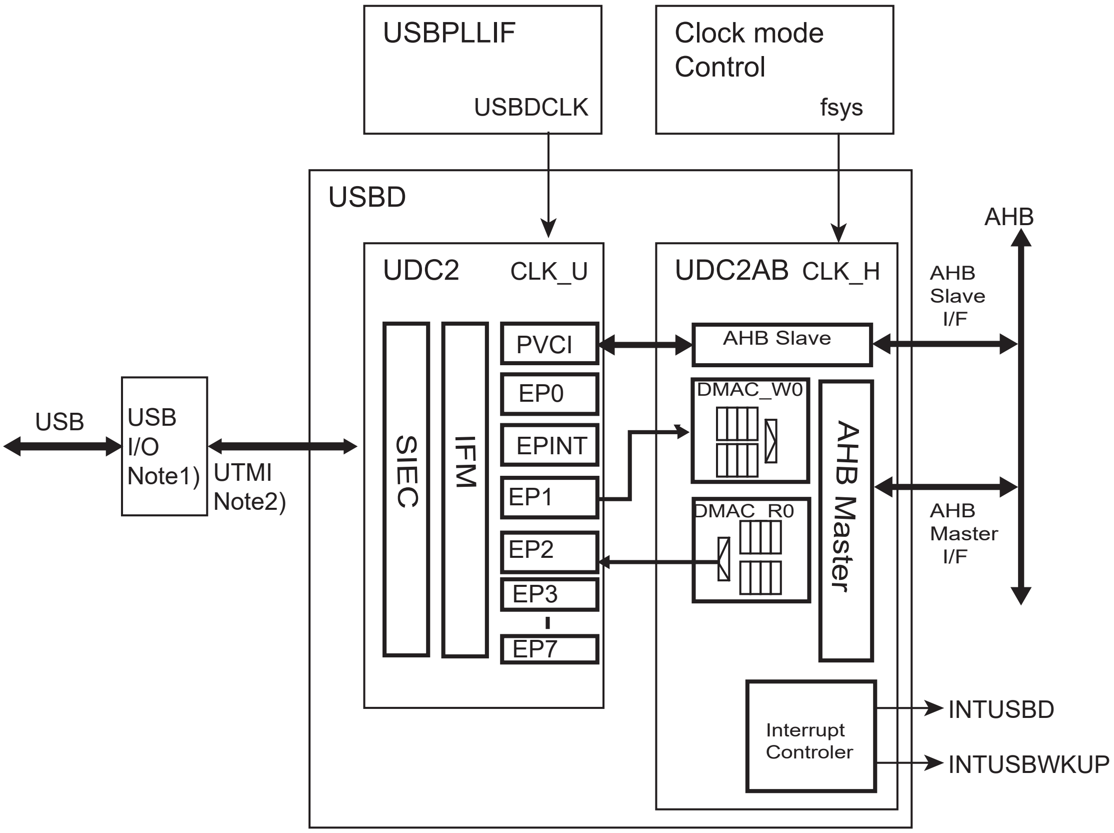
EP0:	Control	64byte × 1 FIFO
EP1:	Control / Interrupt / Bulk / Isochronous (IN)	64byte × 2 FIFO
EP2:	Control / Interrupt / Bulk / Isochronous (OUT)	64byte × 2 FIFO
EP3:	Control / Interrupt / Bulk / Isochronous (IN)	64byte × 2 FIFO
EP4:	Control / Interrupt / Bulk / Isochronous (OUT)	64byte × 2 FIFO
EP5:	Control / Interrupt / Bulk / Isochronous (IN)	64byte × 2 FIFO
EP6:	Control / Interrupt / Bulk / Isochronous (OUT)	64byte × 2 FIFO
EP7:	Control / Interrupt / Bulk / Isochronous (IN)	64byte × 2 FIFO

9. Supports Dual Packets Mode (except for EP 0)
10. Interrupt source signals to the interrupt controller: INTUSBD, INTUSBWKUP

17.2 System Structure

The USB device controller consists of the USB-Spec2.0 device controller (hereinafter called UDC2) and the bus bridge (hereinafter called UDC2AB) which connects the UDC2 and the AHB bus.

In this section, "17.2.1 AHB Bus Bridge (UDC2AB)" describes the configuration of the UDC2AB, and "17.2.2 Toshiba USB-Spec2.0 Device Controller (UDC2)" describes that of the UDC2.


USBPLLIF
USBDCLK
Clock mode Control
fsys
USBD
AHB
UDC2
CLK_U
UDC2AB
CLK_H
AHB Slave I/F
AHB Slave
PVCI
EP0
EPINT
EP1
EP2
EP3
EP7
SIEC
IFM
DMAC_W0
DMAC_R0
AHB Master
AHB Master I/F
USB
USB I/O Note1)
UTMI Note2)
Interrupt Controler
INTUSBD
INTUSBWKUP

Note1) TMPM367FDFG has USB I/O which can be used in Full Speed mode not Low Speed mode.
In this section, "PHY" should be read as USB I/O.
Note2) USB2.0 Transceiver Macrocell Interface

Figure 17-1 Block diagram of the USB device controller

17.2.1 AHB Bus Bridge (UDC2AB)

UDC2AB is the bus bridge between the Toshiba USB-Spec2.0 device controller (hereinafter called UDC2) and the AHB.

UDC2AB has the DMA controller that supports the AHB master transfer and controls transfer between the specified address on the AHB and the Endpoints-FIFO (EP I/F) in the UDC2.

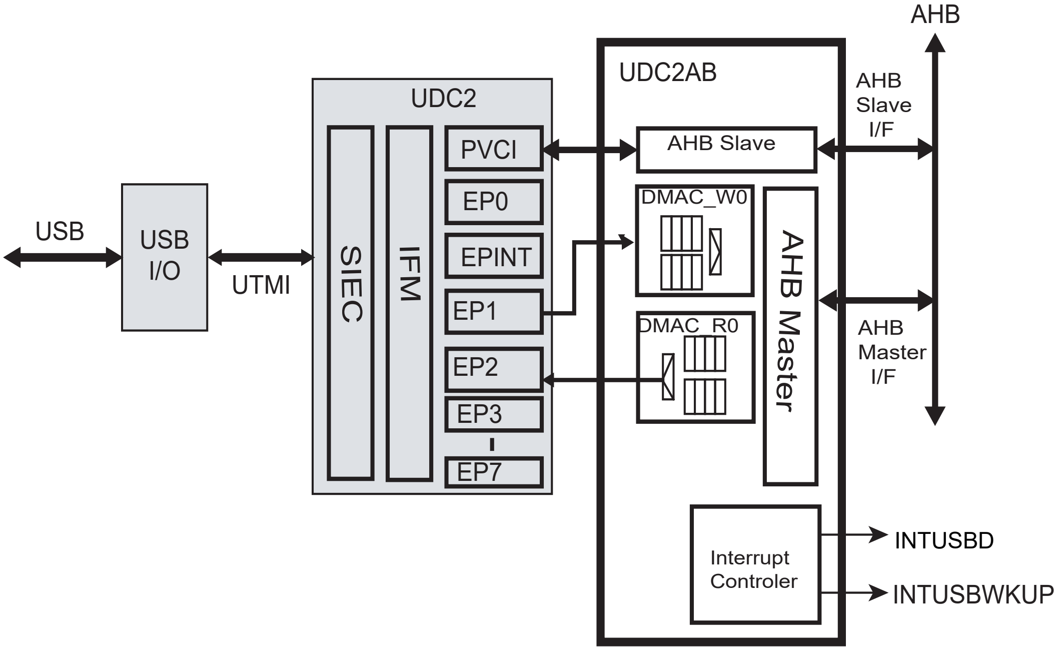

AHB
UDC2AB
AHB Slave I/F
UDC2
PVCI
AHB Slave
EP0
DMAC_W0
USB
USB I/O
EPINT
UTMI
SIEC
IFM
AHB Master
EP1
DMAC_R0
AHB Master I/F
EP2
EP3
EP7
Interrupt Controler
INTUSBD
INTUSBWKUP

Figure 17-2 UDC2AB Block Diagram

17.2.1.1 Functions and Features

UDC2AB has the following functions and features:

1. Connections with UDC2

   There is no specific restriction on the EP configuration for the UDC2 to be connected. However, the DMA controller in UDC2AB (AHB master function) can be connected with one Rx-EP and one Tx-EP. Accesses to other EPs (including EP0) should be made through PVCI I/F of UDC2 using the AHB slave function. Please note the EPx_FIFO register of a UDC2 EP in master transfer with the DMA controller cannot be accessed through PVCI I/F.

   If the maximum packet size of the EP to be connected with the AHB master read function is an odd number, there are some restrictions on the usage. See "(3) Setting the maximum packet size in Master Read transfers" for more information.

2. AHB functions

   AHB master and AHB slave functions are provided.

   a. AHB master function

      There are two DMA channels available for the USB device controller. One channel is allocated to the Rx-Ep and the Tx-Ep each.

Table 17-2 AHB Master Functions

Single Burst (INCR/INCR8) transactions	Supported
Split transaction	Supported
Little Endian	Supported
Protection Control	Supported
Early Burst Termination	Supported
Address bus width	32-bit
Data bus width	32-bit
Byte Word Transaction	Supported

The image of Endian conversion is as shown below.

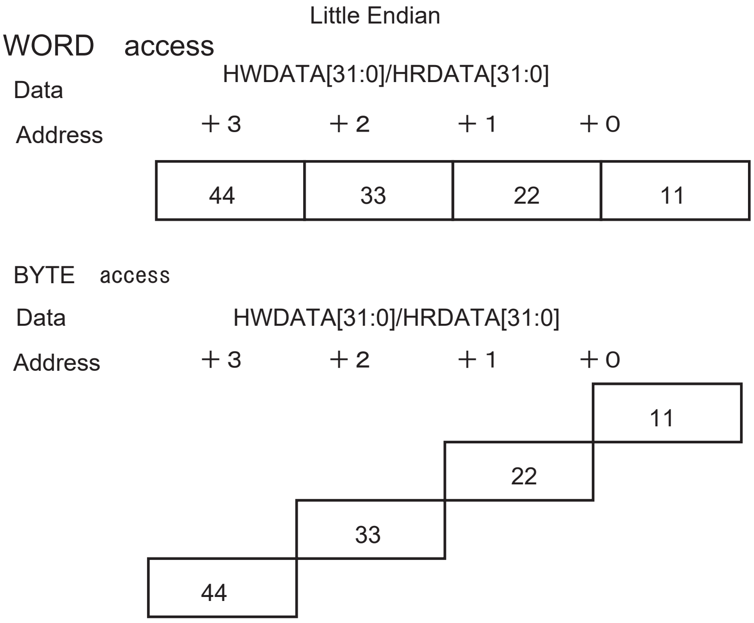

Little Endian
WORD access
Data
HWDATA[31:0]/HRDATA[31:0]
Address
+3
+2
+1
+0
44
33
22
11
BYTE access
Data
HWDATA[31:0]/HRDATA[31:0]
Address
+3
+2
+1
+0
11
22
33
44

Figure 17-3 Image of Endian conversion in AHB Master function

b. AHB Slave Function

Specification of the AHB Slave function.

Little Endian	Supported
Single transaction	Supported
Address width	32-bit
Data width	32-bit
Transaction in bytes or words	Supported

The image of Endian conversion is as shown below.

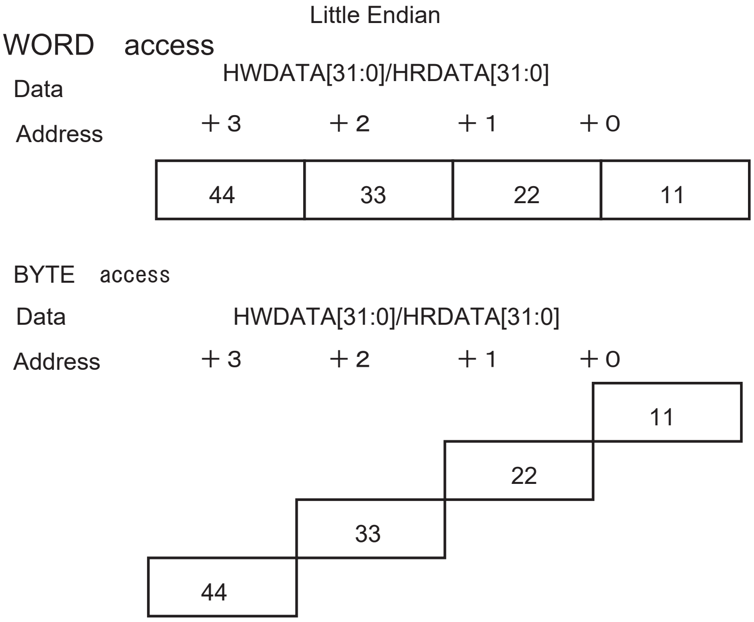


Figure 17-4 Image of Endian conversion in AHB Slave function

17.2.1.2 Configuration

UDC2AB mainly consists of the AHB Slave function that controls the access to the UDC2AB internal registers and UDC2 registers (UDC2 PVCI I/F) and the AHB Master function that controls the DMA access to the UDC2 EP I/F.

The AHB Master function has two built-in channels; Master Read Channel (AHB to UDC2) and Master Write Channel (UDC2 to AHB), which enable DMA transfer between the EP I/F of Rx-EP and Tx-EP of UDC2. Each channel has two built-in 8-word buffers (four in total).

17.2.1.3 Clock Domain

fsys provided by the clock/mode control circuit is connected to CLK_H of the UDC2AB. fsys stops or starts according to the low-power consumption mode of the TMPM367FDFG.

Since CLK_H is not provided during the fsys being stopped in the low-power consumption mode, INTUSBD will not be generated.

Therefore, to detect the connection and disconnection of the VBUS, an interrupt to used needs to be selected from INTUSBPOND generated by the USBDPON pin and INTUSBD at the moment when CLK_H starts or stops.

Refer to "17.5.5.2 Sequence of USB Bus Power (VBUS) Connection/Disconnection" for more information.

USBDCLK provided from USB clock control circuit is connected to CLK_U of UDC2. USBDCLK stops or starts according to the register. To stop or start CLK_U by detecting status such as suspend and resume, configure USB clock control circuit using software.

17.2.2 Toshiba USB-Spec2.0 Device Controller (UDC2)

UDC2 controls the connection of USB functions to the Universal Serial Bus. UDC2 automatically processes the USB protocol and its PHY-end interface can be accessed via UTMI.

1. SIEC (Serial Interface Engine Control) block

   This block manages the protocol in USB. Its major functions are:

   - Checks and generates PIDs
   - Checks and generates CRCs
   - Checks device addresses

2. IFM block

   This block controls SIEC and EPs. Its major functions are:

   - Writes the received data to the relevant EPs when received an OUT-Token.
   - Reads the transmit data from the relevant EPs when an IN-Token is received.
   - Controls and manages the status of UDC2.0.

3. PVCI-I/F block

   This block controls reading and writing between IFM and external register access bus (PVCI).

   PVCI bus accesses via UDC2AB.

4. EP0 block

   This block controls sending and receiving data in Control transfers. When sending or receiving data with DATA-Stage of Control transfers, you should access the FIFO in this block via PVCI-I/F.

5. EPx block

   This block controls sending and receiving data of EPx (x = 1 to 7). FIFO can be directly accessed via the EP-I/F. The EP-I/F can make burst transfers.

   Please note there are two EPs; one for sending (EPTX) and another for receiving (EPRX). Direction of EPs (send/receive) will be fixed on a hardware basis.

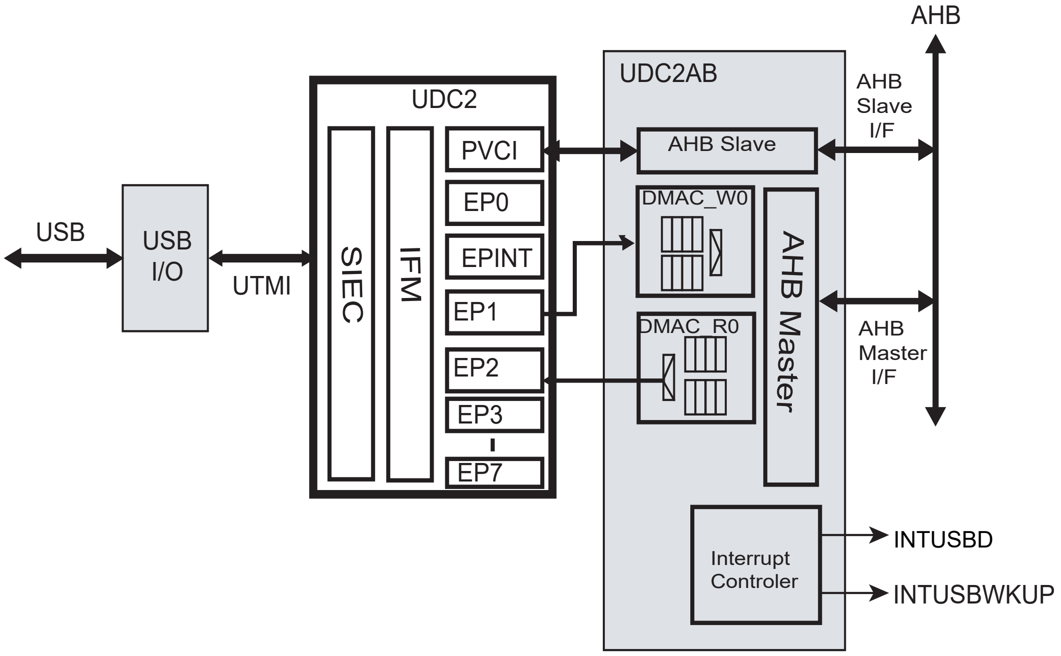

AHB
UDC2AB
AHB Slave I/F
UDC2
PVCI
AHB Slave
EP0
DMAC_W0
USB
USB I/O
UTMI
SIEC
IFM
EPINT
AHB Master
EP1
DMAC_R0
AHB Master I/F
EP2
EP3
EP7
INTUSBD
Interrupt Controler
INTUSBWKUP

Figure 17-5 Block diagram of UDC2

17.2.2.1 Features and Functions

The main features and functions are as follows:

1. Complies with Universal Serial Bus Specification Rev. 2.0.
2. Complies with Full-Speed (FS) (not complies with Low-Speed).
3. USB protocol processing
4. Detects SOF/USB_RESET/SUSPEND/RESUME.
5. Generates and checks packet IDs.
6. Checks CRC5. Generates and checks CRC16.
7. Supports 4 transfer modes (Control/Interrupt/Bulk/Isochronous).
8. Supports up to 8 EPs.
9. Supports Dual Packet Mode (EP 0 excluded)
10. EP 1 to 7 can directly access FIFO (EP-I/F)
11. Complies with USB 2.0 Transceiver Macrocell Interface (UTMI) (8 bits @ 48 MHz)

17.2.2.2 Specifications of Flags

The UDC2 core outputs various events on USB as flags when they occur. This section discusses those flags.

1. USB_RESET

   Asserts "High" while receiving USB_RESET. Since UDC2 returns to the Default-State by receiving USB_RESET, the application also needs to return to the Default-State.

   In Full-Speed operation, UDC2 asserts this flag when SE0 on the USB bus was recognized for 2.5 s or longer. Then, after UDC2 has driven Chirp-K for about 1.5 ms the flag will be deasserted when either one of the following states was recognized:

   a. Chirp from the host (K-J-K-J-K-J) was recognized.
   b. 2 ms or longer has passed without recognizing Chirp from the host (K-J-K-J-K-J).

Note: While the time when the host begins Chirp and the driving time of Chirp-K and Chirp-J depend on the host, asserting period of the USB_RESET flag is around 1.74 ms to 3.5 ms.

2. INT_SETUP

   In Control transfers, asserts "High" after receiving the Setup-Token. When this interrupt is recognized, the software should read the Setup-Data storage register (8 bytes) to make judgment of request.This interrupt will be deasserted by writing 1 into the UDFS2INT<i_setup>. UDFS2INT should be cleared at the point the interrupt was recognized.

3. INT_STATUS_NAK

   In Control transfers, when the host proceeds to the STATUS-Stage and transmits packets while UDC2 is processing the DATA-Stage (before issuing the "Setup_Fin" command), UDC2 will return "NAK" and asserts this flag to "High". When this interrupt is recognized, the software should issue the "Setup_Fin" command from the Command register to make the STATUS-Stage of UDC2 end. This interrupt will be deasserted by writing 1 into the UDFS2INT<i_status_nak>. UDFS2INT should be cleared at the point the interrupt was recognized.

4. INT_STATUS

   In Control transfers, asserts "High" after finishing the STATUS-Stage normally. This interrupt will be deasserted by writing 1 into the UDFS2INT<i_status>. UDFS2INT should be cleared at the point the interrupt was recognized.

5. INT_EP0

   In the DATA-Stage of Control transfers, asserts "High" when "ACK" was sent or received (when the transaction finished normally). This interrupt will be deasserted by writing 1 to the UDFS2INT<i_ep0>. UDFS2INT should be cleared at the point the interrupt was recognized.

6. INT_EP

   In EPs other than EP 0, asserts "High" when "ACK" was sent or received (when the transaction finished normally). In that case, which EP the transfer was made can be identified by checking UDFS2INTEP. This interrupt will be deasserted by writing 1 into the UDFS2INT<i_ep>, or by writing 1 into all bits set in UDFS2INTEP. UDFS2INT should be cleared at the point the interrupt was recognized.

7. INT_RX_ZERO

   "High" is asserted when Zero-Length data is received. In Control transfers, however, "High" is asserted only when Zero-Length data is received in the DATA-Stage. It will not be asserted when Zero-Length data is received in the STATUS-Stage. Which EP has received the data can be identified by reading the UDFS2CMD<rx_nulpkt_ep> or checking UDFS2INTRX0. This interrupt will be deasserted by writing 1 into the UDFS2INT<i_rx_data0>, or by writing 1 into all bits set in UDF2INTRX0. UDFS2INTRX0 should be cleared at the point the interrupt was recognized.

8. INT_SOF

   Asserts "High" when SOF was received. This interrupt will be deasserted by writing 1 into the UDFS2INT<i_osf>. UDFS2INT should be cleared at the point the interrupt was recognized.

   SOF is a packet indicating the start of a frame. It is transmitted from the host to devices every 1ms in the Full-Speed transfers.

9. INT_NAK

   In EPs other than EP 0, asserts "High" when NAK is transmitted. In that case, which EP has transmitted the NAK can be identified by checking UDFS2INTNAK. This interrupt will be deasserted by writing 1 into the UDFS2INT<i_nak>, or by writing 1 into all bits set in UDFS2INTNAK. By default, this flag will not be asserted when NAK was transmitted. Therefore, you should write 0 into the relevant EP of UDFS2INTNAKMASK to release the mask in order to use this flag.

17.2.2.3 Commands to EP

This section describes about the commands to be issued by UDFS2CMD<com> for the EP specified by UDFS2CMD<ep>.

1. 0x0 : Reserved

   Not to be specified.

2. 0x1 : Setup_Fin

   Should be issued only for EP0.

   This is a command for setting the end of DATA-Stage in Control transfers. As UDC2 continues to send back "NAK" to the STATUS-Stage until this command is issued, the command should be issued when the DATA-Stage finishes or INT_STATUS_NAK was received.

Note: During Control-WR transfer, read all data received during the Data-Stage first, and then Issue the Setup-Fin command.

3. 0x2 : Set_DATA0

   Can be issued to EPs except EP0. Should not be issued to EP0.

A command for clearing toggling of EPs. While toggling is automatically updated by UDC2 in normal transfers, this command should be issued if it needs to be cleared by software.

4. 0x03 : EP_Reset

   Can be issued to any EP.

   A command for clearing the data and status of EPs. Issue this command when you want to reset an EP in such cases as setting EPs of Set_Configuration and Set_Interface or resetting the EP by Clear_Feature. This command will reset the following 5 points:

   a. Clear the UDFS2EP0STS<toggle> / UDFS2EPxSTS<toggle> to DATA0.
   b. Clear the UDFS2EP0STS<status> / UDFS2EPxSTS<status> to Ready.
   c. Clear the UDFS2EP0MSZ<dset> / UDFS2EPxMSZ<dset> and the UDFS2EP0DSZ / UDFS2EPxDSZ.
   d. Clear UDFS2EP0MSZ<tx0_data> / UDFS2EPxMSZ<tx_0data>.
   e. Clear the UDFS2EPxSTS<disable>.

   UDC2 makes toggling control by hardware for every transfer. If this command is issued when a transfer of EPs is in progress, toggling of the relevant EP will also be cleared which may cause the synchronization with the host be lost. As in the case of receiving requests as mentioned above, the command should be issued when it is possible to make synchronization with the host.

5. 0x4 : EP_Stall

   Can be issued to EPs except EP0. Should not be issued to EP0.

   A command for setting the status of EPs to "Stall". Issue this command when you want to set the status of an EP to "Stall" in such cases as stalling an EP by Set_Feature. When this command is issued, "STALL" will be always sent back for the EP set. However, the Stall status of EP0 will be cleared when the Setup-Token is received.

   This command should not be issued for EPs where Isochronous transfers are used, since transfers are made without Handshake in Isochronous transfers. Even if this command is issued for EPs where Isochronous transfers are set (by UDFS2EOxSTS<t_type>), "STALL" will not be sent back.

6. 0x5 : EP_Invalid

   Can be issued to EPs except EP0. Should not be issued to EP0.

   A command for setting the status of EPs to "Invalid". Please issue this command when disabling EPs that will not be used when using Set_Config or Set_Interface to set EPs. When this command is issued, the EPs set will make no response. This command should not be issued while transfers of each EP are in progress.

7. 0x6 : Reserved

   Not to be specified.

8. 0x7 : EP_Disable

   Can be issued to EPs except EP0. Should not be issued to EP0.

   A command for making an EP disabled. When this command is issued, "NAK" will be always sent back from the EP set. This command should not be issued for EPs where Isochronous transfers are used, since transfers are made without Handshake in Isochronous transfers. Even if this command is issued for EPs where Isochronous transfers are set (by UDFS2EPxSTS<t_type>), "NAK" will not be sent back.

9. 0x8 : EP_Enable

Can be issued to EPs except EP0. Should not be issued to EP0.

A command for making an EP enabled. Issue this command to cancel the disabled status set by "EP_Disable" command.

10. 0x9 : All_EP_Invalid

    Setting for EP is invalid.

    A command for setting the status of all EPs other than EP0 to "Invalid". Issue this command when you want to apply the "EP_Invalid" command for all EPs. Issue this command when processing Set_Configuration and Set_Interface like the "EP_Invalid" command.

11. 0xA : USB_Ready

    Should be issued only for EP0.

    A command for making connection with the USB cable. Issue this command at the point when communication with the host has become effective after confirming the connection with the cable. Pull-Up of USB-DDP will be made only after this command is issued, and the status of cable connection will be sent to the host.

    Please note that the device state of UDC2 (UDFS2ADR<configued> <addressed> <default>) will be set to "Default" when this command was issued.

12. 0xB : Setup_Received

    Should be issued for EP0.

    A command for informing UDC2 that the SETUP-Stage of a Control transfer was recognized. Issue this command after accepting the INT_SETUP interrupt and the request code was recognized. As UDC2 continues to send back "NAK" to the DATA-Stage/STATUS-Stage until this command is issued, the command should be issued at the end of the INT_SETUP interrupt processing routine.

13. 0xC : EP_EOP

    Can be issued to any EP.

    A command for informing UDC2 that the transmit data has been written. Issue this command when transmitting data with byte size smaller than the maximum transfer bytes (FIFO capacity of the EP or MaxPacketSize, whichever smaller). Issuing this command will set the Data set flag and the data will be sent back to IN-Token from the host. It should not be used when setting Zero-Length data or data of MaxPacketSize.

14. 0xD : EP_FIFO_Clear

    Can be issued to any EP.

    A command for clearing the data of an EP. The UDFS2EPxMSZ<dset> and the UDFS2EPxDSZ will be cleared at the same time. Issue this command when you want to clear the data currently stored in the FIFO before transmitting the data to the host and set the latest data, for instance in Interrupt transfers. If this command is issued while accessing the EP-I/F, the FIFO of the EP will not be successfully cleared. When issuing this command, epx_val of EP-I/F should be set to 0 before issuing.

15. 0xE : EP_TX_0DATA

    Can be issued to any EP.

    A command for setting Zero-Length data to an EP. Issue this command when you want to transmit Zero-Length data. In the case of transmitting Zero-Length data in Bulk-IN transfers and others to indicate the final transfer, read UDFS2EPxDSZ and confirm it is 0 (no data exists in the FIFO of EPx) before setting this command. In the case of writing data from EP-I/F, set this command after the data was written and epx_val became 0. When this command was set, UDFS2EPxMS<tx_0data> of the EP will be set.

Set the next data when the UDFS2EPxMS<tx_0data> is confirmed to be 0. During Isochronous-IN transfer, Zero -Length data is automatically transferred to the IN-Token if data is not set in the FIFO of EP. Do not issue this command.

16. 0xF : Reserved

    Not to be specified.

Settings for the following commands will be suspended when issued during a USB transfer, which will be executed after the USB transfer has finished. Suspension of the command will take place for each EP.

- 0x2: Set_DATA0
- 0x3: EP_Reset
- 0x4: EP_Stall
- 0x5: EP_Invalid
- 0x7: EP_Disable
- 0x8: EP_Enable
- 0x9: All_EP_Invalid
- 0xD: EP_FIFO_Clear
- 0xE: EP_TX_0DATA

Therefore, when commands were issued successively for the same EP while a USB transfer is in progress, commands will be overwritten and only the one last issued will be valid. If you need to issue commands to an EP successively, poll UDFS2EPxSTS / UDFS2EPxDSZ to confirm that the command has become valid before issuing next ones. Also, when making an access to the EP-I/F immediately after clearing the FIFO using the EP_Reset/EP_FIFO_Clear command, poll UDFS2EPxDSZ to confirm that the command has become valid before resuming the access to the EP-I/F.

For EP 0, the following commands will be invalid until the Setup_Received command is issued after receiving the Setup-Token:

- 0x1: Setup_Fin
- 0x2: Set_DATA0
- 0x3: EP_Reset
- 0x4: EP_Stall
- 0xC: EP_EOP
- 0xD: EP_FIFO_Clear
- 0xE: EP_TX_0DATA

When the "EP_Stall" command was set to EPx, "Stall" will be set to the UDFS2EPxSTS<status>. When EP_Disable was set, 1 will be set to the UDFS2EPxSTS<disable>. When these two commands (EP_Stall and EP_Disable) were set to the same EPx and the status becomes "Stall" with UDFS2EPxSTS<disable> = 1, "STALL" will be transmitted in the transfer.

When the "EP_Invalid" command was set to EPx, "Invalid" will be set to the UDFS2EPxSTS<status>. When the two commands (EP_Invalid and EP_Disable) were set to the same EPx and the status becomes "Invalid" with UDFS2EPxSTS<disable> = 1, no response will be made in the transfer.

When UDFS2EPxSTS<disable> is 1 and UDFS2EPxMSZ<tx_0data> is 1, Zero-Length data will be transmitted once in the transfer. After the Zero-Length data was successfully transferred, "NAK" will be transmitted.

17.3 How to connect with the USB bus

The circuit below shows how to connect TMPM367FDFG with the USB bus.

Input the VBUS signal to USBDPON pin to detect the connection of the USB power (VBUS).

The Pull-Up process using the pull-up resistor of the D+ and the in-line damping resistor to the D- are required. Also, a ON/OFF control using a port needs to be added to the pull-up resistor. The pull-up resistor should be disconnected when voltage is not applied to the VBUS.

If D+ and D- are unstable, we recommend to add a pull-down resistor to R_1.

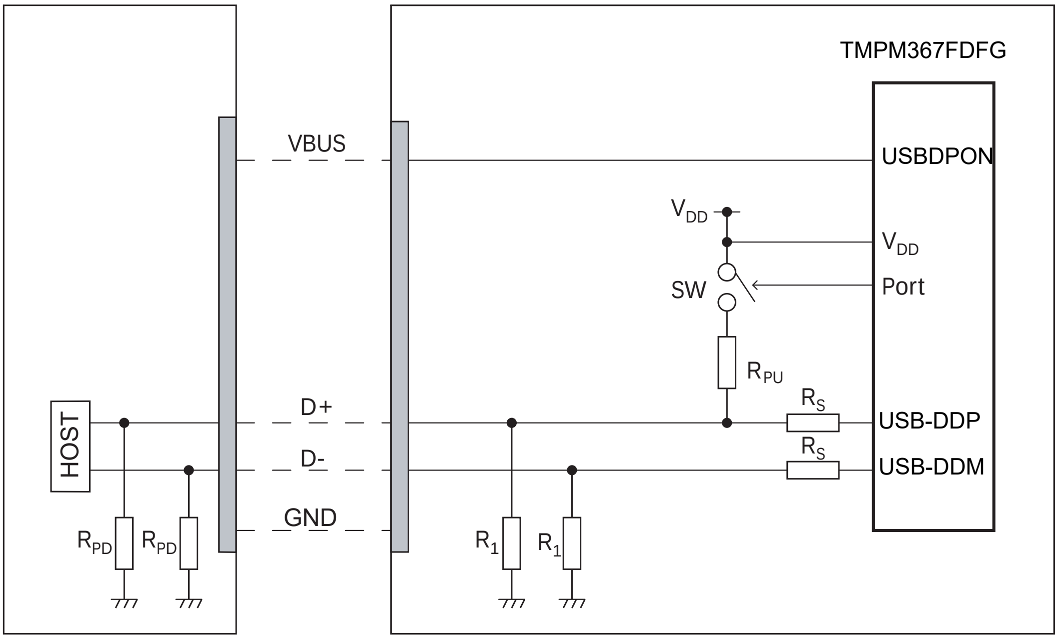

TMPM367FDFG
VBUS
USBDPON
VDD
VDD
SW
Port
RPU
RS
D+
USB-DDP
HOST
D-
RS
USB-DDM
GND
RPD
RPD
R1
R1

Figure 17-6 Connection example of the USB bus and TMPM367FDFG.

Note: R_1=500kΩ or higher (recommended value), R_S=33Ω (recommended value), R_{PU}=1.5kΩ (recommended value)

17.4 Registers

The register map of the USBD consists of registers for setting the UDC2AB and that for the UDC2.

When the registers for setting UDC2 are accessed, UDC2AB automatically accesses UDC2 via PVCI I/F.

The register for setting the UDC2AB is 32-bit width. The register for setting the UDC2 is 16-bit width, which is allocated to [15:0]. [31:16] is allocated to the read-only, for indefinite values.

17.4.1 UDC2AB Register

17.4.1.1 UDC2AB Register list

BaseAddress=0x4000_8000

Register name		Address(Base+)
Interrupt Status Register	UDFSINTSTS	0x0000
Interrupt Enable Register	UDFSINTENB	0x0004
Master Write Timeout Register	UDFSMWTOUT	0x0008
UDC2 Setting Register	UDFSC2STSET	0x000C
DMAC Setting register	UDFSMSTSET	0x0010
DMAC Read Request Register	UDFSDMACRDREQ	0x0014
DMAC Read Value Register	UDFSDMACRDVL	0x0018
UDC2 Read Request Register	UDFSUDC2RDREQ	0x001C
UDC2 Read Value Register	UDFSUDC2RDVL	0x0020
-	Reserved	0x0024 to 0x0038 (note 2)
Arbiter Setting Register	UDFSARBTSET	0x003C
Master Write Start Address Register	UDFSMWSADR	0x0040
Master Write End Address Register	UDFSMWEADR	0x0044
Master Write Current Address Register	UDFSMWCADR	0x0048 (note 1)
Master Write AHB Address Register	UDFSMWAHBADR	0x004C
Master Read Start Address Register	UDFSMRSADR	0x0050
Master Read End Address Register	UDFSMREADR	0x0054
Master Read Current Address Register	UDFSMRCADR	0x0058 (note 1)
Master Read AHB Address Register	UDFSMRAHBADR	0x005C
-	Reserved	0x0060 to 0x007C (note 2)
Power Detect Control Register	UDFSPWCTL	0x0080
Master Status Register	UDFSMSTSTS	0x0084
Timeout Count Register	UDFSTOUTCNT	0x0088 (note 1)
-	Reserved	0x008C to 0x1FC

Note 1: Be sure to make Read accesses via UDFSDMACRDREQ.

Note 2: Those shown as "Reserved" above are prohibited to access. Read / Write is prohibited to those "Reserved" areas.

17.4.1.2 UDFSINTSTS (Interrupt Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	int_mw_rerror	int_ powerdetect	-	-	int_dmac_ reg_rd	int_udc2_ reg_rd
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	int_mr_ahberr	int_mr_ep_ dset	int_mr_end_ add	int_mw_ ahberr	int_mw_ timeout	int_mw_end_ add	int_mw_set_ add	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	int_usb_ reset_end	int_usb_reset	int_suspend_ resume
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	int_nak	int_ep	int_ep0	int_sof	int_rx_zero	int_status	int_status_ nak	int_setup
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-30	-	R	Read as undefined.
29	int_mw_rerror	R/W	Will be set to 1 when the access to the EP has started Master Write transfer during the setting of common bus access (UDFS2EPxSTS<bus_sel> is 0).(UDFS2EPxSTS<bus_sel> is 0) 0: Not detected 1: EP read error occurred during master write.
28	int_ powerdetect	R/W	When the status of VBUSPOWER input of UDC2AB changed, it is set to "1". 0:No changed 1:Status changed
27-26	-	R	Read as undefined.
25	int_dmac_ reg_rd	R/W	Will be set to 1 when the register access executed by the setting of UDFSDMACRDREQ is completed and the value read to UDFSDMACCRDVL is set. 0: Not detected. 1:Register read completed.
24	int_udc2_reg_ rd	R/W	Will be set to 1 when the UDC2 access executed by the setting of UDFSDMACRDREQ is completed and the value read to UDFSDMACRDLV is set. Also set to 1 when Write access to the internal register of UDC2 is completed. 0: Not detected. 1: Register read/write completed.
23	int_mr_ahberr	R/W	This status will be set to 1 when the AHB error has occurred during the operation of Master Read transfer. After this interrupt has occurred, the Master Read transfer block needs to be reset by the UDFSMSTSET<mr_reset >. 0: Not detected. 1: AHB error occurred.
22	int_mr_ep_dset	R/W	Will be set to 1 when the FIFO of EP for UDC2 Tx to be used for Master Read transfer becomes writable (not full). 0: FIFO is not writable 1: FIFO is writable
21	int_mr_end_ add	R/W	Will be set to 1 when the Master Read transfer has finished. 0: Not detected 1: Master Read transfer finished
20	int_mw_ahberr	R/W	This status will be set to 1 when the AHB error has occurred during the operation of Master Write transfer. After this interrupt has occurred, the Master Write transfer block needs to be reset by UDFSMSTSET<mw_reset>. 0: Not detected. 1: AHB error occurred.

Bit	Bit Symbol	Type	Function
19	int_mw_ timeout	R/W	This status will be set to 1 when time-out has occurred during the operation of Master Write transfer. 0: Not detected. 1: Master Write transfer timed out.
18	int_mw_end_ add	R/W	Will be set to 1 when the Master Write transfer has finished. 0: Not detected. 1: Master Write transfer timed out.
17	int_mw_set_ add	R/W	Will be set to 1 when the data to be sent by Master Write transfer is set to the corresponding EP of Rx while the Master Write transfer is disabled. 0: Not detected. 1: Master Write Transfer address request.
16-11	-	R	Read as undefined.
10	int_usb_reset_ end	R/W	Indicates whether UDC2 has deasserted the usb_reset signal. The timing in which UDC2 sets the UDC2 register to the initial value after USB_RESET is after the usb_reset signal is deasserted. To detect this timing, use this bit. The status of the usb_reset signal can be checked using UDFSPWCTL<usb_reset>. 0: UDC2 has not deasserted the usb_reset signal after this bit was cleared. 1: Indicates UDC2 has deasserted the usb_reset signal.
9	int_usb_reset	R/W	Indicates whether UDC2 has asserted the usb_reset signal. The status of the usb_reset signal can be checked using UDFSPWCTL<usb_reset>. 0: UDC2 has not asserted the usb_reset signal after this bit was cleared. 1:Indicates UDC2 has asserted the usb_reset signal.
8	int_suspend_ resume	R/W	Asserts 1 each time the suspend_x signal of UDC2 changes. The status can be checked using the UDFSPWCTL<suspend_x>. 0:Status has not changed. 1: Status has changed.
7	int_nak	R	The int_nak signal of UDC2 can be directly read. To clear it, clear the corresponding bit of UDFS2INT or UDFS2INTNAK of UDC2.
6	int_ep	R	The int_ep signal of UDC2 can be directly read. To clear it, clear the corresponding bit of UDFS2INT or UDFS2INTEP.
5	int_ep0	R	The int_ep0 signal of UDC2 can be directly read. To clear it, clear the corresponding bit of UDFS2INT.
4	int_sof	R	The int_ep0 signal of UDC2 can be directly read. To clear it, clear the corresponding bit of UDFS2INT.
3	int_rx_zero	R	The int_rx_zero signal of UDC2 can be directly read. To clear it, clear the corresponding bit of UDFS2INT or UDFS2INTRX0.
2	int_status	R	The int_status signal of UDC2 can be directly read. To clear it, clear the corresponding bit of UDFS2INT.
1	int_status_nak	R	The int_status_nak signal of UDC2 can be directly read. To clear it, clear the corresponding bit of UDFS2INT.
0	int_setup	R	The int_setup signal of UDC2 can be directly read. To clear it, clear the corresponding bit of UDFS2INT.

The connection between the output signals of UDC2 and the bits [10:9] and [7:0]of this register is shown below.

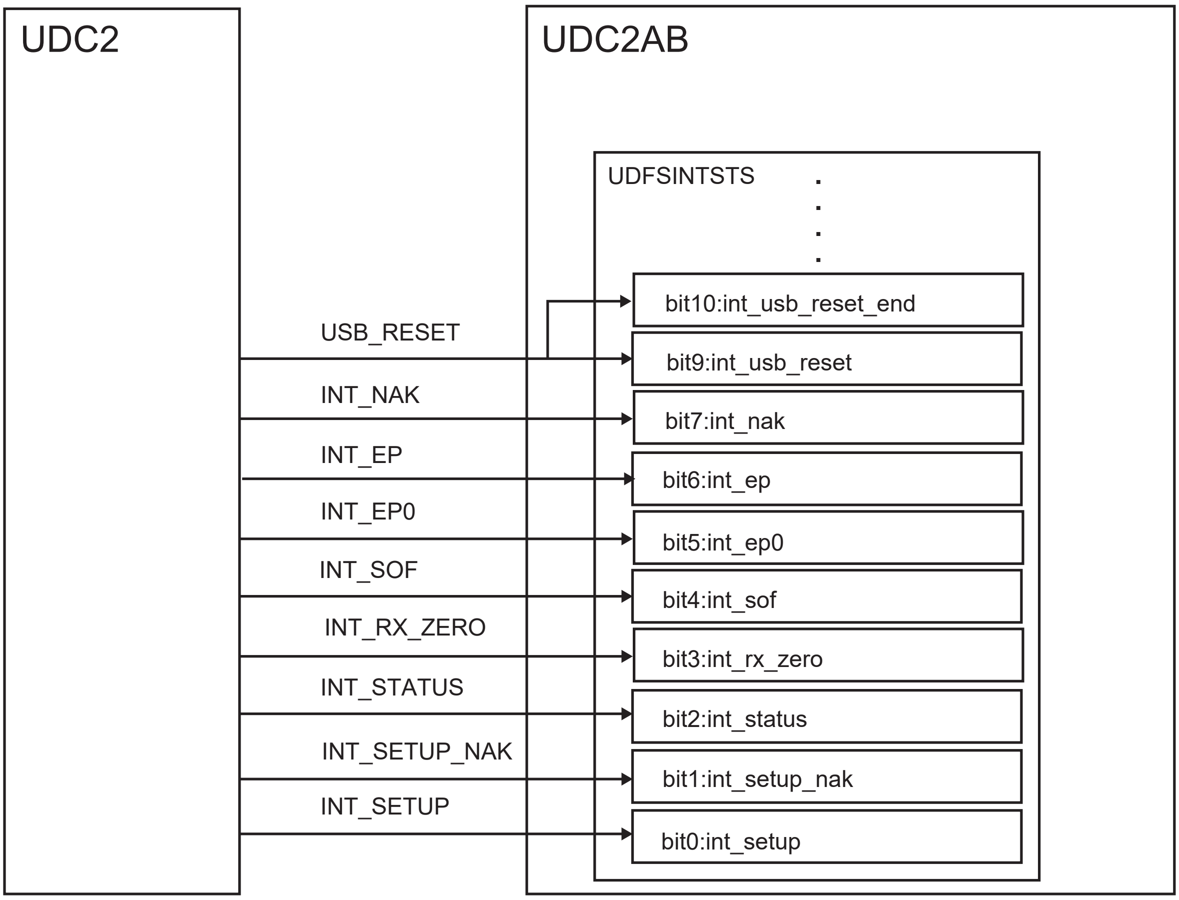

UDC2
UDC2AB
UDFSINTSTS
USB_RESET
INT_NAK
INT_EP
INT_EP0
INT_SOF
INT_RX_ZERO
INT_STATUS
INT_SETUP_NAK
INT_SETUP
bit10:int_usb_reset_end
bit9:int_usb_reset
bit7:int_nak
bit6:int_ep
bit5:int_ep0
bit4:int_sof
bit3:int_rx_zero
bit2:int_status
bit1:int_setup_nak
bit0:int_setup

Figure 17-7 Connection between the flag output signals and interrupt bits.

17.4.1.3 UDFSINTENB(Interrupt Enable Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	mw_rerror_en	power_ detect_en	-	-	dmac_reg_ rd_en	udc2_reg_ rd_en
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	mr_ahberr_en	mr_ep_ dset_en	mr_end_ add_en	mw_ahberr_ en	mw_ timeout_en	mw_end_ add_en	mw_set_ add_en	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	usb_reset_ end_en	usb_reset_en	suspend_ resume_en
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-30	-	R	Read as undefined.
29	mw_rerror_en	R/W	Controls the mw_rerror interrupt. 0: Disable 1: Enable
28	power_detect_ en	R/W	Controls the power_detect interrupt. 0: Disable 1: Enable
27-26	-	R	Read as undefined.
25	dmac_reg_ rd_en	R/W	Controls the dmac_reg_rd interrupt. 0: Disable 1: Enable
24	udc2_reg_ rd_en	R/W	Controls the udc2_reg_rd interrupt. 0: Disable 1: Enable
23	mr_ahberr_en	R/W	Controls the mw_ahberr interrupt. 0: Disable 1: Enable
22	mr_ep_dset_en	R/W	Controls the mr_ep_dset interrupt. 0: Disable 1: Enable
21	mr_end_ add_en	R/W	Controls the mr_end_add interrupt. 0: Disable 1: Enable
20	mw_ahberr_en	R/W	Controls the mw_ahberr interrupt. 0: Disable 1: Enable
19	mw_ timeout_en	R/W	Controls the mw_timeout interrupt. 0: Disable 1: Enable
18	mw_end_ add_en	R/W	mw_end_add interrupt. 0: Disable 1: Enable
17	mw_set_ add_en	R/W	mw_set_add interrupt. 0:Disable 1: Enable
16-11	-	R	Read as undefined.

Bit	Bit Symbol	Type	Function
10	usb_reset_ end_en	R/W	usb_reset_end interrupt. 0: Disable 1: Enable
9	usb_reset_en	R/W	usb_reset interrupt. 0: Disable 1: Enable
8	suspend_ resume_en	R/W	suspend_resume interrupt. 0: Disable 1: Enable
7-0	-	R	Read as undefined.

17.4.1.4 UDFSMWTOUT(Master Write Timeout Register)

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

Bit	Bit Symbol	Type	Function
31-1	timeoutset	R/W	The setting should not be changed during the Master Write transfer. Timeout occurs when the number of times CLK_U was set is counted after the data of Master Write (Rx) EP is exhausted. The timeout counter comprises 32 bits of which upper 31 bits can be set by timeoutset [31:1] of this register, while the lowest bit of the counter is set to 1. As CLK_U is 48 MHz, approximately 20 [ns] to 89 [s] can be set as a timeout value. While CLK_U stopped (PHY is being suspended and so on), no timeout interrupt will occur as the counter does not work.
0	timeout_en	R/W	Used to enable Master Write timeout. It is set to Enable by default. The setting should not be changed during the Master Write transfer. 0: Disable 1: Enable

17.4.1.5 UDFSC2STSET(UDC2 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	-			eopb_enable	-	-	-	tx0
After reset	0	0	0	1	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as undefined.
4	eopb_enable	R/W	Used to enable Master Read EOP. It is set to Enable by default. The setting should not be changed during the Master Read transfer. If this bit is 0, the final data transfer to UDC2 will not take place when the last word is 1 byte. If the last word is 2 bytes, the final data transfer to UDC2 will take place when epx_w_eop = 0. If this bit is 1, the final data transfer to UDC2 will take place when epx_w_eop = 1 regardless of byte number of the last word. Note: See the section"17.5.4.1 Master Read transfer" for more information. 0: Master Read EOP Disabled. 1: Master Read EOP Enabled.
3-1	−	R	Read as undefined.
0	tx0	R/W	Used to transmit NULL packets at an EP connected to the Master Read operation side. Only valid when the UDFSMSTSTS<mrepempty> is 1, otherwise this bit is ignored. It will be automatically cleared to 0 after writing. Setting 1 to this bit will assert the epx_tx0data signal of the UDC2 EP-I/F and the value of 1 is retained during the transmission of NULL packets. After this bit is set, next data setting for Tx-EP should not be made until it is cleared. 0: No operation 1: Transmits NULL packets.

17.4.1.6 UDFSMSTSET(DMAC 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	-	-	-	-	-	-	-	m_burst_type
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	mr_reset	mr_abort	mr_enable	-	mw_reset	mw_abort	mw_enable
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-9	−	R	Read as undefined.
8	m_burst_type	R/W	Selects the type of HBURST[2:0] when making a burst transfer in Master Write/Read transfers. The type of burst transfer made by UDC2AB is INCR8 (burst of 8 beat increment type). Accordingly, 0 (initial value) should be set in normal situation. However, in case INCR can only be used as the type of burst transfer based on the AHB specification of the system, set 1 to this bit. In that case, UDC2AB will make INCR transfer of 8 beat. Please note the number of beat in burst transfers cannot be changed. Setting of this bit should be made in the initial setting of UDC2AB. The setting should not be changed after the Master Write/Read transfers started. Note: UDC2AB does not make burst transfers only in Master Write/Read transfers. It combines burst transfers and single transfers. This bit affects the execution of burst transfers only. 0: INCR8 1: INCR
7	-	R	Read as undefined.
6	mr_reset	R/W	Initializes the Master Read transfer block of UDC2AB. However, as the FIFOs of EPs are not initialized, you need to access the UDFS2CMD of UDC2 to initialize the corresponding EP separately from this reset. This reset should be used after stopping the Master operation. This bit will be automatically cleared to 0 after being set to 1. Subsequent Master Read transfers should not be made until it is cleared. 0: No operation 1: Reset
5	mr_abort	W	Controls Master Read transfers. Master Read operations can be stopped by setting 1 to this bit. When aborted during transfers, transfer of buffers for Master Read to UDC2 is interrupted and the <mr_enable> bit is cleared, stopping the Master Read transfer. Aborting completes when the <mr_enable> bit is disabled to 0 after setting this bit to 1. 0: No operation 1: Abort
4	mr_enable	R/W	Controls Master Read transfers. Enabling should be made when setting the transfer address is completed. It will be automatically disabled as the master transfer finishes. Since Master Read operations cannot be disabled with this register, use the <mr_abort> bit if the Master Read transfer should be stopped. 0: Disable 1: Enable
3	−	R	Read as undefined.
2	mw_reset	R/W	Initializes the Master Write transfer block. However, as the FIFOs of EPs are not initialized, you need to access the UDFS2CMD of UDC2 to initialize the corresponding EP separately from this reset. This reset should be used after stopping the Master operation. This bit will be automatically cleared to 0 after being set to 1. Subsequent Master Write transfers should not be made until it is cleared. 0: No operation 1: Reset
1	mw_abort	W	Controls Master Write transfers. Master Write operations can be stopped by setting 1 to this bit. When aborted during transfers, transfer of buffers for Master Write from UDC2 is interrupted and the <mw_enable> bit is cleared, stopping the Master Write transfer. Aborting completes when the <mw_enable> bit is disabled to 0 after setting this bit to 1. 0:No operation 1: Abort
0	mw_enable	R/W	Controls Master Write transfers. Enabling should be made when setting the transfer address is completed. It will be automatically disabled as the master transfer finishes. Since Master Write operations cannot be disabled with this register, use the <mw_abort> bit if the Master Write transfer should be stopped. 0: Disable 1: Enable

17.4.1.7 UDFSDMACRDREQ(DMAC Read Request Register)

	31	30	29	28	27	26	25	24
bit symbol	dmardreq	dmardclr	-	-	-	-	-	-
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	dmardadr						-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	dmardreq	R/W	The bit for requesting read access to the DMAC registers. Setting this bit to 1 will make a read access to the address specified by <dmardadr>. When the read access is complete and the read value is stored in the UDFSDMACRDVL, this bit will be automatically cleared and the UDFSINTSTS<dmac_reg_rd> will be set to 1. 0: No operation 1: Issue a read request
30	dmardclr	R/W	The bit for forcibly clearing the register read access request associated with DMAC. Setting this bit to 1 will forcibly stop the register read access request by <dmardreq> and the value of <dmardreq> will be cleared to 0. After the forced clearing completes, this bit will be automatically cleared. 0: No operation 1: Issue forced clearing
29-8	−	R	Read as undefined.
7-2	dmardadr[5:0]	R/W	Sets the address of the register (upper 6 bits) to be read. It should be set in combination with the <dmardreq> mentioned above. Any one of the following addresses should be set: 0x48: Read the UDFSMWCADR. 0x58: Read the UDFSMRCADR. 0x88: read the UDFSTOUTCNT.
1-0	−	R	Read as undefined.

17.4.1.8 UDFSDMACRDVL(DMAC Read Value Register)

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

Bit	Bit Symbol	Type	Function
31-0	dmardata[31:0]	R	This register stores the data requested by UDFSDMACRDREQ. This register should not be accessed when the UDFSDMACRDREQ<dmardreq> is set to 1.

17.4.1.9 UDFSUDC2RDREQ(UDC2 Read Request Register)

	31	30	29	28	27	26	25	24
bit symbol	udc2rdreq	udc2rdclr	-	-	-	-	-	-
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	-	-	-	-	-	-	udc2rdadr	
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	udc2rdadr						-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	udc2rdreq	R/W	The bit for requesting read access to the UDC2 registers. Setting this bit to 1 will make a read access to the address set in the udc2rdadr bit. When the read access is complete and the read value is set to UDFSDMACRDVL, this bit will be automatically cleared and the UDINTSTS<int_udc2_reg_rd> bit of Interrupt Status register will be set to 1. During a write access to UDC2 registers, it works as a status bit which indicates the access being made to display the value of 1. Subsequent accesses to UDC2 registers should not be made while this bit is set to 1. 0: No operation 1: Issue a read request
30	udc2rdclr	R/W	The bit for forcibly clearing the read/write access request of UDC2 registers. Setting this bit to 1 will forcibly stop the register read request/UDC2 write access by udc2rdreq and the value of udc2rdreq will be 0. After the forced clearing completes, this bit will be automatically cleared to 0. When interrupted, the read and write values during the access will not be secured. 0 : No operation 1 : Issue forced clearing
29-10	−	R	Read as undefined.
9-2	udc2rdadr[7:0]	R/W	Set the address of the UDC2 register [9:2] to be read. Refer to "17.4.1.1 UDC2AB Register list" for information about the register address of the UDC2. The offset addresses in the above register table (0x0200 to 0x0334) are releant. Set it with the above udc2rdreq.
1-0	−	R	Read as undefined.

17.4.1.10 UDFSUDC2RDVL(UDC2 Read 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	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	udc2rdata							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	udc2rdata							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15-0	udc2rdata[15:0]	R	This register stores the data requested by UDFSDMACRDREQ. This register should not be accessed when the UDFSUDC2RDREQ <udc2rdreq> is set to 1.

17.4.1.11 UDFSARBTSET(Arbiter Setting Register)

	31	30	29	28	27	26	25	24
bit symbol	abt_en	-	-	abtmod	-	-	-	-
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	-	-	abtpri_w1		-	-	abtpri_w0	
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	abtpri_r1		-	-	abtpri_r0	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	abt_en	R/W	Enables the arbiter operation when making an access between DMAC and AHB. 0 should be set to this bit when setting the <abtmod>, <abtpri_w1>, <abtpri_w0>, <abtpri_r1> and <abtpri_r0> of this register. Be sure to set this bit to 1 before starting a DMA access. 0: Disable (DMA access not allowed) 1: Enable
30-29	−	R	Read as undefined.
28	abdmod	R/W	Sets the mode of arbiter. Write access is only available when the <abt_en> bit is set to 0. If 0 is set to this bit, access rights to the AHB bus will be given in a round-robin fashion regardless of the values set to each <abtpri_w1>, <abtpri_w0>, <abtpri_r1> and <abtpri_r0> bit. If 1 is set to this bit, access rights to the AHB bus will be given in accordance with the access priority based on the values set to each <abtpri _ w1>, <abtpri _ w0>, <abtpri _ r1> and <abtpri _ r0> bit. 0: Round-robin 1: Fixed-priority
27-14	−	R	Read as undefined.
13-12	abtpri_w1	R/W	Set the priority of DMA accesses for Master Write 1 when the fixed priority mode is selected. Write access is only available when the <abt_en> bit is set to 0. Priority ranges from 00 (highest) to 11 (lowest).
11-10	−	R	Read as undefined.
9-8	abtpri_w0	R/W	Set the priority of DMA accesses for Master Write 0 when the fixed priority mode is selected. Write access is only available when the <abt_en> bit is set to 0. Priority ranges from 00 (highest) to 11 (lowest).
7-6	−	R	Read as undefined.
5-4	abtpri_r1	R	Set the priority of DMA accesses for Master Read 1 when the fixed priority mode is selected. Write access is only available when the <abt_en> bit is set to 0. Priority ranges from 00 (highest) to 11 (lowest).
3-2	−	R	Read as undefined.
1-0	abtpri_r0	R/W	Set the priority of DMA accesses for Master Read 0 when the fixed priority mode is selected. Write access is only available when the <abt_en> bit is set to 0. Priority ranges from 00 (highest) to 11 (lowest).

Note: Be sure to set different priority values for the <abtpri_w1>, <abtpri_w0>, <abtpri_r1>, and <abtpri_r0> bits. If the same priority values are set, you will not be able to set 1 to <abt_en>.

(1) Relationship of DMAC and the priority area of the Arbiter setting register

Current UDC2AB specification supports one DMAC for Master Write (DMAC_W0) and one DMAC for Master Read (DMAC_R0). The second DMAC for Master Write (DMAC_W1) and the second DMAC for Master Read (DMAC_R1) are not supported.

Accordingly, setting priority for DMAC_W1 and DMAC_R1 has virtually no meaning, but you should be sure to set different priority values for the abtpri_w1, abtpri_w0, abtpri_r1, and abtpri_r0 bits as mentioned above.

There will be no problem to set values for the corresponding register areas of an unpackaged DMAC. The priority areas of Arbiter Setting register correspond with DMAC as shown below.

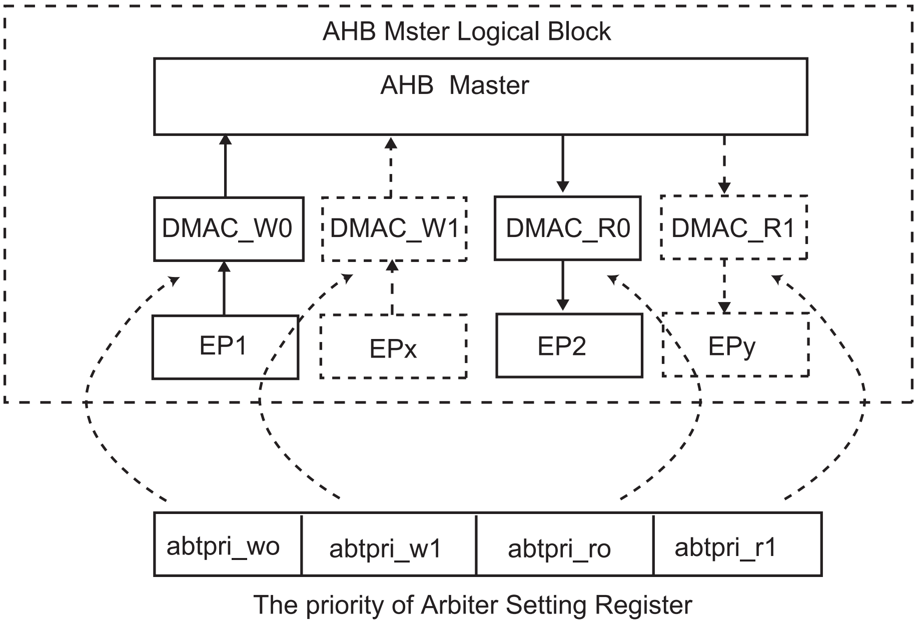

AHB Mster Logical Block
AHB Master
DMAC_W0
DMAC_W1
DMAC_R0
DMAC_R1
EP1
EPx
EP2
EPy
abtpri_wo
abtpri_w1
abtpri_ro
abtpri_r1
The priority of Arbiter Setting Register

Figure 17-8 Relationship between DMAC and priority areas

17.4.1.12 UDFSMWSADR(Master Write Start Address Register)

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

Bit	Bit Symbol	Type	Function
31-0	mwsadr[31:0]	R/W	Set the start address of Master Write transfer. However, as this master operation only supports address increments, values lower than the UDFSMWEADR should be set.

17.4.1.13 UDFSMWEADR(Master Write End Address Register)

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

Bit	Bit Symbol	Type	Function
31-0	mweadr[31:0]	R/W	Set the end address of Master Write transfer. However, as this master only supports address increments, values above the UDFSMWSADR should be set.

17.4.1.14 UDFSMWCADR(Master Write Current Address Register)

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

Bit	Bit Symbol	Type	Function
31-0	mwcadr[31:0]	R	Displays the addresses to which transfers from EPs to the Master Write buffers have been currently completed in Master Write transfers. This can be used in case a timeout interrupt has occurred or an error occurred during the transfer process. This address is incremented at the point when the data is set from the EP to the Master Write buffer, while the data will reside inside the target device or the Master Write buffer during the Master Write transfer process until the displayed address.

17.4.1.15 UDFSMWAHBADR(Master Write AHB Address Register)

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

Bit	Bit Symbol	Type	Function
31-0	mrsadr[31:0]	R	Displays the address where the transfer to the target device has completed in Master Write transfer. This can be used in case a timeout interrupt has occurred or an error occurred during the transfer process. This address is incremented at the point when the data is set to the target device, while the data will reside inside the target device or during the Master Write transfer process until the displayed address.

17.4.1.16 UDFSMRSADR(Master Read Start Address Register)

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

Bit	Bit Symbol	Type	Function
31-0	mrsadr[31:0]	R/W	Set the start address of Master Read transfer. However, as this master only supports address increments, values lower than the UDFSMWEADR should be set.

17.4.1.17 UDFSMREADR(Master Read End Address Register)

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

Bit	Bit Symbol	Type	Function
31-0	mreadr[31:0]	R/W	Set the end address of Master Read transfer. However, as this master only supports address increments, values above the UDFSMRSADR should be set.

17.4.1.18 UDFSMRCADR(Master Read Current Address Register)

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

Bit	Bit Symbol	Type	Function
31-0	mrcadr[31:0]	R	Displays the address to which transfers from the target device to the EP have been currently completed in Master Read transfers. This address is incremented at the point when the data is set from the Master Read buffer to the EP, while the data will reside inside the FIFO for the EP during the Master Read transfer process until the displayed address.

17.4.1.19 UDFSMRAHBADR(Master Read AHB Address Register)

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

Bit	Bit Symbol	Type	Function
31-0	mrahbadr[31:0]	R	Displays the address where the transfer from the target device to UDC2AB has completed in Master Read transfer. This address is incremented at the point when the data is set from the target device, while the data will reside inside the buffer or the FIFO for the EP during the Master Read transfer process until the displayed address.

17.4.1.20 UDFSPWCTL(Power Detect 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	wakeup_en	phy_remote_wkup	phy_resetb	suspend_x	phy_suspend	pw_detect	pw_resetb	usb_reset
After reset	0	0	1	1	0	0	1	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	wakeup_en	R/W	Set this bit to ’1’ if you want to shift the TMPM367FDFG to low-power consumption mode to stop CLK_H when the USB is suspended. If this bit is set to ’1’, the $\overline{\text{WAKEUP}}$ signal will be asserted to 0 asynchronously when the suspended status (<suspend_x>=1) is cancelled. This allows TMPM367FDFG to resuming from the low-power consumption mode using INTUSBWKUP. Refer to "17.5.7 Suspend / Resume" for more information about how to use this bit. 0: Do not assert the $\overline{\text{WAKEUP}}$ signal. 1:Assert the $\overline{\text{WAKEUP}}$ signal.
6	phy_remoto_wkup	R/W	This bit is used to perform the remote wakeup function of USB. Setting this bit to 1 makes it possible to assert the udc2_wakeup output signal (wakeup input pin of UDC2) to 1. However, since setting this bit to 1 while no suspension is detected by UDC2 (when suspend_x = 1) will be ignored (not to be set to 1), be sure to set it only when suspension is detected. It will be automatically cleared to 0 when resuming the USB is completed (when suspend_x is deasserted). See also "1.3.24.8 Suspend / Resume" for more information on using this bit. Refer to "17.5.7 Suspend / Resume" for more information about how to use this bit. 0: No operation 1: Wakeup
5	phy_resetb	R/W	Setting this bit to 0 will make the PHYRESET output signal asserted to 1. The PHYRESET signal can be used to reset PHY. Since this bit will not be automatically released, be sure to clear it to 1 after the specified reset time of PHY. 0: Reset asserted 1: Reset deasserted
4	suspend_x	R	Detects the suspend signal (a value of the suspend_x signal from UDC2 synchronized). 0: Suspended (<suspend_x> = 0) 1: Unsuspended (<suspend_x> = 1)
3	phy_suspend	R/W	Setting this bit to 1 will make the $\overline{\text{PHYSUSPEND}}$ output signal asserted to 0 (CLK_H synchronization). It can be used as a pin for suspending PHY. Setting this bit to 1 makes the UDC2 register and UDFSDMACRDREQ not accessible. It will be automatically cleared to 0 when resumed (when suspend_x of UDC2 is deasserted). Refer to "17.5.7 Suspend / Resume" for more information about how to use this bit. 0: Not suspended. 1: Suspended
2	pw_detect	R	Indicates the status of the VBUSPOWER input of the UDC2AB. 0: USB bus disconnected (VBUSPOWER = 0) 1: USB bus connected (VBUSPOWER = 1)
1	pw_resetb	R/W	Software reset for UDC2AB(See "17.5.1 Reset" for details). Setting this bit to 0 will make the PW_RESETB output pin asserted to 0. Resetting should be made while the master operation is stopped. Since this bit will not be automatically released, be sure to clear it. 0: Reset asserted. 1: Rest deasserted.
0	usb_reset	R	The value of the usb_reset signal from the UDC2 synchronized. 0: usb_reset = 0 1: usb_reset = 1

17.4.1.21 UDFSMSTSTS(Master 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	-	-	-	mrepempty	mrbfemp	mwbfemp	mrepdset	mwepdset
After reset	0	0	0	1	1	1	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Reset as undefined.
4	mrepempty	R	This is a register that indicates the EP for UDC2Rx is empty. Ensure that this bit is set to 1 when sending a NULL packet using the UDFSC2STSET<tx0>. (This bit is the eptx_empty input signal with CLK_H synchronization.) 0: Indicates the EP contains some data. 1: Indicates the EP is empty.
3	mrbfemp	R	Indicates whether or not the buffer for the Master Read DMA in UDC2AB is empty. 0: Indicates the buffer for the Master Read DMA contains some data. 1: Indicates the buffer for the Master Read DMA is empty.
2	mwbfemp	R	Indicates whether or not the buffer for the Master Write DMA in UDC2AB is empty. 0: Indicates the buffer for the Master Write DMA contains some data. 1: Indicates the buffer for the Master Write DMA is empty.
1	mrepdset	R	This bit will be set to 1 when the data to be transmitted is set to the Tx-EP of UDC2 by Master Read DMA transfer, making no room to write in the EP. It will turn to 0 when the data is transferred from UDC2 by the IN-Token from the host. While this bit is set to 0, DMA transfers to the EP can be made. (This bit is the eptx_dataset input signal with CLK_H synchronization.) 0: Data can be transferred into the EP. 1: There is no space to transfer data in the EP.
0	mwepdset	R	This bit will be set to 1 when the data received is set to the Rx-EP of UDC2. It will turn to 0 when the entire data was read by the DMA for Master Write. (This bit is the eprx_dataset input signal with CLK_H synchronization.) 0: No data exists in the EP. 1: There is some data to be read in the EP.

17.4.1.22 UDFSTOUTCNT(Timeout Count Register)

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

Bit	Bit Symbol	Type	Function
31-0	tmoutcnt[31:0]	R	This is used for debugging. Values of the timer can be read when the UDFSMWTOUT<timeout_en> is enabled. It will be decremented each time CLK_U is counted after the EP for Master Write (Rx-EP) becomes empty. This register cannot be read by directly specifying the address. In order to read it, set a value to the UDFSDMACRDREQ and then read the value from UDFSDMACRDVL.

17.4.2 UDC2 Register

17.4.2.1 UDC2 Registers

BaseAddress=0x4000_8000

Register name		Address(Base+)
UDC Address-State Register	UDFS2ADR	0x0200
UDC2 Frame Register	UDFS2FRM	0x0204
-	Reserved	0x0208
UDC2 Command Register	UDFS2CMD	0x020C
UDC2 bRequest-bmRequest Type Register	UDFS2BRQ	0x0210
UDC2 wValue register	UDFS2WVL	0x0214
UDC2 wIndex Register	UDFS2WIDX	0x0218
UDC2 wLength Register	UDFS2WLGTH	0x021C
UDC2 INT Register	UDFS2INT	0x0220
UDC2 INT EP Register	UDFS2INTEP	0x0224
UDC2 INT EP Mask Register	UDFS2INTEPMSK	0x0228
UDC2 INT RX DATA0 Register	UDFS2INTRX0	0x022C
UDC2 EP0 MaxPacketSize Register	UDFS2EP0MSZ	0x0230
UDC2 EP0 Status Register	UDFS2EP0STS	0x0234
UDC2 EP0 Datasize Register	UDFS2EP0DSZ	0x0238
UDC2 EP0 FIFO Register	UDFS2EP0FIFO	0x023C
UDC2 EP1 MaxPacketSize Register	UDFS2EP1MSZ	0x0240
UDC2 EP1 Status Register	UDFS2EP1STS	0x0244
UDC2 EP1 Datasize Register	UDFS2EP1DSZ	0x0248
UDC2 EP1 FIFO Register	UDFS2EP1FIFO	0x024C

BaseAddress=0x4000_8000

Register name		Address(Base+)
UDC2 EP2 MaxPacketSize Register	UDFS2EP2MSZ	0x0250
UDC2 EP2 Status Register	UDFS2EP2STS	0x0254
UDC2 EP2 Datasize Register	UDFS2EP2DSZ	0x0258
UDC2 EP2 FIFO Register	UDFS2EP2FIFO	0x025C
UDC2 EP3 MaxPacketSize Register	UDFS2EP3MSZ	0x0260
UDC2 EP3 Status Register	UDFS2EP3STS	0x0264
UDC2 EP3 Datasize Register	UDFS2EP3DSZ	0x0268
UDC2 EP3 FIFO Register	UDFS2EP3FIFO	0x026C
UDC2 EP4 MaxPacketSize Register	UDFS2EP4MSZ	0x0270
UDC2 EP4 Status Register	UDFS2EP4STS	0x0274
UDC2 EP4 Datasize Register	UDFS2EP4DSZ	0x0278
UDC2 EP4 FIFO Register	UDFS2EP4FIFO	0x027C
UDC2 EP5 MaxPacketSize Register	UDFS2EP5MSZ	0x0280
UDC2 EP5 Status Register	UDFS2EP5STS	0x0284
UDC2 EP5 Datasize Register	UDFS2EP5DSZ	0x0288
UDC2 EP5 FIFO Register	UDFS2EP5FIFO	0x028C
UDC2 EP6 MaxPacketSize Register	UDFS2EP6MSZ	0x0290
UDC2 EP6 Status Register	UDFS2EP6STS	0x0294
UDC2 EP6 Datasize Register	UDFS2EP6DSZ	0x0298
UDC2 EP6 FIFO Register	UDFS2EP6FIFO	0x029C
UDC2 EP7 MaxPacketSize Register	UDFS2EP7MSZ	0x02A0
UDC2 EP7 Status Register	UDFS2EP7STS	0x02A4
UDC2 EP7 Datasize Register	UDFS2EP7DSZ	0x02A8
UDC2 EP7 FIFO Register	UDFS2EP7FIFO	0x02AC
-	Reserved	0x02B0 to 0x32C
UDC2 INT NAK Register	UDFS2INTNAK	0x0330
UDC2 INT NAK MASK Register	UDFS2INTNAKMSK	0x0334
-	Reserved	0x0338 to 0x03FC

Note 1: Those shown as "Reserved" above and the area from 0x0400 to 0x0FFF are prohibited to access. Read / Write is prohibited to these areas.

Note 2: The registers are initialized by reset_x or USB_reset.

17.4.2.2 How to access the UDC2 register

The bits 15-0 of the AHB data bus of UDC2AB are connected to the UDC data bus.

The bit31-16 are read-only (read value: undefined).

Make a WORD (32-bit) access for both write and read. (However, a BYTE (8-bit) access may be made for Write accesses to the EPx_FIFO register. Details will be described later).

It will take some time to complete an access for both write and read.

Be sure to begin subsequent accesses after the previous UDC2 register access is completed, using the int_udc2_reg_rd interrupt. (You can also use the UDFSUDC2RDREQ<udc2rdreq> to confirm the access status when reading.)

- Write access

  When making a write access to the UDC2 register, write it directly in the relevant address.
- Read access

  When making a read access to the UDC2 register, use UDFSUDC2RDREQ and DFSUDC2RDVL.

  First, you set the address to access to the UDFSUDC2RDREQ and then read the data from the UDFSUCS2RDVL for reading. You cannot read the data directly from the address shown in the "17.4.2.1 UDC2 Registers".
- EPx_FIFO register

  When making a write access to the EPx_FIFO register, a lower 1-byte access may be required in UDC2 PVCI I/F. In such a case, make a BYTE access to the lower 1 byte for UDC2AB.

  If a lower 1-byte access is required when making a read access, make an access via UDFSUDC2RDREQ as usual and read the data from UDFSUDC2RDVL. In that case, the access to UDFSAUDC2RDVL can be either by WORD or BYTE.
- Reserved Register in UDC2

  Do not make any access to registers of EPs not supported by UDC2 to be connected and to "Reserved" registers. (In case those registers are accessed, the access from UDC2AB to UDC2 itself will take place. It will be a Dummy write to UDC2 in case of write accesses. In case of read accesses, the read data from UDC2 (udc2_rdata) will be an indefinite value and the indefinite value will be set to the UDFSUDC2RDVL.)
- Accesses when UDC2 is suspended

  When UDC2 is in the suspended status, register accesses to UDC2 become unavailable if the clock (= CLK_U) supply from USB clock control circuit is stopped. Make no register accesses to UDC2 in such cases. If the UDC2 register is accessed when the UDFSPWCTL<phy_suspend> is set to 1, an AHB error will be returned.

Access flow diagram for UDC2 register is shown below.

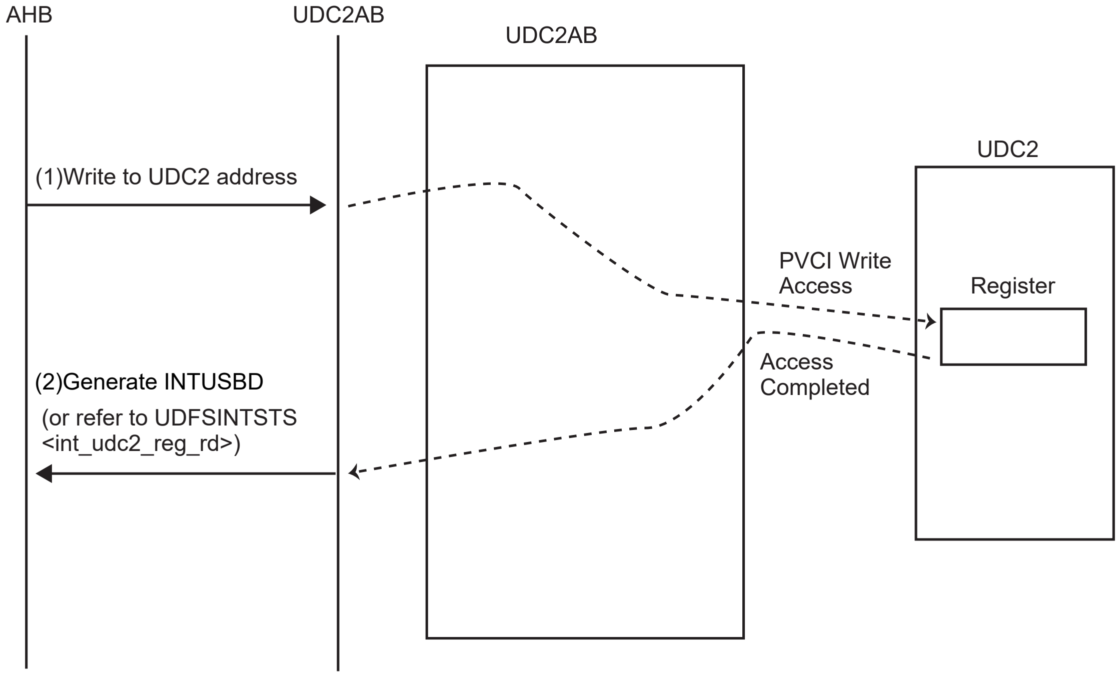

AHB
UDC2AB
UDC2AB
UDC2
(1)Write to UDC2 address
PVCI Write Access
Register
Access Completed
(2)Generate INTUSBD
(or refer to UDFSINTSTS <int_udc2_reg_rd>)

Figure 17-9 Write Access Flow Diagram of the UDC2 Register

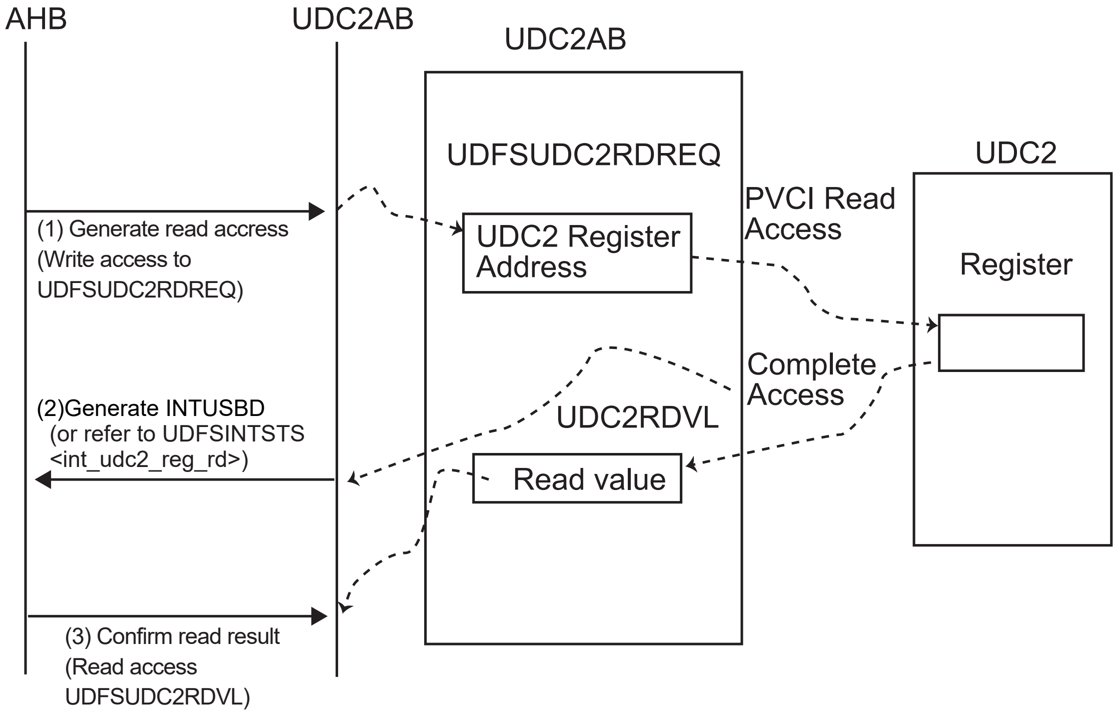

AHB
UDC2AB
UDC2AB
UDFSUDC2RDREQ
UDC2
PVCI Read Access
UDC2 Register Address
Register
(1) Generate read accress
(Write access to UDFSUDC2RDREQ)
Complete Access
UDC2RDVL
(2)Generate INTUSBD
(or refer to UDFSINTSTS <int_udc2_reg_rd>)
Read value
(3) Confirm read result
(Read access UDFSUDC2RDVL)

Figure 17-10 Read Access Flow Diagram of the UDC2 Register

17.4.2.3 UDFS2ADR(Address-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	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	stage_err	ep_bi_mode	cur_speed		suspend	configured	addressed	default
After reset	0	0	0	0	0	0	0	1
	7	6	5	4	3	2	1	0
bit symbol	-	dev_adr						
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15	stage_err	R/W	Indicates whether Control transfers finished normally up to the STATUS-Stage. 1 will be set when the Setup-Token is received in DATA-Stage/STATUS-Stage or in the case of "STALL" transmission. When set, it will be cleared when the next Control transfer has been finished normally. 0: Other than above conditions. 1: Received the Setup-Token in DATA-Stage/STATUS-Stage or "STALL" transmission.
14	ep_bi_mode	R/W	Selects whether to use the EP bidirectionally as a driver. Setting this bit to 1 will enable an EP number to be used bidirectionally in USB communication. 0: Single direction 1: Dual direction
13-12	cur_speed[1:0]	R	Indicates the present transfer mode on the USB bus. 00: Reserved 01: Full-Speed 10: Reserved 11: Reserved
11	suspend	R	Indicates whether or not UDC2 is in suspended state. 0: Normal 1: Suspend
10	configured	R/W	Set the present device state of UDC2. This should be set in accordance with the request received from the host. Please note that you should not set 1 to more than one bit. 001: default (to be set when the DeviceAddress = 0 was specified by the Set_address request in Default / Address state (this will be set by the hardware when USB_RESET is received)) 010: addressed (to be set when ConfigurationValue = 0 was specified by the Set_configuration request after the Set_address request finished normally and in the Address/Configured state). 100: configured (to be set when the Set_configuration request is received).
9	addressed	R/W	
8	default	R/W	
7	−	R	Read as undefined.
6-0	dev_adr[6:0]	R/W	Sets the device address assigned by the host. The device address should be set after Set_address has finished normally (after STATUS-Stage finished normally).

17.4.2.4 UDFS2FRM(Frame 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	create_sof	-	f_status		-	frame		
After reset	0	0	1	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	frame							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15	create_sof	R/W	Sets whether to generate the SOF flag internally when the SOF from the host is unavailable due to a bus error. This should be set if you wish to synchronize frames by SOF in Isochronous transfers. If enabled, the internal frame time counter will operate and the SOF flag will be output even when the SOF-Token could not be received successfully. 0: Generates no flag 1: Generates a flag
14	−	R	read as undefined.
13-12	f_status[1:0]	R	Indicates the status of the frame number. 00: Before: Will be set if the Micro SOF/SOF was not received when 1frame-time (Full-Speed:1 ms) has passed after receiving the Micro SOF/SOF when <create_sof> is enabled. In the Frame register, the frame number received in the last Micro SOF/SOF has been set. 01: Valid: Will be set when the Micro SOF/SOF was received. Indicates a valid frame number is set in the Frame register. 10: Lost: Indicates that the frame number maintained by the host is not synchronized with the value of UDFS2FRM. Accordingly, this will be set in the following cases: 1. When the system was reset or suspended. 2. If the next Micro SOF/SOF was not received when 2 frame-time (Full-Speed: 1 x 2 ms) has passed after receiving the previous Micro SOF/SOF when <create_sof> is enabled. Also note that transition to the Lost status only happens after the system was reset or when it is suspended if <create_sof> is disabled.
11	−	R	Read as undefined.
10-0	frame[10:0]	R	Indicates the frame number when SOF is received. This will be valid when <f_status> is "Valid". Should not be used if <f_status> is "Before" or "Lost" as correct values are not set.

17.4.2.5 UDFS2CMD(Command 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	int_toggle	-	-	-	rx_nulpkt_ep			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ep				com			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15	int_toggle	R/W	Makes the DATA-PID toggle when Handshake is not received in Interrupt-IN transfers. 0: Do not toggle when not received 1: Toggle when not received as well
14-12	−	R	Read as undefined.
11-8	rx_nulpkt_ep [3:0]	R	Indicates the receiving EP when Zero-Length data is received. When the INT_RX_ZERO flag is asserted, read this bit to check to which EP it was asserted. Once Zero-Length data is received and the EP number is retained, the value of this register will be retained until Zero-Length data is received next time or hardware reset is made. If there is more than one EP of OUT direction, this bit will be renewed each time Zero-Length data is received. In that case, UDFS2INTRX0 can be used to identify which EP has received the data.
7-4	ep[3:0]	R/W	Sets the EP where the command to be issued will be valid. (Do not specify an EP not existing.)
3-0	com[3:0]	R/W	Sets the command to be issued for the EP selected in ep[3:0]. Refer to "17.2.2.3 Commands to EP" for more information. 0x0: Reserved 0x1: Setup_Fin 0x2: Set_DATA0 0x3: EP_Reset 0x4: EP_Stall 0x5: EP_Invalid 0x6: Reserved 0x7: EP_Disable 0x8: EP_Enable 0x9: All_EP_Invalid 0xA: USB_Ready 0xB: Setup_Received 0xC: EP_EOP 0xD: EP_FIFO_Clear 0xE: EP_TX_0DATA 0xF: Reserved

17.4.2.6 UDFS2BRQ(bRequest-bmRequest Type 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	request							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	dir	req_type		recipient				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined
15-8	request[7:0]	R	Indicates the data of the second byte received with the Setup-Token (bRequest field).
7	dir	R	Indicates the data of the first byte received with the Setup-Token (bmRequestType field). Direction of Control transfers. 0: Control-WR transfer 1: Control-RD transfer
6-5	req_type[1:0]	R	Type of requests 00: Standard request 01: Class request 10: Vendor request 11: Reserved
4-0	recipient[4:0]	R	Requests are received by 0_0000: Device 0_0001: Interface 0_0010: EP 0_0011: etc. 0_0100-1_1111: Reserved

17.4.2.7 UDFS2WVL(wValue 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	value							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	value							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15-8	value[15:8]	R	Indicates the data of the fourth byte received with the Setup-Token (wValue (High) field).
7-0	value[7:0]	R	Indicates the data of the third byte received with the Setup-Token (wValue (Low) field).

17.4.2.8 UDFS2WIDX(wIndex 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	index							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	index							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined
15-8	index[15:8]	R	Indicates the data of the sixth byte received with the Setup-Token (wIndex (High) field).
7-0	index[7:0]	R	Indicates the data of the fifth byte received with the Setup-Token (wIndex (Low) field).

17.4.2.9 UDFS2WLGTH(wLength 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	length							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	length							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined
15-8	length[15:8]	R	Indicates the data of the eighth byte received with the Setup-Token (wLength (High) field).
7-0	length[7:0]	R	Indicates the data of the seventh byte received with the Setup-Token (wLength (Low) field).

17.4.2.10 UDFS2INT(INT 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	m_nak	m_ep	m_ep0	m_sof	m_rx_data0	m_status	m_status_nak	m_setup
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	i_nak	i_ep	i_ep0	i_sof	i_rx_data0	i_status	i_status_nak	i_setup
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15	m_nak	R/W	Sets whether or not to output <i_nak> to the INT_NAK pin. 0: output 1: no output
14	m_ep	R/W	Sets whether or not to output <i_ep> to INT_EP pin. 0: output 1: no output
13	m_ep0	R/W	Sets whether or not to output <i_ep0> to INT_EP0 pin. 0: output 1: no output
12	m_sof	R/W	Sets whether or not to output <i_sof> to INT_SOF pin. 0: output 1: no output
11	m_rx_data0	R/W	Sets whether or not to output <i_rx_data0> to INT_RX_ZERO pin. 0: output 1: no output
10	m_status	R/W	Sets whether or not to output <i_status> to INT_STATUS pin. 0: output 1: no output
9	m_status_nak	R/W	Sets whether or not to output <i_status_nak> to INT_STATUS_NAK pin. 0: output 1: no output
8	m_setup	R/W	Sets whether or not to output <i_setup> to INT_SETUP pin. 0: output 1: no output
7	i_nak	R/W	This will be set to 1 when NAK is transmitted by EPs except EP0. (EPs to which you wish to output the INT_NAK flag can be selected using UDFS2INTNAKMASK). Writing 1 to this bit will make each bit of UDFS2INTNAK cleared to 0.
6	i_ep	R/W	This will be set to 1 when transfers to EPs other than EP0 have successfully finished (EPs to which you wish to output the flag can be selected using UDFS2INTNAKMASK). Writing 1 to this bit will make each bit of UDFS2INTEP cleared to 0.
5	i_ep0	R/W	This will be set to 1 when the transfer to EP0 has successfully finished.
4	i_sof	R/W	This will be set to 1 when the SOF-token is received or after 1 frame-time was counted in the create_sof mode.
3	i_rx_data0	R/W	This will be set to 1 when Zero-Length data is received. (EPs to which you wish to output the flag can be selected using UDFS2INTNAKMASK). Writing 1 to this bit will make each bit of UDFSINTTRX0 cleared to 0. This will not be set to 1 when Zero-Length data is received in the STATUS-Stage of Control-RD transfers.

Bit	Bit Symbol	Type	Function
2	i_status	R/W	This will be set to 1 when the STATUS-Stage has successfully finished in Control transfers at EP0. (This will be set to 1 when Zero-Length data is received in the STATUS-Stage and successfully finished in Control-RD transfers, and when Zero-Length data is transmitted in the STATUS-Stage and successfully finished in Control-WR transfers.)
1	i_status_nak	R/W	This will be set to 1 when the packet of STATUS-Stage is received in the Control-RD transfers at EP0. When this bit was set which means the DATA-Stage has finished, set the "Setup-Fin" command by the UDFS2CMD to make the stage of UDC2 proceed to the STATUS-Stage. When receiving the data having the size of an integral multiple of MaxPacketSize (64 bytes) in the DATA-Stage of Control-WR transfers, Zero-Length data may be received to indicate the end of the DATA-Stage. After that, as the end of the DATA-Stage can be recognized by this i_status_nak when receiving the In-token in the STATUS-Stage, make UDC2 proceed to the STATUS-Stage.
0	i_setup	R/W	This will be set to 1 when the Setup-Token was received in Control transfers at EP0.

17.4.2.11 UDFS2INTEP(INT_EP 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	i_ep7	i_ep6	i_ep5	i_ep4	i_ep3	i_ep2	i_ep1	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15-8	Reserved	R/W	Write as "0".
7-1	i_ep7 to i_ep1	R/W	Flags to indicate the transmitting/receiving status of EPs (except for EP0). The relevant bit will be set to 1 when the transfer to EPs other than EP0 has successfully finished. (EPs to which you wish to output the int_ep flag can be selected using UDFS2INTEPMSK.). 0: No data transmitted/received. 1: Some data transmitted/received
0	−	R/W	Read as undefined.

17.4.2.12 UDFS2INTEPMSK(INT_EP_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	m_ep7	m_ep6	m_ep5	m_ep4	m_ep3	m_ep2	m_ep1	m_ep0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15-8	Reserved	R/W	Write as "0".
7-0	m_ep7 to m_ep0	R/W	Mask control of flag output. 0 : output 1: no output Sets whether or not to output flags of UDFS2INTEP and UDFS2INTRX0 to the int_ep pin and the int_rx_zero pin respectively. When an EP is masked, each bit of UDFS2INTEP will be set when the transfer of the relevant EP has successfully finished, but the int_ep pin will not be asserted. Similarly, when an EP is masked, each bit of UDFS2INTRX0 will be set when Zero-Length data is received at the relevant EP, but the int_rx_zero pin will not be asserted. However, bit 0 is only valid for UDFS2INTRX0.

(1) How to use UDFS2INT / UDFS2INTEP / UDFS2INTEPMSK

An example of using UDFS2INT / UDFS2INTEP / UDFS2INTEPMSK is provided for EP1 to 3.

1. When using EP 1 and EP 2 with DMA (EP I/F) and using only EP3 via PVCI-I/F

UDFS2INT	<i_ep>	Used as the interrupt source of EP3. This bit is also used when clearing.
	<m_ep>	Used as the mask of the interrupt source of EP3.
UDFS2INTEP	<i_ep1>	Don't care
	<i_ep2>	Don't care
	<i_ep3>	Don't care
UDFS2INTEPMSK	<m_ep1>	Set 1 to mask the bit.
	<m_ep2>	Set 1 to mask the bit.
	<m_ep3>	Write 0.

2. When using EP2 and EP3 via PVCI-I/F and using EP1 with DMA

After initialization, set 1 to UDFS2INTEPMSK of the EP to be used with DMA to mask it. When making interrupt responses for more than one EPs, be sure to use UDFS2INTEP. Ignore UDFS2INT<i_ep> and always enable <m_ep> as 0.

Do not clear the source using UDFS2INT<i_ep>. After the interrupt has occurred, you need to check UDFS2INT and UDFS2INTEP to determine the source. When clearing the source, use each source bit of UDFS2INT interrupt to clear it.

UDFS2INT	<i_ep>	Write as 0.
	<m_ep>	Write as 0.
UDFS2INTEP	<i_ep1>	Don’t care
	<i_ep2>	Used as the interrupt source of EP2. This bit is also used when clearing.
	<i_ep3>	Used as the interrupt source of EP3. This bit is also used when clearing.
UDFS2IN-TEPMSK	<m_ep1>	Set 1 to mask the bit.
	<m_ep2>	Used as the mask of the interrupt source of EP2. Write as "0".
	<m_ep3>	Used as the mask of the interrupt source of EP3. Write as "0".

17.4.2.13 UDFS2INTRX0(INT_RX_DATA0 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	rx_d0_ep7	rx_d0_ep6	rx_d0_ep5	rx_d0_ep4	rx_d0_ep3	rx_d0_ep2	rx_d0_ep1	rx_d0_ep0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15-8	Reserved	R/W	Write as "0".
7-0	rx_d0_ep7 to rx_d0_ep0	R/W	Flags for indicating Zero-Length data received at EP 0:No Zero-Length data received 1:Zero-Length data received The relevant bit will be set to 1 when EPs have received Zero-Length data. (EPs to which you wish to output the int_rx_zero flag can be selected using UDFS2INTEPMSK) For bit 0 (EP 0), it will be set to 1 only when Zero-Length data is received in the DATA-Stage while processing the request. Since it will not be set when Zero-Length data is received in the STATUS-Stage, use the int_status flag.

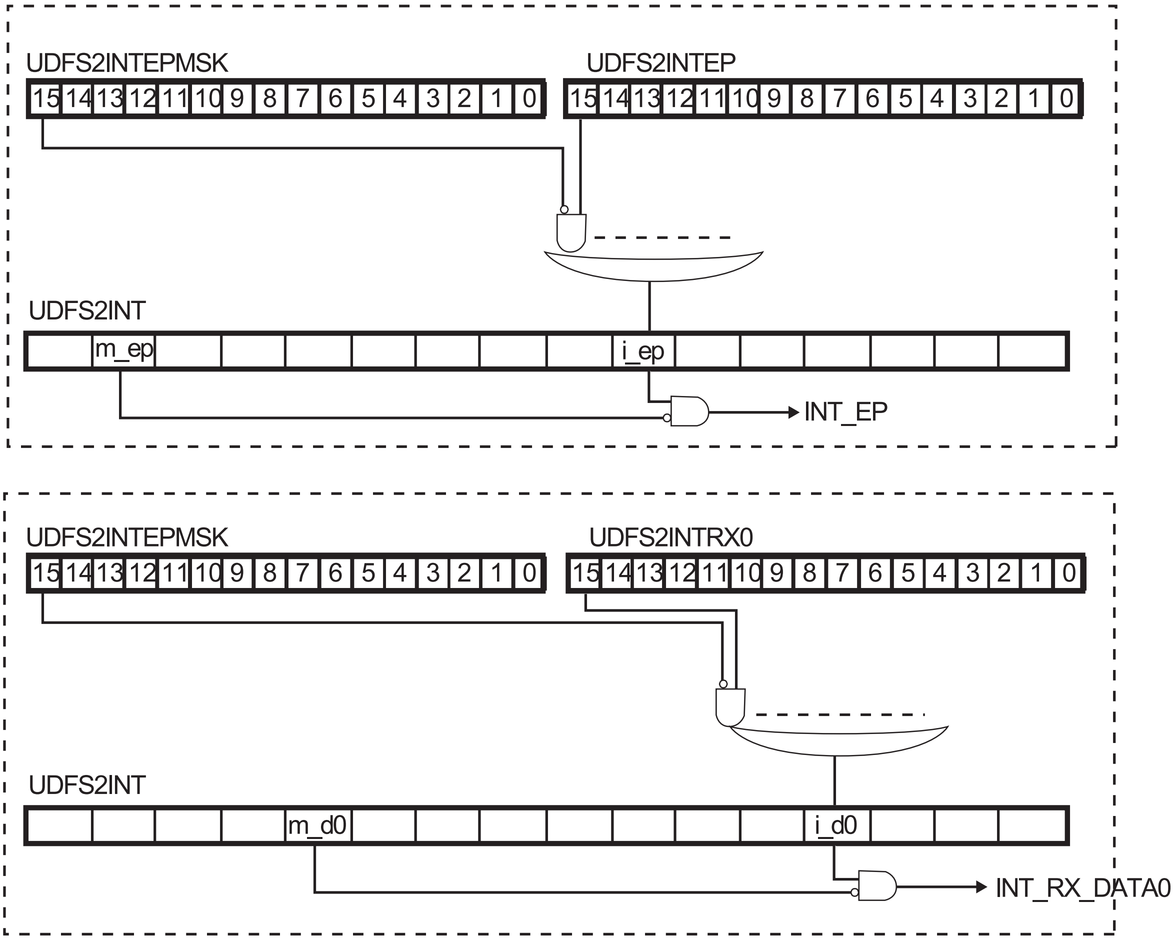

UDFS2INTEPMSK
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
UDFS2INTEP
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
UDFS2INT
m_ep
i_ep
INT_EP
UDFS2INTEPMSK
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
UDFS2INTRX0
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
UDFS2INT
m_d0
i_d0
INT_RX_DATA0

Figure 17-11 Interrupt Status and Mask Register

17.4.2.14 UDFS2INTNAK(INT_NAK 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	i_ep7	i_ep6	i_ep5	i_ep4	i_ep3	i_ep2	i_ep1	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15-8	Reserved	R/W	Write as "0",
7-1	i_ep7 to i_ep1	R/W	Flags to indicate the status of transmitting NAK at EPs (except for EP0) 0: No NAK transmitted 1: NAK transmitted The relevant bit will be set to 1 when NAK is transmitted by EPs other than EP0. (EPs to which you wish to output the INT_NAK flag can be selected using UDFS2INTEPMSK.)
0	−	R	Read as undefined.

17.4.2.15 UDFS2INTNAKMSK(INT_NAK_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	m_ep7	m_ep6	m_ep5	m_ep4	m_ep3	m_ep2	m_ep1	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15-8	Reserved	R/W	Write as "0".
7-1	m_ep7 to m_ep1	R/W	Mask control of flag output 0: output 1: no output Sets whether or not to output flags of UDFS2INTNAK to the int_nak pin respectively. When EPs are masked, each bit of UDFS2INTNAK will be set when NAK is transmitted in the transfer of the relevant EP, but the int_nak pin will not be asserted.
0	−	R	Read as undefined.

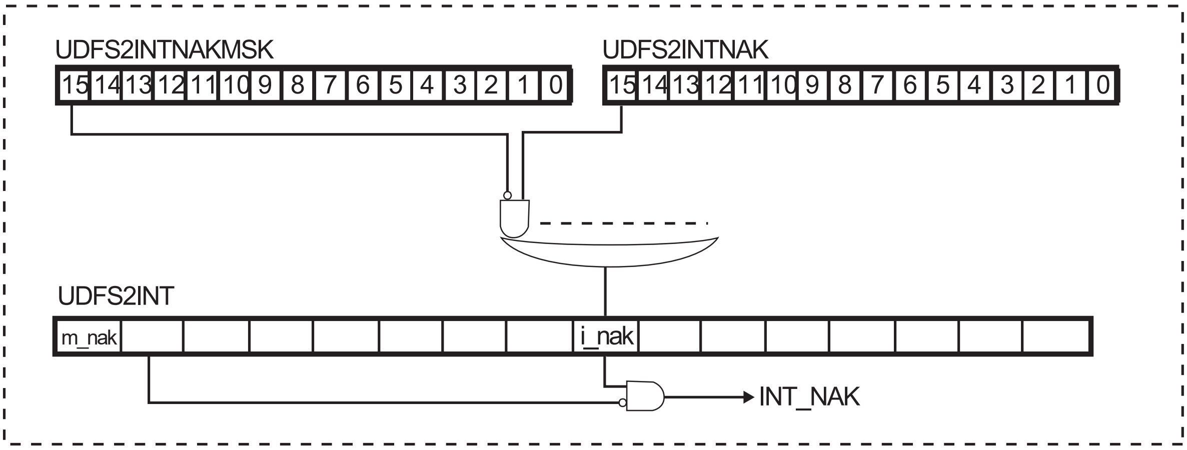

UDFS2INTNAKMSK
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
UDFS2INTNAK
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
UDFS2INT
m_nak
i_nak
INT_NAK

Figure 17-12 Interrupt and Status Register

17.4.2.16 UDFS2EP0MSZ(EP0_MaxPacketSize 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	tx_0data	-	-	dset	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	max_pkt						
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as defined.
15	tx_0data	R	When the "EP_TX_0DATA" command is issued to EP0 by UDFS2CMD, this bit will be set to 1 which will be cleared to 0 after the Zero-Length data has been transmitted.
14-13	−	R	Read as defined.
12	dset	R	Indicates the status of UDFS2EP0FIFO. It will be cleared to 0 when the Setup-Token is received. 0: No valid data exists 1: Valid data exists
11-7	−	R	Read as "0".
6-0	max_pkt[6:0]	R/W	Sets MaxPacketSize of EP0.

17.4.2.17 UDFS2EP0STS(EP0_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	ep0_mask	-	toggle		status			-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15	ep0_mask	R	Will be set to 1 after the Setup-Token is received. Will be cleared to 0 when the "Setup_Received" command is issued. No data will be written into the UDFS2EP0FIFO while this bit is 1. 0: Data can be written into UDFS2EP0FIFO. 1: Data can not be written into UDFS2EP0FIFO.
14	−	R	Read as undefined.
13-12	toggle[1:0]	R	Indicates the present toggle value of EP. 00: DATA0 01: DATA1 10: Reserved 11: Reserved
11-9	status[2:0]	R	Indicates the present status of EP0. It will be cleared to "Ready" when the Setup-Token is received. 000: Ready (Indicates the status is normal) 001: Busy (To be set when returned "NAK" in the STATUS-Stage) 010: Error (To be set in case of CRC error in the received data, as well as when timeout has occurred after transmission of the data) 011: Stall (Returns "STALL" when data longer than the Length was requested in Control-RD transfers) 100 to 111: Reserved
8-0	−	R	Read as undefined.

17.4.2.18 UDFS2EP0DSZ(EP0_Datasize 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	-	size						
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as undefined.
6-0	size[6:0]	R	Indicates the number of valid data bytes stored in UDFS2EP0FIFO. It will be cleared to when the Setup-Token is received.

17.4.2.19 UDFS2EP0FIFO(EP0_FIFO 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	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	−	R	Read as undefined.
15-0	data[15:0]	R/W	Used for accessing data from PVCI-I/F to EP0. For the method of accessing this register, see "17.7.1.1 Control-RD transfer" , "17.7.1.2 Control-WR transfer (without DATA-Stage)" and "17.7.1.3 Control-WR transfer (with DATA-Stage)". The data stored in this register will be cleared when the request is received (when the INT_SETUP interrupt is asserted).

17.4.2.20 UDFS2EPxMSZ(EPx_MaxPacketSizeRegister)

	31	30	29	28	27	26	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	tx_0data	-	-	dset (note1)	-	max_pkt		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	max_pkt							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15	tx_0data	R	When the "EPx_TX_0DATA" command is issued to EPx by UDFS2CMD or Zero-Length data has been set at EP-I/F, this bit will be set to 1. It will be cleared to 0 after the Zero-Length data has been transmitted.
14-13	−	R	Read as undefined.
12	dset	R	Indicates the status of EPx_FIFO. 0: No valid data exists 1: Valid data exists
11	−	R	Read as undefined.
10-0	max_pkt[10:0]	R/W	Sets MaxPacketSize of EPx. Set this when configuring the EP when Set_Configuration and Set_Interface are received. Set an even number for a transmit EP. On USB, when MaxPacketSize of a transmit EP is an odd number, set an even number to max_pkt and make the odd number of accesses to the EP. (For instance, set 1024 to max_pkt when the MaxPacketSize should be 1023 bytes.) Note: For details, refer to "17.9.2 Appendix B About Setting an Odd Number of Bytes as MaxPacketSize".

Note 1: The initial value of <dset> after reset is 1 when the EPx is a transmit EP, while it is 0 when the EPx is a receive EP.

Note 2: The initial value of <dset > after USB_RESET is 1 when the EPx is a transmit EP, while it is "Retain" when the EPx is a receive EP.

Note 3: x=1 to 7

17.4.2.21 UDFS2EPxSTS(EPx_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	pkt_mode	bus_sel	toggle		status			disable
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	dir	-	-	-	t_type		num_mf	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as undefined.
15	pkt_mode	R/W	Selects the packet mode of EPx. Selecting the Dual mode makes it possible to retain two pieces of packet data for the EPx. 0: Single mode 1: Dual mode
14	bus_sel	R/W	Select the bus to access to the FIFO of EPx. 0: Common bus access 1: Direct access
13-12	toggle[1:0]	R	Indicates the present toggle value of EPx. 00: DATA0 01: DATA1 10: DATA2 11: MDATA
11-9	status[2:0]	R	Indicates the present status of EPx. By issuing EP_Reset from UDFSCMD, the status will be "Ready." 000: Ready (Indicates the status is normal) 001: Reserved 010: Error (To be set in case a receive error occurred in the data packet, or when timeout has occurred after transmission. However, it will not be set when "Stall" or "Invalid" has been set.) 011: Stall (To be set when "EP-Stall" was issued by UDFS2CMD.) 100 to 110: Reserved 111: Invalid (Indicates this EP is invalid)
8	disable	R	Indicates whether transfers are allowed for EPx. If "Not Allowed," "NAK" will be always returned for the Token sent to this EP. 0: Allowed 1: Not Allowed
7	dir	R/W	Sets the direction of transfers for this EP. 0: OUT (Host-to-device) 1: IN (Device-to-host)
6-4	−	R	Read as undefined.
3-2	t_type[1:0]	R/W	Sets the transfer mode for this EP. 00: Control 01: Isochronous 10: Bulk 11: Interrupt
1-0	num_mf[1:0]	R/W	When the Isochronous transfer is selected, set how many times the transfer should be made in the frames. 00: 1-transaction 01: 2-transaction 10: 3-transaction 11: Reserved

Note 1: Setting for this register should be made when configuring the EP when Set_Configuration and Set_Interface are received.

Note 2: x=1 to 7

Note 3: Each EP depend on the product specification. For EP1, EP3, EP5, EP7 which is fixed for IN transfers, <dir> can be set to "1" only. For EP2, EP4, EP6 which is fixed for OUT transfers, dir can be set to "0" only.

17.4.2.22 UDFS2EPxDSZ(EPx_Datasize 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	-	-	-	-	-	size		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	size							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-11	−	R	Read as undefined.
10-0	size[10:0]	R	Indicates the number of valid data bytes stored in EP1_FIFO. In the Dual Packet mode, the number of data bytes to be accessed first will be shown.

Note: x=1 to 7

17.4.2.23 UDFS2EPxFIFO(EPx_FIFO 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	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	−	R	0.
15-0	data[15:0]	R/W	Used for accessing data from PVCI-I/F to EPx.

Note: x=1 to 7

17.5 Description of UDC2AB operation

17.5.1 Reset

UDC2AB supports software reset by the UDFSPWCTL<pw_resetb>.

It also supports master channel reset (UDFSMSTSET<mr_reset>/<mw_reset>) for DMAC master transfers.

- Software reset (UDFSPWCTL<pw_resetb>)

  Some bits of each register are initialized by hardware reset but not initialized by software reset with the values retained. As details are provided in the descriptions of each register, refer to "17.4.1.1 UDC2AB Register list".

  When the USB bus power is detected, make software reset as initialization is needed.
- Master channel reset (UDFSMSTSET<mr_reset><mw_reset>

  While the <mw_reset> bit is provided for the Master Write transfer block and the <mr_reset> bit for the Master Read transfer block, only the relevant master blocks are initialized and the UDC2AB register will not be initialized. For more information on using each reset, see "17.4.1.6 UDFSMSTSET(DMAC Setting Register)".

17.5.2 Interrupt Signals

There are two interrupt output signals of UDC2AB, INTUSBD and INTUSBWKUP.

17.5.2.1 INTUSBD Interrupt Signal

Interrupt output signal of INTUSBD consists of interrupts generated by UDC2 and that generated by other sources.

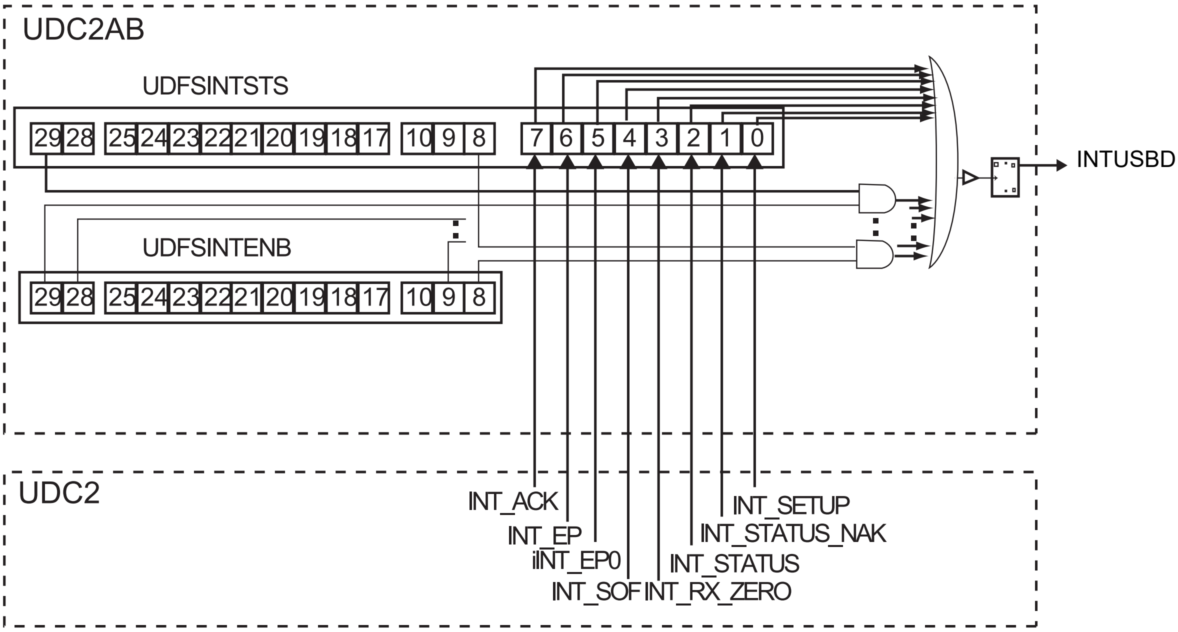
Once the interrupt condition is met, UDC2AB sets the corresponding bit of its UDFSINTSTS. When that bit is set, INTUSBD will be asserted if the relevant bit of UDFSINTENB has been set to "Enable."

When the relevant bit of UDFSINTENB has been set to "Disable," 1 will be set to the corresponding bit of UDFSINTSTS while INTUSBD will not be asserted.

When the relevant bit of UDFSINTENB is set to "Enable" with UDFSINTSTS set, INTUSBD will be asserted immediately after the setting is made.

Initial values for UDFSINTENB are all 0 (Disable).

Interrupt output signal of INTUSBD will not be generated while CLK_H is stopped.


UDC2AB
UDFSINTSTS
29 28 25 24 23 22 21 20 19 18 17 10 9 8 7 6 5 4 3 2 1 0
INTUSBD
UDFSINTENB
29 28 25 24 23 22 21 20 19 18 17 10 9 8
UDC2
INT_ACK
INT_EP
iINT_EP0
INT_SOF
INT_RX_ZERO
INT_STATUS
INT_STATUS_NAK
INT_SETUP

Note) The UDC2 flag is masked by UDFS2INT.

Figure 17-13 Relationship of INTUSBD and registers

17.5.2.2 INTUSBWKUP Interrupt

INTUSBWKUP interrupt occurs at the falling edge of the $\overline{\text{WAKEUP}}$ output signal.

WAKEUP will be asserted when the following conditions match:UDFSPWCTL<wakeup_en> is 1 and the suspended condition is cancelled (UDFSPWCTL<suspend_x>=1). WAKEUP will be asserted when VBUS is disconnected (VBUSPOWER=0) as well.

INTUSBWKUP interrupt occurs regardless of the status of CLK_H.

17.5.3 Operation Sequence

The operation sequence of UDC2AB is as follows:

1. Hardware reset

2. Set the interrupt signal

   Configure the INTUSBD interrupt, the INTUSBWKUP interrupt and the USBDPON interrupt.

3. VBUS detection (connect) and reset

   Refer to "17.5.5.2 Sequence of USB Bus Power (VBUS) Connection/Disconnection" and "17.5.1 Reset" for details.

4. USB enumeration response

   Refer to "17.6 USB Device Response" for details.

5. Master Read / Master Write transfer
   a. Master Read transfer

      Make a Master Read transfer corresponding to the receiving request from the USB host. Refer to "17.5.4.1 Master Read transfer" for details.
   b. Master Write transfer

      Make a Master Write transfer corresponding to the sending request from the USB host. Refer to "17.5.4.2 Master Write transfer" for details.

6. VBUS detection (disconnect)

   The USB bus power supply may be disconnected at any time.

   Refer to "17.5.5.2 Sequence of USB Bus Power (VBUS) Connection/Disconnection" for details.

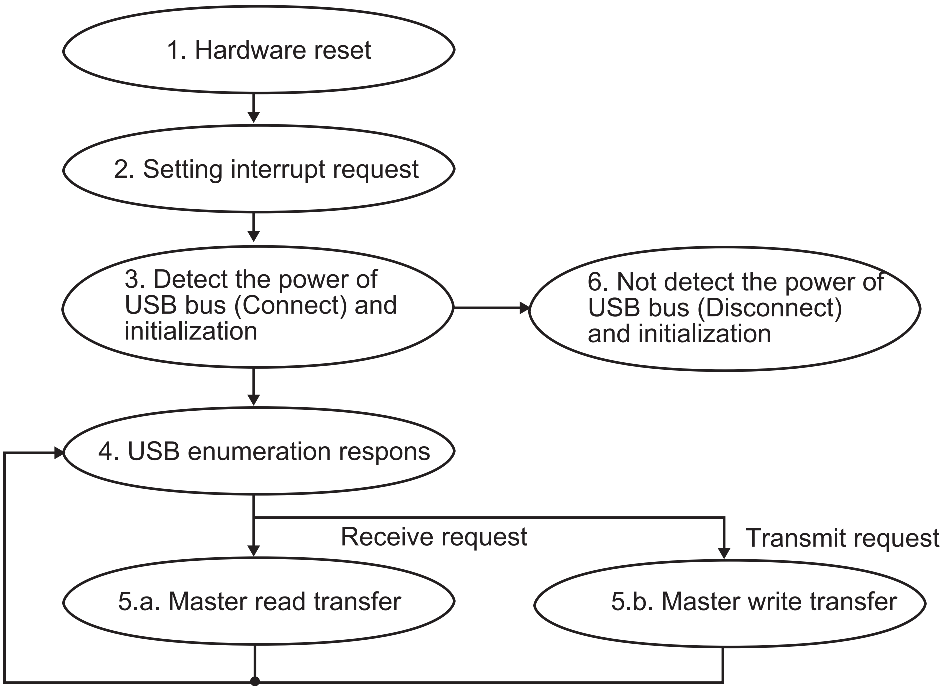

1. Hardware reset
2. Setting interrupt request
3. Detect the power of USB bus (Connect) and initialization
6. Not detect the power of USB bus (Disconnect) and initialization
4. USB enumeration respons
Receive request
Transmit request
5.a. Master read transfer
5.b. Master write transfer

Figure 17-14 Operation Sequence

17.5.4 Master Transfer Operation

This section describes the master transfer operation of UDC2AB.

When you start a master transfer, be sure to set the transfer setting of the relevant EP of UDC2 (UDFS2EPxSTS<bus_sel>) to the direct access mode. It is prohibited to start DMAC when it is set to "Common bus access."

17.5.4.1 Master Read transfer

(1) Master Read mode

There are two modes of the master read mode: EOP enable mode and EOP disable mode.

(a) EOP enable mode

Master Read transfers when UDC2STSET<eopb_enable> is set to 1 (Master Read EOP enable) are described here. Master Read operations will be as follows:

1. Set UDFSMWSADR and UDFSMWEADR.
2. Set the bits associated to the master read operation of UDFSMSTSET and set 1 to <mr_enable>.
3. UDC2AB starts the data transfer to the EP of UDC2. UDC2 transfers the data to the IN token from the USB host.
4. When the Master Read transfer reaches the Master Read end address, UDC2AB asserts the mr_end_add interrupt.
5. After the handling by the software ended, return to 1.

- About Short packets

  If the transfer size (Master Read End Address - Master Read Start Address Å{1) is not the same size as the Max packet size, the last IN transfer will be the transfer of short packets.

Example: In case Master Read transfer size 139 bytes and the Max packet size 64 bytes.

Transfer will take place in:

1st time		2nd time		3rd time
64 bytes	→	64 bytes	→	11 bytes

- About mr_end_add interrupt

  The mr_end_add interrupt occurs when the data transfer to the UDC2 EP is finished. In order to confirm whether the entire data has been transferred from UDC2 to the USB host, check the UDFSMSTSTS<mrepempty>.

(b) EOP Disable mode

Master Read transfers when UDC2STSET<eopb_enable > is set to 0 (Master Read EOP disable) are described here. Master Read operations will be as follows:

1. Set the register associated to the UDFSMWSADR and UDFSMWEADR.

2. Set the bits associated to the Master Read operation of UDFSMSTSET and set 1 to the <mr_enable>.

3. UDC2AB starts the data transfer to the EP of UDC2. UDC2 transfers the data to the IN token from the USB host.

4. When the Master Read transfer reaches the Master Read end address, UDC2AB asserts the mr_end_add interrupt. If the FIFO of the EP has reached the MAX packet size during Master Read transfer, UDC2 transfers the data to the IN token from the USB host. However, if it has not reached yet, data will remain in the FIFO to the next transfer.

5. After the handling by the software ended, return to 1.

Note: When UDC2AB is used in the EOP Disable mode, short packets will not be sent out even if the data string to be sent has been transferred. EOP Disable mode should be used only in case the size of the data string is a multiple of the maximum packet size.

The mode can be used if the total size of data string is a multiple of the maximum packet size. For example, the following transfer may be allowed:

Example:

Size of the first Master Read transfer	:100 bytes
Size of the second Master Read transfer	:28 bytes (total of first and second transfer = 128bytes)
Max packet size	:64 bytes

A transfer of 64 bytes will be made twice for the IN transfer.

(2) Aborting of Master Read transfer

You can abort Master Read transfers with the following operation.

1. Use UDC2 Command register to set the status of the relevant EP to Disabled (EP_Disable). (If aborted without making the EP disabled, unintended data may be sent to the USB host.)

2. In order to stop the Master Read transfer, set 1 (Abort) to UDFSMSTSET <mr_abort>.

3. In order to confirm that the transfer is aborted, check that the UDFSMSTSET<mr_enable> was disabled to 0. Subsequent operations should not be made while the mr_enable bit is 1.

   (Information on the address where the transfer ended when aborted can be confirmed with Master Read Current Address and Master Read AHB Address registers.)

4. In order to initialize the Master Read transfer block, set 1 (Reset) to UDFSMSTSET<mr_reset>.

5. Use the Command register (EP_FIFO_Clear) to initialize the FIFO for the relevant EP.

6. Use the Command register (EP_Enable) to enable the relevant EP.

(3) Setting the maximum packet size in Master Read transfers

If the maximum packet size of the EP to be connected with the Master Read function of UDC2AB will be an odd number, there will be following restrictions to which you should pay attention:

- Even if the maximum packet size of the EP should be handled as an odd number, the setting of the UDFS2EPxMSZ<max_pkt> should be an even number.

Note: Refer to the "17.9.2 Appendix B About Setting an Odd Number of Bytes as MaxPacketSize" for details.

- Set the UDC2STSET<eopb_enable> to 1 (Master Read EOP enable).
- Make the transfer size to be specified for one Master Read transfer (Master Read End Address - Master Read Start Address + 1) not exceed the maximum packet size of an odd number.

Example:
Set the maximum packet size of EP (value to pass to the USB host) to be 63bytes.
Make the setting of the UDFS2EPxMSZ<max_pkt> to be 64 bytes.
Keep the transfer size to be specified for one Master Read transfer to 63 bytes or less.

17.5.4.2 Master Write transfer

(1) Master Write Transfer Sequence

Master Write operations will be as follows:

1. Set UDFSMWSADR and UDFSMWEADR.
2. Set the bits associated to the UDFSMSTSET and set 1 to the <mw_enable>.
3. UDC2AB makes a Master Write transfer to the data in the EP received from the USB host.
4. Since the mw_end_add interrupt will be asserted when the writing ended to reach the Master Write End Address (with no timeout processed), you should make necessary arrangement with the software. UDC2 will return to 1 after receiving the correct packet.

Note: UDC2AB will assert the mw_set_add interrupt when the packet is received normally from the USB host with the UDFSMSTSET<mw_enable disabled.

(2) Timeout

Master Write transfers would not finish if the OUT transfer from the USB host should stagnate before reaching the Master Write End Address during the transfer. In order to cope with such circumstances, you can set the timeout function.

When this timeout function is used, all data stored in the buffer in UDC2AB at the point of timeout will be transferred to AHB.

Timeout can be processed with the following operation.

1. Make an access to the UDFSMWTOUT before starting a Master Write transfer and set timeoutset (timeout time) to make <timeout_en> enabled 1.
2. Start the Master Write transfer in accordance with the instruction in the preceding section.

3. When the timeout has occurred, the mw_timeout interrupt will be asserted. (The mw_end_add interrupt will not be asserted.) In that case, the Master Write transfer is not completed to reach the Master Write End Address. UDC2AB clears the UDFSMSTSET<mw_enable> to 0.

4. In UDFSMWCADR, the address to which the transfer has completed to the AHB end can be confirmed.

Please note that the timeout counter advances during the Master Write transfer with the timeout function enabled, but the counter will be reset to the preset value when the OUT transfer from the USB host to the relevant EP is received and begin recounting (see the Figure 17-15). It means that the time until timeout is "from the point when the last transfer from the USB host to the relevant EP has occurred during the Master Write transfer to the preset time," rather than "from the point when the Master Write transfer has begun to the preset time."

If you do not use the timeout function, be sure to set the UDFSMWTOUT<timeout_en> to "Disable 0" before starting the Master Write transfer. In that case, the transfer will not finish until reaching the preset Master Write End Address.

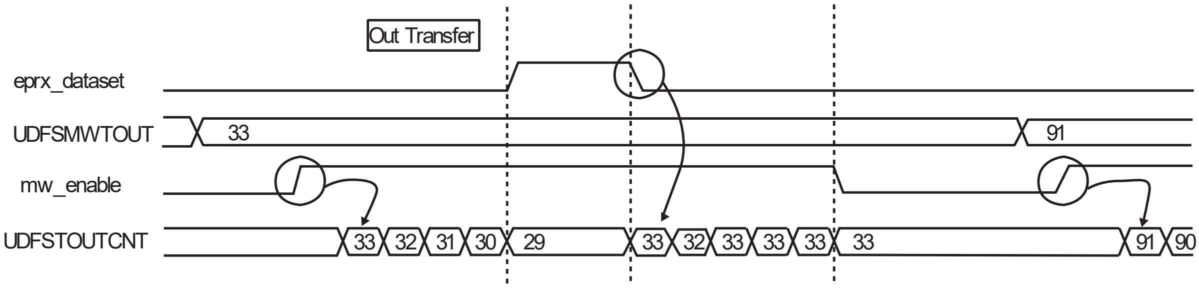


Figure 17-15 Example of MW timeout count

(3) Aborting of Master Write transfers

You can abort Master Write transfers with the following operation.

1. Use UDFS2CMD to set the status of the relevant EP to Disable (EP_Disable).

2. In order to stop the Master Write transfer, set 1 (Abort) to the UDFSMSTSET<mw_abort>.

3. In order to confirm the transfer is aborted, check the UDFSMSTSET<mw_enable> was disabled to 0. Subsequent operations should not be made while the <mw_enable> is 1. (Information on the address where the transfer ended when aborted can be confirmed with Master Write Current Address and Master Write AHB Address registers.)

4. In order to initialize the Master Write transfer block, set 1 (Reset) to the UDFSMSTSET<mw_reset>.

5. Use UDFS2CMD (EP_FIFO_Clear) to initialize the FIFO for the relevant EP.

6. Use UDFS2CMD to set the status of the relevant EP to Enable (EP_Enable).

17.5.5 USB Power Management Control

In USB, operations related to power management including detection of USB bus power supply, suspending and resuming are also prescribed in addition to normal packet transfers. This section discusses about how to control those operations.

Note: Be sure to see the USB 2.0 Specification for details of operations.

17.5.5.1 Connection Diagram of Power Management Control Signal

Below is a connection diagram of signals related to power management control.

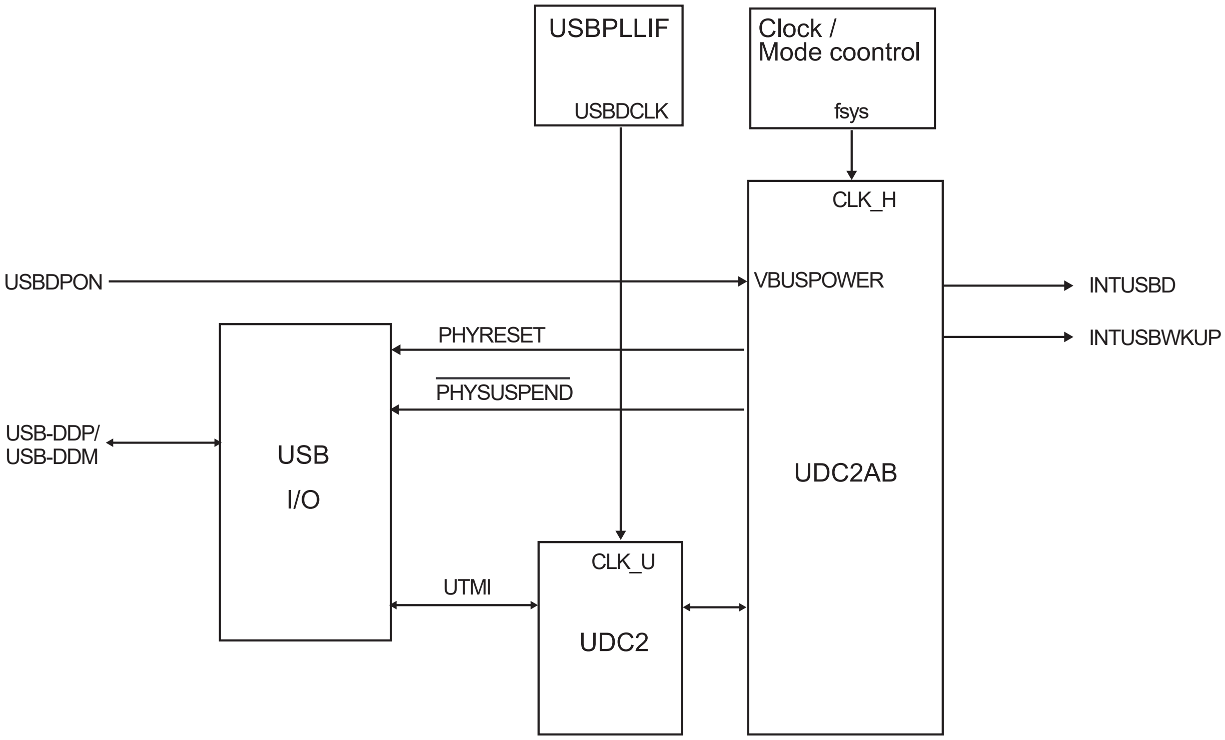

USBPLLIF
USBDCLK
Clock /
Mode coontrol
fsys
CLK_H
USBDPON
VBUSPOWER
INTUSBD
INTUSBWKUP
PHYRESET
PHYSUSPEND
USB-DDP/
USB-DDM
USB
I/O
UDC2AB
CLK_U
UTMI
UDC2

Figure 17-16 Connection Diagram of Power Management Control Signal

17.5.5.2 Sequence of USB Bus Power (VBUS) Connection/Disconnection

(1) Connect

If CLK_H is operating, the USB bus power (VBUS) connection is detected using the INTUSBD (powerdetect) interrupt and UDFSPWCTL<pw_detect>. If UCLK_H is stopped, the USB power connection (VBUS) is detected using the INTUSBDPON interrupt signal.

After detecting bus power (VBUS), initialize UDC2AB and UDC2 following sequence.

1. Use the UDFSPWCTL<pw_resetb> to make software reset. (The <pw_resetb> bit is not automatically released and should be cleared by software.).
2. Make an access to UDC2AB and UDC2 registers to make necessary initial settings.
3. Use UDFS2CMD to issue the USB Ready command. UDC2 notifies the USB host of the connection via PHY. This condition enables UDC2 to accept USB_RESET from the USB host.
4. Once USB_RESET from the USB host is detected, UDC2 initializes the registers inside UDC2 and enumeration with the USB host becomes available. When USB_RESET is detected, the usb_reset / usb_reset_end interrupt occurs.

(2) Disconnect

If CLK_H is operating, the USB bus power (VBUS) disconnection is detected using the INTUSBD(powerdetect) interrupt and UDFSPWCTL<pw_detect>. If CLK_H is stopped, the USB power (VBUS) disconnection is detected using the INTUSBWKUP interrupt.ÅB

When the disconnection of the USB bus power (VBUS) is detected, each master transfer will not automatically stop. Then use the pw_resetb bit of Power Detect Control register to make software reset.

17.5.6 USB Reset

USB_RESET may be received not only when the USB host is connected but also at any timing.

UDC2AB asserts the usb_reset / usb_reset_end interrupt when UDC2 has received USB_RESET and returns to the default state. At this time, master transfers will not automatically stop. Use the abort function to end the transfers. Values are initialized by USB_RESET for some registers of UDC2, while they are retained for other registers (refer to the section of UDC2).

Resetting of UDC2 registers when USB_RESET is recognized should be made after the usb_reset_end interrupt has occurred. This is because UDC2 initializes UDC2 registers at the time it deasserts the usb_reset signal.

17.5.7 Suspend / Resume

17.5.7.1 Shift to the suspended state

UDC2AB makes notification of detecting the suspended state of UDC2 by the INTUSBD (suspend_resume) interrupt and the UDFSPWCTL<suspend_x>.

Since master transfers will not automatically stop in this circumstance, you should use the aborting function of each master transfer to make forcible termination if needed.

In case PHY needs to be suspended (clock stop) after the necessary processes finished by software, you can set the UDFSPWCTL<phy_suspend> to make UDC2AB assert $\overline{\text{PHYSUSPEND}}$ which will put PHY in suspended state.

17.5.7.2 Resuming from suspended state (resuming from the USB host)

The procedures to resume from the suspended state is performed based o the condition of the CLK_H.

When resuming is recognized, make settings again for restarting master transfers.

1. Stopping the CLK_H

The procedures to stop the CLH_H and the signal variation are as shown below.

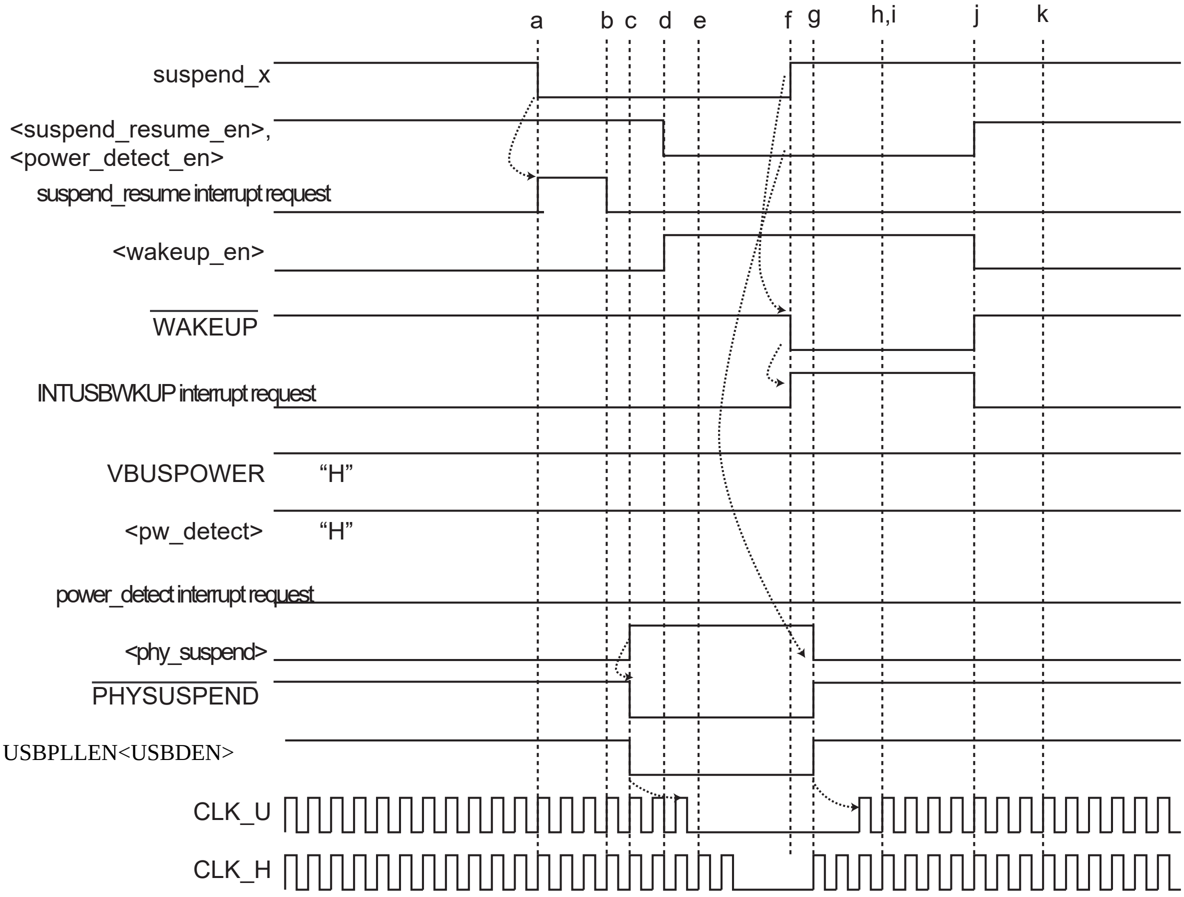

a
b
c
d
e
f
g
h,i
j
k
suspend_x
<suspend_resume_en>,
<power_detect_en>
suspend_resume interrupt request
<wakeup_en>
WAKEUP
INTUSBWKUP interrupt request
VBUSPOWER
"H"
<pw_detect>
"H"
power_detect interrupt request
<phy_suspend>
PHYSUSPEND
USBPLLEN<USBDEN>
CLK_U
CLK_H

Figure 17-17 Signal operations when suspended and resumed (when CLK_H is stopped)

a. The suspend_x of the UDC2 is asserted to zero by detecting the suspend state on the USB bus, and the INTUSBD(suspend_resume) interrupt occurs.

b. The service routine of the INTUSBD(suspend_resume) interrupt clears the interrupt factor.

c. Set the UDFSPWCTL<phy_suspend> to "1". Setting the <phy_suspend> to "1" asserts the $\overline{\text{PHYSUSPEND}}$ output signal to "0".

Zero clear the USBPLLEN<USBDEN> of the USB clock control circuit to stop the CLK_U.

d. Set the UDFSPWCTL<wakeup_en> to "1". Zero clear the UDFSINTENB<power_detect_en><suspend_resume_en> not to generate the INTUSBD(power_detect, suspend_resumu) interrupt.

e. With the INTUSBWKUP interrupt, the operation mode moves into the low-power consumption mode and stops the CLK_H.

f. By detecting Resume on the USB bus, the $\overline{\text{WAKEUP}}$ output signal will be asserted to 0 asynchronously.By $\overline{\text{WAKEUP}}$ output signal, INTUSBWKUP occurs and the low-power consumption mode is cancelled. Then, supply of CLK_H starts.

g. With the supply of CLK_H, $\overline{\text{PHYSUSPEND}}$ output signal is automatically asserted to "1", and <phy_suspend> is zero-cleared.

Set USBPLLEN<USBDEN> of the USB clock control circuit to "1" to activate the CLK_U.

h. 2.5 s after the interrupt is asserted (time required for the signal to stabilize when VBUS is disconnected) and check UDFSPWCTL<pw_detect>. If the UDFSPWCTL<pw_detect> is "1", $\overline{\text{WAKEUP}}$ is asserted by Resume. If UDFSPWCTL<pw_detect> is "0", $\overline{\text{WAKEUP}}$ is asserted by disconnection of the VBUS.

i. To resume, perform the sequences below.To disconnect, perform the sequences of the "17.5.7.3 Resuming from the suspend state (disconnect)".

j. Clears the interrupt factor and <wakeup_en> to deassert the $\overline{\text{WAKEUP}}$ output signal. Set <suspend_resume_en> to "1".

k. Resumes from the suspended state.

2. For CLK_H to work

The procedures to get the CLK_H work and the signal changes are shown as below.

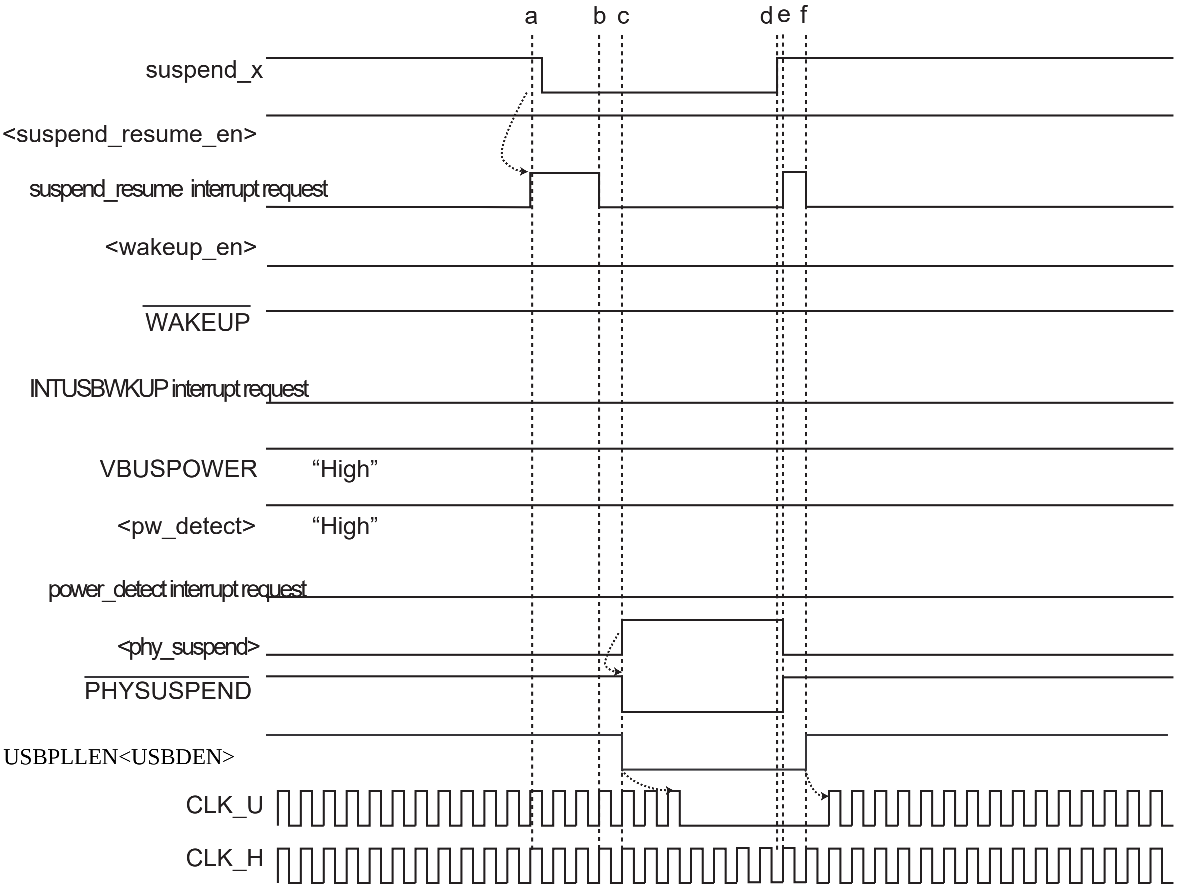

a
b
c
d
e
f
suspend_x
<suspend_resume_en>
suspend_resume interrupt request
<wakeup_en>
WAKEUP
INTUSBWKUP interrupt request
VBUSPOWER "High"
<pw_detect> "High"
power_detect interrupt request
<phy_suspend>
PHYSUSPEND
USBPLLEN<USBDEN>
CLK_U
CLK_H

Figure 17-18 Operation of suspend/resume signals (to get the CLK_H work)

a. INTUSBD(suspend_resume) interrupt occurs by detecting the suspended state on the USB bus.
b. Clears the interrupt source in the INTUSBD(suspend_resume) interrupt service routine.
c. Set "1" to UDFSPWCTL<phy_suspend>. Setting <phy_suspend> to "1" assert the $\overline{\text{PHYSUSPEND}}$ output signal to "0".

   Set USBPLLEN<USBDEN> of the USB clock control circuit to "0" to stop the CLK_U.
d. The suspend_x becomes "1" by detecting resume on the USB bus.

   Aslo, $\overline{\text{PHYSUSPEND}}$ output signal is deasserted to "1" by detecting the rising edge of the suspend_x.
e. INTUSBD(suspend_resume) interrupt occurs.
f. Interrupt source is zero cleared in the service routine of the INTUSBD(suspend_resume).

   Set USBPLLEN<USBDEN> of the USB clock control circuit to "1" to get CLK_U work.
g. Deasserting the $\overline{\text{PHYSUSPEND}}$ output signal will resume the supply of the CLK_U.
h. Resumes from the suspend state.

17.5.7.3 Resuming from the suspend state (disconnect)

The procedures to resume from the suspended state (disconnection) and the signal change are shown as below.

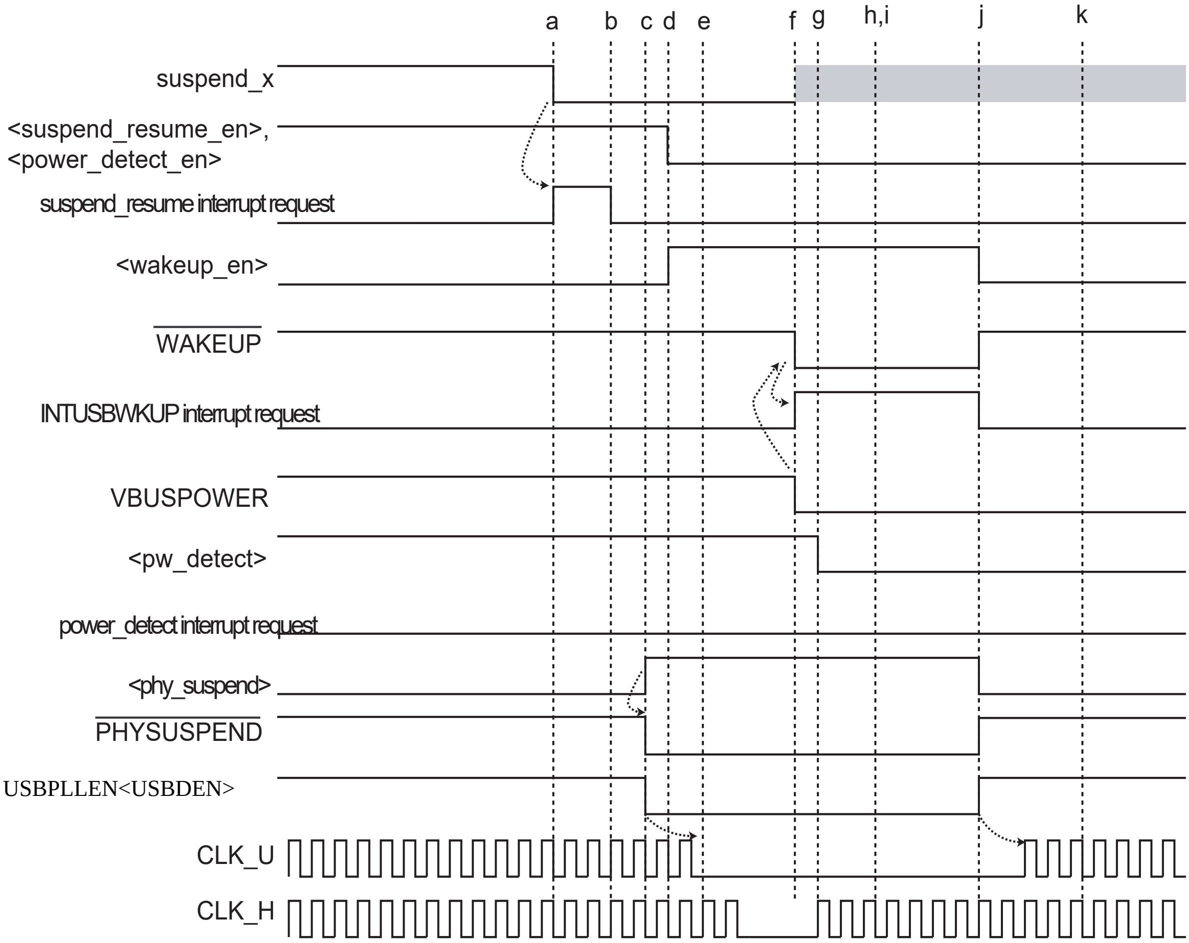

a
b
c
d
e
f
g
h,i
j
k
suspend_x
<suspend_resume_en>,
<power_detect_en>
suspend_resume interrupt request
<wakeup_en>
WAKEUP
INTUSBWKUP interrupt request
VBUSPOWER
<pw_detect>
power_detect interrupt request
<phy_suspend>
PHYSUSPEND
USBPLLEN<USBDEN>
CLK_U
CLK_H

Figure 17-19 Operation of suspend/disconnect signals (to stop CLK_H)

a. 0 is asserted to the suspend_x of the UDC2 by detecting the suspend state on the USB bus and this generates the INTUSBD(suspend_resume) interrupt.
b. Interrupt source is cleared by the service routine of the INTUSBD(suspend_resume) interrupt.
c. Set UDFSPWCTL<phy_suspend> to "1". Setting the <phy_suspend> to "1" asserts the $\overline{\text{PHYSUSPEND}}$ output signal to "0".

   Set USBPLLEN<USBDEN> of the USB clock control circuit to "0" to stop the CLK_U.
d. Set the UDFSPWCTL<wakeup_en> to "1". Zero clear the UDFSINTENB<power_detect_en><suspend_resume_en> not to generate INTUSBD(power_detect, suspend_resumu) interrupt.
e. With the INTUSBWKUP interrupt, the operating mode moves into the low-power consumption mode to stop the CLK_H.
f. If disconnection is detected on the USB bus, the VBUSPOWER pin becomes "0" and the $\overline{\text{WAKEUP}}$ output signal will be asynchronously asserted to "0".
g. $\overline{\text{INTUSBWKUP}}$ interrupt is generated by the $\overline{\text{WAKEUP}}$ output signal and low-power consumption mode is cancelled. The supply of CLK_H starts.

h. 2.5 s after the interrupt is asserted (time required for the signal to stabilize when VBUS is disconnected), check the UDFSPWCTL<pw_detect>. If UDFSPWCTL<pw_detect> is "1", $\overline{\text{WAKEUP}}$ is asserted by resume. If UDFSPWCTL<pw_detect> is "0", $\overline{\text{WAKEUP}}$ is asserted by disconnection of VBUS.

i. If the factor is the resume, perform the sequence written in the "17.5.7.2 Resuming from suspended state (resuming from the USB host)". If the factor is disconnection, perform the sequence below.

j. Zero clear the <phy_suspend> to deassert the $\overline{\text{PHYSYSPEND}}$ output signal.

Set the USBPLLEN<USBDEN> of the USB clock control circuit to "1" to get the CLK_U work.Clear the interrupt factor and <wakeup_en> to deassert the $\overline{\text{WAKEUP}}$ output signal.

k. Set UDFSPWCTL<pw_resetb> using software, initialize the UDC2AB.

17.5.7.4 Remote wakeup from the suspended state

The procedure of remote wakeup from the suspended state and the signal change are shown below.

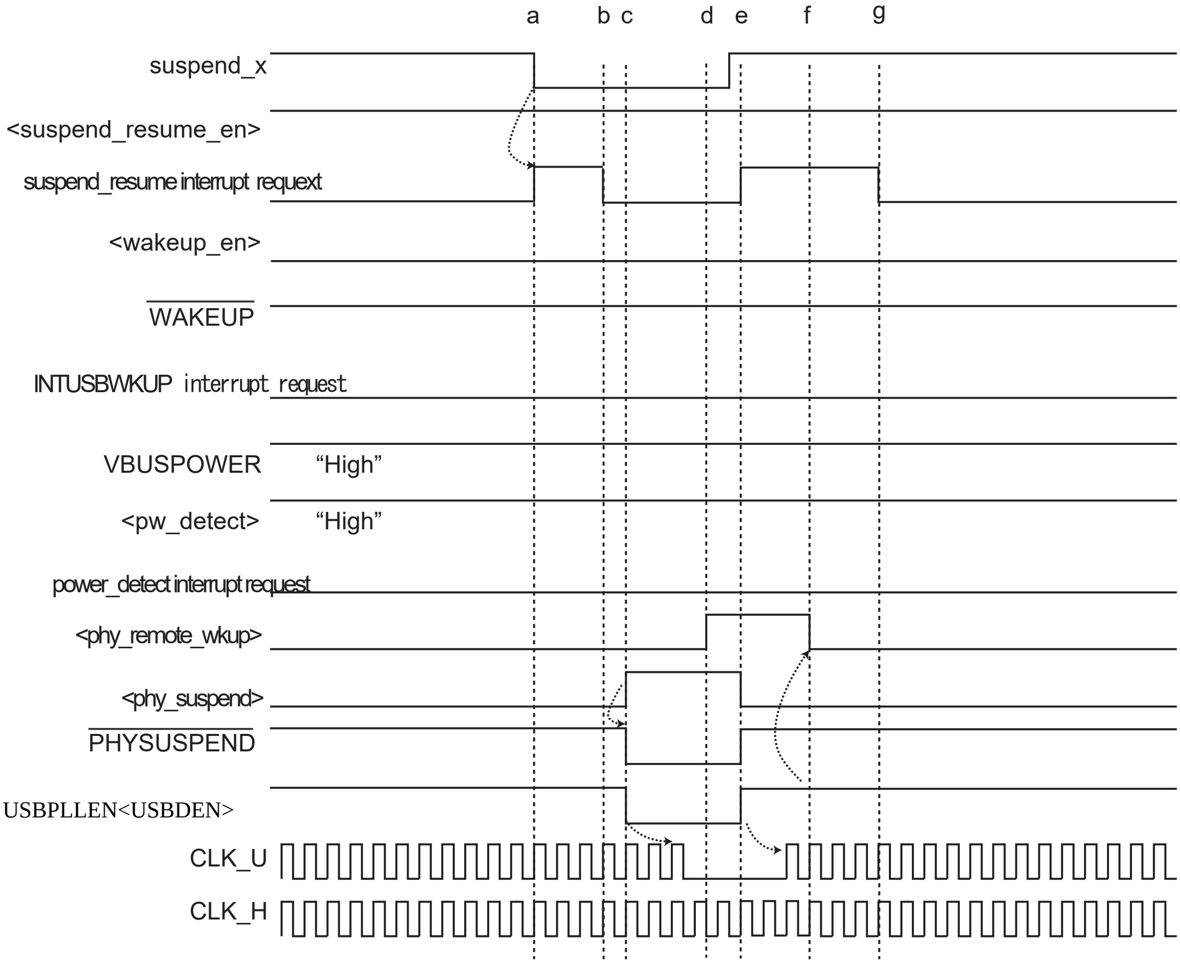

a
b
c
d
e
f
g
suspend_x
<suspend_resume_en>
suspend_resume interrupt requext
<wakeup_en>
$\overline{\text{WAKEUP}}$
INTUSBWKUP interrupt_request
VBUSPOWER
"High"
<pw_detect>
"High"
power_detect interrupt request
<phy_remote_wkup>
<phy_suspend>
$\overline{\text{PHYSUSPEND}}$
USBPLLEN<USBDEN>
CLK_U
CLK_H

Figure 17-20 Operation of suspend/remote wakeup signals

a. suspend_x of the UDC2 is asserted to 0 by detecting the suspended state on the USB bus and the INTUSBD(suspend_resume) interrupt occurs.

b. Clears the interrupt source in the service routine of the INTUSBD(suspend_resume) interrupt.

c. Set the UDFSPWCTL<phy_suspend> to "1". $\overline{\text{PHYSUSPEND}}$ output signal is asserted to "0" by setting <phy_suspend> to "1"

Set the USBPLLEN<USBDEN> of the USB clock control circuit to "0" to stop the CLK_U.

d. When requesting remote wakeup, set the UDFSPWCTL<phy_remote_wkup> to 1. Setting the <phy_remote_wkup> to "1" will cause the UDC2 to make a remote wakeup request on the USB bus. Also, <suspend_x> will be deasserted to 1 asynchronously.

e. Deasserting <suspend_x> will cause the INTUSBD(suspend_resume) interrupt to occur and the $\overline{\text{PHYSUSPEND}}$ output signal to be deasserted to 1.

f. Set the USBPLLEN<USBDEN> of the USB clock control circuit to "1" to get the CLK_U work.

When the CLK_U starts operating, <phy_remote_wkup> is automatically cleared to "0".

g. Clear the interrupt source.

17.6 USB Device Response

UDC2 initializes the inside of UDC2 and sets various registers when hardware reset is detected, USB_RESET is detected, and an enumeration response is made. This section discusses the operations of UDC2 in each status as well as how to control them externally.

1. When hardware reset is detected

   Be sure to reset hardware for UDC2 after the power-on operation. After the hardware reset, UDC2 initializes internal registers and all EPs are in the invalid status, which means the device itself is "Disconnected."

   In order to make the status of UDC2 to "Default," issue the "USB_Ready" command. Issuing this command will put UDC2 in the "Full-Speed" mode, enable the Pull-Up resistance of USB-DDP and notify the host of "Connect".

   In this status, only the USB_RESET signal is accepted from the host.

2. When USB_RESET is detected

   UDC2 initializes internal registers when Bus Reset (USB_RESET) is detected on the USB signal, putting the device in the "Default" status. In this status only EP 0 gets "Ready" enabling enumeration with the host.

3. When "Set_address" request is received

   By setting 010 to the UDFS2ADR<configured> <addressed> <default> and the received address value to the <dev_adr> after receiving the "Set_address" request, UDC2 will be in the "Addressed" status. Setting for this register should be made after the Control transfer has successfully finished (after the STATUS-Stage has ended).

   Transfers to EPs other than EP 0 cannot be made in this status.

4. When "Set_configuration" and "Set_interface" requests are received

   By setting 100 to the UDFS2ADR<configured> <addressed> <default> after receiving the "Set_configuration" and "Set_interface" requests, UDC2 will be in the "Configured" status.

   In the "Configured" status, you can make transfers to the EP to which status settings have been made.

   In order to make the EP "Ready," the following settings should be made:

   - Set the maximum packet size to UDFS2EPxMSZ
   - Set the transfer mode to UDFS2EPxSTS
   - Issue the EP_Reset command to UDFS2CMD

   EPs will be available for transmitting and receiving data after these settings have been made.

   Figure 17-21 shows the "Device State Diagram".

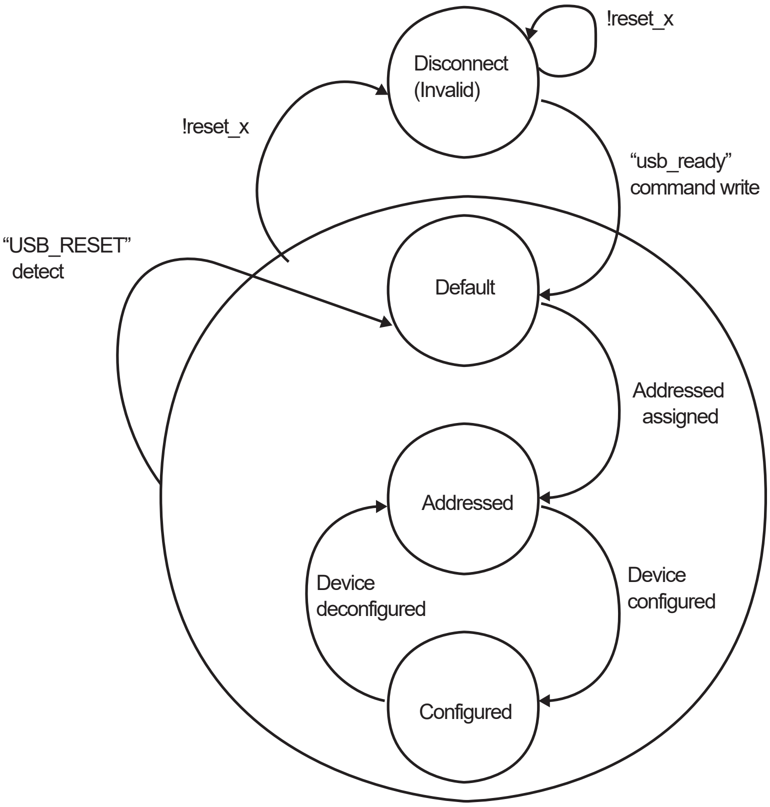

!reset_x
Disconnect (Invalid)
!reset_x
"usb_ready" command write
"USB_RESET" detect
Default
Addressed assigned
Addressed
Device deconfigured
Device configured
Configured

Figure 17-21 Device state diagram

17.7 Flow of Control in Transfer of EPs

17.7.1 EP0

EP0 supports Control transfer and is used as device control for enumeration. EP0 supports only Single packet mode.

Control transfers have SETUP-Stage, DATA-Stage and STATUS-Stage

The types of transfer are categorized into the following major types:

- Control-RD transfer
- Control-WR transfer (without DATA-Stag)
- Control-WR transfer (with DATA-Stage)

UDC2 makes control of those three stages by hardware. Flows in each type of transfer are described below.

17.7.1.1 Control-RD transfer

The flow of control in Control-RD transfers is shown below.

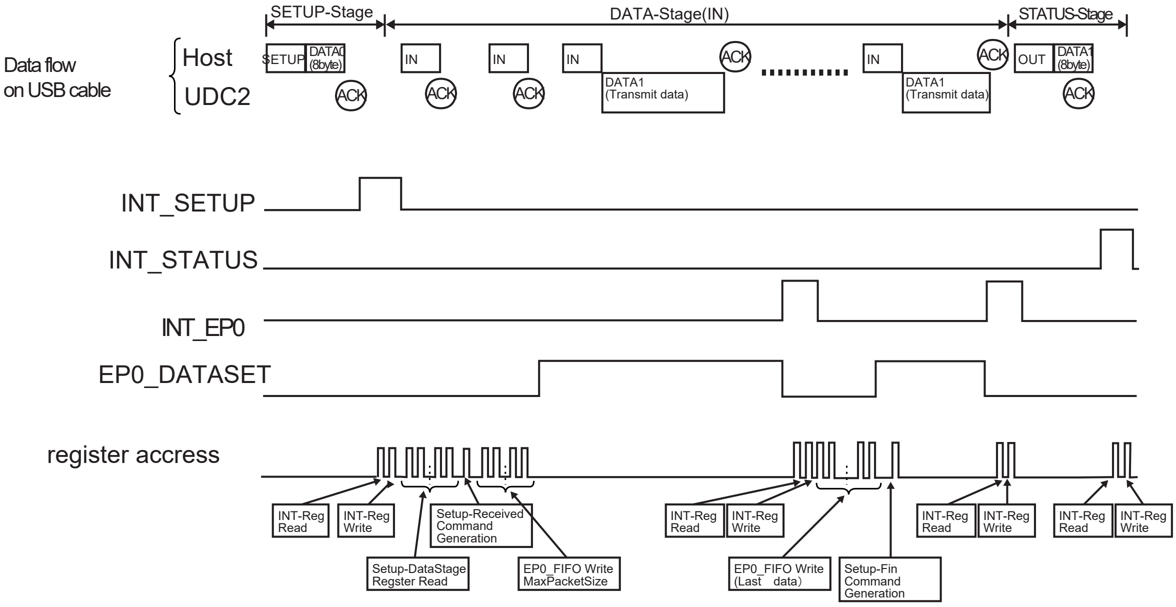

SETUP-Stage
DATA-Stage(IN)
STATUS-Stage
Data flow
on USB cable
Host
UDC2
SETUP
DATA0
(8byte)
ACK
IN
ACK
IN
ACK
IN
DATA1
(Transmit data)
ACK
IN
DATA1
(Transmit data)
ACK
OUT
DATA1
(8byte)
ACK
INT_SETUP
INT_STATUS
INT_EP0
EP0_DATASET
register acrress
INT-Reg
Read
INT-Reg
Write
Setup-DataStage
Regster Read
Setup-Received
Command
Generation
EP0_FIFO Write
MaxPacketSize
INT-Reg
Read
INT-Reg
Write
EP0_FIFO Write
(Last data)
Setup-Fin
Command
Generation
INT-Reg
Read
INT-Reg
Write
INT-Reg
Read
INT-Reg
Write

Figure 17-22 Flow of the control in Control-RD transfer

The following description is based on the assumption that the UDFS2EP0MSZ<dset> is set to "EP0_DATASET flag".

(1) SETUP-Stage

UDC2 asserts the INT_SETUP flag when it has received the Setup-Token. This flag can be cleared by writing 1 into the UDFS2INT<i_setup>. In case flags are combined externally, read the UDFS2INT to confirm which flag is asserted and write "1" into the relevant bit.

Then read Setup-Data storage registers (bRequest-bmRequestType, wValue, wIndex, and wLength registers) to determine the request.

Finally, issue the "Setup_Received" command to inform UDC2 that the SETUP-Stage has finished. Since UDC2 does not allow writing data into the EP0-FIFO before this command is issued, it will keep returning "NAK" to the IN-Token from the host until the command is issued.

(2) DATA-Stage

Write the data to be transmitted to the IN-Token into the EP0-FIFO. If the byte size of the data to send is larger than the MaxPacketSize, divide them into groups of MaxPacketSize before writing. When the number of data reached the MaxPacketSize, the EP0_DATASET flag is asserted.

When the data have been transmitted to the IN-Token from the host with no problem, UDC2 deasserts the EP0_DATASET flag and asserts INT_EP0. Any data remaining to be transmitted should be written into the EP0-FIFO.

If the size of the data to be written is smaller than the MaxPacketSize, issue the "EP_EOP" command to EP0 to inform UDC2 that it is a short packet. With this command, UDC2 recognizes the end of the packet and transmits the short packet data.

Finally, issue the "Setup_Fin" command to inform UDC2 that the DATA-Stage has finished.

(3) STATUS-Stage

When the "Setup_Fin" command is issued, UDC2 will automatically make Handshake for the STATUS-Stage. When the STATUS-Stage finished with no problem, the INT_STATUS flag is asserted. When received a packet of STATUS-Stage from the host before the "Setup_Fin" command is issued, UDC2 will return "NAK" and asserts the INT_STATUS_NAK flag. Therefore, if this flag is asserted, be sure to issue the "Setup_Fin" command.

17.7.1.2 Control-WR transfer (without DATA-Stage)

The flow of control in Control-WR transfer (without DATA-Stage) is shown below.

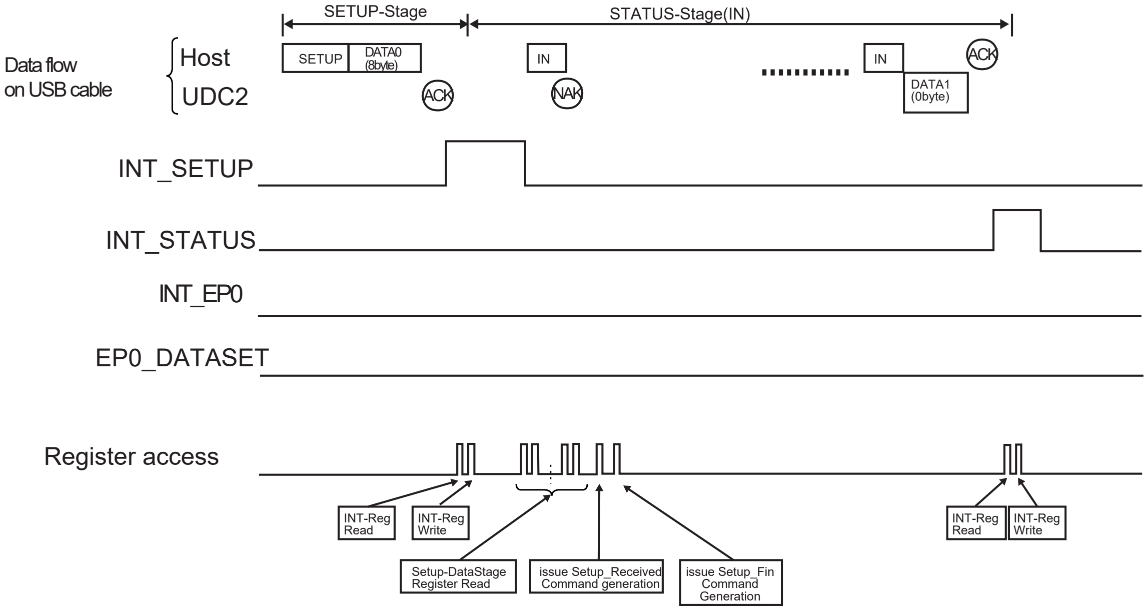

SETUP-Stage
STATUS-Stage(IN)
Data flow
on USB cable
Host
UDC2
SETUP
DATA0
(8byte)
ACK
IN
NAK
IN
ACK
DATA1
(0byte)
INT_SETUP
INT_STATUS
INT_EP0
EP0_DATASET
Register access
INT-Reg
Read
INT-Reg
Write
Setup-DataStage
Register Read
issue Setup_Received
Command generation
issue Setup_Fin
Command
Generation
INT-Reg
Read
INT-Reg
Write

Figure 17-23 Flow of control in Control-WR transfer (without DATA-Stage)

(1) SETUP-Stage

Perform the same procedure described in "17.7.1.1 Control-RD transfer".

(2) STATUS-Stage

After issuing the "Setup_Received" command, make register accesses to UDC2 based on each request. Issue the "Setup_Fin" command when all the register accesses to UDC2 have finished. Subsequent processes are basically the same as the STATUS-Stage described in "17.7.1.1 Control-RD transfer". UDC2 will keep on returning "NAK" until the "Setup_Fin" command is issued.

Note: While register accesses required for each request are made to UDC2 between 'Issuing the "Setup_Received" command' and 'Issuing the "Setup_Fin" command', register accesses are needed after the end of STATUS-Stage in some cases such as Set Address request and Set Feature (TEST_MODE). Processes required for the standard requests are described in "17.7.1.5 Processing when standard request".

17.7.1.3 Control-WR transfer (with DATA-Stage)

The flow of control in Control-WR transfer (with DATA-Stage) is shown below.

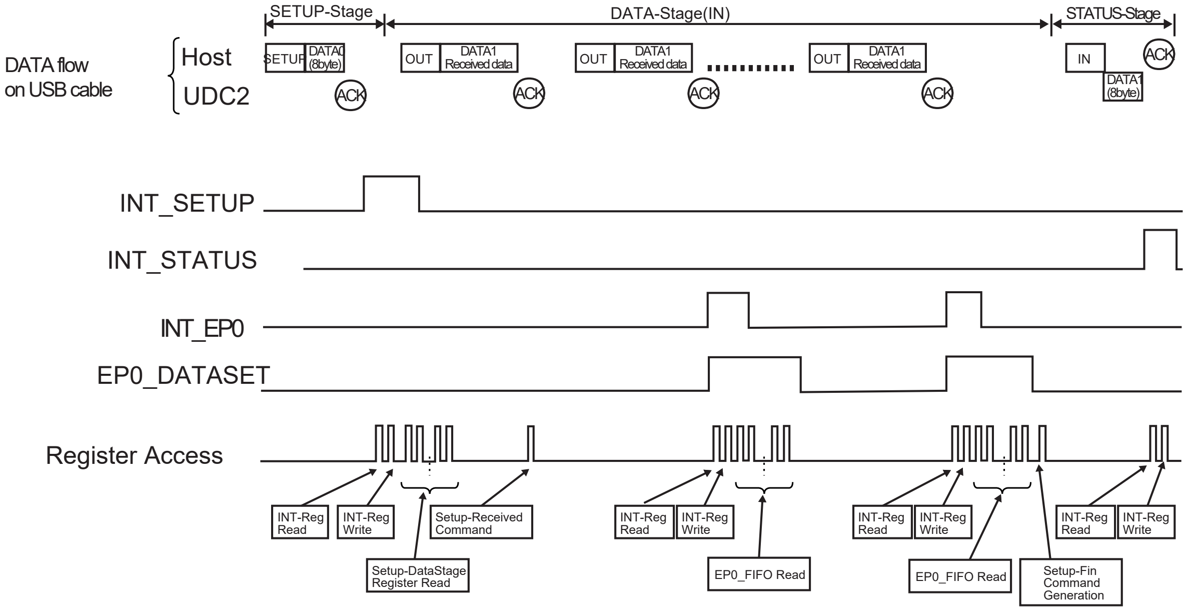

SETUP-Stage
DATA-Stage(IN)
STATUS-Stage
DATA flow on USB cable
Host
UDC2
SETUP
DATA0 (8byte)
ACK
OUT
DATA1 Received data
IN
DATA1 (8byte)
INT_SETUP
INT_STATUS
INT_EP0
EP0_DATASET
Register Access
INT-Reg Read
INT-Reg Write
Setup-Received Command
Setup-DataStage Register Read
EP0_FIFO Read
Setup-Fin Command Generation

Figure 17-24 Flow of control in Control-WR transfers (with DATA-Stage)

(1) SETUP-Stage

To be processed in the same way of SETUP-Stage as described in"17.7.1.1 Control-RD transfer".

(2) DATA-Stage

When the data is received from the host with no problem, UDC2 asserts the EP0_DATASET flag and asserts the INT_EP0 flag. When this flag is asserted, read the data from EP0_FIFO after confirming the received data size in the UDFS2EP0FIFO, or read the data from EP0_FIFO polling the EP0_DATASET flag.

When the byte size of received data has been read, UDC2 deasserts the EP0_DATASET flag.

(3) STATUS-Stage

To be processed in the same way as in the STATUS-Stage described in "17.7.1.1 Control-RD transfer".

17.7.1.4 Example of using the INT_STATUS_NAK flag

When processing requests without DATA-Stage, the INT_STATUS_NAK flag may get asserted by receiving STATUS-Stage from the host before clearing the INT_SETUP flag after it has been asserted, especially in High-Speed transfers. In case such multiple interrupts should be avoided as much as possible, you can use a method to mask the INT_STATUS_NAK flag for request having no DATA-Stage. In such case, basically set 1 to UDFS2INT<m_status_nak>, while 0 should be set only when requests having DATA-Stage are received. (An example for Control-RD transfers is provided below.)

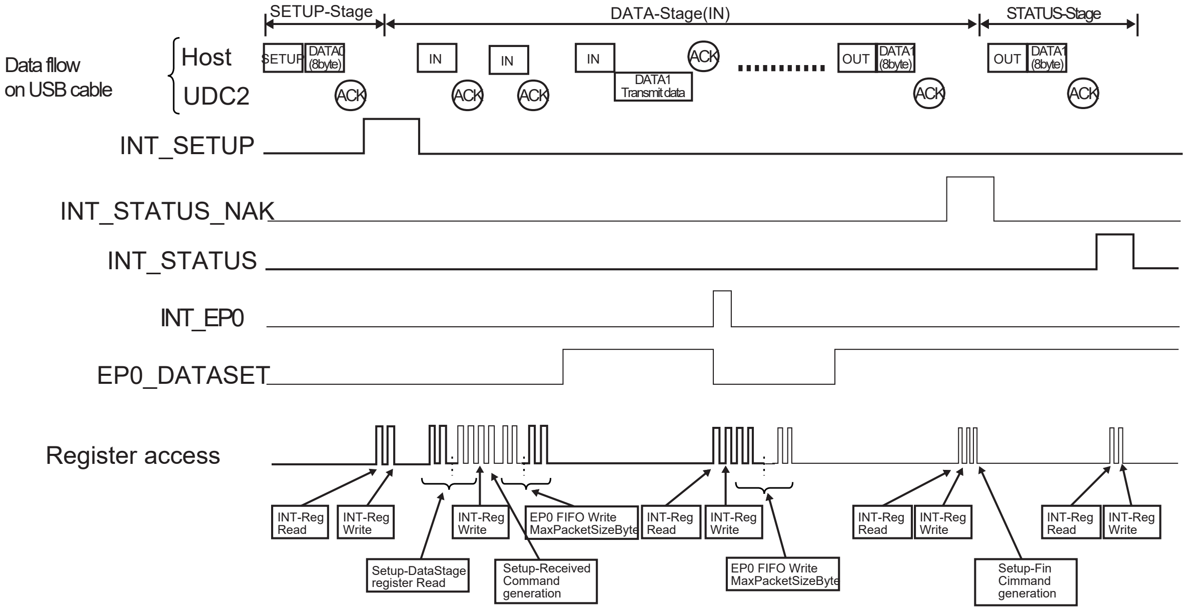

SETUP-Stage
DATA-Stage(IN)
STATUS-Stage
Data flow on USB cable
Host
UDC2
SETUP
DATA0 (8byte)
ACK
IN
ACK
IN
ACK
IN
DATA1 Transmit data
ACK
OUT
DATA1 (8byte)
ACK
OUT
DATA1 (8byte)
ACK
INT_SETUP
INT_STATUS_NAK
INT_STATUS
INT_EP0
EP0_DATASET
Register access
INT-Reg Read
INT-Reg Write
Setup-DataStage register Read
INT-Reg Write
Setup-Received Command generation
EP0 FIFO Write MaxPacketSizeByte
INT-Reg Read
INT-Reg Write
EP0 FIFO Write MaxPacketSizeByte
INT-Reg Read
INT-Reg Write
Setup-Fin Cimmand generation
INT-Reg Read
INT-Reg Write

Figure 17-25 Example of using the INT_STATUS_NAK flag in Control-RD transfers

(1) SETUP-Stage

After the INT_SETUP flag was asserted, clear the UDFS2INT<i_setup> is set to 1, it should be also cleared.

Then, if the request was judged to have DATA-Stage by reading Setup-Data storage registers, set the UDFS2INT<m_status_nak> to 0. Then issue the "Setup_Received" command.

(2) DATA-Stage→STATUS-Stage

When the INT_STATUS_NAK flag was asserted, the device should also proceed to the STATUS-Stage. Clear the UDFS2INT<i_status_nak> and then issue the "Setup_Fin" command. Also, set 1 to the UDFS2INT<m_status_nak> in order to get ready for subsequent transfers.

17.7.1.5 Processing when standard request

Examples of making register accesses to UDC when standard requests are received are provided below. Descriptions of each request are basically provided for each state of the device (Default, Address, and Configured).

For the information on register accesses common to each request, see 17.7.1.1 , 17.7.1.2 and 17.7.1.3.

You should note, however, descriptions provided below do not include the entire details of standard requests in USB 2.0. Since methods to access registers may vary depending on each user's usage, be sure to refer to the USB 2.0 specifications. You should also refer to the USB 2.0 specifications for "Recipient," "Descriptor Types," "Standard Feature Selectors," "Test Mode Selectors" and other terms appear in the descriptions below.

- Standard requests for "17.7.1.1 Control-RD transfer".

  Get Status Get Description Get Configuration

  Get Interface Get Frame
- Standard requests for "17.7.1.2 Control-WR transfer (without DATA-Stage)".

  Clear Feature Set Feature Set Address

  Set Configuration Set Interface
- Standard requests for "17.7.1.3 Control-WR transfer (with DATA-Stage)"

  Set Description

Note 1: Descriptions with double underlines refer to register accessed to UDC2.

Note 2: Writing accesses to UDFS2CMD are described in the following manner for simplicity:

(Example 1) When writing 0x0 to UDFS2CMD<ep> and 0x4 to <com>

→Issue the EP-Stall command to EP0

(Example 2) When writing the relevant EP to UDFS2CMD<ep> and 0x5 to <com>

→Issue the EP-Invalid command to the relevant EP

(1) Get Status Request

To meet this request, the status of the specified receiving end (recipient) is returned.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
1000_0000 1000_0001 1000_0010	GET_STATUS	Zero	Zero Interface EP	Two	Device Interface, or EP Status

- Common to all states:

  If the EP/Interface specified by wIndex does not exist, issue the EP-Stall command to EP0.
- Default state:

  Nothing is specified for the operation of devices by the USB 2.0 specifications.
- Address state:

<recipient> = Device : Write the information on the device (Table 17-3) to UDFS2EP0FIFO.
<recipient> = Interface: Issue the EP-Stall command to EP0
<recipient> = EP : If wIndex=0(EP0), write the information on EP0 (Table 17-5) to UDFS2EP0FIFO. If wIndex≠0(EPx), issue the EP-Stall command to EP0.

- Configured state:

<recipient> = Device : Write the information on the device (Table 17-3) to UDFS2EP0FIFO.

<recipient> = Interface: If the interface specified by lwIndex, write the information on the interface (Table 17-4) to UDFS2EP0FIFO.

<recipient> = EP : If the EP specified by wIndex, write the information on the relevant EP(Table 17-5) to UDFS2EP0FIFO.

Table 17-3 Information on the device to be returned by Get Status request

D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	0	0	0	0
D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	Remote Wakeup	Self Powered

RemoteWakeup (D1)	0 indicates the bus power while 1 indicates the selfpower.
SelfPowered (D0)	0 indicates the remote wakeup function is disabled while 1 indicates it is enabled.

Table 17-4 Information on the interface to be returned by Get Status

D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	0	0	0	0
D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0

Please note that all bits are 0.

Table 17-5 Information on the EP to be returned by Get Status request

D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	0	0	0	0
D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	Halt

Halt (D1)	If this bit is 1, it indicates that the relevant EP is in the "Halt" state.

(2) Clear Feature Request

To meet this request, the particular functions are cleared and disabled.

bmRequesType	bRequest	wValue	wIndex	wLength	Data
1000_0000 1000_0001 1000_0010	CLEAR_FEATURE	Feature Selector	Zero Interface EP	Zero	None

- Common to all states:

  If Feature Selector (wValue) which cannot be cleared (disabled) or does not exist is specified, issue the EP-Stall command to EP0.

  If the EP/Interface specified by wIndex does not exist, issue the EP-Stall command to EP0.

- Default state:

  Nothing is specified for the operation of devices by the USB 2.0 specifications.

- Address state:

<recipient> = Device :	If wValue=1, disable the DEVICE_REMOTE_WAKEUP function at the user program. No register access to UDC2 is required.
<recipient> = Interface:	Issue the EP-Stall command to EP0.
<recipient> = EP :	If wIndex≠0(EPx), issue the EP-Stall command to EP0. If wValue=0 andwIndex=0(EP0), clear the Halt state of EP0 but no register access to UDC2 is required.

- Configured state:

<recipient> = Device :	If wValue=1, disable the DEVICE_REMOTE_WAKEUP function at the user program. No register access to UDC2 is required.
<recipient> = Interface:	Issue the EP-Stall command to EP0.(note)
<recipient> = EP :	If wValue=0 and wIndex=0(EPx), issue the EP-Reset command to the relevant command. If wValue=0 and wIndex=0(EP0), clear the Halt state of EP0 but no register access to UDC2 is required.

Note: EP 0 is to be stalled based on the interpretation of the USB 2.0 specifications that "No Feature Selector exists for Interface" here. For more information, see the USB Specification.

(3) Set Feature Request

To meet this request, the specific functions are set or enabled.

BmRequesetType	BRequesdt	wValue	wIndex		wLength	Data
1000_0000 1000_0001 1000_0010	SET_FEATURE	Feature Selector	Zero Interface EP	Test Selector	Zero	None

- Common to all state:

  If Feature Selector (wValue) which cannot be set (enabled) or does not exist is specified, Eissue the EP-Stall command to the EP0.

  If the EP/Interface specified by the lower byte of wIndex does not exist, issue the EP-Stall command to EP0.

Note: When using a vendor-specific nonstandard Test Selector, the appropriate operation should be made.

- Default state:

  Nothing is specified for the operation of devices by the USB 2.0 specifications except for the above-mentioned TEST_MODE.

- Address state:

<recipient> = Device : If wValue=1, disable the DEVICE_REMOTE_WAKEUP function at the userÅfs end. No register access to UDC2 is required.
<recipient> = Interface: Issue the EP-Stall command to EP0.
<recipient> = EP : If the lower byte of wIndex ≠0 (EPx), issue the EP-Stall to EP0.
If wValue=0 and the lower byte of wIndex=0 (EP0), make EP0 to Halt state. (note 2)

- Configured state:

<recipient> = Device : If wValue=1, enalbe the DEVICE_REMOTE_WAKEUP function at the userÅfs end. No register access to UDC2 is required.
<recipient> = Interface: Issue the EP-Stall command to EP0.(note 1)
<recipient> = EP : If wValue=0 and the lower byte of wIndex≠0(EPx), issue the EP-Stall command to EP0.
If wValue=0 and the lower byte of wIndex=0(EP0), make EP0 to Halt state.(note 2)

Note 1: EP 0 is to be stalled based on the interpretation of the USB specifications that "No Feature Selector exists for Interface" here. For more information, see the USB specifications.

Note 2: USB 2.0 specifications include such description that "Performing the Halt function for EP 0 is neither necessary nor recommended." Accordingly, it can be interpreted that it is not necessary to set UDC2 to the Stall state in this case.

In order to actually make EP 0 be in the Halt state, users have to manage the "Halt state.

Then, when a request is received in the "Halt state", such processes as to issue the EP-Stall command to EP0 in DATA-Stage/STATUS-Stage will be required. (Even if EP0 is set to the Stall state, UDC2 will cancel the Stall state when the Setup-Token is received and will return "ACK.")

As such, the process when SetFeature/ClearFeature is received for EP 0 varies depending on user's usage.

(4) Set Address Request

To meet this request, device addresses are set.

BmRequesetType	BRequesdt	wValue	wIndex	wLength	Data
0000_0000	SET_ADDRESS	Device Address	Zero	Zero	None

For this request, make register accesses shown below within 2 ms after the STATUS-Stage has ended.

(The device address should not be changed before the Setup_Fin command is issued.)

- Default state:

wValue=0: Keep the default state. No register access to UDC2 is required.

wValue≠0: Set wValue to UDFS2ADR<dev_adr> and set 010 to <configured>, <addressed> and <default>. UDC2 will be put in the address state.

- Address state:

wValue=0: Set 0x00 to UDFS2ADR<dev_adr> and 010 to <configured>, <addressed> and <default>. UDC2 will be put in the default state.

wValue≠0: Set wValue to UDFS2ADR<dev_adr>. UDC2 will be set to a new device address.

- Configured state:

Nothing is specified for the operation of devices by the USB 2.0 specification.

(5) Get Descriptor Request

To this request, the specified descriptor is returned.

BmRequesetType	BRequesdt	wValue	wIndex	wLength	Data
1000_0000	GET_DESCRIPTOR	Descriptor Type and Descriptor Index	Zero or Language ID	Descriptor Length	Descriptor

Common to all states:

Write the descriptor information specified by wValue to UDFS2EP0FIFO for the byte size specified by wLength. If the byte size to write is larger than the MaxPacketSize of EP 0, you need to divide the data to write it several times (refer to "17.7.1.1 Control-RD transfer"for details). (If the length of the descriptor is longer than wLength, write the information for wLength bytes from the beginning of the descriptor. If the length of the descriptor is shorter than wLength, write the full information for the descriptor.)

If the descriptor specified by wValue is not supported by the user, issue the EP-Stall command to EP0.

(6) Set Descriptor Request

BmRequesetType	BRequesdt	wValue	wIndex	wLength	Data
0000_0000	SET_DESCRIPTOR	Device Type and Descriptor Index	Language ID or Zero	Descriptor Length	Descriptor

- Common to all states:

  When this request is not supported, issue the EP-Stall command to EP0.
- Default state:

  Nothing is specified for the operation of devices by the USB 2.0 specifications.
- Address state & Configured state:

  Read the information on the description received by UDC2 from UDFS2EP0FIFO.

(7) Get Configuration Request

BmRequesetType	BRequesdt	wValue	wIndex	wLength	Data
1000 0000	GET_CONFIGURATION	Zero	Zero	One	Configuration Value

- Default state:

  To this request, the Configuration value of the current device is returned.
- Address state:

  Write 0x00 to UDFS2EP0FIFO. As this is not configured, 0 should be returned.
- Configured state:

  Write the current configuration value to the UDFS2EP0FIFO.

  Since this has been configured, values other than 0 should be returned.

(8) Set Configuration Request

To meet this request, Device Configuration is set.

BmRequesetType	BRequesdt	wValue	wIndex	wLength	Data
0000 0000	SET_CONFIGURATION	Configuration Value	Zero	Zero	None

- Default state:

  Nothing is specified for the operation of devices by the USB 2.0 specifications.

- Address state:

  When wValue = 0:

  - Keeps the address state. No register access to UDC2 is required.

  When wValue≠0 and the wValue is a Configuration value matching the descriptor :

  - Set 100 to UDFS2ADR<configured> <addressed> <default>.

  <For EPs to use>

  - Set MaxPacketSize to UDFS2EPxMSZ<max_pkt>.
  - Set respective values to UDFS2EPxSTS<pkt_mode>, <bus_sel>, <dir>, <t_type> and <num_mf>.
  - Issue the EP-Reset coomand to the relevant EPs.

  When wValue≠0 and the wValue is a Configuration value not matching the descriptor:

  - Issue the EP-Stall command to EP0.

- Configured state:

  When wValue = 0:

  - Set 010 to UDFS2ADR<configured> <addressed> <default>.
  - Issue the All-EP-Invalid command.

  When Value≠0 and it is a Configuration value matching the descriptor:

  <For EPs to use>

  - Set the MaxPacketSize to UDFS2EPxMSZ<max_pkt>.
  - Set respective values to UDFS2EPxSTS<pkt_mode>, <bus_sel>, <dir>, <t_type> and <num_mf>.
  - Issue the EP-Reset command to the relevant EPs.

  <For EPs to become unused>

  - Issue the EP-Invalid command to the relevant EPs.

  When wValue≠0 and the wValue is a Configuration value not matching the descriptor :

  - Issue the EP-Stall command to EP0.

(9) Get Interface Request

To meet this request, the AlternateSetting value set by the specified interface is returned.

BmRequesetType	BRequesdt	wValue	wIndex	wLength	Data
1000_0001	GET_INTERFACE	Zero	Interface	One	Alternate Setting

- Common to all states:

  If the interface specified by wIndex, issue the EP-Stall command to EP0.
- Default state:

  Nothing is specified for the operation of devices by the USB 2.0 specifications.
- Address state:

  Issue the EP-Stall to EP0.
- Configured state:

  Write the current alternate setting value of the interface specified by the wIndex to UDFS2EP0FIFO.

(10) Set Interface Request

To meet this request, the Alternate Setting value of the specified interface is set.

BmRequesetType	BRequesdt	wValue	wIndex	wLength	Data
0000_0001	SET_INTERFACE	Alternate Setting	Interface	Zero	None

- Common to all states:

  If the interface specified by wIndex does not exist or if the Alternate Setting specified by wValue does not exist, issue the EP-Stall command to EP0.
- Default state:

  Nothing is specified for the operation of devices by the USB 2.0 specifications.
- Address state:

  Issue the EP-Stall command to EP0.
- Configured state:

  <For the EPs to use in Alternate Setting of the specified interface>
  - Set MaxPacketSize to UDFS2EPxMSZ<max_pkt>.
  - Set respective values to UDFS2EPxSTS<pkt_mode>, <bus_sel>, <dir>, <t_type> and <num_mf>.
  - EP-ResetIssue the EP-Reset command to the relevant EPs.

  <For EPs to become unused>
  - Issue the EP-Invalid command to the relevant EPs.

(11) Synch Frame Request

To meet this request, the Synch Frame of the EP is returned.

BmRequesetType	BRequesdt	wValue	wIndex	wLength	Data
1000 0010	SYNCH_FRAME	Zero	EP	Two	Frame Number

- Common to all states:

  If this request is not supported by the EP specified by wIndex, issue the EP-Stall command to EP0.
- Default state:

  Nothing is specified for the operation of devices by the USB 2.0 specifications.
- Address state:

  Issue the EP-Stall command to EP0.
- Configured state:

  Write the Frame Number of the EP specified by wIndex to UDFS2EP0FIFO.

17.7.2 EPs other than EP0

EPs other than EP 0 support Bulk (send/receive), Interrupt (send/receive), and Isochronous (send/receive) transfers and are used to transmit and receive data. They also support the Dual Packet mode which enables high-speed data communication.

17.8 Suspend/Resume State

UDC2 enters into a suspended state based on the signal condition from the host. It also returns from the suspended state by resuming operation by the host or UDC2.

Shifting between the states is described below.

17.8.1 Shift to the suspended state

Though the host issues SOF with given intervals (FS: 1 ms) in the normal state, it will stop issuing this SOF to the device when it tries to make the device suspended and the data on the USB signal line will be unchanged keeping the idle state. UDC2 is always monitoring the "line_state" from PHY and makes judgment of whether it is in the suspended state or USB_RESET when the idle state is detected for 3 ms or longer. If judged to be in the suspended state, it will assert "suspend_x" to "Low" and enter in the suspended state.

Please note accesses to registers will be unavailable while UDC2 is suspended, since supply of CLK from USB clock control circuit.

17.8.2 Resuming from suspended state

Resuming from the suspended state can be made in two ways; by outputting a resuming state from the host and by way of remote wakeup from UDC2 (outputting a resuming state).

Resuming process in each case is described below.

17.8.2.1 Resuming by an output from the host

When a resuming state is output by the host, UDC2 deasserts suspend_x to "High" to declare resuming from the suspend state.

17.8.2.2 Resuming by way remote wakeup from UDC2

The remote wakeup function may not be supported by some applications, and it needs to be permitted by the USB host at the time of bus enumeration. You should not assert "wakeup" unless permitted by the system.

If permitted by the system, asserting the "wakeup" pin will make UDC2 output a resuming state to the host to start resuming. Please note that the clock supply from USB clock control circuit is stopped when UDC2 is suspended, so you should keep asserting wakeup until it resumes. The remote wakeup should be operated after 2 ms or more has passed after suspend_x was asserted to "Low".

17.9 USB-Spec2.0 Device Controller Appendix

17.9.1 Appendix A System Power Management

In USB, operations related to the enumeration and power control signals (USB-DDP/USB-DDM) for reset and suspend from the host are also prescribed, in addition to normal transfer operations. This Appendix provides information about the specifications of USB 2.0 PHY to be connected and clock control on the system level required for processes related to the USB-DDP/USB-DDM signals. For details of each process, please be sure to check the USB Specification Revision 2.0, USB-I/O specification.

The words in Appendix A are described below.

1. Reset:

   The operation of the USB-DDP/USB-DDM signals for initializing the USB device (hereafter called "the device") from the USB host (hereafter called "the host"). After reset, enumeration is performed and then normal transfer operations such as Bulk transfers begin. Upon being connected, the device is always reset. The device also needs to support reset operation at any other arbitrary timing.

2. Suspend

   If no bus activity on the USB-DDP/USB-DDM lines including SOF is initiated by the host for 3 ms or longer, the device needs to be put in the suspend mode to reduce power consumption. In this case, the device is required to perform certain operations such as stopping the clock.

3. Resume

   The operation of the USB-DDP/USB-DDM signals for resuming normal operation from the suspend mode. Resume operation can be initiated either by the host or the device. Resume operation from the device is called "remote wakeup".

The each operation is described below. The time in ( ) is value in the USB 2.0 Specification.

17.9.1.1 Connect / Disconnect Operations

(1) Connect Operation

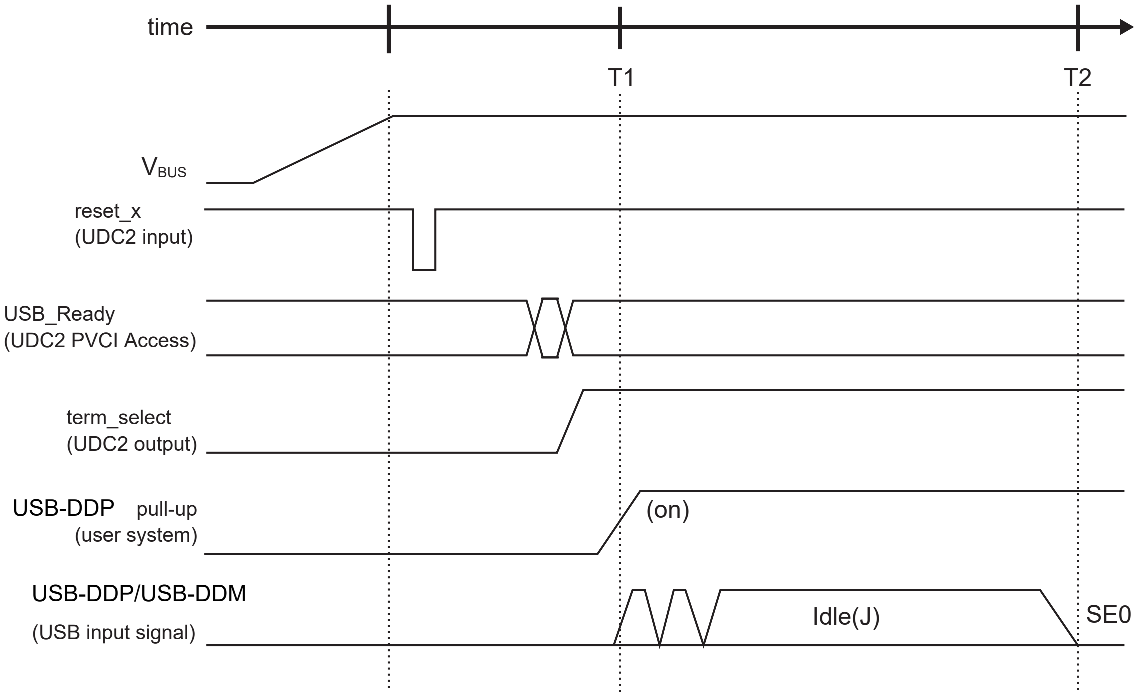

time
T1
T2
V_{BUS}
reset_x
(UDC2 input)
USB_Ready
(UDC2 PVCI Access)
term_select
(UDC2 output)
USB-DDP pull-up
(user system)
(on)
USB-DDP/USB-DDM
(USB input signal)
Idle(J)
SE0

Figure 17-26 Connect operation timing

- T0: VBUS detection

  When Vbus is detected, a system reset (reset_x input) should be applied to UDC2.

  xcvr_select is "High" and term_select is "Low".
- T1: Device connect (no later than 100ms after T0)

  The device must enable USB-DDP no later than 100 ms after Vbus detection (T0) to notify the host of the connected state. Therefore, when Vbus is detected and the device is ready to communicate with the host, the system should access the UDFS2CMD in UDC2 to set the USB_Ready command. After that, the user system sets the port using software to enable the USB-DDP pull-up.
- T2:USB Reset Start (more than 100ms after 100ms)

(2) Disconnect Operation

When a disconnected state is detected, it is recommended to apply a system reset to UDC2.

17.9.1.2 Reset Operation

The "reset" here refers to the "Reset Signaling" defined in the USB 2.0 Specification, not the system reset (reset_x) to UDC2.

(1) When Operation in FS Mode after Reset

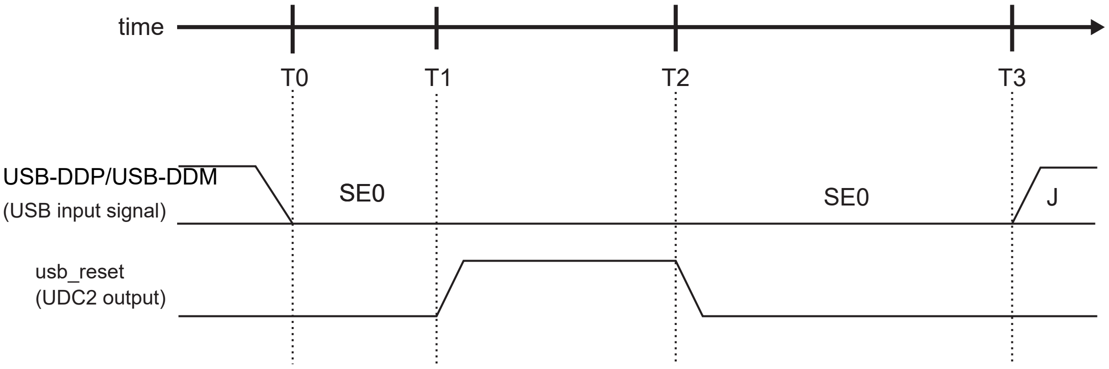


Figure 17-27 Reset Operation Timing

- T0: Reset start

  Upon recognizing SE0 from the host, UDC2 starts counting to recognize the reset.
- T1: Reset recognition (more than 2.5 μs after T0)

  When UDC2 detects SE0 for more than approximately 68 μs after T0, it recognizes the reset from the host and drives usb_reset "High".
- T2: deassert of USB reset

  At this point, usb_reset is driven "Low" more than 3.5ms from T1.
- T3: Reset end (more than 10ms after T0)

  When SE0 from the host finishes and the device enters an idle state, it indicates the end of reset operation. The reset period from the host lasts a minimum of 10 ms.

(2) Notes on Reset Operation

- Initialization of registers after reset

  When the reset from the host is completed (when usb_reset changes from "High" to "Low"), all the internal registers of UDC2 are initialized (For the initial value of each register, refer to "17.4 Registers").

  Note that registers that are set while usb_reset is "High" are also initialized. Therefore, the UDC2 registers should be set after the reset period is completed.
- DMA transfer (EP-I/F access) after reset

  When a reset from the host occurs during DMA transfer, the UDFS2EPxSTS is initialized and the bus access mode is set to "common bus access". Therefore, DMA transfer cannot be continued properly. When a reset occurs, the DMA controller must also be initialized.

  In the enumeration operation after reset, configure each EP and then initialize the EPs by setting the EP_Reset command in the UDFS2CMD.

17.9.1.3 Suspend Operation

(1) Suspend operation

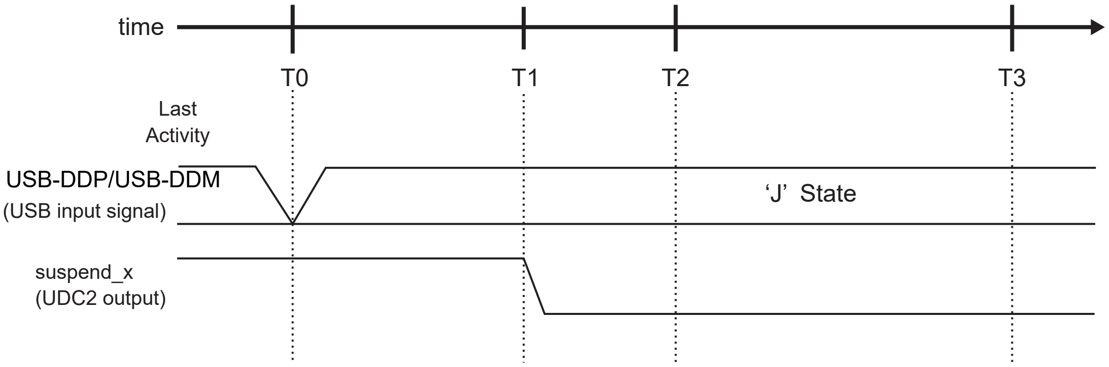


Figure 17-28 Suspend operation timing

- T0: End of bus activity

  When the end of bus activity from the host (the end of packet) is detected, UDC2 starts counting to recognize suspend.

- T1: Recognition of suspend (3 ms after T0)

  When the "FS-J" is detected for more than 3 ms after T0, UDC2 recognizes suspend and drives suspend_x "Low".

- T2: Remote wakeup start enable (5 ms after T0)

  Resume operation from the device (remote wakeup) is enabled 5 ms after T0.

- T3: Transition to suspend state (10 ms after T0)

  The device must enter the suspend state no later than 10 ms after T0. Processes required of the device system to enter the suspend state, such as stopping the CLK_U, must be performed during this period.

  It is necessary to control USB clock control circuit to stop the CLK_U to UDC2.

(2) Notes on Suspend Operation

- Internal registers during the suspend state

  During the suspend state, UDC2 retains the internal register values, the contents of FIFOs, and the state of each flag. These values and states are also retained after the suspend state is exited by resume operation.

  When the CLK_H to UDC2 is stopped, the internal registers in UDC2 cannot be accessed via PVC-I/F and EP-I/F.

17.9.1.4 Resume Operation

(1) Resume Operation by the Host

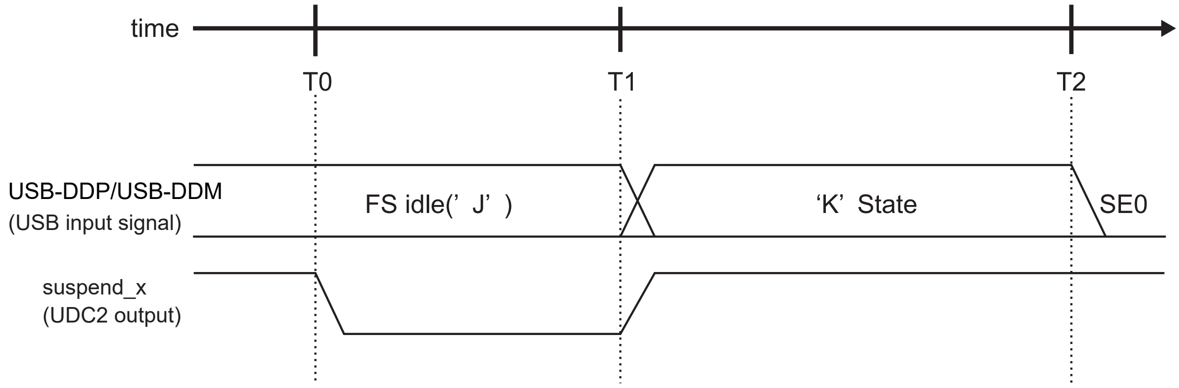

time
T0
T1
T2
USB-DDP/USB-DDM
(USB input signal)
FS idle(' J')
'K' State
SE0
suspend_x
(UDC2 output)

Figure 17-29 Resume operation timing by the host

- T0: suspend_x output of UDC2 is "Low".
- T1: Start of host resume (No timing specifications)

  The host starts resume operation ("FS-K") at arbitrary timing to wake up the device from the suspend state. At this point, UDC2 sets suspend_x to "High". (Even if the CLK_U to UDC2 is stopped, suspend_x becomes "High").

  During suspend, when CLK_H to UDC2 stops, resume the CLK_H by controlling USB clock control circuit.

  When CLK to UDC2 is stopped, it is necessary to control clk_em.
- T2: End of host resume (more than 20 ms after T1)

  The host resume operation ("FS-K") lasts for more than 20 ms, and completes after "SE0".

(2) Resume Operation by the Device (Remote Wakeup)

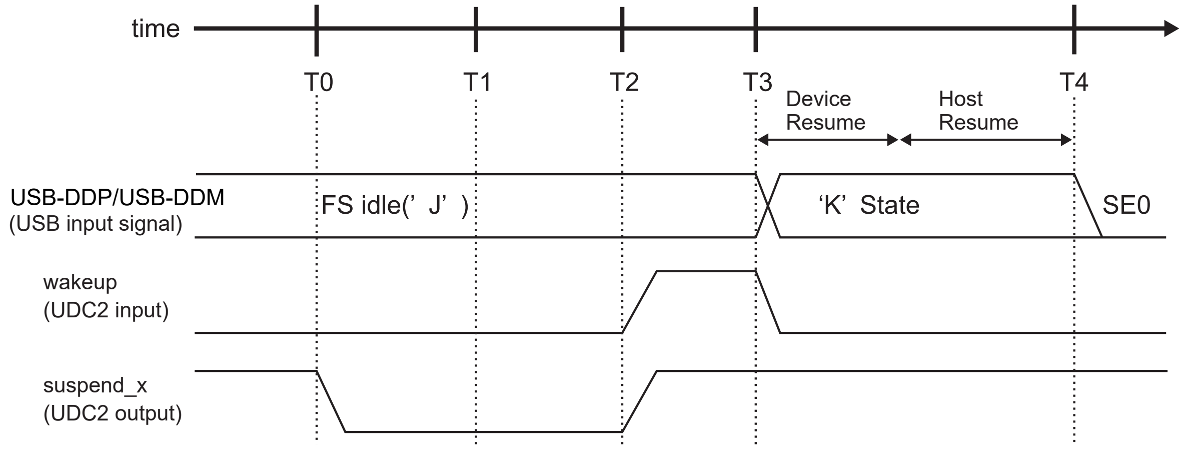

time
T0
T1
T2
T3
T4
Device Resume
Host Resume
USB-DDP/USB-DDM
(USB input signal)
FS idle(' J')
'K' State
SE0
wakeup
(UDC2 input)
suspend_x
(UDC2 output)

Figure 17-30 Remote wakeup operation timing

- T0: suspend_x output of UDC2 is "Low".
- T1: Remote wakeup start enable (more than 2 ms after T0)

  The device can be brought out of the suspend state by using the wakeup input of UDC2. Note that the USB specification prohibits remote wakeup for 5 ms after start of the suspend state. The wakeup signal should be set to "High" a minimum of 2 ms after T0 as 3 ms have already elapsed from the start of suspend operation to T0.
- T2: Wakeup input to UDC2 is "High" (after T1)

  Set the wakeup signal to "High". No timing requirements are specified for this operation. At this point, UDC2 sets suspend_x to "High". (Even if the CLK_H input to UDC2 is stopped, suspend_x becomes "High".) UDC2 requires the clock input to start resume operation ("FSK"). Then, keep wakeup at "High" until clock supply is resumed.
- T3: Start of device resume

  When the CLK_H input to UDC2 is resumed, UDC2 starts the device resume ("FS-K"). The device resume period is approximately 2 ms. After confirming the device resume, the host starts the host resume operation.
- T4: End of host resume (more than 20 ms after T3)

  The host resume operation ("FS-K") lasts for more than 20 ms, and completes after "SE0".

(3) Notes on Resume Operation

The restriction on use of remote wakeup are shown as follows.

To support remote wakeup as the device system, the device must notify the host in the Configuration descriptor that the remote wakeup function is enabled. Even if remote wakeup is supported, it is disabled by default. Remote wakeup can only be used after it is enabled by a request from the host. Use of remote wakeup using the wakeup input is allowed only when these conditions are satisfied.

When using this function, be sure to refer to 17.8 of the USB 2.0 Specification which offers detailed description.

17.9.2 Appendix B About Setting an Odd Number of Bytes as MaxPacketSize

17.9.2.1 Setting an odd number in the UDFS2EPxMSZ

The USB specification allows MaxPacketSize (hereafter referred to as MPS) of each EP to be set as either an odd or even number of bytes for Isochronous and Interrupt transfers. (For Control and Bulk transfers, only an even number can be set.)

In UDC2, MPS is set through UDFS2EPxMSZ<max_pkt>. The EP FIFOs of UDC2 only support even numbers of bytes. It is therefore recommended that MSP be set as an even number of bytes as a general rule.

When using MPS by odd bytes, it is possible to make <max_pkt> into odd number. However, there are restrictions shown in Table 17-6 by the access method of a bus. In the case of EP direct access, an odd number cannot be set in <max_pkt> for a transmit EP. In this case, an even number should be set in <max_pkt> and write accesses to the EP FIFO should be controlled to implement an odd number of maximum write bytes. (For example, when MPS is 1023 bytes, <max_pkt> should be set to 1024 bytes.)

Table 17-6 Restrictions on the setting of max_pkt

	Receive EP	Transmit EP
Common bus access (PVCI-IF)	An odd or even number can be set	An odd or even number can be set
EP direct access (EP-I/F)	An odd or even number can be set	Only an even number can be set.

Based on the above, the following pages describe how to set an odd number of bytes as MPS for each bus access method.

(1) Receive EP and common bus access

Either an odd or even number of bytes can be set in <max_pkt>. The access method is the same for both cases.

(2) Transmit EP and common bus access

Either an odd or even number of bytes can be set in <max_pkt>.

However, the following points must be observed in making common bus accesses for writing the maximum number of bytes with max_pkt = odd number.

The following shows an example in which <max_pkt> = 5 and the maximum number of bytes (5 bytes) are to be written.

- In the last access (5th byte), make sure that udc_be = 01.
- Because it is access of MPS, Do not issue the EP_EOP command in the UDFS2CMD.

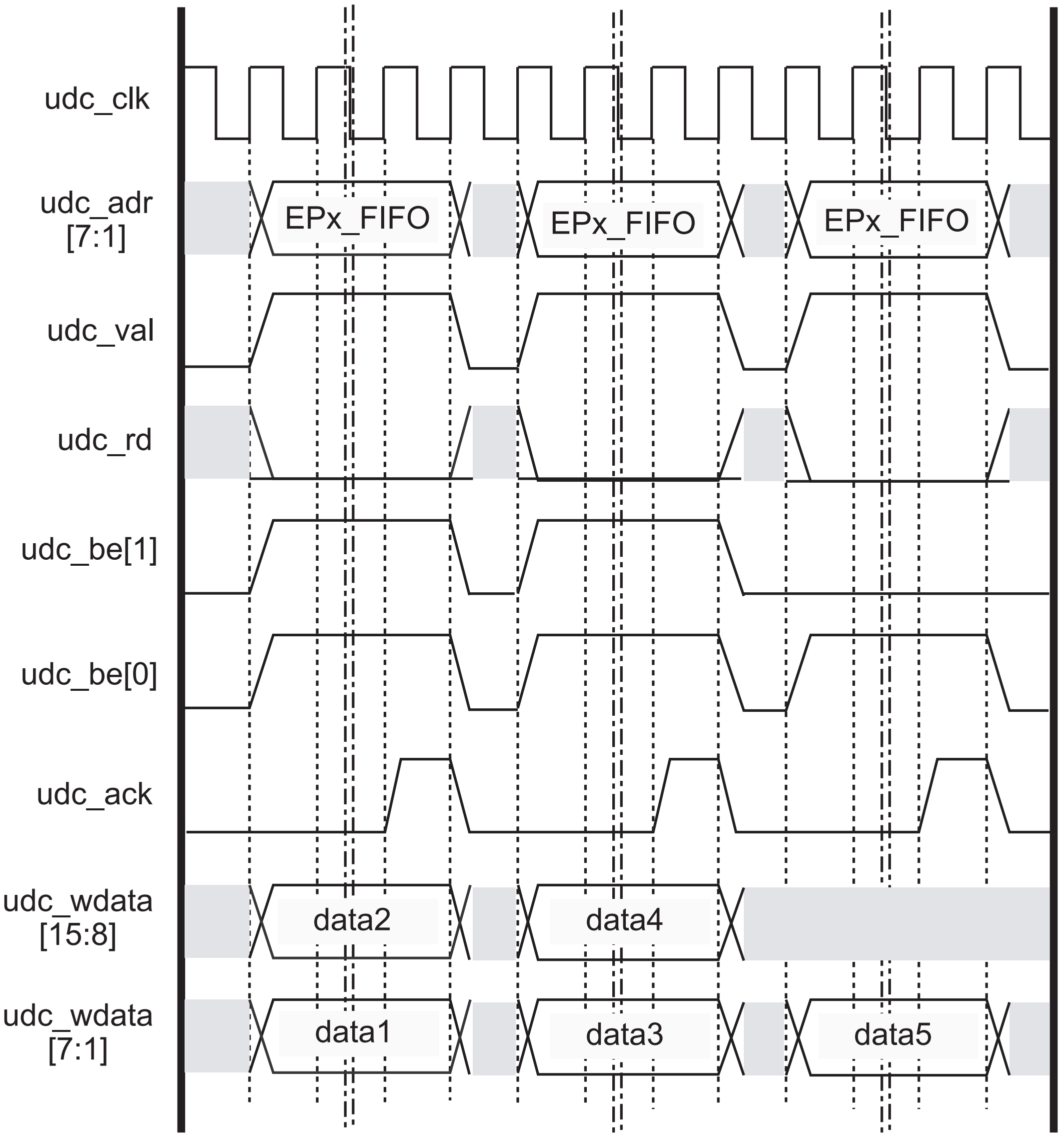

udc_clk
udc_adr [7:1]
EPx_FIFO
EPx_FIFO
EPx_FIFO
udc_val
udc_rd
udc_be[1]
udc_be[0]
udc_ack
udc_wdata [15:8]
data2
data4
udc_wdata [7:1]
data1
data3
data5

Figure 17-31 MPS write access with max_pkt = odd number (common bus access)

(3) Receive EP and EP direct access

Either an odd or even number can be set in <max_pkt>. The access method is the same for both cases.

(4) Transmit EP and EP direst access

Only an even number of bytes can be set in <max_pkt>. To use an odd number of bytes as MPS for a transmit EP, the following settings are required.

- When MPS is 1023
    - Set <max_pkt> is 1024.
    - The maximum number of bytes that can be written to the EP is 1023 bytes. (It is not allowed to write the 1024th byte.)
    - "wMaxPacketSize" of the EP descriptor to be managed by firmware should be set to 1023. (This is the value to be sent to the USB host by the Get Descriptor request.)

The following shows an example in which max_pkt = 1024 and the maximum number of bytes (1023 bytes) are to be written.

- In the last access (1023rd byte), make sure that epx_w_be = 01.

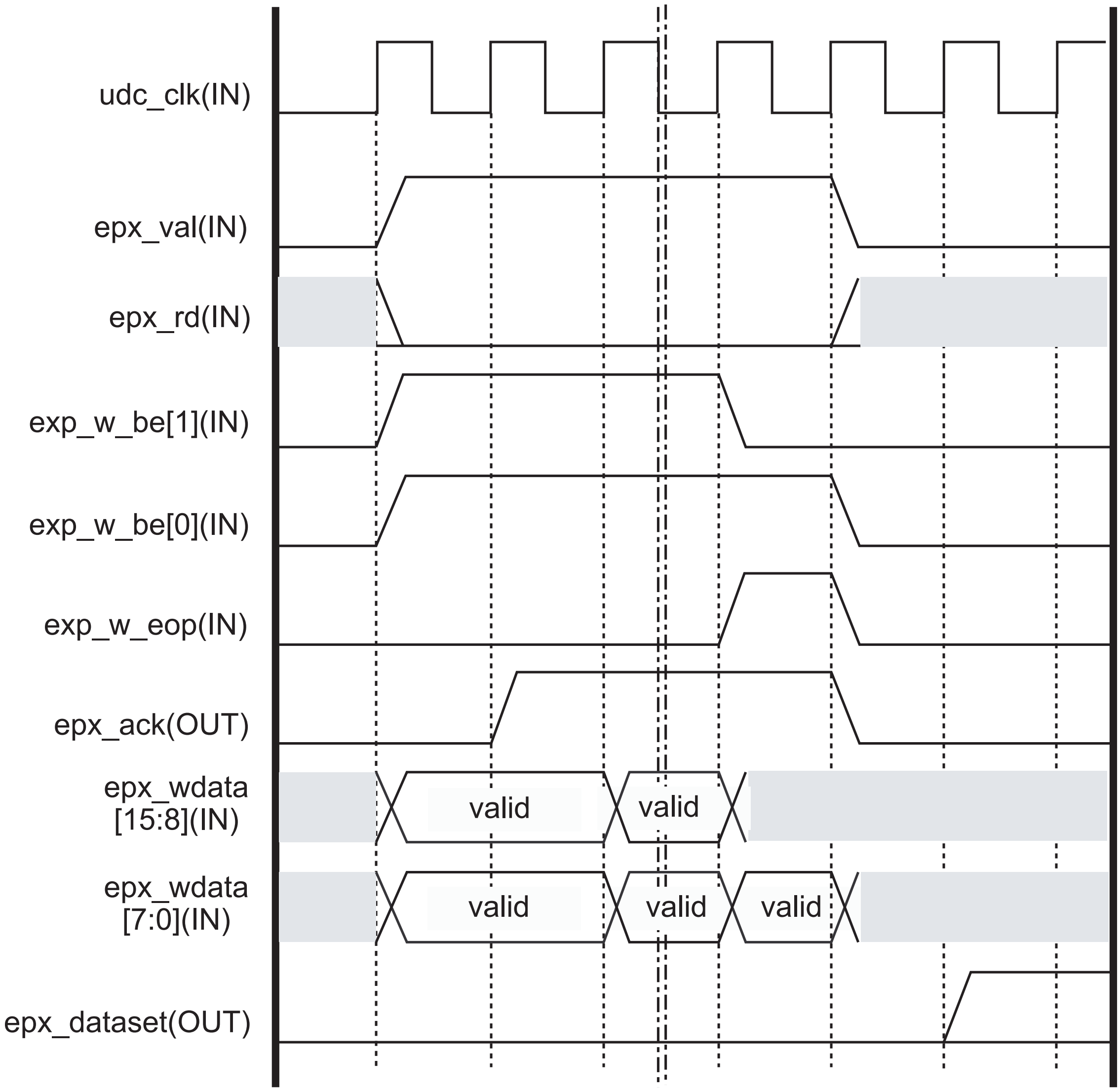

udc_clk(IN)
epx_val(IN)
epx_rd(IN)
exp_w_be[1](IN)
exp_w_be[0](IN)
exp_w_eop(IN)
epx_ack(OUT)
epx_wdata [15:8](IN)
valid
valid
epx_wdata [7:0](IN)
valid
valid
valid
epx_dataset(OUT)

Figure 17-32 MPS (odd number) write access with max_pkt = even number (EP direct access)

17.9.3 Appendix C Isochronous Translator

In Isochronous transfers, the isochronism of data is critical and transfers occur per frame. Therefore, accesses to an EP (FIFO) using Isochronous transfers require a certain level of performance (speed). In UDC2, the access method to each EP can be selected from PVCI-I/F and EP-I/F. The FIFO configuration can be selected from Single mode and Dual mode. However, for an EP using Isochronous transfers, it is recommended to use EP-IF and Dual mode.

17.9.3.1 Accessing an EP using Isochronous transfer

The maximum data payload size is 1023 bytes in FS mode. To transfer 1023 bytes using Dual mode, 2048 bytes of RAM are required. Transfers are performed per frame (1 ms) in FS mode. In FS mode, One transactions can be made in one frame.

(Information such as the payload size and the number of transactions must be set in the relevant UDC2 register. This information must also be managed by software as the EP descriptor information to be sent to the host.)

17.9.3.2 Restrictions on command usage to EP when using Isochronous transfer

Compared to other transfers, Isochronous transfers have certain restrictions on handshake, toggle, the number of transactions in a frame, etc., limiting the types of commands that can be used. As a general rule, commands must not be issued to EPs during Isochronous transfers. While a request is being processed, the EP_Reset or EP_Invalid command may be used as necessary.

(When using PVCI-I/F as the EP access method, use the EP_EOP command.)

(About the Appendix)

For descriptions concerning the USB Specification, be sure to check the USB Specification (revision 2.0).

18. Remote control signal preprocessor(RMC)

18.1 Basic operation

Remote control signal preprocessor (hereafter referred to as RMC) receives a remote control signal of which carrier is removed.

18.1.1 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

18.2 Block Diagram

Figure 18-1 shows the block diagram of RMC.

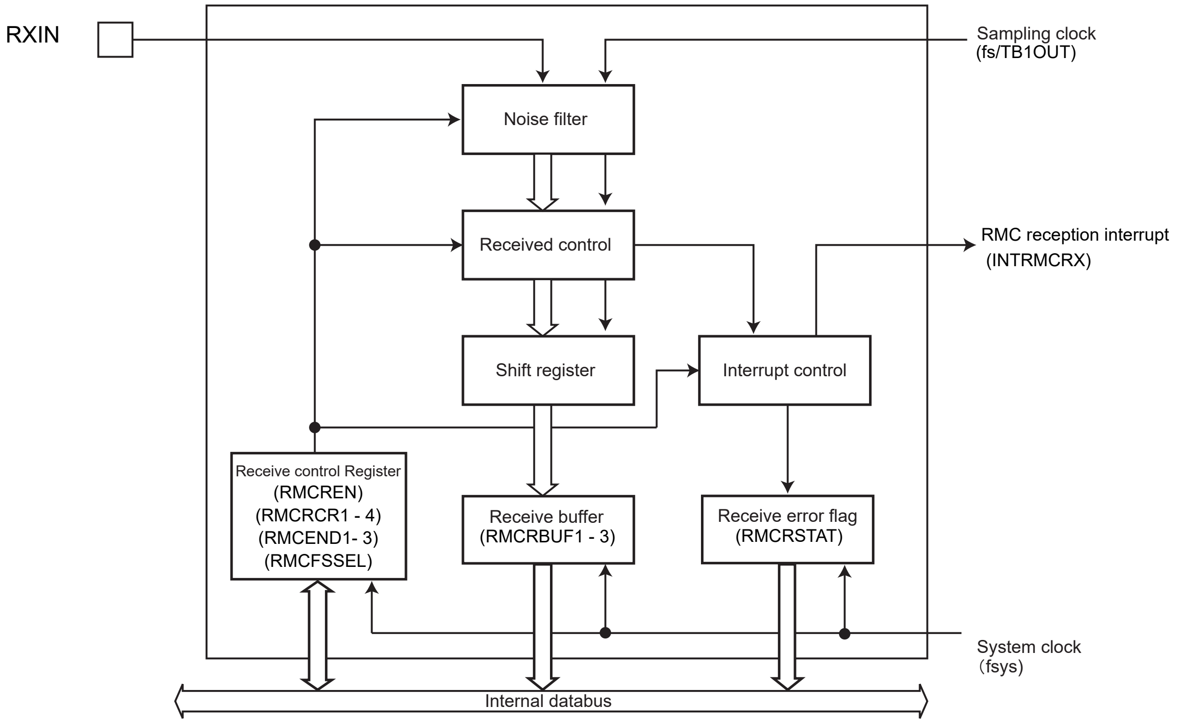

RXIN
Sampling clock
(fs/TB1OUT)
Noise filter
Received control
RMC reception interrupt
(INTRMCRX)
Shift register
Interrupt control
Receive control Register
(RMCREN)
(RMCRCR1 - 4)
(RMCEND1- 3)
(RMCFSSEL)
Receive buffer
(RMCRBUF1 - 3)
Receive error flag
(RMCRSTAT)
System clock
(fsys)
Internal databus

Figure 18-1 Block diagram of RMC

18.3 Registers

18.3.1 Register List

Addresses and names of RMC control registers are shown below.

Base Address = 0x400E_7000

Register		Address(Base+)
Enable Register	RMCEN	0x0000
Receive Enable Register	RMCREN	0x0004
Receive Data Buffer Register 1	RMCRBUF1	0x0008
Receive Data Buffer Register 2	RMCRBUF2	0x000C
Receive Data Buffer Register 3	RMCRBUF3	0x0010
Receive Control Register 1	RMCRCR1	0x0014
Receive Control Register 2	RMCRCR2	0x0018
Receive Control Register 3	RMCRCR3	0x001C
Receive Control Register 4	RMCRCR4	0x0020
Receive Status Register	RMCRSTAT	0x0024
Receive End bit Number Register 1	RMCEND1	0x0028
Receive End bit Number Register 2	RMCEND2	0x002C
Receive End bit Number Register 3	RMCEND3	0x0030
Source Clock selection Register	RMCFSSEL	0x0034

18.3.2 RMCEN(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 bit first. If the operation is disabled, all the clocks for RMC except for the enable register are stopped, and it can reduce power consumption. If RMC is enabled and then disabled, the settings in each register remain intact.

18.3.3 RMCREN(Receive Enable Register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	RMCREN	R/W	Reception 0: Disabled 1: Enabled Controls reception of RMC. Setting this bit to "1" enables reception.

Note: Enable the <RMCREN> bit after setting the RMCRCR1, RMCRCR2, and RMCRCR3.

18.3.4 RMCRBUF1(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)

18.3.5 RMCRBUF2(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)

18.3.6 RMCRBUF3(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.

18.3.7 RMCRCR1(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 RMCRCR2<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

18.3.8 RMCRCR2(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

18.3.9 RMCRCR3(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".

18.3.10 RMCRCR4(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]

18.3.11 RMCRSTAT(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.

18.3.12 RMCEND1(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

18.3.13 RMCEND2(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

18.3.14 RMCEND3(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 RMCEND1, RMCEND2 and RMCEND3, it is able to set three kinds of the receive data bit.

Note 2: To use the RMCEND1, RMCEND2 and RMCEND3 is in combination with the maximum data bit cycle.

18.3.15 RMCFSSEL(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(TB1OUT) For the Sampling of RMC function, It is able to set the Low Frequency Clock (32.768kHz) or Timer output (TB1OUT). The Setting range of Timer output by TB1OUT is from 30 to 34kHz.

Note: To Change the sampling clock by using the RMCFSSEL, disable the RMC operation first by using the RMCEN<RMCEN>. Then, enable it again, and set the RMCFSSEL before setting other RMC registers.

18.4 Operation Description

18.4.1 Reception of Remote Control Signal

18.4.1.1 Sampling clock

A remote control signal is sampled by using low-speed 32.768kHz clock (fs).

18.4.1.2 Basic operation

RMC set RMCRSTAT<RMCRLDR> bit when a leader is detected.

At this time, if you set the RMCRCR2<RMCLIEN> bit, leader detection will generate a leader detection interrupt. When a leader detection interrupt occurs, RMCRSTAT<RMCRLIF> bit is set.

After the leader detecting, each data bit is determined as "0" or "1" in sequence. The results are stored in RMCRBUF1, RMCRBUF2 and RMCRBUF3 registers up to 72 bits. By setting RMCRCR2< RMCEDIEN> bit, a remote control signal input falling edge interrupt can be generated in each falling edge of data bit. When a remote control signal input falling edge interrupt is generated, RMCRSTAT< RMCEDIF > bit 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.

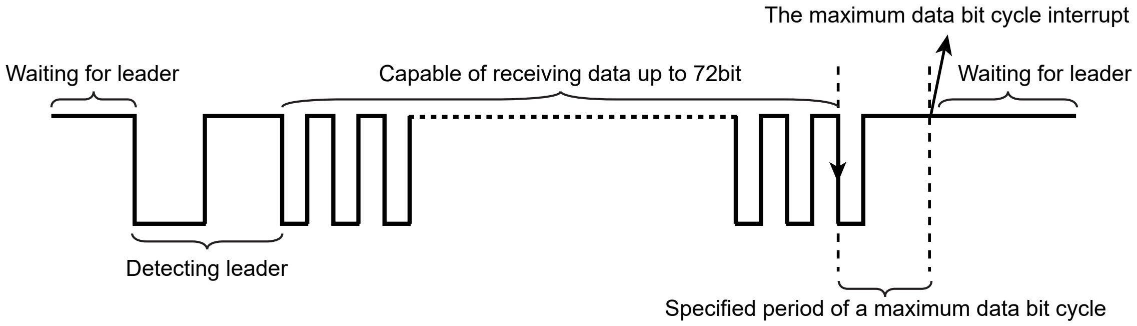

The maximum data bit cycle interrupt
Waiting for leader
Capable of receiving data up to 72bit
Waiting for leader
Detecting leader
Specified period of a maximum data bit cycle

Figure 18-2 Data reception completed by detecting the max data bit cycle

18.4.1.3 Preparation

Before starting receiving process, configure how to receive remote control signal using the Remote Control Signal Receive Control Registers (RMCRCR1, RMCRCR2 and RMCRCR3, RMCRCR4).

(1) Settings of Noise Cancelling Time

Configure noise cancelling time with the RMCRCR4 <RMCNC[3:0]> bit.

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".

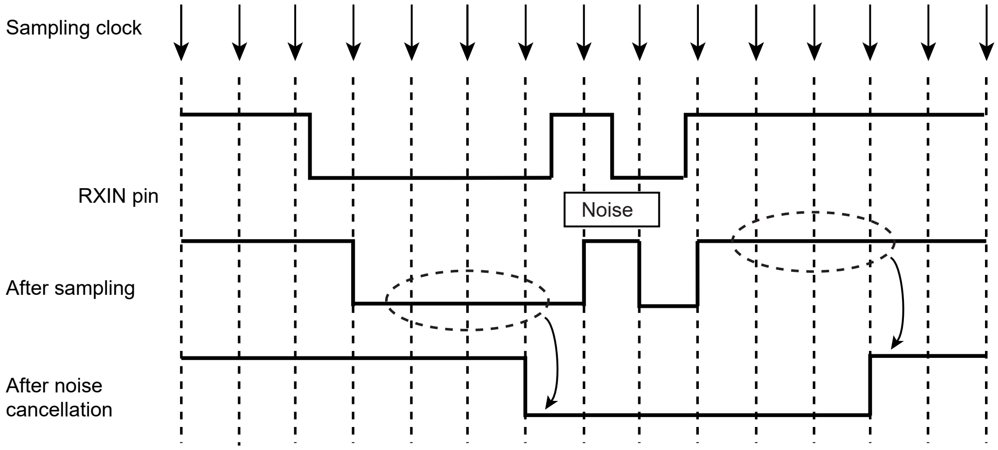


Figure 18-3 Noise Cancel (In the case of RMCRCR4="0011" (3 Cycles))

(2) Settings of Detecting Leader

Set the leader cycle and a low width of the leader to RMCRCR1 <RMCLLMIN[7:0]> <RMCLLMAX[7:0]> <RMCLCMIN[7:0]> <RMCLCMAX[7:0]> bits. 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

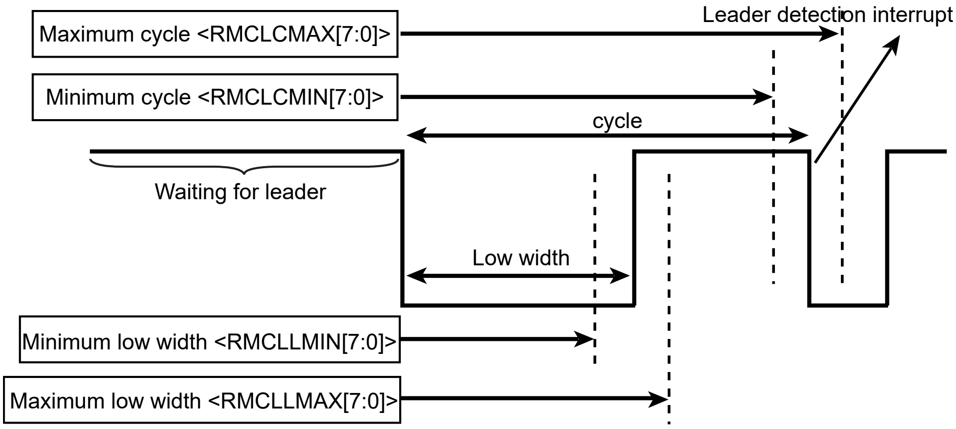

Maximum cycle <RMCLCMAX[7:0]>
Leader detection interrupt
Minimum cycle <RMCLCMIN[7:0]>
cycle
Waiting for leader
Low width
Minimum low width <RMCLLMIN[7:0]>
Maximum low width <RMCLLMAX[7:0]>

Figure 18-4 Leader wave form and the RMCRCR1 register settings

If you want to generate an interrupt when detecting a leader, configure the RMCRCR2 <RMCLIEN> bit.

A remote control signal without a leader cannot generate a leader detection interrupt.

(3) Setting of 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"18.4.1.8 Receiving a Remote Control Signal in a Phase Method".

1. Determination by threshold.

   Configure a threshold value to RMCRCR3<RMCDATL[6:0]> bit which determines data bit as "0" or "1." If the determination value is equal to threshold value or more, it is determined as "1." If the determination value is less than threshold value, it is determined as "0."

2. Determination by falling edge interrupt inputs.

   By setting "1" to the RMCRCR2<RMCEDIEN> bit, a remote control signal input falling edge interrupt can be generated in each falling edge of the data bit. Using this interrupt together with a timer enables the determination to be done by software.

The followings shows the determination method of data bit.

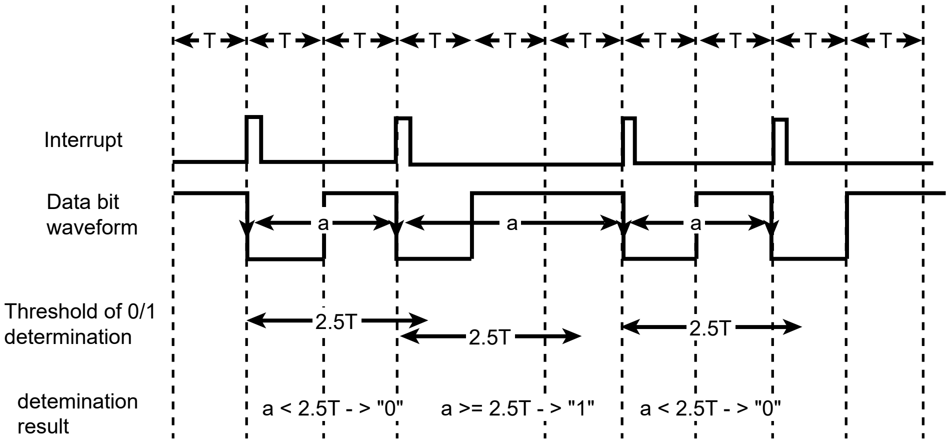

T
T
T
T
T
T
T
T
T
T
Interrupt
Data bit
waveform
a
a
a
Threshold of 0/1
determination
2.5T
2.5T
2.5T
detemination
result
a < 2.5T - > "0"
a >= 2.5T - > "1"
a < 2.5T - > "0"

Figure 18-5 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. Completetion by the maximum data bit cycle

   To complete reception by detecting a maximum data bit cycle, you need to configure the RMCRCR2 <RMCDMAX[7:0]> bits.

   If the falling edge of the data bit cycle isn't monitored after time specified as threshold in the <RMCDMAX[7:0]> bits, a maximum data bit cycle is detected. The detection completes reception and generates an interrupt.After interrupt inputs generated, RMCRSTAT< RMCDMAXIF > bit is set to "1".

   To complete reception by setting the number of receive data is set a RMCEND 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 RMCEND 1 to 3 register of each <RMCEND1>, <RMCEND2>, <RMCEND3>, it occurs by an MAX interrupt in data bit period.

   As specified to RMCEND3 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.

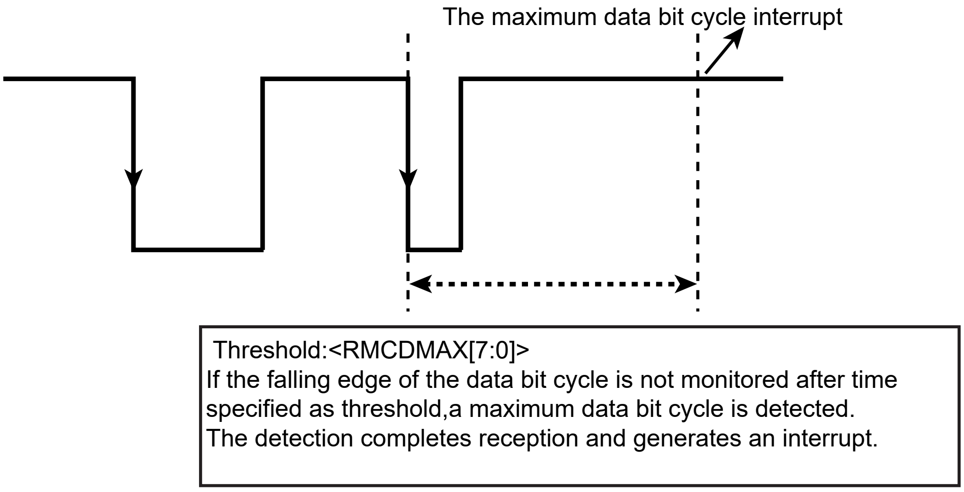

The maximum data bit cycle interrupt
Threshold:<RMCDMAX[7:0]>
If the falling edge of the data bit cycle is not monitored after time specified as threshold,a maximum data bit cycle is detected.
The detection completes reception and generates an interrupt.

Figure 18-6 Completetion by the maximum data bit cycle

2. Completetion by detecting low width

   To complete reception by detecting the low width, you need to configure the RMCRCR2 <RMCLL[7:0]> bits.

   After the falling edge of the data bit is detected, if the signal stays low longer than specified, excess low width is detected. The detection completes reception and generates an interrupt.

   After interrupt inputs generated, RMCRSTAT<RMCLOIF> bit is set to "1."

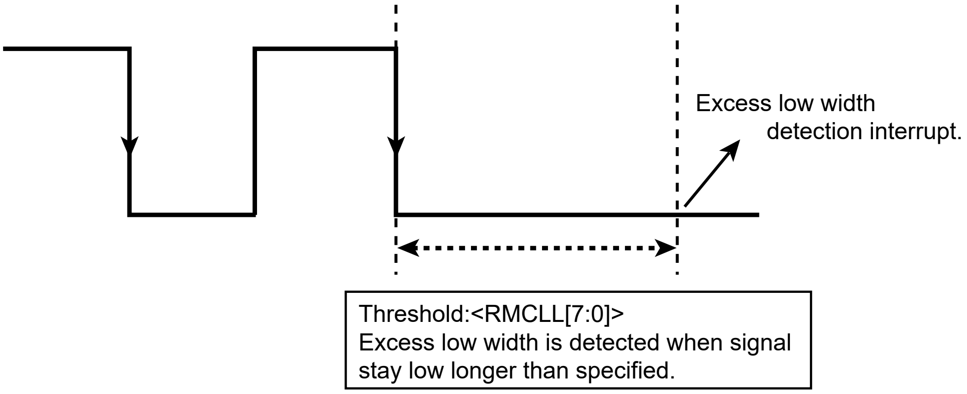

Excess low width
detection interrupt.
Threshold:<RMCLL[7:0]>
Excess low width is detected when signal stay low longer than specified.

Figure 18-7 Completetion by detecting low width

18.4.1.4 Enabling Reception

By enabling the RMCREN <RMCREN> bit after configuring the RMCRCR1, RMCRCR2, RMCRCR3 and RMCRCR4 registers, RMC is ready for reception. Detecting a leader initiates reception.

Note: Changing the configurations of the RMCRCR1, RMCRCR2, RMCRCR3, RMCRCR4, RMCEND1, RMCEND2 or RMCEND3 registers during reception may harm their proper operation. Be careful if you change them during reception.

18.4.1.5 Stopping Reception

RMC stops reception by clearing the RMCREN <RMCREN> bit to "0" (reception disabled).

Clearing this bit during reception stops reception immediately and the received data is discarded.

18.4.1.6 Receiving Remote Control Signal without Leader in Waiting Leader

Setting RMCRCR2 <RMCLD> enables RMC to receive signals with or without a leader.

By setting RMCRCR2 <RMCLD>, RMC starts receiving data if it recognizes a signal of which low width is shorter than a maximum low width of leader detection specified in the RMCRCR1 <RMCLLMAX [7:0]> bits. RMC keeps receiving data until the final data bit is received.

If RMCRCR2 <RMCLD> is enabled, the same settings of error detection, reception completion and data bit determination of 0 or 1 are applied regardless of whether a signal has a leader or not.

Thus receivable remote control signals are limited.

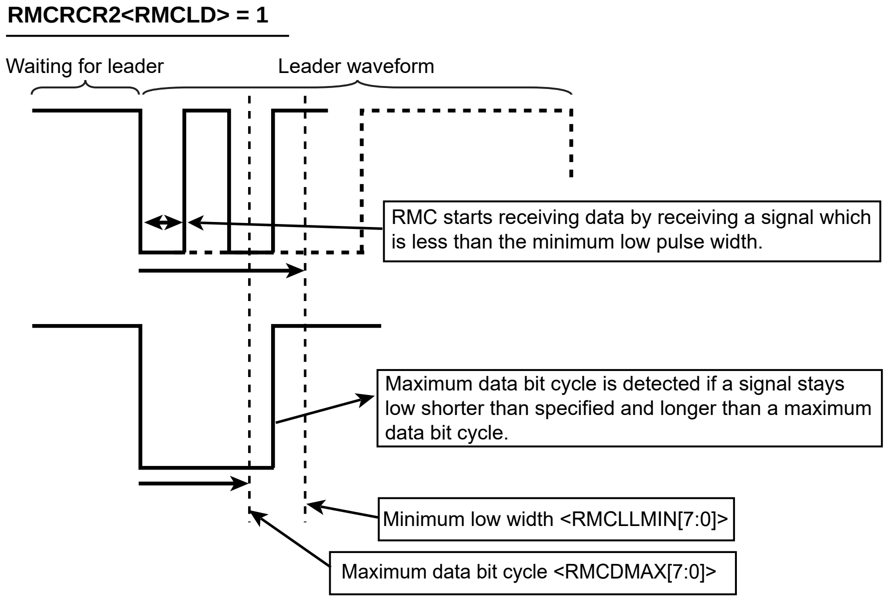

RMCRCR2<RMCLD> = 1
Waiting for leader
Leader waveform
RMC starts receiving data by receiving a signal which is less than the minimum low pulse width.
Maximum data bit cycle is detected if a signal stays low shorter than specified and longer than a maximum data bit cycle.
Minimum low width <RMCLLMIN[7:0]>
Maximum data bit cycle <RMCDMAX[7:0]>

Figure 18-8 Receiving Remote ControlRCR2<RMCLD>="1")

18.4.1.7 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 RMCRCR4 <RMCPO> bit to "1".

This is because RMC is configured to detect a data bit cycle from the falling edge

To detect a leader, configure only low-pulse width of the leader with the <RMCLLMAX[7:0]> = "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 RMCRCR3 <RMCDATL[6:0]>.

The maximum data bit cycle is configured with the <RMCDMAX[7:0]> of the RMCRCR2.

To complete data reception, configure the maximum data bit cycle with <RMCDMAX[7:0]> of the RMCRCR2, 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.

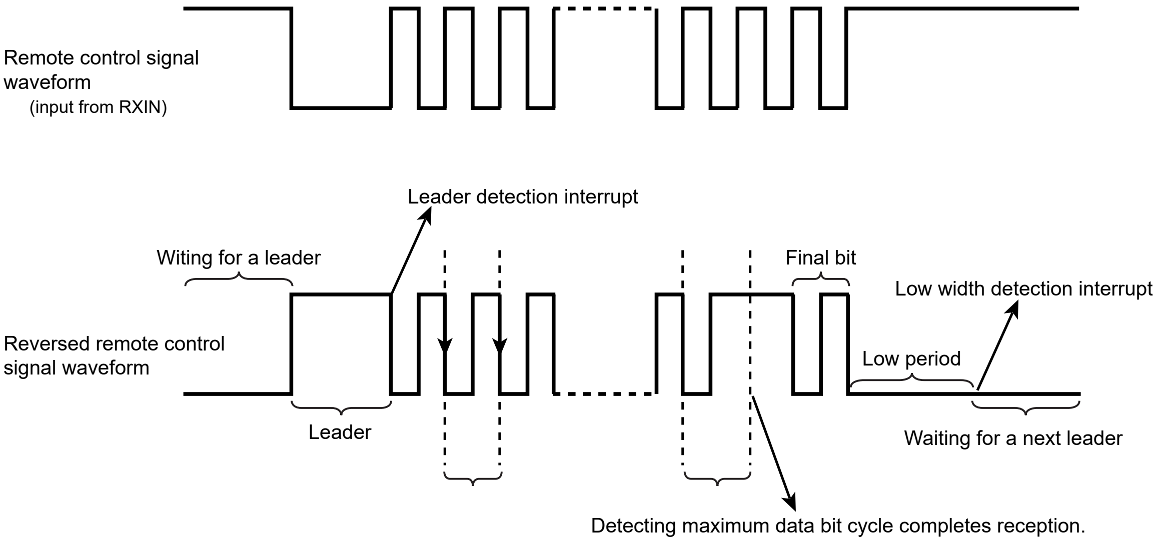

Remote control signal waveform
(input from RXIN)
Leader detection interrupt
Witing for a leader
Final bit
Low width detection interrupt
Reversed remote control signal waveform
Low period
Leader
Waiting for a next leader
Detecting maximum data bit cycle completes reception.

Figure 18-9 A Leader only with Low Width

18.4.1.8 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 RMCRBUF1, RMCRBUF 2 and RMCRBUF3.

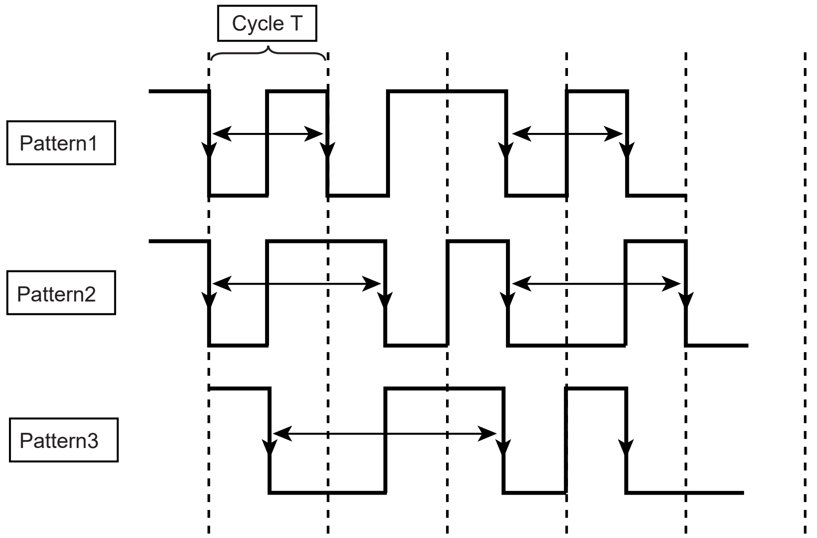
By setting RMCRCR2<RMCPHM> = "1", RMC enables to receive a remote control signal in the phase method. Each threshold can be configured with the RMCRCR3 <RMCDATL[6:0]> bits and <RMCDATH[6:0]> bits.

Two thresholds are used to distinguish three waveform patterns. On condition that a cycle between two falling edges is "T", three patterns show cycles of 1T, 1.5T and 2T. Details of the two thresholds are shown below.

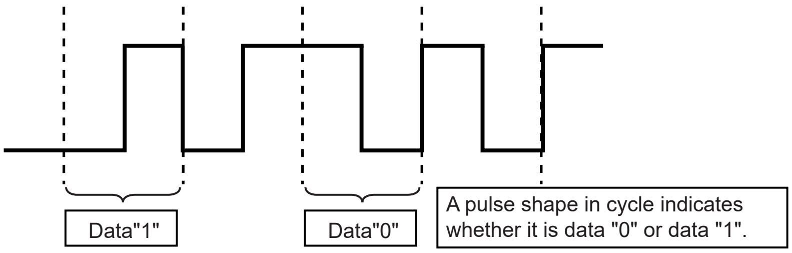
	Determined by	Threshold	Register bits to set
Threshold 1	Pattern 1 & pattern 2	1T to 1.5T	RMCRCR3<RMCDATL[6:0]>
Threshold 2	Pattern 2 & pattern 3	1.5T to 2T	RMCRCR3<RMCDATH[6:0]>

To determine a remote control signal in the phase method, three patterns of data waveform and preceding data are required. In addition, the signal needs to start from data "11".

Waveform pattern in phase method



Remote control signal data in phase method



A pulse shape in cycle indicates whether it is data "0" or data "1".

Figure 18-10 Waveform pattern in phase method and the example of data

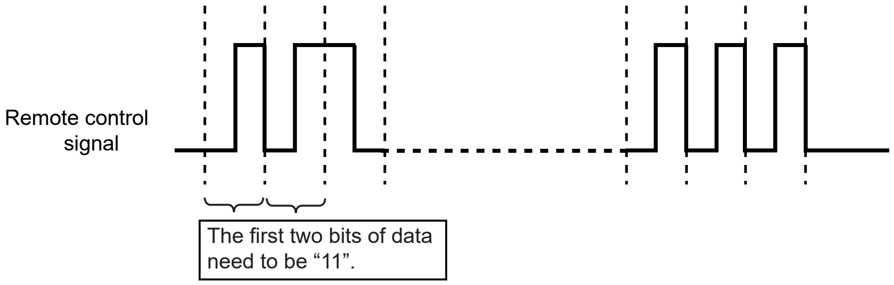

Remote control signal
The first two bits of data need to be “11”.

Figure 18-11 The waveform pattern in phase method

19. Analog / Digital Converter (ADC)

19.1 Features

TMPM367FDFG contain two units of 12-bit sequential-conversion analog/digital converters (ADC) with 8 analog input channels.

These 8 analog input channels (AINA0 to AINA3, AINB0 to AINB3) are also used as input/output ports.

There are two operation modes: one is the single unit mode operation, in which each unit operate separately, and the other is the dual unit mode, in which the interleave control circuit (ADCINTLV) generates trigger signals for AD conversion performed by two AD conversion units.

The dual unit mode offers two operation modes: interleave mode and trigger start mode.

Each unit receives trigger signals from ADCINTLV and operates according to the specified conditions.

In the interleave mode, ADCINTLV receives a trigger signal and outputs a trigger signal to the unit A. After the specified time, ADCINTLV outputs a trigger signal to the unit B.

In the trigger start mode, the ADCINTLV outputs a trigger signal to the unit A for the odd number of trigger input, and outputs a trigger signal to the unit B for the even number of trigger input.

This operation enables the result of AD conversion to be returned in half the time that each conversion takes (min. 0.5μs).

A 12-bit AD converter has the features shown below.

- Starting normal AD conversion and top-priority AD conversion
  - Software activation
  - Hardware activation with an external trigger input ($\overline{\text{ADTRGA}}$)
  - Activation with a 16-bit timer (dual unit mode only)
- Operation modes of Normal AD conversion
  - Fixed-channel single conversion mode
  - Channel scan single conversion mode (single unit mode only)
  - Fixed-channel repeat conversion mode
  - Channel scan repeat conversion mode (single unit mode only)
- Operation modes of top-priority AD conversion
  - Fixed-channel single conversion mode (single unit mode only)
- Normal AD conversion end interrupt and top-priority AD conversion end interrupt
- Normal AD conversion function and top-priority AD conversion contain the below status flags.
  - A flag that indicates AD conversion results are valid and a flag that indicates overwrite.
  - AD conversion completion flag and AD conversion busy flag.
- AD monitor function
  - If an arbitrary condition for comparison is satisfied, an interrupt occurs.
- AD conversion clock is controllable from fc to fc/16.
- Stand-by mode is supported.

19.2 Configuration

Figure 19-1 shown the block diagram of the AD converter.

ADC Pin Connection

Dual Unit mode (Setting Interleave / Trigger start mode)

Unit A and Unit B are operated in the same setting dependly each other.
An AD conversion is started by an external pin, timer or soft trigger <SWATRG>.
Select fixed-channel mode.

Single Unit mode 1 (Setting single unit mode)

Unit A and Unit B are operated in the each setting independly.
An AD conversion is started by an external pin or soft trigger <ADS> <HPADS>.
Timer can not be selected as starting conversion trigger

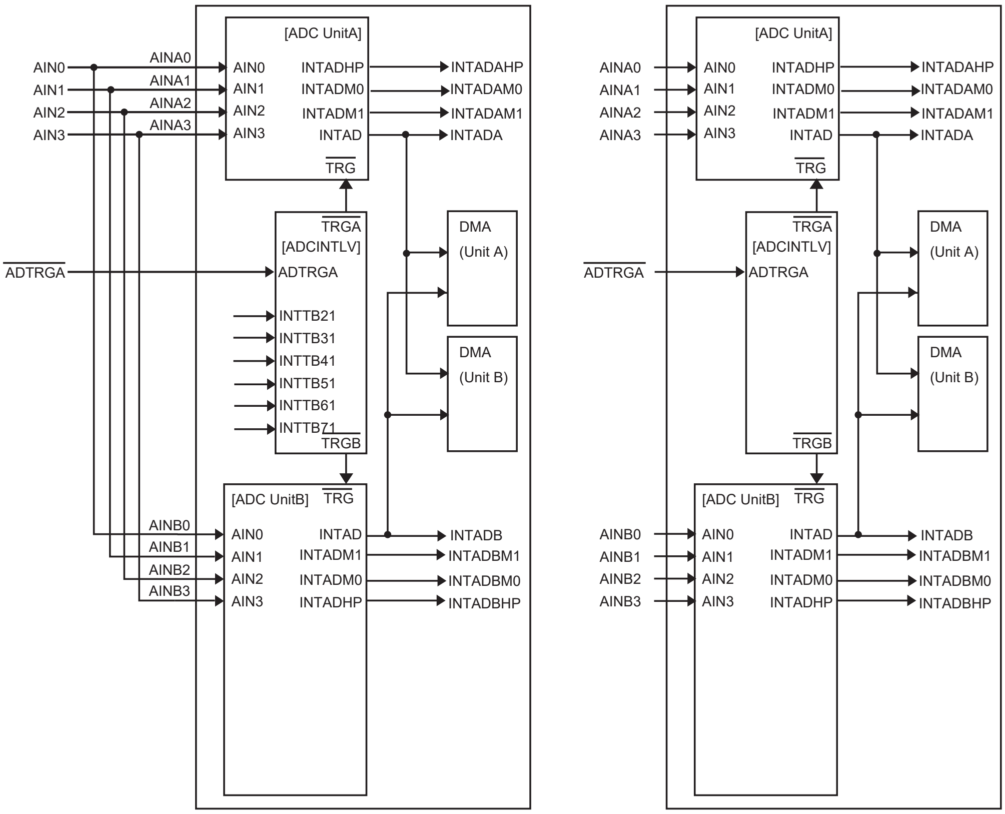

AIN0
AIN1
AIN2
AIN3
AINA0
AINA1
AINA2
AINA3
[ADC UnitA]
AIN0
AIN1
AIN2
AIN3
INTADHP
INTADM0
INTADM1
INTAD
TRG
INTADAHP
INTADAM0
INTADAM1
INTADA
TRGA
[ADCINTLV]
ADTRGA
ADTRGA
INTTB21
INTTB31
INTTB41
INTTB51
INTTB61
INTTB71
TRGB
DMA
(Unit A)
DMA
(Unit B)
[ADC UnitB]
TRG
AINB0
AINB1
AINB2
AINB3
AIN0
AIN1
AIN2
AIN3
INTAD
INTADM1
INTADM0
INTADHP
INTADB
INTADBM1
INTADBM0
INTADBHP
AINA0
AINA1
AINA2
AINA3
[ADC UnitA]
AIN0
AIN1
AIN2
AIN3
INTADHP
INTADM0
INTADM1
INTAD
TRG
INTADAHP
INTADAM0
INTADAM1
INTADA
TRGA
[ADCINTLV]
ADTRGA
ADTRGA
TRGB
DMA
(Unit A)
DMA
(Unit B)
[ADC UnitB]
TRG
AINB0
AINB1
AINB2
AINB3
AIN0
AIN1
AIN2
AIN3
INTAD
INTADM1
INTADM0
INTADHP
INTADB
INTADBM1
INTADBM0
INTADBHP

Figure 19-1 AD Converter Block Diagram

ADC Pin Connection

Single Unit mode 2 (Setting Interleave mode)

Unit A is operated by the trigger signal $\overline{\text{TRGA}}$ of [ADCINTLV].

Unit B is operated by the soft trigger <ADS> <HPADS>.

Single Unit mode 3 (Setting interleave mode)

Unit A is operated by the soft trigger <ADS> <HPADS>.

Unit B is operated by the trigger signal $\overline{\text{TRGB}}$ of [ADCINTLV].

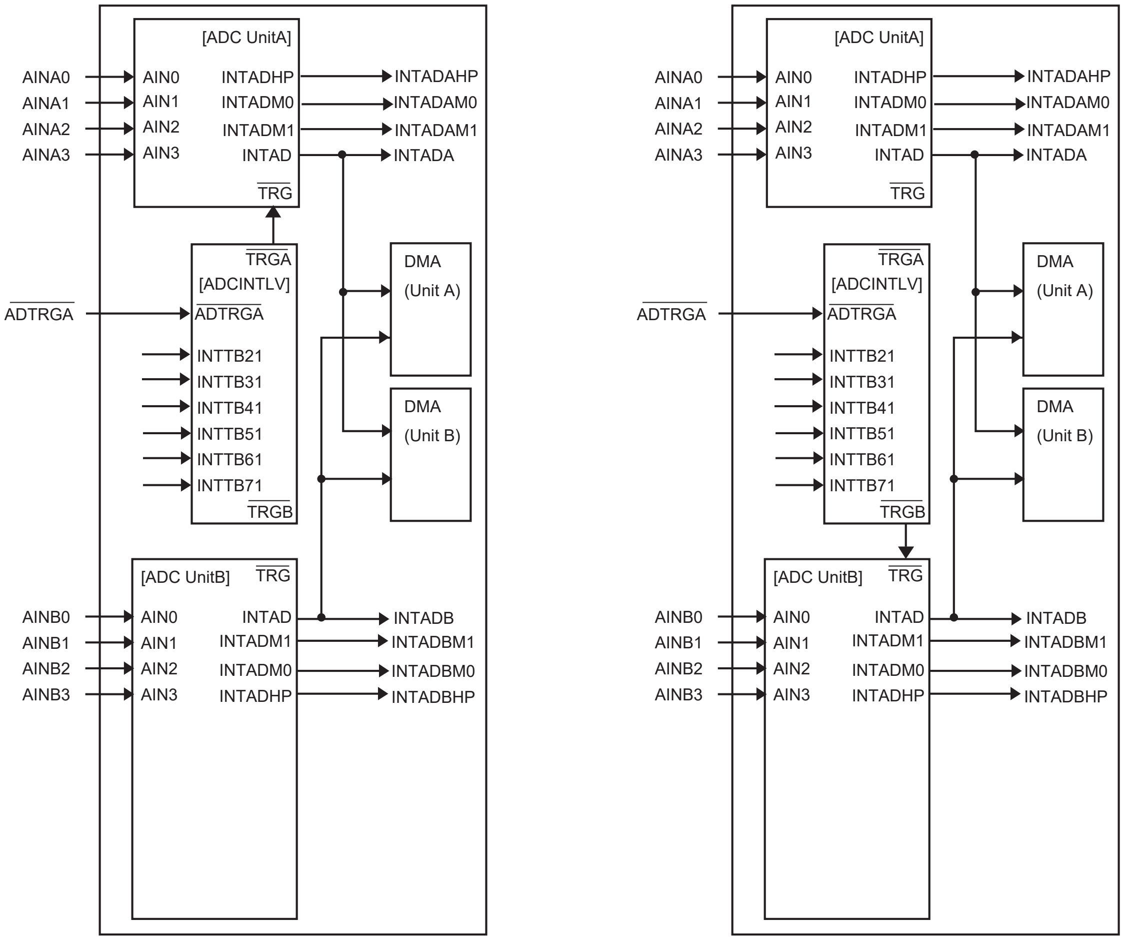

[ADC UnitA]
AINA0
AINA1
AINA2
AINA3
AIN0
AIN1
AIN2
AIN3
INTADHP
INTADM0
INTADM1
INTAD
TRG
INTADAHP
INTADAM0
INTADAM1
INTADA
TRGA
[ADCINTLV]
ADTRGA
ADTRGA
INTTB21
INTTB31
INTTB41
INTTB51
INTTB61
INTTB71
TRGB
DMA
(Unit A)
DMA
(Unit B)
[ADC UnitB]
TRG
AINB0
AINB1
AINB2
AINB3
AIN0
AIN1
AIN2
AIN3
INTAD
INTADM1
INTADM0
INTADHP
INTADB
INTADBM1
INTADBM0
INTADBHP

Figure 19-2 AD Converter Block Diagram

Figure 19-3 shows the block diagram of Unit A.

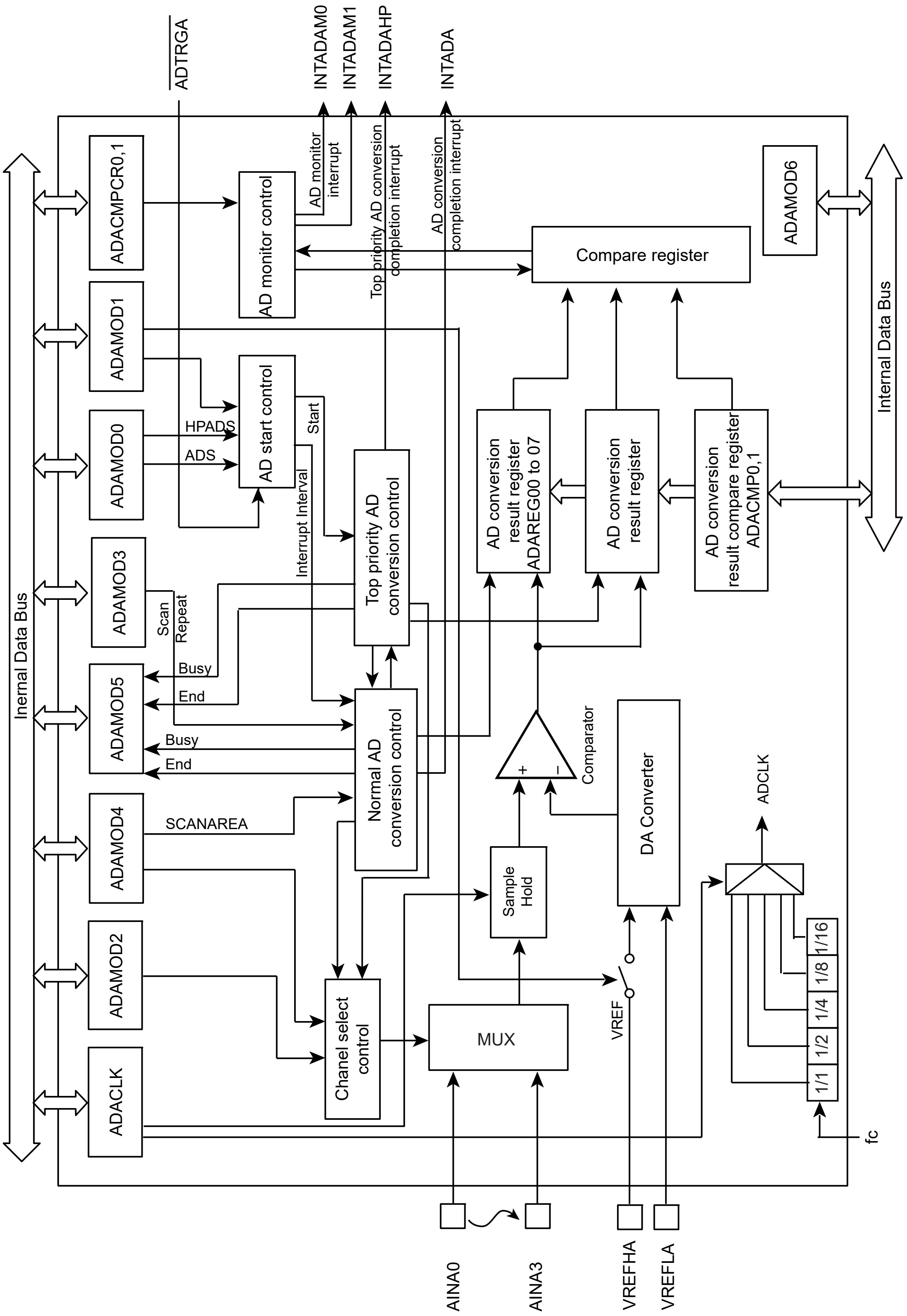

ADTRGA
INTADAM0
INTADAM1
INTADAHP
INTADA
AD monitor interrupt
Top priority AD conversion completion interrupt
AD conversion completion interrupt
ADACMPCR0,1
AD monitor control
Compare register
ADAMOD6
Internal Data Bus
ADAMOD1
AD start control
Start
HPADS
ADS
ADAMOD0
Interrupt Interval
Top priority AD conversion control
AD conversion result register ADAREG00 to 07
AD conversion result register
AD conversion result compare register ADACMP0,1
ADAMOD3
Scan
Repeat
Busy
End
ADAMOD5
Busy
End
Normal AD conversion control
Comparator
+
-
DA Converter
ADCLK
ADAMOD4
SCANAREA
Sample Hold
ADAMOD2
Chanel select control
MUX
VREF
1/16
1/8
1/4
1/2
1/1
fc
ADACLK
Inernal Data Bus
AINA0
AINA3
VREFHA
VREFLA

Figure 19-3 Block Diagram of Unit A

19.3 Registers

19.3.1 Register list

The AD converter is controlled by the Mode Setting Registers (ADAMOD0 through ADAMOD6/ADBMOD0 through ADBMOD6). The results of normal conversion are stored in the conversion result registers (ADAREG00 through ADAREG07/ADBREG00 through ADBREG8). The results of top-priority conversion are stored in the register ADAREGSP/ADBREGSP.

AD monitor function controls the setting value of the conversion result comparison registers (ADACMP0 through ADACMP1/ADBCMP0 through ADBCMP1) according to the setting values in the monitoring interrupt setting register (ADACMPCR0 through ADACMPCR1/ADBCMPCR0 through ADBCMPCR1).

Unit A : Base Address = 0x4005_0000

Unit B : Base Address = 0x4005_1000

Registers			Address (Base+)
Clock Setting Register	ADACLK	ADBCLK	0x0000
Mode Setting Register 0	ADAMOD0	ADBMOD0	0x0004
Mode Setting Register 1	ADAMOD1	ADBMOD1	0x0008
Mode Setting Register 2	ADAMOD2	ADBMOD2	0x000C
Mode Setting Register 3	ADAMOD3	ADBMOD3	0x0010
Mode Setting Register 4	ADAMOD4	ADBMOD4	0x0014
Mode Setting Register 5	ADAMOD5	ADBMOD5	0x0018
Mode Setting Register 6	ADAMOD6	ADBMOD6	0x001C
Reserved	−	−	0x0020
Monitoring Setting Register 0	ADACMPCR0	ADBCMPCR0	0x0024
Monitoring Setting Register 1	ADACMPCR1	ADBCMPCR1	0x0028
AD Conversion Result Compare Register 0	ADACMP0	ADBCMP0	0x002C
AD Conversion Result Compare Register 1	ADACMP1	ADBCMP1	0x0030
AD Conversion Result Register 0	ADAREG00	ADBREG00	0x0034
AD Conversion Result Register 1	ADAREG01	ADBREG01	0x0038
AD Conversion Result Register 2	ADAREG02	ADBREG02	0x003C
AD Conversion Result Register 3	ADAREG03	ADBREG03	0x0040
AD Conversion Result Register 4	ADAREG04	ADBREG04	0x0044
AD Conversion Result Register 5	ADAREG05	ADBREG05	0x0048
AD Conversion Result Register 6	ADAREG06	ADBREG06	0x004C
AD Conversion Result Register 7	ADAREG07	ADBREG07	0x0050
AD Conversion Result Register 8	−	Reserved	0x0054
AD Conversion Result Register 9	−	Reserved	0x0058
AD Conversion Result Register 10	−	Reserved	0x005C
AD Conversion Result Register 11	−	Reserved	0x0060
Reserved	−	−	0x0064
Reserved	−	−	0x0068
Reserved	−	−	0x006C
Reserved	−	−	0x0070
Top-priority Conversion Result Register	ADAREGSP	ADBREGSP	0x0074

Base Address = 0x4005_2000

Registers		Address (Base+)
Control Register for Dual Unit Mode 1	ADILVMO1	0x0000
Control Register for Dual Unit Mode 2	ADILVMO2	0x0004
Control Register for Dual Unit Mode 3	ADILVMO3	0x0008

Note: Access to the "Reserved" address is prohibited.

19.3.2 ADACLK/ADBCLK (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	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	ADSH				-	ADCLK		
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as zero.
7-4	ADSH[3:0]	R/W	Select the ADC sample hold time. 0000: 10 × <ADCLK> 0001: 20 × <ADCLK> 0010: 30 × <ADCLK> 0011: 40 × <ADCLK> 0100: 80 × <ADCLK> 0101: 160 × <ADCLK> 0110: 320 × <ADCLK> 0111 to 1111: Reserved
3	−	R	Read as zero.
2-0	ADCLK[2:0]	R/W	Select the ADC prescaler output. 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101 to 111: Reserved

Note 1: Use in the range of 4MHz ≤ ADCLK ≤ 40MHz. For example, if the settings are fosc = 12MHz and PLL = multiply-by-4, the value is fc = 48MHz. In this case, do not use the condition ADACLK/ADBCLK<ADCLK[2:0]> = "000".

Note 2: To select the <ADCLK >ADC prescaler output, stop AD conversion and the condition must be as follows: ADAMOD1/ADBMOD1<DACON>="0".

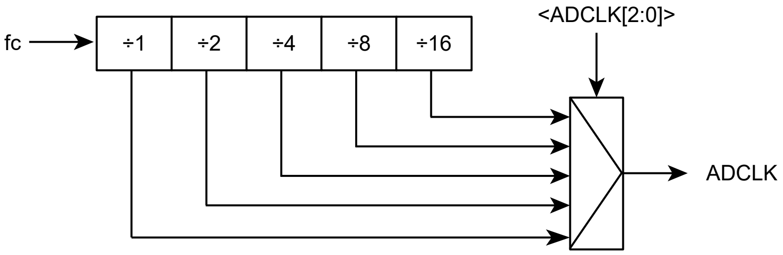


Figure 19-4 AD Conversion Clock (ADCLK)

The below table shows the conversion time. If you select the shortest conversion time, the conversion clock is 40 clocks.

ADCLK setting <ADCLK[2:0]>	Sample hold time setting <ADSH[3:0]>	Conversion time (Tconv)			
		fc=16MHz	fc=48MHz	fc=72MHz	fc=80MHz
000 (fc)	0000 (ADCLK × 10)	2.50 μs	−	−	−
	0001 (ADCLK × 20)	3.13 μs	−	−	−
	0010 (ADCLK × 30)	3.75 μs	−	−	−
	0011 (ADCLK × 40)	4.38 μs	−	−	−
	0100 (ADCLK × 80)	6.88 μs	−	−	−
001 (fc/2)	0000 (ADCLK × 10)	5.00 μs	1.67 μs	1.11 μs	1.00 μs
	0001 (ADCLK × 20)	6.25 μs	2.01 μs	1.39 μs	1.25 μs
	0010 (ADCLK × 30)	7.50 μs	2.50 μs	1.67 μs	1.50 μs
	0011 (ADCLK × 40)	8.75 μs	2.92 μs	1.95μs	1.75 μs
	0100 (ADCLK × 80)	−	4.58 μs	3.06 μs	2.75 μs
	0101 (ADCLK × 160)	−	7.92 μs	5.28 μs	4.75 μs
	0110 (ADCLK × 320)	−	−	9.72 μs	8.75 μs
010 (fc/4)	0000 (ADCLK × 10)	10.00 μs	3.33 μs	2.22 μs	2.00 μs
	0001 (ADCLK × 20)	−	4.17 μs	2.78 μs	2.50 μs
	0010 (ADCLK × 30)	−	5.00 μs	3.34 μs	3.00 μs
	0011 (ADCLK × 40)	−	5.83 μs	3.89 μs	3.50 μs
	0100 (ADCLK × 80)	−	9.17 μs	6.12 μs	5.50 μs
	0101 (ADCLK × 160)	−	−	−	9.50 μs
011 (fc/8)	0000 (ADCLK × 10)	−	6.67 μs	4.45 μs	4.00 μs
	0001 (ADCLK × 20)	−	8.33 μs	5.56 μs	5.00 μs
	0010 (ADCLK × 30)	−	10.0 μs	6.67 μs	6.00 μs
	0011 (ADCLK × 40)	−	−	7.78 μs	7.00 μs
	0100 (ADCLK × 80)	−	−	−	−
100 (fc/16)	0000 (ADCLK × 10)	−	−	8.89 μs	8.00 μs
	0001 (ADCLK × 20)	−	−	−	10.00 μs
	0010 (ADCLK × 30)	−	−	−	−

Note 1: Do not change the setting of the AD conversion clock during AD conversion.

Note 2: Setting the element indicated by "−" in the above table is prohibited. Specify the <ADCLK> settings in the range of conversion time from 1μs to 10μs.

19.3.3 ADAMOD0/ADBMOD0 (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	-	-	-	-	-	-	HPADS	ADS
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as zero.
1	HPADS	W	When the conversion mode is in single unit mode, activate the top-priority AD conversion. 0: Don't care 1: Start conversion "0" is always read.
0	ADS	W	When the conversion mode is in single unit mode, activate normal (software) AD conversion. 0: Don't care 1: Start conversion "0" is always read.

Note 1: To start AD conversion, set ADAMOD1/ADBMOD1<DACON>="1" first, and then, perform software/hardware triggered start by setting ADAMOD0/ADBMOD0<ADS>,<HPADS>. After starting ADAMOD1/ADBMOD1<DACON> = ON("1"), wait for 3μs for stabilization.

Note 2: If the top-priority AD conversion <HPADS> and the normal AD conversion <ADS> start at the same time, the top-priority AD conversion receives preference to start. The normal AD conversion does not start.

19.3.4 ADAMOD1/ADBMOD1 (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	DACON	I2AD	RCUT	-	HPADHWS	HPADHWE	ADHWS	ADHWE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as zero.
7	DACON	R/W	Circuit ON/OFF control 0: OFF 1: ON
6	I2AD	R/W	ADC operation control in IDLE mode (WFI (Controls the Wait For Interrupt) instruction execution time). 0: Stop 1: Operate
5	RCUT	R/W	Controls the reference current between VREFH and VREFL. 0: Energizing only during conversion. 1: Always energized excepting the reset time.
4	−	R	Read as zero.
3	HPADHWS	R/W	Select hardware activation source of top-priority AD conversion. 0: Trigger output of ADCINTLV (interleave control circuit) 1: Reserved Write zero.
2	HPADHWE	R/W	Controls top-priority AD conversion triggered by hardware factors (ADCINTLV). 0: Disable 1: Enable
1	ADHWS	R/W	Hardware activation source to start normal AD conversion. 0: Trigger output of ADCINTLV (interleave control circuit). 1: Reserved Write zero.
0	ADHWE	R/W	Activate normal AD conversion triggered by hardware factor (ADCINTLV). 0: Disable 1: Enable

Note 1: To reduce consumption current used when shifting to IDLE mode with setting <I2AD>="0", or in STOP1/STOP2 mode, set "0" to <DACON> and <RCUT> after AD conversion, and then execute an instruction to move on to stand-by mode.

Note 2: In single unit mode, enable a hardware activation source, either <HPADHWE> or <ADHWE>, of a unit to use. Only an external trigger pin can be used for hardware activation source.

Note 3: In dual unit mode, enable the <ADHWE> of both units. Disable the <HPADHWE>. Activation source can be selected using ADILVMO2<TRGASEL>.

19.3.5 ADAMOD2/ADBMOD2 (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	HPADCH				ADCH			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as zero.
7-4	HPADCH[3:0]	R/W	Select an analog input channel for top-priority AD conversion.(refer toTable 19-1,Table 19-2)
3-0	ADCH[3:0]	R/W	Select an analog input channel for normal AD conversion (refer to Table 19-1,Table 19-2).

Table 19-1 Select input channels for normal AD conversion and top-priority AD conversion (ADAMOD2)

<HPADCH[3:0]>	Analog input channel for top-priority AD conversion	<ADCH[3:0]>	Analog input channel for normal AD conversion
0000	AINA0	0000	AINA0
0001	AINA1	0001	AINA1
0010	AINA2	0010	AINA2
0011	AINA3	0011	AINA3

Table 19-2 Select input channels for normal AD conversion and top-priority AD conversion (ADBMOD2)

<HPADCH[3:0]>	Analog input channel for top-priority AD conversion	<ADCH[3:0]>	Analog input channel for normal AD conversion
0000	AINB0	0000	AINB0
0001	AINB1	0001	AINB1
0010	AINB2	0010	AINB2
0011	AINB3	0011	AINB3
0100 to 1011	Reserved	0100 to 1011	Reserved

19.3.6 ADAMOD3/ADBMOD3 (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	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	ITM			-	-	REPEAT	SCAN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as zero.
6-4	ITM[2:0]	R/W	Set interrupt generation timing in channel fixed repeated conversion mode.(refer to Table 19-3)
3-2	−	R	Read as zero.
1	REPEAT	R/W	Sets repeat mode. 0 : Single conversion 1 : Repeat conversion
0	SCAN	R/W	Sets scan mode. 0 : Fixed channel mode 1 : Channel scan mode

Table 19-3 Interrupt generation timing in fixed channel mode

<ITM[2:0]>	Fixed channel repeat conversion mode <SCAN> = "0", <REPEAT> = "1"
000	Each time one conversion is completed.
001	Each time when conversion is completed twice.
010	Each time when conversion is completed three times.
011	Each time when conversion is completed four times.
100	Each time when conversion is completed five times.
101	Each time when conversion is completed six times.
110	Each time when conversion is completed seven times.
111	Each time when conversion is completed eight times.

Note 1: <ITM[2:0]> is valid only in fixed channel repeat mode (<REPEAT>=1,<SCAN>=0).

Note 2: To stop the conversion during repeat conversion (when <REPEAT>=1, fixed channel and channel scan), zero clear the <REPEAT> (<REPEAT>=0). Do not change any bits excepting the <REPEAT> bit.

19.3.7 ADAMOD4/ADBMOD4 (Mode Setting 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	SCANAREA				SCANSTA			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as zero.
7-4	SCANAREA [3:0]	R/W	Sets the range of channel scan (refer to Table 19-4 andTable 19-5.)
3-0	SCANSTA[3:0]	R/W	Sets the beginning channel of the channel scan (refer to Table 19-4 andTable 19-5).

The following setting sets the channel scan single mode: ADAMOD3/ADBMOD3<SCAN> = "1" and <REPEAT> = "0". Also, the following setting sets the channel scan repeat mode: ADAMOD3/ADBMOD3<SCAN> = "1" and <REPEAT> = "1". Then, select the channel for channel scan. For example, if you would like to set ADAMOD4<SCANSTA> = "0001"(AINA01), <SCANAREA>="0010"(3 channel scan), perform channel scan from AIN01 to AIN03 (for 3 channels).

Table 19-4 shows the <SCANSTA> setting in relation to the range of assignable value of <SCANAREA>.

Table 19-4 The range of assignable channel scan values (ADAMOD4)

<SCANSTA[3:0]>	Start channel	<SCANAREA[3:0]>	The range of assignable channel scan value
0000	(AINA0)	0000 to 0011	(1ch to 4ch)
0001	(AINA1)	0000 to 0010	(1ch to 3ch)
0010	(AINA2)	0000 to 0001	(1ch to 2ch)
0011	(AINA3)	0000	(1ch)

Note: Settings other than above is prohibited. Use assignable <SCANAREA>.

Table 19-5 The range of assignable channel scan values (ADBMOD4)

<SCANSTA[3:0]>	Start channel	<SCANAREA[3:0]>	The range of assignable channel scan
0000	(AINB0)	0000 to 0011	(1ch to 4ch)
0001	(AINB1)	0000 to 0010	(1ch to 3ch)
0010	(AINB2)	0000 to 0001	(1ch to 2ch)
0011	(AINB3)	0000	(1ch)

Note: Settings other than above is prohibited. Use assignable <SCANAREA>.

19.3.8 ADAMOD5/ADBMOD5 (Mode Setting Register 5)

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

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as zero.
3	HPEOCF	R	Top-priority AD conversion completion flag (note 1) 0: Before or during conversion 1: Completion
2	HPADBF	R	Top-priority AD conversion BUSY flag 0: Conversion stop 1: During conversion
1	EOCF	R	Normal AD conversion end flag (note 1) 0: Before or during conversion 1: Completion
0	ADBF	R	Normal AD conversion BUSY flag 0: Conversion stop 1: During conversion

Note 1: <EOCF> and <HPEOCF> zero cleared by reading them.

Note 2: To reduce consumption current used when shifting to IDLE mode with setting <I2AD>="0", or in STOP1/STOP2 mode, set "0" to <DACON> and <RCUT> after AD conversion, and then execute an instruction to move on to stand-by mode.

19.3.9 ADAMOD6/ADBMOD6 (Mode Setting 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	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	ADRST	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	W	Write as zero.
1-0	ADRST[1:0]	W	Overwriting 10 with 01 allows ADC to be software reset. All registers excepting <ADCLK> bit are initialized.

Note: To perform software, initialization takes 3μs.

19.3.10 ADACMPCR0/ADBCMPCR0 (Monitor 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	-	-	-	-	CMPCNT0			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP0EN	-	CMPCOND0	ADBIG0	AINS0			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	−	R	Read as zero.
11-8	CMPCNT0[3:0]	R/W	The number of counting is configured. 0000 : 1 count 0001 : 2 counts 0010 : 3 counts 0011 : 4 counts 0100 : 5 counts 0101 : 6 counts 0110 : 7 counts 0111 : 8 counts 1000 : 9 counts 1001 : 10 counts 1010 : 11 counts 1011 : 12 counts 1100 : 13 counts 1101 : 14 counts 1110 : 15 counts 1111 : 16 counts
7	CMP0EN	R/W	AD monitor function 0 0: Disable (the number of counting for judgement is cleared) 1: Enable (if condition is satisfied, an AD monitor interrupt INTADAM0/INTADBM0 is generated)
6	−	R	Read as zero.
5	CMPCOND0	R/W	Sets the condition for judgement count. 0: Serial 1: Cumulative Using serial method, an AD monitor interrupt occurs when the condition set to the <ADBIG0> continues and counts up to the number set to the <CMPCNT0>. After exceeding the setting value, an AD monitor interrupt occurs every time when the judgement condition is true. If the condition is different from the condition set to the <ADBIG0>, the counter is cleared. Using cumulative method, an AD monitor interrupt occurs and the counter is cleared when the condition set to the <ADBIG0> is accumulated and reaches the number set to the <CMPCNT0>. Even if the condition is different from the value set to the <ADBIG0>, the value of the counter is held.
4	ADBIG0	R/W	Sets judging large and small. 0: Larger than the value of the comparison register (ADACMP0/ADBCMP0) 1: Smaller than the value of the comparison register (ADACMP0/ADBCMP0) Sets the conversion results of the target analog input larger/smaller than the value of the comparison register. Every time when the AD conversion set to AINS0[3:0] ends, judges large and small. If the result of the judgement matches to the setting of the <ADBIG0>, the counter is counted up.
3-0	AINS0[3:0]	R/W	Sets analog input as the comparison target. ADACMPCR0 0000 : AINA0 0001 : AINA1 0010 : AINA2 0011 : AINA3 0100 to1111: prohibited ADBCMPCR0 0000 : AINB0 0001 : AINB1 0010 : AINB2 0011 : AINB3 0100 : Reserved 0101 : Reserved 0110 : Reserved 0111 : Reserved 1000 : Reserved 1001 : Reserved 1010 : Reserved 1011 : Reserved 1100 to 1111: prohibited

Note: AD monitor function is used in the fixed repeat conversion mode and the scan repeat mode.

19.3.11 ADACMPCR1/ADBCMPCR1 (Monitor 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	-	-	-	-	CMPCNT1			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP1EN	-	CMPCOND1	ADBIG1	AINS1			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	−	R	Read as zero.
11-8	CMPCNT1[3:0]	R/W	Sets judging large and small. 0000 : 1 count 0001 : 2 counts 0010 : 3 counts 0011 : 4 counts 0100 : 5 counts 0101 : 6 counts 0110 : 7 counts 0111 : 8 counts 1000 : 9 counts 1001 : 10 counts 1010 : 11 counts 1011 : 12 counts 1100 : 13 counts 1101 : 14 counts 1110 : 15 counts 1111 : 16 counts
7	CMP1EN	R/W	AD monitor function 1 0: Disable (the number of counting for judgement is cleared) 1: Enable (if condition is satisfied, an AD monitor interrupt INTADAM1/INTADBM1 is generated)
6	−	R	Read as zero.
5	CMPCOND1	R/W	Sets the condition for judgement count. 0: Serial 1: Cumulative Using serial method, an AD monitor interrupt occurs when the condition set to the <ADBIG1> continues and counts up to the number set to the <CMPCNT1>. After exceeding the setting value, an AD monitor interrupt occurs every time when the judgement condition is true. If the condition is different from the condition set to the <ADBIG1>, the counter is cleared. Using cumulative method, an AD monitor interrupt occurs and the counter is cleared when the condition set to the <ADBIG1> is accumulated and reaches the number set to the <CMPCNT1>. Even if the condition is different from the value set to the <ADBIG1>, the value of the counter is held.
4	ADBIG1	R/W	Sets judging large and small. 0: Larger than the value of the comparison register (ADACMP1/ADBCMP1). 1: Smaller than the value of the comparison register (ADACMP1/ADBCMP1). Sets the conversion results of the target analog input larger/smaller than the value of the comparison register. Every time when the AD conversion set to AINS1[3:0] ends, judges large and small. If the result of the judgement matches to the setting of the <ADBIG1>, the counter is counted up.
3-0	AINS1[3:0]	R/W	Sets analog input as the comparison target. ADACMPCR0 0000 : AINA0 0001 : AINA1 0010 : AINA2 0011 : AINA3 0100 to 1111: prohibited ADBCMPCR0 0000 : AINB0 0001 : AINB1 0010 : AINB2 0011 : AINB3 0100 : Reserved 0101 : Reserved 0110 : Reserved 0111 : Reserved 1000 : Reserved 1001 : Reserved 1010 : Reserved 1011 : Reserved 1100 to 1111: prohibited

Note: AD monitor function is used in the fixed repeat conversion mode and the scan repeat mode.

19.3.12 ADACMP0/ADBCMP0 (AD 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	-	-	-	-	AD0CMP			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AD0CMP							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	−	R	Read as zero.
11-0	AD0CMP[11:0]	R/W	Sets the comparison value of AD conversion.

Note: To write a value to this register or to change the settings, disable the AD monitor function first ÅiADACMPCR0/ADBCMPCR0<CMP0EN> = "0", ADACMPCR1/ADBCMPCR1<CMP1EN> = "0").

19.3.13 ADACMP1/ADBCMP1 (AD 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	-	-	-	-	AD1CMP			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AD1CMP							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	−	R	Read as zero.
11-0	AD1CMP[11:0]	R/W	Sets the comparison value of AD conversion.

Note: To write a value to this register or to change the settings, disable the AD monitor function first (ADACMPCR0/ADBCMPCR0<CMP0EN> = "0", ADACMPCR1/ADBCMPCR1<CMP1EN> = "0").

19.3.14 ADAREG00 to ADAREG07/ADBREG00 to ADBREG07 (AD Conversion Result Register)

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

Bit	Bit Symbol	Type	Function
31-20	ADR[11:0] _MIR	R	12-bit normal AD conversion results are stored. If you read the ADREGx register during AD conversion, the previous conversion result is read.
19-18	−	R	Read as zero.
17	ADOVRF_MIR	R	Overrun flag 0: Not generated 1: Generated If AD conversion result is over written before reading the AD conversion result register (ADREGx), this bit is set to "1". This flag is zero cleared when reading the ADREGx register.
16	ADRF_MIR	R	AD conversion result storage flag 0: Conversion result is not stored. 1: Conversion result is stored. If the conversion result is stored, this bit is set to "1". This flag is zero cleared when the ADREGx register is read.
15-14	−	R	Read as zero.
13	ADOVRF	R	Overrun flag 0: Not generated. 1: Generated. If the conversion result is overwritten before reading the AD conversion result register (ADREGx), this bit is set to "1". This flag is zero cleared when the ADREGx register is read.
12	ADRF	R	AD conversion result storage flag 0: Conversion result is NOT stored. 1: Conversion result is stored. If a conversion result is stored, this bit is set to "1". This flag is zero cleared when the ADREGx register is read.
11-0	ADR[11:0]	R	12-bit normal AD conversion result is stored. If you read the ADREGx register during AD conversion, the previous conversion result is read.

Note: Since <ADR_MIR>, <ADOVRF_MIR> and <ADRF_MIR> are read as same as <ADR>, <ADOVRF> and <ADRF>. Use one of them.

19.3.15 ADAREGSP/ADBREGSP (Top-priority AD Conversion Result Register)

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

Bit	Bit Symbol	Type	Function
31-20	ADSPR[11:0]_ MIR	R	12-bit top-priority AD conversion results are stored. If you read the ADAREGSP/ADBREGSP registers during AD conversion, the previous conversion result is read.
19-18	−	R	Read as zero.
17	ADOVRSPF_ MIR	R	Overrun flag 0: Not generated 1: Generated If AD conversion result is over written before reading the AD conversion result registers (ADAREGSP/ADBREGSP), this bit is set to "1". This flag is zero cleared when reading the ADREGSP register.
16	ADSPRF_MIR	R	Top-priority AD conversion result storage flag 0: Conversion result is NOT stored. 1: Conversion result is stored. If a result of top-priority conversion is stored, this bit is set to "1". This flag is zero cleared when reading the ADAREGSP/ADBREGSP registers.
15-14	−	R	Read as zero.
13	ADOVRSPF	R	Overrun flag 0: Not generated. 1: Generated. If the result of top-priority conversion is overwritten before reading the top-priority AD Conversion Result Register (ADAREGSP/ADBREGSP), this bit is set to "1". This flag is zero cleared when reading the ADREGSP register.
12	ADSPRF	R	Top-priority AD conversion storage flag 0: Conversion result is NOT stored. 1: Conversion result is stored. When a conversion result of top-priority AD conversion is stored, this bit is set to "1". This flag is zero cleared when reading the ADAREGSP/ADBREGSP registers.
11-0	ADSPR[11:0]	R	12-bit top-priority AD conversion result is stored. If you read the ADAREGSP/ADBREGSP registers during AD conversion, the previous conversion result is read.

Note: Since <ADSPR_MIR>, <ADOVRSPF_MIR> and <ADSPRF_MIR> are read as same as <ADSPR>, <ADOVRSPF> and <ADSPRF>. Use one of them.

19.3.16 ADILVMO1 (Control Register for Dual Unit Mode 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	SWATRG	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as zero.
7	SWATRG	W	Software trigger in dual unit mode 0: Writing a zero is ignored. 1: Starts a trigger output of ADCINTLV. Read as zero.
6-0	−	R	Read as zero.

Note 1: When you use <SWATRG>, enable the ADILVMO2<TRGAEN> and set the ADILVMO2<TRGASEL> to "software".

Note 2: Do not write to the register before the completion of output to the unit A or unit B corresponding to a trigger input.

19.3.17 ADILVMO2 (Control Register for Dual Unit Mode 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	ADILV	-	-	TRGASTA	TRGASEL			TRGAEN
After reset	0	0	0	0	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as zero.
7	ADILV	R/W	Selects control mode in dual unit mode. 0: Trigger start mode 1: Interleave mode
6-5	−	R	Read as zero.
4	TRGASTA	R	Indicates the destination unit of the next trigger output in trigger start mode. 0: Unit A 1: Unit B In other modes (interleave mode and trigger disabled mode), this bit is fixed to a zero.
3-1	TRGASEL[2:0]	R/W	Selects a trigger input in dual unit mode. 000 : Software 001 : External trigger 010 : INTTB21 (TB2RG1 of the timer 2 matches) 011 : INTTB31 (TB3RG1 of the timer 3 matches) 100 : INTTB41 (TB4RG1 of the timer 4 matches) 101 : INTTB51 (TB5RG1 of the timer 5 matches) 110 : INTTB61 (TB6RG1 of the timer 6 matches) 111 : INTTB71 (TB7RG1 of the timer 7 matches)
0	TRGAEN	R/W	Enable /disable the trigger function in dual unit mode. 0:Disable 1:Enable Set "1" in dual unit mode (trigger start mode and interleave mode). Set "0" if you use in single unit mode (trigger function disable mode)

Note 1: During the operation of ADCINTLV circuit, if modify <ADLIV> or clear <TRGAEN> to "0", the operation of ADCINTLV circuit is stopped.

Note 2: Do not modify the registers other than <ADLIV> and <TRGAEN> during the operation of ADCINTLV circuit.

19.3.18 ADILVMO3 (Control Register for Dual Unit Mode 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	CORCNT							
After reset	0	0	1	0	0	1	1	1

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as zero.
7-0	CORCNT[7:0]	R/W	Sets the trigger start adjusting counter. Sets an arbitrary value up to 0x01to0xFF. The initial value is 0x27.

Note 1: The actual delay time from the start of the unit A conversion to the start of the unit B conversion is calculated by multiplying 1/fc by (register setting value +1). Thus, the initial condition when fc=80MHz, 0.5μs=40 ÷ 80MHz (initial count value: 0x28).

Note 2: Do not input a trigger to unit B before a trigger signal is output to unit B in operating interleave mode.

19.4 Description of Operations

This chapter describes the unit operation in single unit mode and dual unit mode.

19.4.1 Usage note about the activation of analog conversion

To start AD conversion, write "1" to the ADAMOD1/ADBMOD1<DACON>, and wait for 3 μs until the internal circuit stabilizes. Then, write "1" to the ADAMOD0/ADBMOD0<ADS>. If you do not use this function, write "0" to the ADAMOD1/ADBMOD1<DACON>. The consumption current of the analog circuit is reduced.

19.4.2 AD conversion mode

Two types of AD conversion are supported: normal AD conversion and top-priority AD conversion.

19.4.2.1 Normal AD conversion

Normal AD conversion supports the following four operation types. Operation modes are selected by setting ADAMOD3/ADBMOD3<REPEAT>, <SCAN>.

- Fixed channel single conversion mode
- Channel scan single conversion mode
- Fixed channel repeat conversion mode
- Channel scan repeat conversion mode

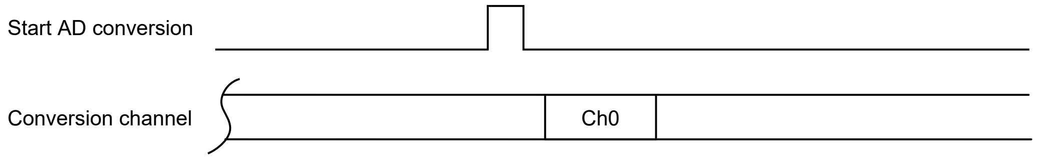
Only the fixed channel single conversion mode and fixed channel repeat conversion mode can be used in the dual unit mode.

(1) Fixed channel single conversion mode

If ADAMOD3/ADBMOD3<REPEAT>, <SCAN> is set to "00", AD conversion is performed in the fixed channel single conversion mode.

In this mode, AD conversion is performed once for one channel which is selected by ADAMOD2/ADBMOD2<ADCH>. After the AD conversion, ADAMOD5/ADBMOD5<EOCF> is set to "1", ADAMOD5/ADBMOD5<ADBF> is zero cleared, and an interrupt request of INTADA/INTADB is generated. <EOCF> is cleared to zero upon read.

The figure below shows an example of converting AINA0 in fixed channel single conversion mode.



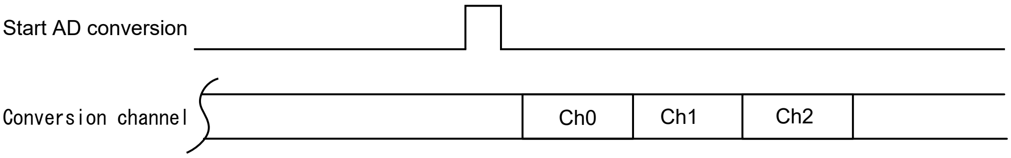
(2) Channel scan single conversion mode

If ADAMOD3/ADBMOD3<REPEAT>, <SCAN> is set to "01", AD conversion is performed in the channel scan single conversion mode.

In this mode, AD conversion is performed once for the scan channel area selected by ADMOD4/ADBMOD4<SCANAREA> from the start channel selected by ADAMOD4/ADBMOD4<SCAN-

STA>. After AD scan conversion, ADAMOD5/ADBMOD5<EOCF> is set to "1", ADAMOD5/ADBMOD5<ADBF> is zero cleared, and an interrupt request of INTADA/INTADB is generated. <EOCF> is zero cleared upon read.

The figure below shows an example of converting from AINA0 to AINA2 in channel scan single conversion mode.

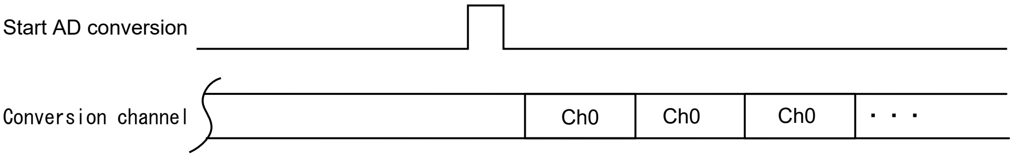


(3) Fixed channel repeat conversion mode

If ADAMOD3/ADBMOD3<REPEAT>, <SCAN> is set to "10", AD conversion is performed in fixed channel repeat conversion mode.

In this mode, AD conversion is repeated for the number of times set to ADAMOD3/ADBMOD3<ITM> (interrupt request generating timing for INTADA/INTADB can be selected) for the one channel selected by ADAMOD2/ADBMOD2<ADCH>. After the number of AD conversion set to <ITM> is repeated, ADAMOD5/ADBMOD5<EOCF> is set to "1", but ADAMOD5/ADBMOD5<ADBF> is not zero cleared and keeps "1". <EOCF> is zero cleared upon read.

The figure below is an conversion example of AINA0 in fixed channel repeat conversion mode.

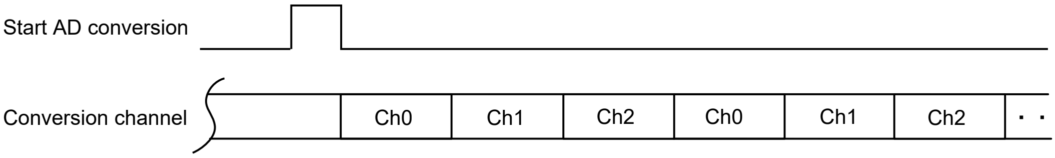


(4) Channel scan repeat conversion mode

If ADAMOD3/ADBMOD3<REPEAT>, <SCAN> is set to "11", AD conversion is performed in the channel scan repeat conversion mode.

In this mode, AD conversion is performed repeatedly for the scan channel area in the channel selected by ADAMOD4/ADBMOD4<SCANSTA> from the start channel selected by ADAMOD4/ADBMOD4<SCANAREA>. Each time one AD scan conversion is completed, ADAMOD5/ADBMOD5<EOCF> is set to "1" and the INTADA/INTADB interrupt request is generated. ADAMOD5/ADBMOD5<ADBF> is not zero cleared and remains at "1". <EOCF> is cleared to "0" upon read.

The figure below shows an operation example of converting AINA0 to AINA2 in the channel scan repeat conversion mode.



19.4.2.2 Top-priority AD conversion

Top-priority AD conversion can be performed by interrupting ongoing normal AD conversion. If the top-priority AD conversion interrupts into normal AD conversion, the normal AD conversion starts from the stopped channel after the top-priority conversion.

The fixed channel single conversion is the only operation mode. The settings to ADAMOD3/ADBMOD3<REPEAT>, <SCAN> are invalid. When conditions to start operation are met, a conversion is performed just once for a channel selected by ADAMOD2/ADBMOD2<HPADCH>. When conversion is completed, the top-priority AD conversion completion interrupt (INTADAHP/INTADBHP) is generated, ADAMOD5/ADBMOD5<HPEOCF> is set to "1" and <HPADBF> is zero cleared. <HPEOCF> returns to "0" upon read.

If a top-priority AD conversion is activated during the operation of another top-priority conversion, the previous conversion becomes invalid and the new operation becomes valid. The top-priority AD conversion cannot be used in the dual unit AD conversion mode.

19.4.3 AD monitor function

This function is used for configuring the fixed channel repeat mode and the scan repeat mode.

If ADACMPCR0/ADBCMPCR0 <CMP0EN> and ADACMPCR1/ADBCMPCR1<CMP1EN> are set to "1", AD monitor function is enabled. Two monitor function can be enabled concurrently.

Here is an example of ADACMPCR0/ADBCMPCR0 (same as ADACMPCR1/ADBCMPCR1).

Analog input for comparison is set to ADACMPCR0/ADBCMPCR0<AINS0[3:0]>. Large or small judgement is set to <ADBIG0>. Conditions of this comparison count is set to <CMPCOND0>. The number of comparison count is set to <CMPCNT0[3:0]>.

Once AD conversion starts, the AD converter checks comparison conditions (smaller/larger than values of the compare register) for each AD conversion. If the result of the comparison meets the settings of <ADBIG0>, the AD converter counts up the counter value.

There are two types of comparison condition: sequential method and cumulative method.

In the sequential method, an AD monitor interrupt (INTADAM0/INTADBM0) occurs if the condition set to <ADBIG0> continues and reaches the number of counts set to <CMPCNT0[3:0]>. An interrupt occurs without clearing the counter if the result of comparison meets the condition even after reaching the the number of counts set to <CMPCNT0[3:0]>. The value of the counter is zero cleared only when the condition does not meet the set condition of <ADBIG0>.

In the cumulative method, the counter is zero cleared when the total number of times that the condition set to <ADBIG0> meets reaches the number set to <CMPCNT0[3:0]>. An AD monitor interrupt (INTADAM0/INTADBM0) occurs at this time. The counter value is kept even if the result of the comparison is different from the set value of the counter. If values of the Conversion Result Register configured by the ADACMPCR0/ADBCMPCR0 register are the same as those of the tar get register, the AD converter does not count up. An AD conversion interrupt (INTADAM0/INTADBM0) does not occur either.

This comparison operation is performed each time a result is stored in a corresponding conversion result register. An interrupt (INTADAM0/INTADBM0) occurs when conditions meet (including counting). Since the conversion result register assigned to perform the AD monitor function is usually not read by software, the conversion result storage flag ADAREG/ADBREG<ADRF> and the overrun flag ADAREG/ADBREG<ADOVRF> remain being set. Do not use the conversion result register when you use the AD monitor function.

1. AINA0 input is set to the fixed channel repeat conversion mode and compare values of the AD Conversion Result Compare Register (0x0888).
    - ADAMOD3=0x0002: fixed channel repeat conversion

      AD conversion completion interrupt (INTADA) is disabled.
    - ADACMPCR0 =0x0280: compare target channel: AINA0, size determination: larger than a value of the compare register. Compare counting condition: sequential method. AD monitor function: enabled. Size determination count: 3 counts.
    - ADACMP0=0x0888: AD Conversion Result Compare Register (compared value 0x0888)

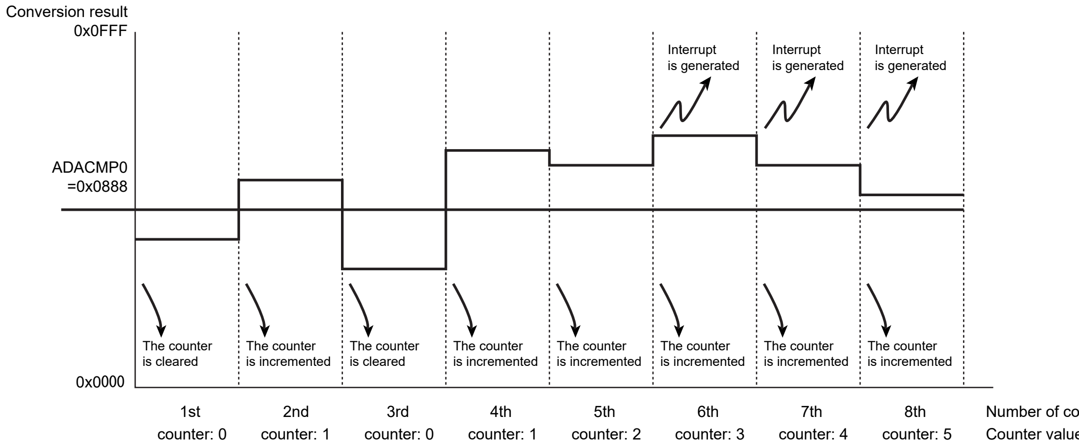

Conversion result
0x0FFF
ADACMP0
=0x0888
0x0000
Interrupt
is generated
Interrupt
is generated
Interrupt
is generated
The counter
is cleared
The counter
is incremented
The counter
is cleared
The counter
is incremented
The counter
is incremented
The counter
is incremented
The counter
is incremented
The counter
is incremented
1st
counter: 0
2nd
counter: 1
3rd
counter: 0
4th
counter: 1
5th
counter: 2
6th
counter: 3
7th
counter: 4
8th
counter: 5
Number of co
Counter value

Figure 19-5 AD monitor function (fixed channel repeat and sequential method)

2. AINA0 is set to fixed channel repeat conversion and compare the value of the AD Conversion Result Compare Register (0x0888).
   - ADAMOD3=0x0002: fixed channel repeat conversion

     AD conversion completion interrupt (INTADA) is disabled.
   - ADACMPCR0 =0x02A0: compare target channel: AINA0, size determination: larger than a value of the compare register. compare counting condition: cumulative method. AD monitor function: enabled. size determination count: 3 counts
   - ADACMP0=0x0888: AD Conversion Result Compare Register (compared value 0x0888)

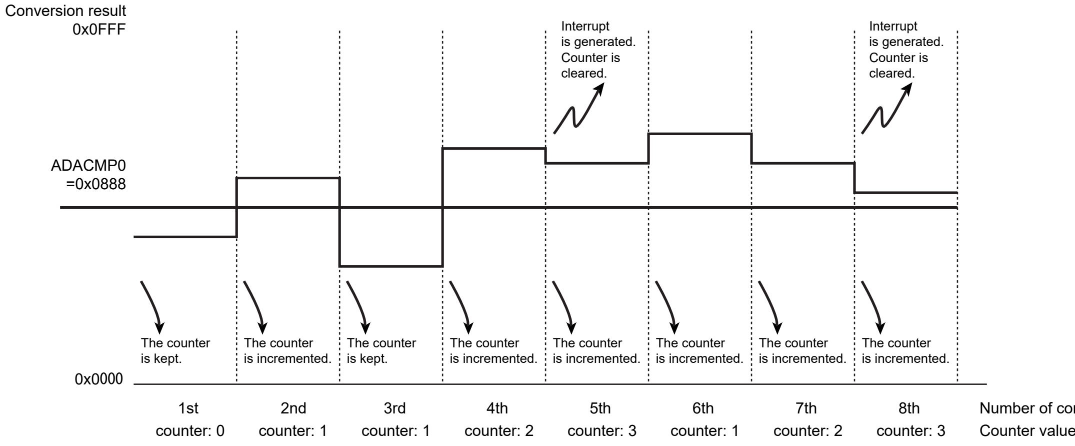

Conversion result
0x0FFF
ADACMP0
=0x0888
0x0000
Interrupt
is generated.
Counter is
cleared.
Interrupt
is generated.
Counter is
cleared.
The counter
is kept.
The counter
is incremented.
The counter
is kept.
The counter
is incremented.
The counter
is incremented.
The counter
is incremented.
The counter
is incremented.
The counter
is incremented.
1st
counter: 0
2nd
counter: 1
3rd
counter: 1
4th
counter: 2
5th
counter: 3
6th
counter: 1
7th
counter: 2
8th
counter: 3
Number of co
Counter value

Figure 19-6 AD Monitor Function (fixed channel repeat and cumulative method)

19.4.4 Selecting the input channel

After reset, ADAMOD3/ADBMOD3<REPEAT>, <SCAN> is initialized to "00" and ADAMOD2/ADBMOD2<ADCH[3:0]> is initialized to "0000".

The channels to be converted are selected according to the operation mode of the AD converter as shown below.

1. Normal AD conversion mode
   - If the analog input channel is used in a fixed state (ADAMOD3/ADBMOD3<SCAN> = "0")

     One channel is selected from analog input pins AIN0 through AINA3/AINB0 through AINB3 by setting ADAMOD2/ADBMOD2<ADCH> to an appropriate setting.
   - If the analog input channel is used in a scan state (ADAMOD3/ADBMOD3<SCAN> = "1")

     One scan mode is selected from the scan modes by setting ADAMOD4/ADBMOD4<SCANSTA> and ADAMOD4 /ADBMOD4<SCANAREA> to an appropriate setting.

2. Top-priority AD conversion mode

   One channel is selected from analog input pins from AIN0 through AIN3/AINB0 through AINB3 by setting ADAMOD2/ADBMOD2<HPADCH> to an appropriate setting. In this mode, if top-priority AD conversion has been activated during normal AD conversion, ongoing normal AD conversion is suspended, and restarts normal AD conversion after top-priority AD conversion is completed. The normal AD conversion restarts from the channel previously stopped after the top-priority AD conversion is completed.

19.4.5 Details of AD Conversion

19.4.5.1 Starting AD conversion

The normal AD conversion is enabled by setting "1" to ADAMOD0/ADBMOD0<ADS>. The top-priority AD conversion is enabled by setting "1" to ADAMOD0/ADBMOD0<HPADS>. Starting by setting values is called software start.

The normal AD conversion has four operation modes specified by ADAMOD3/ADBMOD3<REPEAT>, <SCAN>, and one of those mode is selected. The operation mode for the top-priority AD conversion is the fixed channel single conversion only.

Also, hardware activation can be selected. Set "0" to ADAMOD1/ADBMOD1<ADHWS> for the normal AD conversion and to ADAMOD1/ADBMOD1<HPADHWS> for the top-priority AD conversion. In single unit mode, use the external trigger pin ($\overline{\text{ADTRGA}}$) for activation. In dual unit mode, set ADILVMO2<TRGASEL> to select from software trigger, external trigger pin ($\overline{\text{ADTRGA}}$) and timer trigger activation.

To enable hardware activation, set "1" to ADAMOD1/ADBMOD1<ADHWE> for normal AD conversion and to ADAMOD1/ADBMOD1<HPADHWE> for top-priority AD conversion.

Even if hardware start is enabled, start with software can be done.

Note: When using external trigger is used as a hardware activating source for the top-priority AD conversion, external trigger cannot be selected for the normal AD conversion hardware start.

19.4.5.2 AD conversion

When normal AD conversion starts, the AD conversion Busy flag (ADAMOD5/ADBMOD5 <ADBF>) which indicates that AD conversion is under way is set to "1".

When top-priority AD conversion starts, the top-priority AD conversion Busy flag (ADAMOD5/ADBMOD5<HPADBF>) showing that AD conversion is underway is set to "1". At that time, the value of the Busy flag ADAMOD5/ADBMOD5<EOCF> and <ADBF> for normal AD conversion before the start of top-priority AD conversions are retained.

Note: Do not execute the normal AD conversion when the top-priority AD conversion is underway. A top-priority AD conversion completion flag is not set. Also, a previous normal AD conversion flag is not cleared.

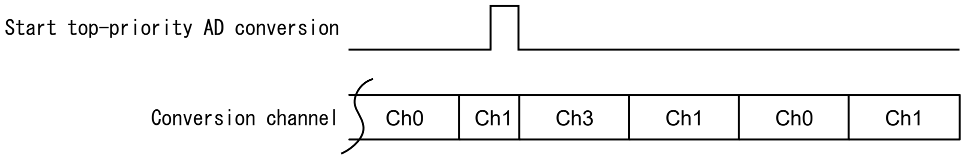
19.4.5.3 Top-priority AD conversion during normal AD conversion

If top-priority AD conversion has been activated during normal AD conversion, ongoing normal AD conversion is suspended, and restarts normal AD conversion after top-priority AD conversion is completed.

If ADAMOD0/ADBMOD0<HPADS> is set to "1" during normal AD conversion, ongoing normal AD conversion is suspended, and the top-priority AD conversion starts; specifically, AD conversion (fixed-channel single conversion) is executed for a channel designated by MOD2/ADBMOD2<HPADCH>. After the result of this top-priority AD conversion is stored in the storage register ADAREGSP/ADBREGSP, normal AD conversion is resumed.

If H/W activation of top-priority AD conversion is authorized during normal AD conversion, ongoing AD conversion is discontinued when requirements for activation using a H/W activation resource are met, and top-priority AD conversion (fixed-channel single conversion) starts for a channel designated by <HPADCH>. After the result of this top-priority AD conversion is stored in the storage register ADAREGSP/ADBREGSP, normal AD conversion is resumed.

For example, if channel repeat conversion is activated for channels AINB0 through AINB2 and if <HPADS> is set to "1" during AINB1 conversion, AIN1 conversion is suspended, and conversion is performed for a channel designated by <HPADCH> (AINB3 in the case shown below). After the result of conversion is stored in ADBREGSP, channel repeat conversion is resumed, starting from AINB1.



19.4.5.4 Stopping Repeat Conversion Mode

To stop the AD conversion operation in the repeat conversion mode (fixed-channel repeat conversion mode or channel scan repeat conversion mode), write "0" to ADAMOD3/ADBMOD3<REPEAT>. When ongoing AD conversion is completed, the repeat conversion mode terminates, and ADAMOD5/ADBMOD5<ADBF> is set to "0".

19.4.5.5 Reactivating normal AD conversion

Setting "1" to ADAMOD0/ADBMOD0<ADS> during normal AD conversion reactivates the normal AD conversion. The previous normal AD conversion is immediately discontinued when a new normal AD conversion is reactivated. At this time, the normal AD conversion busy flag ADAMOD5/ADBMOD5<ADBF>, the normal AD conversion completion flag ADAMOD5/ ADBMOD5<EOCF> and the AD conversion result storage flag ADAREG00 to 03/ADBREG00 to 03<ADOVRF> <ADRF> are zero cleared.

If H/W activation of top-priority AD conversion is authorized during normal AD conversion, ongoing AD conversion is reactivated when requirements for activation using a H/W activation resource are met. At this time, the previous normal AD conversion is immediately discontinued. <ADBF>, <EOCF>, <AOVRF> and <ADRF> are zero cleared.

19.4.5.6 Conversion completion

(1) Completing normal AD conversion

When normal AD conversion is completed, the AD conversion completion interrupt (INTADA/INTADB) is generated. The result of AD conversion is stored in the storage register, and values of two registers change: the register ADMOD0<EOCFN> which indicates the completion of AD conversion and the register ADMOD0<ADBFN> which indicates conversion is ongoin.Interrupt requests, conversion register storage register and <EOCFN><ADBFN> change with a different timing according to a mode selected.

In mode other than fixed-channel repeat conversion mode, conversion results are stored in the AD conversion result registers (ADAREG00 to 03/ADBREG00 to 03) corresponding to a channel.

In fixed-channel repeat conversion mode, the conversion results are sequentially stored in storage registers ADAREG00/ADBREG00 through ADAREG07/ADBREG07, according to the interrupt condition set to ADAMOD3/ADBMOD3<ITM>.

Interrupt requests, flag changes and conversion result registers in each mode are as shown below.

- Fixed-channel single conversion mode

  After AD conversion completed, <EOCF> is set to "1", <ADBF> is cleared to "0", and the interrupt request is generated.

  Conversion results are stored a conversion result register correspond to a channel.

- Channel scan single conversion mode

  After the channel scan conversion is completed, <EOCF> is set to "1", <ADBF> is set to "0", and an interrupt request is generated.

  Conversion results are stored a conversion result register correspond to a channel.

- Fixed-channel repeat conversion mode

  The <ADBF> is not cleared to "0". It remains at "1". The timing with which the interrupt request INTAD is generated can be selected by setting <ITM> to an appropriate setting. ADAREG04-ADAREG07 can be used only in the fixed channel repeat conversion mode.

  a. One-time conversion

With <ITM> set to "00", an interrupt request is generated each time one AD conversion set to <ADCH> is completed. In this case, the conversion results are always stored in the storage register s ADAREG00/ADBREG00 in sequential order. After the conversion result is stored, <EOCF> changes to "1".

b. Eight-time conversions

With <ITM> set to "111", an interrupt request is generated each time eight AD conversions set to <ADCH> are completed. In this case, the conversion results are always stored in the storage registers ADAREG00/ADBREG00 through ADAREG07/ ADBREG07 in sequential order. After the conversion results are stored in ADAREG07/ ADBREG07, <EOCF> is set to "1", and storage of subsequent conversion results from ADAREG00/ADBREG00.

- Fixed-channel repeat conversion mode

With <EOCF> set to "1", a INTADA/INTADB interrupt request is generated each time one scan conversion is completed. <ADBF> is not cleared to zero and remains at "1".

With ADAMOD4/ADBMOD4 <SCANAREA> set to "0011" (4-channel scan), four channel scans are performed from the Start Channel designated by ADAMOD4/ADBMOD4<SCANSTA>. Each time when a conversion of the final channel is completed, <EOCF> is set to "1", an interrupt request is generated, and four channel scanning starts from the Start Channel again. Since this mode is a repeat mode, <ADBF> is not zero cleared and maintains "1".

Conversion results are stored in a conversion result register corresponding to the channel.

(2) Completing top-priority AD conversion

After the top-priority AD conversion is completed, the top-priority AD conversion completion interrupt (INTADAHP/INTADBHP) is generated. ADAMOD5/ADBMOD5<HPEOCF> which indicates the completion of top-priority AD conversion is set to "1".

Conversion results are stored in the conversion result register ADAREGSP/ADBREGSP.

(3) Data polling

To confirm the completion of AD conversion without using interrupts, data polling can be used. When AD conversion is completed, ADAMOD5/ADBMOD5<EOCF> are set to "1". To confirm the completion of AD conversion and to obtain the results, poll this bit.

AD conversion result storage register must be read by word access. If <ADOVRF> = "0" and <ADRF> = "1" in ADAREG00 to ADAREG07 / ADBREG00 to ADBREG07, a correct conversion result has been successfully obtained.

A top-priority AD conversion can use data polling, too.

19.4.5.7 Interrupt Timing and Conversion Result Register

Table 19-6 shows the correlation between AD conversion modes, interrupt timing and flags. Table 19-7, Table 19-8 and Table 19-9 show the correlation between analog input channels and conversion result registers.

Table 19-6 Correlation between AD conversion mode, interrupt timing and flag operation

Conversion mode		Scan / repeat mode setting (ADAMOD3/ADBMOD3)			Interrupt generation timing	Conversion Status Flag (ADAMOD5/ADBMOD5)		
		<REPEAT>	<SCAN>	<ITM[2:0]>		<EOCF>/<HPEOCF> set timing (note 1)	<ADBF> (after interrupt)	<HPADBF> (after interrupt)
Normal conversion	Fixed channel single conversion	0	0	–	After conversion	After conversion	0	–
	Fixed channel repeat conversion	1	0	000	Each 1 conversion	After 1 conversion is completed	1	–
				001	Each 2 conversions	After 2 conversions are completed	1	–
				010	Each 3 conversions	After 3 conversions are completed	1	–
				011	Each 4 conversions	After 4 conversions are completed	1	–
				100	Each 5 conversions	After 5 conversions are completed	1	–
				101	Each 6 conversions	After 6 conversions are completed	1	–
				110	Each 7 conversions	After 7 conversions are completed	1	–
				111	Each 8 conversions	After 8 conversions are completed„	1	–
	Channel scan single conversion	0	1	–	After scan conversion is completed	After scan conversion is completed	0	–
	Channel scan repeat conversion	1	1	–	One scan conversion is completed.	One scan conversion is completed.	1	–
Top-priority conversion		–	–	–	After completion of conversion	After completion of conversion	–	0

Note 1: ADAMOD5/ADBMOD5<EOCF><HPEOCF> are cleared to zero upon read.

Note 2: In repeat mode, ADAMOD5/ADBMOD5<ADBF> are not zero cleared even if the interrupt is generated. To stop repeat mode, write zero to ADAMOD3/ADBMOD3<REPEAT>, then <ADBF> is zero cleared when the AD conversion is completed.

Table 19-7 Analog input channels and AD conversion result registers (Fixed channel single mode)

Fixed channel single mode		
Channel		Storage registers
Unit A	AINA0	ADAREG00
	AINA1	ADAREG01
	AINA2	ADAREG02
	AINA3	ADAREG03
Unit B	AINB0	ADBREG00
	AINB1	ADBREG01
	AINB2	ADBREG02
	AINB3	ADBREG03

Table 19-8 Analog input channels and AD conversion result registers (Fixed channel repeat mode)

Fixed channel repeat mode			
ADAMOD3/ADBMOD3<ITM[2:0]>			Storage register
Unit A	000	An interrupt occurs each time.	ADAREG00
	001	An interrupt occurs every twice.	ADAREG00 to ADAREG01
	010	An interrupt occurs every 3 times.	ADAREG00 to ADAREG02
	011	An interrupt occurs every 4 times.	ADAREG00 to ADAREG03
	100	An interrupt occurs every 5 times.	ADAREG00 to ADAREG04
	101	An interrupt occurs every 6 times.	ADAREG00 to ADAREG05
	110	An interrupt occurs every 7 times.	ADAREG00 to ADAREG06
	111	An interrupt occurs every 8 times.	ADAREG00 to ADAREG07
Unit B	000	An interrupt occurs each time.	ADBREG00
	001	An interrupt occurs every twice.	ADBREG00 to ADBREG01
	010	An interrupt occurs every 3 times.	ADBREG00 to ADBREG02
	011	An interrupt occurs every 4 times.	ADBREG00 to ADBREG03
	100	An interrupt occurs every 5 times.	ADBREG00 to ADBREG04
	101	An interrupt occurs every 6 times.	ADBREG00 to ADBREG05
	110	An interrupt occurs every 7 times.	ADBREG00 to ADBREG06
	111	An interrupt occurs every 8 times.	ADBREG00 to ADBREG07

Table 19-9 Analog input channels and AD conversion result registers (Channel scan single mode / repeat mode)

Channel scan single mode / repeat mode (Example: ADBREG03 to arbitrary width of scan channel)					
Unit	<SCANSTA> (Start Channel)		<SCANAREA> (Scan channel width)		Storage registers
Unit A	0000	AINA0	0000 to 0011	(1ch to 4ch)	ADBREG00 to ADBREG03
	0001	AINA1	0000 to 0010	(1ch to 3ch)	ADBREG01 to ADBREG03
	0010	AINA2	0000 to 0001	(1ch to 2ch)	ADBREG02 to ADBREG03
	0011	AINA3	0000	(1ch)	ADBREG03
Unit B	0000	AINB0	0000 to 0011	(1ch to 4ch)	ADBREG00 to ADBREG03
	0001	AINB1	0000 to 0010	(1ch to 3ch)	ADBREG01 to ADBREG03
	0010	AINB2	0000 to 0001	(1ch to 2ch)	ADBREG02 to ADBREG03
	0011	AINB3	0000	(1ch)	ADBRE03

19.4.6 Dual Unit Mode

19.4.6.1 Outline of Dual Unit Mode

Dual unit mode is controlled by ADCINTLV which generates two trigger signal for AD conversion start (unit A and unit B). AD conversions set to each unit are performed at the timing of trigger signals. Dual unit mode has two operation modes.

- Interleave mode
- Trigger start mode

Trigger source can be selected from software trigger, external trigger input and timer trigger input.

In interleave mode, a trigger signal generated based on a selected trigger input is output to the unit A. After that, a trigger signal for the unit B is generated after the elapse of a specified period of time.

In trigger start mode, a trigger signal is output to the unit A and the unit B in alternate shifts each time when a selected trigger become active.

19.4.6.2 Interleave Mode

To perform data polling in interleave mode, set ADILVMO2<ADILV><TRGAEN> to "1". Trigger source can be selected from software trigger, timer trigger and external trigger on the ADILVMO2<TRGASEL>.

If software trigger is selected as trigger source, generated pulses can be used as a trigger source (trigger output for the unit A) by setting "1" to ADILVMO1<SWATRG>. Simultaneously, the counter starts operation and a trigger for the unit B is output when the counter value becomes the set value of the ADILVMO3<CORCNT>.

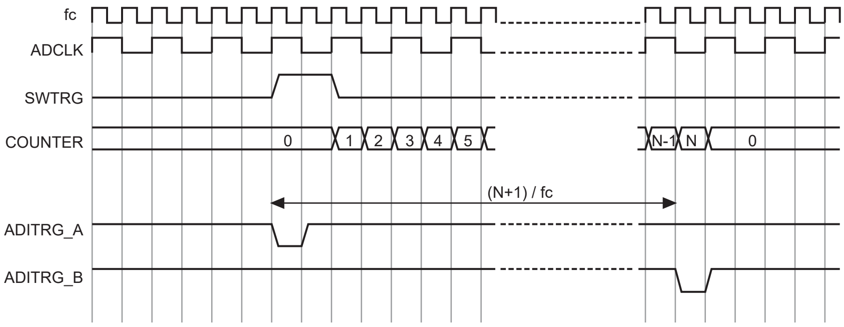


Figure 19-7 Interleave operation using a software trigger (ADCLK=fc/2)

If a timer trigger is selected as trigger source, a pulse generated by a timer trigger set to ADILVMO2<TRGASEL> becomes a trigger source, the first trigger output for the unit A. Simultaneously, the counter starts operation and a trigger for the unit B is output when the counter value becomes the set value of the ADILVMO3<CORCNT>.

If an external trigger is selected as a trigger source, th external trigger is a trigger source and a trigger output for the unitA. Simultaneously, the counter starts operation and a trigger for the unit B is output when the counter value becomes the set value of the ADILVMO3<CORCNT>.

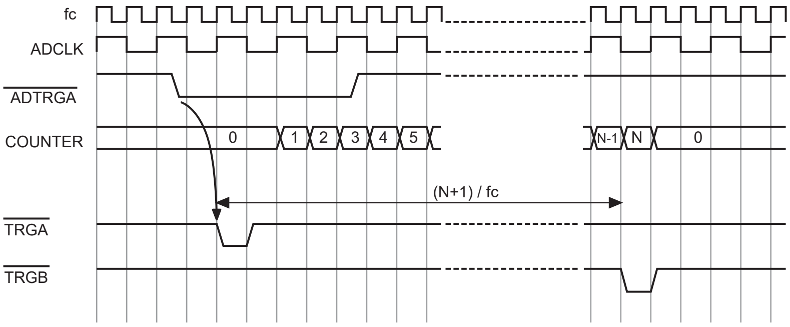


Figure 19-8 Interleave operation using an external trigger (ADCLK=fc/2)

19.4.6.3 Discontinuance and Resume of Interleave mode

After a trigger output for the unit B in interleave mode, conversion results of the unit B is stored and the operation stops if the below operation is performed before a trigger signal for the unit A is output. Also, after a trigger output for the unit A is output, the unit B does not start conversion and stops if the below operations are performed before a trigger signal for the unit B is output.

- ADILVMO2<ADILV> is set to "0".
- ADILVMO2<TRGAEN> is set to "0".

Interleave mode restarts if "1" is set to both registers above and a signal selected as a trigger source becomes active.

19.4.6.4 Trigger Start Mode

To start operation in trigger start mode, set ADILVMO2<ADILV> to "0", <TRGAEN> to "1". Trigger source can be selected from software trigger, timer trigger and external trigger by setting ADILVMO2<TRGASEL>.

If the software trigger is selected as a trigger source, generated pulse is used as trigger source by setting ADILVMO1<SWATRG> to "1". Trigger signals for the unit A and the unit B are output in alternate shift every time when this pulse is generated.

If the timer trigger is selected as a trigger source, the pulse generated by the timer trigger specified by the ADILVMO2<TRGASEL> becomes trigger source, and starts conversion at the falling edges. Trigger signals for the unit A and the unit B are output in alternate shift every time when a timer trigger is generated.

If the external trigger is selected as a trigger source, Trigger signals for the unit A and the unit B are output in alternate shift every time when an external trigger is input.

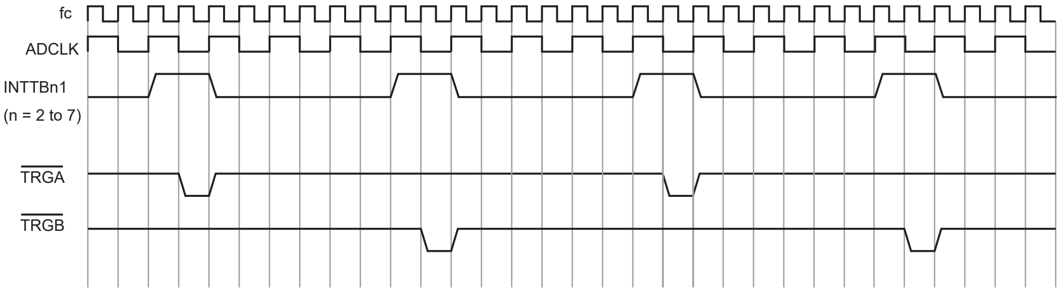


Figure 19-9 Trigger Start Operation by Timer Trigger (ADCLK = fc/2)

19.4.6.5 Notes on stopping and performing dual unit mode

Set "0" to ADILVMO2<TRGAEN> to stop the dual unit mode.

Also, do not write to the register during ongoing operation, without setting the above condition.

When interleave mode is active, do not set the next trigger source to output before ADCINTLV outputs a trigger signal for the unit B. Also, input of the next trigger source before the conversion of the unit A is prohibited. Ongoing operation stops as is the case in the single unit mode.

During an operation in trigger start mode, do not input a trigger source in odd number of time before the conversion for the unit A is completed. Do not input a trigger source in even number of time before the conversion for the unit B is completed, either. Ongoing operation stops as is the case in the single unit mode.

If you use the fixed channel repeat mode, settings of interrupt generation timing ADAMOD3/ADBMOD3<ITM> and the overrun flag ADAREG00-07/ADBREG00-07<ADOVRF> require attention to avoid an overrun error during the process of reading conversion results.

Notes on designing for AD converter inputs

<An output impedance of the external signal source which is connected with AIN pin>

An output impedance of the external signal source which is connected with AIN pin is equal or less than R_{EXAIN} shown below formula.

- Calculating formula of allowable value of output impedance of the external signal source -

The maxmum value of an output impedance connected with AIN pin : $R_{EXAIN} < Tscyc \div (ADCLK \times C_{ADC} \times \ln(2^{14})) - R_{AIN}$

MCU information	Symbol	Min	Typ	Max	Unit
ADC clock frequency	ADCLK	4	-	40	MHz
Total AIN input capacity in MCU	C_{ADC}	-	-	12.2	pF
AIN resistance in MCU	R_{AIN}	-	-	1	kΩ
Cycle number in the sample hold period	Tscyc	10	-	320	Cycle

R_{EXAIN} maximum value list (ADCLK = 40MHz)

Tcsyc	R_{EXAIN}	Unit
10	1.1	kΩ
20	3.2	kΩ
30	5.3	kΩ
40	7.5	kΩ
80	15.9	kΩ
160	32.8	kΩ
320	66.6	kΩ

< Addition of stabilizing capacity >

If high-speed AD conversion is required and the sample hold period cannot meet the conditions of calculating formula of allowable values of output impedance of external signal source, add stabilizing capacity to the AIN pin. The additional capacity depends on external circuitd. Although the capacity depended on the external circuit is different from the each board set, add the capacity from about 0.1μF to 1μF, appropriate amount for your circuit board.

Set the capacity to be added next to the AIN pin.

< Adjustment of sample hold period>

Generally, by setting the sample hold period long, you can make the input voltage of the comparator in the ADC circuit as same as the input voltage of the AIN pin can reduce the error of an AD conversion.

Although, in case that the sample hold period is too long, the error of an AD conversion may be increased because the voltage held in sample hold circuit is changed.

Because the suitable sample hold period is depended on the each board set, please decided the suitable sample hold period on your board set.

Notes of the use of the AD converter

The result value of AD conversion may vary depending on the fluctuation of the supply voltage, or may be affected by noise.

When using analog input pins and ports alternately, do not read and write ports during conversion because the conversion accuracy may be reduced. Also the conversion accuracy may be reduced if the output ports current fluctuate during AD conversion.

Please take counteractive measures with the program such as averaging the AD conversion results.

20. Digital Analog Converter (DAC)

20.1 Outline

The TMPM367FDFG has two channels of 10-bit digital-analog converter.

- 2-channel synchronous output
- Trigger function
- Waveform generation function
- Power-down function

20.2 Block Diagram

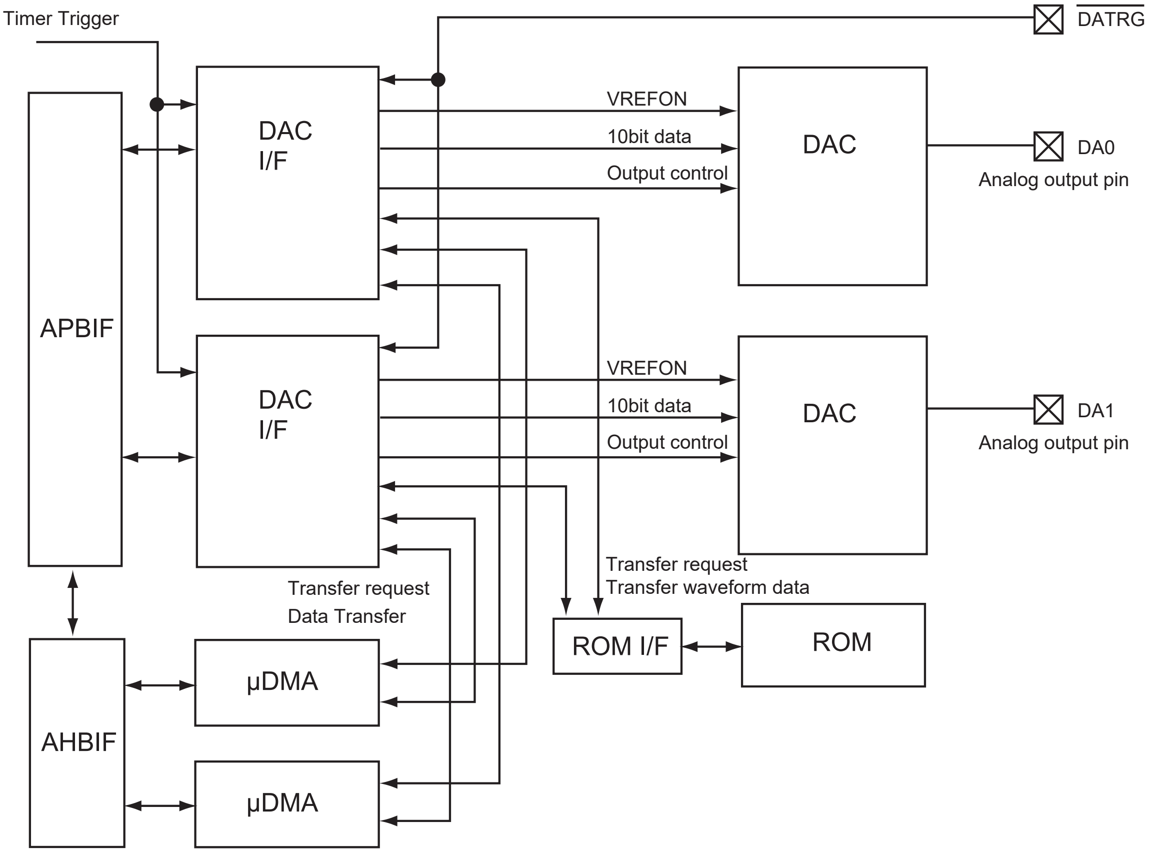

Timer Trigger
DATRG
APBIF
DAC I/F
VREFON
10bit data
Output control
DAC
DA0
Analog output pin
DAC I/F
VREFON
10bit data
Output control
DAC
DA1
Analog output pin
Transfer request
Data Transfer
Transfer request
Transfer waveform data
ROM I/F
ROM
AHBIF
μDMA
μDMA

Figure 20-1 Block diagram of DAC

20.3 Registers

20.3.1 List of Registers

DA convertor is controlled by DACCNTx and DACVCTLx. It outputs an analog voltage depend on DACREGx.

When wave generation function is used, DACDCTLx is set the condition and operation mode.

DACTCTLx initialize DA convertor (<DACCLR>) and controls the software output control (SWTRG>).

Channel x	Base Address
Channel0	0x4005_4000
Channel1	0x4005_5000

Register name (x=0 to 1)		Address (Base+)
DAC control register	DACCNTx	0x0000
DAC data register	DACREGx	0x0004
Waveform output control register	DACDCTLx	0x0008
Waveform trigger control register	DACTCTLx	0x000C
VOUTHOLD adjustment register	DACVCTLx	0x0010

20.3.2 DACCNTx (DAC 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	-	-	-	-	-	-	VREFON	OP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	VREFON	R/W	VREF control 0: VREF off 1: VREF on When this bit is set to "1", AVREFH is connected to the DAC circuit.
0	OP	R/W	DAC operation 0: Stop 1: Operation Controls DAC operation. When this bit is set to "1", a voltage which is set with DACREGx register is output to analog output pins. When this bit is set to "0", the operation is stopped and outputs becomes Hi-Z.

20.3.3 DACREGx (DAC 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	DAC							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	DAC		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-6	DAC[9:0]	R/W	Variable value setting Sets analog voltage outputs. Output voltage are calculated with a formula in below: Output voltage = (AVDD3_DA - AVSS_DA) × <DAC> / 1024 (Note) When output current is equal or less than 1mA, output voltage is equal or less than 0.2V in minimum and is equal or more than AVDD3_DA-0.2V in maximum.
5-0	−	R	Read as "0".

20.3.4 DACDCTLx (wave output control register)

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

Bit	Bit Symbol	Type	Function
31-21	−	R	Read as "0".
20-18	OFFSET[2:0]	R/W	Offset setting of output waveform 000:0VDD 001:1/8VDD 010:2/8VDD 011:3/8VDD 100:4/8VDD 101:5/8VDD 110:6/8VDD 111:reserved
17-16	AMPSEL[1:0]	R/W	Amplitude setting of output waveform 00:1/1VDD 01:1/2VDD 10:1/4VDD 11:reserved
15-12	−	R	Read as "0".
11-9	TRGSEL[2:0]	R/W	Trigger selection 000: Software 001: $\overline{\text{DATRG}}$ (External trigger inputs) 010:INTTB21 (TMRB2 : match with TB2RG1) 011:INTTB31 (TMRB3 : match with TB3RG1) 100:INTTB41 (TMRB4 : match with TB4RG1) 101:INTTB51 (TMRB5 : match with TB5RG1) 110:INTTB61 (TMRB6 : match with TB6RG1) 111:INTTB71 (TMRB7 : match with TB7RG1)
8	TRGEN	R/W	Trigger function 0: Disabled 1: Enabled
7	DMAEN	R/W	DMA enable 0: Disabled 1: Enabled
6-2	-	R	Read as "0".
1-0	WAVE[1:0]	R/W	Output waveform selection 00: No waveform generation 01: Triangle wave 10: Noise waveform 11: Sine wave

20.3.5 DACTCTLx (waveform trigger 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	DACCLR	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	SWTRG
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	W	Write as "0".
15	DACCLR	W	Initializing control of DAC 0:- 1:Clear requests Clears output buffer and initialize the waveform generation circuit
14-1	−	W	Write as "0".
0	SWTRG	W	Software output control 0:- 1: Output While <TRGSEL>=000 is set, if <SWTRG>=1 is set, data transfer is executed. (Note) Writable only when<TRGEN>=1 is set.

20.3.6 DACVCTLx (VOUTHOLD adjustment 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	VHOLDCTB				VHOLDCTF			
After reset	1	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	VHOLDCTB	R/W	Post adjustment register for VOUTHOLD time 0000: Prohibited 0001 to 1111: Adjustment value (0x1 to 0xF)
3-0	VHOLDCTF	R/W	Pre adjustment register for VOUTHOLD time 0000: Prohibited 0001 to 1111: Adjustment value (0x1 to 0xF)

20.4 Operational Description

20.4.1 Trigger Function

When the digital-analog converter is used, analog output values are required to update using the trigger function. To use the trigger function, set a trigger source to DACDCTLx<TRGSEL> then set "1" to <TRGEN>. When triggers are generated, analog output values are updated according to a value of DAREGx<DAC>. The trigger sources can be chosen among a 16-bit timer, an external trigger pin and software.

20.4.2 µDMA controller interlocking function

When DACDCTLx<DMAEN>="1" is set, data transfer to DACREGx<DAC> is enabled using µDMA controller. Waveform output interlocking with trigger function is enabled as well. How to use µDMA controller is described in Chapter "µDMA controller".

The following procedure is an example how to output an arbitrary waveform using external trigger and µDMA controller.

1. Sets µDMA controller
2. Sets DACCNTx<VREFON>="1" and <OP>="1"
3. Sets DACTCTLx<DACCLR>="1"
4. Sets VOUTHOLD time to DACVCTLx<VHOLDCTB>, <VHOLDCTF>
5. Sets DACDCTLx<OFFSET>="000", <AMPSEL>="00", <TRGSEL>="001", <TRGEN>="1", <DMAEN>="1" and <WAVE>="00"

20.4.3 Waveform Generation Function

The waveform generation function can be output sine waves, triangle waves and noise waveforms by setting output waveform to DACDCTLx<WAVE>. When sine waves or triangle waves are set, amplitude of output value and offset value can be set toDACDCTLx<AMPSEL> and <OFFSET> respectively.

Once the trigger which is set to DACDCTLx<TRGSEL> occurs, data in ROM table is read via ROM interface to output waveforms. Data setting is executed per one trigger. To output one cycle waveform, N time data setting is required.

Note: A value of N varies according to output waveforms.

Table 20-1 The value of <OFFSET> and <AMPSEL> when sine waves and triangle waves are output

<AMPSEL>	<OFFSET>	Output level (min.)	Output level (max)
00	000	AVDD3_DA × 0/8 (note)	AVDD3_DA × 8/8 (note)
01 (When the amplitude is 50%)	000	AVDD3_DA × 0/8 (note)	AVDD3_DA × 4/8
	001	AVDD3_DA × 1/8	AVDD3_DA × 5/8
	010	AVDD3_DA × 2/8	AVDD3_DA × 6/8
	011	AVDD3_DA × 3/8	AVDD3_DA × 7/8
	100	AVDD3_DA × 4/8	AVDD3_DA × 8/8 (note)
10 (When the amplitude is 25%)	000	AVDD3_DA × 0/8 (note)	AVDD3_DA × 2/8
	001	AVDD3_DA × 1/8	AVDD3_DA × 3/8
	010	AVDD3_DA × 2/8	AVDD3_DA × 4/8
	011	AVDD3_DA × 3/8	AVDD3_DA × 5/8
	100	AVDD3_DA × 4/8	AVDD3_DA × 6/8
	101	AVDD3_DA × 5/8	AVDD3_DA × 7/8
	110	AVDD3_DA × 6/8	AVDD3_DA × 8/8 (note)

Note: About the output level near the full scale or zero point, refer to the 10-bit D/A Convertor Electrical characteristics in Electrical characteristics.

20.4.3.1 Sine Wave Output

When sine waves are output, set DACDCTLx<WAVE>="11". A waveform cycle is determined by the trigger generation interval. To output one cycle waveform, 2048-time data read is required. The cycle is calculated as follows; the trigger generation interval × 2048.

The following shows an example that the 50% amplitude triggered by 16-bit timer and sine waves with 1/8 offset are configured.

1. Sets DACCNTx<VREFON>="1", <OP>="1"
2. Sets DACTCTLx<DACCLR>="1"
3. Sets VOUTHOLD time to DACVCTLx<VHOLDCTB>, <VHOLDCTF>
4. Sets DACDCTLx<OFFSET>="001", <AMPSEL>="01", <TRGSEL>="010", <TRGEN>="1", <DMAEN>="0" and <WAVE>="11"
5. Sets the 16-bit timer interrupt

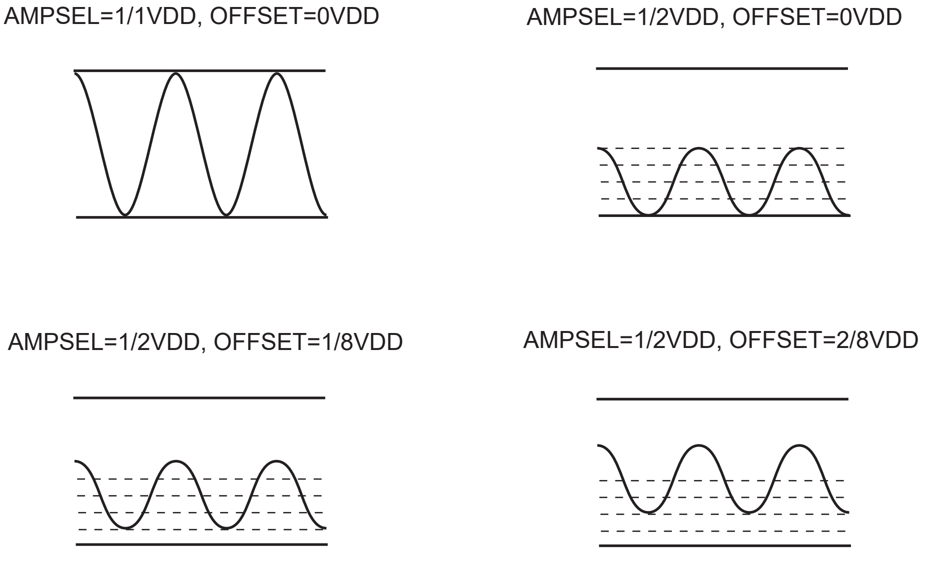

AMPSEL=1/1VDD, OFFSET=0VDD
AMPSEL=1/2VDD, OFFSET=0VDD
AMPSEL=1/2VDD, OFFSET=1/8VDD
AMPSEL=1/2VDD, OFFSET=2/8VDD

Figure 20-2 Sine wave output

20.4.3.2 Triangle Wave Output

When triangle waveforms are output, set DACDCTLx<WAVE>="01". A waveform cycle is determined by the set amplitude and the trigger generation intervals. The cycle is calculated as follows:

The following shows an example that the 50% amplitude triggered by 16-bit timer and the triangle waves with 1/8 offset are configured.

1. Sets DACCNTx<VREFON>="1", <OP>="1"
2. Sets DACTCTLx<DACCLR>="1"
3. Sets VOUTHOLD time to DACVCTLx<VHOLDCTB>, <VHOLDCTF>
4. Sets DACDCTLx<OFFSET>="001", <AMPSEL>="01", <TRGSEL>="010", <TRGEN>="1", <DMAEN>="0" and <WAVE>="01"
5. Sets the 16-bit timer interrupt

Table 20-2 Relationship between amplitude of triangle wave and waveform cycles

Amplitude of triangle wave	Number of data reads per one cycle	Waveform cycle
00	2046	Interrupt interval × 2046
01 (When the amplitude is 50%)	1022	Interrupt interval × 1022
10 (When the amplitude is 25%)	510	Interrupt interval × 510

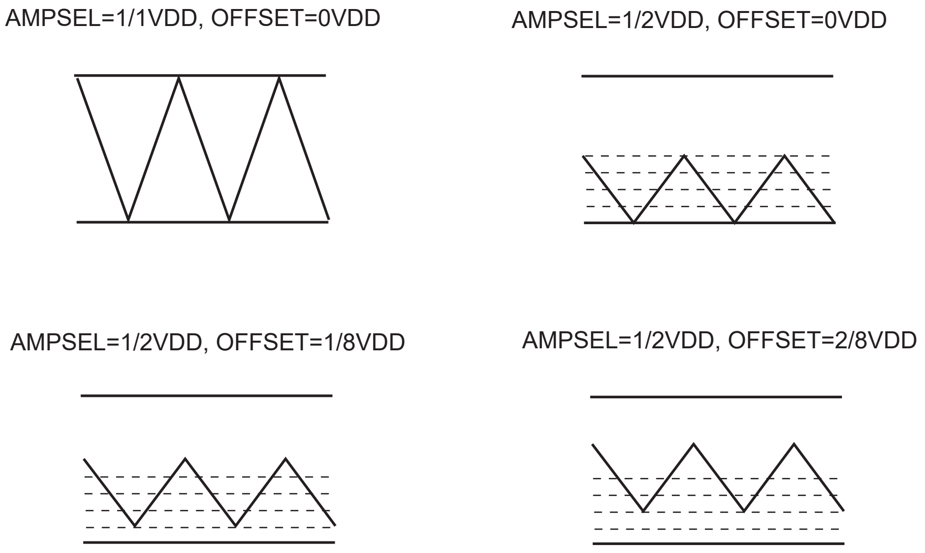

AMPSEL=1/1VDD, OFFSET=0VDD
AMPSEL=1/2VDD, OFFSET=0VDD
AMPSEL=1/2VDD, OFFSET=1/8VDD
AMPSEL=1/2VDD, OFFSET=2/8VDD

Figure 20-3 Triangle wave output

20.4.3.3 Noise Waveform Output

When "10" is set to DACDCTLx<WAVE>, white noise is output. A waveform cycle is determined by the trigger interval. To output one cycle waveform, 1023-time data read is required. The cycle is calculated as follows; the trigger generation interval ×1023.

The following shows an example that noise waveforms is output triggered by software trigger.

1. Sets DACCNTx<VREFON>="1", <OP>="1"
2. Set DACTCTLx<DACCLR>="1"
3. Sets VOUTHOLD time to DACVCTLx<VHOLDCTB>, <VHOLDCTF>
4. Sets DACDCTLx<OFFSET>="00", <AMPSEL>="00", <TRGSEL>="000", <TRGEN>="1", <DMAEN>="0" and <WAVE>="10"
5. Sets DACTCTLx<SWTRG>="1" periodically

20.4.4 Low Consumption Mode

When DACCNTx<VREFON>="0" is set, VREF is cut. This causes that VREF are becomes same voltage as AVREFL, so that the consumption current will be reduced. When DACCTLx<OP>="0" is set, the DAC operation is stopped and analog output pin becomes Hi-Z.

When transferring to the low power consumption mode, set DACCNT<OP>="0" and <VREFON>="0".

Table 20-3 Relationship between DACCNTx register setting value and analog output value

DACCNTx register		Analog output value
<OP>	<VREFON>	
0	0	Hi-Z
0	1	Hi-Z
1	0	0V (to 0.2V) (Note)depending on current value.
1	1	DACREGx<DAC> register value

20.4.5 VOUTHOLD Time Adjustment Function

DACVCTLx<VHOLDCTB>, <VHOLDCTF> is a register to set a waiting time for waveform outputs until stabilizing. Note that DACVCTLx<VHOLDCTB>, <VHOLDCTF> is not modified while VOUTHOLD signal is ON.

20.4.6 Settling Time

The settling time Tstlng is a time from rising of VOUTHOLD signal to stabilization of waveform. The setling time Tstlng is depended on DACVCTLx<VHOLDCTB>, <VHOLDCTF>.

When fsys is 80MHz, VHOLDCTB=200ns and VHOLDCTF=25ns is set, the settling time tstlng is 3μs in maximum. If the amount of variable is within 16LSB, the settling time is 1μsec in maximum.

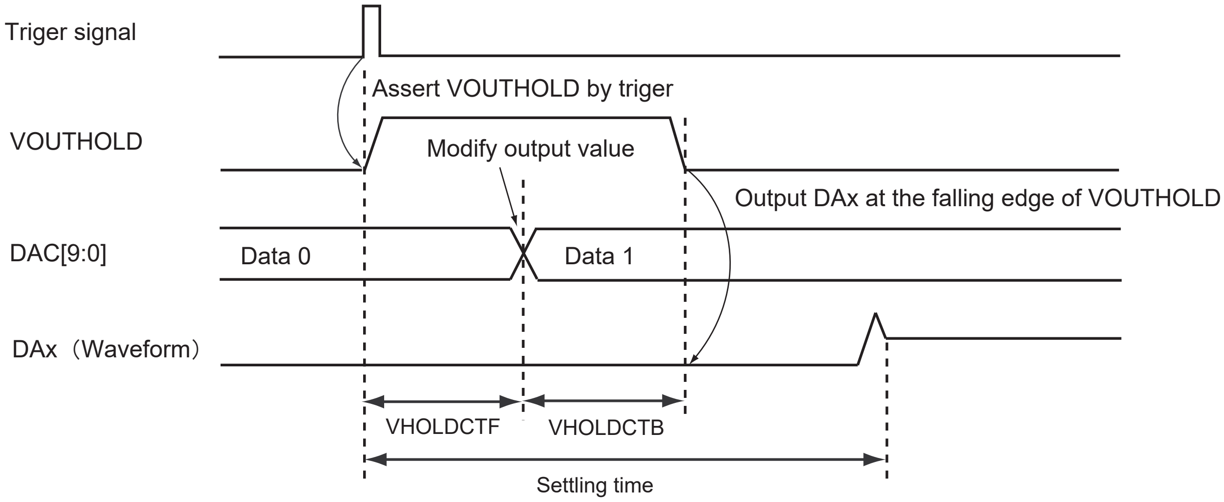

Triger signal
Assert VOUTHOLD by triger
VOUTHOLD
Modify output value
Output DAx at the falling edge of VOUTHOLD
DAC[9:0]
Data 0
Data 1
DAx（Waveform）
VHOLDCTF
VHOLDCTB
Settling time

Figure 20-4 VOUTHOLD and settling time

Table 20-4 The value of VHOLDCTB/VHOLDCTF (min.)

fsys [MHz]	VHOLD time (min.)			Tstlng(max) - Tvhld [μs]	
	<VHOLDCTB>	<VHOLDCTF>	Tvhld [s]	16LSB	max
70<fsys≤80	1000	0001	2/fsys × 9	0.775	2.775
60<fsys≤70	0111	0001	2/fsys × 8		
50<fsys≤60	0110	0001	2/fsys × 7		
40<fsys≤50	0101	0001	2/fsys × 6		
30<fsys≤40	0100	0001	2/fsys × 5		
20<fsys≤30	0011	0001	2/fsys × 4		
10<fsys≤20	0010	0001	2/fsys × 3		
1<fsys≤10	0001	0001	2/fsys × 2		

21. 16-bit Multi-Purpose Timer (MPT)

21.1 Outline

TMPM367FDFG has four channels of multi-purpose timer (MPT).

The MPT provides three operational modes as follows.

<Timer mode>

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable rectangular waveform output (PPG, one output) mode
- Pulse width measurement (capture)

<IGBT mode>

- 16-bit programmable rectangular waveform output (PPG, two outputs) mode
- External trigger start
- Cycle match detection
- Emergency stop function

<PMD mode>

- 3-phase motor control mode

Hereafter, "x" indicates a channel number.

Note: **MPT1, MPT2 and MPT3 do not have PMD mode.**

21.2 Specification Differences in Channel line-up

Each channel (MPT0-MPT3) operates independently and identically except the differences shown in the Table 21-1.

Table 21-1 MPT specification differences in channel line-up.

Specification / Channel	External pin					
	External clock/ capture trigger input pin	Timer flip-flop output pin	IGBT input pin	IGBT output pin	PMD input pin	PMD output pin
Channel 0	MTTB0IN	MTTB0OUT	$\overline{\text{GEMG0}}$ MT0IN	MTOUT00 MTOUT10	$\overline{\text{EMG}}$	UO,VO, WO,XO, YO,ZO
Channel 1	MTTB1IN	MTTB1OUT	$\overline{\text{GEMG1}}$ MT1IN	MTOUT01 MTOUT11	-	-
Channel 2	MTTB2IN	MTTB2OUT	$\overline{\text{GEMG2}}$ MT2IN	MTOUT02 MTOUT12	-	-
Channel 3	MTTB3IN	MTTB3OUT	$\overline{\text{GEMG3}}$ MT3IN	MTOUT03 MTOUT13	-	-

21.3 Block Diagram

The MPT consists of three modules including a timer, IGBT and PMD. Each module is switched by registers.

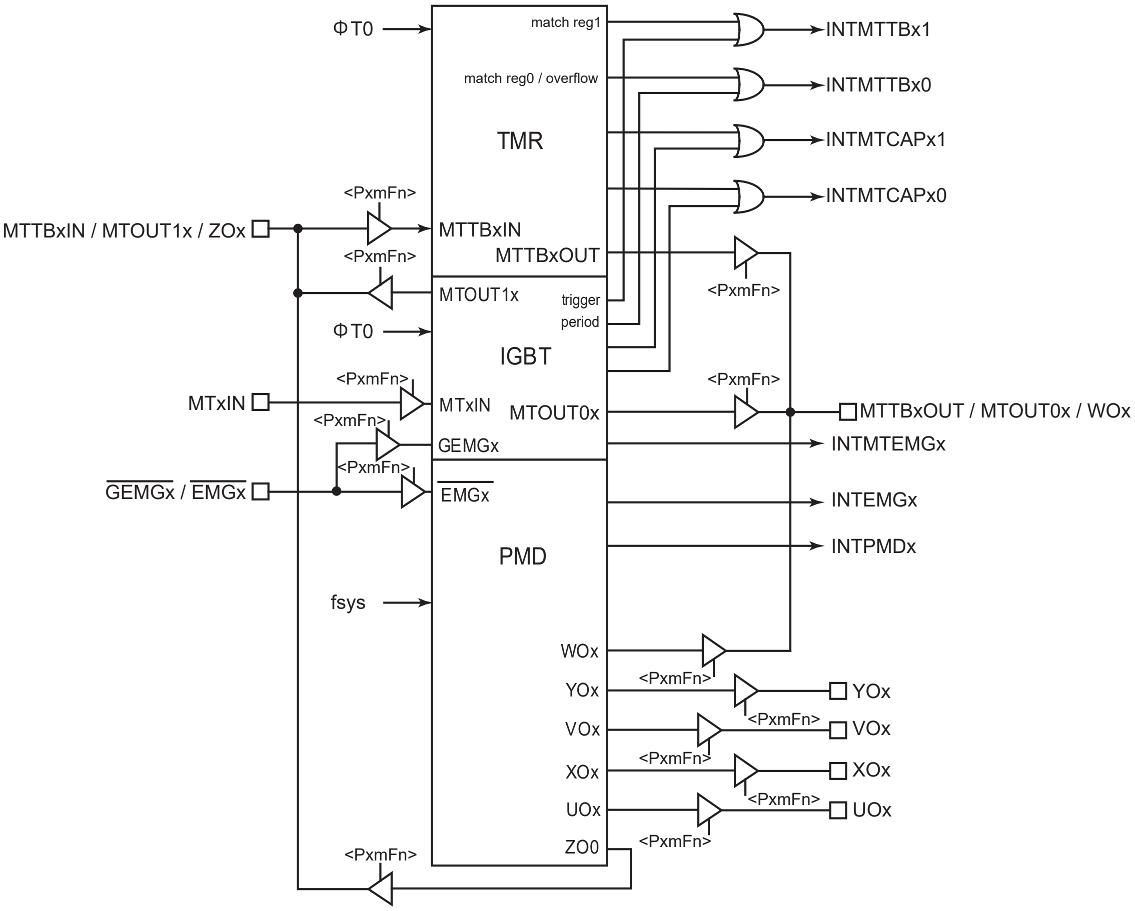

ΦT0
match reg1
match reg0 / overflow
TMR
INTMTTBx1
INTMTTBx0
INTMTCAPx1
INTMTCAPx0
<PxmFn>
MTTBxIN / MTOUT1x / ZOx
MTTBxIN
MTTBxOUT
MTOUT1x
trigger
period
ΦT0
IGBT
MTxIN
MTxIN
MTOUT0x
GEMGx
MTTBxOUT / MTOUT0x / WOx
INTMTEMGx
$\overline{\text{GEMGx}}$ / $\overline{\text{EMGx}}$
$\overline{\text{EMGx}}$
INTEMGx
INTPMDx
PMD
fsys
WOx
YOx
VOx
XOx
UOx
ZO0
YOx
VOx
XOx
UOx

Figure 21-1 Block Diagram of MPTx

Note: MPT1, MPT2 and MPT3 do not have PMD mode.

21.4 Operation Description of Timer Mode

21.4.1 Block Diagram

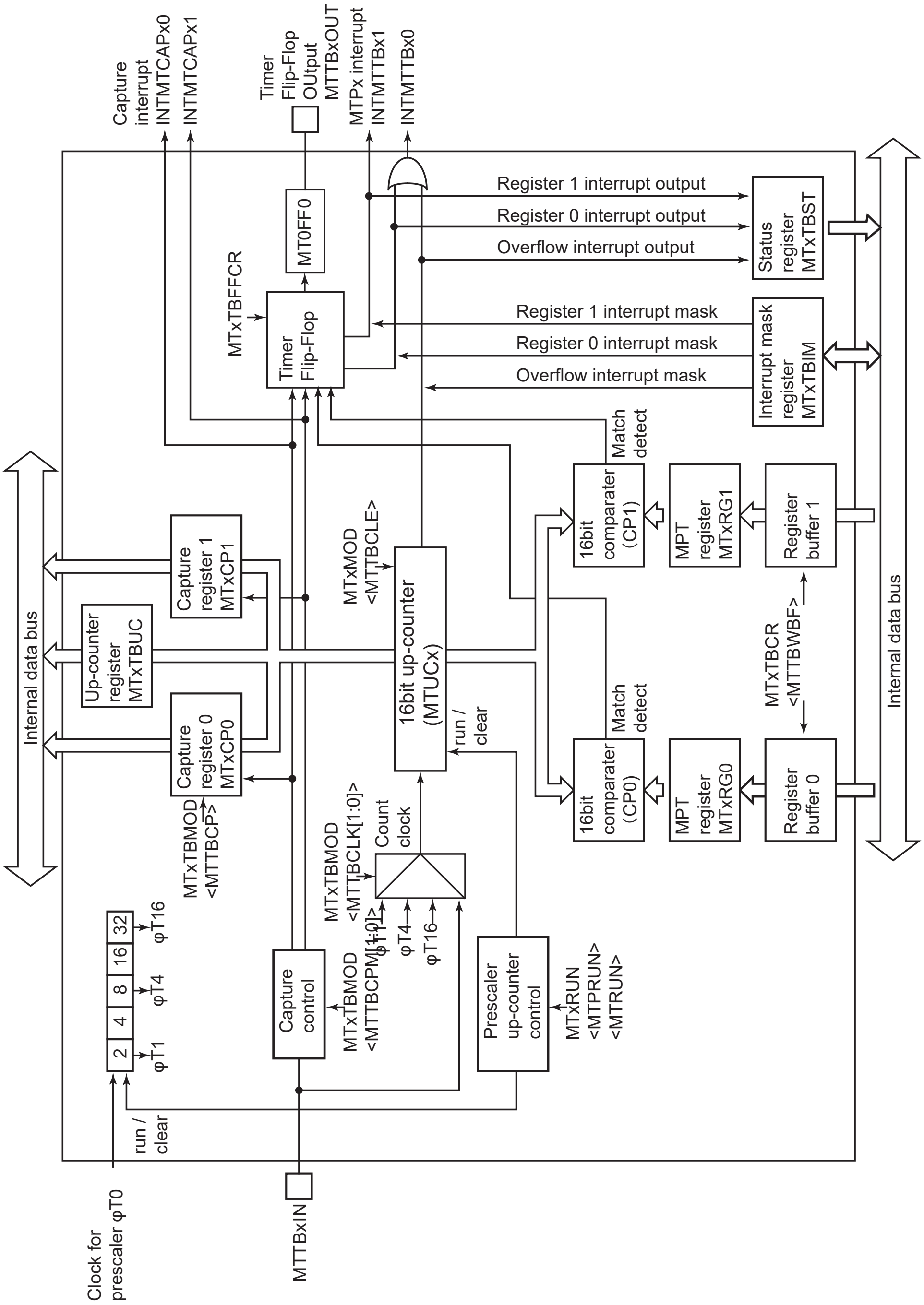

Capture interrupt INTMTCAPx0
INTMTCAPx1
Timer Flip-Flop OUtput MTTBxOUT
MTPx interrupt INTMTTBx1
INTMTTBx0
Register 1 interrupt output
Register 0 interrupt output
Overflow interrupt output
Status register MTxTBST
MT0FF0
MTxTBFFCR
Timer Flip-Flop
Register 1 interrupt mask
Register 0 interrupt mask
Overflow interrupt mask
Interrupt mask register MTxTBIM
Match detect
16bit comparater (CP1)
MPT register MTxRG1
Register buffer 1
Internal data bus
Capture register 1 MTxCP1
MTxMOD <MTTBCLE>
16bit up-counter (MTUCx)
Up-counter register MTxTBUC
MTxTBCR <MTTBWBF>
Internal data bus
run / clear
Match detect
Capture register 0 MTxCP0
16bit comparater (CP0)
MPT register MTxRG0
Register buffer 0
MTxTBMOD <MTTBCP>
MTxTBMOD <MTTBCLK[1:0]>
Count clock
2
4
8
16
32
φT1
φT4
φT16
φT1
φT4
φT16
Capture control
MTxTBMOD <MTTBCPM[1:0]>
Prescaler up-counter control
MTxRUN <MTPRUN> <MTRUN>
run / clear
Clock for prescaler φT0
MTTBxIN

Figure 21-2 Block Diagram of Timer Mode

21.4.2 Registers categorized by timer mode channel

This section describes registers and addresses of each channel.

Channel x	Base Address
Channel 0	0x400C_7000
Channel 1	0x400C_7100
Channel 2	0x400C_7200
Channel 3	0x400C_7300

Register name (x=0 to 3)		Address (Base+)
MPT enable register	MTxEN	0x0000
MPT RUN register	MTxRUN	0x0004
MPT control register	MTxTBCR	0x0008
MPT mode register	MTxTBMOD	0x000C
MPT flip-flop control register	MTxTBFFCR	0x0010
MPT status register	MTxTBST	0x0014
MPT interrupt mask register	MTxTBIM	0x0018
MPT up-counter register	MTxTBUC	0x001C
MPT register	MTxRG0 MTxRG1	0x0020 0x0024
MPT capture register	MTxCP0 MTxCP1	0x0028 0x002C

21.4.3 MTxEN (MPT 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	MTEN	MTHALT	-	-	-	-	-	MTMODE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	MTEN	R/W	Specifies MPT operation. 0: Disable 1: Enable When MTEN is disabled, feeding clock to other registers of MPT module is stopped, so that power consumption can be reduced. (Read or write to other registers cannot be done.)
6	MTHALT	R/W	Specifies MPT operation when core halts (debug break). [TMR function] 0: Clock stopping operation is disabled while core halts. 1: Clock stopping operation is enabled while core halts. [IGBT function] 0: Not control clock stopping operation and MTOUT0x/MTOUT1x output. 1: Clock stopping operation is enabled while core halt. It controls MTOUT0x/MTOUT1x output according to the MTxIGEMGCR<IGEMGOC> setting.
5-1	-	R	Read as "0".
0	MTMODE	R/W	Specifies operation modes 0: Timer mode 1: IGBT mode

Note: **When MPT is used, MPT operation is enabled (<MTEN>="1")before each register of MPT module is set. Even if MPT operation is disabled after MPT is stopped, each register setting is maintained.**

21.4.4 MTxRUN (MPT 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	-	-	-	-	-	MTPRUN	-	MTRUN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	-	R	Read as "0".
2	MTPRUN	R/W	Controls MPT prescaler operation 0: Stops prescaler operation. Prescaler is cleared to "0". 1: Starts prescaler operation.
1	-	R	Read as "0".
0	MTRUN	R/W	Controls MPT counting operation 0: Stops counting operation. Counter is cleared to "0". 1: Starts counting operation.

21.4.5 MTxTBCR (MPT 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	MTTBWBF	-	-	-	MTI2TB	-	MTTB TRGSEL	MTTBCSSEL
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	MTTBWBF	R/W	Specifies double buffer to enable/disable 0:Disabled 1:Enabled
6-5	-	R/W	Write "0".
4	-	R	Read as "0".
3	MTI2TB	R/W	Controls clock operation to star/stop in IDLE mode 0:Stop 1:Start
2	-	R	Read as "0".
1	MTTBTRGSEL	R/W	Selects rising or falling edge of external trigger. 0:Rising edge 1:Falling edge
0	MTTBCSSEL	R/W	Selects how to start counting 0:Soft start 1:External trigger

Note 1: **Do not modify MTxTBCR during timer in operation (MTxRUN<MTRUN>="1").**

Note 2: **In the IGBT mode, double-buffering is automatically enabled regardless of <MTTBWBF> setting.**

21.4.6 MTxTBMOD (MPT 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	-	MTTBRSWR	MTTBCP	MTTBCPM		MTTBCLE	MTTBCLK	
After reset	0	0	1	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	-	R	Read as "0".
6	MTTBRSWR	R/W	Controls the write timing to timer register 0 and 1 when double-buffer is used. 0: If either timer register 0 or timer register 1 is prepared to be written, one register can be written at a time. 1: If both timer register 0 and timer register 1 are not prepared, timer register cannot be written.
5	MTTBCP	W	Controls software capture 0: Capture count values to the capture register 0 (MTxCP0) 1: Don't care
4-3	MTTBCPM[1:0]	R/W	Sets capture timing 00: Capture is disabled. 01: At the rising edge of MTTBxIN input, counter values are captured to the capture register 0 (MTxCP0) 10: At the rising edge of MTTBxIN input, counter values are captured to the capture register 0 (MTxCP0). At the falling edge of MTTBxIN input, counter values are captured to the capture register 1 (MTxCP1). 11: Capture is disabled.
2	MTTBCLE	R/W	Clear MPT up-counter 0: Clear is disabled. 1: Clear MPT up-counter by matching with timer register 1 (MTxRG1)
1-0	MTTBCLK[1:0]	R/W	Selects timer count clock of MPT 00: MTTBxIN input 01: φT1 10: φT4 11: φT16

Note 1: **MTxTBMOD<MTTBCP> reads as "1".**

Note 2: **Do not modify MTxTBMOD during timer in operation (MTxRUN<MTRUN>="1").**

21.4.7 MTxTBFFCR (MPT 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	-	-	MTTBC1T1	MTTBC0T1	MTTBE1T1	MTTBE0T1	MTTBFF0C	
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 "11".
5	MTTBC1T1	R/W	Controls timer flip-flop reverse when up-counter values are captured to the capture register 1 (MTxCP1). 0: MTxFF0 does not reverse. 1: MTxFF0 reverses.
4	MTTBC0T1	R/W	Controls timer flip-flop reverse when up-counter values are captured to the capture register 1 (MTxCP0) 0: MTxFF0 does not reverse. 1: MTxFF0 reverses.
3	MTTBE1T1	R/W	Controls timer flip-flop reverse when up-counter values and the timer register 1 (MTxRG1) are matched. 0: MTxFF0 does not reverse. 1: MTxFF0 reverses.
2	MTTBE0T1	R/W	Controls timer flip-flop reverse when up-counter values and the timer register 1 (MTxRG0) are matched. 0: MTxFF0 does not reverse. 1: MTxFF0 reverses.
1-0	MTTBFF0C	R/W	Controls timer flip-flop 00: Reverses a value of MTxFF0. 01: Sets "1" to MTxFF0. 10: Sets "0" MTxFF0 to clear. 11: Don't care. Read as "11".

Note: Do not modify **MTxTBFFCR during timer in operation (MTxRUN<MTRUN>="1").**

21.4.8 MTxTBST (MPT 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	-	-	-	-	-	MTTBINT TBOF	MTTBINTTB1	MTTBINTTB0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	-	R	Read as "0".
2	MTTBINTTBOF	R	Indicates the status of up-counter overflow interrupt generation. 0: No interrupt generation 1: Interrupt generation
1	MTTBINTTB1	R	Indicates the interrupt generation by matching with timer register 1 (MTxRG1) 0: No interrupt generation 1: Interrupt generation
0	MTTBINTTB0	R	Indicates the interrupt generation by matching with timer register 0 (MTxRG0) 0: No interrupt generation 1: Interrupt generation

Note: **Once any interrupt generates, corresponding flag in MTxTBST register is set to notify CPU of an interrupt generation. If MTxTBST register is read, the flag is cleared to "0".**

21.4.9 MTxTBIM (MPT 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	-	-	-	-	-	MTTBIMOF	MTTBIM1	MTTBIM0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	-	R	Read as "0".
2	MTTBIMOF	R/W	Controls up-counter overflow interrupt to mask 0: Not mask interrupt 1: Masks interrupt
1	MTTBIM1	R/W	Controls to mask the interrupt when the match between timer register 1 (MTxRG1) and up-counter. 0: Not mask interrupt 1: Masks interrupt
0	MTTBIM0	R/W	Controls to mask the interrupt when the match between timer register 0 (MTxRG0) and up-counter. 0: Not mask interrupt 1: Masks interrupt

Note: MTxTBST reflects interrupt requests even though **MTxTBIM masks interrupts.**

21.4.10 MTxTBUC (MPT read 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	MTUC							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTUC							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTUC[15:0]	R	Reading MTxTBUC captures the current up-counter value.

21.4.11 MTxRG0/MTxRG1 (MPT timer register)

MTxRG0

	31	30	29	28	27	26	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	MTRG0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTRG0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTRG0[15:0]	R/W	Timer count value [Timer mode] When up-counter values match with MTRG0[15:0], match detection interrupt (INTMTTBx0) occurs. Also, MTTBxOUT can be reversed when matching, [IGBT mode] When up-counter values match with MTRG0[15:0], MTOUT0x becomes active level.

Note 1: **Use half word access or word access.**

Note 2: **Set to the condition of 0x0000 < MTxRG0 < MTxRG1 ≤ MTxIRG40 ≤ 0xFFFF.**

MTxRG1

	31	30	29	28	27	26	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	MTRG1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTRG1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTRG1[15:0]	R/W	Timer count value [Timer mode] When up-counter values match with MTRG1[15:0], match detection interrupt (INTMTTBx1) occurs. Also, MTTBxOUT can be reversed when matching. [IGBT mode] When up-counter values match with MTRG1[15:0], MTOUT0x becomes inactive level.

Note 1: **Use half word access or word access.**

Note 2: **Set to the condition of 0x0000 < MTxRG0 < MTxRG1 ≤ MTxIRG40 ≤ 0xFFFF.**

21.4.12 MTxCP0 /MTxCP1 (MPT capture register)

MTnCP0

	31	30	29	28	27	26	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	MTCP0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTCP0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTCP0[15:0]	R	Read captured up-counter values.

MTnCP1

	31	30	29	28	27	26	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	MTCP1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTCP1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTCP1[15:0]	R	Read captured up-counter values.

Note: **During the timer stopping, a value of timer counter (MTUCx) cannot be read. When the timer stops, a value previously captured is held and the value can be read.**

21.5 Operational Description categorized by circuit

21.5.1 Prescaler

This 4-bit prescaler generates the source clock for up-counter MTUCx.

Input clock φT0 to the prescaler is chosen among fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16 and fperiph/32 by specifying with CGSYSCR<PRCK[2:0]>. This peripheral clock (fperiph) is either fgear specified with CG SYSCR<FPSEL> or fc that is pre-divided clock gear.

Prescaler is set to enable/disable with MTxRUN<MTPRUN>. When MTxRUN<MTPRUN> is set to "1", counting starts. When MTxRUN<MTPRUN> is set to "0", the counter is stopped and cleared. Table 21-2 shows prescaler output clock resolutions.

Table 21-2 Prescaler output clock resolutions (fc = 80MHz)

Peripheral clock selection <FPSEL>	Clock gear value <GEAR[2:0]>	Prescaler clock selection <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
0 (fgear)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.025 μs)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)
		001 (fperiph/2)	fc/2^{2} (0.05 μs)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)
		010 (fperiph/4)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		011 (fperiph/8)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		100 (fperiph/16)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		101 (fperiph/32)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^{2} (0.05 μs)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)
		001 (fperiph/2)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		010 (fperiph/4)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		011 (fperiph/8)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		100 (fperiph/16)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
		101 (fperiph/32)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		001 (fperiph/2)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		010 (fperiph/4)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		011 (fperiph/8)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
		100 (fperiph/16)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)
		101 (fperiph/32)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
	110 (fc/8)	000 (fperiph/1)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		001 (fperiph/2)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		010 (fperiph/4)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
		011 (fperiph/8)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)
		100 (fperiph/16)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
		101 (fperiph/32)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)	fc/2^{13} (102.4 μs)
	111 (fc/16)	000 (fperiph/1)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		001 (fperiph/2)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
		010 (fperiph/4)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)
		011 (fperiph/8)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
		100 (fperiph/16)	fc/2^{9} (6.4 μs)	fc/2^{11} (25.6 μs)	fc/2^{13} (102.4 μs)
		101 (fperiph/32)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)	fc/2^{14} (204.8 μs)

Table 21-2 Prescaler output clock resolutions (fc = 80MHz)

Peripheral clock selection <FPSEL>	Clock gear value <GEAR[2:0]>	Prescaler clock selection <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
1 (fc)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.025 μs)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)
		001 (fperiph/2)	fc/2^{2} (0.05 μs)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)
		010 (fperiph/4)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		011 (fperiph/8)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		100 (fperiph/16)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		101 (fperiph/32)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
	100 (fc/2)	000 (fperiph/1)	–	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)
		001 (fperiph/2)	fc/2^{2} (0.05 μs)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)
		010 (fperiph/4)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		011 (fperiph/8)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		100 (fperiph/16)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		101 (fperiph/32)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
	101 (fc/4)	000 (fperiph/1)	–	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)
		001 (fperiph/2)	–	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)
		010 (fperiph/4)	fc/2^{3} (0.1 μs)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		011 (fperiph/8)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		100 (fperiph/16)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		101 (fperiph/32)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
	110 (fc/8)	000 (fperiph/1)	–	–	fc/2^{5} (0.4 μs)
		001 (fperiph/2)	–	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)
		010 (fperiph/4)	–	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		011 (fperiph/8)	fc/2^{4} (0.2 μs)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		100 (fperiph/16)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		101 (fperiph/32)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)
	111 (fc/16)	000 (fperiph/1)	–	–	fc/2^{5} (0.4 μs)
		001 (fperiph/2)	–	–	fc/2^{6} (0.8 μs)
		010 (fperiph/4)	–	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)
		011 (fperiph/8)	–	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)
		100 (fperiph/16)	fc/2^{5} (0.4 μs)	fc/2^{7} (1.6 μs)	fc/2^{9} (6.4 μs)
		101 (fperiph/32)	fc/2^{6} (0.8 μs)	fc/2^{8} (3.2 μs)	fc/2^{10} (12.8 μs)

Note 1: **Prescaler output clock φTn must satisfy the condition of φTn <fsys. (φTn must be slower than fsys.)**

Note 2: **Do not change the clock gear during timer in operation.**

Note 3: **In the above table, "−" indicates prohibition.**

21.5.2 Up-Counter (MTUC0)

This counter is a 16-bit binary counter.

- Source clock

  The source clock can be set with MTxTBMOD<MTTBCLK[1:0]>.

  Prescaler output clock can choose among φT1, φT4, φT16 or external clock of MTTBxIN pin.

- Start/Stop counter operation

  Counter operation is set with MTxRUN<MTRUN>. When <MTRUN>="1" is set, counter operation starts. When <MTRUN>="1" is set, the counter is stopped and cleared at the same time.

  When a value of up-counter MTUCx detects the match with a setting value of timer register MTxRG0/MTxRG1, INTMTTB0x or INTMTTB1x occurs.

- Counter clear timing

  1. Comparing a match

     If MTxTBMOD<MTTBCLE>="1" is set, the counter is cleared when comparing matches with MTxRG1.

     If MTxTBMOD<MTTBCLE>="0" is set, the counter becomes a free-running counter.

  2. Counter stopping

     If MTxRUN<MTRUN>="0" is set, the counter is stopped and cleared.

- Overflow of the counter

  If MTUCx is overflowed, an overflow interrupt INTMTTB0x occurs.

21.5.3 Timer Register (MTxRG0, MTxRG1)

Timer register sets a values to compare with up-counter MTUCx. Comparator compares a value of timer register with a value of up-counter. If these two are matched, the match detection signal is output.

- Structure

  In the timer register, MTxRG0/1 is double-buffering structure paired with register buffer.

  Double-buffer is set to enable/disable with MTxTBCR<MTTBWBF>. If <MTTBWBF>="0" is set, double-buffer is disabled. If <MTTBWBF>="1" is set, double-buffer is enabled.

  While double-buffer is enabled, data transfer is taken place from register buffer 0 to timer register MTxRG0/1 when MTUCx matches with MTxRG1.

- Initial state

  After reset, MTxRG0 and MTxRG1 are undefined and double-buffer is disabled.

- How to set

  1. If double-buffer is not used.

     Use half-word access or word access

  2. If double-buffer is used.

     MTxRG0 and 1, and register buffer 0 and 1 are assigned to the same address respectively.

     When <MTTBWBF> is "0", MTxRG0 and 1 and each register buffer are written the same value. When <MTTBWBF> is "1", only corresponding register buffer is written data. Thus when writing the initial value to timer register, set as follows; firstly register buffer is disabled, secondly timer register is written data, thirdly <MTTBWBF> is set to "1". Finally next data is written to register buffer.

21.5.4 Capture Control

This circuit controls the timing when a value of up-counter MTUCx is latched by capture register MTxCP0/MTxCP1. This latch timing is set with MTxTBMOD<MTTBCPM[1:0]>.

Also the timing is controlled by software. Every time MTxTBMOD<MTTBCP> is set to "0", a value of MTUCx is captured to the capture register MTxCP0 at the time. Note that prescaler must be set to RUN status (MTxRUN<MTPRUN> "1").

21.5.5 Capture Register (MTxCAP0、MTxCAP1)

This register captures a value of up-counter MTUCx.

21.5.6 Up-counter Capture Register (MTxTBUC)

Besides the capture function using capture control circuit, current counter value of up-counter (MTUC0) is also captured by reading MTxTBUC register.

21.5.7 Comparators (CP0, CP1)

This comparator detects the match comparing a value of up-counter (MTUCx) with a setting value of timer register MTxRG0/MTxRG1. If these values are matched, INTMTTBx0 or INTMTTBx1 occurs.

21.5.8 Timer Flip-flop (MTxFF0)

Timer flip-flop circuit (MTxFF0) reverses by a match signal from comparators or a latch signal to the capture register. This reverse is enabled/disabled with MTxTBFFCR<MTTBC1T1, MTTBC0T1, MTTBE1T1, MTTBE0T1>.

After reset, a value of MTxFF0 is undefined. If MTxTBFFCR<MTTBFF0C[1:0]> is set to "00", the reverse is enabled. If MTxTBFFCR<MTTBFF0C[1:0]> is set to "01", MTxFF0 is set to "1". MTxTBFFCR<MTTBFF0C[1:0]> is set to "10", MTxFF0 is set to "0" to clear.

A value of MTxFF0 can be output to timer output pin MTTBxOUT. If timer output is used, port related registers (PxCR and PxFR) must be set beforehand.

21.5.9 Capture Interrupts (INTMTCAPx0, INTMTCAPx1)

Capture interrupts (INTMTCAPx0 and INTMTCAPx1) occur respectively at the timing when data is latched to each capture register (MTxCP0 and MTxCP1). Interrupt setting is set by CPU.

21.6 Operational Description in IGBT mode

21.6.1 Block Diagram

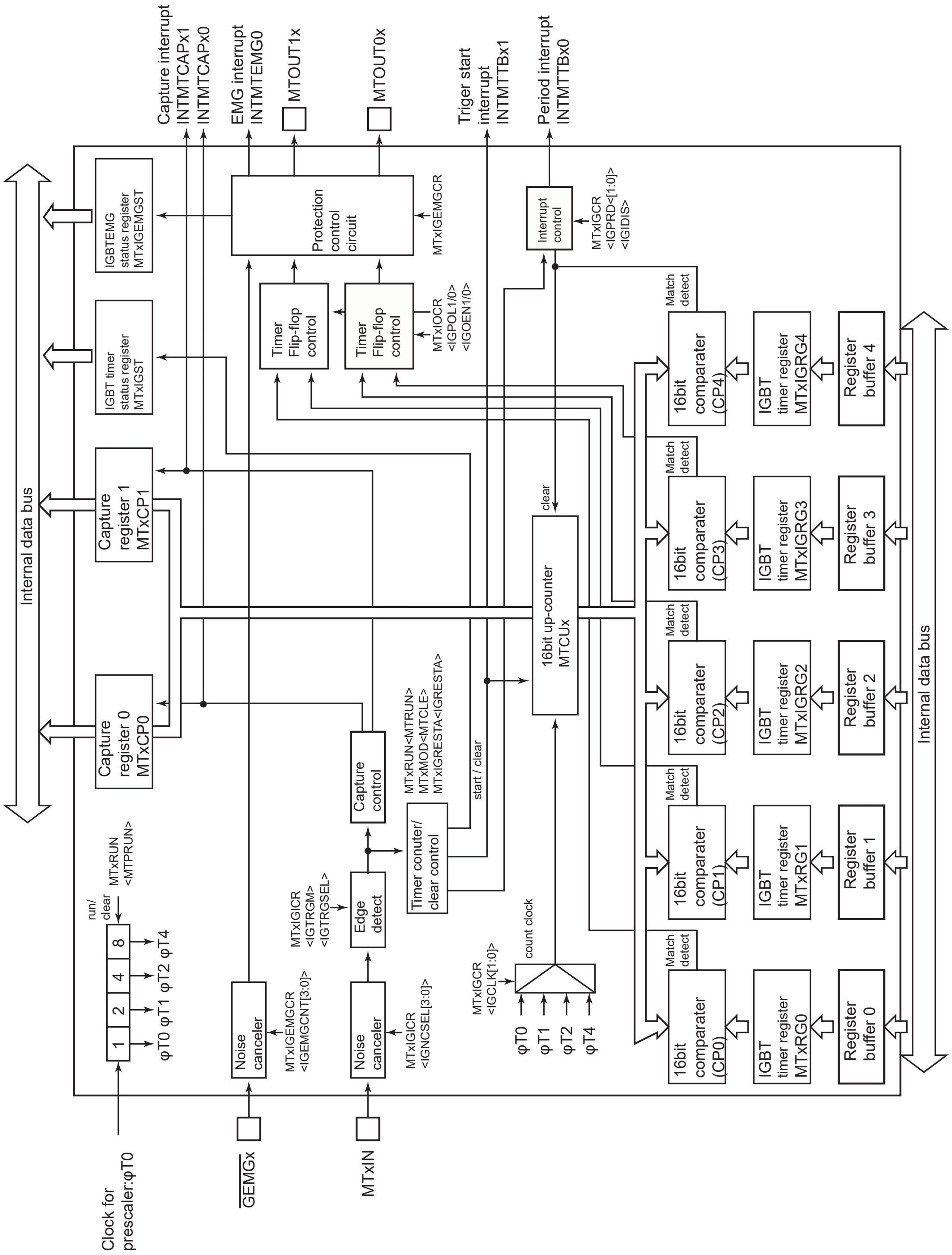

Capture interrupt
INTMTCAPx1
INTMTCAPx0
EMG interrupt
INTMTEMG0
MTOUT1x
MTOUT0x
Triger start
interrupt
INTMTTBx1
Period interrupt
INTMTTBx0
IGBTEMG
status register
MTxIGEMGST
Protection
control
circuit
MTxIGEMGCR
Interrupt
control
MTxIGCR
<IGPRD<1:0>>
<IGIDIS>
IGBT timer
status register
MTxIGST
Timer
Flip-flop
control
Timer
Flip-flop
control
MTxIOCR
<IGPOL1/0>
<IGOEN1/0>
Match
detect
16bit
comparater
(CP4)
IGBT
timer register
MTxIGRG4
Register
buffer 4
Internal data bus
Capture
register 1
MTxCP1
clear
16bit
comparater
(CP3)
IGBT
timer register
MTxIGRG3
Register
buffer 3
16bit up-counter
MTCUx
16bit
comparater
(CP2)
IGBT
timer register
MTxIGRG2
Register
buffer 2
Internal data bus
Capture
register 0
MTxCP0
MTxRUN<MTRUN>
MTxMOD<MTCLE>
MTxIGRESTA<IGRESTA>
start / clear
Capture
control
Timer conuter/
clear control
16bit
comparater
(CP1)
IGBT
timer register
MTxRG1
Register
buffer 1
run/
clear
MTxRUN
<MTPRUN>
MTxIGICR
<IGTRGM>
<IGTRGSEL>
Edge
detect
count clock
MTxIGCR
<IGCLK[1:0]>
1
2
4
8
φT0 φT1 φT2 φT4
Noise
canceler
MTxIGEMGCR
<IGEMGCNT[3:0]>
Noise
canceler
MTxIGICR
<IGNCSEL[3:0]>
φT0
φT1
φT2
φT4
16bit
comparater
(CP0)
IGBT
timer register
MTxRG0
Register
buffer 0
Clock for
prescaler:φT0
GEMGx
MTxIN

Figure 21-3 Block Diagram in IGBT mode

21.6.2 Registers in IGBT mode categorized by channel

This section describes registers and addresses of each channel.

Channel x	Base Address
Channel 0	0x400C_7000
Channel 1	0x400C_7100
Channel 2	0x400C_7200
Channel 3	0x400C_7300

Register name (x=0 to 3)		Address (Base+)
MPT enable register	MTxEN	0x0000
MPT RUN register	MTxRUN	0x0004
MPT register	MTxRG0 MTxRG1	0x0020 0x0024
MPT capture register	MTxCP0 MTxCP1	0x0028 0x002C
IGBT control register	MTxIGCR	0x0030
IGBT timer restart register	MTxIGRESTA	0x0034
IGBT timer status register	MTxIGST	0x0038
IGBT input control register	MTxIGICR	0x003C
IGBT output control register	MTxIGOCR	0x0040
IGBT timer register 2, 3, 4	MTxIGRG2 MTxIGRG3 MTxIGRG4	0x0044 0x0048 0x004C
IGBT EMG control register	MTxIGEMGCR	0x0050
IGBT EMG status register	MTxIGEMGST	0x0054

21.6.3 MTxEN (MPT 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	MTEN	MTHALT	-	-	-	-	-	MTMODE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	MTEN	R/W	Specifies MPT operation. 0: Disabled 1: Enabled When MTEN is disabled, feeding clock to other registers of MPT module is stopped, so that power consumption can be reduced. (Read or write to other registers cannot be done.)
6	MTHALT	R/W	Specifies MPT operation when core halts (debug break). [TMR function] 0: Clock stopping operation is disabled while core halts. 1: Clock stopping operation is enabled while core halts. [IGBT function] 0: Not control clock stopping operation and MTOUT0x/MTOUT1x output. 1: Clock stopping operation is enabled while core halt. It controls MTOUT0x/MTOUT1x output according to the MTxIGEMGCR<IGEMGOC> setting.
5-1	-	R	Read as "0".
0	MTMODE	R/W	Specifies operation mode. 0: Timer mode 1: IGBT mode

Note: When MPT is used, MPT operation is enabled (<MTEN>="1")before each register of MPT module is set. If MPT operation is disabled after MPT is stopped, each register setting is maintained.

21.6.4 MTxRUN (MPT 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	-	-	-	-	-	MTPRUN	-	MTRUN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	-	R	Read as "0".
2	MTPRUN	R/W	Controls MPT prescaler operation 0: Stops prescaler operation. Prescaler is cleared to "0". 1: Starts prescaler operation.
1	-	R	Read as "0".
0	MTRUN	R/W	Controls MPT counting operation 0: Stops counting operation. Counter is cleared to "0". 1: Starts counting operation.

21.6.5 MTxRG0/MTxRG1 (MPT timer register)

MTxRG0

	31	30	29	28	27	26	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	MTRG0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTRG0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTRG0[15:0]	R/W	Timer count value [Timer mode] When up-counter values match with MTRG0[15:0], match detection interrupt (INTMTTBx0) occurs. Also, when matching, MTTBxOUT can be reversed. [IGBT mode] When up-counter values match with MTRG0[15:0], MTOUT0x becomes active level.

Note 1: **Use half word access or word access.**

Note 2: **Set to the condition of 0x0000 < MTxRG0 < MTxRG1 ≤ MTxIRG40 ≤ 0xFFFF.**

MTxRG1

	31	30	29	28	27	26	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	MTRG1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTRG1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTRG1[15:0]	R/W	Timer count value [Timer mode] When up-counter values match with MTRG1[15:0], match detection interrupt (INTMTTBx1) occurs. Also, MTTBxOUT can be reversed when matching. [IGBT mode] When up-counter values match with MTRG1[15:0], MTOUT0x becomes inactive level.

Note 1: **Use half word access or word access.**

Note 2: **Set to the condition of 0x0000 < MTxRG0 < MTxRG1 ≤ MTxIRG40 ≤ 0xFFFF.**

21.6.6 MTxCP0 /MTxCP1 (MPT capture register)

MTnCP0

	31	30	29	28	27	26	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	MTCP0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTCP0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTCP0[15:0]	R	Read captured up-counter values.

MTnCP1

	31	30	29	28	27	26	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	MTCP1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTCP1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTCP1[15:0]	R	Read captured up-counter values.

Note: **During the timer stopping, a value of timer counter (MTUCx) cannot be read. When the timer stops, a value previously captured is held and the value can be read.**

21.6.7 MTxIGCR (IGBT 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	-	-	-	-	-	IGIDIS	IGPRD	
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	IGSNGL	IGSTP		IGSTA		IGCLK	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-11	-	R	Read as "0".
10	IGIDIS	R/W	Controls an interrupt when commands start 0: Enabled 1: Disabled
9-8	IGPRD[1:0]	R/W	Chooses an interrupt cycle 00: Every one cycle 01: Every two cycles 10: Every four cycles 11: Reserved
7	-	R	Read as "0".
6	IGSNGL	R/W	Chooses IGBT operation 0: Continuous operation 1: Single operation
5-4	IGSTP[1:0]	R/W	Chooses stopping status 00: Initial output status and counter immediately stops to clear 01: Sustains output status and counter immediately stops to clear 10: After cycle time has elapsed then counter stops to clear 11: Reserved
3-2	IGSTA[1:0]	R/W	Chooses start mode 00: Command start and trigger capture 01: Command start and trigger start 10: Trigger start 11: Reserved
1-0	IGCLK[1:0]	R/W	Chooses a source clock of IGBT 00: φT0 01: φT1 10: φT2 11: φT4

Note 1: **Do not modify MTxIGCR during timer in operation (MTxRUN<MTRUN>="1").**

Note 2: **When the counter stops after specified cycle time has elapsed, or counter is stopped with (MTxIGCR<IGSTP>="10") and cleared with MTxRUN<MTRUN>, check if the timer is stopped by cycle interrupt generation. Then change the setting and restart.**

21.6.8 MTxIGRESTA (IGBT timer restart 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	-	-	-	-	-	-	-	IGRESTA
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as "0".
0	IGRESTA	W	Controls counting restart 0: Don't care 1: Restart Read as "0".

Note: **If MTxIGRESTA<IGRESTA> is set to "1" during timer in operation, timer counter can be cleared and restart. Please check the status of output waveform before setting is changed.**

21.6.9 MTxIGST (IGBT timer 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	-	-	-	-	-	-	-	IGST
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as "0".
0	IGST	R	Counter operation status 0: Stop 1: Operating

21.6.10 MTxIGICR (IGBT input control register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	IGTRGM	R/W	Controls trigger edge accept mode 0: Always accept 1: Acceptance is disabled during active level
6	IGTRGSEL	R/W	Chooses start trigger edges and its active levels. 0: Rising edge start and active level is "High". 1: Falling edge start and active level is "Low".
5-4	-	R	Read as "0".
3-0	IGNCSEL[3:0]	R/W	Trigger input noise elimination time selection Noise elimination time is calculated with the following formula: IGNCSEL[3:0] × 16 / fsys 0000: Noise filter is not used. 0001: Noise elimination time 16 / fsys [s] 0010: Noise elimination time 32 / fsys [s] 0011: Noise elimination time 48 / fsys [s] 0100: Noise elimination time 64 / fsys [s] 0101: Noise elimination time 80 / fsys [s] 0110: Noise elimination time 96 / fsys [s] 0111: Noise elimination time 112 / fsys [s] 1000: Noise elimination time 128 / fsys [s] 1001: Noise elimination time 144 / fsys [s] 1010: Noise elimination time 160 / fsys [s] 1011: Noise elimination time 176 / fsys [s] 1100: Noise elimination time 192 / fsys [s] 1101: Noise elimination time 208 / fsys [s] 1110: Noise elimination time 224 / fsys [s] 1111: Noise elimination time 240 / fsys [s]

Note 1: **Do not modify MTxIGCR during timer in operation (MTxRUN<MTRUN>="1").**

Note 2: **When MTxGCR<IGNCSEL[3:0]> is used, EMG protection circuit must be disabled (MTxIGEMGCR<IGEMGEN>="0").**

Note 3: **When MTxIGCR<IGNCSEL[3:0]> is changed, specified noise elimination time or more is required to start the timer with (MTxRUN<MTRUN>="1").**

21.6.11 MTxIGOCR (IGBT output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	IGPOL1	IGPOL0	-	-	IGOEN1	IGOEN0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-6	-	R	Read as "0".
5	IGPOL1	R/W	Indicates the initial state of MTOUT1x 0: Low 1: High
4	IGPOL0	R/W	Indicates the initial state of MTOUT0x 0: Low 1: High
3-2	-	R	Read as "0".
1	IGOEN1	R/W	Controls MTOUT1x output 0: Disable 1: Enable
0	IGOEN0	R/W	Control MTOUT0x output 0: Disabled 1: Enabled

Note: **MTOUT0x/MTOUT1x output is changing according to a content of IGBT output control register (MTxIGOCR) regardless of timer in operation/stopping. Check the operation status before MTxGOCR is set.**

21.6.12 MTxIGRG2 (IGBT timer 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	IGRG2							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	IGRG2							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	IGRG2[15:0]	R/W	Timer count value When up-counter matches with IGRG2[15:0], MTOUT1x becomes active level.

Note 1: **Use half-word access or word access.**

Note 2: **Set the value to the condition of 0<MTxIGRG2<MTxIGRG3≤MTxIGRG4≤0xFFFF.**

21.6.13 MTxIGRG3 (IGBT timer 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	IGRG3							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	IGRG3							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	IGRG3[15:0]	R/W	Timer count value When up-counter matches with <IGRG3[15:0]>, MTOUT1x becomes inactive level.

Note 1: **Use half-word access or word access.**

Note 2: **Set the value to the condition of 0<MTxIGRG2<MTxIGRG3≤MTxIGRG4≤0xFFFF.**

21.6.14 MTxIGRG4 (IGBT timer 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	IGRG4							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	IGRG4							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	IGRG4[15:0]	R/W	Timer count value Specifies IGBT mode cycle

Note 1: **Use half-word access or word access.**

Note 2: **Set the value to the condition of 0<MTxRG0<MTxRG1≤MTxIGRG4≤0xFFFF.**

Note 3: **Set the value to the condition of 0<MTxIGRG2<MTxIGRG3≤MTxIGRG4≤0xFFFF.**

21.6.15 MTxIGEMGCR (IGBT EMG 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	IGEMGCNT				-	IGEMGRS	IGEMGOC	IGEMGEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-4	IGEMGCNT[3:0]	R/W	GEMG input noise elimination time selection Noise elimination time is calculated with the following formula: IGEMGCNT[3:0]×16 / fsys 0000: Noise filter is not used. 0001: Input noise elimination time 16 / fsys [s] 0010: Input noise elimination time 32 / fsys [s] 0011: Input noise elimination time 48 / fsys [s] 0100: Input noise elimination time 64 / fsys [s] 0101: Input noise elimination time 80 / fsys [s] 0110: Input noise elimination time 96 / fsys [s] 0111: Input noise elimination time 112 / fsys [s] 1000: Input noise elimination time 128 / fsys [s] 1001: Input noise elimination time 144 / fsys [s] 1010: Input noise elimination time 160 / fsys [s] 1011: Input noise elimination time 176 / fsys [s] 1100: Input noise elimination time 192 / fsys [s] 1101: Input noise elimination time 208 / fsys [s] 1110: Input noise elimination time 224 / fsys [s] 1111: Input noise elimination time 240 / fsys [s]
3	-	R	Read as "0".
2	IGEMGRS	W	Return from EMG protection status 0: Don't care 1: Returned (automatically cleared to "0".) (Read as "0".)
1	IGEMGOC	R/W	Set the polarity of MTOUT0x/MTOUT1x at EMG protection 0: Inactive level 1: High-impedance
0	IGEMGEN	R/W	Controls EMG protection circuit operation 0: Disable 1: Enable

Note: **Do not modify MTxIGEMGCR during timer in operation (MTxRUN<MTRUN>="1").**

21.6.16 MTxIGEMGST (IGBT EMG 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	-	-	-	-	-	-	IGEMGIN	IGEMGST
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1	IGEMGIN	R	EMG input status after noise elimination 0: Low 1: High
0	IGEMGST	R	EMG protection status 0: Normal operation 1: During in protection Read value indicates EMG protection status

21.7 Operation Description categorized by circuit

21.7.1 Prescaler

This 4-bit prescaler generates the source clock for up-counter MTUCx.

Input clock φT0 to the prescaler is chosen among fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16 and fperiph/32 by specifying with CGSYSCR<PRCK[2:0]>. This peripheral clock (fperiph) is either fgear specified with CGSYSCR<FPSEL> or fc that is a pre-dividing clock gear.

Prescaler is set to enable/disable with MTxRUN<MTPRUN>. When MTxRUN<MTPRUN> is set to "1", counting starts. When MTxRUN<MTPRUN> is set to "0", the counter is stopped and cleared. Table 21-3 shows prescaler output clock resolutions.

Table 21-3 Prescaler output clock resolutions (fc = 80MHz)

Peripheral clock selection <FPSEL>	Clock gear value <GEAR[2:0]>	Prescaler clock selection <PRCK[2:0]>	Prescaler output clock function			
			φT0	φT1	φT2	φT4
0 (fgear)	000 (fc)	000 (fperiph/1)	fc (0.0125 μs)	fc/2^{1} (0.025 μs)	fc/2^{2} (0.05 μs)	fc/2^{3} (0.1 μs)
		001 (fperiph/2)	fc/2^{1} (0.025 μs)	fc/2^{2} (0.05 μs)	fc/2^{3} (0.1 μs)	fc/2^{4} (0.2 μs)
		010 (fperiph/4)	fc/2^{2} (0.05 μs)	fc/2^{3} (0.1 μs)	fc/2^{4} (0.2 μs)	fc/2^{5} (0.4 μs)
		011 (fperiph/8)	fc/2^{3} (0.1 μs)	fc/2^{4} (0.2 μs)	fc/2^{5} (0.4 μs)	fc/2^{6} (0.8 μs)
		100 (fperiph/16)	fc/2^{4} (0.2 μs)	fc/2^{5} (0.4 μs)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)
		101 (fperiph/32)	fc/2^{5} (0.4 μs)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^{1} (0.025 μs)	fc/2^{2} (0.05 μs)	fc/2^{3} (0.1 μs)	fc/2^{4} (0.2 μs)
		001 (fperiph/2)	fc/2^{2} (0.05 μs)	fc/2^{3} (0.1 μs)	fc/2^{4} (0.2 μs)	fc/2^{5} (0.4 μs)
		010 (fperiph/4)	fc/2^{3} (0.1 μs)	fc/2^{4} (0.2 μs)	fc/2^{5} (0.4 μs)	fc/2^{6} (0.8 μs)
		011 (fperiph/8)	fc/2^{4} (0.2 μs)	fc/2^{5} (0.4 μs)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)
		100 (fperiph/16)	fc/2^{5} (0.4 μs)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)
		101 (fperiph/32)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^{2} (0.05 μs)	fc/2^{3} (0.1 μs)	fc/2^{4} (0.2 μs)	fc/2^{5} (0.4 μs)
		001 (fperiph/2)	fc/2^{3} (0.1 μs)	fc/2^{4} (0.2 μs)	fc/2^{5} (0.4 μs)	fc/2^{6} (0.8 μs)
		010 (fperiph/4)	fc/2^{4} (0.2 μs)	fc/2^{5} (0.4 μs)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)
		011 (fperiph/8)	fc/2^{5} (0.4 μs)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)
		100 (fperiph/16)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)
		101 (fperiph/32)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)
	110 (fc/8)	000 (fperiph/1)	fc/2^{3} (0.1 μs)	fc/2^{4} (0.2 μs)	fc/2^{5} (0.4 μs)	fc/2^{6} (0.8 μs)
		001 (fperiph/2)	fc/2^{4} (0.2 μs)	fc/2^{5} (0.4 μs)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)
		010 (fperiph/4)	fc/2^{5} (0.4 μs)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)
		011 (fperiph/8)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)
		100 (fperiph/16)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)
		101 (fperiph/32)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)
	111 (fc/16)	000 (fperiph/1)	fc/2^{4} (0.2 μs)	fc/2^{5} (0.4 μs)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)
		001 (fperiph/2)	fc/2^{5} (0.4 μs)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)
		010 (fperiph/4)	fc/2^{6} (0.8 μs)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)
		011 (fperiph/8)	fc/2^{7} (1.6 μs)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)
		100 (fperiph/16)	fc/2^{8} (3.2 μs)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)
		101 (fperiph/32)	fc/2^{9} (6.4 μs)	fc/2^{10} (12.8 μs)	fc/2^{11} (25.6 μs)	fc/2^{12} (51.2 μs)

Table 21-3 Prescaler output clock resolutions (fc = 80MHz)

Peripheral clock selection <FPSEL>	Clock gear value <GEAR[2:0]>	Prescaler clock selection <PRCK[2:0]>	Prescaler output clock function			
			φT0	φT1	φT2	φT4
1 (fc)	000 (fc)	000 (fperiph/1)	fc (0.0125 μs)	fc/2^1 (0.025 μs)	fc/2^2 (0.05 μs)	fc/2^3 (0.1 μs)
		001 (fperiph/2)	fc/2^1 (0.025 μs)	fc/2^2 (0.05 μs)	fc/2^3 (0.1 μs)	fc/2^4 (0.2 μs)
		010 (fperiph/4)	fc/2^2 (0.05 μs)	fc/2^3 (0.1 μs)	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)
		011 (fperiph/8)	fc/2^3 (0.1 μs)	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)
		100 (fperiph/16)	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)	fc/2^7 (1.6 μs)
		101 (fperiph/32)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)	fc/2^7 (1.6 μs)	fc/2^8 (3.2 μs)
	100 (fc/2)	000 (fperiph/1)	–	fc/2^1 (0.025 μs)	fc/2^2 (0.05 μs)	fc/2^3 (0.1 μs)
		001 (fperiph/2)	fc/2^1 (0.025 μs)	fc/2^2 (0.05 μs)	fc/2^3 (0.1 μs)	fc/2^4 (0.2 μs)
		010 (fperiph/4)	fc/2^2 (0.05 μs)	fc/2^3 (0.1 μs)	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)
		011 (fperiph/8)	fc/2^3 (0.1 μs)	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)
		100 (fperiph/16)	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)	fc/2^7 (1.6 μs)
		101 (fperiph/32)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)	fc/2^7 (1.6 μs)	fc/2^8 (3.2 μs)
	101 (fc/4)	000 (fperiph/1)	–	–	fc/2^2 (0.05 μs)	fc/2^3 (0.1 μs)
		001 (fperiph/2)	–	fc/2^2 (0.05 μs)	fc/2^3 (0.1 μs)	fc/2^4 (0.2 μs)
		010 (fperiph/4)	fc/2^2 (0.05 μs)	fc/2^3 (0.1 μs)	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)
		011 (fperiph/8)	fc/2^3 (0.1 μs)	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)
		100 (fperiph/16)	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)	fc/2^7 (1.6 μs)
		101 (fperiph/32)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)	fc/2^7 (1.6 μs)	fc/2^8 (3.2 μs)
	110 (fc/8)	000 (fperiph/1)	–	–	–	fc/2^3 (0.1 μs)
		001 (fperiph/2)	–	–	fc/2^3 (0.1 μs)	fc/2^4 (0.2 μs)
		010 (fperiph/4)	–	fc/2^3 (0.1 μs)	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)
		011 (fperiph/8)	fc/2^3 (0.1 μs)	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)
		100 (fperiph/16)	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)	fc/2^7 (1.6 μs)
		101 (fperiph/32)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)	fc/2^7 (1.6 μs)	fc/2^8 (3.2 μs)
	111 (fc/16)	000 (fperiph/1)	–	–	–	–
		001 (fperiph/2)	–	–	–	fc/2^4 (0.2 μs)
		010 (fperiph/4)	–	–	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)
		011 (fperiph/8)	–	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)
		100 (fperiph/16)	fc/2^4 (0.2 μs)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)	fc/2^7 (1.6 μs)
		101 (fperiph/32)	fc/2^5 (0.4 μs)	fc/2^6 (0.8 μs)	fc/2^7 (1.6 μs)	fc/2^8 (3.2 μs)

Note 1: **Prescaler output clock f φTn must satisfy the condition of φTn <fsys. (φTn must be slower than fsys.)**

Note 2: **In the above table, "−" indicates prohibition.**

Note 3: **Do not change the clock gear during timer in operation.**

21.7.2 Up-Counter (MTUCx)

This counter is a 16-bit binary counter.

- Source clock

  The source clock can be set with MTxIGCR<IGCLK[1:0]>.

  Prescaler output clock can be chosen among φT0, φT1, φT2, φT4
- Start/Stop counter operation

  Counter operation is set with MTxRUN<MTRUN>. When <MTRUN>="1" is set, counter operation starts. When <MTRUN>="1" is set, the counter is stopped and cleared at the same time.

  And when MTxIGRESTA<IGRESTA>="1" is set, counter is cleared and started count-up from zero.
- Counter clear timing

  1. Comparing a match

     The counter is cleared when a value of up-counter (MTUCx) is match with MTxIGRG4.

  2. Counter stopping

     If Mx0RUN<MTRUN>="0" is set, the counter is stopped and cleared.

  3. Counter restarts

     If MTxIGRESTA<IGRESTA>="1" is set, the counter is cleared and counted-up from 0.

  4. In trigger start mode

     In trigger start mode, the counter is stopped and cleared when MTxIN pin becomes the stopping to clear level.

- Count-up & clear operation

  Count-up & clear operation and setting cycle are described in the two cases respectively; one is the case that φT0 is chosen as a source clock, the other case is that φT1, φT2 or φT4 is chosen as a source clock.

  1. φT0 is selected as a source clock

     When φT0 is selected as a source clock, two source clocks are required for match counting and clear counting, so that setting cycle is M+1.

  2. φT1, φT2 or φT4 is selected as a source clock

     When φT1, φT2 or φT4 is selected as a source clock, one source clock is required for match counting and clear counting, so that setting cycle is M.

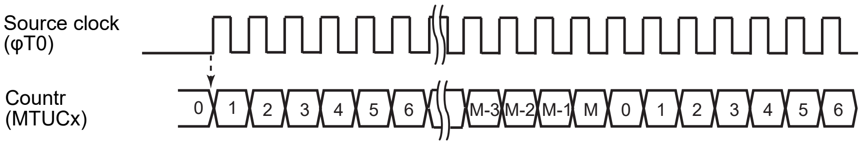


Figure 21-4 Count-up/clear operation when φT0 is selected as a source clock

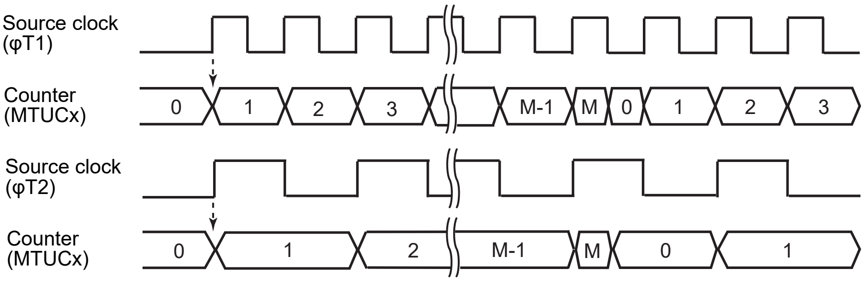


Figure 21-5 Count-up/clear operation when φT1, φT2 or φT4 is selected as a source clock

21.7.3 Cycle Setting Register (MTxIGRG4)

This register sets the cycle of PPG output consisting of double-buffering structure. Data update timing is one cycle after when MTxIGRG4 matches with up-counter MTUCx clearing the counter. At this time, data transfer is taken place from register buffer 4 to timer register MTxIGRG4.

21.7.4 Timer register (MTxRG0, MTxRG1, MTxIGRG2, MTxIGRG3, MTxIGRG4)

This register sets a value to compare with up-counter MTUCx. When these are matched, the match detect signal is output. Timer register, MTxRG0/1, MTxIGRG2/3 are double-buffering structure paired with each register buffer. When MTxIGRG4 matches with up-counter MTUCx, the counter is cleared and data is updated at the same time. Also at this time, data transfer is taken place from register buffer 2/3 to timer register MTxIGRG2/3.

In IGBT mode, MTxRG0/1 is always double-buffering structure.

- Write/read operation of timer registers (MTxRG0, MTxRG1, MTxIGRG2 and MTxIGRG3) and cycle register (MTxIGRG4)

  1. Write

     When timer is stopping, above registers can be written directly. In timer in operation, data is latched in each register. When MTxIGRG4 matches with up-counter MTUCx, the counter is cleared and data is updated at the same time.

  2. Read

     Read the current value of target register comparing with 16-bit comparator. A value of register buffer cannot be read.

Note: **Use half-word access or word access.**

21.7.5 Capture Control

If command start or capture mode is set, this circuit captures up-counter values (MTUCx) at the.risinges and falling edges of MTxIN to MTxCP0 and MTxCP1 respectively.

21.7.6 Capture Register (MTxCAP0, MTxCAP1)

This register captures a value of up-counter MTUCx.

21.7.7 Comparators (CP0, CP1, CP2, CP3, CP4)

This comparator detects the match comparing a value of up-counter (MTUCx) with a setting value of timer register MTxRG0, MTxRG1, MTxIGRG2, MTxIGRG3 and MTxIGRG4.

21.7.8 MTOUT0x, MTOUT1x Output Control

When up-counter matches with timer register, MTOUT0x or MTOUT1x is output.

Initial setting of output pin is set with MTxIGOCR<IGPOL0,1>. After reset, initial state is low. When MTxIGOCR<IGPOL0,1>=0 is set, initial state is low. When MTxIGOCR<IGPOL0,1>=1 is set, initial state is high. Output control is set with MTxIGOCR<IGOEN0,1>. After reset, MTxIGOCR<IGOEN0,1> is disabled. If MTxIGOCR<IGOEN0,1> is enabled, set to 1.

21.7.9 Capture Interrupts (INTMTCAPx0,INTMTCAPx1)

Capture interrupts (INTMTCAPx0 and INTMTCAPx1) occur respectively when each capture register (MTxCP0 and MTxCP1) latches data. Interrupt setting is set by CPU.

21.7.10 Trigger Start Interrupt (INTMTTBx1)

When command start & trigger start mode or trigger start mode is chosen, trigger interrupt occurs when the edge specified with MTxIGCR<IGTRGSEL> is input and the counter starts. In the trigger capture mode, INTMTTBx1 interrupt does not generate at the trigger edge. When emergency output is stopping, a start trigger interrupt occurs.

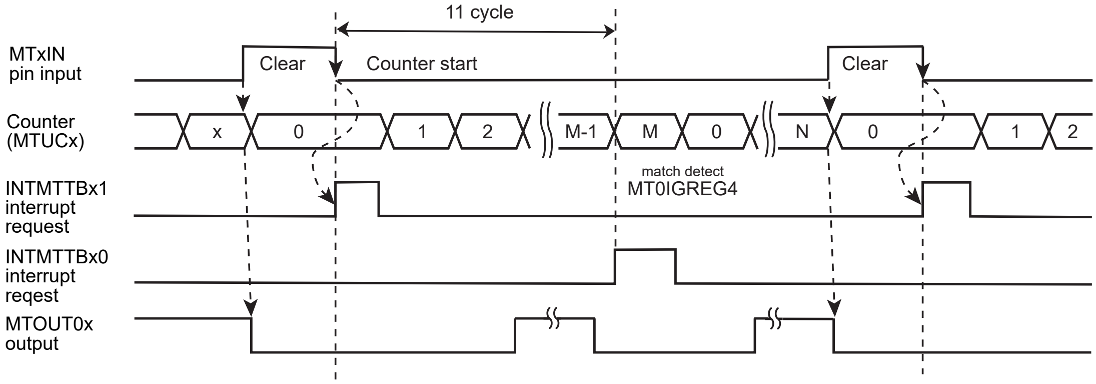


Figure 21-6 Trigger start interrupt operation

21.7.11 Cycle Interrupt (INTMTTBx0)

When command start & trigger capture mode or command start & trigger start mode is chosen, a cycle interrupt occurs when count starts in command start or counter reaches to a value of counter cycle setting (MTxIGREG4) (cycle finishes by matching with a value of cycle setting). Also, a cycle interrupt occurs by matching with a value of counter cycle when emergency output is stopping. Interrupt cycle can be set to among every one cycle, every two cycles or every four cycles with MTxIGCR<IGPRD[1:0]>.

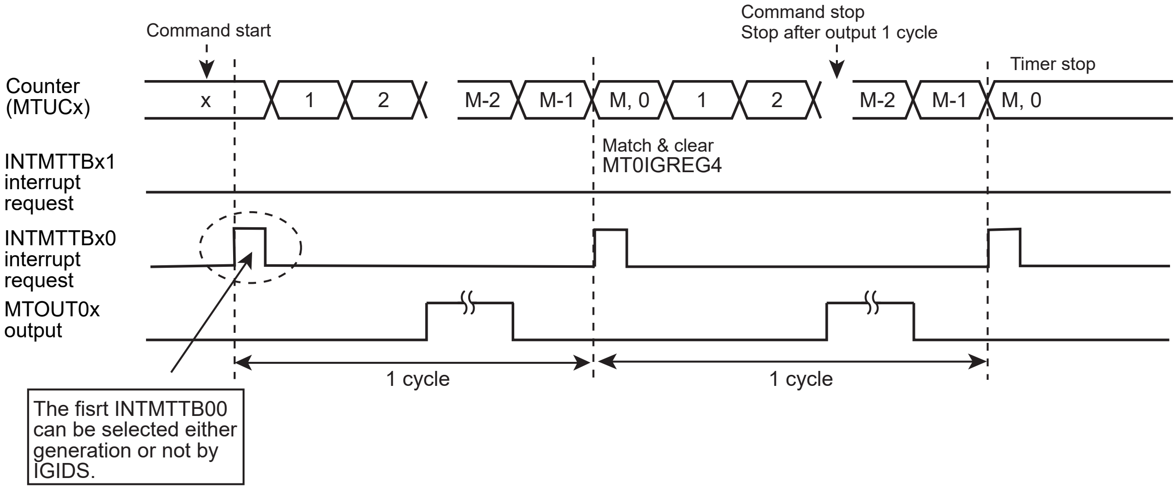

Command start
Command stop
Stop after output 1 cycle
Timer stop
Counter
(MTUCx)
x
1
2
M-2
M-1
M, 0
1
2
M-2
M-1
M, 0
Match & clear
MT0IGREG4
INTMTTBx1
interrupt
request
INTMTTBx0
interrupt
request
MTOUT0x
output
1 cycle
1 cycle
The fisrt INTMTTB00
can be selected either
generation or not by
IGIDS.

Figure 21-7 Cycle interrupt operation

In command start, a cycle interrupt at the starting count is set to enable/disable with interrupt control register MTxIGCR<IGIDIS>. At starting command (MTxRUN<MTRUN> is set to "1"), if MTxIN pin is stopping level, counting does not start (INTMTTBx0 does not occur). Counting starts by trigger start edge and NTMTTBx1 occurs.

21.7.12 Basic Operation

Each MTOUT0x pin and MTOUT1x pin output PPG.

This circuit controls waveform by comparing data set in the timer register (MTxRG0/1, MTxIGRG2/3/4) with a value of 16-bit up-counter.

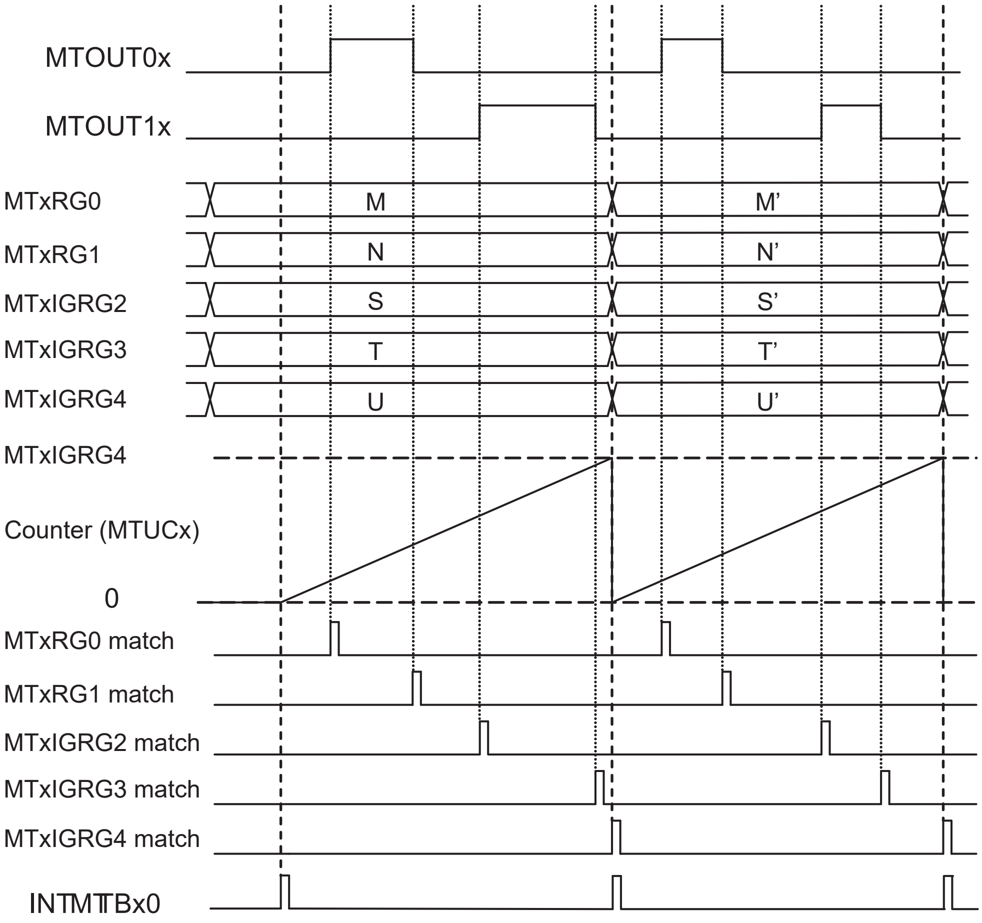

MTOUT0x
MTOUT1x
MTxRG0
M
M'
MTxRG1
N
N'
MTxIGRG2
S
S'
MTxIGRG3
T
T'
MTxIGRG4
U
U'
MTxIGRG4
Counter (MTUCx)
0
MTxRG0 match
MTxRG1 match
MTxIGRG2 match
MTxIGRG3 match
MTxIGRG4 match
INTMTTBx0

Figure 21-8 IGBT mode basic timing

21.7.13 Start modes

In IGBT mode, three start modes are available.

21.7.13.1 Command Start & Trigger Capture Mode

When MTxRUN<MTRUN> is set to "1", counting-up starts. If the counter reaches to the setting cycle, the counter is cleared. At this time, continuous mode is set with MTxIGCR<IGSNGL>, count-up starts again. If single mode is set, counting stops.

If MTxIGRESTA<IGRESTA> is set to "1" before reaching to the setting cycle, counter is cleared at this point and count-up continues.

Counter value at the rising edge/falling edge of MTxIN input can be stored to capture registers.

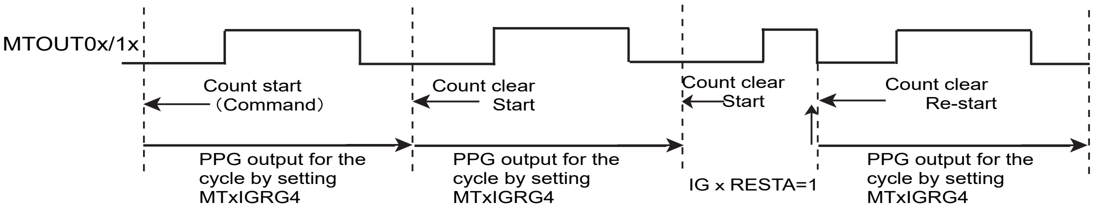

MTOUT0x/1x
Count start
（Command）
Count clear
Start
Count clear
Start
Count clear
Re-start
PPG output for the cycle by setting MTxIGRG4
PPG output for the cycle by setting MTxIGRG4
IG x RESTA=1
PPG output for the cycle by setting MTxIGRG4

Figure 21-9 Continuous mode in command start

In the command start & trigger capture mode, when the counter starts, a counter value is captured at the each rising/falling edge of MTxIN input to each capture register (MTxCAP0 and MTxCAP1).When capture operation is done, INTMTCAPx0 and INTMTCAPx1 occur at each edge.

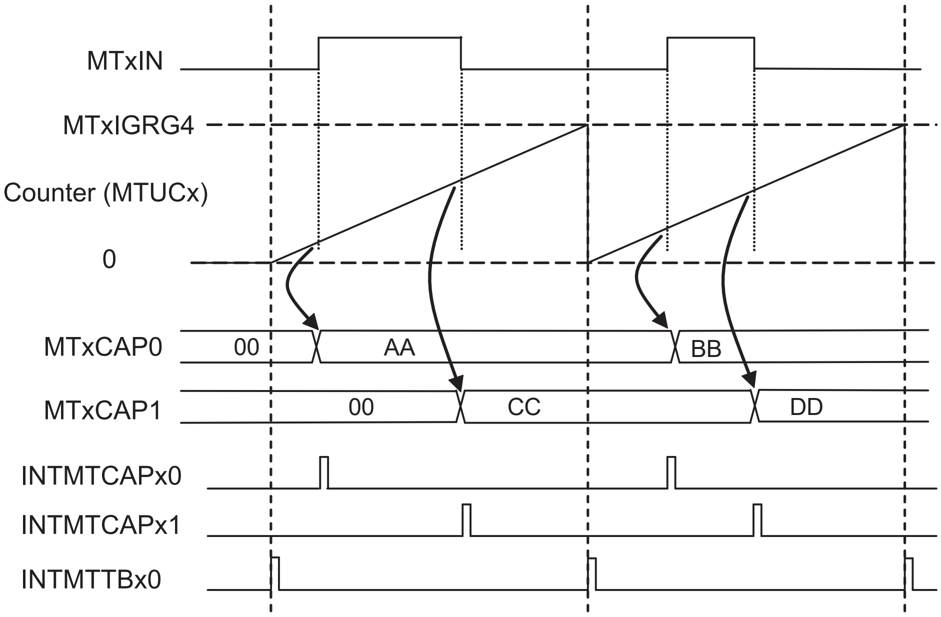

MTxIN
MTxIGRG4
Counter (MTUCx)
0
MTxCAP0
00
AA
BB
MTxCAP1
00
CC
DD
INTMTCAPx0
INTMTCAPx1
INTMTTBx0

Figure 21-10 Capture operation

21.7.13.2 Command Start & Trigger Start Mode

When MTxRUN<MTRUN> is set to "1", count-up starts. If there is no trigger inputs to MTxIN input, same operation previously described in command start & capture mode starts. If an edge input specified with MTxIGICR<IGTRGSEL> to MTxIN pin exists, timer counting starts. While specified clear stopping level is input, the counter is not cleared. At the starting command (when MTxRUN<MTRUN> is set to "1"), if MTxIN pin is in the stopping level, the counter does not start (INTMTTBx1 does not generate). Counting starts by trigger start edge and NTMTTBx1 occurs. (Trigger input is prior to command start.)

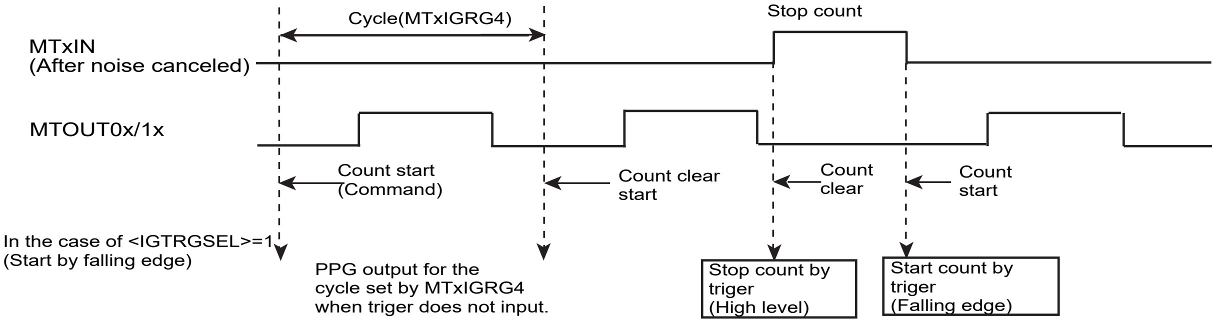

Cycle(MTxIGRG4)
Stop count
MTxIN
(After noise canceled)
MTOUT0x/1x
Count start
(Command)
Count clear
start
Count
clear
Count
start
In the case of <IGTRGSEL>=1
(Start by falling edge)
PPG output for the
cycle set by MTxIGRG4
when triger does not input.
Stop count by
triger
(High level)
Start count by
triger
(Falling edge)

Figure 21-11 Command start & trigger start

21.7.13.3 Trigger Start Mode

If an edge input specified with MTxIGICR<IGTRGSEL> exists, timer counting starts.While specified clear stopping level is input, the counter is not cleared.

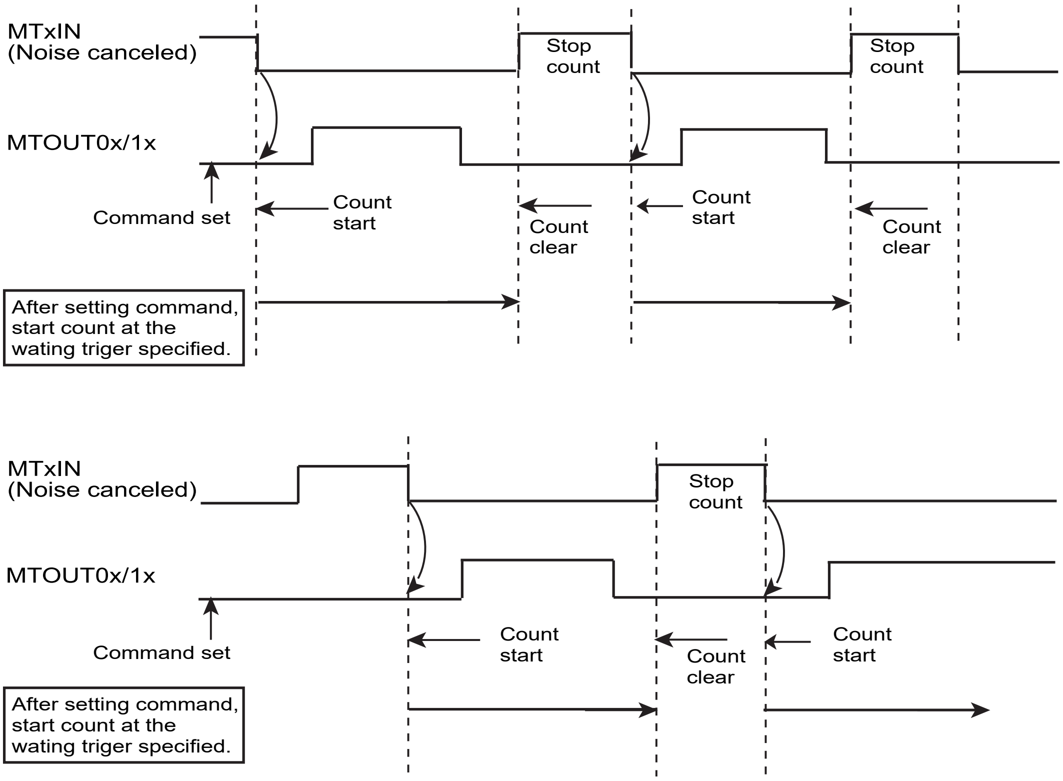

MTxIN
(Noise canceled)
MTOUT0x/1x
Command set
Count start
Stop count
Count clear
Count start
Stop count
Count clear
After setting command, start count at the wating triger specified.
MTxIN
(Noise canceled)
MTOUT0x/1x
Command set
Count start
Stop count
Count clear
Count start
After setting command, start count at the wating triger specified.

Figure 21-12 Trigger Start

21.7.14 Single/Continuous Output Mode

Single/continuous output mode can be set with IGBT output.

21.7.14.1 Continuous Output Mode

At the starting timer (MTxRUN<MTRUN>="1"), if MTxIGCR<IGSNGL>="0" is set, continuous output mode is chosen. In the continuous output mode, specified continuous waveform can be output.

21.7.14.2 Single Output Mode

At the starting timer (MTxRUN<MTRUN>="1"), if MTxIGCR<IGSNGL>="1" is set, single output mode is chosen. In the single output mode, the counter stops after output every single cycle.

At the trigger starting, the counter stops until triggers are input. Counting starts by the specified trigger input, and after one cycle has elapsed, counting stops. If trigger starts again, set MTxRUN<MTRUN>="1".

21.7.15 Stopping Type

By setting "0" to MTxRUN<MTRUN>, outputs and timers stop according to MTxIGCR<IGSTP[1:0]> setting.

21.7.15.1 Counter Stops with Initial State Output

When MTxIGCR<IGSTP[1:0]> is set to "00", the counter immediately stops and MTOUT0x/1x output becomes an initial value set with MTxIGOCR<IGPOL[1:0]>.

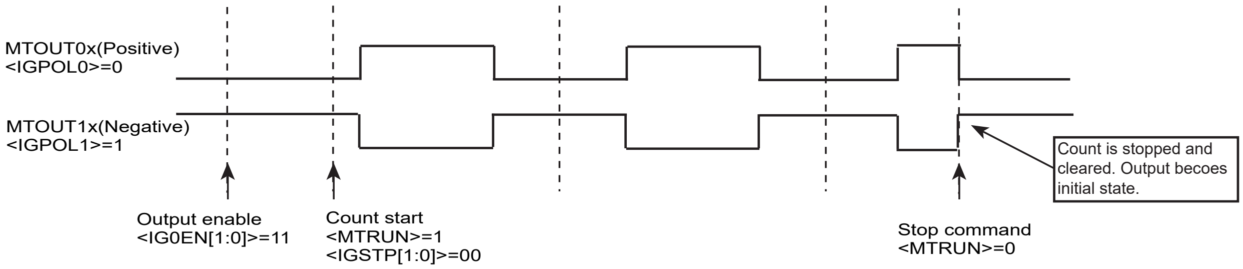

MTOUT0x(Positive)
<IGPOL0>=0
MTOUT1x(Negative)
<IGPOL1>=1
Count is stopped and cleared. Output becoes initial state.
Output enable
<IG0EN[1:0]>=11
Count start
<MTRUN>=1
<IGSTP[1:0]>=00
Stop command
<MTRUN>=0

Figure 21-13 Counter stops with initial state output

21.7.15.2 Counter Stops with maintaining the output status

When <IGSTP[1:0]> is set to “01”, the counter immediately stops and MTOUT0x/1x output is maintained.

If the counter starts again, set MTxRUN<MTRUN>="1". At this time, outputs become an initial value (setting value of <IGPOL0> or <IGPOL1>) and then restarts.

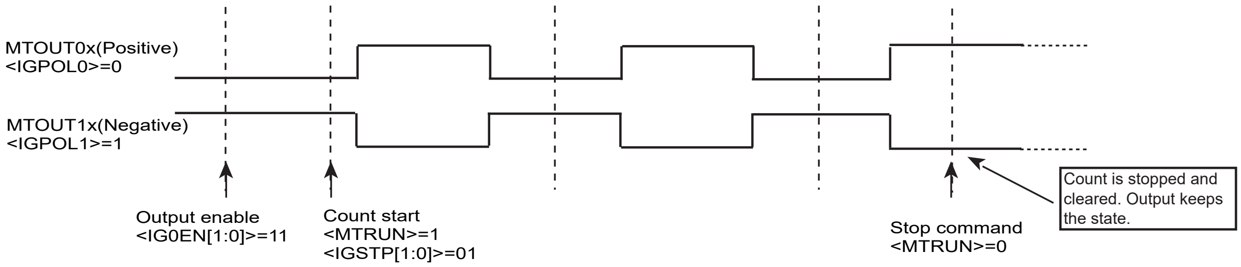

MTOUT0x(Positive)
<IGPOL0>=0
MTOUT1x(Negative)
<IGPOL1>=1
Count is stopped and cleared. Output keeps the state.
Output enable
<IG0EN[1:0]>=11
Count start
<MTRUN>=1
<IGSTP[1:0]>=01
Stop command
<MTRUN>=0

Figure 21-14 Counter stops with maintaining the output status

21.7.15.3 Counter Stops with Initial State after Cycle finished

When <IGSTP[1:0]> is set to "10", the counter operates until the cycle has finished. After cycle has finished, the counter stops. However, if trigger input becomes stop level until the cycle finishes, the counter stops at this point.

If the timer is set again, check if the counter stops after cycle has finished.

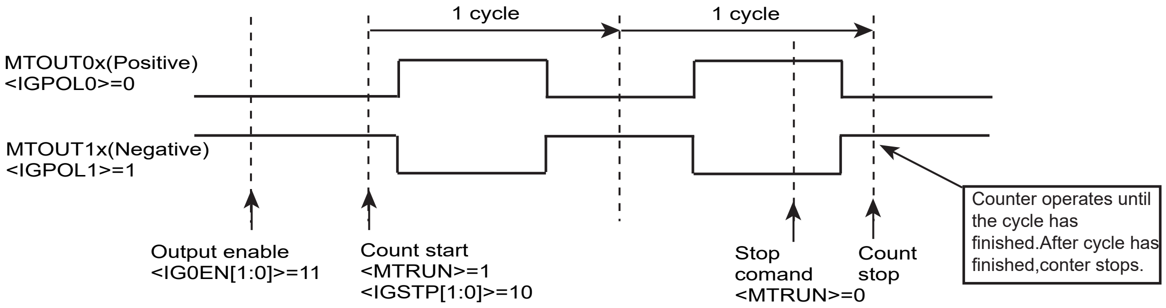

1 cycle
1 cycle
MTOUT0x(Positive)
<IGPOL0>=0
MTOUT1x(Negative)
<IGPOL1>=1
Counter operates until the cycle has finished.After cycle has finished,conter stops.
Output enable
<IG0EN[1:0]>=11
Count start
<MTRUN>=1
<IGSTP[1:0]>=10
Stop comand
<MTRUN>=0
Count stop

Figure 21-15 Counter stops with initial state after cycle has finished

21.7.16 Trigger Input

21.7.16.1 Logic of Trigger Input

The valid condition of MTxIN input is set with MTxIGICR<IGTRGSEL>.

- <IGTRGSEL>=0 : Rising edge detection to start counting

  During "High" level, count-up is performed. During "Low" level, counter stops.
- <IGTRGSEL>=1 : Falling edge detection to start counting

  During "Low" level, count-up is performed. During "High" level, counter stops.

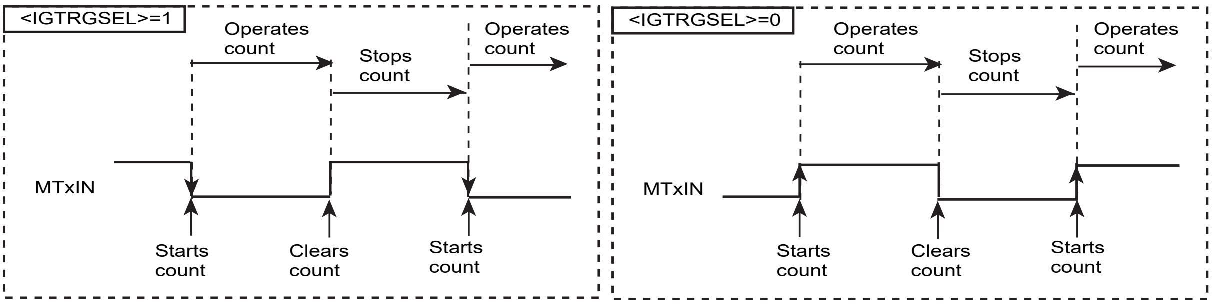

<IGTRGSEL>=1
Operates count
Stops count
Operates count
MTxIN
Starts count
Clears count
Starts count
<IGTRGSEL>=0
Operates count
Stops count
Operates count
MTxIN
Starts count
Clears count
Starts count

Figure 21-16 Logic of trigger input

While cycles are stopping, a stop trigger signal is accepted but a start signal is not. (Once a stop trigger signal is accepted while cycles are stopping, outputs become an initial value then the counter stops.)

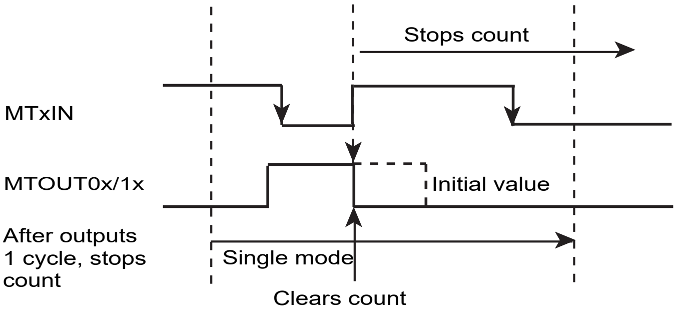

Stops count
MTxIN
MTOUT0x/1x
Initial value
After outputs 1 cycle, stops count
Single mode
Clears count

Figure 21-17 Trigger acceptance while cycles are stopping

21.7.16.2 Trigger Constant Acceptance/prohibit accessing during active level

MTxIGICR<IGTRGM> can choose either condition; one is a trigger from MTxIN is always accepted during PPG output, or another is a trigger is prohibited accessing during PPG output in active. This setting is only valid for enabled pin with MTxIGOCR <IGOEN[1:0]>.

When <IGTRGM>="0" is set, a trigger input from MTxIN is always accepted regardless of MTOUT0x/1x in active/non-active. During this period, timer is started/stopped to clear and MTOUT0x/1x output becomes non-active.

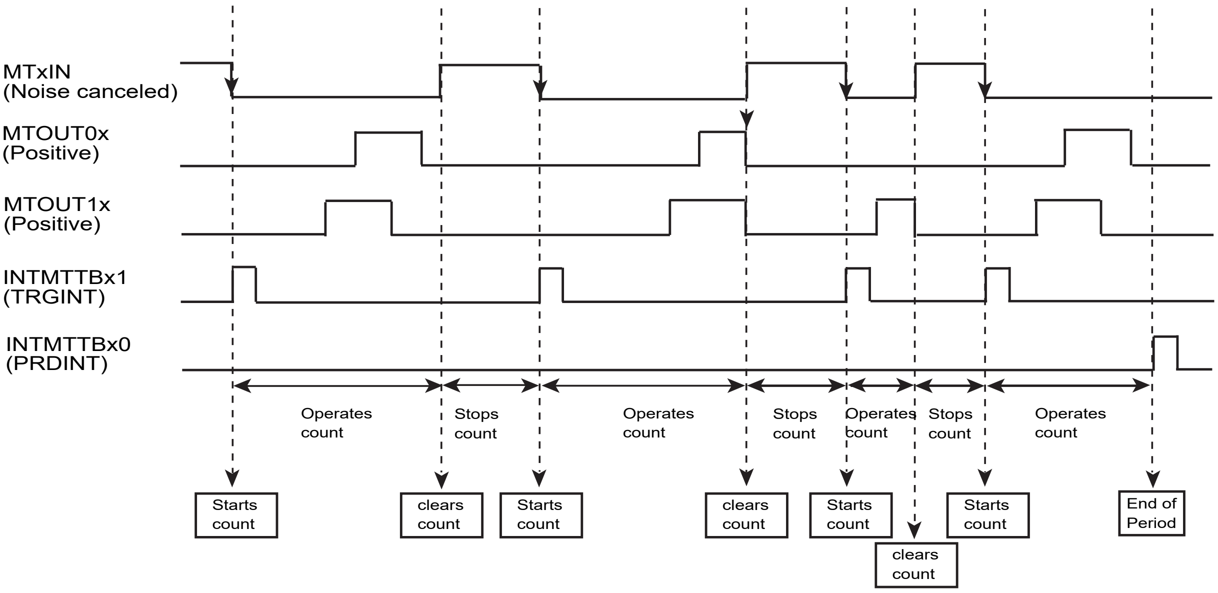

MTxIN
(Noise canceled)
MTOUT0x
(Positive)
MTOUT1x
(Positive)
INTMTTBx1
(TRGINT)
INTMTTBx0
(PRDINT)
Operates count
Stops count
Operates count
Stops count
Operates count
Stops count
Operates count
Starts count
clears count
Starts count
clears count
Starts count
clears count
Starts count
End of Period

Figure 21-18 Trigger constant acceptance

When <IGTRGM>="1" is set, input edge at MTOUT0x/1x output in non-active is accepted and cleared to stop.

If input edge at MTOUT0x/1x output in active, the counter does not immediately stops. It continues to count until MTOUT0x/1x output becomes non-active. When MTOUT0x/1x output is non-active, if trigger signal is not in active level, the counter is cleared to stop and waits next start trigger signal.

If the counter operates when both MTOUT0x and MTOUT1x are enabled, both outputs must be in non-active. Otherwise triggers are not accepted.

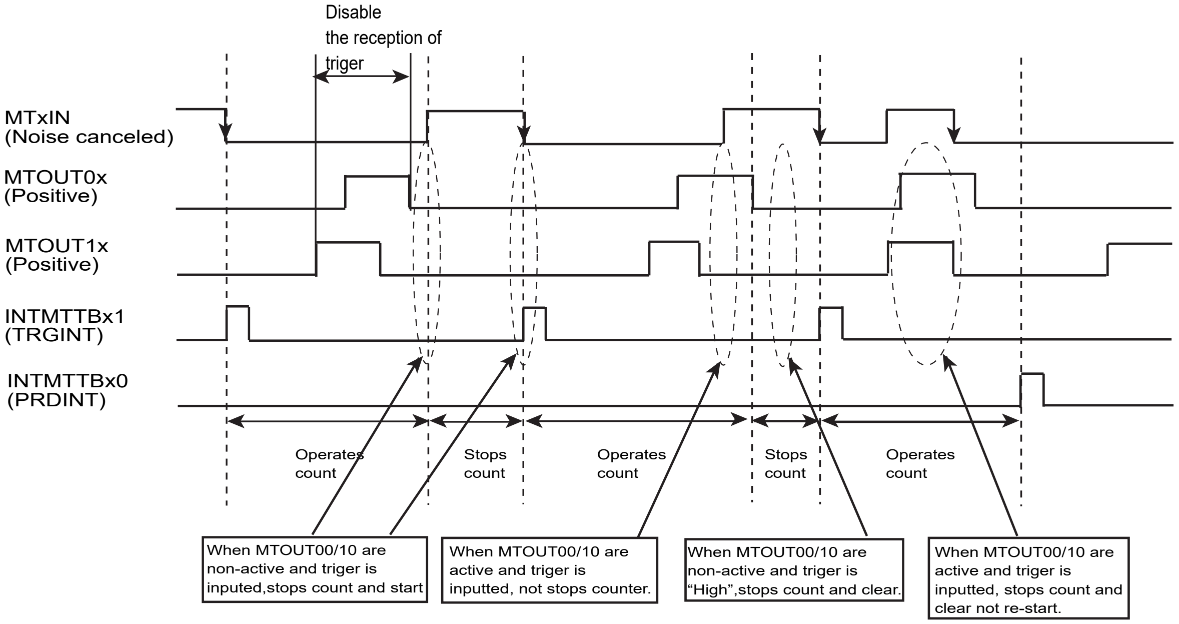

Disable
the reception of
triger
MTxIN
(Noise canceled)
MTOUT0x
(Positive)
MTOUT1x
(Positive)
INTMTTBx1
(TRGINT)
INTMTTBx0
(PRDINT)
Operates count
Stops count
Operates count
Stops count
Operates count
When MTOUT00/10 are non-active and triger is inputed,stops count and start
When MTOUT00/10 are active and triger is inputted, not stops counter.
When MTOUT00/10 are non-active and triger is "High",stops count and clear.
When MTOUT00/10 are active and triger is inputted, stops count and clear not re-start.

Figure 21-19 Prohibit accessing during active level

21.7.17 Emergency Stop Function

21.7.17.1 Operation Description

When MTxIGEMGCR<IGEMGEN>="1" is set, the emergency stop function is enabled (GEMGx pin is enabled to input).

If GEMGx pin detects a low level input, MTOUT0x/ MTOUT1x waveform becomes initial state (set with IGPOL0/IGPOL1) according to MTxIGEMGCR<IGEMGOC> setting or becomes high-impedance and generates a GEMGx interrupt.

This function prohibits only MTOUT0x/ MTOUT1x output. The counter does not stop so that timer must be stopped in the GEMG interrupt service routine.

21.7.17.2 Emergency stop monitor

On the emergency stop condition, MTxIGEMGST<IGEMGST> is set to "1". When IGEMGST is read, "1" indicates of emergency stopping.

21.7.17.3 GEMG interrupts

When an emergency stop input is received, a GEMG interrupt (INTMTEMGx) occurs. If this process uses interrupt service routine, the INTMTEMGx interrupt must be enabled in advance.

If GEMGx pin is "Low" and exits emergency stop status, GEMG interrupt occurs again, and MCU is in emergency stop condition again.

21.7.17.4 Exiting Emergency Stop Condition

When MCU exits emergency stop condition, check if GEMGx input is high and MTxRUN<MTRUN> is set to "0". Then confirm the timer stops (MTxIGST<IGST>=0), later MTxIGEMGCR<IGEMGRS>="1" is set for exiting emergency stop condition.

When MTxIGCR<IGSTP[1:0]>="01" or "10" is set in the stopping type selection register, set the initial setting with MTxIGOCR<IGPOL[1:0]> before writing MTxIGEMGCR<IGEMGRS>="1".

21.7.18 Noise Canceller

The digital noise canceller eliminate noise inputting to external input pins (MTxIN and GEMGx).

It can be chosen the noise elimination time with MTxIGICR<IGNCSEL[3:0]> or MTxIGEMGCR <IGEMGCNT[3:0]> setting.

21.8 Operation Description of Motor Control Circuit (PMD : Programmable Motor Driver)

TMPM367FDFG has one channel of motor control circuits (PMD).

This PMD is added the applying current control and the DC over-voltage detection input to fabricate one-shunt sensor less motor control.

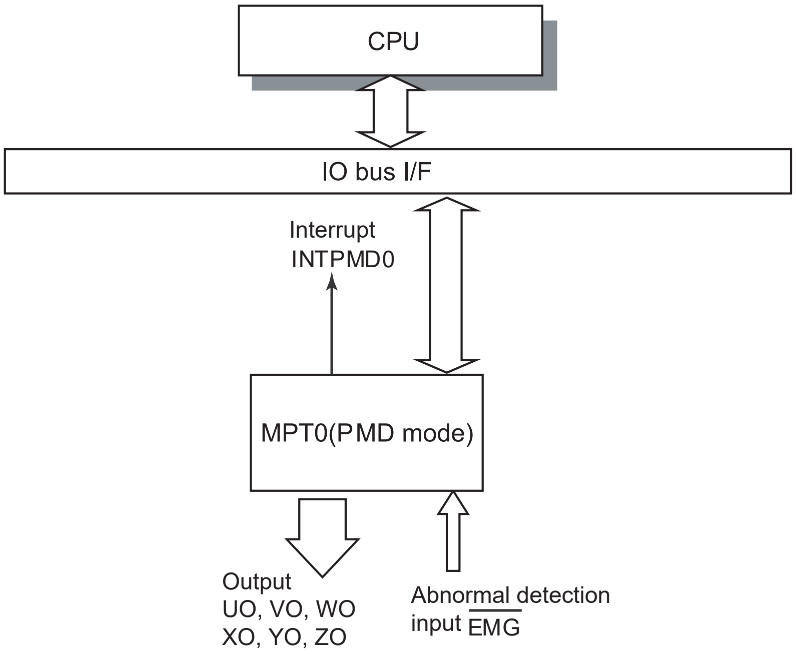

CPU
IO bus I/F
Interrupt
INTPMD0
MPT0(PMD mode)
Output
UO, VO, WO
XO, YO, ZO
Abnormal detection
input $\overline{\text{EMG}}$

Figure 21-20 Block diagram of related with motor control

21.8.1 Input/output signal to PMD circuit

The following table describes input and output signals categorized by channel in motor control circuit.

Table 21-4 Input/output signals

CH	Pin name	**PMD signal name**	**Function**
PMD0	PG7/UO	UO	U-phase output
	PG6/XO	XO	X-phase output
	PG5/VO	VO	V-phase output
	PG4/YO	YO	Y-phase output
	PG3/WO	WO	W-phase output
	PG2/ZO	ZO	Z-phase output
	PG1/$\overline{\text{EMG0}}$	$\overline{\text{EMG}}$	Abnormal detection input signal

21.8.2 Structure

The PMD (programmable motor driver) contains a waveform generation circuit which is comprised of a pulse width modulation circuit, an applying current control circuit, a protection circuit and a dead time control circuit.

- Pulse width modulation circuit generates identical 3-phase independent PWM waveforms.
- Applying current circuit determines each upper/lower output pattern of U-, V-, W-phase.
- Protection control circuit takes place emergency stop by detecting abnormal detect input.
- Dead time control prevent a short circuit at switching upper/lower phase.

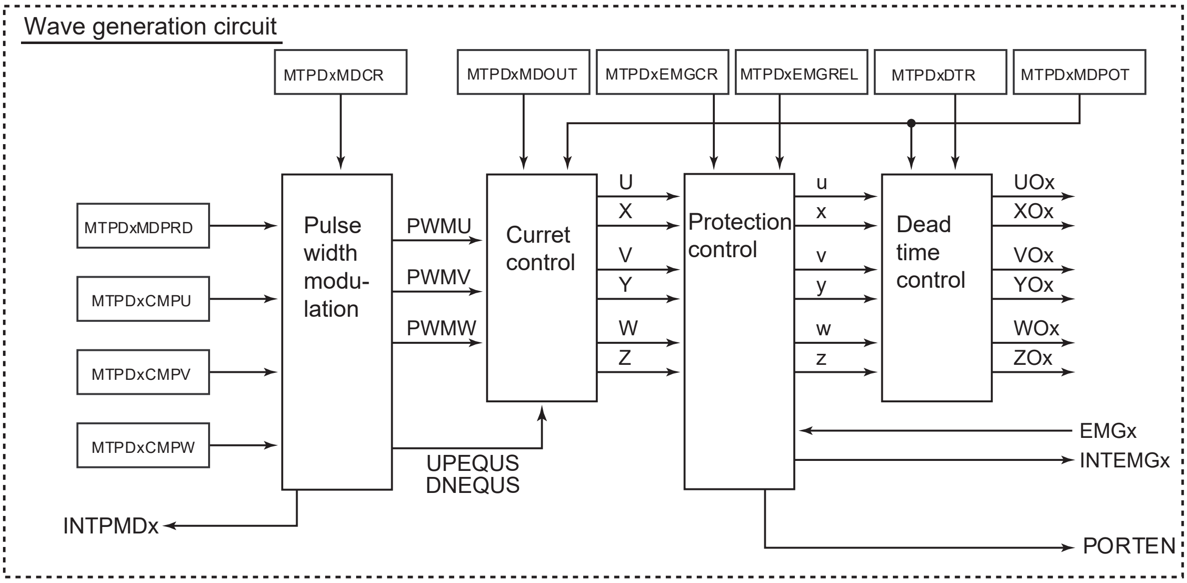

Wave generation circuit
MTPDxMDCR
MTPDxMDOUT
MTPDxEMGCR
MTPDxEMGREL
MTPDxDTR
MTPDxMDPOT
MTPDxMDPRD
MTPDxCMPU
MTPDxCMPV
MTPDxCMPW
Pulse width modu- lation
PWMU
PWMV
PWMW
Curret control
U
X
V
Y
W
Z
Protection control
u
x
v
y
w
z
Dead time control
UOx
XOx
VOx
YOx
WOx
ZOx
EMGx
INTEMGx
UPEQUS
DNEQUS
INTPMDx
PORTEN

Figure 21-21 Schematic diagram of PMD circuit

21.8.3 Registers

21.8.3.1 Registers categorized by channel

This section describes registers and addresses of each channel.

Base address = 0x400F_6000

Register name		Address (Base+)
PMD enable register	MTPDMDEN	0x0000
Port output mode register	MTPDPORTMD	0x0004
PMD control register	MTPDMDCR	0x0008
PWM count status register	MTPDCNTSTA	0x000C
PWM count register	MTPDMDCNT	0x0010
PWM cycle register	MTPDMDPRD	0x0014
PWM compare U register	MTPDCMPU	0x0018
PWM compare V register	MTPDCMPV	0x001C
PWM compare W register	MTPDCMPW	0x0020
Reserved	−	0x0024
PMD output control register	MTPDMDOUT	0x0028
PMD output setting register	MTPDMDPOT	0x002C
EMG release register	MTPDEMGREL	0x0030
EMG control register	MTPDEMGCR	0x0034
EMG status register	MTPDEMGSTA	0x0038
Reserved	−	0x003C
Reserved	−	0x0040
Dead time register	MTPDDTR	0x0044
Reserved	−	0x0048
Reserved	−	0x004C
Reserved	−	0x0050
Reserved	−	0x0054
Reserved	−	0x0058
Reserved	−	0x005C
Reserved	−	0x0060
Reserved	−	0x007C

21.8.3.2 MTPDMDEN (PMD 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	-	-	-	-	-	-	-	PWMEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as "0".
0	PWMEN	R/W	Controls waveform generation to enable/disable 0: Disabled 1: Enabled While ports are set to PWM output, if <PWMEN>="0"(disable)is set, output ports become high-impedance. Initial settings other than <PWMEN> such as output port polarity must be done before <PWMEN>="1" (Enable) is set.

21.8.3.3 MTPDPORTMD (Port output 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	-	-	-	-	-	-	-	PORTMD
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 "0".
0	PORTMD	R/W	Sets port control 0: High-impedance 1: PMD output Six output ports with all phase output control signals to external port by setting <PORTMD>. If tool break occurs when high-impedance is chosen, external output port becomes high-impedance. Otherwise PMD output is set. Note 1) When MTPDMDEN<PWMEN>=0 is set, high-impedance output is set regardless of output port setting. Note 2) External port output control can be done according to PMDxEMGMD setting, even when EMG input.

21.8.3.4 MTPDMDCR (PMD 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	-	PWMCK	SYNTMD	DTYMD	PINT	INTPRD		PWMMD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	-	R	Read as "0".
6	PWMCK	R/W	Specifies PWM cycle extension mode 0: Normal cycle 1: 4-fold cycle In the normal setting, PWM counter operates at a resolution of 12.5ns@fsys=80MHz. *Sawtooth wave 12.5ns, Triangle wave 25ns In the 4-fold cycle setting, PWM counter operates at a resolution of 50ns@2-bit counter (fsys=80MHz) *Sawtooth wave 50ns, Triangle wave 100ns
5	SYNTMD	R/W	Sets port output of U-, V- and W-phases.Å¶Refer toTable 21-6.
4	DTYMD	R/W	Chooses DUTY mode 0: U-phase in common 1: 3-phase independent Chooses duty setting among either each 3-phase (PMDxCMPU, V and W) is independent or PMDxCMPU register in each 3-phase is used in common.
3	PINT	R/W	Chooses PWM interrupt timing when PWM mode 1 (triangle wave) is set. 0: When PWM count MDCNT="1" is set, (minimum) interrupt request occurs. 1:When PWM count MDCNT=MTPDMDPRD<MDPRD> is set, (maximum) interrupt request occurs. User can be choose the interrupt generation timing in the PWM mode 1 (triangle wave) either when PWM counter MDCNT becomes "1" (minimum) or when PWM counter becomes MTPDMDPRD<MDPRD> (maximum). If PWM interrupt cycle is set as 0.5 cycle with <INTPRD>, PWM interrupt occurs both when PWM counter MDCNT becomes "1" (minimum) and <MDPRD>(maximum) regardless of this register. Also in PWM mode 0 (sawtooth wave), PWM interrupt occurs when PWM counter MDCNT becomes <MDPRD> (maximum) regardless of this register.
2-1	INTPRD	R/W	Chooses PWM interrupt cycle 00: Every PWM 0.5 cycles (can be set in PWM mode1 (triangle wave) 01: Every PWM 1 cycle 10: Every PWM 2 cycles 11: Every PWM 4 cycles Frequency of PWM interrupt generation can be chosen among every 0.5 cycles, 1 cycles, 2 cycles or 4cycles.
0	PWMMD	R/W	Specifies PWM carrier wave 0: PWM mode 0 (edge PWM, sawtooth) 1: PWM mode 1 (center PWM, triangle wave)

21.8.3.5 MTPDCNTSTA (PWM count 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	-	-	-	-	-	-	-	UPDWN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as "0".
0	UPDWN	R	PWM counter flag 0: Up-counting 1: Down-counting Indicate PWM counter is up-counting or down-counting. When PWM mode 0 (sawtooth) is chosen, always read "0".

21.8.3.6 MTPDMDCNT (PWM count 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	MDCNT							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MDCNT							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MDCNT	R	16-bit read-only register for counting PWM cycles. When MTPDMDCR<PWMCK>="0" is set, PWM counter resolution is 12.5ns@fsys=80MHz in PWM mode 0 (sawtooth) or 25ns@fsys=80MHz in PWM mode 1 (triangle wave). When <PWMCK>="1" is set, PWM counter resolution is 50ns@fsys=80MHz in PWM mode 0 (sawtooth) or 100ns@fsys=80MHz in PWM mode 1 (triangle).

21.8.3.7 MTPDMDPRD (PWM cycle 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	MDPRD							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MDPRD							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MDPRD	R/W	Sets PWM cycles. PWM counter resolution is 12.5ns@fsys=80MHz in PWM mode 0 (sawtooth) or 25ns@fsys=80MHz in PWM mode 1 (triangle wave). When MTPDMDCR<PWMCK>="1" is set, PWM counter resolution is 50ns@fsys=80MHz in PWM mode 0 (sawtooth) or 100ns@fsys=80MHz in PWM mode 1 (triangle wave). <MDPRD> is a PWM cycle setting register with double-buffering structure. Thus even if PWM counter is operating, it can be changed. Transfer timing from register to latch circuit is when PWM counter MDCNT matches with MTPDMDPRD <MDPRD>. If interrupt timing is set to 0.5 cycles (MTPDMCR<INTPRD>="00"), the transfer is taken place when PWM counter MDCNT is set to "1" or matches with <MDPRD>. Sets a value of 0x10 or more to <MDPRD>. Even if a value less than 0x10 is set, the register assumes 0x10 is set. (User specified value is set in the register.) Read a value of register (data set from bus) when read.

Note: **Use half-word access or word access.**

21.8.3.8 MTPDCMPU (PWM compare 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	CMPU							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMPU							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	CMPU	R/W	Controls PWM pulse width Resolutions are 12.5ns@fsys=80MHz in PWM mode 0 (sawtooth) and 25ns@fsys=80MHz in PWM mode 1 (triangle wave). When MTPDMDCR<PWMCK> is set to "1", a resolution is set to 50ns@fsys=80MHz in PWM mode 0 (sawtooth) or 100ns@fsys=80MHz in PWM mode 1 (triangle). <CMPU> is the compare register that determines the pulse width outputting to U-phase. This register compares a pulse large or small with PWM counter MDCNT to determine the pulse width. <CMPU> is double-buffering structure, so that even if PWM counter is operating, it can be changed. Transfer timing from buffers to registers is when PWM counter MDCNT matches with MTPDMDPRD <MDPRD>. If interrupt timing is set to 0.5 cycles (MTPDMCR <INTPRD>="00"), the transfer is taken place when PWM counter MDCNT is set to "1" or matches with <MDPRD>. Read a value of buffer (data set from bus) when read.

Note: **Use half-word access or word access.**

21.8.3.9 MTPDCMPV (PWM compare 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	CMPV							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMPV							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	CMPV	R/W	Sets PWM pulse width Resolutions are 12.5ns@fsys=80MHz in PWM mode 0 (sawtooth) and 25ns@fsys=80MHz in PWM mode 1 (triangle wave). When MTPDMDCR<PWMCK> is set to "1", a resolution is set to 50ns@fsys=80MHz in PWM mode 0 (sawtooth) or 100ns@fsys=80MHz in PWM mode 1 (triangle). <CMPV> is the compare register that determines the pulse width outputting to U-phase. This register compares a pulse large or small with PWM counter MDCNT to determine the pulse width. <CMPV> is double-buffering structure, so that even if PWM counter is operating, it can be changed. Transfer timing from buffers to registers is when PWM counter MDCNT matches with MTPDMDPRD <MDPRD>. If interrupt timing is set to 0.5 cycles (MTPDMCR <INTPRD>="00"), the transfer is taken place when PWM counter MDCNT is set to "1" or matches with <MDPRD>. Read a value of buffer (data set from bus) when read.

Note: **Use half-word access or word access.**

21.8.3.10 MTPDCMPW (PWM compare 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	CMPW							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMPW							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	CMPW	R/W	Sets PWM pulse width Resolutions are 12.5ns@fsys=80MHz in PWM mode 0 (sawtooth) and 25ns@fsys=80MHz in PWM mode 1 (triangle wave). When MTPDMDCR<PWMCK> is set to "1", a resolution is set to 50ns@fsys=80MHz in PWM mode 0 (sawtooth) or 100ns@fsys=80MHz in PWM mode 1 (triangle). <CMPW> is the compare register that determines the pulse width outputting to W-phase. This register compares a pulse large or small with PWM counter MDCNT to determine the pulse width. <CMPW> is double-buffering structure, so that even if PWM counter is operating, it can be changed. Transfer timing from buffers to registers is when PWM counter MDCNT matches with MTPDMDPRD <MDPRD>. If interrupt timing is set to 0.5 cycles (MTPDMCR <INTPRD>="00"), the transfer is taken place when PWM counter MDCNT is set to "1" or matches with <MDPRD>. Read a value of buffer (data set from bus) when read.

Note: **Use half-word access or word access.**

21.8.3.11 MTPDMDOUT (PMD output control register)

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

Bit	Bit Symbol	Type	Function
31-11	-	R	Read as "0".
10	WPWM	R/W	Controls U-, V- and W-phase outputs 0: H/L output 1: PWM output For details, refer to Table 21-6.
9	VPWN	R/W	
8	UPWN	R/W	
7-6	-	R	Read as "0".
5-4	WOC[1:0]	R/W	Controls U-, V- and W-phase outputs For details, refer to Table 21-6.
3-2	VOC[1:0]	R/W	
1-0	UOC[1:0]	R/W	

Note: **Use half-word access or word access.**

21.8.3.12 MTPDMDPOT (PMD output 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	-	-	-	-	POLH	POLL	PSYNCS	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	-	R	Read as "0".
3	POLH	R/W	Chooses polarity of upper phase output port 0: Low active 1: High active *Chooses the setting when MTPDMDEN<PWMEN>=0 is set.
2	POLL	R/W	Choose polarity of lower phase output port 0: Low active 1: High active *Chooses the setting when MTPDMDEN<PWMEN>=0 is set.
1-0	PSYNCS	R/W	Chooses the timing when port outputs of U-, V- and W-phase output setting is reflected. 00: Reflects when write 01: Reflects when PWM counter MDCNT="1"(minimum) 10: Reflects when PWM counter MDCNT=MTPDMDPRD<MDPRD>(maximum) 11: Reflects when PWM counter MDCNT="1"(minimum) and MTPDMDPRD<MDPRD>(maximum) *Chooses the setting when MTPDMDEN<PWMEN>="0" is set.

21.8.3.13 MTPDEMGREL (EMG release 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	EMGREL							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-0	EMGREL[7:0]	W	Writes EMG prohibition code To prohibit EMG function, set the procedure as follows; set "0x5A"→"0xA5" to <EMGREL[7:0]>, then set "0" to MTPDEMGCR<EMGEN>.

Note: **To prohibit EMG function, three instructions must be executed consecutively. Three instructions are as follows; set "0x5A", change to "0xA5" and set "0" to MTPDEMGCR<EMGEN>.**

21.8.3.14 MTPDEMGCR (EMG 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	-	-	-	-	EMGCNT			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	INHEN	EMGMD		-	EMGRS	EMGEN
After reset	0	0	1	1	1	0	0	1

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as "0".
11-8	EMGCNT[3:0]	R/W	Set the noise elimination time for abnormal condition detection input Noise elimination time is calculated with the following formula: EMGCNT[3:0]×16/fsys 0000: Noise filter is not used. 0001: Input noise elimination time 16 / fsys [s] 0010: Input noise elimination time 32 / fsys [s] 0011: Input noise elimination time 48 / fsys [s] 0100: Input noise elimination time 64 / fsys [s] 0101: Input noise elimination time 80 / fsys [s] 0110: Input noise elimination time 96 / fsys [s] 0111: Input noise elimination time 112 / fsys [s] 1000: Input noise elimination time 128 / fsys [s] 1001: Input noise elimination time 144 / fsys [s] 1010: Input noise elimination time 160 / fsys [s] 1011: Input noise elimination time 176 / fsys [s] 1100: Input noise elimination time 192 / fsys [s] 1101: Input noise elimination time 208 / fsys [s] 1110: Input noise elimination time 224 / fsys [s] 1111: Input noise elimination time 240 / fsys [s]
7-6	-	R	Read as "0".
5	INHEN	R/W	Chooses a PMD output status at tool break 0: PMD output is continued 1: High-impedance Initial state is high-impedance.
4-3	EMGMD	R/W	EMG protection mode selection 00: All phase are on/PORT output and high-impedance 01: All phase are off/PORT output and high-impedance 10: All phase are on/PORT output is enabled. 11: All phase are off/PORT output is high-impedance. *On=PWM output (No output control), Off=Low (@high-active (POLL/H=1)) Upon EMG occurred, this bit controls that five PWM outputs in all phase (upper and lower) are On/Off. Also, it controls that PORT output is disabled/enabled when EMG occurred.
2	-	R/W	Write "0".
1	EMGRS	R/W	Returns from EMG protection status 0: − 1: Returns from protection status When after MTPDMDOUT<WPWM><VPWM><UPWM><WOC[1:0]><VOC[1:0]><UOC[1:0]> are set to 0, MCU returns from EMG protection status by setting <EMGRS> to "1". Always reads as "0".
0	EMGEN	R/W	Sets EMG protection circuit to disable/enable 0: Disabled 1: Enabled When this bit is set to "1", EMG protection circuit operates. Initial state is enabled. To disable EMG protection circuit, set as follows; set "0x5A"→"0xA5" to MTPDEMGREL<EMGREL>in order, then set "0" to <EMGEN>.(These three instruction must be executed consecutively.)

Note: **When returning from EMG protection status by setting MTPDEMG<EMGRS>, read MTPDEMGSTA<EMGI> to confirm if abnormal detection input level is "H".**

21.8.3.15 MTPDEMGSTA (EMG 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	-	-	-	-	-	-	EMGI	EMGST
After reset	0	0	0	0	0	0	-	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1	EMGI	R	Monitors the level of abnormal condition input 0: Abnormal condition input level is "L" 1: Abnormal condition input level is "H"
0	EMGST	R	Monitors EMG protection condition 0: Normal operation 1: During in EMG protection

21.8.3.16 MTPDDTR (Dead Time 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	DTR							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-0	DTR[7:0]	R/W	Sets dead time Dead time is calculated with the following formula: 100nsec x <DTR[7:0]> (fsys=80MHz)

Note: **Do not modify MTPDDTR<DTR[7:0]> when MTPDMDEN<PWMEN>="1" is set.**

21.9 Operation Description categorized by Circuit

21.9.1 Pulse Width Modulation Circuit

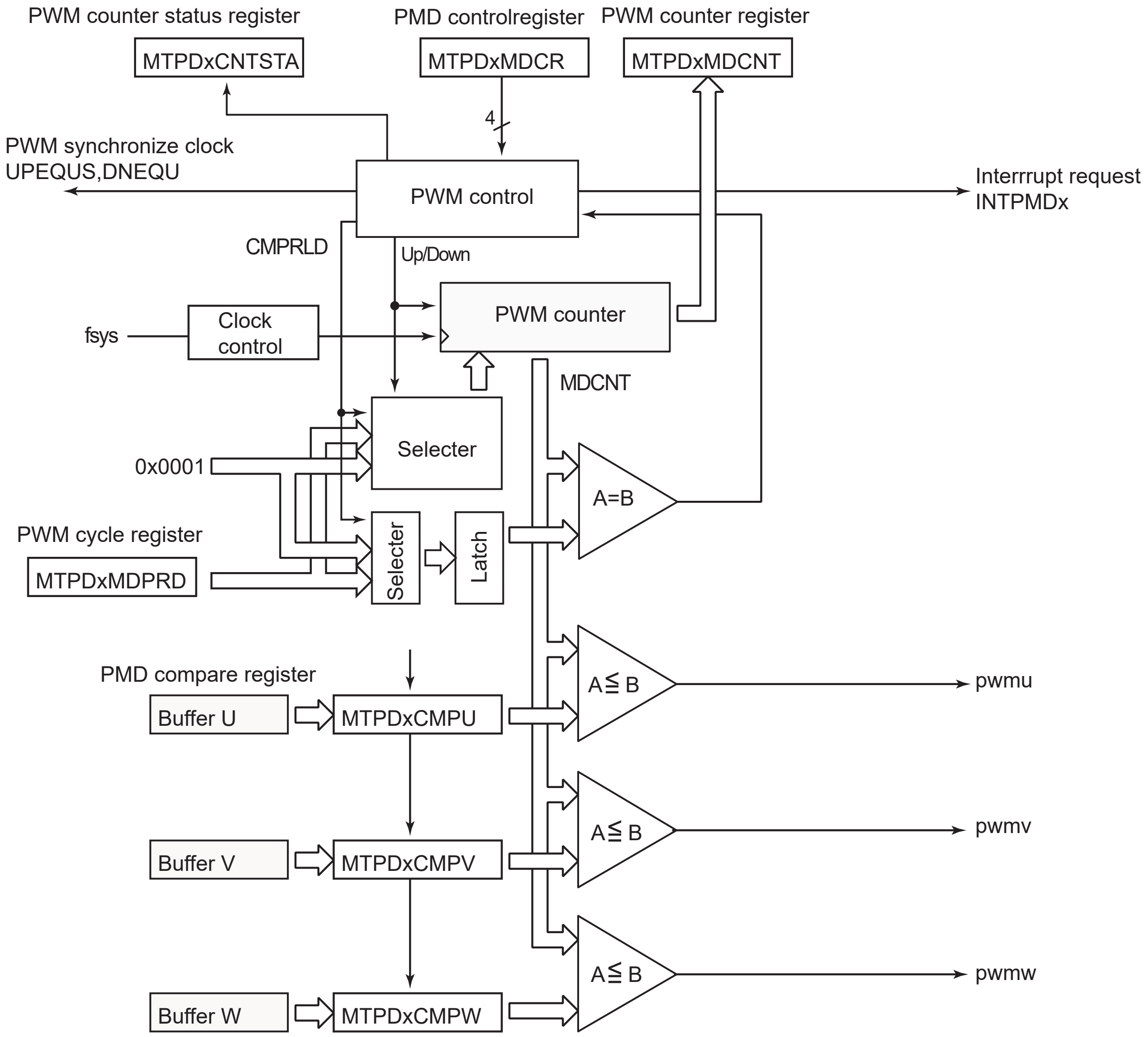

PWM counter status register
MTPDxCNTSTA
PMD controlregister
MTPDxMDCR
PWM counter register
MTPDxMDCNT
4
PWM synchronize clock
UPEQUS,DNEQU
PWM control
Interrrupt request
INTPMDx
CMPRLD
Up/Down
fsys
Clock control
PWM counter
MDCNT
Selecter
0x0001
A=B
PWM cycle register
MTPDxMDPRD
Selecter
Latch
PMD compare register
Buffer U
MTPDxCMPU
A≦B
pwmu
Buffer V
MTPDxCMPV
A≦B
pwmv
Buffer W
MTPDxCMPW
A≦B
pwmw

Figure 21-22 Schematic diagram of pulse width modulation circuit

Pulse width modulation circuit contains PWM counter MDCNT which is a 16-bit up-/down-counter. In PWM mode 0 (sawtooth), the counter resolution is 12.5ns@fsys=80MHz. In PWM mode 1(triangle wave), the counter resolution is 25ns@fsys=80MHz to generate PWM carrier. When MTPDMDCR<PWMCK>="1" is set, the counter resolution is 50ns@fsys=80MHz in PWM mode 0 (sawtooth). In the PWM mode 1 (triangle wave), the counter resolution is 100ns@fsys=80MHz to generate PWM carrier.

PWM carrier waveform mode can be chosen either an edge PWM (sawtooth) in PWM mode 0 or a center PWM (triangle wave) in mode 1.

1. PWM cycle setting

   MTPDMDPRD<MDPRD> register determines PWM cycles.

   MTPDxMDPRD register contains a latch circuit with double-buffering structure. Register values are synchronously transferred as the comparator input (latch) when PWM counter MDCNT matches with <MDPRD>. If MTPDMDCR<INTPRD> is set to "00", updating on every PWM half cycle can be chosen.

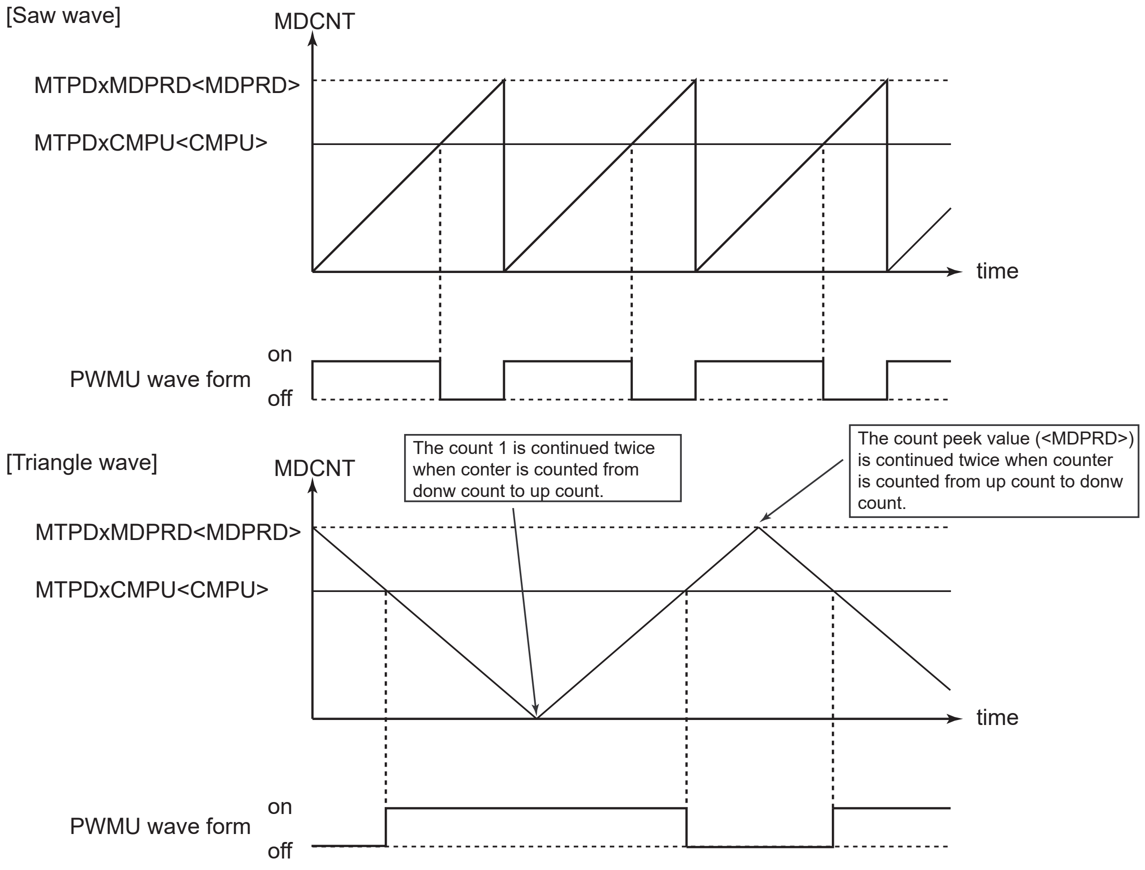
$$\text{Saw wave PWM : Value of MDPRD register} = \frac{\text{Oscillation frequency [Hz]}}{\text{PWM frequency [Hz]}}$$

$$\text{Triangle wave modulation PWM : Value of MDPRD register} = \frac{\text{Oscillation frequency [Hz]}}{\text{PWM frequency [Hz]} \times 2}$$

2. Compare function

   This compare function generates desirable duty PWM waveforms by small/large comparing a value of 3-phase PWM compare register (PMDxCMPU/V/W) with a carrier generated by PWM counter MDCNT.

   PWM compare registers in each phase have double-buffering structure. Buffer values are synchronously transferred to PWM compare register when internal counter value matches with <MDPRD>. If MTPDMDCR<INTPRD> is set to "00", loading on every PWM half cycle can be chosen.



The count 1 is continued twice when conter is counted from donw count to up count.

The count peek value (<MDPRD>) is continued twice when counter is counted from up count to donw count.

Figure 21-23 PWM waveform

3. Waveform mode

   Two kinds of 3-phase PWM generation can be chosen.

   1. 3-phase independent duty mode: Set independent values to each 3-phase compare register to generate 3-phase independent PWM waveforms. This is used for generating arbitrary drive waveform such as a sine wave.

   2. 3-phase common duty mode: Set only U-phase PWM compare register. By setting a value in U-phase, identical PWM waveform in 3-phase common. This is used for square waveform drive used in DC motor.

4. Interrupt service routine

   In pulse width modulation circuit, PWM interrupts synchronously occurs with PWM waveforms. A frequency of PWM interrupt is chosen among once every half PWM cycle, once every one PWM cycle, once every two PWM cycles or once every four PWM cycles.

21.9.2 Applying Current Control Circuit

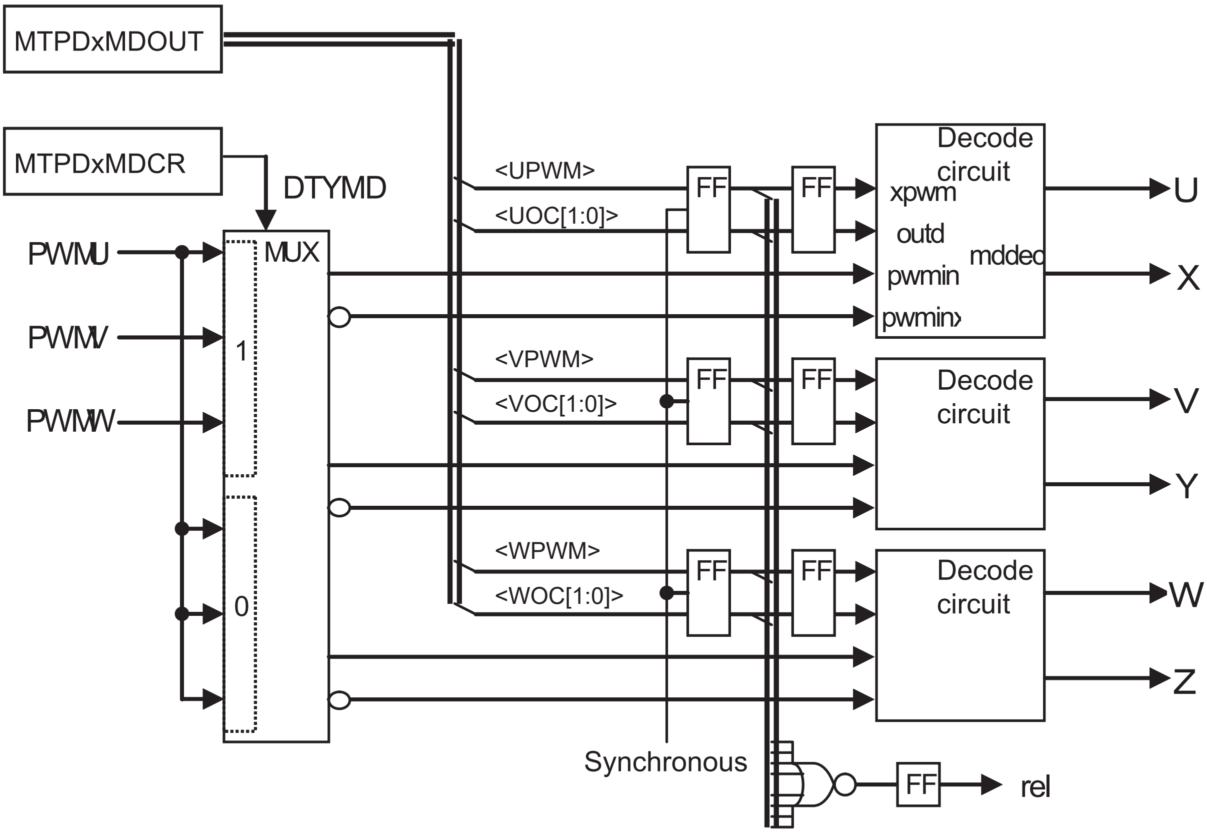


Figure 21-24 Schematic diagram of applying current control circuit

This circuit controls output port according to the content set in the PMD output register (MTPDMDOUT). The settings can be divided two contents such as the selection of synchronous signal at port output, and the port output setting. The port output setting is double-buffering structure, so that the update timing can be chosen between synchronous update or asynchronous update with PWM.

The output setting of six ports can be set independently between active or inactive using MTPDMDPOT<POLH><POLL>. In addition, PWM output or H/L output can be chosen in each three phase (U-, V- and W-phase) using MTPDMDOUT<WPWM><VPWM><UPWM>. If PWM output is chosen, PWM waveform generates. If H/L output is chosen, fixed-high output or fixed-low output generates. For the relation between the port output settings using MTPDMDOUT and the pin output using the polarity of PMD control register(MTPDMDCR)., refer to"Table 21-6 Port outputs setting by UOC, VOC, WOC, UPWN, VPWN and WPWM bits".

Also, one shunt current can be detected as follows:

Table 21-5 Settings for one shunt current detection

	Normal	U-phase PWM shift	V-phase PWM shift	W-phase PWM shift
CMPU	duty_U	MTPDMDPRD <MDPRD>-duty_U	duty_U	duty_U
CMPV	duty_V	duty_V	MTPDMDPRD <MDPRD>D-duty_V	duty_V
CMPW	duty_W	duty_W	duty_W	MTPDMDPRD <MDPRD>-duty_W
<UOC>	11	00	11	11
<VOC>	11	11	00	11
<WOC>	11	11	11	00

Table 21-6 Port outputs setting by UOC, VOC, WOC, UPWN, VPWN and WPWM bits

MTPDMDCR<SYNTMD>=0

Polarity: high-active(MTPDMDPOT<POLH><POLL>="11")

MDOUT output control		MTPDMDOUT <WPWM><VPWM><UPWM> H/L/PWM output selection			
<WOC[1]> <VOC[1]> <UOC[1]> (Upper)	<WOC[0]> <VOC[0] ><UOC[0]> (Lower)	0 : H/L output		1 : PWM output	
		Upper output	Lower output	Upper output	Lower output
0	0	L	L	$\overline{\text{PWM}}$	PWM
0	1	L	H	L	PWM
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{\text{PWM}}$

MTPDMDCR<SYNTMD>=0

Polarity: low-active(MTPDMDPOT<POLH><POLL>="00")

MDOUT output control		MTPDMDOUT <WPWM><VPWM><UPWM> H/L/PWM output selection			
<WOC[1]> <VOC[1]> <UOC[1]> (Upper)	<WOC[0]> <VOC[0] ><UOC[0]> (Lower)	0 : H/L output		1 : PWM output	
		Upper output	Lower output	Upper output	Lower output
0	0	H	H	PWM	$\overline{\text{PWM}}$
0	1	H	L	H	$\overline{\text{PWM}}$
1	0	L	H	$\overline{\text{PWM}}$	H
1	1	L	L	$\overline{\text{PWM}}$	PWM

MTPDMDCR<SYNTMD>=1

Polarity: high-active(MTPDMDPOT<POLH><POLL>="11")

MDOUT output control		MTPDxMDOUT <WPWM><VPWM><UPWM> H/L/PWM output selection			
<WOC[1]> <VOC[1]> <UOC[1]> (Upper)	<WOC[0]> <VOC[0] ><UOC[0]> (Lower)	0 : H/L output		1 : PWM output	
		Upper output	Lower output	Upper output	Lower output
0	0	L	L	$\overline{\text{PWM}}$	PWM
0	1	L	H	L	$\overline{\text{PWM}}$
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{\text{PWM}}$

MTPDMDCR<SYNTMD>=1

Polarity: low-active(MTPDMDPOT<POLH><POLL>="00")

MDOUT output control		MTPDxMDOUT <WPWM><VPWM><UPWM> H/L/PWM output selection			
<WOC[1]> <VOC[1]> <UOC[1]> (Upper)	<WOC[0]> <VOC[0] ><UOC[0]> (Lower)	0 : H/L output		1 : PWM output	
		Upper output	Lower output	Upper output	Lower output
0	0	H	H	PWM	$\overline{\text{PWM}}$
0	1	H	L	H	PWM
1	0	L	H	$\overline{\text{PWM}}$	H
1	1	L	L	$\overline{\text{PWM}}$	PWM

21.9.3 Protection Control Circuit

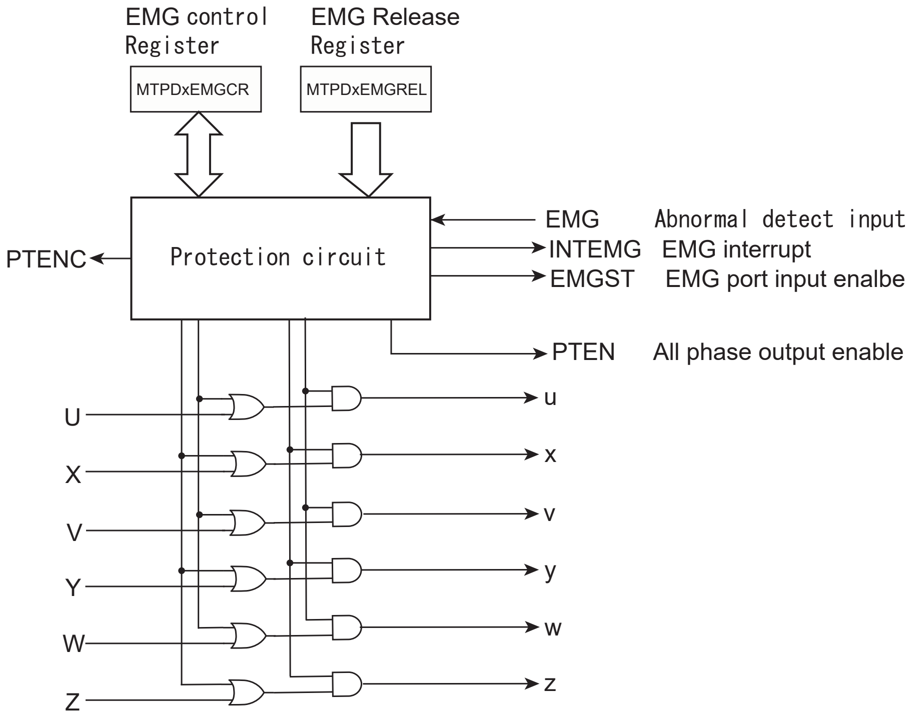

EMG control Register
EMG Release Register
MTPDxEMGCR
MTPDxEMGREL
Protection circuit
PTENC
EMG Abnormal detect input
INTEMG EMG interrupt
EMGST EMG port input enalbe
PTEN All phase output enable
U
X
V
Y
W
Z
u
x
v
y
w
z

Figure 21-25 Schematic diagram of protection control circuit

The protection control circuit consists of the protection control part and the port output prohibition circuit part. It operates when abnormal detection input is low level. The EMG protection circuit is used for emergency stop. If abnormal detection input is found (high level → low level), six PWM outputs are immediately prohibited (depending on MTPDxEMGCR<EMGMD> setting) and an EMG interrupt (INTEMG) occurs.

In addition, this circuit outputs control signals that bemuse external output ports to be high-impedance by setting <EMGMD>.

This circuit prohibits six PWM outputs when PMD is stopped caused by tool break as well. This prohibition depends on <EMGMD> setting. At tool break, it can choose the high-impedance control of external output port by setting MTPDPORTMD<PORTMD>.

When MTPDEMGSTA<EMGST> reads as "1", this indicates MCU is in the EMG protection status.

To return from the EMG protection status, set as follows; all ports are set to in-active (MTPDMDOUT <WPWM> <VPWM><UPWM><WOC[1:0]><VOC[1:0]><UOC[1:0]>="0"); then MTPDEMGCR<EMGRS> is set to "1".

To prohibit the EMG function, set as follows; set 0x5A and 0xA5 to the EMG prohibition code register (MTPDEMGREL<EMGREL[7:0]>)in order; set "0" to MTPDEEMGCR<EMGEN> (These three instruction must be executed consecutively.) However, the returning service routine is ignored while abnormal detection inputs are low. The returning service routine must be taken place when abnormal detection input level becomes high after confirming MTPDEMGSTA<EMGI> is high.

By setting specified key codes (0x5A and 0xA5) to <EMGREL[7:0]>, the EMG protection circuit is enabled to prevent unintentional operation.

21.9.4 Dead Time Circuit

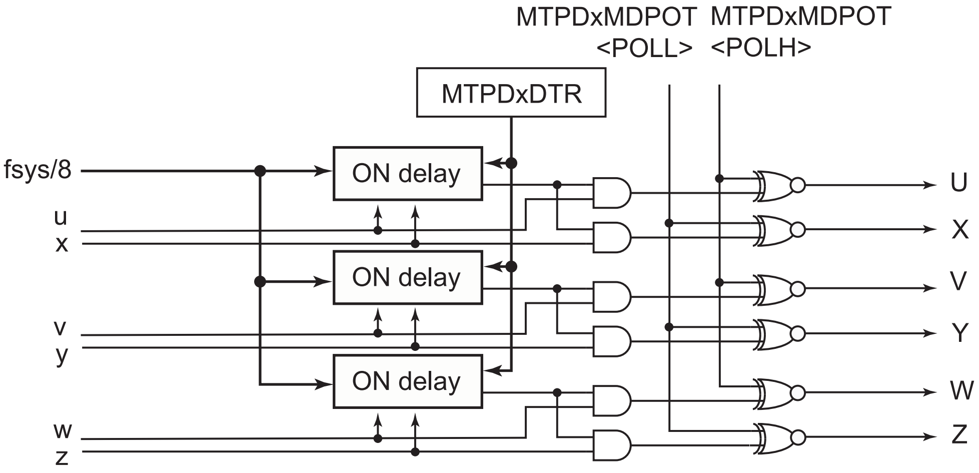

MTPDxMDPOT <POLL>
MTPDxMDPOT <POLH>
MTPDxDTR
fsys/8
u
x
v
y
w
z
ON delay
ON delay
ON delay
U
X
V
Y
W
Z

Figure 21-26 Schematic diagram of dead time circuit

The dead time circuit consists of the dead time circuit part and the output polarity switching part.

In each U-, V- and W-phase, this circuit prevent short circuit with delaying on-time using the dead time counter in case that upper phase and lower phase are reversed. A delay time is set to the dead time register (MTPDDTR<DTR>) and can be set to 100ns @ fsys=80MHz resolution with 8 bits.

The output polarity switching circuit can be set upper or lower signals to be high-active or low-active respectively using TPDxMDPOT<POLH><POLL>.

22. Encoder Input Circuit (ENC)

22.1 Outline

The encoder input circuit supports four operation modes including encoder mode, sensor mode (two types) and timer mode. And the functions are as follows:

- Supports incremental encoders and Hall sensor ICs. (signals of Hall sensor IC can be input directly)
- 24-bit general-purpose timer mode
- Multiply-by-4 (multiply-by-6) circuit
- Rotational direction detection circuit
- 24-bit counter
- Comparator enable/disable
- Interrupt request output:1
- Digital noise filters for input signals

22.2 Differences between channels

The TMPM367FDFG has a one-channel incremental encoder interface (ENC0), which can obtain the absolute position of the motor, based on input signals from the incremental encoder.

These channels operate identical except the differences in below.

Table 22-1 Differences between channels

Channel	Input pin			Encoder input interrupt
	A-phase	B-phase	Z-phase	
Channel0	PF7 / ENCA	PF6 / ENCB	PF5 / ENCZ	INTENC

22.3 Block Diagram

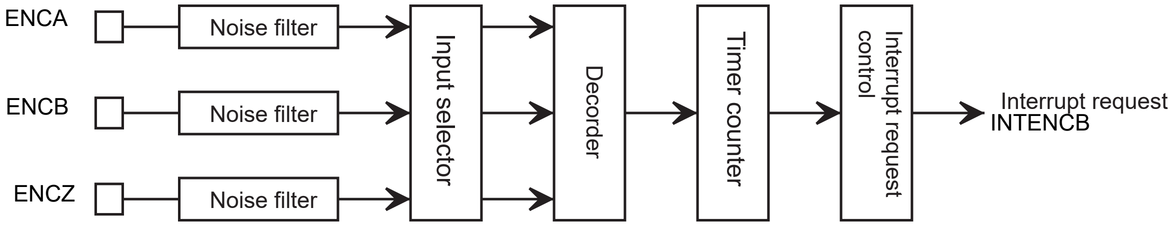

ENCA
ENCB
ENCZ
Noise filter
Noise filter
Noise filter
Input selector
Decorder
Timer counter
Interrupt request control
Interrupt request
INTENCB

Figure 22-1 Block diagram of encoder input circuit

22.4 Registers

22.4.1 List of Registers

The following is control registers and addresses of encoder input circuit.

Base Address = 0x400F_7000

Register name		Address (Base+)
Encoder Input Control Register	ENTNCR	0x0000
Encoder Counter Reload Register	ENRELOAD	0x0004
Encoder Compare Register	ENINT	0x0008
Encoder Counter	ENCNT	0x000C

22.4.2 ENTNCR (Encoder Input Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	MODE		P3EN
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CMP	REVERR	UD	ZDET	SFTCAP	ENCLR	ZESEL	CMPEN
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ZEN	ENRUN	NR		INTEN	ENDEV		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-19	−	R	Read as "0".
18-17	MODE[1:0]	R/W	Encoder input mode setting 00:Encoder mode 01:Sensor mode (event count)) 10:Sensor mode (timer count)) 11:Timer mode
16	P3EN	R/W	2-phase / 3-phase input selection (sensor mode) (Note 1) 0:2-phase input 1:3-phase input Sets the number of input signals.
15	CMP	R	Compare flag 0:- 1:Compare (Clear by RD) If comparing is executed, <CMP> is set to "1". Flag is cleared by reading the values. When <ENRUN> = "0" is set, always "0" is set. Writing to this bit is no effect.
14	REVERR	R	Reverse error flag (Sensor mode (at timer count)) (Note 2) 0:- 1:Error (Clear by RD) In sensor mode (at timer count), when a reverse error occurs, <REVERR> is set to "1". Flag is cleared by reading the values. When <ENRUN> = "0" is set, always "0" is set. Writing to this bit is no effect. In the encoder mode, sensor mode (event count) and timer mode, this bit has no meaning.
13	UD	R	Rotation direction 0:CCW (A-phase has the 90-degree phase lead to B-phase using incremental encoder) 1:CW (A-phase has the 90-degree phase lag to B-phase using incremental encoder) <UD> is set to "0", when <ENRUN> = 0.

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>12</td><td>ZDET</td><td>R</td><td>Z-Detected
0:Not detected
1:Z-phase detected

<ZDET> is set to 1 on the first edge of Z input signal (ENCZ) after <ENRUN> is written from 0 to 1.
This occurs on a rising edge of the signal Z during CW rotation or on a falling edge of Z during CCW rotation.
<ZDET> is set to "0" when <ENRUN> = 0.
<ZEN> has no influence on the value of <ZDET>.
<ZDET> is set to "0" in the sensor event count and the sensor timer count modes.
In the sensor mode (event count) and sensor mode (timer count), this bit is always set to "0".</td></tr>
<tr><td>11</td><td>SFTCAP</td><td>W</td><td>Executes software capture (timer mode/sensor mode (at timer count))
0:-
1:Software capture

If <SFTCAP> is set to 1, the value of the encoder counter is captured into the ENCNT register.
Writing "0" to <SFTCAP> has no effect. Reading <SFTCAP> always returns to "0".
In Encoder and Sensor Event Count modes, <SFTCAP> has no effect; writing "1" to this bit is ignored.</td></tr>
<tr><td>10</td><td>ENCLR</td><td>W</td><td>Encoder pulse counter clear
0:-
1:Clear

Writing a 1 to <ENCLR> clears the encoder counter to 0. Once cleared, the encoder counter restarts counting from 0. Writing "0" to <ENCLR> has no effect. Reading <ENCLR> always returns to "0".</td></tr>
<tr><td>9</td><td>ZESEL</td><td>R/W</td><td>Edge selection of ENCZ (timer mode)
0:Rising edge
1:Falling edge

In timer mode, this bit selects inputs edge of ENCZ used as external trigger.
In the other mode, this bit has no meaning.</td></tr>
<tr><td>8</td><td>CMPEN</td><td>R/W</td><td>Compare enable
0:Disable
1:Enable

When "1" is set to <CMPEN>, this bit compares counter values of encoder counter with register value of ENINT. When "0" is set to <CMPEN>, this compare is disabled.</td></tr>
<tr><td>7</td><td>ZEN</td><td>R/W</td><td>Z-phase enable (Encoder mode/timer mode)
0:Disable
1:Enable

In the other mode, this bit has no meaning
<table>
<tr><td><Encoder mode>
Clear setting of encoder counter using ENCZ input</td><td>When <ZEN> = "1" is set, if a rising edge of ENCZ is detected during rotating clockwise, the encoder counter is cleared to "0".
If a falling edge of ENCZ is detected during rotating counter- clockwise, the encoder counter is cleared to "0".
If the edges of ENCLK (multiply by 4 clock derived from the decoded A and B signals) and the edge of ENCZ coincide, the encoder counter is cleared to 0 without incrementing or decrementing (i.e., the clear takes precedence).</td></tr>
<tr><td><Timer mode>
Sets ENCZ input to use as an external trigger.</td><td>When <ZEN> = 1, the value of the encoder counter is captured into the ENINT register and cleared to 0 on the edge of ENCZ selected by <ZESEL>.</td></tr>
</table></td></tr>
<tr><td>6</td><td>ENRUN</td><td>R/W</td><td>Encoder operation enable
0:Disable
1:Enable

Setting <ENRUN> to 1 and clearing <ZDET> to 0 enables the encoder operation.
Clearing <ENRUN> to 0 disables the encoder operation.
There are counters and flags that are cleared and not cleared when <ENRUN> bit is cleared to 0.</td></tr>
</table>

Bit	Bit Symbol	Type	Function
5-4	NR[1:0]	R/W	Noise filter 00:No filtering 01:Filters out pulses narrower than 31/fsys as noise (387.5ns@80MHz) 10:Filters out pulses narrower than 63/fsys as noise (787.5ns@80MHz) 11:Filters out pulses narrower than 127/fsys as noise (1587ns@80MHz) The digital noise filters remove pulses narrower than the width selected by <NR[1:0]>.
3	INTEN	R/W	Encoder interrupt enable 0:Disable 1:Enable <INTEN> enables or disables the ENC interrupt. Setting <INTEN> to "1" enables interrupt generation. Setting <INTEN> to "0" disables interrupt generation.
2-0	ENDEV[2:0]	R/W	Encoder pulse division factor 000:divided by 1 100:divided by 16 001:divided by 2 101:divided by 32 010:divided by 4 110:divided by 64 011:divided by 8 111:divided by 128 Sets encoder pulse division factor The frequency of the encoder pulse is divided by the factor specified by <ENDEV[2:0]>. The divided signal determines the interval of the event interrupt.

Note 1: In the encoder mode or timer mode, <P3EN> must be set to "0".

Note 2: If changing the mode, first read the flag to clear.

The operation mode has eight modes specified with<MODE[1:0]>, <P3EN> and <ZEN>.

The operation mode settings are as follows:

<MODE[1:0]>	<ZEN>	<P3EN>	Input pin	Mode
00	0	0	A, B	Encoder mode
	1		A,B,Z	Encoder mode (use of Z)
01	0	0	U,V	Sensor mode (event count, 2-phase input)
		1	U,V,W	Sensor mode (event count, 3-phase input)
10	0	0	U,V	Sensor mode (timer count, 2-phase input)
		1	U,V,W	Sensor mode (timer count, 3-phase input)
11	0	0	-	Timer mode
	1		Z	Timer mode (use of Z)

The following is the status of <ENRUN> and corresponding signals.

Counter/flag	<ENRUN> = 0 (After reset)	<ENRUN> = 1 (Operating)	<ENRUN> = 0 (Stopping)	<ENRUN> = 0 Object flag/counter clear procedure
Encoder counter	0x000000	Count operation	Maintains a value when stopping	Software clear (<ENCLR> = 1 WR)
Noise filter counter	000_0000	Count-up operation	Count-up operation (Always filtering)	Only reset
Encoder pulse division counter	0x00	Count-down operation	Stopped and cleared	Clear when <ENRUN> = 0
Compare flag <CMP>	0	"1" is set when comparing Clear when read.	Cleared	Clear when <ENRUN> = 0
Reverse error flag <REVERR>	0	"1" is set when error occurs. Clear when read.	Cleared	Clear when <ENRUN> = 0
Z detection flag <ZDET>	0	"1" is set when Z is detected.	Cleared	Clear when <ENRUN> = 0
Rotation direction bit <UD>	0	"0"/"1" is set depending on the direction	Cleared	Clear when <ENRUN> = 0

22.4.3 ENRELOAD (Encoder Counter Reload 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	RELOAD							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RELOAD							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-0	RELOAD[15:0]	R/W	Sets the Encoder counter period (after multiplied by 4 or 6) 0x0000 to 0xFFFF Z-phase is used : Sets the number of count pulses for one rotation Z-phase is not used : Sets the number of count pulses minus one for one rotation <RELOAD[15:0]> defines the encoder counter period multiplied by 4. If the encoder counter is configured as an up-counter, it increments up to the value programmed in <RELOAD[15:0]> and then wraps around to 0 on the next ENCLK. If the encoder counter is configured as a down-counter, it decrements to 0 and then is reloaded with the value of <RELOAD[15:0]> on the next ENCLK.

The RELOAD register is only used in Encoder mode.

22.4.4 ENINT (Encoder Compare 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	INT							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	INT							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	INT							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function		
31-24	−	R	Read as "0".		
23-0	INT[23:0]	R/W	Counter compare value setting		
			Encoder mode:	Interrupt condition of the encoder pulse position.	0x0000 to 0xFFFF
				While <CMPEN> = 1 is set, if an encoder counter value matches a value of <INT[15:0]>, <CMP> is set to "1". If <INTEN> = 1is set, an interrupt request (INTENCx) occurs. However if <ZEN> = 1 is set, an interrupt request does not occur until <ZDET> = 1.	
			Sensor mode: (event count)	Interrupt condition of the encoder pulse position.	0x0000 to 0xFFFF
				While <CMPEN> = 1 is set, if an encoder counter value matches a value of <INT[15:0]>, <CMP> is set to "1". If <INTEN> = 1is set, an interrupt request (INTENCx) occurs. This bit has no effect on a value of <ZEN>.	
			Sensor mode: (Timer count)	Interrupt condition of abnormal pulse detection time	0x000000 to 0xFFFFFF
				When <CMPEN> = 1 is set, an internal counter value matches a value of <INT[23:0]>, abnormal pulse detection time error is determined and <CMP> is set to "1". If <INTEN> = 1 is set, an interrupt request (INTENCx) occurs. This bit has no effect on a value of <ZEN>.	
			Timer mode	Interrupt condition of timer compare	0x000000 to 0xFFFFFF
				When <CMPEN> = 1 is set, an internal counter value matches a value of <INT[23:0]>, abnormal pulse detection time error is determined and <CMP> is set to "1". If <INTEN> = 1 is set, an interrupt request (INTENCx) occurs. This bit has no effect on a value of <ZEN>.	

<INT[23:16]> is used only in Sensor mode (timer count) and Timer mode.

22.4.5 ENCNT (Encoder Counter)

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

Bit	Bit Symbol	Type	Function		
31-24	−	R	Read as "0".		
23-0	CNT[23:0]	R/W	Encoder counter/capture value		
			Encoder mode:	Counter value of encoder pulse	0x0000 to 0xFFFF
				The value of encoder count can be read. In Encoder mode, the encoder counter counts up or down on each encoder pulse (ENCLK). During CW rotation, encoder counter counts up; when it has reached to the value of <RELOAD[15:0]>, it wraps around to 0 on the next ENCLK. During CCW rotation, encoder counter counts down; when it has reached to 0, it is reloaded with the value of <RELOAD[15:0]> on the next ENCLK.	
			Sensor mode: (event count)	Counter value of encoder pulse	0x0000 to 0xFFFF
				The value of encoder count can be read. In Sensor Event Count mode, the encoder counter counts up or down on each encoder pulse (ENCLK). During CW rotation, encoder counter counts up; when it has reached to 0xFFFF, it wraps around to 0 on the next ENCLK. During CCW rotation, encoder counter counts down; when it has reached to 0, it wraps around to 0xFFFF on the next ENCLK.	
			Sensor mode: (Timer count)	Pulse detection time or captured value by software	0x000000 to 0xFFFFFF
				The value of encoder counter can be read. In Sensor mode, the value of encoder counter can be read and captured by software on each encoder pulse (ENCLK) by writing "1" to <SFTCAP>. The captured value is cleared to 0 by system reset. It can also be cleared by clearing the counter by setting <ENCLR> to 1 and then setting <SFTCAP> to 1. In Sensor Timer Count mode, the encoder counter is configured as a free-running counter that counts up with fsys. The encoder counter is cleared to 0 when the encoder pulse (ENCLK) is detected. When it has reached to 0xFFFFFF, it wraps around to 0 automatically.	
			Timer mode	Capture value of internal counter or captured value by software	0x000000 to 0xFFFFFF
				The value of encoder counter can be read and captured by software by writing "1" to <SFTCAP>.When <ZEN> = 1, the value of the encoder counter is also captured into <CNT[23:0]> on the Z edge selected by <ZESEL>. The captured value is cleared to "0" by reset. It can also be cleared by clearing the counter by setting <ENCLR> to 1 and then setting <SFTCAP> to 1. In Timer mode, the encoder counter is configured as a free-running counter that counts up with fsys. When it has reached to 0xFFFFFF, it wraps around to 0 automatically.	

<CNT[23:16]> is used only in the sensor mode (Timer counting) or timer mode. In the encoder mode or sensor mode (event counting), always reads as "0".

22.5 Operational Description

22.5.1 Encoder mode

The high-speed position sensor determines the phase input from the AB encoder and the ABZ encoder.

- Event detection (rotation pulse) → interrupt generation
- Event count → match detection interrupt generation (measures the amount of transferring)
- Detects rotation direction
- Up/down-count (changeable in operation)
- Settable counter cycle

22.5.2 Sensor mode

The low-speed position sensor determines (zero-cross determination) the phase input from UV Hall sensor and UVW Hall sensor.

There are two kinds of sensor modes such as event count mode and timer count mode (counts with fsys).

22.5.2.1 Event Count Mode

- Event detection (rotation pulse) → interrupt generation
- Event count → match interrupt occurs (measuring the amount of transfer)
- Rotation direction detection

22.5.2.2 Timer count mode

- Event detection (rotation pulse) → interrupt generation
- Timer count
- Rotation direction detection
- Capture function → event capture (measures event intervals) → interrupt generation

  software capture
- Abnormal detection time error (timer compare) → match detection interrupt generation
- Reverse detection error → error flag caused by changing rotation direction

22.5.3 Timer mode

This mode can be used as a general-purpose 24-bit timer.

- 24-bit up counter
- Counter clear control (software clear, timer clear, external trigger and free-run count)
- Compare function → match detection interrupt generation
- Capture function → external trigger capture → interrupt generation

  software capture

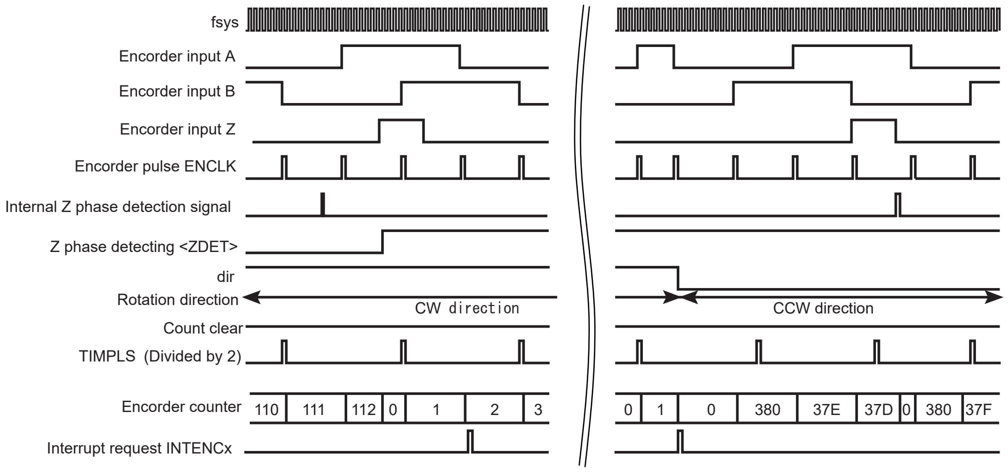
22.6 Function

22.6.1 Mode operation outline

22.6.1.1 Encoder mode

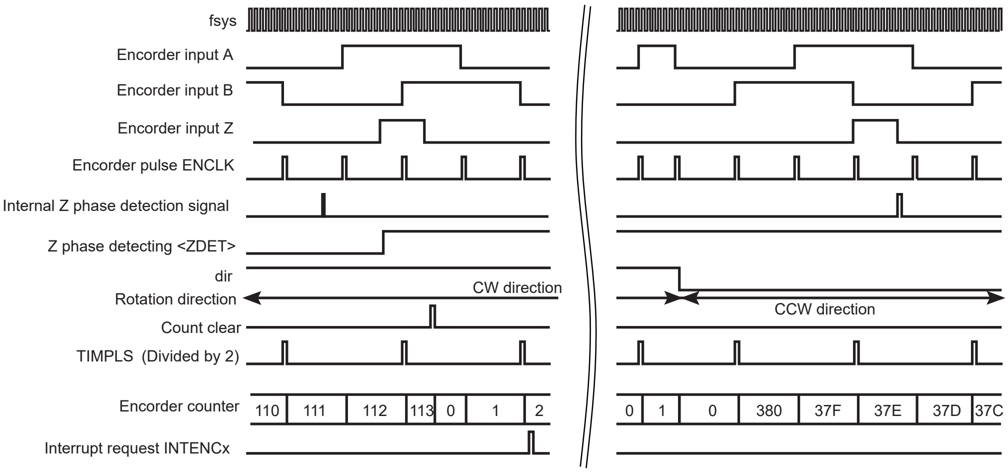
1. If ENTNCR<ZEN> = 1

   (ENRELOAD<RELOAD[15:0]> = 0x0380, ENINT<INT[15:0]> = 0x0002)


fsys
Encorder input A
Encorder input B
Encorder input Z
Encorder pulse ENCLK
Internal Z phase detection signal
Z phase detecting <ZDET>
dir
Rotation direction
CW direction
CCW direction
Count clear
TIMPLS (Divided by 2)
Encorder counter
110
111
112
0
1
2
3
0
1
0
380
37E
37D
0
380
37F
Interrupt request INTENCx

2. If ENTNCR<ZEN> = 0

   (ENRELOAD<RELOAD[15:0]> = 0x0380, ENINT<INT[15:0]> = 0x0002)

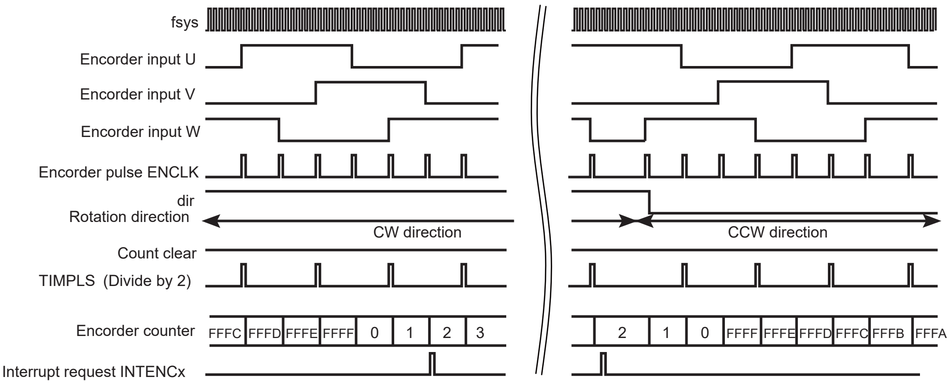

fsys
Encorder input A
Encorder input B
Encorder input Z
Encorder pulse ENCLK
Internal Z phase detection signal
Z phase detecting <ZDET>
dir
Rotation direction
CW direction
CCW direction
Count clear
TIMPLS (Divided by 2)
Encorder counter
110
111
112
113
0
1
2
0
1
0
380
37F
37E
37D
37C
Interrupt request INTENCx

- The incremental encoder inputs of the MCU should be connected to the A, B and Z channels. The encoder counter counts pulses of ENCLK, which is multiplied by 4 clock derived from the decoded A and B quadrature signals.
- During CW rotation (i.e., A has the 90-degree phase lead to B), the encoder counter counts up; when it has reached to the value of ENRELOAD<RELOAD[15:0]>, it wraps around to 0 on the next ENCLK.

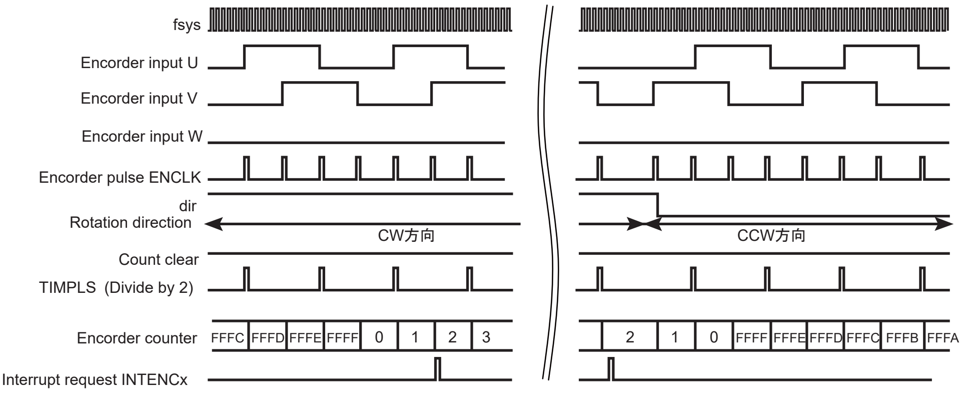
- During CCW rotation (i.e., A has the 90-degree phase lag to B), the encoder counter counts down; when it has reached to 0x0000, it is reloaded with the value of ENRELOAD<RELOAD [15:0]> on the next ENCLK.
- Additionally, when ENTNCR<ZEN> = 1, the encoder counter is cleared to 0 on the rising edge of Z during CW rotation and on the falling edge of Z during CCW rotation (at the internal Z_Detected timing). If the ENCLK edge matches Z edge, the encoder counter is cleared to 0 without incrementing or decrementing.
- When ENTNCR<ENCLR> is set to 1, the encoder counter is cleared to 0.
- ENTNCR<UD> is set to 1 during CW rotation and cleared to 0 during CCW rotation.
- TIMPLS, which is derived by dividing ENCLK by a programmed factor, can be driven out externally.
- If ENTNCR<CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached to the value of ENINT<INT[15:0]>. When ENTNCR<ZEN> = 1, however, an interrupt does not occur while ENTNCR<ZDET> = 0.
- When <ZDET> and <UD> are set to "0", ENTNCR<ENRUN> is cleared to "0".

22.6.1.2 Sensor mode (event count)

1. If ENTNCR<P3EN> = 1 (ENINT<INT[15:0]> = 0x0002)


fsys
Encorder input U
Encorder input V
Encorder input W
Encorder pulse ENCLK
dir
Rotation direction
CW direction
CCW direction
Count clear
TIMPLS (Divide by 2)
Encorder counter
FFFC
FFFD
FFFE
FFFF
0
1
2
3
2
1
0
FFFF
FFFE
FFFD
FFFC
FFFB
FFFA
Interrupt request INTENCx

2. If ENTNCR<P3EN> = 0 (ENINT<INT[15:0]> = 0x0002)

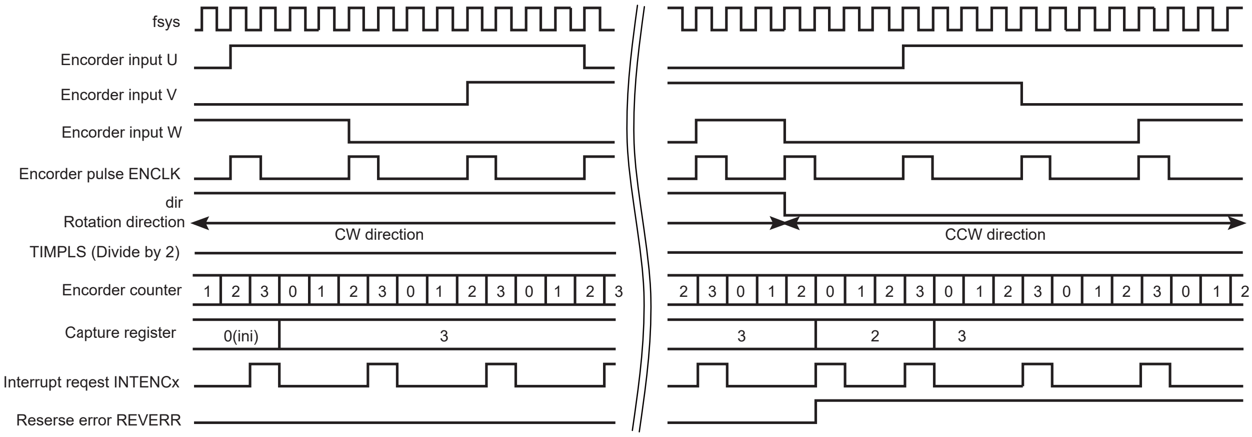

fsys
Encorder input U
Encorder input V
Encorder input W
Encorder pulse ENCLK
dir
Rotation direction
CW方向
CCW方向
Count clear
TIMPLS (Divide by 2)
Encorder counter
FFFC
FFFD
FFFE
FFFF
0
1
2
3
2
1
0
FFFF
FFFE
FFFD
FFFC
FFFB
FFFA
Interrupt request INTENCx

- The Hall sensor inputs of the MCU should be connected to the U, V and W channels. The encoder counter counts the pulses of ENCLK, which is either multiplied by 4 clock (when ENTNCR<P3EN> = 0) derived from the decoded U and V signals or multiplied by 6 clock (when ENTNCR<P3EN> = 1) derived from the decoded U, V and W signals.

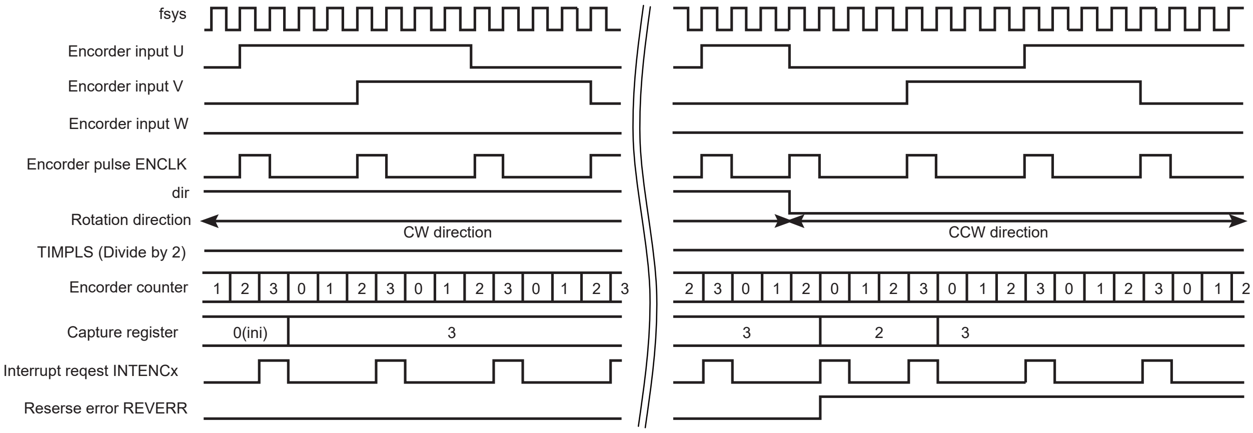
- During CW rotation (i.e., U channel has the 90-degree phase lead to V channel; V channel has the 90-degree phase lead to W channel), the encoder counter counts up; when it has reached to 0xFFFF, it wraps around to 0 on the next ENCLK.
- During CCW rotation (i.e., U channel has the 90-degree phase lag to V channel; V channel has the 90-degree phase lag to W), the encoder counter counts down; when it has reached to 0x0000, it wraps around to 0xFFFF on the next ENCLK.
- When ENTNCR<ENCLR> is set to 1, the internal counter is cleared to 0.
- ENTNCR<UD> is set to 1 during CW rotation and cleared to 0 during CCW rotation.
- TIMPLS, which is derived by dividing ENCLK by a programmed factor, can be driven out externally.
- If ENTNCR<CMPEN> is set to 1, an interrupt is generated when the value of the internal counter has reached to the value of ENINT<INT[15:0]>.
- When ENTNCR<UD> and ENTNCR<ENRUN> are set to "0", <UD> is cleared to "0".

22.6.1.3 Sensor mode (Timer count)

1. If ENTNCR<P3EN> = 1 (ENINT<INT[23:0]> = 0x000002)


fsys
Encoder input U
Encoder input V
Encoder input W
Encorder pulse ENCLK
dir
Rotation direction
TIMPLS (Divide by 2)
Encorder counter
Capture register
Interrupt reqest INTENCx
Reserse error REVERR
CW direction
CCW direction
1 2 3 0 1 2 3 0 1 2 3 0 1 2 3
2 3 0 1 2 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2
0(ini)
3
3
2
3

2. If ENTNCR<P3EN> = 0 (ENINT<INT[23:0]> = 0x000002)

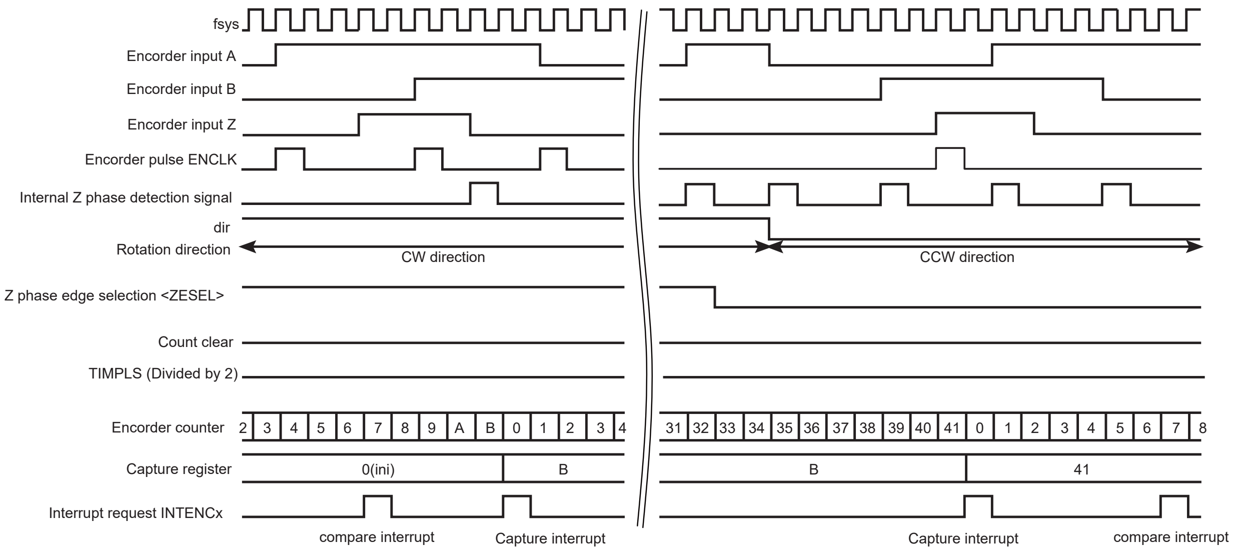

fsys
Encoder input U
Encoder input V
Encoder input W
Encorder pulse ENCLK
dir
Rotation direction
TIMPLS (Divide by 2)
Encorder counter
Capture register
Interrupt reqest INTENCx
Reserse error REVERR
CW direction
CCW direction
1 2 3 0 1 2 3 0 1 2 3 0 1 2 3
2 3 0 1 2 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2
0(ini)
3
3
2
3

- In Sensor Timer Count mode, the Hall sensor inputs of the MCU should be connected to the U, V and W channels. The encoder counter measures the interval between two contiguous pulses of ENCLK, which is either multiplied by 4 clock (when ENTNCR<P3EN> = 0) derived from the decoded U and V signals or multiplied by 6 clock (when ENTNCR<P3EN> = 1) derived from the decoded U, V and W signals.

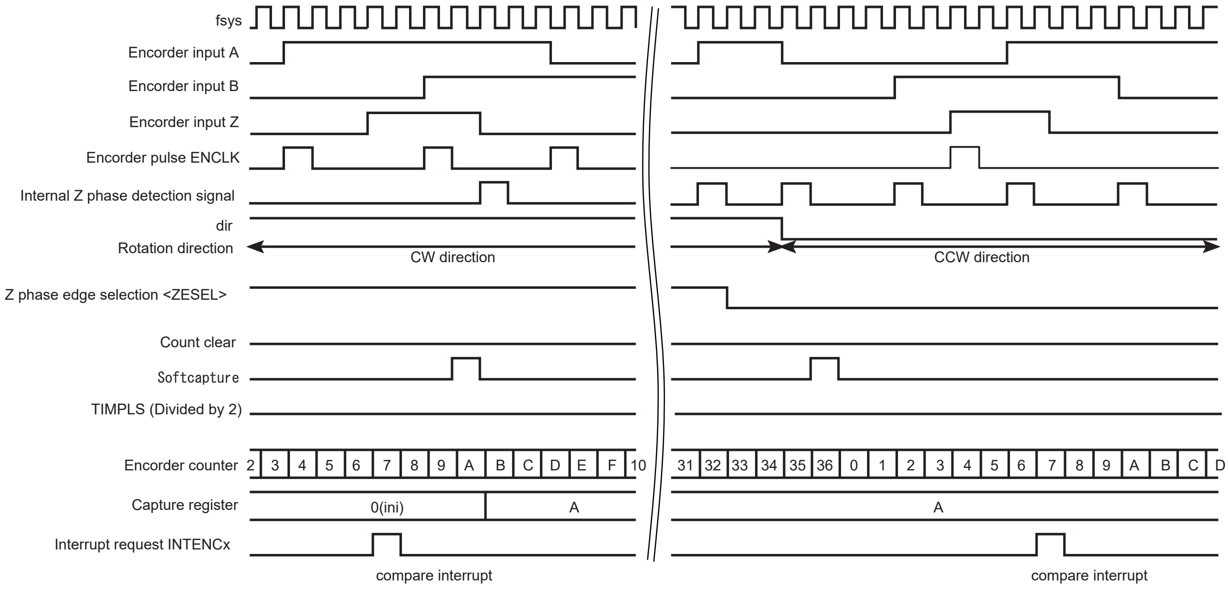
- The encoder counter always counts up; it is cleared to 0 on ENCLK. When the encoder counter has reached to0xFFFFFF, it wraps around to 0.
- When ENTNCR<ENCLR> is set to 1, the encoder counter is cleared to 0.
- ENCLK captures the value of the encoder counter into the ENCNT register. The captured counter value can be read out of ENCNT.
- Setting the software capture bit, ENTNCR<SFTCAP>, to 1 causes the value of the encoder counter to be captured into the ENCNT register. This capture operation can be performed at any time. The captured counter value can be read out of ENCNT.
- ENTNCR<UD> is set to 1 during CW rotation and cleared to 0 during CCW rotation.
- If ENTNCR<CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached to the value of ENINT<INT[23:0]>.
- When ENTNCR<ENRUN> is set to "0", ENTNCR<UD> is cleared to "0".
- ENTNCR<REVERR> is set to 1 when the rotation direction has changed. This bit is cleared to 0 on a read.
- The value of the ENCNT register (the captured value) is retained, regardless of the value of ENTNCR<ENRUN>. The ENCNT register is only cleared by a reset.

22.6.1.4 Timer mode

1. If ENTNCR<ZEN> = 1 (ENINT<INT[23:0]> = 0x000006)


fsys
Encorder input A
Encorder input B
Encorder input Z
Encorder pulse ENCLK
Internal Z phase detection signal
dir
Rotation direction
CW direction
CCW direction
Z phase edge selection <ZESEL>
Count clear
TIMPLS (Divided by 2)
Encorder counter
2 3 4 5 6 7 8 9 A B 0 1 2 3 4
31 32 33 34 35 36 37 38 39 40 41 0 1 2 3 4 5 6 7 8
Capture register
0(ini)
B
B
41
Interrupt request INTENCx
compare interrupt
Capture interrupt
Capture interrupt
compare interrupt

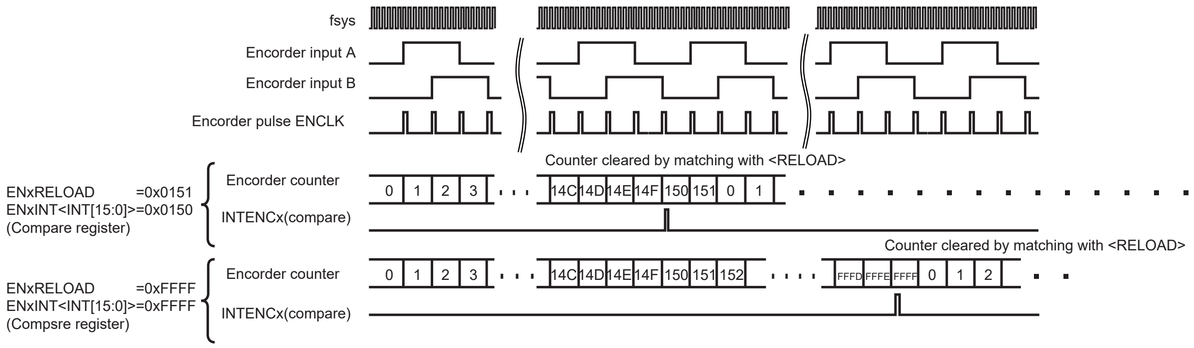
2. If ENTNCR<ZEN> = 0 (ENINT<INT[23:0]> = 0x000006)


fsys
Encoder input A
Encoder input B
Encoder input Z
Encoder pulse ENCLK
Internal Z phase detection signal
dir
Rotation direction
CW direction
CCW direction
Z phase edge selection <ZESEL>
Count clear
Softcapture
TIMPLS (Divided by 2)
Encoder counter
2 3 4 5 6 7 8 9 A B C D E F 10
31 32 33 34 35 36 0 1 2 3 4 5 6 7 8 9 A B C D
Capture register
0(ini)
A
A
Interrupt request INTENCx
compare interrupt
compare interrupt

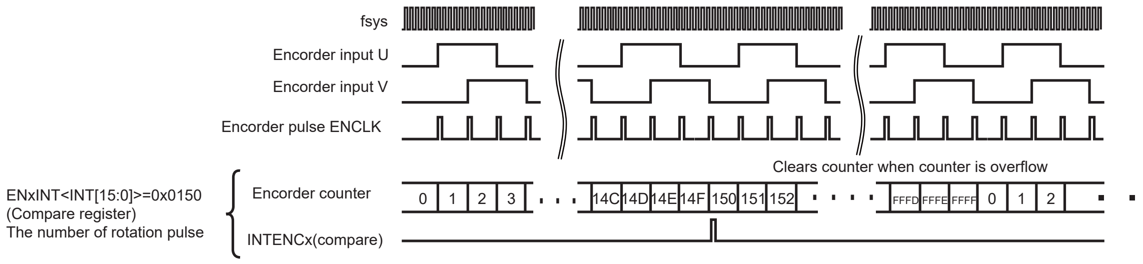
- When ENTNCR<ZEN> = 1, the Z input pin is used as an external trigger. When ENTNCR<ZEN> = 0, no external input is used to trigger the timer.
- The encoder counter always counts up. If ENTNCR<ZEN> = 1, the counter is cleared to 0 on the rising edge of Z when ENTNCR<ZESEL> is set to "0" and a falling edge when ENTNCR<ZESEL> is set to "1". When the encoder counter has reached to 0xFFFFFF, it wraps around to 0.
- When ENTNCR<ENCLR> is set to 1, the encoder counter is cleared to 0.
- Z-Detected causes the value of the encoder counter to be captured into the ENCNT register. The captured counter value can be read out of ENCNT.
- Setting the software capture bit, ENTNCR<SFTCAP>, to 1 causes the value of the encoder counter to be captured into the ENCNT register. This capture operation can be performed at any time. The captured counter value can be read out of ENCNT.
- ENTNCR<UD> is set to 1 during CW rotation and cleared to 0 during CCW rotation.
- If ENTNCR<CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached to the value of ENINT<INT[23:0]>.
- When ENTNCR<ENRUN> is set to "0", ENTNCR<UD> is cleared to "0".
- The value of the ENCNT register (the captured value) is retained, regardless of the value of ENTNCR<ENRUN>. The ENCNT register is only cleared by a reset.

22.6.2 Counter and interrupt generate operation when ENTNCR<CMPEN> = 1

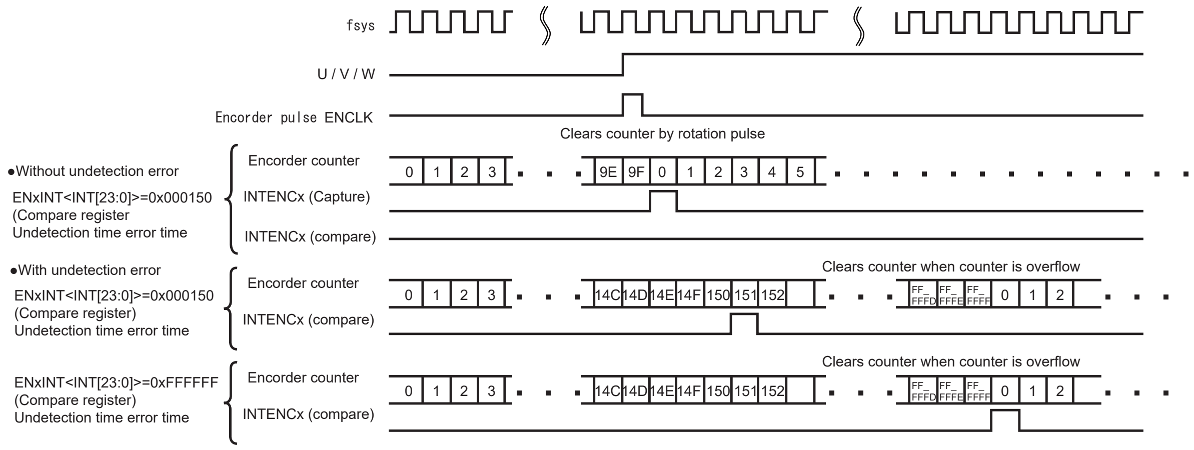
22.6.2.1 Encoder mode


fsys
Encoder input A
Encoder input B
Encoder pulse ENCLK
Counter cleared by matching with <RELOAD>
ENxRELOAD =0x0151
ENxINT<INT[15:0]>=0x0150
(Compare register)
Encorder counter
0 1 2 3 14C 14D 14E 14F 150 151 0 1
INTENCx(compare)
Counter cleared by matching with <RELOAD>
ENxRELOAD =0xFFFF
ENxINT<INT[15:0]>=0xFFFF
(Compsre register)
Encorder counter
0 1 2 3 14C 14D 14E 14F 150 151 152 FFFD FFFE FFFF 0 1 2
INTENCx(compare)

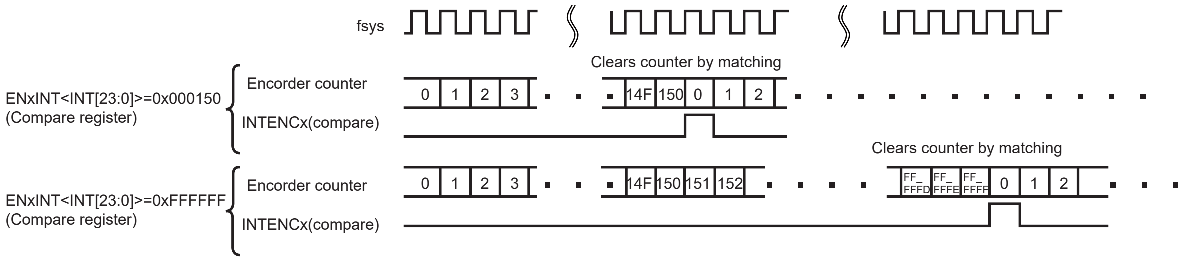
22.6.2.2 Sensor mode (event count)


fsys
Encorder input U
Encorder input V
Encorder pulse ENCLK
Clears counter when counter is overflow
ENxINT<INT[15:0]>=0x0150
(Compare register)
The number of rotation pulse
Encorder counter
0 1 2 3 14C 14D 14E 14F 150 151 152 FFFD FFFE FFFF 0 1 2
INTENCx(compare)

22.6.2.3 Sensor mode (Timer count)


fsys
U / V / W
Encorder pulse ENCLK
Clears counter by rotation pulse
●Without undetection error
ENxINT<INT[23:0]>=0x000150
(Compare register
Undetection time error time
Encorder counter
0 1 2 3 9E 9F 0 1 2 3 4 5
INTENCx (Capture)
INTENCx (compare)
●With undetection error
ENxINT<INT[23:0]>=0x000150
(Compare register)
Undetection time error time
Encorder counter
0 1 2 3 14C 14D 14E 14F 150 151 152 FF_FFFD FF_FFFE FF_FFFF 0 1 2
INTENCx (compare)
Clears counter when counter is overflow
ENxINT<INT[23:0]>=0xFFFFFF
(Compare register)
Undetection time error time
Encorder counter
0 1 2 3 14C 14D 14E 14F 150 151 152 FF_FFFD FF_FFFE FF_FFFF 0 1 2
INTENCx (compare)
Clears counter when counter is overflow

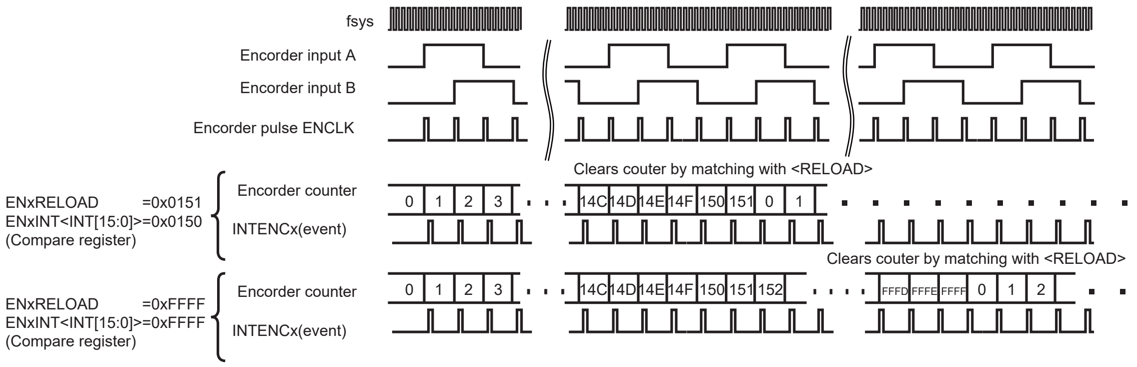
22.6.2.4 Timer mode


fsys
Clears counter by matching
ENxINT<INT[23:0]>=0x000150
(Compare register)
Encorder counter
INTENCx(compare)
0
1
2
3
14F
150
0
1
2
Clears counter by matching
ENxINT<INT[23:0]>=0xFFFFFF
(Compare register)
Encorder counter
INTENCx(compare)
0
1
2
3
14F
150
151
152
FF_FFFD
FF_FFFE
FF_FFFF
0
1
2

22.6.3 Counter and interrupt generate operation when ENTNCR<CMPEN> = 0

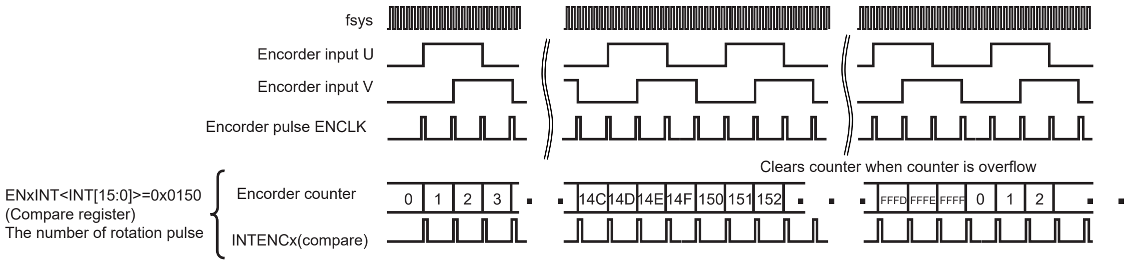
22.6.3.1 Encoder mode

ENTNCR<ENDEV[2:0]>="000"

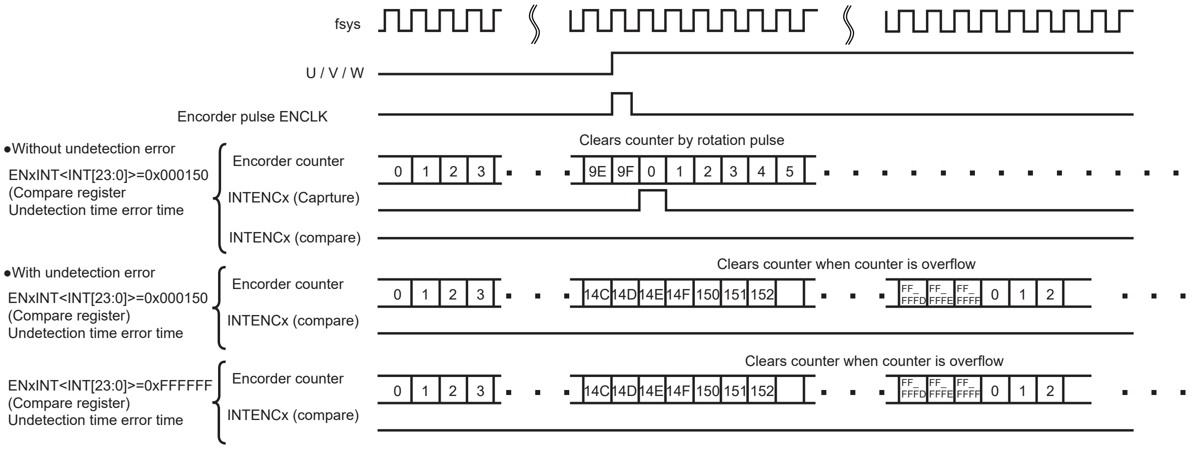

fsys
Encorder input A
Encorder input B
Encorder pulse ENCLK
Clears couter by matching with <RELOAD>
ENxRELOAD =0x0151
ENxINT<INT[15:0]>=0x0150
(Compare register)
Encorder counter
INTENCx(event)
0 1 2 3
14C 14D 14E 14F 150 151 0 1
Clears couter by matching with <RELOAD>
ENxRELOAD =0xFFFF
ENxINT<INT[15:0]>=0xFFFF
(Compare register)
Encorder counter
INTENCx(event)
0 1 2 3
14C 14D 14E 14F 150 151 152
FFFD FFFE FFFF 0 1 2

22.6.3.2 Sensor mode (event count)

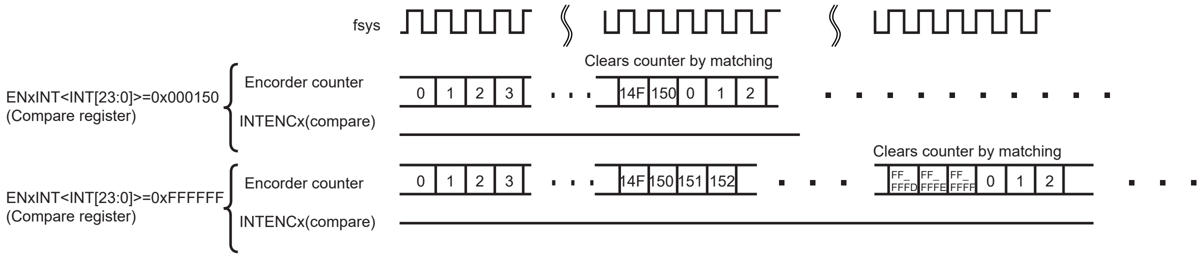
ENTNCR<ENDEV[2:0]>="000"


fsys
Encorder input U
Encorder input V
Encorder pulse ENCLK
Clears counter when counter is overflow
ENxINT<INT[15:0]>=0x0150
(Compare register)
The number of rotation pulse
Encorder counter
INTENCx(compare)
0 1 2 3
14C 14D 14E 14F 150 151 152
FFFD FFFE FFFF 0 1 2

22.6.3.3 Sensor mode (Timer count)


fsys
U / V / W
Encorder pulse ENCLK
●Without undetection error
ENxINT<INT[23:0]>=0x000150
(Compare register
Undetection time error time
Clears counter by rotation pulse
Encorder counter
INTENCx (Caprture)
INTENCx (compare)
0 1 2 3
9E 9F 0 1 2 3 4 5
●With undetection error
ENxINT<INT[23:0]>=0x000150
(Compare register)
Undetection time error time
Clears counter when counter is overflow
Encorder counter
INTENCx (compare)
0 1 2 3
14C 14D 14E 14F 150 151 152
FF_ FFFD FF_ FFFE FF_ FFFF 0 1 2
ENxINT<INT[23:0]>=0xFFFFFF
(Compare register)
Undetection time error time
Clears counter when counter is overflow
Encorder counter
INTENCx (compare)
0 1 2 3
14C 14D 14E 14F 150 151 152
FF_ FFFD FF_ FFFE FF_ FFFF 0 1 2

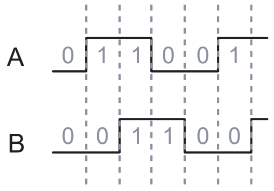
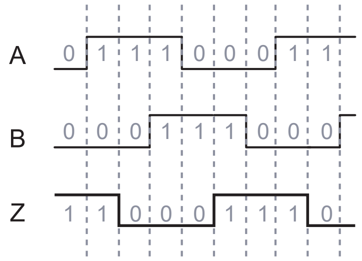
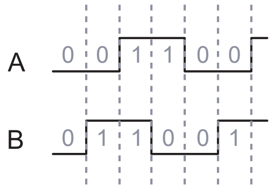
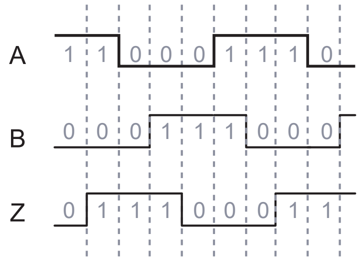
22.6.3.4 Timer mode


fsys
Clears counter by matching
ENxINT<INT[23:0]>=0x000150
(Compare register)
Encorder counter
0
1
2
3
14F
150
0
1
2
INTENCx(compare)
Clears counter by matching
ENxINT<INT[23:0]>=0xFFFFFF
(Compare register)
Encorder counter
0
1
2
3
14F
150
151
152
FF_FFFD
FF_FFFE
FF_FFFF
0
1
2
INTENCx(compare)

22.6.4 Encoder rotation direction

This circuit determines a phase either A-, B- or Z-phase.

It is used as 2-phase input (A,B) and 3-phase input (A,B,Z) in common. When 3-phase input is used, set ENTNCR<P3EN> = 1.

	2-phase input	3-phase input
CW direction	 A 0 1 1 0 0 1 B 0 0 1 1 0 0	 A 0 1 1 1 0 0 0 1 1 B 0 0 0 1 1 1 0 0 0 Z 1 1 0 0 0 1 1 1 0
CCW direction	 A 0 0 1 1 0 0 B 0 1 1 0 0 1	 A 1 1 0 0 0 1 1 1 0 B 0 0 0 1 1 1 0 0 0 Z 0 1 1 1 0 0 0 1 1

22.6.5 Counter Circuit

The counter circuit has a 24-bit up/down counter.

22.6.5.1 Operation Description

Depending on the operation modes, counting, clearing and reloading operation are controlled as described in Table 22-2.

Table 22-2 Counter control

Mode <MODE[1:0]>	<ZEN>	<P3EN>	Input pin	Count	Operation	Counter clear condition	Counter reload condition	Operational range of counter (Reload value)
Encoder mode 00	0	0	A,B	Encoder pulse (ENCLK)	UP	[1]<ENCLR> = 1 WR [2] Matches with <RELOAD>	-	0x0000 to <RELOAD>
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
	1		A,B,Z		UP	[1]<ENCLR> = 1 WR [2] Matches with <RELOAD> [3] Z-trigger	-	
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
Sensor mode (event count) 01	0	0	U,V		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFF	-	0x0000 to 0xFFFF
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
		1	U,V,W		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFF	-	
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
Sensor mode (Timer count) 10	0	0	U,V	fsys	UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFFFF [3] Encoder pulse (ENCLK)	-	0x000000 to 0xFFFFFF
		1	U,V,W		UP		-	
Timer mode 11	0	×	-		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFFFF [3] Matches with <INT[23:0]>	-	0x000000 to 0xFFFFFF
	1		Z		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFFFF [3] Matches with <INT[23:0]> [4] Z-trigger	-	

Note: The counter value is not cleared by writing "0" to ENTNCR<ENRUN>. If <ENRUN> = 1 is set again, the counter restarts from the counter value which has stopped. If clear the counter value, write "1" to <ENCLR> to execute software clear.

22.6.6 Interrupt

The interrupt consists of four interrupts including Event (divide pulse and capture), Abnormal detecting time, Timer compare and Capture interrupts.

22.6.6.1 Operational Description

When ENTNCR<INTEN> = 1 is set, interrupts occurs by counter value and encoder pulses.

Interrupt factor setting consists of six kinds setting with operation modes and the setting of ENTNCR<CMPEN> and <ZEN>.Table 22-3 shows interrupt factors.

Table 22-3 Interrupt factors

	Interrupt factor	Description	Mode	Interrupt output	Status flag
1	Event count interrupt	When <CMPEN> = 1, the encoder counter counts events (encoder pulses). When it has reached to the value programmed in <INT[15:0]>, an interrupt occurs.	Encoder mode and Sensor mode (event count)	<INTEN> = 1 and <CMPEN> = 1	<CMP>
2	Event interrupt (divide pulse)	An interrupt occurs on each divided clock pulse (1 to 128 divide), which is derived by dividing the encoder pulse by a factor programmed in <ENDEV>.		<INTEN> = 1	Not available
3	Event interrupt (capture interrupt)	An interrupt occurs to indicate that an event (encoder pulse) has occurred, causing the counter value to be captured on the rotation pulse timing.	Sensor mode (Timer count)	<INTEN> = 1	Not available
4	Abnormal detection time error interrupt	When <CMPEN> = 1, the ENC uses a counter that counts up with fsys and is cleared by an event (encoder pulse). If no event occurs for a period of time programmed in <INT[23:0]>, an interrupt occurs.		<INTEN> = 1 and <CMPEN> = 1	<CMP>
5	Timer compare interrupt	When <CMPEN> = 1, an interrupt occurs when the timer has reached to the value programmed in <INT[23:0]>.	Timer mode	<INTEN> = 1 and <CMPEN> = 1	<CMP>
6	Capture interrupt	An interrupt occurs when the counter value has been captured on an external trigger (Z input).		<INTEN> = 1	Not available

In Sensor Timer Count mode and Timer mode, the value of the encoder counter can be captured into the ENCNT register.

The captured counter value can be read out of the ENCNT register.

In Sensor Timer Count mode, the value of the encoder counter is captured into the ENCNT register upon occurrence of an event (encoder pulse). The counter value can also be captured by writing a 1 to ENTNCR<SFTCAP> by software.

In Timer mode, the counter value can be captured by writing a 1 to ENTNCR<SFTCAP> by software. If ENTNCR<ZEN> is set to 1, the counter value can also be captured by an edge of the Z signal input selected according to ENTNCR<ZESEL> by external trigger.

23. Real Time Clock (RTC)

23.1 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. Interrupt

23.2 Block Diagram

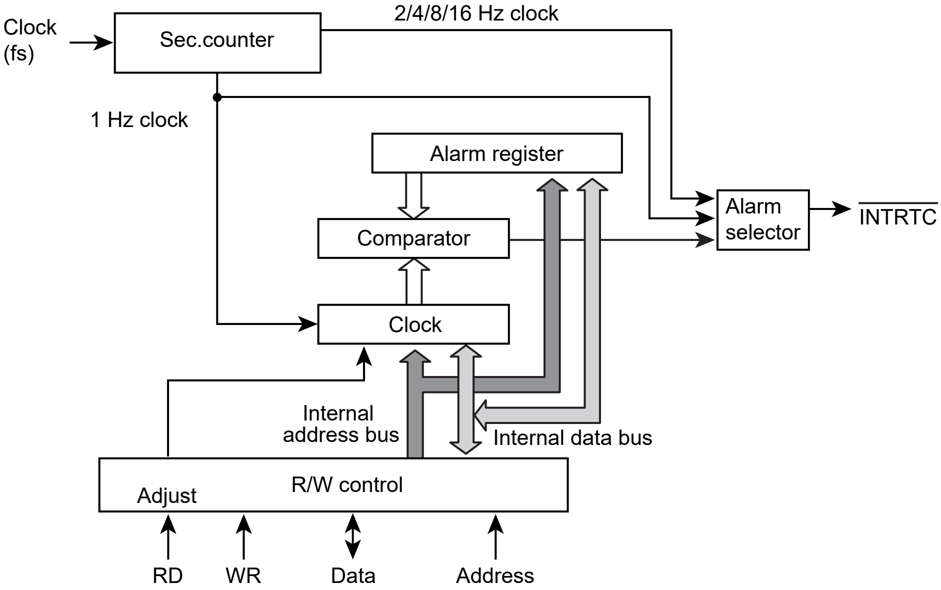

Clock
(fs)
Sec.counter
2/4/8/16 Hz clock
1 Hz clock
Alarm register
Comparator
Alarm
selector
INTRTC
Clock
Internal
address bus
Internal data bus
Adjust
R/W control
RD
WR
Data
Address

Figure 23-1 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.**

23.3 Detailed Description Register

23.3.1 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 = 0x400C_C000

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.**

23.3.2 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 starting the RTC, set the time, month, day, day of the week, year and leap year in the relevant registers.

Caution is required in setting clock data, adjusting seconds or resetting the clock.

Refer to "23.4.3 Entering the Low Power Consumption Mode" for more information.

Table 23-1 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 23-2 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.**

23.3.3 Detailed Description of Control Register

23.3.3.1 RTCSECR (Second column register (for PAGE0 only))

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

Bit	Bit Symbol	Type	Functon
7	−	R	Read as 0.
6-0	SE	R/W	Setting digit register of second 000_0000 : 00sec. 000_0001 : 01sec. 000_0010 : 02sec. 000_0011 : 03sec. 000_0100 : 04sec. 000_0101 : 05sec. 000_0110 : 06sec. 000_0111 : 07sec. 000_1000 : 08sec. 000_1001 : 09sec. 001_0000 : 10sec. 001_0001 : 11sec. 001_0010 : 12sec. 001_0011 : 13sec. 001_0100 : 14sec. 001_0101 : 15sec. 001_0110 : 16sec. 001_0111 : 17sec. 001_1000 : 18sec. 001_1001 : 19sec. 010_0000 : 20sec. · 011_0000 : 30sec. · 100_0000 : 40sec. · 101_0000 : 50sec. · · 101_1001 : 59sec.

Note: **The setting other than listed above is prohibited.**

23.3.3.2 RTCMINR (Minute column register (PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	-	MI						
After reset	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Functon
7	−	R	Read as 0.
6-0	MI	R/W	Setting digit register of Minutes. 000_0000 : 00min. 000_0001 : 01min. 000_0010 : 02min. 000_0011 : 03min. 000_0100 : 04min. 000_0101 : 05min. 000_0110 : 06min. 000_0111 : 07min. 000_1000 : 08min. 000_1001 : 09min. 001_0000 : 10min. 001_0001 : 11min. 001_0010 : 12min. 001_0011 : 13min. 001_0100 : 14min. 001_0101 : 15min. 001_0110 : 16min. 001_0111 : 17min. 001_1000 : 18min. 001_1001 : 19min. 010_0000 : 20min. · 011_0000 : 30min. · 100_0000 : 40min. · 101_0000 : 50min. · · 101_1001 : 59min. 111_1111 :don’t care (Only PAGE1)

Note: **The setting other than listed above is prohibited.**

23.3.3.3 RTCHOURR (Hour column register(PAGE0/1))

(1) 24-hour clock mode (RTCMONTHR<MO0>= "1")

	7	6	5	4	3	2	1	0
Bit symbol	-	-	HO					
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Functon
7-6	−	R	Read as 0.
5-0	HO	R/W	Setting digit register of Hour. 00_0000 : 0 o'clock 00_0001 : 1 o'clock 00_0010 : 2 o'clock 00_0011 : 3 o'clock 00_0100 : 4 o'clock 00_0101 : 5 o'clock 00_0110 : 6 o'clock 00_0111 : 7 o'clock 00_1000 : 8 o'clock 00_1001 : 9 o'clock 01_0000 : 10 o'clock 01_0001 : 11 o'clock 01_0010 : 12 o'clock 01_0011 : 13 o'clock 01_0100 : 14 o'clock 01_0101 : 15 o'clock 01_0110 : 16 o'clock 01_0111 : 17 o'clock 01_1000 : 18 o'clock 01_1001 : 19 o'clock 10_0000 : 20 o'clock 10_0001 : 21 o'clock 10_0010 : 22 o'clock 10_0011 : 23 o'clock 11_1111 : don't care (Only PAGE1)

Note: **The setting other than listed above is prohibited.**

(2) 12-hour clock mode (RTCMONTHR<MO0> = "0")

	7	6	5	4	3	2	1	0
Bit symbol	-	-	HO					
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Functon
7-6	−	R	Read as 0.
5-0	HO	R/W	Setting digit register of Hour. (AM) 00_0000 : 0 o'clock 00_0001 : 1 o'clock 00_0010 : 2 o'clock 00_0011 : 3 o'clock 00_0100 : 4 o'clock 00_0101 : 5 o'clock 00_0110 : 6 o'clock 00_0111 : 7 o'clock 00_1000 : 8 o'clock 00_1001 : 9 o'clock 01_0000 : 10 o'clock 01_0001 : 11 o'clock (PM) 10_0000 : 0 o'clock 10_0001 : 1 o'clock 10_0010 : 2 o'clock 10_0011 : 3 o'clock 10_0100 : 4 o'clock 10_0101 : 5 o'clock 10_0110 : 6 o'clock 10_0111 : 7 o'clock 10_1000 : 8 o'clock 10_1001 : 9 o'clock 11_0000 : 10 o'clock 11_0001 : 11 o'clock 11_1111 : don't care (Only PAGE1)

Note: **The setting other than listed above is prohibited.**

23.3.3.4 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.**

23.3.3.5 RTCDATER (Day column register (for PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	-	-	DA					
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Functon
7-6	−	R	Read as 0.
5-0	DA	R/W	Setting digit register of day. 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 : 21st day 10_0010 : 22nd day 10_0011 : 23rd 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 : 31st 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.).**

23.3.3.6 RTCMONTHR (Month column register (for PAGE0 only))

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	MO				
After reset	0	0	0	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Functon
7-5	−	R	Read as 0.
4-0	MO	R/W	Setting digit register of Month. 0_0001 : January 0_0111 : July 0_0010 : February 0_1000 : August 0_0011 : March 0_1001 : September 0_0100 : April 1_0000 : October 0_0101 : May 1_0001 : November 0_0110 : June 1_0010 : December

Note: **The setting other than listed above is prohibited.**

23.3.3.7 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.**

23.3.3.8 RTCYEARR (Year column register (for PAGE0 only))

	7	6	5	4	3	2	1	0
bit symbol	YE							
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-0	YE	R/W	Setting digit register of Year. 0000_0000 : 00 years 0001_0000 : 10 years 0110_0000 : 60 years 0000_0001 : 01 year · · 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.**

23.3.3.9 RTCYEARR (Leap year register (for PAGE1 only))

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

Bit	Bit Symbol	Type	Functon
7-2	−	R	Read as 0.
1-0	LEAP	R/W	00 : leap year 01 : one year after leap year 10 : two years after leap year 11 : three years after leap year

23.3.3.10 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

23.3.3.11 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 second 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 Initialize alarm registers (Minute column, hour column, day column and day of the week column) as follows. Minus: 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 1: A read-modify-write operation cannot be performed for this register.

The setting of RTCPAGER<ENAALM> and <DIS1HZ>, <DIS16HZ>, <DS8HZ>, <DS4MZ>, <DS2MZ> used for alarm, 1Hz interrupt and 2Hz interrupt, 4Hz interrupt, 8Hz interrupt, 16Hz interrupt are shown as below.

Table 23-3 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

23.4 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.

23.4.1 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.

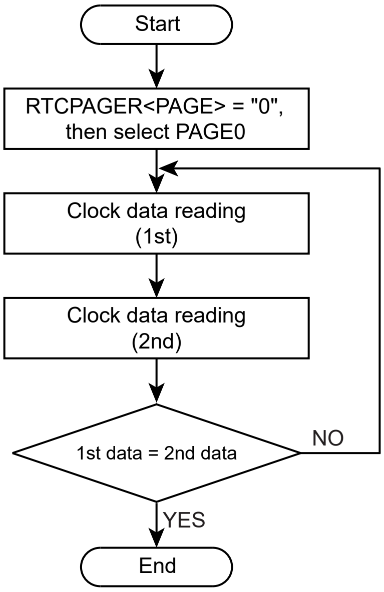


Figure 23-2 Flowchart of the clock data reading

23.4.2 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.

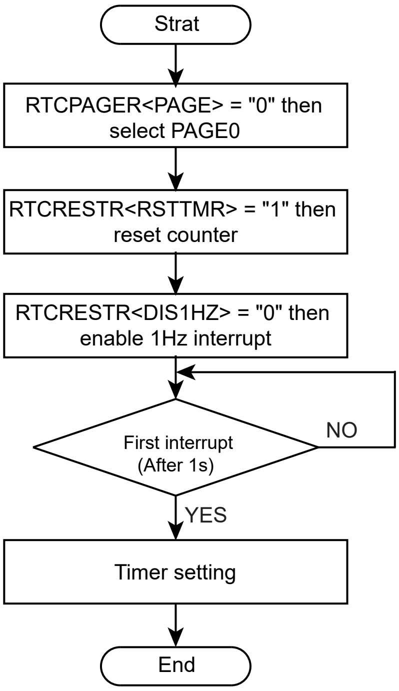

Strat
RTCPAGER<PAGE> = "0" then select PAGE0
RTCRESTR<RSTTMR> = "1" then reset counter
RTCRESTR<DIS1HZ> = "0" then enable 1Hz interrupt
First interrupt (After 1s)
NO
YES
Timer setting
End

Figure 23-3 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

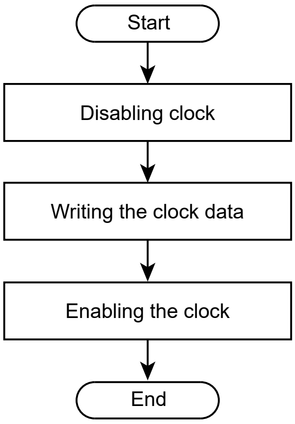

Start
Disabling clock
Writing the clock data
Enabling the clock
End

Figure 23-4 Flowchart of the disabling clock

23.4.3 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.

23.5 Alarm function

By writing "1" to RTCPAGER<PAGE>, the alarm function of the PAGE1 registers is enabled.

The INTRTC interrupt signal is falling edge triggered. Specify the falling edge as the active state in the CG Interrupt Mode Control Register.

23.5.1 The INTRTC interrupt is generated when the values of the PAGE0 clock register and the PAGE1 alarm register correspond

when the values of the PAGE0 clock register and the PAGE1 alarm register correspond, the INTRTC interrupt is generated.

The alarm settings are shown as follows.

Initialize the alarm with alarm prohibited. Write "1" to RTCRESTR<RSTALM>.

It makes the alarm setting to be 00 minus, 00 hour, 01 day and Sunday.

Setting alarm for minus, 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 generating INTRTC interrupt at noon (12:00) on Monday 5th.

		7	6	5	4	3	2	1	0	
RTCPAGER	←	0	0	0	0	1	0	0	1	Disable alarm, sets PAGE1
RTCRESTR	←	1	1	0	1	0	0	0	0	Initialize 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 minus
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 be become valid.

Note: To make the alarm work repeatedly (e.g. every Wednesday at 12:00), next alarm must be set during the INTETC interrupt routine that is generated when the time set for the alarm matches the RTC count.

24. 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 DVDD3.

24.1 Structure

Power-on-reset circuit consists of the reference voltage generation circuit, comparators, the LVD 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.

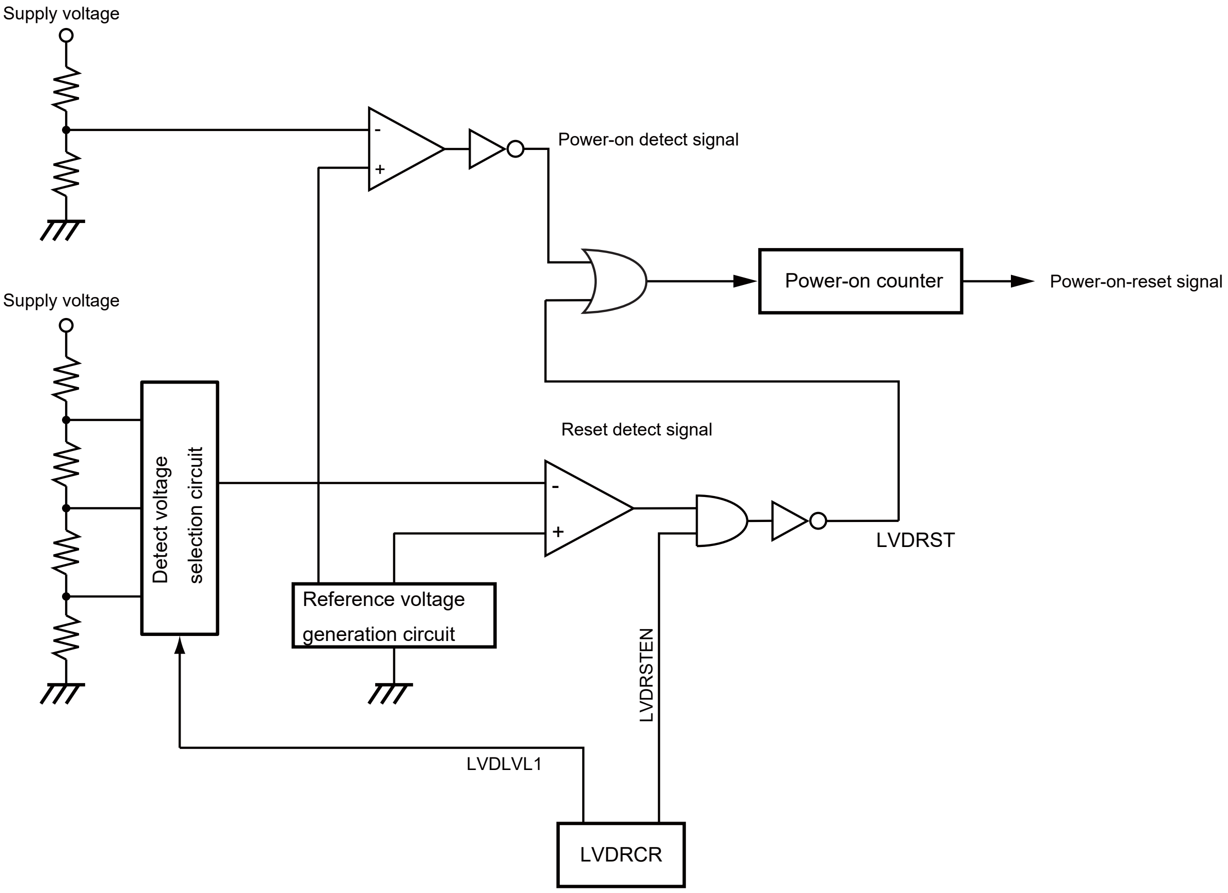

Supply voltage
Power-on detect signal
Power-on counter
Power-on-reset signal
Supply voltage
Reset detect signal
Detect voltage selection circuit
LVDRST
Reference voltage generation circuit
LVDRSTEN
LVDLVL1
LVDRCR

Figure 24-1 Power-on-reset circuit

For details of LVDRCR in LVD reset circuit, refer to Section "Voltage detection circuit (LVD)".

24.2 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 DVDD3 is over 2.3 ± 0.2 V.

If the power-on detection signal is released and the reset detection signal is also released, the power-on counter starts to operate. After waiting time (approximately 0.8ms) 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, TMPM367FDFG cannot operate properly.

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.

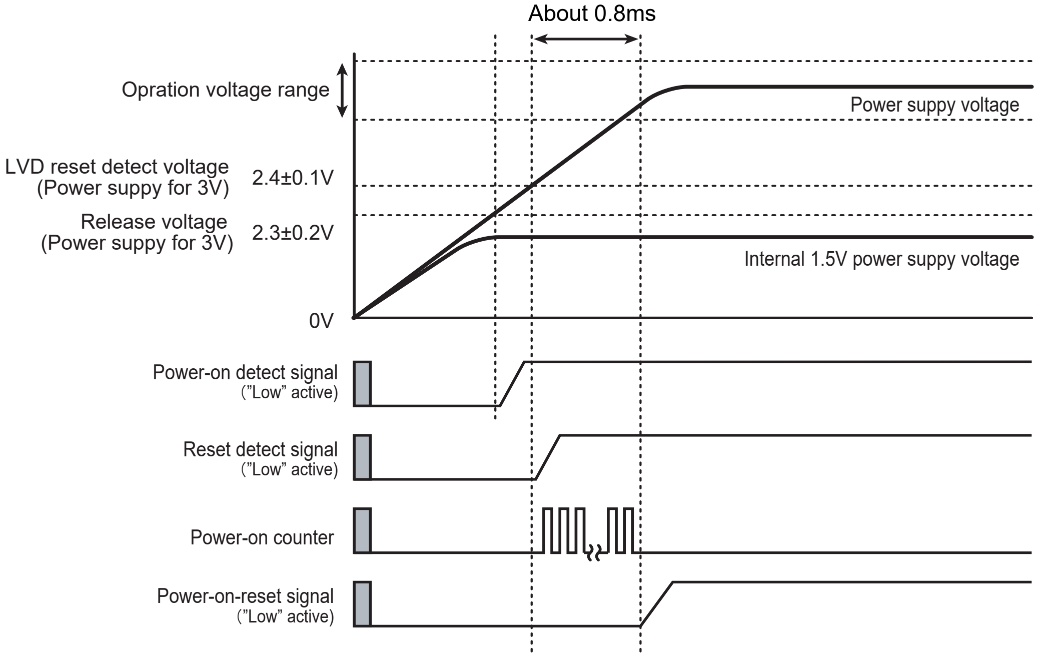

About 0.8ms
Opration voltage range
Power suppy voltage
LVD reset detect voltage
(Power suppy for 3V)
2.4±0.1V
Release voltage
(Power suppy for 3V)
2.3±0.2V
Internal 1.5V power suppy voltage
0V
Power-on detect signal
("Low" active)
Reset detect signal
("Low" active)
Power-on counter
Power-on-reset signal
("Low" active)

Figure 24-2 Power-on-reset operation timing

25. Low Voltage Detection Circuit (LVD)

Low voltage detection circuit generates a reset signal or an interrupt signal by detecting a decreasing/increasing voltage.

Supply voltage is indicated as DVDD3.

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.

25.1 Structure

The low voltage detection circuit consists of a reference voltage generation circuit, comparators and control registers.

Supply voltage is divided by a ladder resistor and input to the voltage selection circuit. In the voltage selection circuit, a voltage is chosen according to the detected voltage then compared with the reference voltage in the comparator. If the supply voltage is upper/lower than the detected voltage, a reset/interrupt signal occurs.

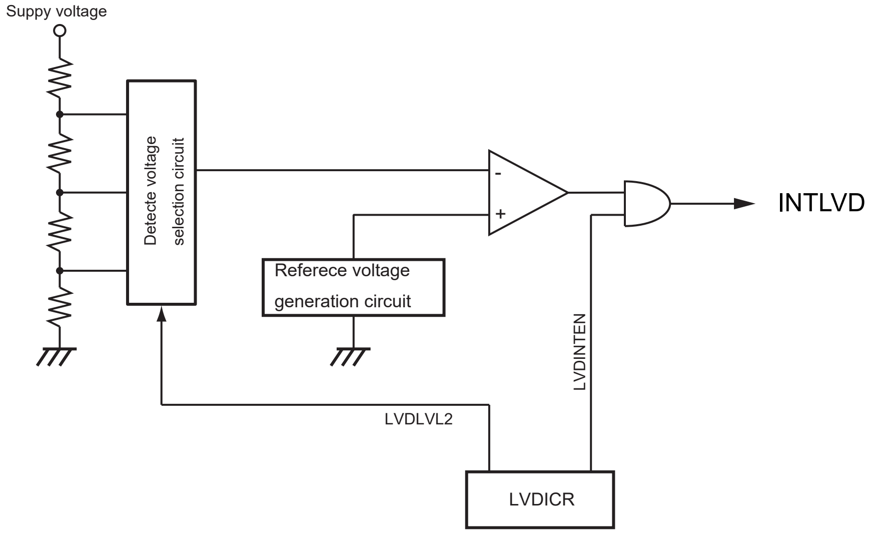

Suppy voltage
Detecte voltage selection circuit
-
+
Referece voltage generation circuit
INTLVD
LVDINTEN
LVDLVL2
LVDICR

Figure 25-1 Block diagram of LVD (described only LVD interrupt circuit)

Refer to Section "Power-on-reset circuit" about the LVD reset circuit block diagram.

25.2 Registers

25.2.1 Register List

Base Address = 0x400F_4000

Register name		Address(Base+)
LVD reset control register	LVDRCR	0x0000
LVD interrupt control register	LVDICR	0x0004
LVD status register	LVDSR	0x0008

25.2.2 LVDRCR (LVD 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	−	−	LVDRSTEN	−	LVDLVL1			LVDEN1
After reset	0	0	1	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31 - 6	−	R	Read as "0".
5	LVDRSTEN	R/W	Controls LVDRST output 0: Disabled 1: Enabled
4	−	R	Read as "0".
3 - 1	LVDLVL1[2:0]	R/W	Detected voltage 000: 2.4 ± 0.1V 001: 2.5 ± 0.1V 010: 2.6 ± 0.1V 011: 2.7 ± 0.1V 100: 2.8 ± 0.1V 101: 2.9 ± 0.1V 110: Reserved 111: Reserved
0	LVDEN1	R/W	Low voltage detection operation 0: Disabled 1: Enabled

Note: LVDRCR is initialized by power-on reset operation.

25.2.3 LVDICR (LVD interrupt 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	−	−	LVDINTEN	INTSEL	LVDLVL2			LVDEN2
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31 - 6	−	R	Read as "0".
5	LVDINTEN	R/W	Controls INTLVD output 0: Disabled 1: Enabled
4	INTSEL	R/W	INT generation condition 0: Only lower than the setting voltage when voltage decreasing. 1: Both lower and upper than the setting voltage when voltage decreasing.
3 - 1	LVDLVL2[2:0]	R/W	Detected voltage 000: 2.80± 0.1V 001: 2.85 ± 0.1V 010: 2.90 ± 0.1V 011: 2.95 ± 0.1V 100: 3.00 ±0.1V 101: 3.05 ± 0.1V 110: 3.10 ± 0.1V 111: 3.15 ± 0.1V
0	LVDEN2	R/W	Low voltage detection operation 0: Disabled 1: Enabled

Note: LVDICR is initialized by power-on reset, LVD reset and reset with $\overline{\text{RESET}}$ pin.

25.2.4 LVDSR (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	−	−	−	−	−	−	LVDST2	LVDST1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31 - 2	−	R	Read as "0".
1	LVDST2	R	Indicates LVDLVL2 low voltage detection status 0: Power supply voltage is upper than the detection voltage. 1: Power supply voltage is lower than the detection voltage.
0	LVDST1	R	Indicates LVDLVL1 low voltage detection status 0: Power supply voltage is upper than the detection voltage. 1: Power supply voltage is lower than the detection voltage.

Note: LVDSR is initialized by power-on-reset, LVD reset and reset with $\overline{\text{RESET}}$ pin.

25.3 Operation Description

25.3.1 Detection Voltage Selection and Enabling/Disabling the Operation

The LVDICR register sets the following; choosing the detection voltage, setting the operation to enable/disable, choosing output conditions and setting the output to enable/disable. The LVDICR register is initialized with the power-on reset, the LVD reset or the reset by RESET pin.

The LVDICR<LVDLVL2[2:0]> bit chooses the detection voltage. If LVDICR<LVDEN2> is set to "1", low voltage detection operation is enabled.

Note: While supply voltage is lower than the detection voltage, if low voltage detection operation is enabled, INTLVD will generate at this timing.

25.3.2 Lower Voltage Detection

If supply voltage is lower than the detection voltage level, INTLVD generates. When the LVDICR<INTSEL> is set to "1", if supply voltage is upper than the detection voltage, INTLVD generates.

After lower voltage detection, to detect INTLVD is required a certain time. If this period is shorter than expected, INTLVD may not generate.

If supply voltage is lower than 2.7V, MCU operation is not guaranteed. In this case, supply voltage must be decreased to 0V and then power-on.

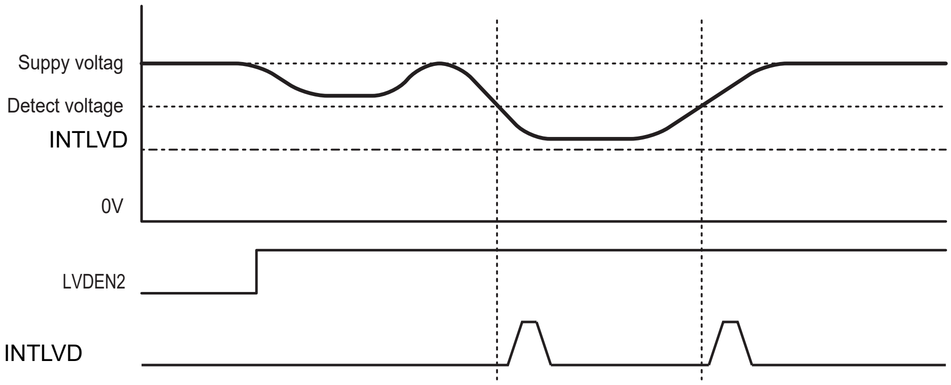


Figure 25-2 low voltage detection Timing

26. Ocsillation 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 subharmonic 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. TMPM367FDFG uses internal high-speed oscillator clock 2 (f_{IHOSC2}) as a reference and the target clock are an internal high-speed oscillator clock 1 (f_{IHOSC1}) and an external high-speed oscillator clock (f_{EHOSC}) . They are selected by CGOSCCR<OSCSEL>.

Note: It is not guaranteed that OFD can detect all defects at any time, and it is not a circuit to measure error frequency.

26.1 Block diagram

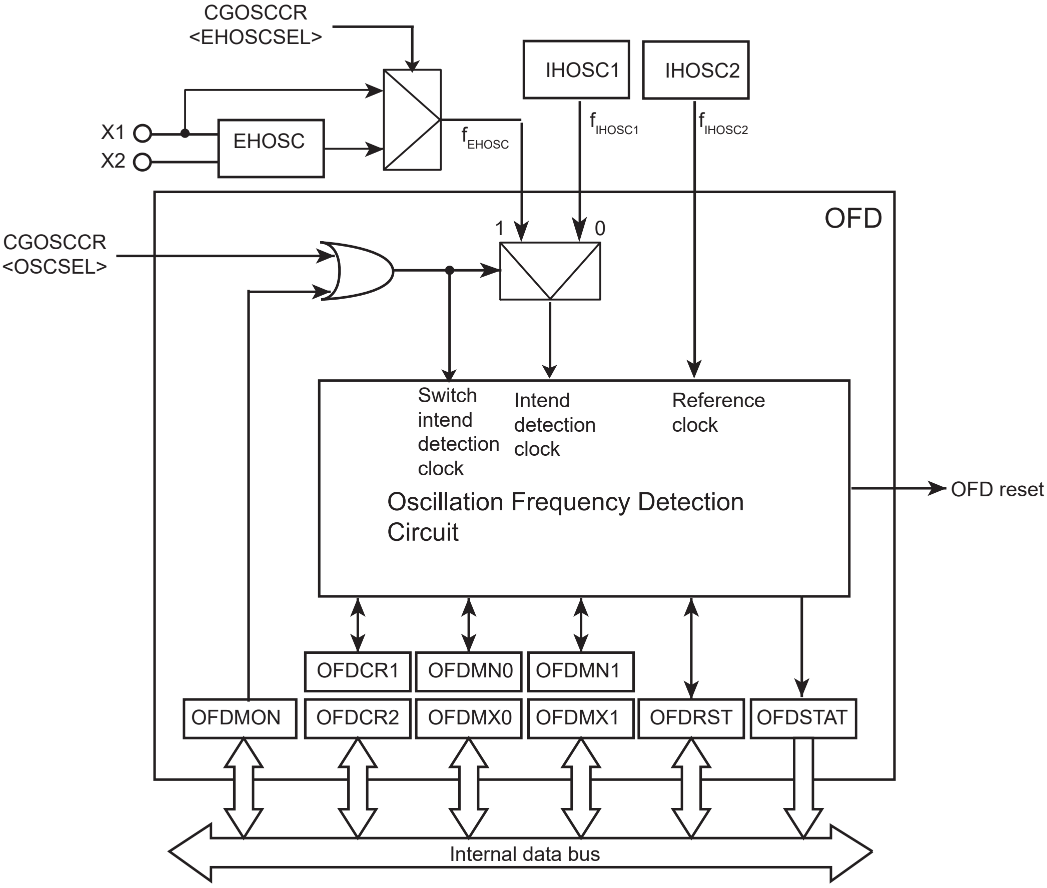

CGOSCCR
<EHOSCSEL>
IHOSC1
IHOSC2
X1
X2
EHOSC
fEHOSC
fIHOSC1
fIHOSC2
OFD
CGOSCCR
<OSCSEL>
1
0
Switch intend detection clock
Intend detection clock
Reference clock
Oscillation Frequency Detection Circuit
OFD reset
OFDCR1
OFDMN0
OFDMN1
OFDMON
OFDCR2
OFDMX0
OFDMX1
OFDRST
OFDSTAT
Internal data bus

Figure 26-1 Ocsillation Frequency Detector Block diagram

26.2 Registers

26.2.1 Register List

Base Address = 0x400F_1000

Register name		Address(Base+)
Control register 1	OFDCR1	0x0000
Control register 2	OFDCR2	0x0004
Lower detection frequency setting register0 (IHOSC1)	OFDMN0	0x0008
Lower detection frequency setting register1 (EHOSC)	OFDMN1	0x000C
Higher detection frequency setting register0 (IHOSC1)	OFDMX0	0x0010
Higher detection frequency setting register1 (EHOSC)	OFDMX1	0x0014
Reset control register	OFDRST	0x0018
Status register	OFDSTAT	0x001C
External high frequency oscillaion clock monitor register	OFDMON	0x0020

Note: Access to the "Reserved" area is prohibited.

26.2.1.1 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.

26.2.1.2 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.

26.2.1.3 OFDMN0 (Lower detection frequency 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	-	-	-	-	-	-	-	OFDMN0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	OFDMN0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMN0[8:0]	R/W	Sets internal lower detection frequency.

Note: Writing to the register of OFDMN0 is protected while OFD circuit is operating.

26.2.1.4 OFDMN1 (Lower detection frequency 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	-	-	-	-	-	-	-	OFDMN1
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	OFDMN1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMN1[8:0]	R/W	Sets external lower detection frequency.

Note: Writing to the register of OFDMN1 is protected while OFD circuit is operating.

26.2.1.5 OFDMX0 (Higher detection frequency 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	-	-	-	-	-	-	-	OFDMX0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	OFDMX0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMX0[8:0]	R/W	Sets internal higher detection frequency.

Note: Writing to the register of OFDMX0 is protected while OFD circuit is operating.

26.2.1.6 OFDMX1 (Higher detection frequency 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	-	-	-	-	-	-	-	OFDMX1
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	OFDMX1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMX1[8:0]	R/W	Sets external higher detection frequency.

Note: Writing to the register of OFDMX1 is protected while OFD circuit is operating.

26.2.1.7 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.

26.2.1.8 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: Run 1: Stop
0	FRQERR	R	Error detecting flag 0: No Error 1: Error

26.2.1.9 OFDMON (External high frequency odcillation clock monitor 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	-	-	-	-	-	-	-	OFDMON
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Description
31-1	−	R	Read as 0.
0	OFDMON	R/W	Select intended detection frequency 0: Normal detection mode Target clock which is selected by CGOSCCR<OSCSEL> 1: Monitor mode Monitor the oscillation condition of EHOSC when system is opearated by IHOSC1

Note: Writing to the register of OFDRST is protected while OFD circuit is operating.

26.3 Operational Description

26.3.1 Setting

All register except OFDCR1 can not be written by reset. They are can be written by writing "0xF9" to OFDCR1.

The range of detection frequency is setting by OFDMX and OFDMN for each clock. The reset generation is set for enabling or disabling by OFDRST, and writing "0xE4" to OFDCR2 enables the oscillation frequency detection.

To protect the mistaken writting, shout be written "0x06" to OFDCR1. And the register should be modified when OFD is sttoped.

26.3.2 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 256/reference clock frequency.

The oscillation condition can be confirmed by monitor function before system clock is chaged to the external oscilattion clock EHOSC. At this time, disable reset generation and monitor the oscillation condition by OFDSTAT<FRQERR>. Since the OFDSTAT<OFDBSY> changes to operating until the state of OFDSTAT<FRQERR> changes valid, time length as two cycle of detecting clock is needed.

When generating reset is enabled, OFD generates reset if the target clock frequency exceeds the frequency limit set by OFDMN and OFDMX. From detection of abnormal to reset generation, time length as two cycle of detecting clock is needed.

The reset generated by OFD does not make itself and OFD continues detected operation. Therefore, fosc is initialized to IHOSC1 and the target clock changes to IHOSC1 and detected operation is continued. When the target clock frequency is IHOSC1 and the reset generated by OFD, the reset is generated continuously until OFD detects correctly. From detection of abnormal to reset generation, time length as two cycle of detecting clock is needed.

Note: There are several factors of reset. Clock generator register CGRSTFLG can confirm the factors. For details of CGRSTFLG see chapter "exception."

26.3.3 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.

The upper and lower limit of detecting frequency are calcuratrd by the maxmum error of a target clock and a reference clock.

By the way of rounding the calcurated result when OFDMX and OFDMN are decided, the upper and lower limit of detecting and undetecting range shown as follows. The way of rouding is selected depending on the uneveness of the detected clock.

- In case of rounding up OFDMX and rouding down OFDMN

  The target clock is higher than the upper limit of undetecting ragne and lower than the lower limit of undetecting range.
- In case of rounding down OFDMX and rouding down OFDMM

  The target clock is lower than the upper limit of undetecting ragne and higher than the lower limit of undetecting range.

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 ±3% (undetecting range) and the reference clock error is ± 5%. In this example, OFDMX is rounded up and OFDMX is rounded down.

target clock	10MHz ± 3%	Max.10.3MHz	-------- c
		Min. 9.7MHz	-------- b
reference clock	10MHz ± 5%	Max.10.5MHz	-------- f
		Min. 9.5MHz	-------- e

OFDMX = c ÷ e × 64 = 69.39... = 70 (Rounding up to nearest decimal) = 0x46

OFDMN = b ÷ f × 64 = 59.12... = 59 (Rounding down to nearest decimal) = 0x3B

At this time, the detecting range is calculated shown below.

a = e × OFDMN ÷ 64 = 8.76

d = f × OFDMX ÷ 64 = 11.5

And the undetecting range is calculated shown below.

g = e × OFDMX ÷ 64 = 10.4

h = f × OFDMN ÷ 64 = 9.68

Setting "0x46" to the register OFDMX and "0x3B" to the register OFDMN, when the target clock of higher than 11.5MHz or lower than 8.76MHz is detected, the oscillation frequency detector outputs a reset signal. And whe the target clock of higher than 9.68MHz and lower than 10.4MHz is detected, the oscillation frequency detector does not output a reset signal.

Figure 26-2 shows the detection or undetectable and detectable frequency range.

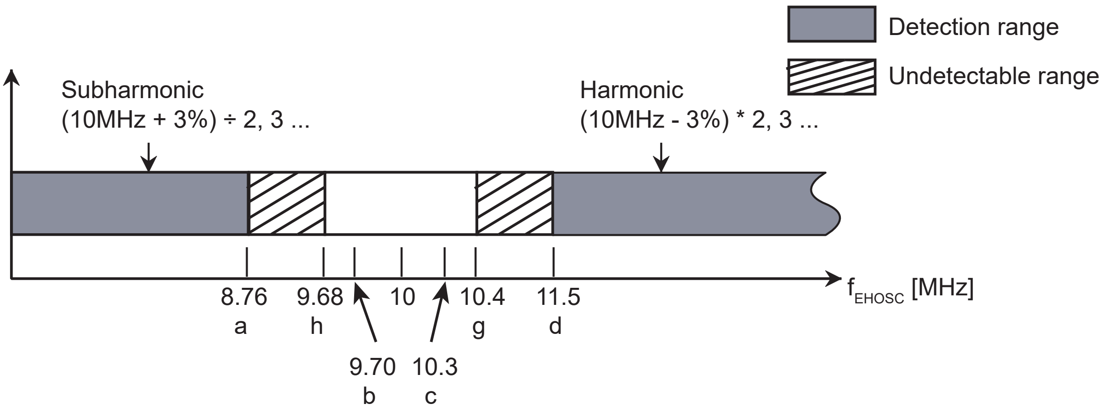

Detection range
Undetectable range
Subharmonic
(10MHz + 3%) ÷ 2, 3 ...
Harmonic
(10MHz - 3%) * 2, 3 ...
8.76
a
9.68
h
10
10.4
g
11.5
d
9.70
b
10.3
c
f_{EHOSC} [MHz]

Figure 26-2 Example of detection frequency range (in case of 10MHz)

26.3.4 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.

26.3.5 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.

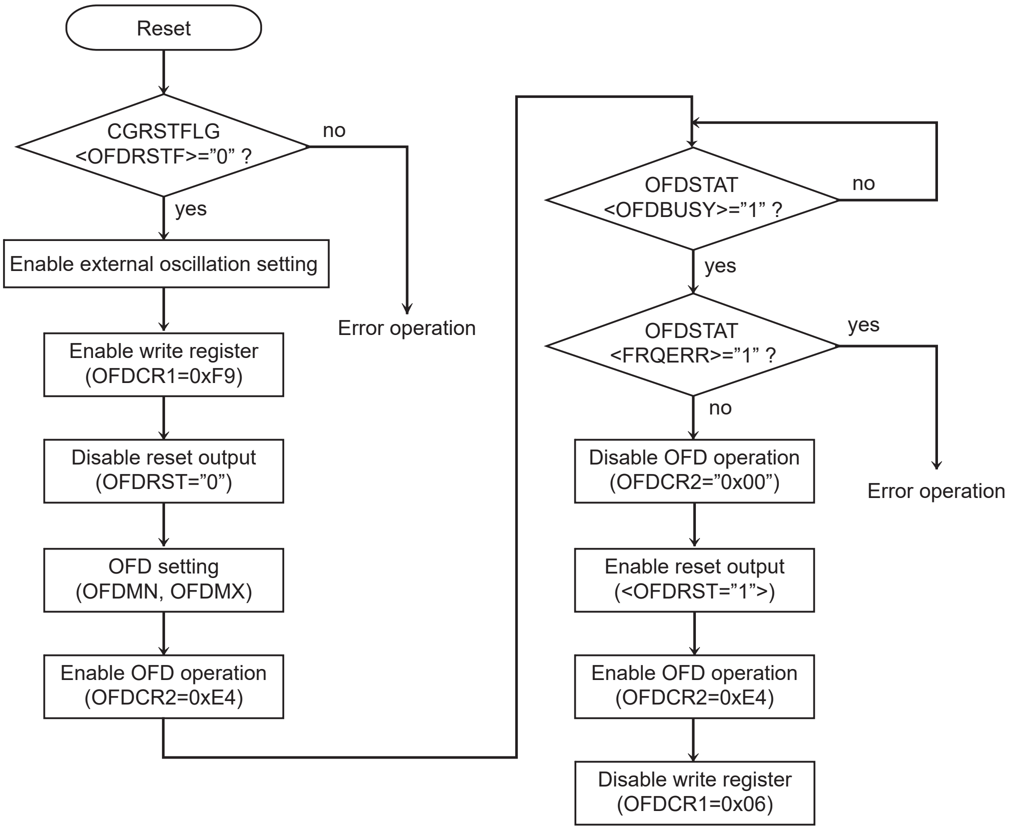

Reset
CGRSTFLG <OFDRSTF>="0" ?
no
yes
Enable external oscillation setting
Error operation
Enable write register (OFDCR1=0xF9)
Disable reset output (OFDRST="0")
OFD setting (OFDMN, OFDMX)
Enable OFD operation (OFDCR2=0xE4)
OFDSTAT <OFDBUSY>="1" ?
no
yes
OFDSTAT <FRQERR>="1" ?
yes
no
Error operation
Disable OFD operation (OFDCR2="0x00")
Enable reset output (<OFDRST="1">)
Enable OFD operation (OFDCR2=0xE4)
Disable write register (OFDCR1=0x06)

Figure 26-3 Example of operational procedure

27. 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.

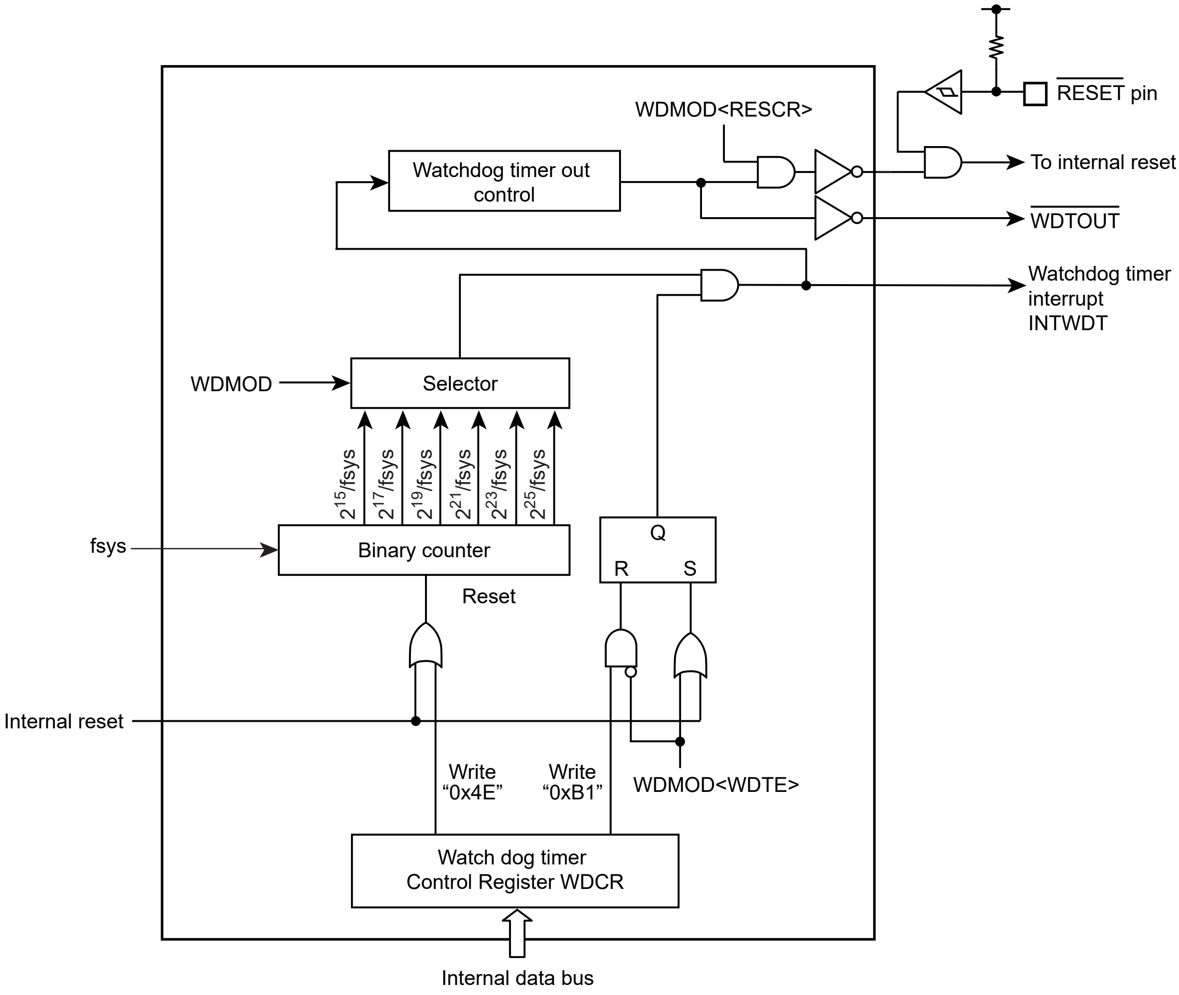
Note: INTWDT interrupt is a factor of the non-maskable interrupts (NMI).

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 ($\overline{\text{WDTOUT}}$).

27.1 Configuration

Figure 27-1shows the block diagram of the watchdog timer.



$\overline{\text{RESET}}$ pin
WDMOD<RESCR>
Watchdog timer out control
To internal reset
$\overline{\text{WDTOUT}}$
Watchdog timer interrupt INTWDT
WDMOD
Selector
2^{15}/fsys
2^{17}/fsys
2^{19}/fsys
2^{21}/fsys
2^{23}/fsys
2^{25}/fsys
fsys
Binary counter
Q
R
S
Reset
Internal reset
Write "0x4E"
Write "0xB1"
WDMOD<WDTE>
Watch dog timer Control Register WDCR
Internal data bus

Figure 27-1 Block Diagram of the Watchdog Timer

27.2 Register

The followings are the watchdog timer control registers and addresses.

Base Address = 0x400F_2000

Register name		Address(Base+)
Watchdog Timer Mode Register	WDMOD	0x0000
Watchdog Timer Control Register	WDCR	0x0004

27.2.1 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 27-1) 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 27-1 Detection time of watchdog timer (fc = 80MHz)

Clock gear value CGSYSCR<GEAR[2:0]>	WDMOD<WDTP[2:0]>					
	000	001	010	011	100	101
000 (fc)	0.41 ms	1.64 ms	6.55 ms	26.21 ms	104.86 ms	419.43 ms
100 (fc/2)	0.82 ms	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms
101 (fc/4)	1.64 ms	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s
110 (fc/8)	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s
111 (fc/16)	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s	6.71 s

27.2.2 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

27.3 Operations

27.3.1 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 ($\overline{\text{WDTOUT}}$).

27.3.2 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.

- STOP1 mode
- STOP2 mode

Also, the binary counter is automatically stopped during debug mode.

27.4 Operation when malfunction (runaway) is detected

27.4.1 INTWDT interrupt generation

In the Figure 27-2 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($\overline{\text{WDTOUT}}$).

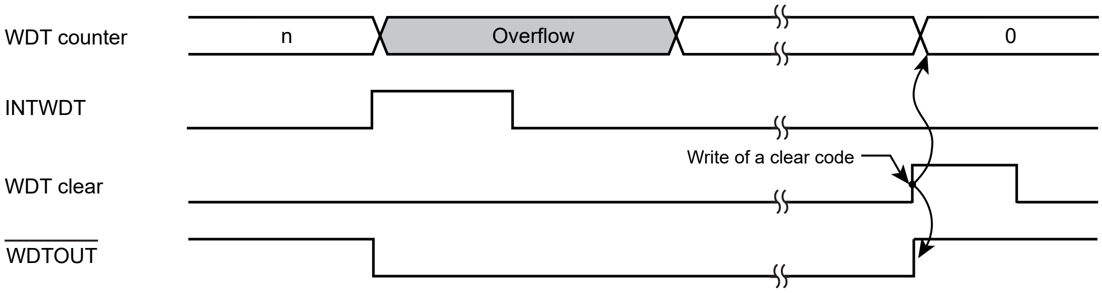


Figure 27-2 INTWDT interrupt generation

27.4.2 Internal reset generation

Figure 27-3 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.

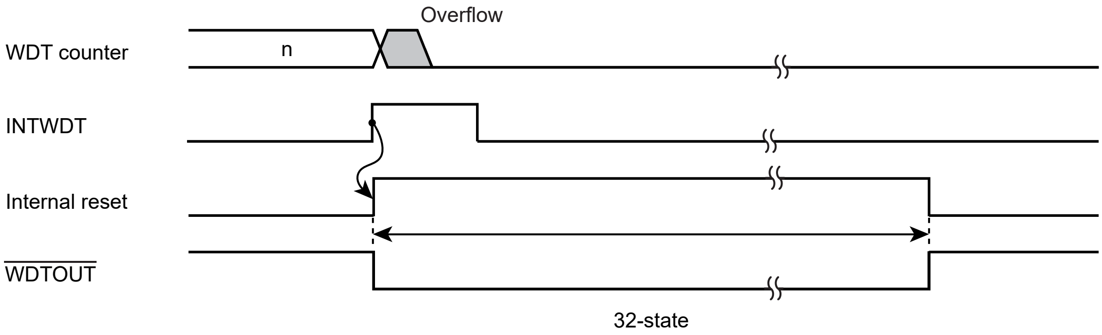

Overflow
WDT counter
n
INTWDT
Internal reset
WDTOUT
32-state

Figure 27-3 Internal reset generation

27.5 Control register

The watchdog timer (WDT) is controlled by two control registers WDMOD and WDCR.

27.5.1 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.

27.5.2 Watchdog Timer Control Register(WDCR)

This is a register for disabling the watchdog timer function and controlling the clearing function of the binary counter.

27.5.3 Setting example

27.5.3.1 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).

27.5.3.2 Enabling control

Set WDMOD <WDTE> to "1".

		7	6	5	4	3	2	1	0	
WDMOD	←	1	−	−	−	−	−	−	−	Set <WDTE> to "1".

27.5.3.3 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).

27.5.3.4 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	−	−	−	−

28. Flash

This section describes the hardware configuration and operation of the flash memory.

28.1 Flash Memory

28.1.1 Features

1. Memory capacity

   TMPM367FDFG contains flash memory. The memory sizes and configurations are shown in the table below.

   Independent write access to each block is available. When the CPU is to access the internal flash memory, 32-bit data bus width is used.

2. Write / erase time

   Writing is executed per page. TMPM367FDFG contains 128 words.

   Page writing requires 1.25ms (typical) regardless of number of words.

   A block erase requires 0.1 sec. (typical).

   The following table shows write and erase time per chip.

Product Name	Memory size	Block Configuration			# of words	Write time	Erase time
		128 KB	64 KB	32 KB			
TMPM367FDFG	512 KB	3	1	2	128	1.28 s	0.4 s

Note: **The above values are theoretical values not including data transfer time. The write time per chip depends on the write method to be used by users.**

3. Programming method

   There are two types of the onboard programming mode for users to program (rewrite) the device while it is mounted on the user's board:

   a. User boot mode

      The use’s original rewriting method can be supported.

   b. Single boot mode

      The rewriting method to use serial data transfer (Toshiba's unique method) can be supported.

4. Rewriting method

The flash memory included in this device is generally compliant with the applicable JEDEC standards except for some specific functions. Therefore, if a user is currently using an external flash memory device, it is easy to implement the functions into this device. Furthermore, the user is not required to build his/her own programs to realize complicated write and erase functions because such functions are automatically performed using the circuits already built-in the flash memory chip.

JEDEC compliant functions	Modified, added, or deleted functions
・ Automatic programming ・ Automatic chip erase ・ Automatic block erase ・ Data polling / toggle bit	<Modified> Block protect (only software protection is supported) <Deleted> Erase resume - suspend function

5. Protect/ Security Function

This device is also implemented with a read-protect function to inhibit reading flash memory data from any external writer device. On the other hand, rewrite protection is available only through command-based software programming; any hardware setting method to apply +12VDC is not supported. See the chapter "ROM protection" for details of ROM protection and security function.

Note: **If a password is set to 0xFF (erased data), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.**

28.1.2 Block Diagram of the Flash Memory Section

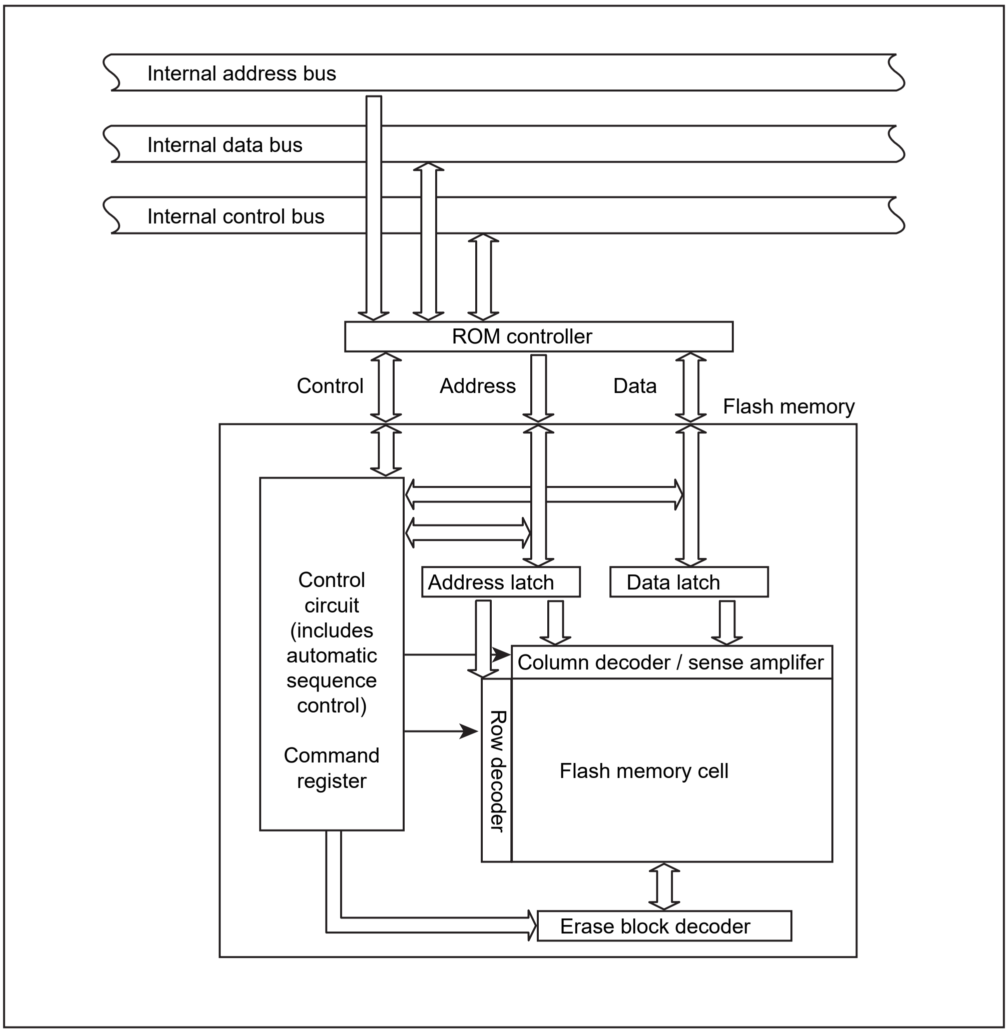

Internal address bus
Internal data bus
Internal control bus
ROM controller
Control
Address
Data
Flash memory
Control circuit (includes automatic sequence control)
Command register
Address latch
Data latch
Column decoder / sense amplifer
Row decoder
Flash memory cell
Erase block decoder

Figure 28-1 Block Diagram of the Flash Memory Section

28.2 Operation Mode

This device has three operation modes including the mode not to use the internal flash memory.

Table 28-1 Operation modes

Operation mode	Operation details
Single chip mode	After reset is cleared, it starts up from the internal flash memory.
Normal mode	In this operation mode, two different modes, i.e., the mode to execute user application programs and the mode to rewrite the flash memory onboard the user's set, are defined.The former is referred to as "normal mode" and the latter "user boot mode".
User boot mode	A user can uniquely configure the system to switch between these two modes. For example, a user can freely design the system such that the normal mode is selected when the port "A0" is set to "1" and the user boot mode is selected when it is set to "0". A user should prepare a routine as part of the application program to make the decision on the selection of the modes.
Single boot mode	After reset is cleared, it starts up from the internal Boot ROM (Mask ROM). In the Boot ROM, an algorithm to enable flash memory rewriting on the user's set through the serial port of this device is programmed. By connecting to an external host computer through the serial port, the internal flash memory can be programmed by transferring data in accordance with predefined protocols.

Among the flash memory operation modes listed in the above table, the User Boot mode and the Single Boot mode are the programmable modes. These two modes, the User Boot mode and the Single Boot mode, are referred to as "Onboard Programming" modes where onboard rewriting of internal flash memory can be made on the user's set.

Either the Single Chip or Single Boot operation mode can be selected by externally setting the level of the $\overline{\text{BOOT}}$ (PB6) pin while the device is in reset status.

Table 28-2 Operating Mode Setting

Operation mode	Pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$ (PB6)
Single chip mode	0 → 1	1
Single boot mode	0 → 1	0

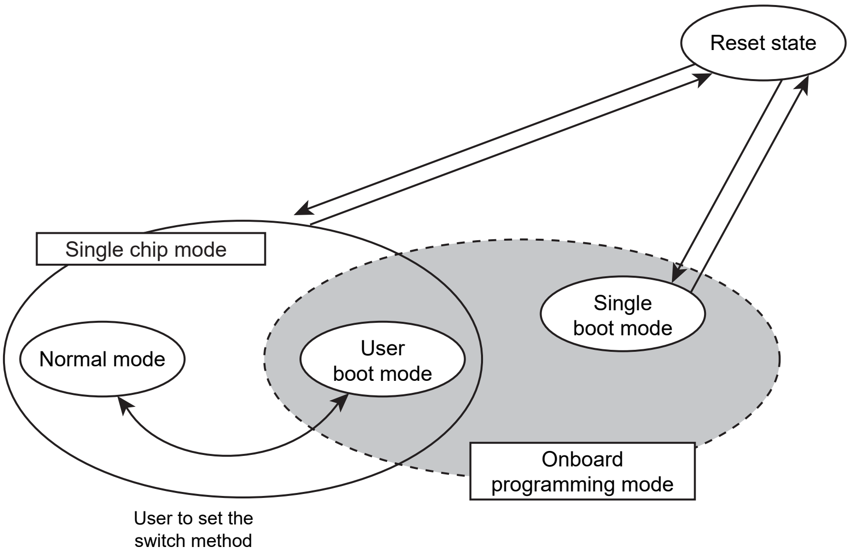


Figure 28-2 Mode Transition Diagram

28.2.1 Reset

Regarding to reset, refer to "Reset Operation".

Note: **While flash auto programming or erasing is in progress, at least 0.5 μs of reset period is required regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.**

28.2.2 User Boot Mode (Single chip mode)

User Boot mode is to use flash memory programming routine defined by users. It is used when the data transfer buses for flash memory program code on the old application and for serial I/O are different. It operates at the single chip mode; therefore, a switch from normal mode in which user application is activated at the single chip mode to User Boot Mode for programming flash is required. Specifically, add a mode judgment routine to a reset program in the user application.

The condition to switch the modes needs to be set by using the I/O of TMPM367FDFG in conformity with the user's system setup condition. Also, flash memory programming routine that the user uniquely makes up needs to be set in the new application. This routine is used for programming after being switched to User Boot Mode. The execution of the programming routine must take place while it is stored in the area other than the flash memory since the data in the internal flash memory cannot be read out during delete / writing mode. Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations. Be sure not to cause any exceptions including a non-maskable while User Boot Mode.

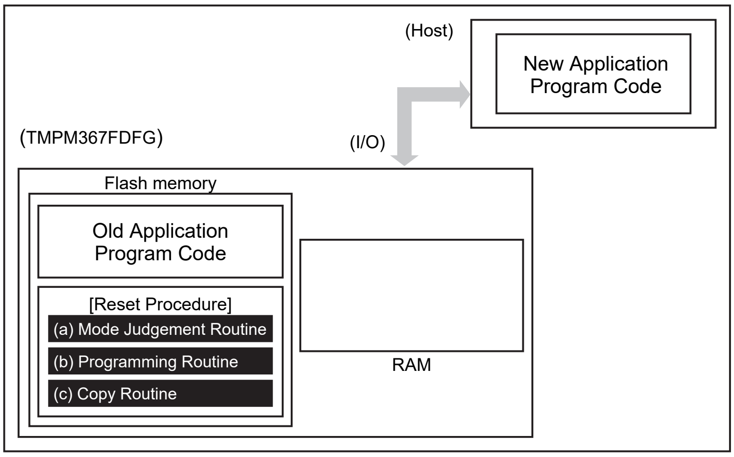
(1-A) and (1-B) are the examples of programming with routines in the internal flash memory and in the external memory. For a detailed description of the erase and program sequence, refer to "28.3 On-board Programming of Flash Memory (Rewrite/Erase)".

28.2.2.1 (1-A) Method 1: Storing a Programming Routine in the Flash Memory

(1) Step-1

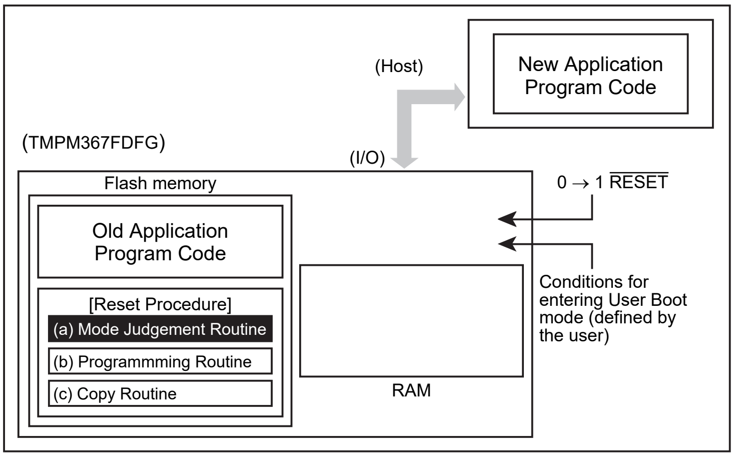
Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM367FDFG on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine:	Code to determine whether or not to switch to User Boot mode
(b) Programming routine:	Code to download new program code from a host controller and re-program the flash memory
(c) Copy routine:	Code to copy the data described in (b) from the TMPM367FDFG flash memory to either the TMPM367FDFG on-chip RAM or external memory device.


(Host)
New Application Program Code
(I/O)
(TMPM367FDFG)
Flash memory
Old Application Program Code
[Reset Procedure]
(a) Mode Judgement Routine
(b) Programming Routine
(c) Copy Routine
RAM

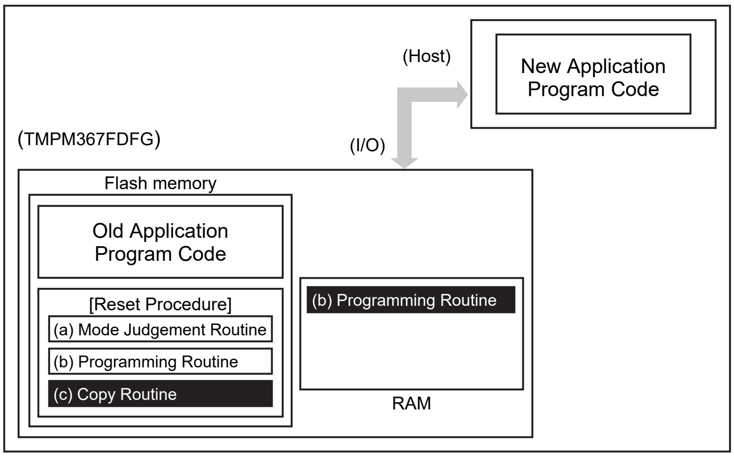
(2) Step-2

The following description is the case that programming routines are installed in the reset processing program. After $\overline{\text{RESET}}$ pin is released, the reset procedure determines whether to put the TMPM367FDFG flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be not used while in User Boot mode.)


(Host)
New Application Program Code
(TMPM367FDFG)
(I/O)
Flash memory
0 → 1 RESET
Old Application Program Code
[Reset Procedure]
(a) Mode Judgement Routine
(b) Programmming Routine
(c) Copy Routine
RAM
Conditions for entering User Boot mode (defined by the user)

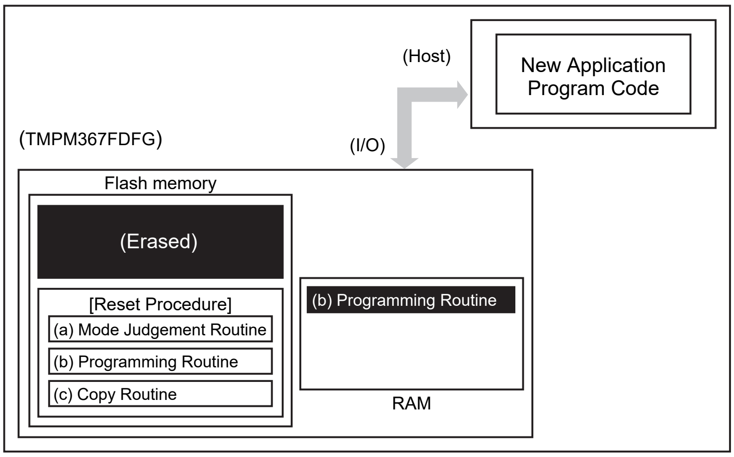
(3) Step-3

Once transition to User Boot mode is occurred, execute the copy routine (c) to copy the flash programming routine (b) to the TMPM367FDFG on-chip RAM.


(Host)
New Application Program Code
(TMPM367FDFG)
(I/O)
Flash memory
Old Application Program Code
[Reset Procedure]
(a) Mode Judgement Routine
(b) Programming Routine
(c) Copy Routine
(b) Programming Routine
RAM

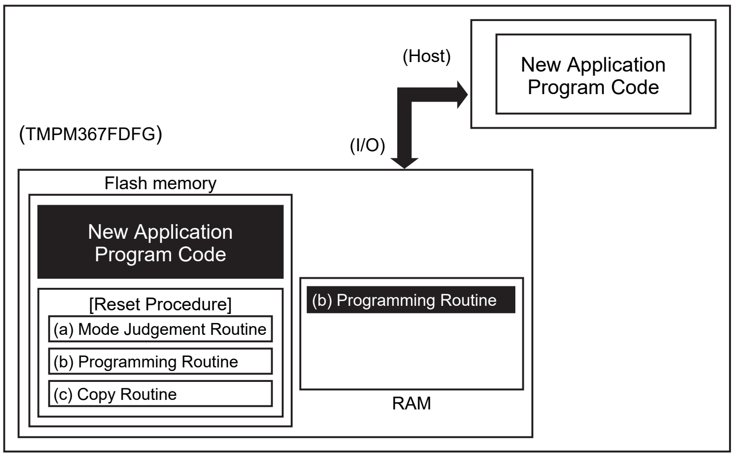
(4) Step-4

Jump program execution to the flash programming routine in the on-chip RAM to clear write or erase protection and erase a flash block containing the old application program code.


(Host)
New Application Program Code
(TMPM367FDFG)
(I/O)
Flash memory
(Erased)
[Reset Procedure]
(a) Mode Judgement Routine
(b) Programming Routine
(c) Copy Routine
(b) Programming Routine
RAM

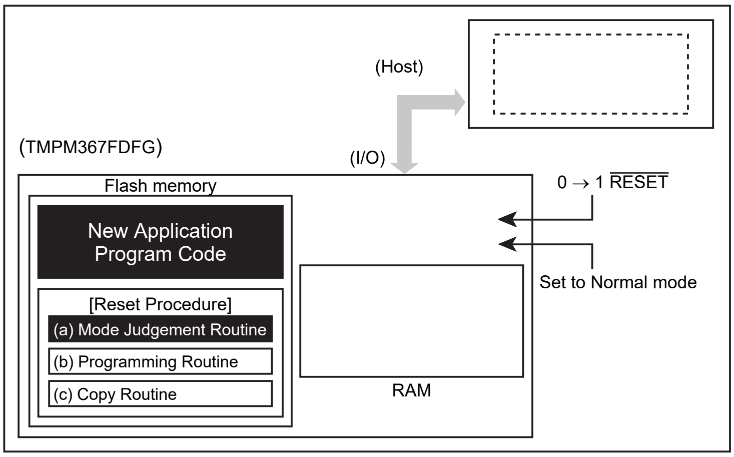
(5) Step-5

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user's program area must be set.


(Host)
New Application Program Code
(TMPM367FDFG)
(I/O)
Flash memory
New Application Program Code
[Reset Procedure]
(a) Mode Judgement Routine
(b) Programming Routine
(c) Copy Routine
(b) Programming Routine
RAM

(6) Step-6

Set $\overline{\text{RESET}}$ to "0" to reset the TMPM367FDFG. Upon reset, the on-chip flash memory is set to Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.


(Host)
(TMPM367FDFG)
(I/O)
Flash memory
0 → 1 RESET
New Application Program Code
Set to Normal mode
[Reset Procedure]
(a) Mode Judgement Routine
(b) Programming Routine
(c) Copy Routine
RAM

28.2.2.2 (1-B) Method 2: Transferring a Programming Routine from an External Host

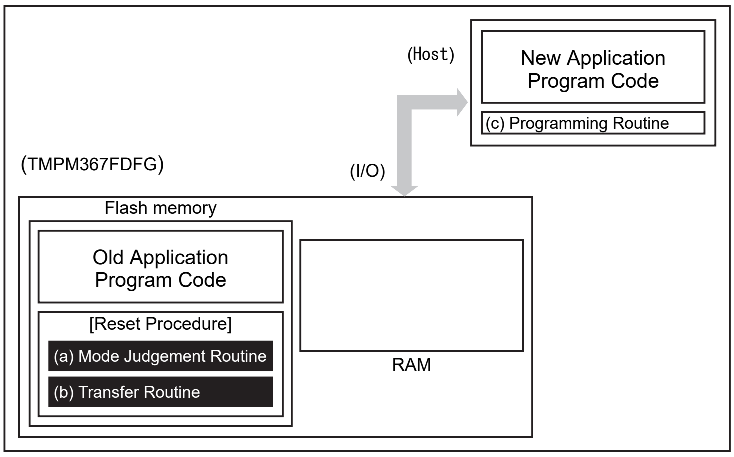
(1) Step-1

Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM367FDFG on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine:	Code to determine whether or not to switch to User Boot mode
(b) Transfer routine:	Code to download new program code from a host controller

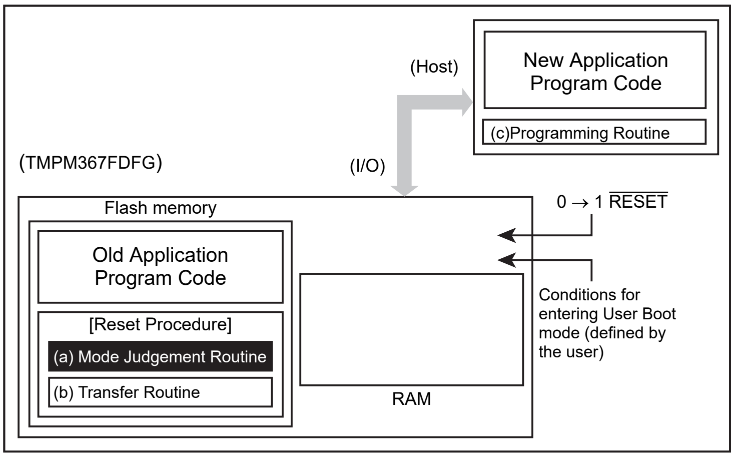
Also, prepare a programming routine shown below on the host controller:

(c) Programming routine:	Code to download new program code from an external host controller and re-program the flash memory


(Host)
New Application Program Code
(c) Programming Routine
(TMPM367FDFG)
(I/O)
Flash memory
Old Application Program Code
[Reset Procedure]
(a) Mode Judgement Routine
(b) Transfer Routine
RAM

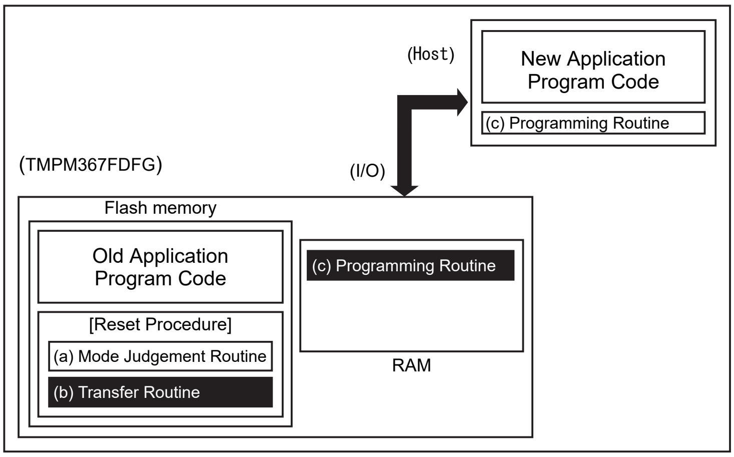
(2) Step-2

The following description is the case that programming routines are installed in the reset processing program. After RESET is released, the reset procedure determines whether to put the TMPM367FDFG flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be not used while in User Boot mode).


(Host)
New Application Program Code
(c)Programming Routine
(TMPM367FDFG)
(I/O)
Flash memory
0 → 1 $\overline{\text{RESET}}$
Old Application Program Code
[Reset Procedure]
(a) Mode Judgement Routine
(b) Transfer Routine
RAM
Conditions for entering User Boot mode (defined by the user)

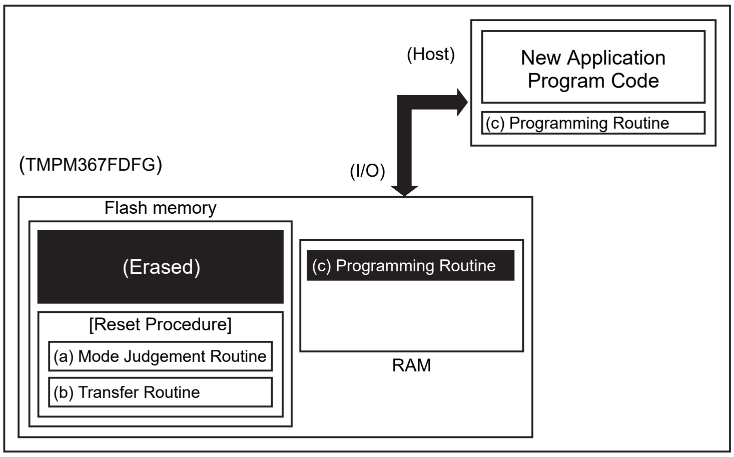
(3) Step-3

Once User Boot mode is entered, execute the transfer routine (b) to download the flash programming routine (c) from the host controller to the TMPM367FDFG on-chip RAM.


(Host)
New Application Program Code
(c) Programming Routine
(TMPM367FDFG)
(I/O)
Flash memory
Old Application Program Code
(c) Programming Routine
[Reset Procedure]
(a) Mode Judgement Routine
(b) Transfer Routine
RAM

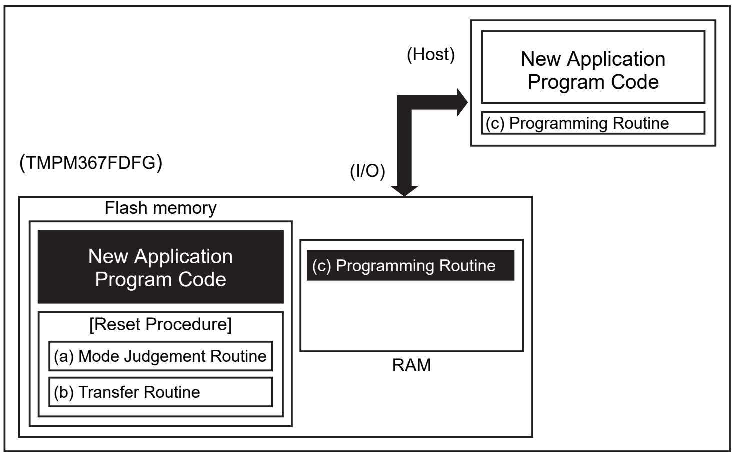
(4) Step-4

Jump program execution to the flash programming routine in the on-chip RAM to clear write or erase protection and erase a flash block containing the old application program code.


(Host)
New Application Program Code
(c) Programming Routine
(TMPM367FDFG)
(I/O)
Flash memory
(Erased)
[Reset Procedure]
(a) Mode Judgement Routine
(b) Transfer Routine
(c) Programming Routine
RAM

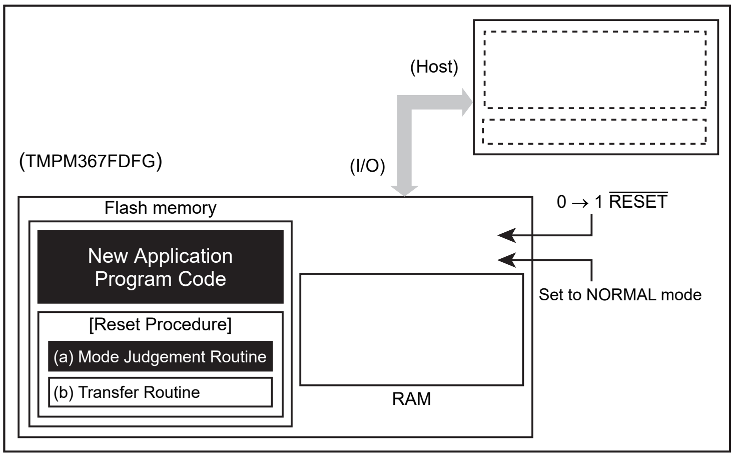
(5) Step-5

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user program area must be set.


(Host)
New Application Program Code
(c) Programming Routine
(TMPM367FDFG)
(I/O)
Flash memory
New Application Program Code
[Reset Procedure]
(a) Mode Judgement Routine
(b) Transfer Routine
(c) Programming Routine
RAM

(6) Step-6

Set $\overline{\text{RESET}}$ to "0" low to reset the TMPM367FDFG. Upon reset, the on-chip flash memory is set to Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.


(Host)
(TMPM367FDFG)
(I/O)
Flash memory
0 → 1 $\overline{\text{RESET}}$
New Application Program Code
Set to NORMAL mode
[Reset Procedure]
(a) Mode Judgement Routine
(b) Transfer Routine
RAM

28.2.3 Single Boot Mode

In Single Boot mode, the flash memory can be re-programmed by using a program contained in the TMPM367FDFG on-chip boot ROM. This boot ROM is a masked ROM. When Single Boot mode is selected upon reset, the boot ROM is mapped to the address region including the interrupt vector table while the flash memory is mapped to an address region different from it.

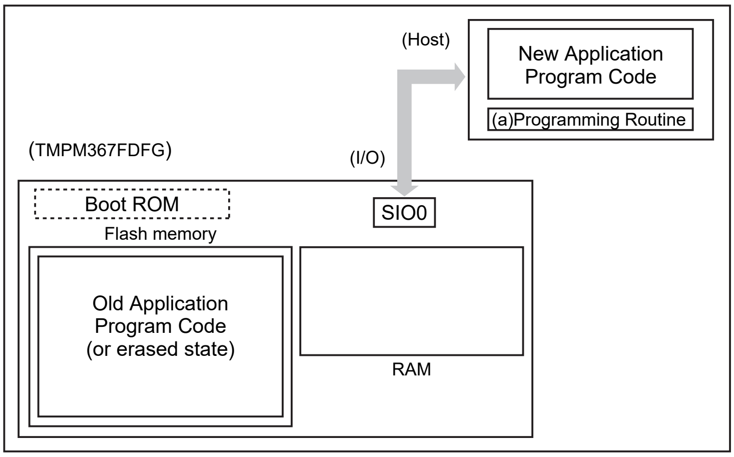
Single Boot mode allows for serial programming of the flash memory. Channel 0 of the SIO (SIO0) of the TMPM367FDFG is connected to an external host controller. Via this serial link, a programming routine is downloaded from the host controller to the TMPM367FDFG on-chip RAM. Then, the flash memory is re-programmed by executing the programming routine. The host sends out both commands and programming data to re-program the flash memory. Communications between the SIO0 and the host must follow the protocol described later. To secure the contents of the flash memory, the validity of the application's password is verified before a programming routine is downloaded into the on-chip RAM. If password matching fails, the transfer of a programming routine itself is aborted. As in the case of User Boot mode, all interrupts including the non-maskable interrupt (NMI) must be disabled in Single Boot mode while the flash memory is being erased or programmed. In Single Boot mode, the boot-ROM programs 33are executed in Normal mode.

Once re-programming is complete, it is recommended to set the write/erase protection to the relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations.

28.2.3.1 (2-A) Using the Program in the On-Chip Boot ROM

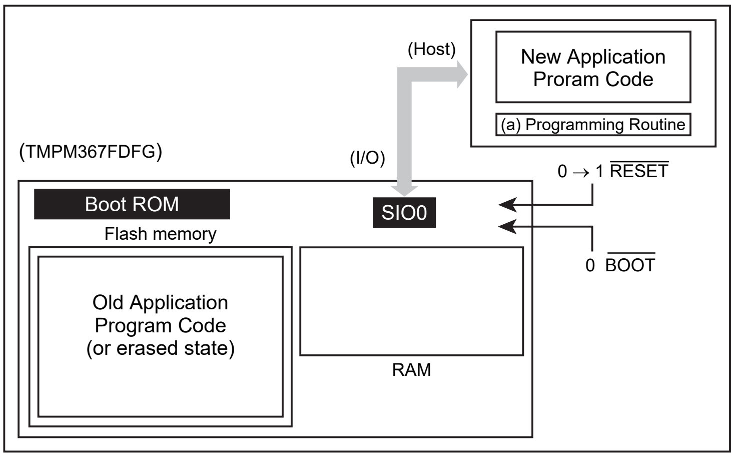
(1) Step-1

The flash block containing the old version of the program code does not need to be erased before executing the programming routine. Since a programming routine and programming data are transferred via the SIO (SIO0), the SIO0 must be connected to a host controller. Prepare a programming routine (a) on the host controller.


(Host)
New Application Program Code
(a)Programming Routine
(TMPM367FDFG)
(I/O)
Boot ROM
SIO0
Flash memory
Old Application Program Code (or erased state)
RAM

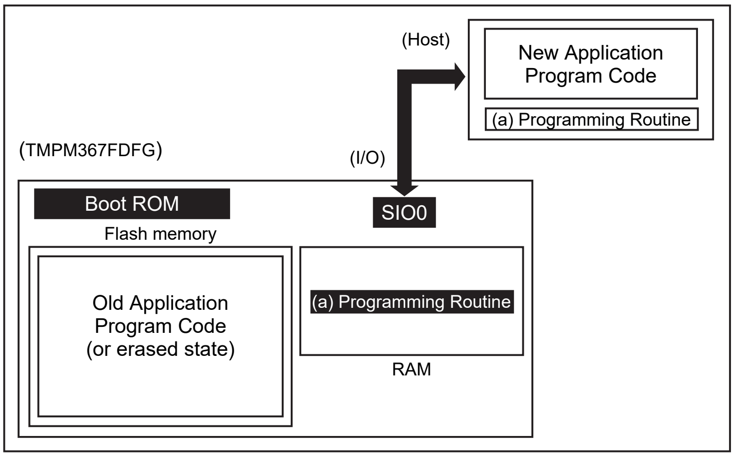
(2) Step-2

Set the $\overline{\text{RESET}}$ pin to "1" to cancel the reset of the TMPM367FDFG when the $\overline{\text{BOOT}}$ pin has already been set to "0". After reset, CPU reboots from the on-chip boot ROM. The 12-byte password transferred from the host controller via SIO0 is firstly compared to the contents of the special flash memory locations. (If the flash block has already been erased, the password is 0xFF).


(Host)
New Application Proram Code
(a) Programming Routine
(TMPM367FDFG)
(I/O)
0 → 1 RESET
Boot ROM
SIO0
Flash memory
0 BOOT
Old Application Program Code (or erased state)
RAM

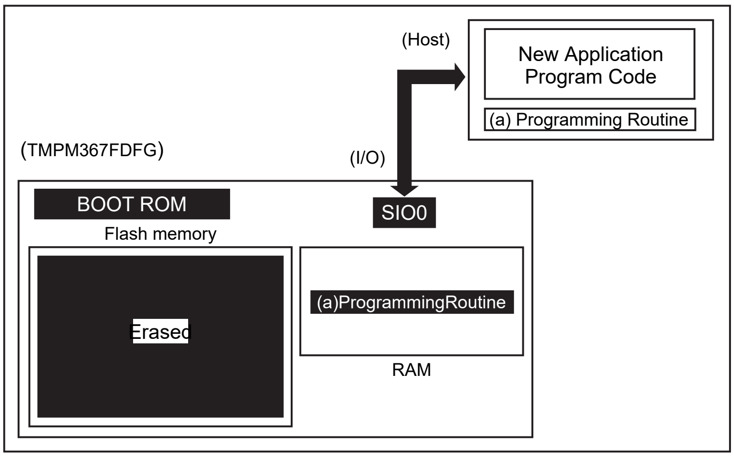
(3) Step-3

If the password is correct, the boot program downloads the programming routine (a) from the host controller into the on-chip RAM of the TMPM367FDFG. Regarding the address stored programming routine, refer to "28.2.5 Memory Map".


(Host)
New Application Program Code
(a) Programming Routine
(TMPM367FDFG)
(I/O)
Boot ROM
SIO0
Flash memory
Old Application Program Code (or erased state)
(a) Programming Routine
RAM

(4) Step-4

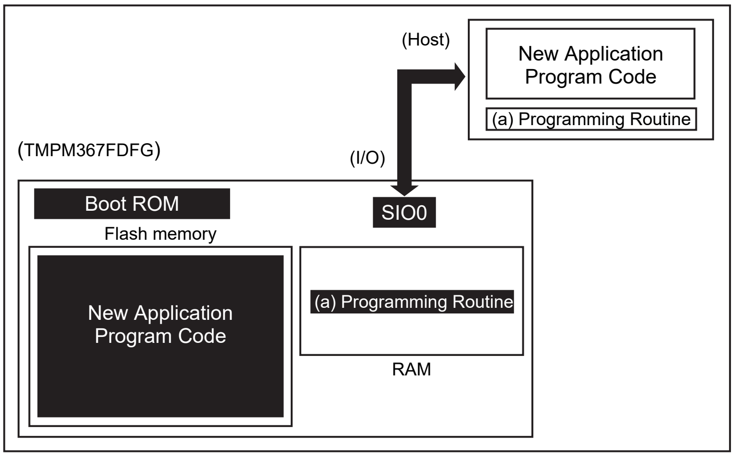
The CPU jumps to the programming routine (a) in the on-chip RAM to erase the flash block containing the old application program code. The Block Erase or Chip Erase command may be used.


(Host)
New Application Program Code
(a) Programming Routine
(TMPM367FDFG)
(I/O)
BOOT ROM
SIO0
Flash memory
Erased
(a)ProgrammingRoutine
RAM

(5) Step-5

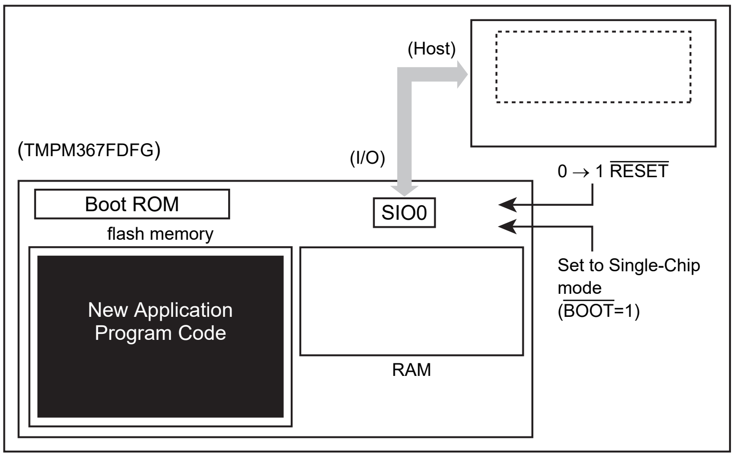
Next, the programming routine (a) downloads new application program code from the host controller and programs it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user's program area must be set.

In the example below, new program code comes from the same host controller via the same SIO0 channel as for the programming routine. However, once the programming routine has begun to execute in the on-chip RAM, it is free to change the transfer path and the source of the transfer. Create board hardware and a programming routine to suit your particular needs.


(Host)
New Application Program Code
(a) Programming Routine
(TMPM367FDFG)
(I/O)
Boot ROM
SIO0
Flash memory
New Application Program Code
(a) Programming Routine
RAM

(6) Step-6

When programming of the flash memory is complete, power off the board and disconnect the cable between the host and the target board. Turn on the power again so that the TMPM367FDFG reboots in Single-Chip (Normal) mode to execute the new program.


(Host)
(TMPM367FDFG)
(I/O)
0 → 1 $\overline{\text{RESET}}$
Boot ROM
SIO0
flash memory
New Application Program Code
RAM
Set to Single-Chip mode ($\overline{\text{BOOT}}$=1)

28.2.4 Configuration for Single Boot Mode

To execute the on-board programming, boot the TMPM367FDFG with Single Boot mode following the configuration shown below.

$\overline{\text{BOOT}}$(PB6) = 0
$\overline{\text{RESET}}$ = 0 → 1

Set the $\overline{\text{RESET}}$ input to "0", and set the each $\overline{\text{BOOT}}$ (PB6) pins to values shown above, and then release $\overline{\text{RESET}}$ pin (high).

28.2.5 Memory Map

Figure 28-3 shows a comparison of the memory maps in Normal and Single Boot modes. In Single Boot mode, the internal flash memory is mapped to 0x3F80_0000 and later addresses, and the Internal boot ROM (Mask ROM) is mapped to 0x0000_0000 through 0x0000_27FF.

The internal flash memory and RAM addresses of each device are shown below.

Product Name	Flash Size	RAM Size	Flash Address (Single Chip / Single Boot Mode)	RAM Address
TMPM367FDFG	512 KB	128 KB	0x0000_0000 to 0x0007_FFFF 0x3F80_0000 to 0x3F87_FFFF	0x2000_0000 to 0x2001_FFFF note)

Note 1: In case of TMPM367FDFG a programming routine is stored from 0x2000_0800 to the end of RAM.

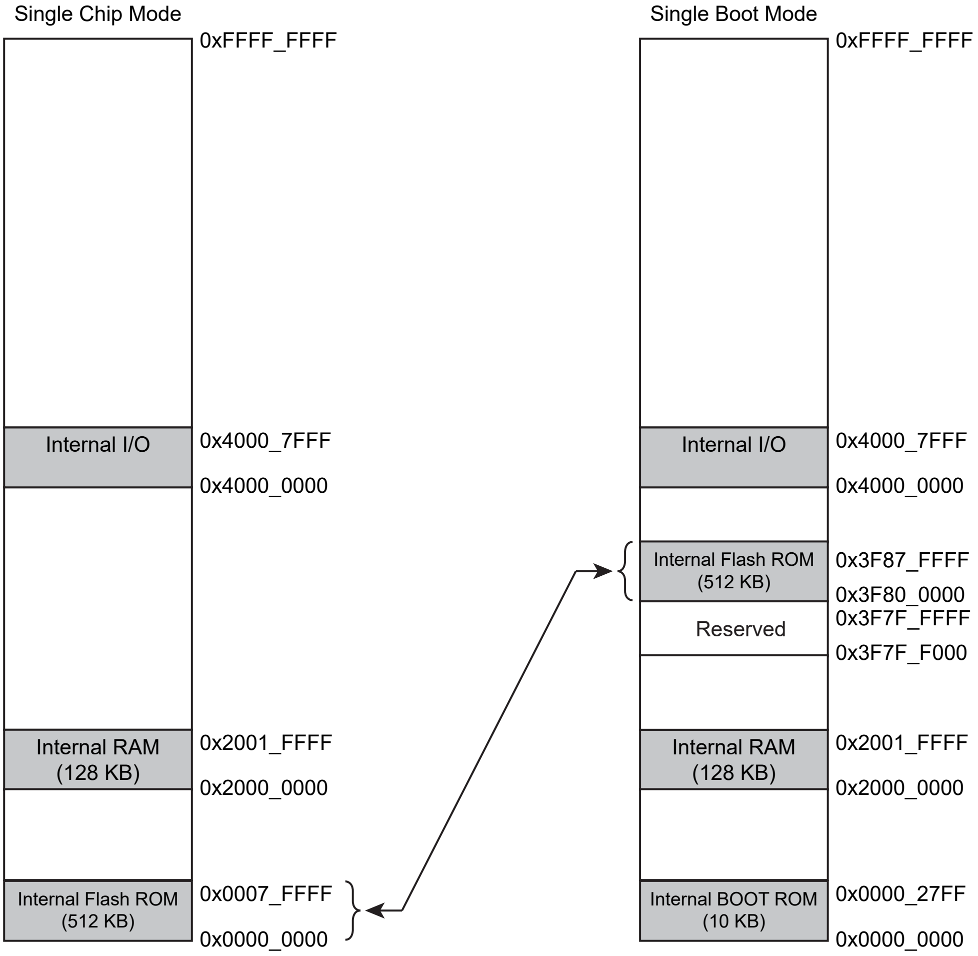

Single Chip Mode
0xFFFF_FFFF
Internal I/O
0x4000_7FFF
0x4000_0000
Internal RAM (128 KB)
0x2001_FFFF
0x2000_0000
Internal Flash ROM (512 KB)
0x0007_FFFF
0x0000_0000
Single Boot Mode
0xFFFF_FFFF
Internal I/O
0x4000_7FFF
0x4000_0000
Internal Flash ROM (512 KB)
0x3F87_FFFF
0x3F80_0000
Reserved
0x3F7F_FFFF
0x3F7F_F000
Internal RAM (128 KB)
0x2001_FFFF
0x2000_0000
Internal BOOT ROM (10 KB)
0x0000_27FF
0x0000_0000

Figure 28-3 Memory Maps for TMPM367FDFG

28.2.6 Interface specification

In Single Boot mode, an SIO channel is used for communications with a programming controller. Both UART (asynchronous) and I/O Interface (synchronous) modes are supported.

In USB Boot mode, USB port is used for communications with programming controller. The same configuration is applied to a communication format on a programming controller to execute the on-board programming in this mode.

The communication formats are shown below.

- UART communication

  Communication channel : SIO channel 0

  Serial transfer mode : UART (asynchronous), half -duplex, LSB first

  Data length : 8 bits

  Parity bit : None

  STOP bit : 1 bit

  Baud rate : Arbitrary baud rate

- I/O Interface mode

  Communication channel : SIO channel 0

  Serial transfer mode : I/O interface mode, full -duplex, LSB first

  Synchronization clock (SCLK0) : Input mode

  Handshaking signal : PE4 configured as an output mode

  Baud rate : Arbitrary baud rate

- USB Boot mode

  Communication port : USB-DDP, USB-DDM

  Only full speed available

  Transfer mode : control / bulk

  USB clock : 48MHz (fosc=8/12/16MHz crystal oscillator with PLL or external clock input)

Table 28-3 Required Pin Connections

Pin		Interface		
		UART	I/O Interface Mode	USB
Voltage supply pin	DVDD3A	o	o	o
	DVSSA	o	o	o
	DVDD3B	o	o	o
	DVSSB	o	o	o
	AVDD3A	o	o	o
	AVSSA	o	o	o
	AVDD3B	o	o	o
	AVSSB	o	o	o
	AVDD3_DA	o	o	o
	AVSS_DA	o	o	o
	RVDD3	o	o	o

Table 28-3 Required Pin Connections

Pin		Interface		
		UART	I/O Interface Mode	USB
Mode setting pin	MODE	Fixed to "L".		
	PE5	o (Fixed to "L")	o (Fixed to "L")	o (Fixed to "H")
	PK0	×	×	o (Detects Vbus connection) (note 1)
	$\overline{\text{BOOT}}$ (PB6)	o	o	o
	X1	-	-	o (8/12/16 MHz)
	PK2	×	×	o (USB_ECLK)
reset pin	$\overline{\text{RESET}}$	o	o	o
Communication pin	PE0	×	o (Output mode) (note 2)	o (Output mode) (note 3)
	PE1	o (RXD0, Input mode)	o (RXD, Input mode)	o (Input mode) (note 4)
	PE2	o (TXD, Output mode)	o (TXD, Output mode)	o (Input mode) (note 4)
	PE3	×	o (SCLK0, Input mode)	o (Input mode) (note 4)
	USB-DDP	×	×	o
	USB-DDM	×	×	o

Note 1: This port is not used in Boot mode.

Note 2: In I/O interface mode, this port is used for hand shake signal.

Note 3: This port is used for control signal of the switch which is make USB-DDP pin to pull-up.

Note 4: In USB Boot mode, specifies the USB clock. Refer to below table.

PE3	PE2	PE1	fosc (MHz)	PLL	USB clock
0	0	0	8	6 times	8MHz x 3 = 48MHz
0	0	1	12	4 times	12MHz x 4 = 48MHz
0	1	0	-	-	USB_ECLK pin
0	1	1	-	-	USB_ECLK pin
1	0	0	16	3 times	16MHz x 3 = 48MHz
1	0	1	-	-	Reserved
1	1	0	-	-	USB_ECLK pin
1	1	1	-	-	USB_ECLK pin

28.2.7 Data Transfer Format

Table 28-4, Table 28-6 to Table 28-7 illustrate the operation commands and data transfer formats at each operation mode. In conjunction with this section, refer to "28.2.10 Operation of Boot Program".

Table 28-4 Single Boot Mode Commands

Code	Command
0x10	RAM transfer
0x40	Chip and protection bit erase

28.2.8 Restrictions on internal memories

Single Boot Mode places restrictions on the internal RAM and ROM as shown in Table 28-5.

Table 28-5 Restrictions in Single Boot Mode

Memory	Details
Internal RAM	A program contained in the BOOT ROM uses the area, through 0x2000_0000 to 0x2000_07FF, as a work area. Regarding the address stored programming routine, refer to "28.2.5 Memory Map"
Internal ROM	The following addresses are assigned for storing software ID information and passwords. Storing program in these addresses is not recommendable. 0x3F87_FFF0 to 0x3F87_FFFF

28.2.9 Transfer Format for Boot Program

The following tables shows the transfer format for each Boot program command. Use this section in conjunction with Chapter "28.2.10 Operation of Boot Program".

28.2.9.1 RAM Transfer

Table 28-6 Transfer Format for the RAM Transfer Command

	Byte	Data Transferred from the Controller to the TMPM367FDFG	Baud rate	Data Transferred from the TMPM367FDFG to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode : 0x86 For I/O Interface mode : 0x30	Desired baud rate (Note 1)	–
	2 byte	–		ACK for the serial operation mode byte ・For UART mode - Normal acknowledge : 0x86 (The boot program aborts if the baud rate can not be set correctly.) ・For I/O Interface mode - Normal acknowledge :0x30
	3 byte	Command code (0x10)		–
	4 byte	–		ACK for the command code byte (Note 2) - Normal acknowledge : 0x10 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	5 byte to 16 byte	Password sequence (12 bytes)) 0x3F87_FFF4 to 0x3F87_FFFF		–
	17 byte	Check SUM value for bytes 5 to 16		–
	18 byte	–		ACK for the checksum byte (Note 2) - Normal acknowledge : 0x10 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	19 byte	RAM storage start address 31 to 24		–
	20 byte	RAM storage start address 23 to 16		–
	21 byte	RAM storage start address 15 to 8		–
	22 byte	RAM storage start address 7 to 0		–
	23 byte	RAM storage start address 15 to 8		–
	24 byte	RAM storage start address 7 to 0		–
	25 byte	Check SUM value for bytes 19 to 24		–
	26 byte	–		ACK for the checksum byte (Note 2) - Normal acknowledge : 0x10 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	27 byte to mbyte	RAM storage data		–
	m+ 1 byte	Checksum value for bytes 27 to m		–
	m+ 2 byte	–		ACK for the checksum byte (Note 2) - Normal acknowledge : 0x10 - Negative acknowledge : 0xX1 - Communication error : 0xX8
RAM	m+ 3 byte	–		Jump to RAM storage start address

Note 1: **In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.**

Note 2: **In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.**

Note 3: **The 19th to 25th bytes must be within the RAM address range which is determined depend on each products. For detail of the RAM address, refer to "28.2.5 Memory Map".**

28.2.9.2 Chip Erase and Protect Bit Erase

Table 28-7 Transfer Format for the Chip and Protection Bit Erase Command

	Byte	Data Transferred from the Controller to the TMPM367FDFG	Baud rate	Data Transferred from the TMPM367FDFG to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode : 0x86 For I/O Interface mode : 0x30	Desired baud rate (Note 1)	–
	2 byte	–		ACK for the serial operation mode byte ・For UART mode - Normal acknowledge : 0x86 (The boot program aborts if the baud rate can not be set correctly.) ・For I/O Interface mode - Normal acknowledge :0x30
	3 byte	Command code (0x40)		–
	4 byte	–		ACK for the command code byte (Note 2) - Normal acknowledge : 0x10 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	5 to 16 byte	Password data (12 bytes) when the data of 0x3F87_FFF0 is not 0xFF 0x3F87_FFF4 to 0x3F87_FFFF Dummy data (12 bytes) when the data of 0x3F87_FFF0 is 0xFF		–
	17 byte	Check Sum value for byte 5 to 16		–
	18 byte	–		ACK for the checksum byte (Note 2) - Normal acknowledge : 0x40 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	19 byte	Chip erase command code (0x54)		–
	20 byte	–		ACK for the command code byte (Note 2) - Normal acknowledge : 0x54 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	21 byte	–		ACK for the chip erase command code byte - Normal acknowledge : 0x4F - Negative acknowledge : 0x4C
	22 byte	(Wait for the next command code.)		–

Note 1: **In I/O Interface mode, the baud rate for the transfers of the first and second byte must be 1/16 of the desired baud rate.**

Note 2: **In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.**

28.2.10 Operation of Boot Program

When Single Boot mode is selected, the boot program is automatically executed on startup. The boot program offers these four commands, of which the details are provided on the following subsections.

1. RAM Transfer command

   The RAM Transfer command stores program code transferred from the host controller to the on-chip RAM and executes the program once the transfer is successfully completed. The user program RAM space can be assigned to the RAM space except boot program area (0x2000_0000 to 0x2000_07FF). For the detail of the RAM space, refer to "28.2.5 Memory Map".

   The user program starts at the assigned RAM address.

   The RAM Transfer command can be used to download a flash programming routine of your own; this provides the ability to control on-board programming of the flash memory in a unique manner. The programming routine must utilize the flash memory command sequences described in Section 28.3. Before initiating a transfer, the RAM Transfer command verifies a password sequence coming from the controller against that stored in the flash memory.

Note: **If a password is set to 0xFF (erased data), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.**

2. Flash Memory Chip Erase and Protection Bit Erase command

   This command erases the entire area of the flash memory automatically. All the blocks in the memory cell and their protection conditions are erased even when any of the blocks are prohibited from writing and erasing. This command can select whether a password is verified.

28.2.10.1 RAM Transfer Command

See Table 28-6 for the transfer format of this command.

1. The 1st byte specifies which one of the two serial operation modes is used. For a detailed description of how the serial operation mode is determined, see "28.2.10.4 Determination of a Serial Operation Mode" described later. If the mode is determined as UART mode, the boot program checks if the baud rate setting can be performed. During the first-byte processing, receiving operation is prohibited. (SC0MOD0<RXE>=0)
   - To communicate in UART mode

     The 1st byte is set to "0x86" and is transmitted from the controller to the target board at the specified baud rate by setting UART. If the serial operation mode is determined as UART, then the boot program checks if the baud rate setting can be performed. If that baud rate cannot be set, the boot program aborts and any subsequent communications cannot be done. Please refer to "Baud rate setting" for the method of judging whether the setting of the baud rate is possible.

   - To communicate in I/O Interface mode

     The 1st byte is set to "0x30" and is transmitted from the controller to the target board at 1/16 of the desired baud rate by the synchronous setting. Same as the 1st byte, a 1/16 of the specified baud rate is used in the 2nd transmission. From the 3rd byte (operation command data), users can transmit data at specified baud rate.

     In I/O interface mode, CPU considers the reception terminal to be an input port and monitors the level of I/O port. If the baud rate is high or operation frequency is high, CPU may not distinguish the level of I/O port. To avoid this situation, the baud rate is set at the 1/16 of desired baud rate in the I/O interface. When the serial operation mode is determined as I/O Interface mode, SCLK Input mode is set. The controller must ensure that its AC timing restrictions are satisfied at the selected baud rate. In the case of I/O Interface mode, the boot program does not check the receive error flag; thus there is no error acknowledge response (bit 3, 0xX8).

2. The 2nd byte, transmitted from the target board to the controller, is an acknowledge response to the 1st byte where the serial operation mode is set. When 1st byte is determined as UART and can be set at the specified baud rate, data "0x86" is transmitted. When 1st byte is determined as I/O interface, data "0x30" is transmitted.
   - UART mode

     The 2nd byte is used for distinguishing whether the baud rate can be set. If the baud rate can be set, a value of SC0BRCE is renewed and data "0x86" is sent to the controller. If the baud rate cannot be set, transmit operation is stopped and no data is transmitted. After transmission of 1st byte completed, the controller allows for five seconds of time-out. If it does not receive 0x86 within the allowed time-out period, the controller should give up the communication. Receiving operation is permitted by setting SC0MOD0<RXE>=1, before loading 0x86 to the SIO transmit buffer.

   - I/O Interface mode

     The boot program sets a value of the SC0MOD0 and SC0CR registers to configure the the I/O Interface mode and writes 0x30 to the SC0BUF. Then, the SIO0 waits for the SCLK4 signal to come from the controller. After the transmission of the 1st byte completed, the controller should send the SCLK clock to the target board after a certain idle time (several microseconds). This must be done at 1/16 of the desired baud rate. If the 2nd byte, which is from the target board to the controller, is 0x30, then the controller regards it as communication possible. From the 3rd byte, users can transmit data at specified baud rate. Receiving operation is permitted by setting SC0MOD0<RXE>=1, before loading 0x30 to the SIO.

3. The 3rd byte transmitted from the controller to the target board is a command. The code for the RAM Transfer command is 0x10.

4. The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there is a receive error, the boot program transmits 0xX8 (bit 3) and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

   If the 3rd byte is equal to any of the command codes listed in Table 28-4, the boot program echoes it back to the controller. When the RAM Transfer command is received, the boot program echoes back a value of 0x10 and then branches to the RAM Transfer routine. Once this branch is taken, password verification is done. Password verification is detailed in the later Section "Password". If the 3rd byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

5. The 5th to 16th bytes transmitted from the controller to the target board, are a 12-byte password. Each byte is compared to the contents of following addresses in the flash memory. The verification is started with the 5th byte. If the password verification fails, the RAM Transfer routine sets the password error flag.

Product name	Area
TMPM367FDFG	0x3F87_FFF4 to 0x3F87_FFFF

6. The 17th byte is a checksum value for the password sequence (5th to 16th bytes). To calculate the checksum value for the 12-byte password, add the 12 bytes together, ignore the carries and calculate the 8-bit two's complement by using lower 8 bits then transmit this checksum value from the controller. The checksum calculation is described in details in the later Section "Checksum Calculation".

7. The 18th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th to 17th bytes. First, the RAM Transfer routine checks for a receive error in the 5th to 17th byte. If there is a receive error, the boot program sends back 0x18 (bit 3) and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

   Next, the RAM Transfer routine performs the checksum operation to ensure 17th byte data integrity. Adding the series of the 5th to 16th bytes must result in 0x00 (with the carry dropped). In case of a checksum error, the RAM Transfer routine sends back 0x11 to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

   Finally, the password verification result is checked. If the following case is generated, the boot program transmits an acknowledge response (bit 0, 0x11) as a password error and waits for next operation command (3rd byte).

   - Irrespective of the result of the password comparison, all the 12 bytes of a password in the flash memory are the same value other than 0xFF.
   - Not the entire password bytes transmitted from the controller matched those contained in the flash memory.

   When all the above verification has been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

8. The 19th to 22nd bytes, transmitted from the controller the target board, indicate the start address of the RAM region where subsequent data (e.g., a flash programming routine) should be stored. The 19th byte corresponds to bits 31 to 24 of the address and the 22nd byte corresponds to bits 7 to 0 of the address. The start address of the stored RAM must be even address.

9. The 23rd and 24th bytes, transmitted from the controller to the target board, indicate the number of bytes that will be transferred from the controller to be stored in the RAM. The 23rd byte corresponds to bits 15 to 8 of the number of bytes to be transferred, and the 24th byte corresponds to bits 7 to 0 of the number of bytes.

10. The 25th byte is a checksum value for the 19th to 24th bytes. To calculate the checksum value, add all these bytes together, ignore the carries and calculate the 8-bit two's complement by using lower 8 bits then transmit this checksum value from the controller. The checksum calculation is described in detail in the later Section "28.2.10.6 Checksum Calculation".

11. The 26th byte, transmitted from the target board to the controller, is an acknowledge response to the 19th to 25th bytes of data. First, the RAM Transfer routine checks for a receive error in the 19th to 25th bytes. If there is a receive error, the RAM Transfer routine sends back 0x18 and returns to the command wait state (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

    Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 19th to 24th bytes must result in 0x00 (with the carry dropped). In case of a checksum error, the RAM Transfer routine sends back 0x11 to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

    - The 19th to 25th bytes data must be determined by referring "28.2.5 Memory Map". They are set within the suitable range for each product to the end address of RAM.

    When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

12. The 27th to mth bytes from the controller are stored in the on-chip RAM of the TMPM367FDFG. Storage begins at the address specified by the 19th to 22nd bytes and continues for the number of bytes specified by the 23rd to 24th bytes.

13. The (m+1) th byte is a checksum value. To calculate the checksum value, add the 27th to mth bytes together, ignore the carries and calculate the 8-bit two’s complement by using lower 8 bits then transmit this checksum value from the controller. The checksum calculation is described in detail in later Section "28.2.10.6 Checksum Calculation".

14. The (m+2) th byte is a acknowledge response to the 27th to (m+1) th bytes. First, the RAM Transfer routine checks for a receive error in the 27th to (m+1) th bytes. If there is a receive error, the RAM Transfer routine sends back 0x18 (bit 3) and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

    Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 27th to (m+1) th bytes must result in 0x00 (with the carry dropped). In case of a checksum error, the RAM Transfer routine sends back 0x11 (bit 0) to the controller and returns to the command wait state (i.e., the 3rd byte) again. When the above checks have been completed successfully, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

15. If the (m+2) th byte was a normal acknowledge response, a branch is made to the address specified by the 19th to 22nd bytes.

28.2.10.2 Chip and Protection Bit Erase Command

See Table 28-7 for the transfer format of this command.

1. The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

2. From the Controller to the TMPM367FDFG

   The 3rd byte, which the target board receives from the controller, is a command. The code for the Show Product Information command is 0x40.

3. From TMPM367FDFG to the Controller

   The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte.

   Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

   If the 3rd byte is equal to any of the command codes listed in Table 28-4, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was received, the boot program echoes back a value of 0x40. If the 3rd byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

4. The 5th byte to 16th byte, transmitted from the target board to the controller, differ according to the Chip Erase Enable / Disable area (0x3F87_FFF0).

   If 0x3F87_FFF0 is not the same value as 0xFF, a erase password is required. The 5th byte from the 16th byte becomes password data (12bytes). The verification is started with the 5th byte to compare with the addresses in the flash memory shown in the table below. If the password verification fails, the password error flag is set.

Product name	Password area
TMPM367FDFG	0x3F87_FFF4 to 0x3F87_FFFF

   If 0x3F87_FFF0 is 0xFF, a password is not required. The 5th byte from 16th byte becomes dummy data.

5. The 17th byte is a checksum value. To calculate the checksum value, perform 8-bit unsigned addition of transmits data (5th byte to 16th byte), drop the carries and take the two’s complement of the total sum. Transmit this checksum value from controller to the target board. The checksum calculation is described in details in a latter section "Checksum Calculation".

6. The 18th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th to 17th bytes. First, the Chip and Protection bit Erase routine checks for a receive error in the 5th to 17th bytes. If there was a receive error, the boot program sends back 0x48 (bit3) and returns to the state in which it for a ccommand(i.e., the 3rd byte) again. Since the up-

per four bits of the transmitted data are the same as those of the previously issued command (i.e., 4). When the SIO0 is configured for I/O Interface mode, the Chip and Protection bit Erase routine does not check for a receive error.

Next, the Chip and Protection bit Erase routine performs the checksum operation to ensure data integrity. Adding the series of the 5th to 16th bytes must result in 0x00 (with the carry dropped). If it is not 0x00, one or more bytes of data have been corrupted. In case of a checksum error, the Chip and Protection bit Erase routine sends back the ACK response data (0x41) to the control and returns to the state in which it for a command (i.e., the 3rd byte) again.

Finally, the Chip and Protection bit Erase routine examines he result of the password verification. The following two cases are treated as a password error. In these cases, the Chip and Protection bit Erase routine sends back the ACK response error 0x41 (bit 0) to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

- Irrespective of the result of the password comparison (from 5th byte to 16th byte), all the 12 bytes of a password in the flash memory are the same value other than 0xFF.
- Not the entire password bytes transmitted from the controller matched those contained in the flash memory.

When all the above verification has been successful, the Chip and Protection bit Erase routine returns a normal acknowledge response (0x40) to the controller.

7. From the controller to the TMPM367FDFG

   The 19th byte, transmitted from the target board to the controller, is the Chip Erase Enable command code (0x54).

8. From TMPM367FDFG to the Controller

   The 20th byte, transmitted from the target board to the controller, is an acknowledge response to the 19th byte.

   Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

   If the 19th byte is equal to any of the command codes to enable erasing, the boot program echoes it back to the controller. When the Chip and Protection Erase command was received, the boot program echoes back a value of 0x54 and then branches to the Chip and Protection bit Erase routine. If the 5th byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

9. From TMPM367FDFG to the Controller

   The 21st byte indicates whether the Chip and Protection bit Erase command is normally completed or not.

   At normal completion, completion code (0x4F) is sent.

   When an error was detected, error code (0x4C) is sent.

10. The 22nd byte is the next command code.

28.2.10.3 Acknowledge Responses

The boot program represents processing states with specific codes. Table 28-8 to show the values of possible acknowledge responses to the received data. The upper four bits of the acknowledge response are equal to those of the command being executed. The 3rd bit indicates a receive error. The 0th bit indicates an invalid command error, a checksum error or a password error. The 1st bit and 2nd bit are always "0". Receive error checking is not done in I/O Interface mode.

Table 28-8 ACK Response to the Serial Operation Mode Byte

Return Value	Meaning
0x86	The SIO can be configured to operate in UART mode. (See Note)
0x30	The SIO can be configured to operate in I/O Interface mode.

Note: **In the UART mode, if the baud rate setting cannot be set, the communication is stopped without any response.**

Table 28-9 ACK Response to the Command Byte

Return Value	Meaning
0xX8 (See note)	A receive error occurred while receiving a command code.
0xX1 (See note)	An undefined command code was received. (Reception was completed normally.)
0x10	The RAM Transfer command was received.
0x40	The Chip Erase command was received.

Note: **The upper four bits of the ACK response are the same as those of the previous command code.**

Table 28-10 ACK Response to the Checksum Byte

Return Value	Meaning
0xN8 (See note)	A receive error occurred.
0xN1 (See note)	A checksum or password error occurred.
0xN0 (See note)	The checksum was correct.

Note: **The upper four bits of the ACK response are the same as those of the operation command code. For example, it is 1 (N ; RAM transfer command data [7:4]) when password error occurs.**

Table 28-11 ACK Response to Chip and Protection Bit Erase Byte

Return Value	Meaning
0x54	The Chip Erase enabling command was received.
0x4F	The Chip Erase command was completed.
0x4C	The Chip Erase command was abnormally completed.

28.2.10.4 Determination of a Serial Operation Mode

The first byte from the controller determines the serial operation mode. To use UART mode for communications between the controller and the target board, the controller must firstly send a value of 0x86 at a desired baud rate to the target board. To use I/O Interface mode, the controller must send a value of 0x30 at 1/16 of the desired baud rate. Figure 28-4 shows the waveforms for the first byte in each mode.

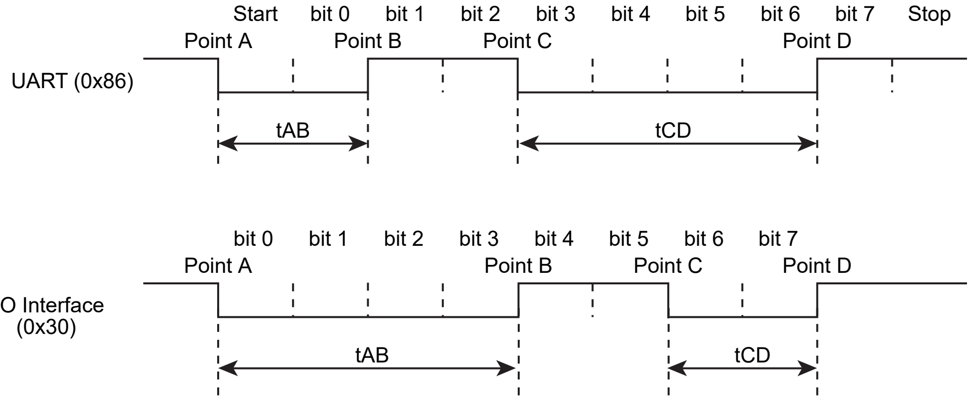


Figure 28-4 Serial Operation Mode Byte

After $\overline{\text{RESET}}$ is released, the boot program monitors the first serial byte from the controller, with the SIO reception disabled, and calculates the intervals of tAB, tAC and tAD. Figure 28-5 shows a flowchart describing the steps to determine the intervals of tAB, tAC and tAD. As shown in the flowchart, the boot program captures timer counts when each time the logic transition occurs in the first serial byte. Consequently, the calculated tAB, tAC and tAD intervals tend to have slight errors. If the transfer goes at a high baud rate, the CPU might not be able to keep up with the speed of logic transitions at the serial receive pin. In particular, I/O Interface mode may have this problem since its baud rate is generally much higher than that for UART mode. To avoid such a situation, the controller should send the first serial byte at 1/16 of the desired baud rate.

The flowchart in Figure 28-5 shows how the boot program distinguishes between UART and I/O Interface modes. If the length of tAB is equal to or less than the length of tCD, the serial operation mode is determined as UART mode. If the length of tAB is greater than the length of tCD, the serial operation mode is determined as I/O Interface mode. Note that if the baud rate is too high or the timer operating frequency is too low, each timer value becomes small. It causes an unintentional behavior of the controller. To prevent this problem, reset UART mode within the programming routine.

For example, the serial operation mode may be determined to be I/O Interface mode when the intended mode is UART mode. To avoid such a situation, when UART mode is utilized, the controller should allow for a time-out period within which it expects to receive an echo-back (0x86) from the target board. The controller should give up the communication if it fails to get that echo-back within the allowed time. When I/O Interface mode is utilized, once the first serial byte has been transmitted, the controller should send the SCLK clock after a certain idle time to get an acknowledge response. If the received acknowledge response is not 0x30, the controller should give up further communications.

When the intended mode is I/O interface mode, the first byte does not have to be 0x30 as long as tAB is greater than tCD as shown above. 0x91, 0xA1 or 0xB1 can be sent as the first byte code to determine the falling edges of Point A and Point C and the rising edges of Point B and Point D. If tAB is greater than tCD and SIO is selected by the resolution of the operation mode determination, the second byte code is 0x30 even though the transmitted code on the first byte is not 0x30 (The first byte code to determine I/O interface mode is described as 0x30).

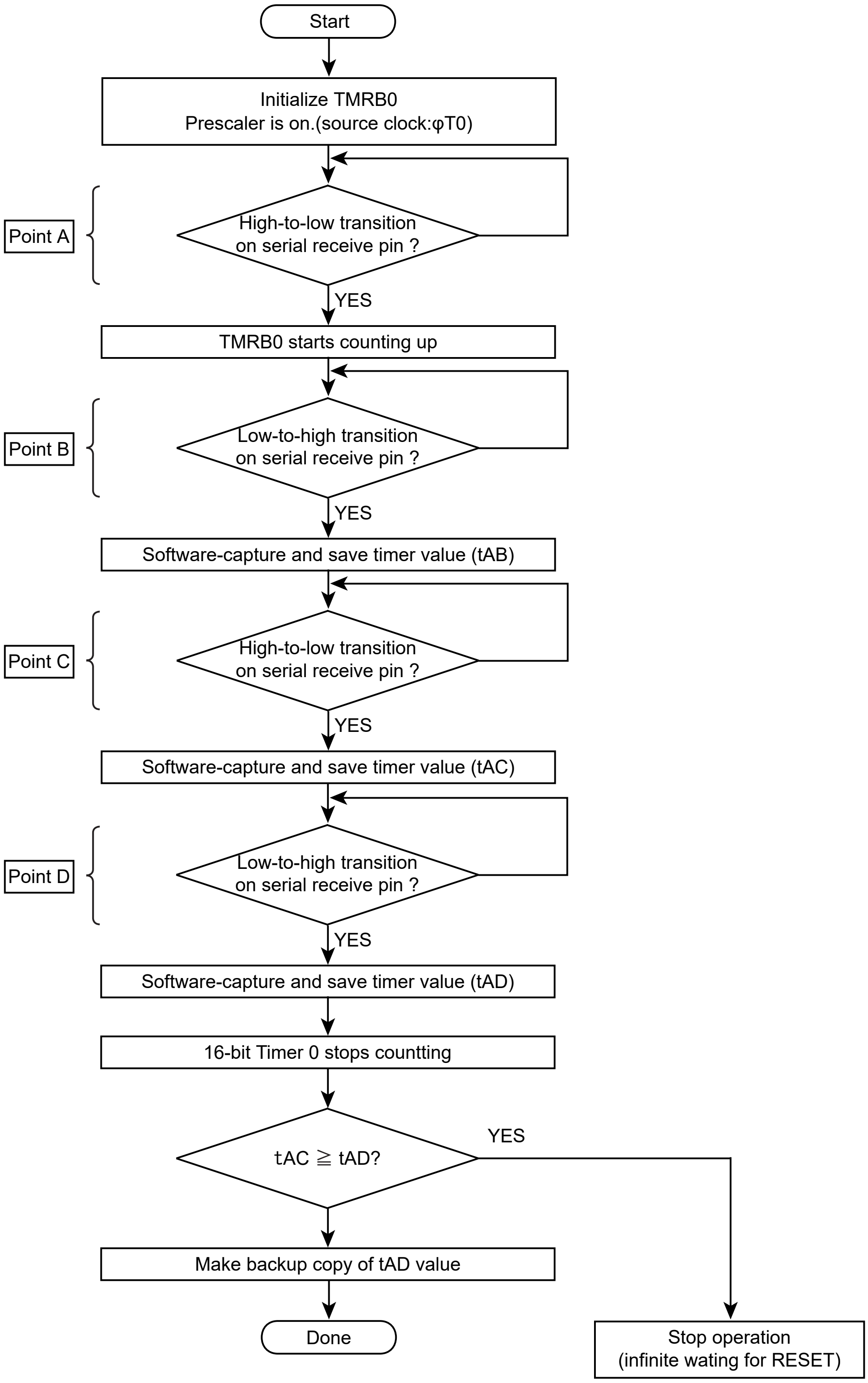

Start
Initialize TMRB0
Prescaler is on.(source clock:φT0)
Point A
High-to-low transition
on serial receive pin ?
YES
TMRB0 starts counting up
Point B
Low-to-high transition
on serial receive pin ?
YES
Software-capture and save timer value (tAB)
Point C
High-to-low transition
on serial receive pin ?
YES
Software-capture and save timer value (tAC)
Point D
Low-to-high transition
on serial receive pin ?
YES
Software-capture and save timer value (tAD)
16-bit Timer 0 stops countting
tAC ≧ tAD?
YES
Make backup copy of tAD value
Done
Stop operation
(infinite wating for RESET)

Figure 28-5 Serial Operation Mode Byte Reception Flowchart

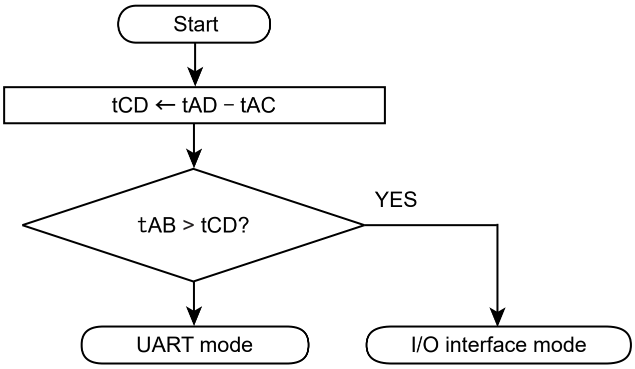


Figure 28-6 Serial Operation Mode Determination Flowchart

28.2.10.5 Password

Verification methods differ according to operation commands. The password area is common to all commands, as shown below. Password verification is performed even if the security is enabled.

Product name	Area
TMPM367FDFG	0x3F87_FFF4 to 0x3F87_FFFF

Note: **If a password is set to 0xFF (erased data area), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.**

(1) RAM Transfer command

If the password is set erased data (0xFF), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.

If all these address locations contain the same bytes of data other than 0xFF, a password area error occurs as shown in Figure 28-7. In this case, the boot program returns an error acknowledge (0x11) in response to the checksum byte (the 17th byte).

The password sequence received from the controller (5th to 16th bytes) is compared to the password stored in the flash memory. All of the 12 bytes must match to pass the password verification. Otherwise, a password error occurs, which causes the boot program to reply an error acknowledge in response to the checksum byte (the 17th byte).

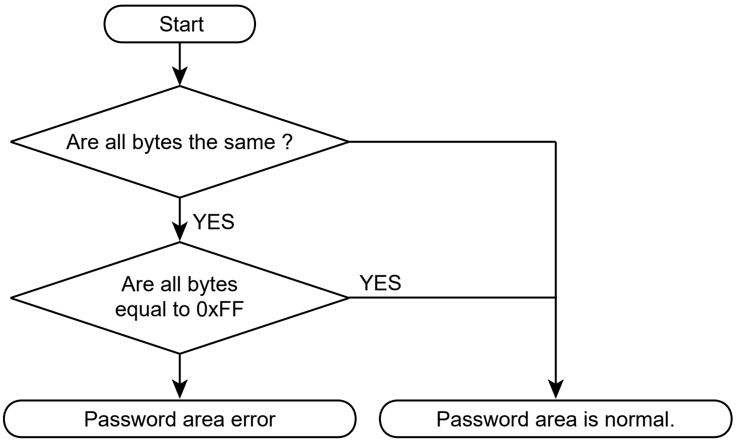

Start
Are all bytes the same ?
YES
Are all bytes equal to 0xFF
YES
Password area error
Password area is normal.

Figure 28-7 Password Area Verification Flowchart (1)

(2) Flash Memory Chip Erase and Protection Bit Erase

The Chip Erase Enable / Disable area specifies whether password verification is performed or not. The Chip Erase Enable / Disable area is as shown below.

Product name	Chip Erase Enable / Disable area
TMPM367FDFG	0x3F87_FFF4 to 0x3F87_FFFF

As shown in the Figure 28-8, if the data contained in the Chip Erase Enable / Disable area is not 0xFF, password verification is executed. If all data in the password area are the same, it is determined as an error. The boot program returns an error acknowledge (0x41) in response to the checksum byte (the 17th byte).

The password sequence received from the controller (5th to 16th bytes) is compared to the password stored in the flash memory. All of the 12 bytes must match to pass the password verification. Otherwise, a password error occurs, which causes the boot program to reply error acknowledges in response to the checksum byte (the 17th byte).

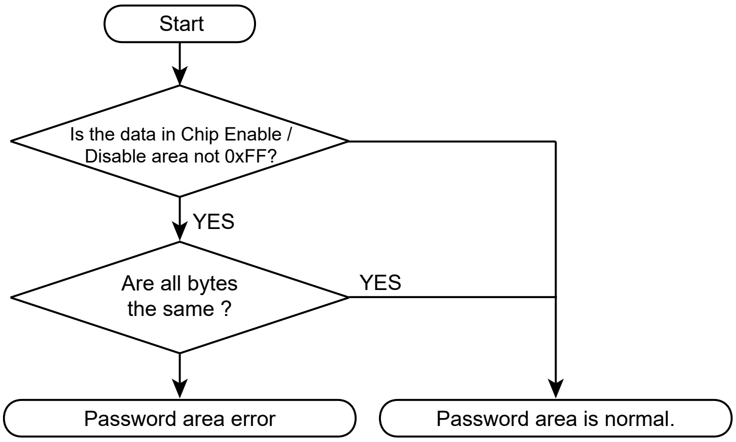

Start
Is the data in Chip Enable / Disable area not 0xFF?
YES
Are all bytes the same ?
YES
Password area error
Password area is normal.

Figure 28-8 Password Area Verification Flowchart (2)

28.2.10.6 Checksum Calculation

The checksum byte for a series of bytes of data is calculated by adding the bytes together with 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) To calculate the checksum for a series of 0xE5 and 0xF6:

Add the bytes together

0xE5 + 0xF6 = 0x1DB

Calculate the two's complement by using lower 8 bits, and that is the checksum byte. Then send 0x25 to the controller.

0 - 0xDB = 0x25

28.2.11 General Boot Program Flowchart

Figure 28-9 shows an overall flowchart of the boot program.

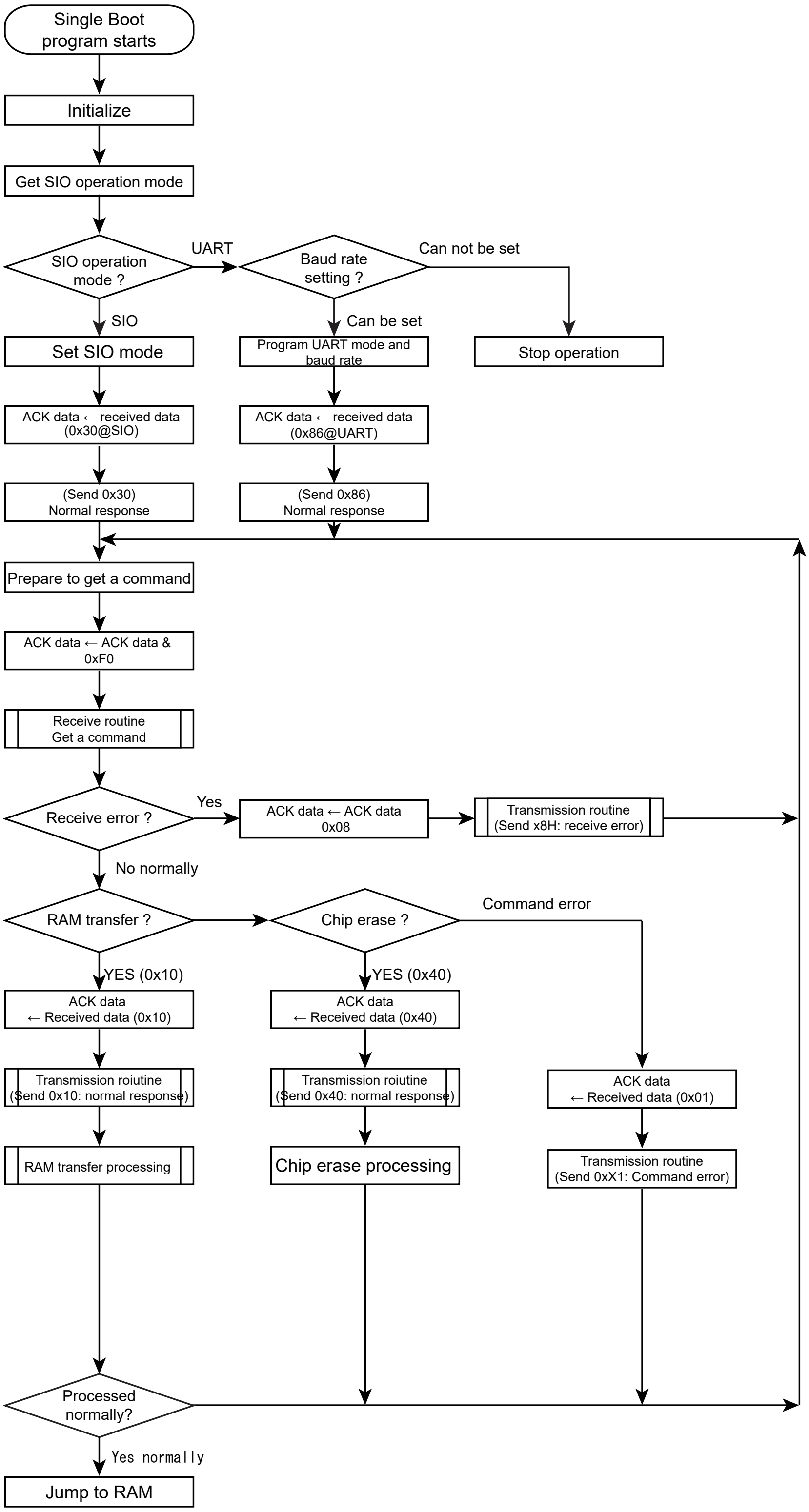

Single Boot program starts
Initialize
Get SIO operation mode
SIO operation mode ?
UART
Baud rate setting ?
Can not be set
SIO
Can be set
Set SIO mode
Program UART mode and baud rate
Stop operation
ACK data ← received data (0x30@SIO)
ACK data ← received data (0x86@UART)
(Send 0x30) Normal response
(Send 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 x8H: 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 roiutine (Send 0x10: normal response)
Transmission roiutine (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 28-9 Overall Boot Program Flowchart

28.2.12 USB Boot

28.2.12.1 Boot Sequence

USB BOOT Sequence is shown as following table.

Table 28-12 USB Boot Sequence

USB BOOT Protocol		
PC		TMPM367FDFG
Start USB BOOT Protocol	→	
	←	Response
Send Password	→	
Confirm Password	→	
	←	Response
Send Boot Information	→	
Confirm Boot Information	→	
	←	Response
Plane Data	→	
Send Sum Data	→	
Confirm Sum Data	→	
	←	Response
Flash Erase Protocol		
Start Flash Erase Protocol	→	
	←	Response
Send Password	→	
Confirm Password	→	
	←	Response
Run Flash Erase	→	
	←	Response
Confirm Flash Erase	→	
	←	Response

28.2.12.2 USB Boot Command

USB Boot Command is shown as following table.

Table 28-13 Boot Command List

			Start USB Boot Protocol	Send Password	Confirm Password	Send Boot Information	Confirm Boot Information
bmRequestType	1byte	Vendor Class	in	out	in	out	in
bRequest	1byte	Command	0x18	0x20	0x28	0x30	0x38
wValue	2byte	0x0000	-	-	-	-	-
wIndex	2byte	Sequence ID	any data	Same as Start Protocol	Same as Start Protocol	Same as Start Protocol	Same as Start Protocol
wLength	2byte	Data Length	1	12	1	6	1
Data Stage	0 to 64 byte		0x18: OK 0x19: NG	Password[0]	0x28:OK 0x29:NG	RAM Address<31:24>	0x38:OK 0x39:NG
				Passwoed[1]		RAM Address<23:16>	
				Password[2]		RAM Address<15:8>	
				Password[3]		RAM Address<7:0>	
				Password[4]		Transfer Size<15:8>	
				Password[5]		Transfer Size<7:0>	
				Password[6]			
				Password[7]			
				Password[8]			
				Password[9]			
				Password[10]			
				Password[11]			

			Send Sum Data	Confirm Sum Data	Start Flash Erase Protocol	Run Flash Erase	Confirm Flash Erase
bmRequestType	1byte	Vendor Class	out	in	in	in	in
bRequest	1byte	Command	0x40	0x48	0x58	0x68	0x78
wValue	2byte	0x0000	-	-	-	-	-
wIndex	2byte	Sequence ID	Same as Start Protocol	Same as Start Protocol	any data	Same as Start Protocol	Same as Start Protocol
wLength	2byte	Data Length	1	1	1	1	1
Data Stage	0 - 64 byte		Sum Data	0x48: OK 0x49: NG	0x58:OK 0x59:NG	0x68: OK 0x69: NG	0x78:OK 0x79:NG

28.2.13 Descriptor

Descriptor in USB Boot Mode is shown as following table.

Table 28-14 Device Descriptor

Offset	Filed	value	Description
0	bLentgth	0x12	18 bytes
1	bDescripotType	0x01	Device descriptor
2	bcdUSB	0x00	USB version 2.0
3		0x02	
4	bDeviceClass	0x00	Device class (Not used)
5	bDeviceSubClass	0x00	Sub command (Not used)
6	bDeviceProtocol	0x00	Protocol (Not used)
7	bMaxPacketSize0	0x40	EP0 Maximum packet size 64 bytes
8	idVendor	0x30	Vender ID
9		0x09	
10	idProduct	0x69	Product ID
11		0x65	
12	bcdDevice	0x00	Device version
13		0x01	
14	iManufacture	0x00	String descriptor index shown as manufacturer
15	iProduct	0x00	String descriptor index shown as Product
16	iSerialNumber	0x00	String descriptor index shown as product's serial number
17	bNumConfigurations	0x01	The number of configuration 1

Table 28-15 Configuration Descriptor

Offset	Filed	value	Description
0	bLentgth	0x09	9 bytes
1	bDescripotType	0x02	Configuration descripor
2	bTotal Lenght	0x20	The length which is added with configuration and end point (32 bytes)
3		0x00	
4	bNumInterfaces	0x01	The number of interface 1
5	bConfigurationValue	0x01	The number of configuration 1
6	iConfiguration	0x00	String descriptor index shown this configuration name (Not used)
7	bmAttributes	0x80	Bus power
8	MaxPower	0x31	Maximum power consumption (49mA)

Table 28-16 Interface Descriptor

Offset	Filed	value	Description
0	bLentgth	0x09	9 bytes
1	bDescripotType	0x04	Interface descriptor
2	bInterdfaceNumber	0x00	The number of interface 1
3	bAlternateSetting	0x00	The number of alternate setting 0
4	bNumEndpoints	0x02	Two end point
5	bInterfaceClass	0xFF	
6	bInterfaceSubClass	0x00	
7	bInterfaceProtocol	0x50	Bulk only protocol
8	iInterface	0x00	String descriptor index shown this interface name (Not used)

Table 28-17 Bulk-In Endpoint Descriptor

Offset	Filed	value	Description
0	bLentgth	0x07	7 bytes
1	bDescripotType	0x05	End point descriptor
2	bEndpointAddress	0x81	End point 1 is used as input
3	bmAttributes	0x02	Bulk transfer
4	wMaxPacketSize	0x40	
5		0x00	
6	bInterval	0x00	(Ignore because of bulk transfer)

Table 28-18 Bulk-Out Endpoint Descriptor

Offset	Filed	value	Description
0	bLentgth	0x07	7 bytes
1	bDescripotType	0x05	End point descriptor
2	bEndpointAddress	0x02	End point 2 is used as output
3	bmAttributes	0x02	Bulk transfer
4	wMaxPacketSize	0x40	Payload 64 bytes
5		0x00	
6	bInterval	0x00	(Ignore because of bulk transfer)

28.3 On-board Programming of Flash Memory (Rewrite/Erase)

In on-board programming, the CPU is to execute software commands for rewriting or erasing the flash memory. The rewrite/erase control program should be prepared by the user beforehand. Because the flash memory content cannot be read while it is being written or erased, it is necessary to run the rewrite/erase program from the internal RAM after shifting to the user boot mode.

28.3.1 Flash Memory

Except for some functions, writing and erasing flash memory data are in accordance with the standard JEDEC commands.

In writing or erasing, use 32-bit data transfer command of the CPU to enter commands to the flash memory. Once the command is entered, the actual write or erase operation is automatically performed internally.

Table 28-19 Flash Memory Functions

Major functions	Description
Automatic page program	Writes data automatically per page.
Automatic chip erase	Erase the entire area of the flash memory automatically.
Automatic block erase	Erases a selected block automatically.
Protect function	The write or erase operation can be individually inhibited for each block.

28.3.1.1 Block Configuration

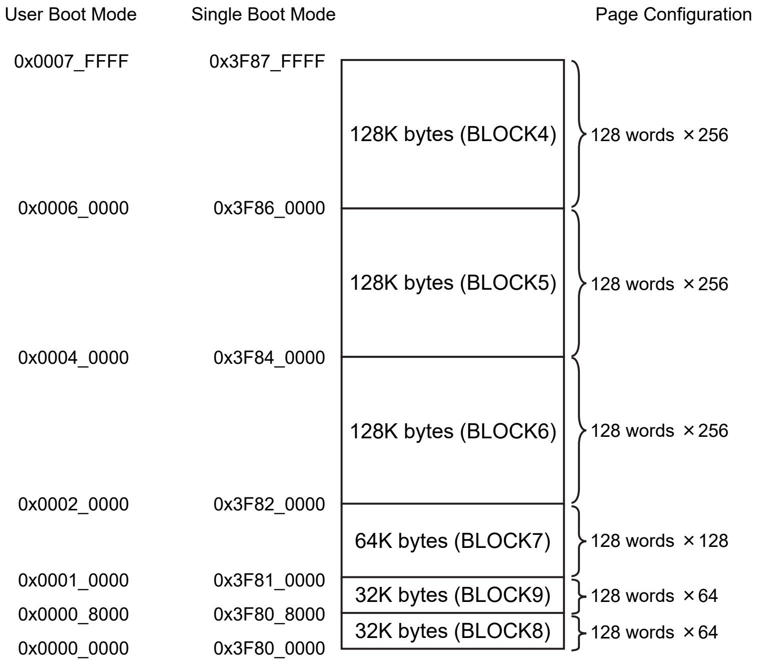

User Boot Mode
Single Boot Mode
Page Configuration
0x0007_FFFF
0x3F87_FFFF
128K bytes (BLOCK4)
128 words ×256
0x0006_0000
0x3F86_0000
128K bytes (BLOCK5)
128 words ×256
0x0004_0000
0x3F84_0000
128K bytes (BLOCK6)
128 words ×256
0x0002_0000
0x3F82_0000
64K bytes (BLOCK7)
128 words ×128
0x0001_0000
0x3F81_0000
32K bytes (BLOCK9)
128 words ×64
0x0000_8000
0x3F80_8000
32K bytes (BLOCK8)
128 words ×64
0x0000_0000
0x3F80_0000

Figure 28-10 Block Configuration of Flash Memory (TMPM367FDFG)

28.3.1.2 Basic Operation

This flash memory device has the following two operation modes:

- The mode to read memory data (Read mode)
- The mode to automatically erase or rewrite memory data (Automatic operation)

Transition to the automatic mode is made by executing a command sequence while it is in the memory read mode. In the automatic operation mode, flash memory data cannot be read and any commands stored in the flash memory cannot be executed. In the automatic operation mode, any interrupt or exception generation cannot set the device to the read mode except when a hardware reset is generated. During automatic operation, be sure not to cause any exception other than reset and debug exceptions while a debug port is connected. Any exception generation cannot set the device to the read mode except when a hardware reset is generated.

(1) Read

When data is to be read, the flash memory must be set to the read mode. The flash memory will be set to the read mode immediately after power is applied, when CPU reset is removed, or when an automatic operation is normally terminated. In order to return to the read mode from other modes or after an automatic operation has been abnormally terminated, either the Read/reset command (a software command to be described later) or a hardware reset is used. The device must also be in the read mode when any command written on the flash memory is to be executed.

- Read / reset command and Read command (software reset)

  When ID-Read command is used, the reading operation is terminated instead of automatically returning to the read mode. In this case, the Read/reset command can be used to return the flash memory to the read mode. Also, when a command that has not been completely written has to be canceled, the Read/reset command must be used. The Read command is used to return to the read mode after executing 32-bit data transfer command to write the data "0x0000_00F0" to an arbitrary address of the flash memory.
- With the Read/reset command, the device is returned to the read mode after completing the third bus write cycle.

(2) Command write

This flash memory uses the command control method. Commands are executed by executing a command sequence to the flash memory. The flash memory executes automatic operation commands according to the address and data combinations applied (refer to Command Sequence).

If it is desired to cancel a command write operation already in progress or when any incorrect command sequence has been entered, the Read/reset command is to be executed. Then, the flash memory will terminate the command execution and return to the read.

While commands are generally comprised of several bus cycles and the operation applying to the 32-bit (word) data transmission command to the flash memory is called "bus write cycle". The bus write cycles have a specific sequential order and the flash memory will perform an automatic operation when the sequence of the bus write cycle data and address of command write is operated in accordance with a predefined specific order. If any bus write cycle does not follow a predefined command write sequence, the flash memory will terminate the command execution and return to the read mode.

Note 1: **Command sequences are executed from outside the flash memory area.**

Note 2: **Each bus write cycle must be sequentially executed by 32-bit data transmit command. While a command sequence is being executed, access to the flash memory is prohibited. Also, do not generate any interrupt (except debug exceptions when a debug port is connected). If such an operation is made, it may result in an unexpected read access to the flash memory, and the command sequencer may not be able to correctly recognize**

the command. While it may cause an abnormal termination of the command sequence, it also may cause an incorrect recognition of the command.

Note 3: **For the command sequencer to recognize a command, the device must be in the read mode prior to executing the command. Be sure to check before the first bus write cycle where FCFLCS <RDY / BSY> is set to "1". It is recommended to subsequently execute a Read command.**

Note 4: **Upon issuing a command, if any address or data is incorrectly written, be sure to perform a software reset to return to the read mode again.**

28.3.1.3 Commands

(1) Automatic Page Program

Writing to a flash memory device is to change "1" data cells to "0" data cells. Any "0" data cell cannot be changed to a "1" data cell. For changing "0" data cells to "1" data cells, it is necessary to perform an erase operation.

The automatic page programming function of this device writes data of each page. The TMPM367FDFG contains 128 words in a page. A 128 word block is defined by the same [31:9] address. It starts from the address [8:0] = 0x00 and ends at the address [8:0] = 0x1FF. This programming unit is hereafter referred to as a "page".

Writing to data cells is automatically performed by an internal sequencer and no external control by the CPU is required. The state of automatic page programming (whether it is in writing operation or not) can be checked by FCFLCS [0] <RDY_BSY>.

Also, any new command sequence is not accepted while it is in the automatic page programming mode. If it is desired to interrupt the automatic page programming, use the hardware reset function. If the operation is stopped by a hardware reset operation, it is necessary to once erase the page and then perform the automatic page programming again because writing to the page has not been normally terminated.

The automatic page programming operation is allowed only once for a page already erased. No programming can be performed twice or more. Note that rewriting to a page that has been once written requires execution of the automatic block erase or automatic chip erase command before executing the automatic page programming command again.

No automatic verify operation is performed internally to the device. So, be sure to read the data programmed to confirm that it has been correctly written.

The automatic page programming operation starts when the third bus write cycle of the command cycle is completed. After the fifth bus write cycle, data will be written sequentially starting from the next address of the address specified in the fourth bus write cycle (in the fourth bus write cycle, the page top address will be command written) (32 bits of data is input at one time). Be sure to use the 32-bit data transfer command in writing commands after the fourth bus cycle. At this time, any 32-bit data transfer commands shall not be placed across word boundary. After the fifth bus write cycle, data is command written to the same page area. Even if it is desired to write the page only partially, it is required to perform the automatic page programming for the entire page. In this case, the address input for the fourth bus write cycle shall be set to the top address of the page. Be sure to perform command write operation with the input data set to "1" for the data cells not to be set to "0". For example, if the top address of a page is not to be written, set the input data in the fourth bus write cycle to 0xFFFFFFFF as a command write.

Once the third bus cycle is executed, the automatic page programming is in operation. This condition can be checked by monitoring FCFLCS<RDY / BSY>. Any new command sequence is not accepted while it is in automatic page programming mode. If it is desired to stop operation, use the hardware reset function. Be careful in doing so because data cannot be written normally if the operation is interrupted. When a single page has been command written with normally terminating the automatic page writing process, FCFLCS<RDY / BSY> is set to "1" then it returns to the read mode.

When multiple pages are to be written, it is necessary to execute the page programming command for each page because the number of pages to be written by a single execution of the automatic page program command is limited to only one page. It is not allowed for automatic page programming to process input data across pages.

Data cannot be written to a protected block. When automatic programming is finished, it automatically returns to the read mode. This condition can be checked by monitoring FCFLCS<RDY_BSY>. If automatic programming has failed, the flash memory is locked in the current mode and will not return to the read mode. For returning to the read mode, it is necessary to execute hardware reset to reset the flash memory or the device. In this case, while writing to the address has failed, it is recommended not to use the device or not to use the block that includes the failed address.

Note: **Software reset becomes ineffective after the fourth bus write cycle of the automatic page programming command.**

(2) Automatic chip erase

The automatic chip erase operation starts when the sixth bus write cycle of the command cycle is completed.

This condition can be checked by monitoring FCFLCS<RDY / BSY>. While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic chip erase operation. If it is desired to stop operation, use the hardware reset function. If the operation is forced to stop, it is necessary to perform the automatic chip erase operation again because the data erasing operation has not been normally terminated.

Also, any protected block cannot be erased. If all the blocks are protected, the automatic chip erase operation will not be performed and it returns to the read mode after completing the sixth bus read cycle of the command sequence. When an automatic chip erase operation is normally terminated, it automatically returns to the read mode. If an automatic chip erase operation has failed, the flash memory is locked in the current mode and will not return to the read mode.

For returning to the read mode, it is necessary to execute hardware reset to reset the device. In this case, the failed block cannot be detected. It is recommended not to use the device anymore or to identify the failed block by using the block erase function for not to use the identified block anymore.

(3) Automatic block erase (for each block)

The automatic block erase operation starts when the sixth bus write cycle of the command cycle is completed.

This status of the automatic block erase operation can be checked by monitoring FCFLCS <RDY / BSY>. While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic block erase operation. If it is desired to stop operation, use the hardware reset function. In this case, it is necessary to perform the automatic block erase operation again because the data erasing operation has not been normally terminated.

Also, any protected block cannot be erased. If an automatic block erase operation has failed, the flash memory is locked in the mode and will not return to the read mode. In this case, execute hardware reset to reset the device.

(4) Automatic programming of protection bits (for each block)

This device is implemented with protection bits. This protection can be set for each block. See Table 28-23 for table of protection bit addresses. This device assigns 1 bit to 1 block as a protection bit. The applicable protection bit is specified by PBA in the seventh bus write cycle. By automatically programming the protection bits, write and/or erase functions can be inhibited (for protection) individually for each block. The protection status of each block can be checked by FCFLCS <BLPRO> to be described later. This status of the automatic programming operation to set protection bits can be checked by monitoring FCFLCS <RDY_BSY>. Any new command sequence is not accepted while automatic programming is in progress to program the protection bits. If it is desired to stop the programming operation, use the hardware reset function. In this case, it is necessary to perform the programming operation again because the protection bits may not have been correctly programmed. If all the protection bits have been programmed, all FCFLCS <BLPRO> are set to "1" indicating that it is in the protected state. This disables subsequent writing and erasing of all blocks.

Note: **Software reset is ineffective in the seventh bus write cycle of the automatic protection bit programming command. FCFLCS <RDY_BSY> turns to "0" after entering the seventh bus write cycle.**

(5) Automatic erasing of protection bits

Different results will be obtained when the automatic protection bit erase command is executed depending on the status of the protection bits and the security bits. It depends on whether all <BLPRO> in the FCFLCS register are set to "1" or not, when FCSECBIT<FCSECBIT> is set to "1". Be sure to check the value of FCFLCS <BLPRO> before executing the automatic protection bit erase command. See Chapter "Protect/security function" for details.

- When all the FCFLCS <BLPRO> are set to "1" (all the protection bits are programmed):

  When the automatic protection bit erase command is command written, the flash memory is automatically initialized within the device. When the seventh bus write cycle is completed, the entire area of the flash memory data cells is erased and then the protection bits are erased. This operation can be checked by monitoring FCFLCS <RDY_BSY>. If the automatic operation to erase protection bits is normally terminated, FCFLCS will be set to "0x00000001". Since no automatic verify operation is performed internally to the device, be sure to read the data to confirm that it has been correctly erased. For returning to the read mode while the automatic operation after the seventh bus cycle is in progress, it is necessary to use the hardware reset to reset the device. If this is done, it is necessary to check the status of protection bits by FCFLCS <BLPRO> after retuning to the read mode and perform either the automatic protection bit erase, automatic chip erase, or automatic block erase operation, as required.
- When FCFLCS <BLPRO> include "0" (not all the protection bits are programmed):

  If the automatic protection bit is cleared to "0", the protection condition is canceled. With this device, protection bits can be programmed to an individual block and performed bit-erase operation in the four bits unit as shown in Table 28-23. The target bits are specified in the seventh bus write cycle.The protection status of each block can be checked by FCFLCS <BLPRO> to be described later. This status of the programming operation for automatic protection bits can be checked by monitoring FCFLCS <RDY_BSY>. When the automatic operation to erase protection bits is normally terminated, the protection bits of FCFLCS <BLPRO> selected for erasure are set to "0".

In any case, any new command sequence is not accepted while it is in an automatic operation to erase protection bits. If it is desired to stop the operation, use the hardware reset function. When the automatic operation to erase protection bits is normally terminated, it returns to the read mode.

Note: **The FCFLCS <RDY / BSY> bit is "0" while in automatic operation and it turns to "1" when the automatic operation is terminated.**

(6) ID-Read

Using the ID-Read command, you can obtain the type and other information on the flash memory contained in the device. The data to be loaded will be different depending on the address [15:14] of the fourth and subsequent bus write cycles (recommended input data is 0x00). After the fourth bus write cycle, when an arbitrary flash memory area is read, the ID value will be loaded. Once the fourth bus write cycle of an ID-Read command has passed, the device will not automatically return to the read mode. In this condition, the set of the fourth bus write cycle and ID-Read commands can be repeatedly executed. For returning to the read mode, use the Read/reset command or hardware reset command.

28.3.1.4 Flash control / status register

Base Address = 0x41FF_F000

Register name		Address (Base+)
Reserved	-	0x0000, 0x0004
Security bit register	FCSECBIT	0x0010
Reserved	-	0x0014
Flash control register	FCFLCS	0x0020
Reserved	-	0x0024, 0x0028
Reserved	-	0x0040, 0x0044
Reserved	-	0x0050, 0x0058
Reserved	-	0x0060 to 0x00B8

Note: **Do not access to the reserved address.**

(1) FCFLCS (Flash control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
After reset	0	0	(Note 2)	(Note 2)	(Note 2)	(Note 2)	(Note 2)	(Note 2)
	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 to 22	−	R	Read as 0.
21to 16	BLPRO5 to BLPRO0	R	Protection for Block 5 to 0 0: disabled 1: enabled Each of the protection bits represents the protection status of the corresponding block. When a bit is set to "1", it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.
15 to 1	−	R	Read as 0.
0	RDY_BSY	R	Ready / Busy (Note 1) 0:Auto operating 1:Auto operation terminated. Ready/Busy flag bit The RDY_BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1".

Note 1: **This command must be issued in the ready state. Issuing the command in the busy state may disable both correct command transmission and further command input. To exit from the condition, execute system reset. Refer to "Reset Operation" regarding to reset in this case.**

Note 2: **The value varies depending on protection applied.**

(2) FCSECBIT (Security bit register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	SECBIT	R/W	Security bits 0:disabled 1:enabled

Note: **This register is initialized by cold reset or releasing STOP2 mode of low power consumption mode.**

28.3.1.5 List of Command Sequences

Table 28-20 shows the address and the data of each command of flash memory.

Bus cycles are "bus write cycles" except for the second bus cycle of the Read command, the fourth bus-cycle of the Read/reset command, and the fifth bus cycle of the ID-Read command. Bus write cycles are executed by 32-bit (word) data transfer commands. (In the following table, only lower 8 bits data are shown.)

See Table 28-21 for the detail of the address bit configuration. Use a value of "Addr." in the Table 28-20 for the address [15:8] of the normal command in the Table 28-21.

Note: **Always set "0" to the address bits [1:0] in the entire bus cycle.**

Table 28-20 Flash Memory Access from the Internal CPU

Command sequence	First bus cycle	Second bus cycle	Third bus cycle	Fourth bus cycle	Fifth bus cycle	Sixth bus cycle	Seventh bus cycle
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	Data	Data	Data	Data	Data	Data	Data
Read	0xXX	−	−	−	−	−	−
	0xF0	−	−	−	−	−	−
Read / Reset	0x54XX	0xAAXX	0x54XX	RA	−	−	−
	0xAA	0x55	0xF0	RD	−	−	−
ID-Read	0x54XX	0xAAXX	0x54XX	IA	0xXX	−	−
	0xAA	0x55	0x90	0x00	ID	−	−
Automatic page programming	0x54XX	0xAAXX	0x54XX	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic chip erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	−
	0xAA	0x55	0x80	0xAA	0x55	0x10	−
Auto block erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	BA	−
	0xAA	0x55	0x80	0xAA	0x55	0x30	−
Protection bit programming	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
Protection bit erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

Supplementary explanation

- RA: Read address
- RD: Read data
- IA: ID address
- ID: ID data
- PA: Program page address
  PD: Program data (32 bit data)

  After fourth bus cycle, enter data in the order of the address for a page.
- BA: Block address
- PBA: Protection bit address

28.3.2 Address bit configuration for bus write cycles

Table 28-21 is used in conjunction with "Table 28-20 Flash Memory Access from the Internal CPU".

Address setting can be performed according to the normal bus write cycle address configuration from the first bus cycle. "0" is recommended" in the Table 28-21 Address Bit Configuration for Bus Write Cycles can be changed as necessary.

<table>
<tr><th>Address</th><th>Addr
[31:19]</th><th>Addr
[18]</th><th>Addr
[17]</th><th>Addr
[16]</th><th>Addr
[15]</th><th>Addr
[14]</th><th>Addr
[13:11]</th><th>Addr
[10]</th><th>Addr
[9]</th><th>Addr
[8]</th><th>Addr
[7:0]</th></tr>
<tr><td rowspan="2">Normal commands</td><td colspan="11">Normal bus write cycle address configuration</td></tr>
<tr><td>Flash area</td><td colspan="3">"0" is recommended.</td><td colspan="6">Command</td><td>Addr[1:0]="0" (fixed)
Others:0
(recommended)</td></tr>
<tr><td rowspan="2">ID-READ</td><td colspan="11">IA: ID address (Set the fourth bus write cycle address for ID-Read operation)</td></tr>
<tr><td>Flash area</td><td colspan="3">"0" is recommended.</td><td colspan="2">ID address</td><td colspan="5">Addr[1:0]="0" (fixed), Others:0 (recommended)</td></tr>
<tr><td rowspan="2">Block erase</td><td colspan="11">BA: Block address (Set the sixth bus write cycle address for block erase operation)</td></tr>
<tr><td colspan="6">Block selection (Table 28-21)</td><td colspan="5">Addr[1:0]="0" (fixed), Others:0 (recommended)</td></tr>
<tr><td rowspan="2">Auto page programming</td><td colspan="11">PA: Program page address (Set the fourth bus write cycle address for page programming operation)</td></tr>
<tr><td colspan="9">Page selection</td><td colspan="2">Addr[1:0]="0" (fixed)
Others:0 (recommended)</td></tr>
<tr><td rowspan="2">Protection bit programming</td><td colspan="11">PBA: Protection bit address (Set the seventh bus write cycle address for protection bit programming)</td></tr>
<tr><td>Flash area</td><td colspan="2">Protection bit selection
(Table 28-22)</td><td colspan="4">Fixed to "0".</td><td colspan="2">Protect bit selection
(Table 28-22)</td><td colspan="2">Addr[1:0]="0" (fixed)
Others:0 (recommended)</td></tr>
<tr><td rowspan="2">Protection bit erase</td><td colspan="11">PBA: Protection bit address (Set the seventh bus erase cycle address for protection bit erasure)</td></tr>
<tr><td>Flash area</td><td colspan="2">Protection bit selection
(Table 28-23)</td><td colspan="5">Fixed to "0".</td><td colspan="3">Addr[1:0]="0" (fixed)
Others:0 (recommended)</td></tr>
</table>

As block address, specify any address in the block to be erased.

Table 28-21 Block Address Table

TMPM367FDFG

Block	Address (User boot mode)	Address (Single boot mode)	Size (K byte)
4	0x0000_0000 to 0x0000_7FFF	0x3F80_0000 to 0x3F80_7FFF	32
5	0x0000_8000 to 0x0000_FFFF	0x3F80_8000 to 0x3F80_FFFF	32
3	0x0001_0000 to 0x0001_FFFF	0x3F81_0000 to 0x3F81_FFFF	64
2	0x0002_0000 to 0x0003_FFFF	0x3F82_0000 to 0x3F83_FFFF	128
1	0x0004_0000 to 0x0005_FFFF	0x3F84_0000 to 0x3F85_FFFF	128
0	0x0006_0000 to 0x0007_FFFF	0x3F86_0000 to 0x3F87_FFFF	128

Note: As for the addresses from the first to the fifth bus cycles, specify the upper addresses of the blocks to be erased.

Table 28-22 Block Address Table

TMPM367FDFG

Block	Protect bit	The seventh bus write cycle address					
		Address [18]	Address [17]	Address [16:11]	Address [10]	Address 9]	Address [8]
Block0	<BLPRO[0]>	0	0	Fixed to "0".	0	0	Fixed to "0"
Block1	<BLPRO[1]>	0	0		0	1	
Block2	<BLPRO[2]>	0	0		1	0	
Block3	<BLPRO[3]>	0	0		1	1	
Block4	<BLPRO[4]>	0	1		0	0	
Block5	<BLPRO[5]>	0	1		0	1	

Table 28-23 Protection Bit Erase Address Table

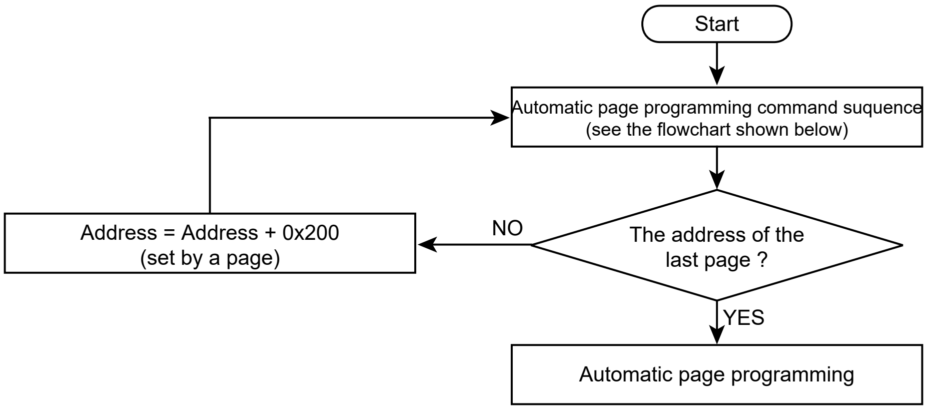
Block	Protection bit	The seventh bus write cycle address [18:17]	
		Address [18]	Address [17]
Block3 to 0	<BLPRO[3:0]>	0	0
Block5 to 4	<BLPRO[5:4]>	0	1

Note: **The protection bit erase command cannot erase by individual block.**

Table 28-24 The ID-Read command's fourth bus write cycle ID address (IA) and the data to be read by the following 32-bit data transfer command (ID)

IA[15:14]	ID[7:0]	Code
00	0x98	Manufacturer code
01	0x5A	Device code
10	Reserved	−
11	0x10	Macro code

28.3.2.1 Flowchart


Start
Automatic page programming command suquence (see the flowchart shown below)
The address of the last page ?
NO
Address = Address + 0x200 (set by a page)
YES
Automatic page programming

Automatic Page Programming Command Sequence (Address / Command)

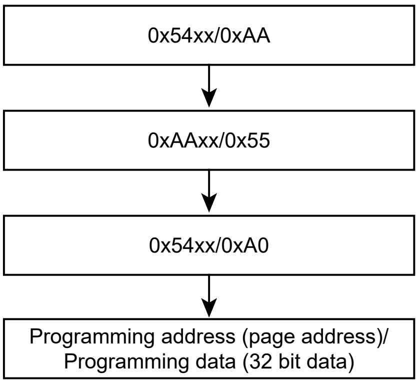

0x54xx/0xAA
0xAAxx/0x55
0x54xx/0xA0
Programming address (page address)/ Programming data (32 bit data)

Figure 28-11 Automatic Programming

Note: **Command sequence is executed by 0x54xx or 0x55xx.**

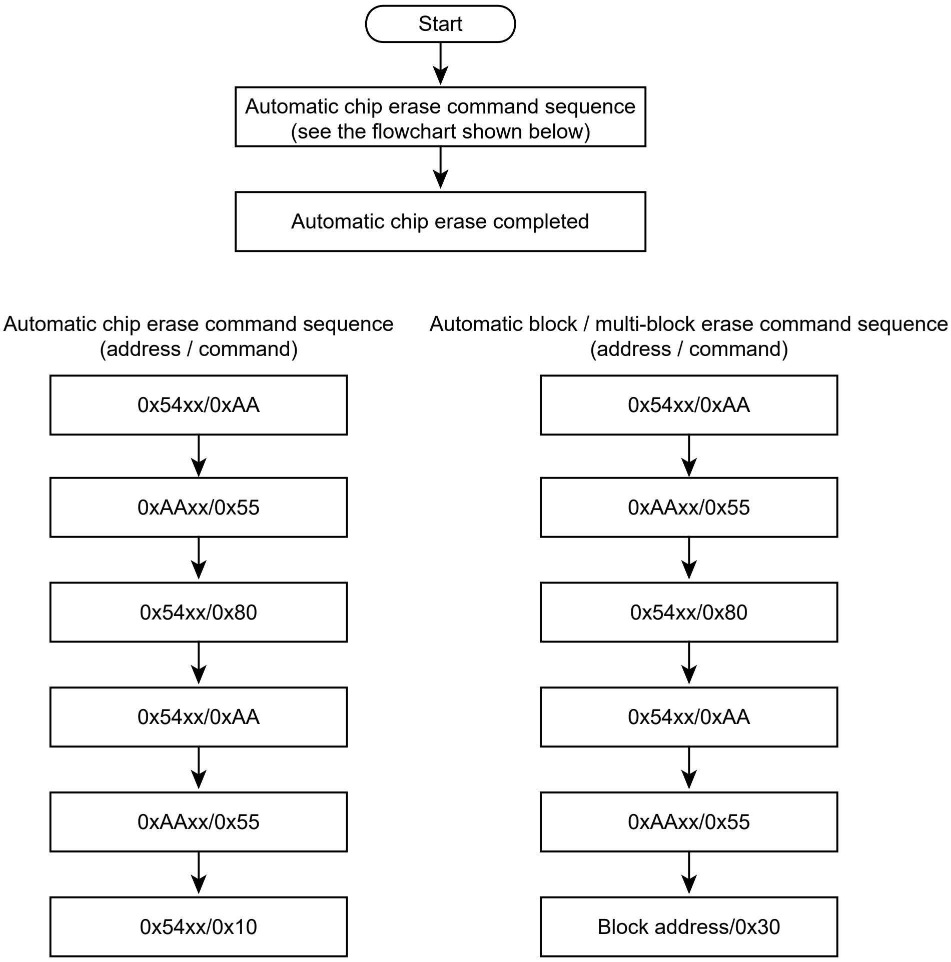

Start
Automatic chip erase command sequence (see the flowchart shown below)
Automatic chip erase completed
Automatic chip erase command sequence (address / command)
0x54xx/0xAA
0xAAxx/0x55
0x54xx/0x80
0x54xx/0xAA
0xAAxx/0x55
0x54xx/0x10
Automatic block / multi-block erase command sequence (address / command)
0x54xx/0xAA
0xAAxx/0x55
0x54xx/0x80
0x54xx/0xAA
0xAAxx/0x55
Block address/0x30

Figure 28-12 Automatic Erase

Note: **Command sequence is executed by 0x54xx or 0x55xx.**

29. ROM protection

29.1 Outline

The TMPM367FDFG offers two kinds of ROM protection/ security functions.

One is a write/ erase-protection function for the internal flash ROM data.

The other is a security function that restricts internal flash ROM data readout and debugging.

29.2 Features

29.2.1 Write/ erase-protection function

The write/ erase-protection function enables the internal flash to prohibit the writing and erasing operation for each block.

To activate the function, write "1" to the corresponding bits to a block to protect. Writing "0" to the bits cancels the protection.

The protection settings of the bits can be monitored by the FCFLCS <BLPRO[5:0]> bit. See the chapter "Flash" for programming details.

29.2.2 Security function

The security function restricts flash ROM data readout and debugging.

This function is available under the conditions shown below.

1. The FCSECBIT <SECBIT> bit is set to "1".
2. All the protection bits (the FCFLCS<BLPRO> bits) used for the write/erase-protection function are set to "1".

Table 29-1 shows details of the restrictions by the security function.

Table 29-1 Restrictions by the security function

Item	Details
1) ROM data readout	Data can be read from CPU.
2) Debug port	Communication of JTAG/SW and trace are prohibited
3) Command for flash memory	Writing a command to the flash memory is prohibited. An attempt to erase the contents in the bits used for the write/ erase-protection erases all the protection bits.

29.3 Register

Base Address = 0x41FF_F000

Register name		Address (Base+)
Reserved	-	0x0000, 0x0004
Security bit register	FCSECBIT	0x0010
Reserved	-	0x0014
Flash control register	FCFLCS	0x0020
Reserved	-	0x0024 to 0x0FFF

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

29.3.1 FCFLCS (Flash control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
After reset	0	0	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	RDY_BSY
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-22	−	R	Read as 0.
21-16	BLPRO5 to BLPRO0	R	Protection for Block5 to 0 0: disabled 1: enabled Protection status bits Each of the protection bits represents the protection status of the corresponding block. When a bit is set to "1", it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.
15-1	−	R	Read as 0.
0	RDY_BSY	R	Ready/Busy (Note 1) 0: Auto operating 1:Auto operation terminated Ready/Busy flag bit The RDY_BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1".

Note 1: **This command must be issued in the ready state. Issuing the command in the busy state may disable both correct command transmission and further command input. To exit from the condition, execute system reset. Refer to "**Reset" in "Flash" section regarding to reset in this case.

Note 2: **The value varies depending on protection applied.**

29.3.2 FCSECBIT (Security bit register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	SECBIT	R/W	Security bit 0: Disabled 1: Enabled

Note: This register is initialized by cold reset and releasing STOP2 mode of the low power consumption mode.

29.4 Writing and erasing

29.4.1 Protection bits

To write the protection bits or to erase the protection bits is used by command sequence.

Writing to the protection bits is done on block-by-block basis. Erasing to the protection bits is done on the unit, from block0 to block3 and from block 4 to block 5.

When the settings for all the blocks are "1" and FCSECBIT<SECBIT> is "1", the security function is enabled. In this condition, the chip erase is done and all protect bits are erased by erasing the protect bits. Therefore, the protect bit should be erased after FCSECBIT<SECBIT> makes "0".

See the chapter "Flash" for details.

29.4.2 Security bit

The FCSECBIT <SECBIT> bit that activates security function is set to "1" at a power-on reset right after power-on.

The bit is rewritten by the following procedure.

1. Write the code 0xa74a9d23 to FCSECBIT register.
2. Write data within 16 clocks from the above.1.

Note: The above procedure is enabled only when using 32-bit data transfer command.

30. Debug Interface

30.1 Specification Overview

TMPM367FDFG contains the Serial Wire JTAG Debug Port (SWJ-DP) unit for interfacing with the debugging tools and the Embedded Trace Macrocell™(ETM) unit for instruction trace output.Trace data is output to the dedicated pins(TRACEDATA[3:0], SWV) for the debugging via the on-chip Trace Port Interface Unit (TPIU).

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

30.2 SWJ-DP

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

30.3 ETM

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

30.4 Pin Functions

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

The PA1 and PA2 pins are shared between the JTAG debug port function and the Serial Wire Debug Port function. The PA0 pin is shared between the JTAG debug port function and the SWV trace output function.

Table 30-1 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	PA1	Input	JTAG Test Mode Selection	I / O	Serial Wire Data Input/Output
TCK / SWCLK	PA2	Input	JTAG Test Check	Input	Serial Wire Clock
TDO / SWV	PA0	Output	JTAG Test Data Output	(Output) (Note)	(Serial Wire Viewer Output)
TDI	PA3	Input	JTAG Test Data Input	-	-
$\overline{\text{TRST}}$	PA4	Input	JTAG Test $\overline{\text{RESET}}$	-	-
TRACECLK	PA5	Output	TRACE Clock Output		
TRACEDATA0	PA6	Output	TRACE DATA Output0		
TRACEDATA1	PA7	Output	TRACE DATA Output1		
TRACEDATA2	PB0	Output	TRACE DATA Output2		
TRACEDATA3	PB1	Output	TRACE DATA Output3		

Note: When SWV function is used.

After reset, PA0, PA1 and, PA2, PA3 and PA4 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: If PA1 and PA0 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>.

Table 30-2 summarizes the debug interface pin and related port settings after reset.

Table 30-2 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)	Open-drain (PxOD)
PA1	TMS/SWDIO	1	1	1	1	0	0
PA2	TCK/SWCLK	1	1	0	0	1	0
PA0	TDO/SWV	1	0	1	0	0	0
PA3	TDI	1	1	0	1	0	0
PA4	$\overline{\text{TRST}}$	1	1	0	1	0	0
PA5	TRACECLK	0	0	0	0	0	0
PA6	TRACEDATA0	0	0	0	0	0	0
PA7	TRACEDATA1	0	0	0	0	0	0
PB0	TRACEDATA2	0	0	0	0	0	0
PB1	TRACEDATA3	0	0	0	0	0	0

30.5 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.

30.6 Connection with a Debug Tool

30.6.1 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.

Note: Ensure that to measure the power-consumption with debug tool connected in STOP1/STOP2 mode is prohibitted.

30.6.2 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 30-3 Example Table of using debug interface pins

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

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

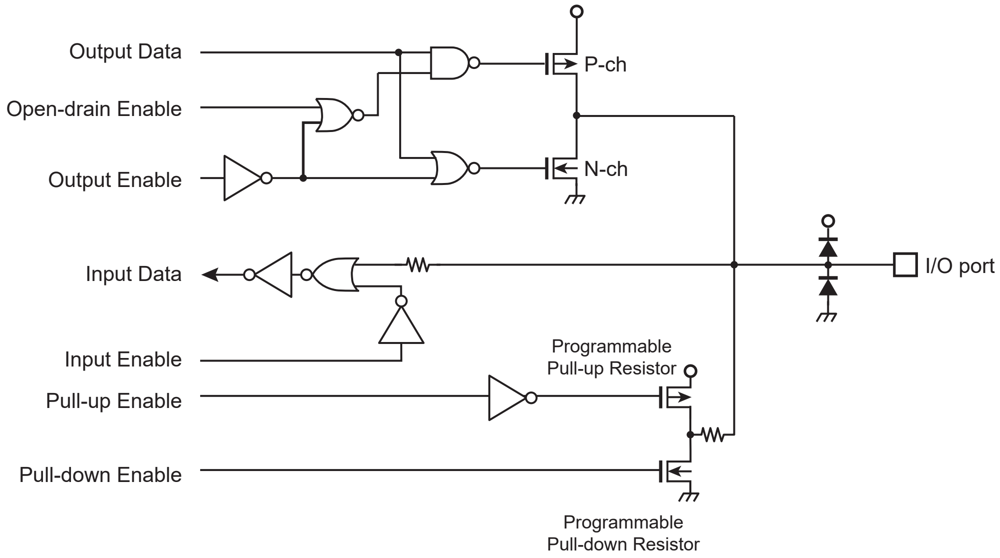
Note: For the treatment of the pin of which the $\overline{TRST}$ function is assigned, select the $\overline{TRST}$ function with the function register and set the pin to OPEN or "High level".

31. Port Section Equivalent Circuit Schematic

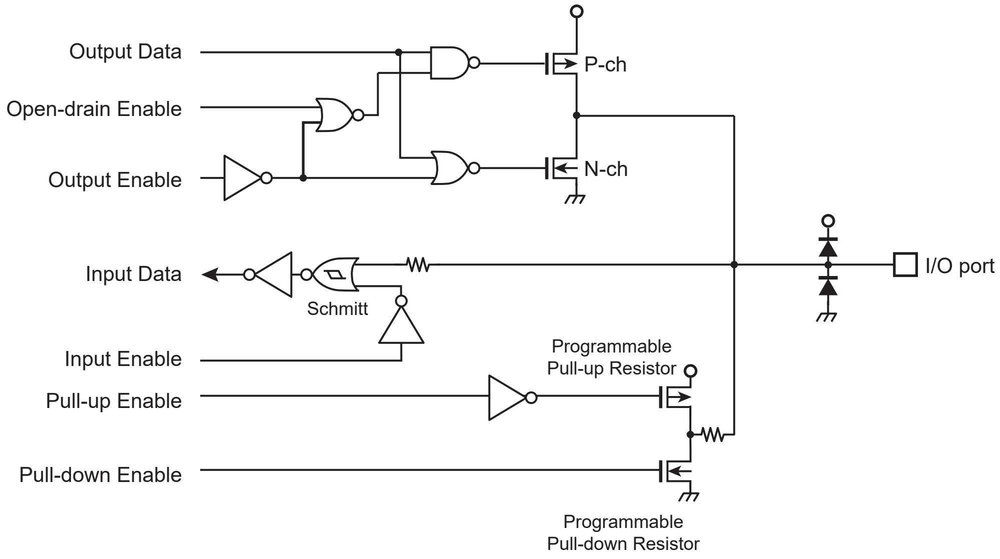
Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

The input protection resistance ranges from several tens of OHM to several hundred OHM. Feedback resistor and Damp-ing resistorare shown with a typical value.

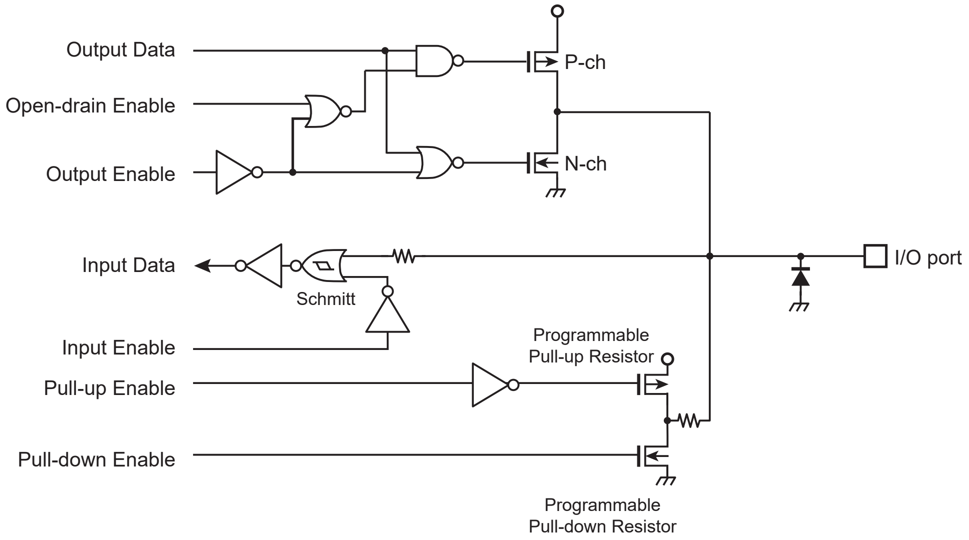
31.1 PB4, PK2


Output Data
Open-drain Enable
Output Enable
Input Data
Input Enable
Pull-up Enable
Pull-down Enable
P-ch
N-ch
I/O port
Programmable
Pull-up Resistor
Programmable
Pull-down Resistor

31.2 PA0 to 7, PB0 to 3, 5, PE0 to 7, PF0 to 7, PG0 to 7, PH0 to 3, PK1, PK3 to 4, PL0 to 3

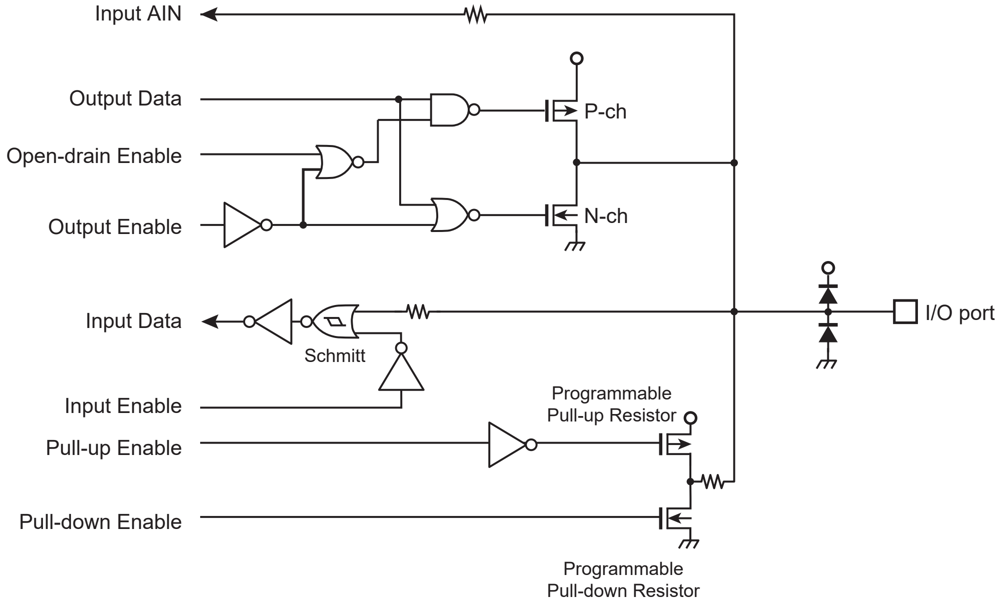

Output Data
Open-drain Enable
Output Enable
Input Data
Schmitt
Input Enable
Pull-up Enable
Pull-down Enable
P-ch
N-ch
I/O port
Programmable
Pull-up Resistor
Programmable
Pull-down Resistor

31.3 PK0

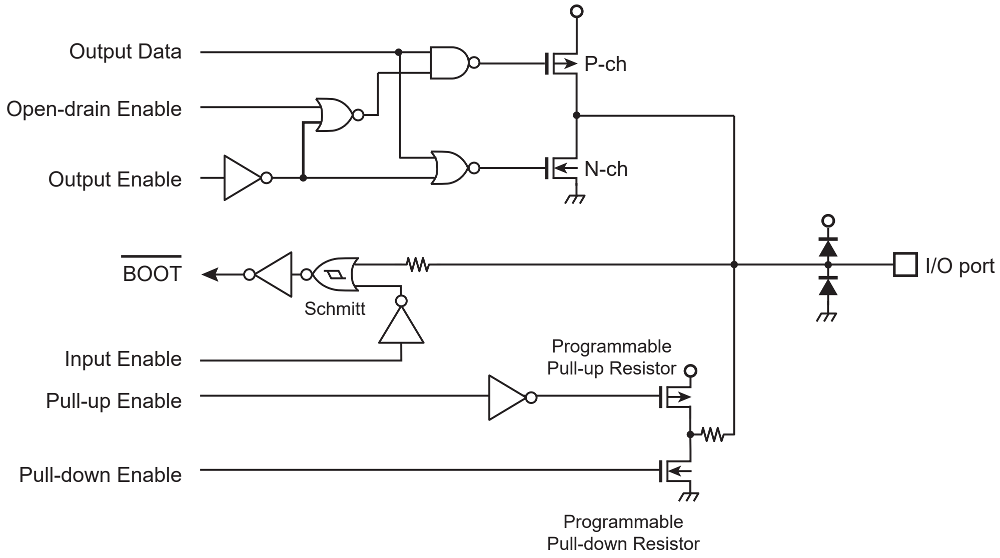

Output Data
Open-drain Enable
Output Enable
Input Data
Input Enable
Pull-up Enable
Pull-down Enable
P-ch
N-ch
I/O port
Schmitt
Programmable
Pull-up Resistor
Programmable
Pull-down Resistor

Note: Only when input is enabled, these pins tolerate 5V inputs.

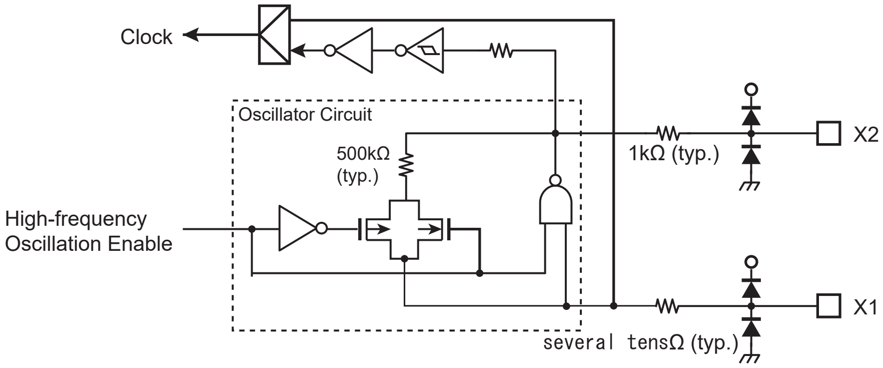
31.4 PI0 to 7


Input AIN
Output Data
Open-drain Enable
Output Enable
Input Data
Input Enable
Pull-up Enable
Pull-down Enable
P-ch
N-ch
I/O port
Schmitt
Programmable
Pull-up Resistor
Programmable
Pull-down Resistor

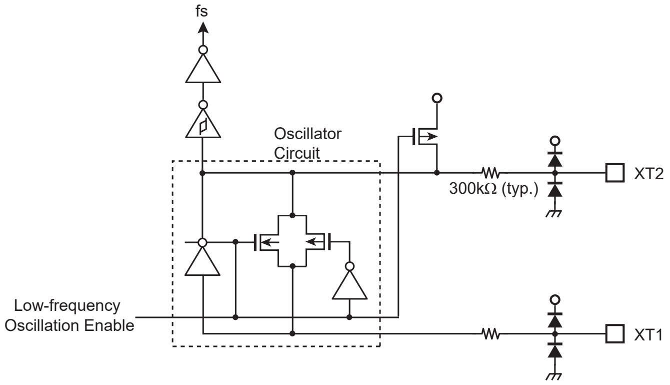
31.5 PB6


Output Data
Open-drain Enable
Output Enable
P-ch
N-ch
BOOT
Schmitt
I/O port
Input Enable
Pull-up Enable
Pull-down Enable
Programmable
Pull-up Resistor
Programmable
Pull-down Resistor

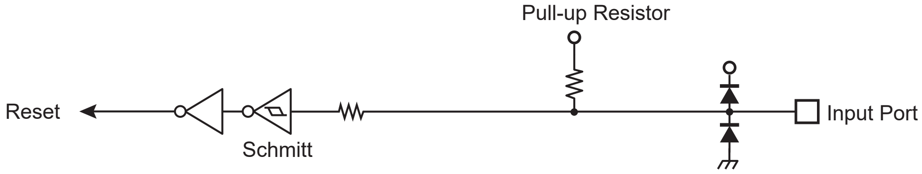
31.6 X1, X2


Clock
Oscillator Circuit
500kΩ
(typ.)
1kΩ (typ.)
X2
High-frequency
Oscillation Enable
X1
several tensΩ (typ.)

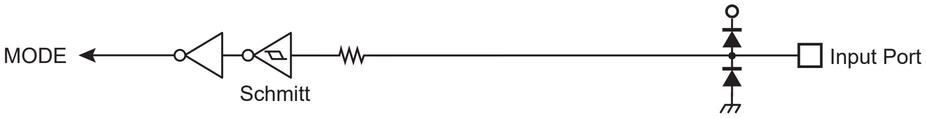
31.7 XT1, XT2


fs
Oscillator
Circuit
300kΩ (typ.)
XT2
Low-frequency
Oscillation Enable
XT1

31.8 $\overline{\text{RESET}}$, $\overline{\text{NMI}}$

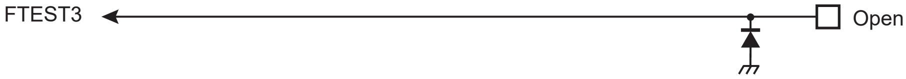

Pull-up Resistor
Reset
Schmitt
Input Port

31.9 MODE


MODE
Schmitt
Input Port

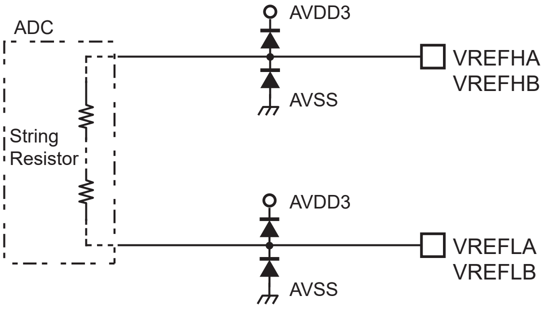
(Note)MODE pin is fixed to GND.

31.10 FTEST3


FTEST3
Open

(Note)FTEST3 pin is fixed to Open.

31.11 VREFHA, VREFHB, VREFLA, VREFLB


ADC
String
Resistor
AVDD3
AVSS
VREFHA
VREFHB
AVDD3
AVSS
VREFLA
VREFLB

32. Electrical Characteristics

32.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		DVDD3A/B	−0.3 to 3.9	V
		RVDD3	−0.3 to 3.9	
		AVDD3A/B	−0.3 to 3.9	
		AVDD3_DA	−0.3 to 3.9	
Input voltage	Except 5V torerant pin	V_{IN1}	−0.3 to VDD + 0.3	V
	5V torerant pin	V_{IN2}	−0.3 to 5.5	
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 Temperature	Except during Flash W/E	T_{OPR}	−40 to 85	°C
	During Flash W/E		0 to 70	

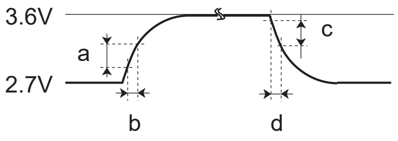
Note: **Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no Absolute maximum rating value is exceeded with respect to current, voltage, power consumption, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning.**

32.2 DC Electrical Characteristics (1/2)

DVDD3A = DVDD3B=RVDD3 = AVDD3A = AVDD3B = AVDD3_DA = 2.7 V to 3.6 V
DVSSA = DVSSB = AVSSA = AVSSB = VREFLA = VREFLB = AVSS_DA = 0V, Ta = −40 to 85 °C

Parameter		Symbol	Condition		Min.	Typ. (Note 1)	Max.	Unit
Supply voltage	DVDD3A DVDD3B AVDD3A,AVDD3B RVDD3	VDD	f_{OSC} = 8 to 16 MHz fsys = 1 to 80 MHz fs = 30 to 34 kHz	without USB	2.7	−	3.6	V
				with USB	3	−	3.45	V
Low-level Input voltage	PB4, PK2	V_{IL1}	2.7 V ≤ VDD ≤ 3.6 V		−0.3	−	0.2 VDD	V
	PA0 to 7, PB0 to 6 PE0 to 7, PF0 to 7, PG0 to 7,PH0 to 3, PI0 to 7 PK1,3 to 4, PL0 to 3 $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, MODE X1	V_{IL2}						
	PK0	V_{IL3}						
High-level Input voltage	PB4, PK2	V_{IH1}	2.7 V ≤ VDD ≤ 3.6 V		0.8 VDD	−	VDD + 0.3	V
	PA0 to 7, PB0 to 6 PE0 to 7, PF0 to 7, PG0 to 7,PH0 to 3, PI0 to 7 PK1,3 to 4, PL0 to 3 $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, MODE X1	V_{IH2}						
	PK0	V_{IH3}					5.5	
Low-level output voltage		V_{OL}	I_{OL} = 2 mA		−	−	0.4	V
High-level output voltage		V_{OH}	I_{OH} = −2 mA		2.4	−	VDD	V
Input leakage current		I_{LI1}	0.0 ≤ V_{IN} ≤ VDD		−	0.02	±5	μA
Output leakage current		I_{LO}	0.2 ≤ V_{IN} ≤ VDD -0.2		−	0.05	±10	
Pull-up resistor at Reset		RRST	−		−	50	75	kΩ
Schmitt trigger input width		VTH1	2.7 V ≤ VDD ≤ 3.6 V		0.3	0.6	−	V
Programmable pull-up/pull-down resistor		PKH	−		−	50	75	kΩ
Power supply variation rate in operation range		VRS	RVDD3 = DVDD3A		−	−	6	mV/μs
		VFS			−	−	-18	
Pin capacitance (Except power supply pins)		C_{IO}	fc = 1 MHz		−	−	10	pF
Low-level output current		I_{OL1}	Per pin		−	−	2	mA
		ΣI_{OL}	Total, all ports		−	−	35	mA
High-level output current		I_{OH1}	Per pin		−	−	-2	mA
		ΣI_{OH}	Total, all ports		−	−	-35	mA

Note 1: Ta = 25 °C, DVDD3A = DVDD3B = RVDD3 = AVDD3A = AVDD3B = 3.3 V, unless otherwise noted.
Note 2: The same voltage must be supplied to DVDD3A, DVDD3B, AVDD3A, AVDD3B, AVDD3_DA and RVDD3.
Note 3: VRS(Rising), VFS(Falling) should be measured at a strict level against a characteristics.



VRS = a / b , VFS = c / d

32.3 DC Electrical Characteristics (2/2)

DVDD3A = DVDD3B=RVDD3 = AVDD3A = AVDD3B = AVDD3_DA = 2.7 V to 3.6 V, Ta = −40 to 85 °C

Parameter	Symbol	Condition	Min.	Typ. (Note 1)	Max.	Unit
NORMAL (Note 2) Gear 1/1	I_{DD}	fsys = 80 MHz	−	60	88	mA
IDLE (Note 3)			−	43	72	
STOP1		−	−	1.0	18	
STOP2		−	−	18	110	μA

Note 1: Ta = 25 °C, DVDD3A = DVDD3B = AVDD3 = RVDD3 = 3.3, unless otherwise noted.

Note 2: Measurement condition of I_{DD} NORMAL :

Execution program: Dhrystone V2.1 (built-in FLASH operation)

All peripheral functions stopped.

Note 3: Measurement condition of I_{DD} IDLE:

All peripheral functions stopped.

The currents flow through DVDD3A, DVDD3B, AVDD3 and RVDD3 are included in IDD.

32.4 12-bit A/D Converter Electrical Characteristics

DVDD3A = RVDD3 = AVDD3A = AVDD3B = 2.7 V to 3.6 V
DVSSA = AVSSA = AVSSB = VREFLA = VREFLB = 0V, Ta = −40 to 85 °C

Parameter		Symbol	Condition	Min.	Typ.	Max	Unit
Analog reference voltage (+)		VREFH	−	AVDD3-0.3	−	AVDD3	V
Analog input voltage		VAIN	−	VREFL	−	VREFH	V
Power supply current of analog reference voltage	AD conversion	IREF	−	−	1.5	2.3	mA
	Non-AD conversion		−	−	0.02	0.1	μA
Consumption current			−	−	1.5	2.5	mA
INL error		−	AIN resistance ≤ 600 Ω AIN load capacitance ≥ 30 pF Conversion time ≥ 1.0 μs	−	5	6	LSB
DNL error				−	3	6	
Zero-scale error				−	3	6	
Full-scale error				−	4	9	
Total error				−	7	9	
INL error		−	AIN resistance ≤ 600 kΩ AIN load capacitance ≥ 0.1μF Conversion time ≥ 1.0 μs	−	5	6	
DNL error				−	3	6	
Zero-scale error				−	3	6	
Full-scale error				−	4	8	
Total error				−	7	8	
INL error		−	AIN resistance ≤ 5 kΩ AIN load capacitance ≥ 0.1μF Conversion time ≥ 1.0 μs	−	5	6	
DNL error				−	3	6	
Zero-scale error				−	3	6	
Full-scale error				−	4	8	
Total error				−	7	8	
Conversion time		Tconv	−	1.0	−	10	μs

Note 1: 1LSB = (VREFH − VREFL)/4096 [V]

Note 2: Peripheral functions are disabled.

32.5 10-bit D/A Converter Electrical Characteristics

DVDD3A = RVDD3 = AVDD3_DA = 2.7 V to 3.6 V
DVSSA = AVSS_DA = 0V, Ta = −40 to 85 °C

Parameter	Symbol	Condition	Min.	Typ.	Max	Unit
Consumption current	IAVDD3_DA	−	−	350	550	μA
Settling time	Tset	−	−	−	3	μs
			−	−	1(Note 2)	
Output current	IOUT	−	−	−	± 1000	μA
Output voltage range	VOUT	AVSS_DA = 0V	+ 0.3	−	AVDD3_DA - 0.3	V
Total error	TERR	−	−	−	2	LSB

Note 1: Ensure that external capacity of output pins (DA0,DA1) are maximum 30pF per one channel.

Note 2: The change margin is within ± 16LSB compared to immediate previous setting value.

Note 3: 1LSB = (AVDD3_DA − AVSS_DA)/1024 [V]

Note 4: Relative accuracy is not guaranteed when two channels operate simultaneously.

32.6 AC Electrical Characteristics

32.6.1 AC Measurement Condition

The AC characteristics data of this chapter is measured under the following conditions unless otherwise noted.

- Output levels: High = 0.8 × DVDD3A, 0.8 × AVDD3A
- Output levels: Low = 0.2 × DVDD3A, 0.2 × AVDD3A
- Input levels: Refer to low-level input voltage and high-level input voltage in "DC Electrical Characteristics".
- Load capacity: CL = 30pF

32.6.2 Serial Channel (SIO/UART)

32.6.2.1 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

[Data Input]

Parameter	Symbol	Equation		48 MHz		80 MHz		Unit
		Min.	Max	Min.	Max	Min.	Max	
SCLK Clock High width (input)	t_{SCH}	4x	−	83.3	−	50	−	ns
SCLK Clock Low width (input)	t_{SCL}	4x	−	83.3	−	50	−	
SCLK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	−	166.6	−	100	−	
Valid Data Input ← SCLK rise or fall (Note 1)	t_{SRD}	30	−	30.0	−	30.0	−	
SCLK rise or fall → Input Data hold (Note 1)	t_{HSR}	x + 30	−	50.8	−	42.5	−	

[Data Output]

Parameter	Symbol	Equation		48 MHz		80 MHz		Unit
		Min.	Max	Min.	Max	Min.	Max	
SCLK Clock High width (input)	t_{SCH}	4x	−	107.5 (Note 3)	−	82.5 (Note 3)	−	ns
SCLK Clock Low width (input)	t_{SCL}	4x	−	107.5 (Note 3)	−	82.5 (Note 3)	−	
SCLK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	−	215	−	165	−	
Output Data ← SCLK rise or fall (Note 1)	t_{OSS}	$t_{SCY}/2 - 3x - 45$	−	0 (Note 2)	−	0 (Note 2)	−	
SCLK rise or fall → Output Data hold (Note 1)	t_{OHS}	$t_{SCY}/2$	−	107.5	−	82.5	−	

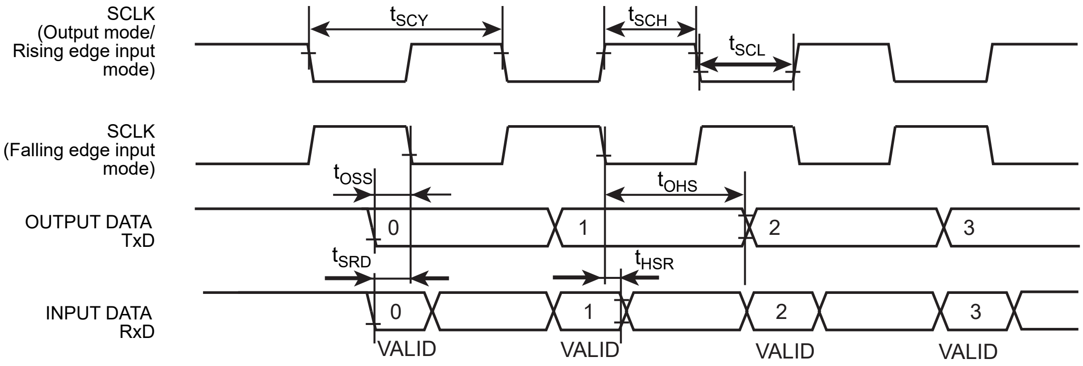
Note 1: SCLK rise/fall : SCLK rise mode uses the rise timing of SCLK. SCLK fall mode uses the fall timing of SCLK.

Note 2: Use the frequency 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		48 MHz		80 MHz		Unit
		Min.	Max	Min.	Max	Min.	Max	
SCLK cycle (programmable)	t_{SCY}	4x	–	83.2	–	50	–	ns
Output Data ← SCLK rise	t_{OSS}	$t_{SCY}/2 - 20$	–	21.6	–	5	–	
SCLK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 - 20$	–	21.6	–	5	–	
Valid Data Input ← SCLK rise	t_{SRD}	45	–	45	–	45	–	
SCLK rise → Input Data hold	t_{HSR}	0	–	0	–	0	–	


SCLK
(Output mode/
Rising edge input
mode)
SCLK
(Falling edge input
mode)
OUTPUT DATA
TxD
INPUT DATA
RxD
tSCY
tSCH
tSCL
tOSS
tOHS
tSRD
tHSR
0
1
2
3
VALID
VALID
VALID
VALID

32.6.3 Serial Bus Interface (I2C/SIO)

32.6.3.1 I2C Mode

In the table below, the letter x represents the I2C operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.It varies depending on the programming of the clock gear function.

n denotes the value of n programmed into the SCK (SCL output frequency select) field in the SBIxCR.

Parameter	Symbol	Equation		Standard Mode		Fast Mode		Unit
		Min.	Max	Min.	Max	Min.	Max	
SCL Clock frequency	t_{SCL}	0	−	0	100	0	400	kHz
Hold time for START condition	$t_{HD;\ STA}$	−	−	4.0	−	0.6	−	μs
SCL Low width (Input) (Note 1)	t_{LOW}	−	−	4.7	−	1.3	−	μs
SCL High width (Input) (Note 2)	t_{HIGH}	−	−	4.0	−	0.6	−	μs
Setup time for a repeated START condition	$t_{SU;\ STA}$	(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): $(2^{n-1} + 14)/x$

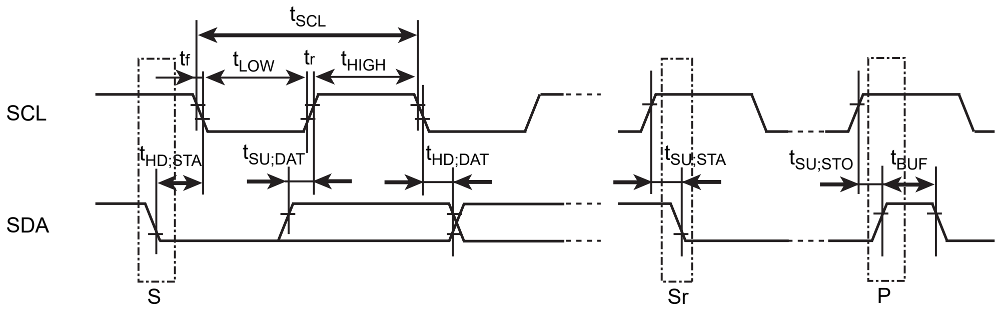
On I2C-bus specification, maximum Speed of Standard Mode/fast mode is 100kHz/400khz. Internal SCL Frequency setting should comply with fsys and Note1 & Note2 shown above.

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

32.6.3.2 Clock-Synchronous 8-bit SIO mode

In the table below, the letter x represents the I2C operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

(1) SCK input mode (SCK duty=50%)

[Data Input]

Parameter	Symbol	Equation		48 MHz		80 MHz		Unit
		Min.	Max	Min.	Max	Min.	Max	
SCK Clock High width (input)	t_{SCH}	4x	–	83.3	–	50	–	ns
SCK Clock Low width (input)	t_{SCL}	4x	–	83.3	–	50	–	
SCK cycle	t_{SCY}	t_{SCH} + t_{SCL}	–	166.6	–	100	–	
Valid Data input ← SCK rise or	t_{SRD}	30 − x	–	9.2	–	17.5	–	
SCK rise → Input Data hold	t_{HSR}	2x + 30	–	71.7	–	55.0	–	

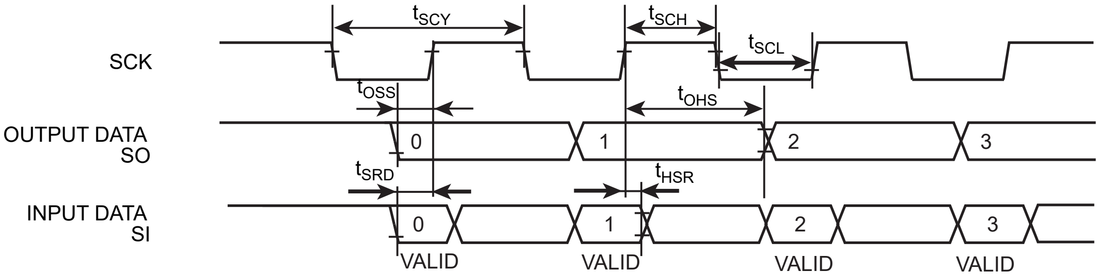
[Data Output]

Parameter	Symbol	Equation		48 MHz		80 MHz		Unit
		Min.	Max	Min.	Max	Min.	Max	
SCK Clock High width (input)	t_{SCH}	4x	–	107.5 (Note1)	–	82.5 (Note1)	–	ns
SCK Clock Low width (input)	t_{SCL}	4x	–	107.5 (Note1)	–	82.5 (Note1)	–	
SCK cycle	t_{SCY}	t_{SCH} + t_{SCL}	–	215	–	165	–	
Output Data ← SCK rise	t_{OSS}	$t_{SCY}/2 - 3x - 45$	–	0 (Note2)	–	0 (Note2)	–	
SCK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 + x$	–	128.3	–	95	–	

Note 1: The value indicates a minimum value that enables t_{OSS} to be zero or more.
Note 2: Keep this value positive by adjusting SCK cycle.

(2) SCK output mode (for an SCK signal with a 50% duty cycle)

Parameter	Symbol	Equation		48 MHz		80 MHz		Unit
		Min.	Max	Min.	Max	Min.	Max	
SCK cycle (programmable)	t_{SCY}	16x	–	333.3	–	200	–	ns
Output Data ← SCK rise	t_{OSS}	$t_{SCY}/2 - 20$	–	146.6	–	80	–	
SCK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 - 20$	–	146.6	–	80	–	
Valid Data input ← SCK rise	t_{SRD}	x + 45	–	65.8	–	57.5	–	
SCK rise → Input Data hold	t_{HSR}	0	–	0	–	0	–	


tSCY
tSCH
tSCL
SCK
tOSS
tOHS
OUTPUT DATA
SO
0
1
2
3
tSRD
tHSR
INPUT DATA
SI
0
1
2
3
VALID
VALID
VALID
VALID

32.6.4 Synchronous Serial Interface (SSP)

32.6.4.1 AC Measurement Condition

The letter "T" used in the equations in the table represents the period of internal bus frequency (fsys).

- Output levels : High = 0.7 × DVDD3A, Low = 0.3 × DVDD3A
- Input levels: High = 0.9 × DVDD3A, Low = 0.1 × DVDD3A
- Load capacitanceCL=30pF

Note: The "Equation" column in the table shows the specifications under the conditions DVDD3A = 2.7V to 3.6V.

10MHz type@ch0/1

Parameter	Symbol	Equation		fsys=48MHz	fsys=80MHz	Unit
		Min.	Max	(m=6,n=16)	(m=8, n=24)	
SPxCLK cycle (master)	T_m	(m)T At least 100ns or more	−	125 (8MHz)	100 (10MHz)	ns
SPxCLK cycle (slave)	T_s	(n)T At least 300ns or more	−	333 (3MHz)	300 (3.3MHz)	
SPxCLK rise up time	t_r	−	15	15	15	
SPxCLK fall down time	t_f	−	15	15	15	
Master mode: SPxCLK low level pulse width	t_{WLM}	(m)T/2 - 15	−	47.5	35	
Master mode: SPxCLK high level pulse width	t_{WHM}	(m)T/2 - 15	−	47.5	35	
Slave mode: SPxCLK low level pulse width	t_{WLS}	(n)T/2 - 15	−	151.7	135	
Slave mode: SPxCLK high level pulse width	t_{WHS}	(n)T/2 - 15	−	151.7	135	
Master mode: SPxCLK rise/fall to output data valid	t_{ODSM}	−	15	15	15	
Master mode: SPxCLK rise/fall to output data hold	t_{ODHM}	(m)T/2 - 15	−	47.5	35	
Master mode: Input data valid to SPxCLK rise/fall	t_{IDSM}	30	−	30	30	
Master mode: SPxCLK rise/fall to input data hold	t_{IDHM}	0	−	0	0	
Master mode: SPxFSS valid to SPxCLK rise/fall	t_{OFSM}	(m)T - 15	(m)T + 15	110 to140	85 to 115	
Slave mode: SPxCLK rise/fall to output data valid delay time	t_{ODSS}	−	(3T) + 40	102.5	77.5	
Slave mode: SPxCLK rise/fall (Output data hold)	t_{ODHS}	(n)T/2 + (2T)	−	208.3	175	
Slave mode: Input data valid to SPxCLK rise/fall	t_{IDSS}	10	−	10	10	
Slave mode: SPxCLK rise/fall to input data hold	t_{IDHS}	(3T) + 15	−	77.5	52.5	
Slave mode: SPxFSS valid to SPxCLK rise/fall	t_{OFSS}	(n)T + 10	−	343.3	310	

20MHz type@ch2

Parameter	Symbol	Equation		fsys=48MHz (m=4,n=12)	fsys=80MHz (m=4, n=12)	Unit
		Min.	Max			
SPxCLK cycle (master)	T_m	(m)T At least 50ns or more	–	83.3 (12MHz)	50 (20MHz)	ns
SPxCLK cycle (slave)	T_s	(n)T At least 150ns or more	–	250 (4MHz)	150 (6.6MHz)	
SPxCLK rise up time	t_r	–	10	10	10	
SPxCLK fall down time	t_f	–	10	10	10	
Master mode: SPxCLK low level pulse width	t_{WLM}	(m)T/2 - 10	–	31.6	15	
Master mode: SPxCLK high level pulse width	t_{WHM}	(m)T/2 - 10	–	31.6	15	
Slave mode: SPxCLK low level pulse width	t_{WLS}	(n)T/2 - 10	–	115	65	
Slave mode: SPxCLK high level pulse width	t_{WHS}	(n)T/2 - 10	–	115	65	
Master mode: SPxCLK rise/fall to output data valid	t_{ODSM}	–	10	10	10	
Master mode: SPxCLK rise/fall to output data hold	t_{ODHM}	(m)T/2 - 10	–	31.6	15	
Master mode: Input data valid to SPxCLK rise/fall	t_{IDSM}	15	–	15	15	
Master mode: SPxCLK rise/fall to input data hold	t_{IDHM}	0	–	0	0	
Master mode: SPxFSS valid to SPxCLK rise/fall	t_{OFSM}	(m)T - 15	(m)T + 15	68 to 98	35 to 65	
Slave mode: SPxCLK rise/fall to output data valid delay time	t_{ODSS}	–	(3T) + 30	92.5	67.5	
Slave mode: SPxCLK rise/fall (Output data hold)	t_{ODHS}	(n)T/2 + (2T)	–	166.6	100	
Slave mode: Input data valid to SPxCLK rise/fall	t_{IDSS}	10	–	10	10	
Slave mode: SPxCLK rise/fall to input data hold	t_{IDHS}	(3T) + 15	–	77.5	52.5	
Slave mode: SPxFSS valid to SPxCLK rise/fall	t_{OFSS}	(n)T + 10	–	260	160	

Note: Baud rate clock is set under below condition:

- Master mode:

  $m = (\text{<CPSDVSR>} \times (1 + \text{<SCR>})) = fsys/f_{SPxCLK}$

  <CPSDVR>is set only even number and "m" must set between the range of $65024 \geq m \geq 2$.

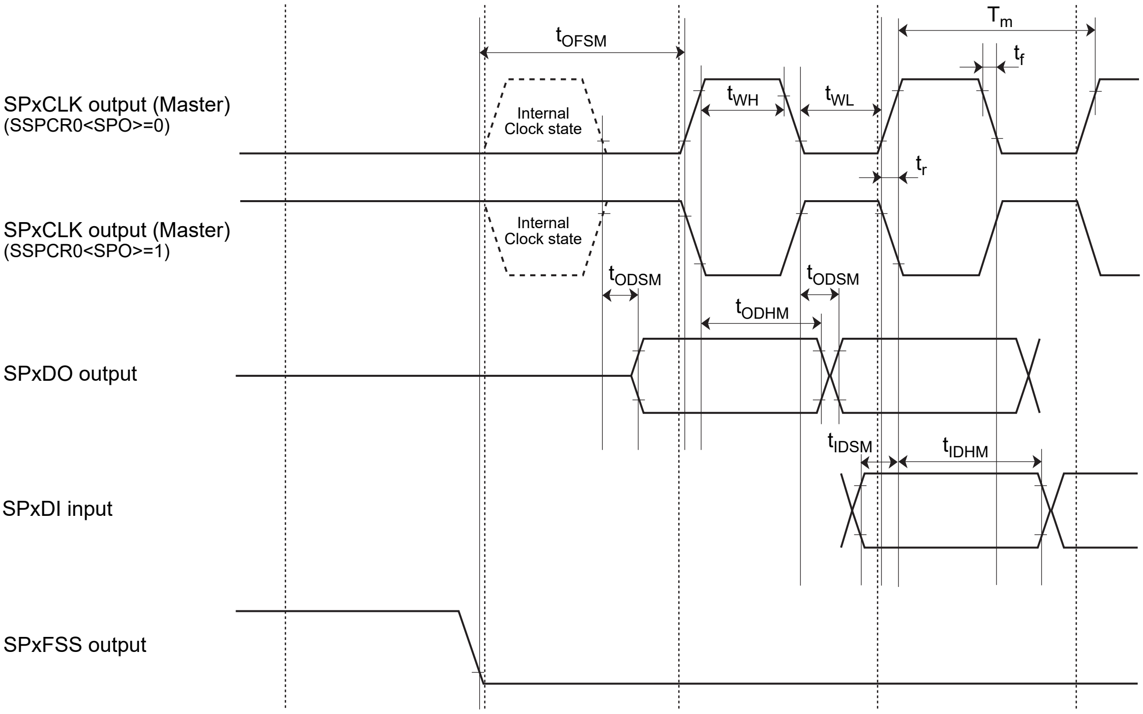
- Slave mode

  $n = fsys/f_{SPxCLK}$ $(65024 \geq n \geq 12)$

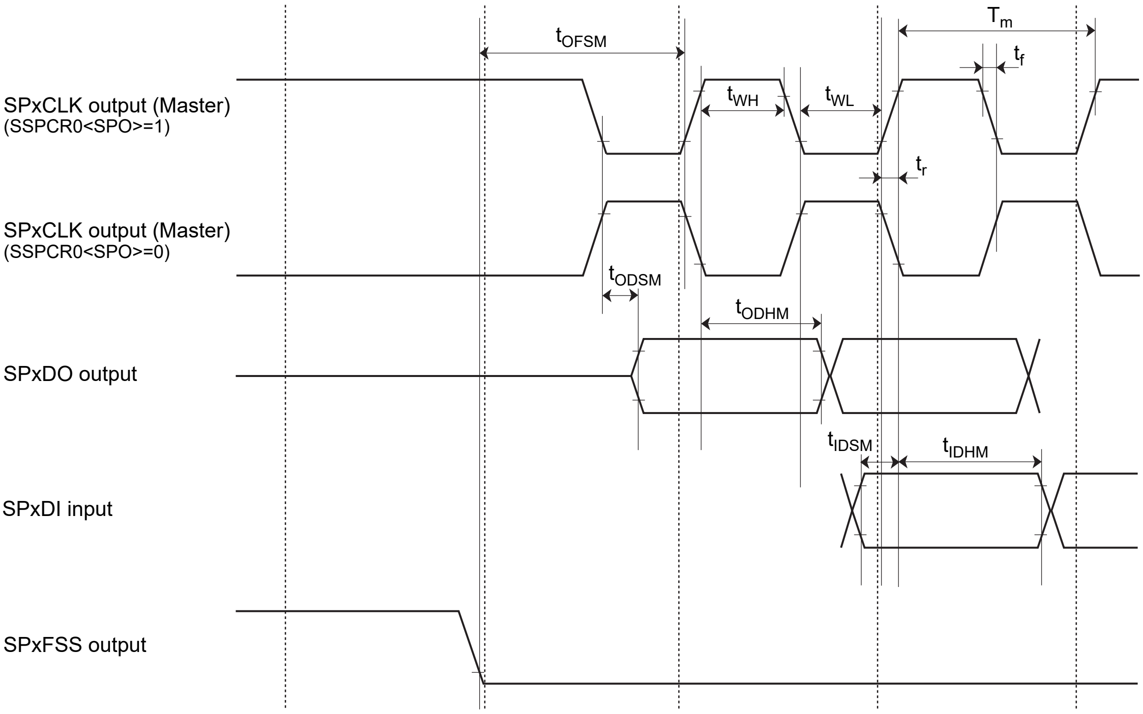
32.6.4.2 SSP SPI mode (Master)

- fsys ≥ 2 × f_{SPxCLK} (Maximum)
- fsys ≤ 65024 × f_{SPxCLK} (Minimum)

(1) Master SSPCR0<SPH>="0" (Data is latched on the first edge.)


SPxCLK output (Master)
(SSPCR0<SPO>=0)
SPxCLK output (Master)
(SSPCR0<SPO>=1)
SPxDO output
SPxDI input
SPxFSS output
Internal Clock state
Internal Clock state
t_{OFSM}
T_m
t_f
t_{WH}
t_{WL}
t_r
t_{ODSM}
t_{ODSM}
t_{ODHM}
t_{IDSM}
t_{IDHM}

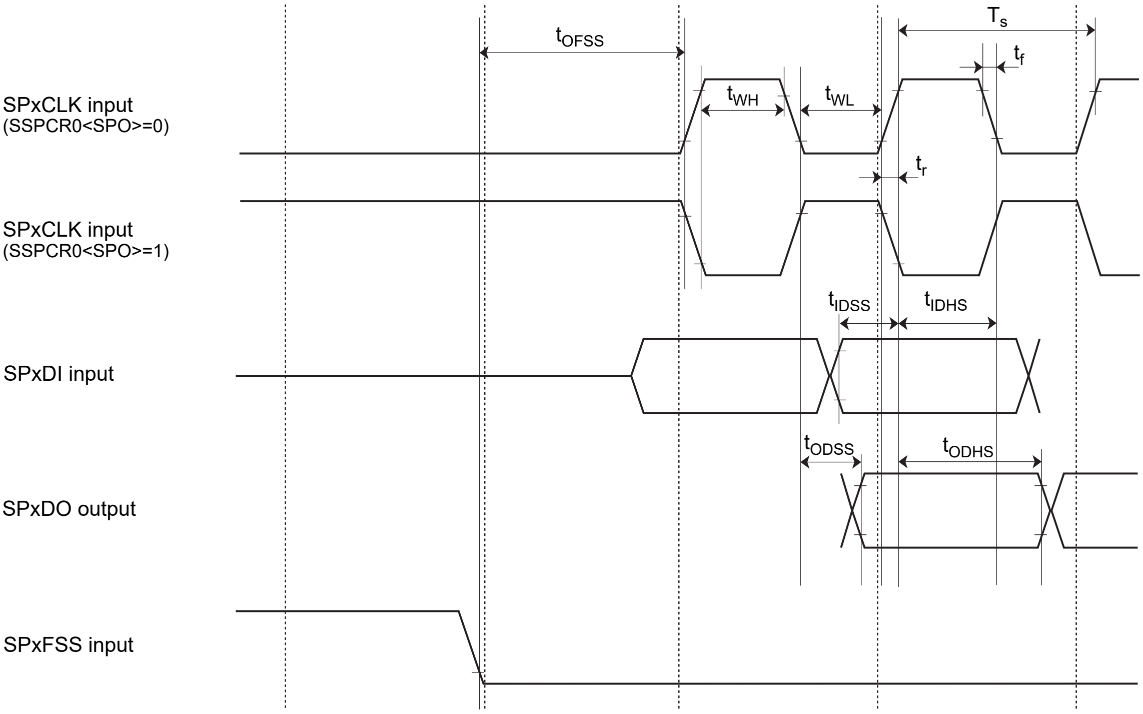
(2) Master SSPCR0<SPH>="1" (Data is latched on the second edge)


SPxCLK output (Master)
(SSPCR0<SPO>=1)
SPxCLK output (Master)
(SSPCR0<SPO>=0)
SPxDO output
SPxDI input
SPxFSS output
tOFSM
Tm
tf
tWH
tWL
tr
tODSM
tODHM
tIDSM
tIDHM

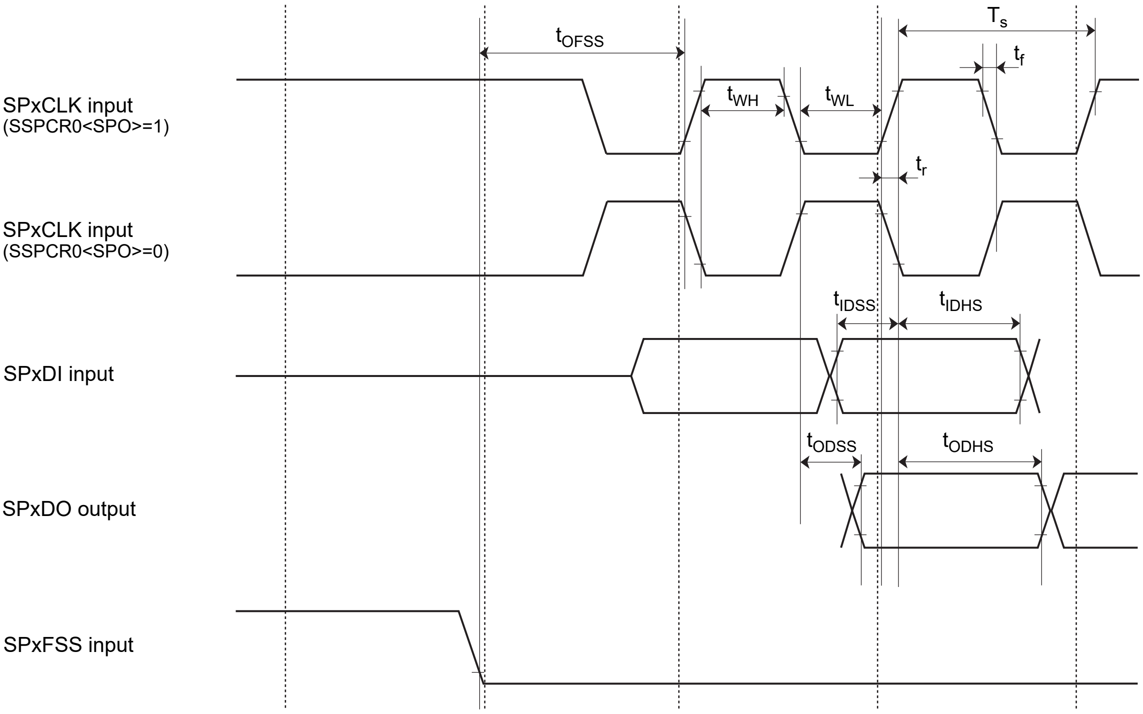
32.6.4.3 SSP SPI mode (Slave)

- fsys ≥ 12 × f_{SPxCLK} (Maximum)
- fsys ≤ 65024 × f_{SPxCLK} (Minimum)

(3) Slave SSPCR0<SPH>="0" (Data is latched on the first edge.)


SPxCLK input
(SSPCR0<SPO>=0)
SPxCLK input
(SSPCR0<SPO>=1)
SPxDI input
SPxDO output
SPxFSS input
tOFSS
Ts
tf
tWH
tWL
tr
tIDSS
tIDHS
tODSS
tODHS

(4) Slave SSPCR0<SPH>="1" (Data is latched on the second edge.)


T_s
t_OFSS
t_f
t_WH
t_WL
SPxCLK input
(SSPCR0<SPO>=1)
t_r
SPxCLK input
(SSPCR0<SPO>=0)
t_IDSS
t_IDHS
SPxDI input
t_ODSS
t_ODHS
SPxDO output
SPxFSS input

32.6.5 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		48 MHz		80 MHz		Unit
		Min.	Max	Min.	Max	Min.	Max	
Clock low pulse width	t_{VCKL}	2x + 100	–	141.7	–	125	–	ns
Clock high pulse width	t_{VCKH}	2x + 100	–	141.7	–	125	–	ns

32.6.6 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		48 MHz		80 MHz		Unit
		Min.	Max	Min.	Max	Min.	Max	
Low pulse width	t_{CPL}	2x + 100	–	141.7	–	125	–	ns
High pulse width	t_{CPH}	2x + 100	–	141.7	–	125	–	ns

32.6.7 External Interrupt

In the table below, the letter x represents the fsys cycle time.

1. Except STOP1and STOP2 release interrupts

Parameter	Symbol	Equation		48 MHz		80 MHz		Unit
		Min.	Max	Min.	Max	Min.	Max	
INT0 to D low-level pulse width	t_{INTAL}	x + 100	–	120.8	–	112.5	–	ns
INT0 to D high-level pulse width	t_{INTAH}	x + 100	–	120.8	–	112.5	–	ns

2. STOP1 release interrupts

Parameter	Symbol	Min.	Max	Unit
$\overline{\text{NMI}}$, INT0 to B,D low-level pulse width	t_{INTBL}	100	–	μs
INT0 to B,D high-level pulse width	t_{INTBH}	100	–	

3. STOP2 release interrupts

Parameter	Symbol	Min.	Max	Unit
$\overline{\text{NMI}}$, INT0 to B,D low-level pulse width	t_{INTCL}	1	–	ms
INT0 to B,D high-level pulse width	t_{INTCH}	1	–	

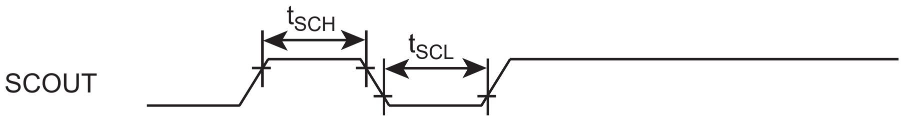
32.6.8 $\overline{\text{NMI}}$

Parameter	Symbol	Min.	Max	Unit
$\overline{\text{NIM}}$ Low-level pulse width	t_{INTCL}	100	−	ns

32.6.9 SCOUT

Parameter	Symbol	Equation		48 MHz		80 MHz		Unit
		Min.	Max	Min.	Max	Min.	Max	
High-level pulse width	t_{SCH}	0.5T − 5	−	5.4	−	1.25	−	ns
Low-level pulse width	t_{SCL}	0.5T − 5	−	5.4	−	1.25	−	ns

Note: In the above table, the letter T represents the cycle time of the SCOUT output clock.



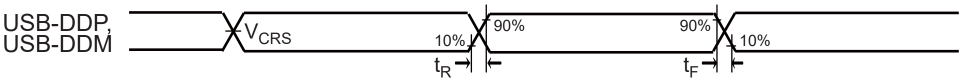
32.6.10 ADC and DAC trigger Input

Parameter	Symbol	Formula		48MHz		80MHz		Unit
		Min.	Max	Min.	Max	Min.	Max	
Low-level pulse width	T_{ADL}	2/fsys + 20	–	62	–	45	–	ns
High-level pulse interval	T_{ADH}	2/fsys + 20	–	62	–	45	–	

32.6.11 USB Timing (Full-speed)

DVDD3B = 3.0 V to 3.45 V@ fusbclk = 48MHz

Parameter	Symbol	Min.	Max	Unit
USB-DDP, USB-DDM Supply rise time	t_R	4	20	ns
USB-DDP, USB-DDM Supply withdraw time	t_F	4	20	
Data Line crossover voltage	V_{CRS}	1.3	2.0	V



32.6.12 External Bus Interface AC Characteristics

32.6.12.1 AC Measurement Condition

- DVDD3A=2.7 to 3.6V
- Output levels: High = 0.7 × DVDD3A,Low = 0.3 × DVDD3A
- Input levels: High = 0.7 × DVDD3A,Low = 0.3 × DVDD3A
- Load capacitance: CL = 30pF

32.6.12.2 Multiplex Bus mode

Conditional variable : ALE = 1, RWS = 1, TW = 2, RWH = 1, CSH =1 @tsys=tcyc=20.8ns

ALE = 1, RWS = 1, TW = 3, RWH = 1, CSH =1 @tsys=tcyc=12.5ns

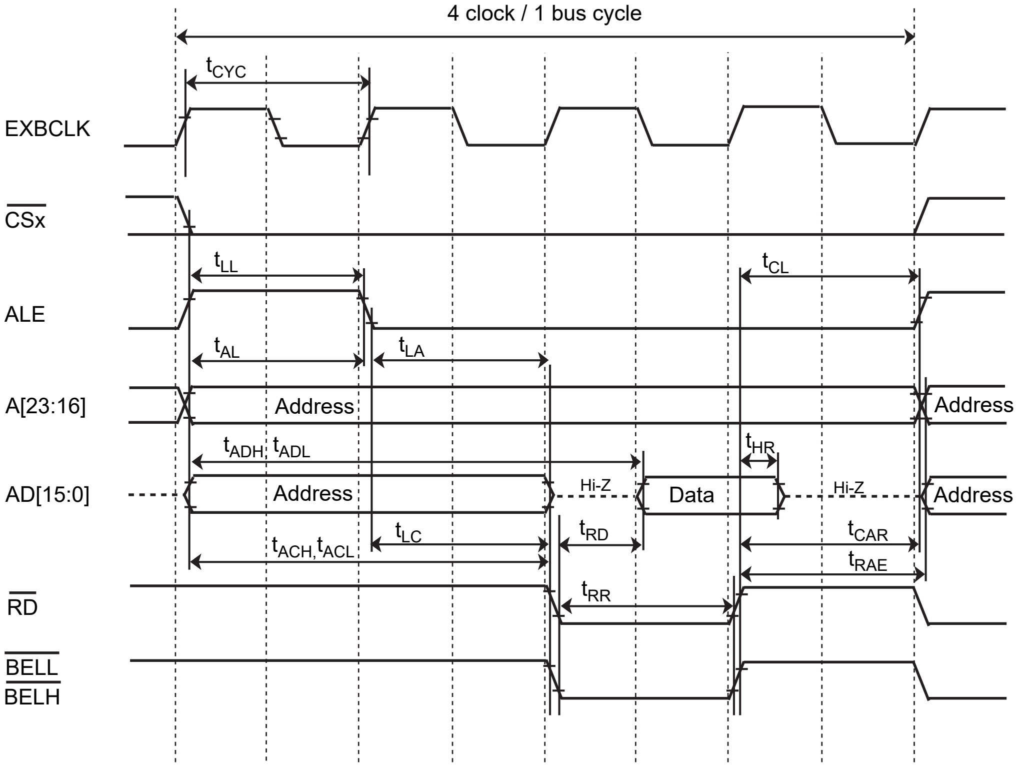
- ALE : Conditional variable (ALE = 1 + n; n = 0, 1, 2, 4)
- RWS: Number of setup cycle insertion before $\overline{\text{RD}}$, $\overline{\text{WR}}$ asserted (TW = 0, 1, 2 or 4)
- TW : Number of internal wait insertion (TW = 0 to 15)
- RWH: Number of $\overline{\text{RD}}$, $\overline{\text{WR}}$ hold cycle insertion (RWH = 0 to 6 or 8)
- CSH :Number of $\overline{\text{CSx}}$ hold cycle insertion (CSH = 0, 1, 2 or 4)

DVDD3A = 3.6V to 2.7V

Parameter	Sym-bol	Equation		48MHz		80MHz		Unit
		Min.	Max	Min.	Max	Min.	Max	
System clock period (x)	t_{SYS}	x	−	20.8	−	12.5	−	ns
External bus clock (EXBCLK)	t_{CYC}	x	−	20.8	−	12.5	−	
A[23:0] valid to ALE low	t_{AL}	x (1+ALE)-15	−	26.6	−	10	−	
ALE hold after A[23:0] hold	t_{LA}	x (1+RWS)-7	−	34.6	−	18	−	
ALE width high	t_{LL}	x (1+ALE)-15	−	26.6	−	10	−	
ALE low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ asserted	t_{LC}	x (1+RWS)-7	−	34.6	−	18	−	
$\overline{\text{RD}}$ or $\overline{\text{WR}}$ negated to ALE high	t_{CL}	x (1+RWH+CSH)-15	−	47.5	−	22.5	−	
$\overline{\text{RD}}$ or $\overline{\text{WR}}$ negated to A[23:16] hold	t_{CAR}	x (1+RWH+CSH)-15	−	47.5	−	22.5	−	
A[15:0] valid to D[15:0] input A[23:16] valid to D[15:0] input	t_{ADL} t_{ADH}	−	x(3+ALE+RWS+TW)-45	−	100.8	−	55	
A[15:0] valid to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ asserted A[23:16] valid to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ asserted	t_{ACL} t_{ACH}	x (2+ALE+RWS)-19	−	64.3	−	31	−	
$\overline{\text{RD}}$ asserted to D[15:0] data in	t_{RD}	−	x (1+TW)-35	−	27.5	−	15	
$\overline{\text{RD}}$ width low	t_{RR}	x (1+TW)-12	−	50.4	−	38	−	
$\overline{\text{RD}}$ negated to D[15:0] hold	t_{HR}	0	−	0	−	0	−	
$\overline{\text{RD}}$ negated to A[23:0] output	t_{RAE}	x (1+RWH+CSH)-15	−	47.4	−	22.5	−	
$\overline{\text{WR}}$ width low	t_{WW}	x (1+TW)-15	−	47.4	−	35	−	
D[15:0] valid to $\overline{\text{WR}}$ negated	t_{DW}	x (1+TW)-15	−	47.4	−	35	−	
$\overline{\text{WR}}$ negated to D[15:0] hold	t_{WD}	x (1+RWH)-10	−	31.6	−	15	−	

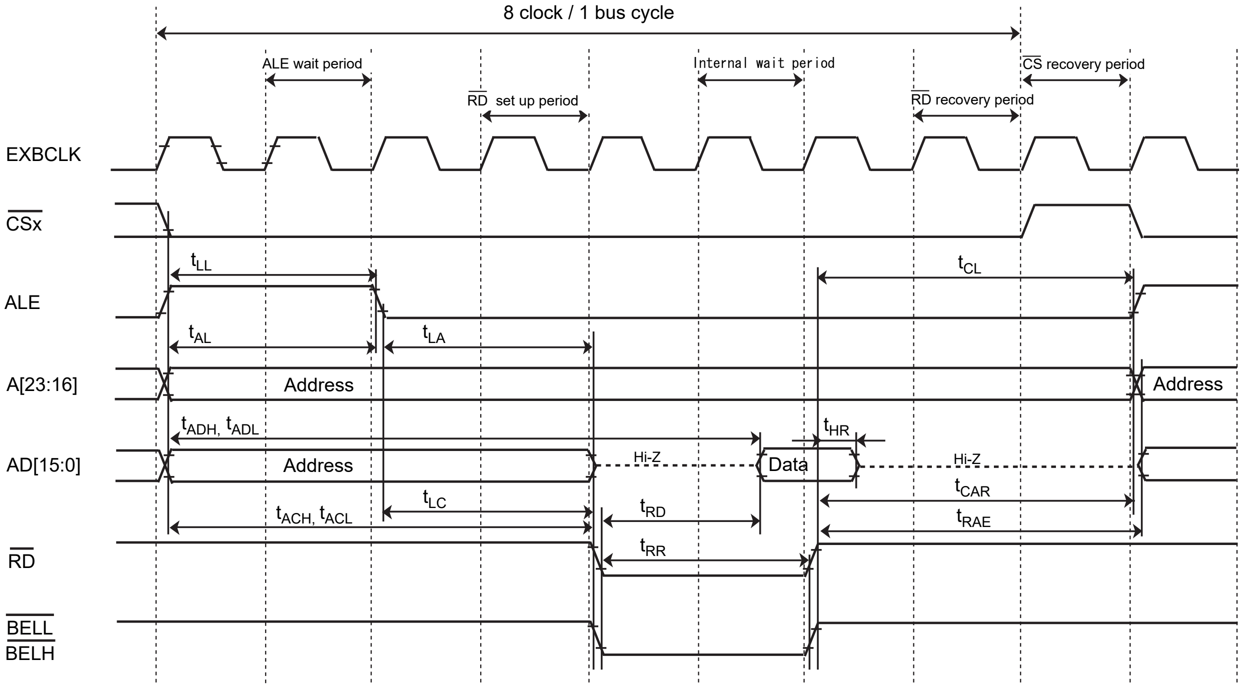
1. Read cycle timing (minimum cycle)

   (Neither Cycle expander, ALE wait, RD setup, Internal wait, CS recovery nor RD recovery function are used.)


4 clock / 1 bus cycle
tCYC
EXBCLK
CSx
tLL
tCL
ALE
tAL
tLA
A[23:16]
Address
Address
tADH, tADL
tHR
AD[15:0]
Address
Hi-Z
Data
Hi-Z
Address
tLC
tRD
tCAR
tACH,tACL
tRAE
tRR
RD
BELL
BELH

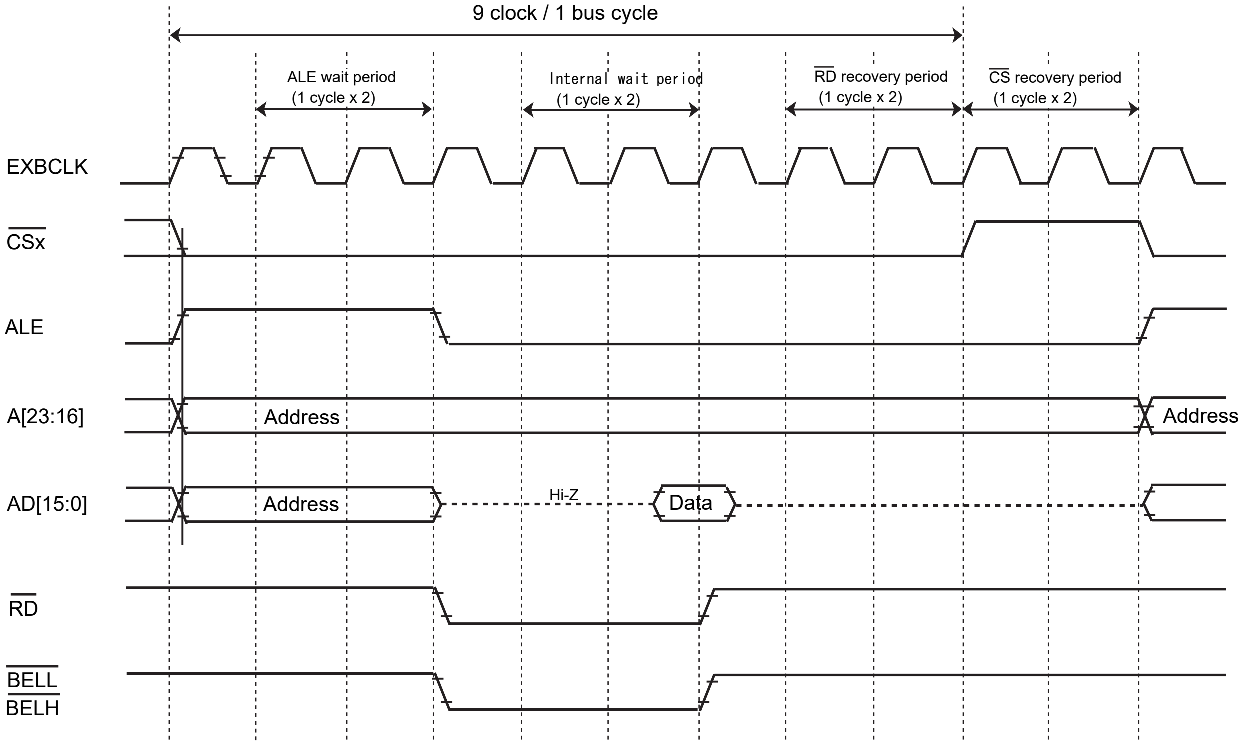
2. Read cycle timing (1 bus cycle per 8 clock)

(ALE wait, RD setup, Internal wait, CS recovery and RD recovery function are set to 1 cycle though Cycle expander function is not used.)


8 clock / 1 bus cycle
ALE wait period
$\overline{RD}$ set up period
Internal wait period
$\overline{CS}$ recovery period
$\overline{RD}$ recovery period
EXBCLK
$\overline{CSx}$
ALE
A[23:16]
AD[15:0]
$\overline{RD}$
$\overline{BELL}$
$\overline{BELH}$
t_{LL}
t_{CL}
t_{AL}
t_{LA}
Address
Address
t_{ADH}, t_{ADL}
t_{HR}
Address
Hi-Z
Data
Hi-Z
t_{CAR}
t_{LC}
t_{ACH}, t_{ACL}
t_{RD}
t_{RAE}
t_{RR}

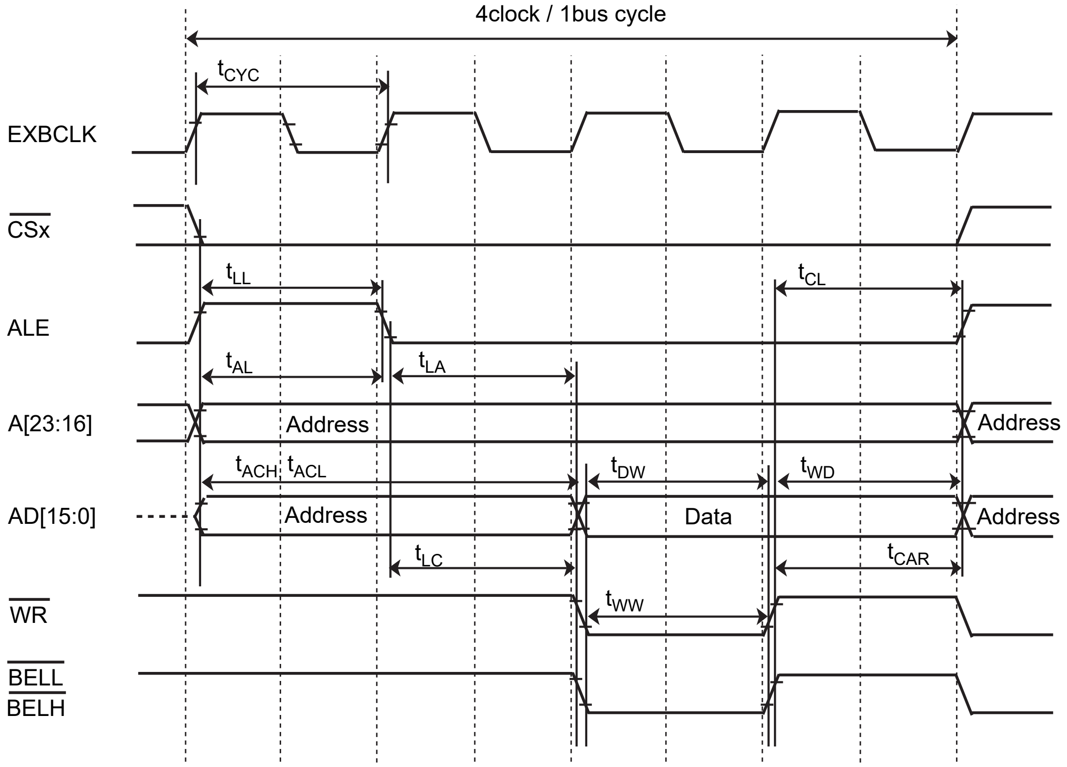
3. Read cycle timing (1 bus cycle per 9 clock)

   (ALE wait, RD setup, Internal wait, CS recovery and RD recovery function are set to 1 cycle though Cycle expander function is set double.)


9 clock / 1 bus cycle
ALE wait period
(1 cycle x 2)
Internal wait period
(1 cycle x 2)
RD recovery period
(1 cycle x 2)
CS recovery period
(1 cycle x 2)
EXBCLK
CSx
ALE
A[23:16]
Address
Address
AD[15:0]
Address
Hi-Z
Data
RD
BELL
BELH

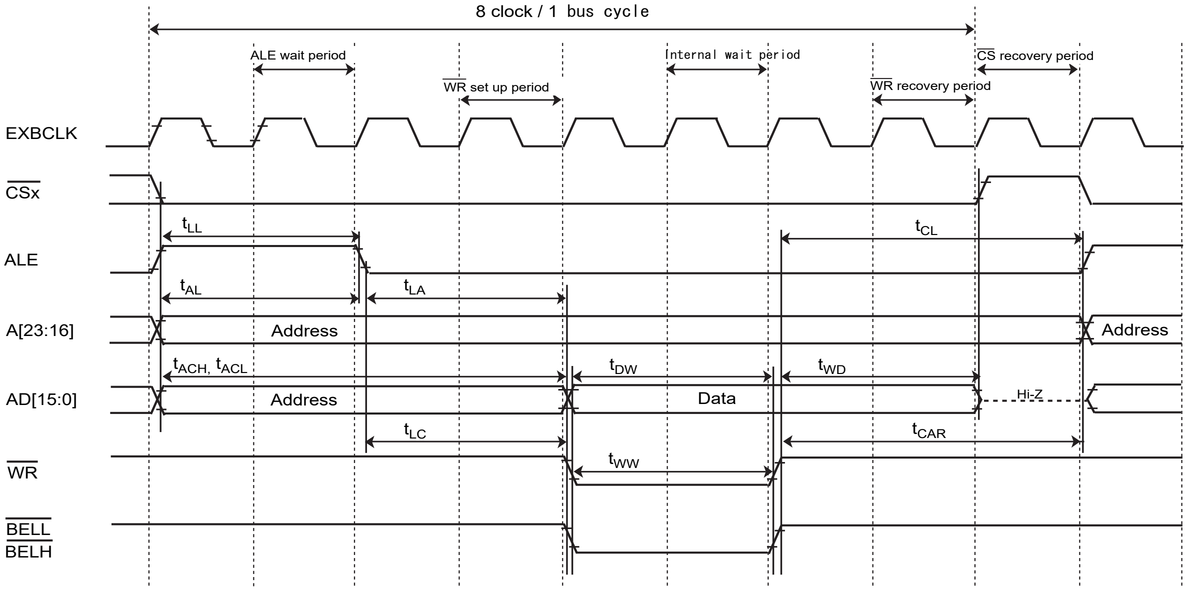
4. Write cycle timing (minimum bus cycle)

   (Neither Cycle expander, ALE wait, WR setup, Internal wait, CS recovery nor WR recovery function are used.)


4clock / 1bus cycle
t_{CYC}
EXBCLK
$\overline{CSx}$
t_{LL}
t_{CL}
ALE
t_{AL}
t_{LA}
A[23:16]
Address
Address
t_{ACH} t_{ACL}
t_{DW}
t_{WD}
AD[15:0]
Address
Data
Address
t_{LC}
t_{CAR}
$\overline{WR}$
t_{WW}
$\overline{BELL}$
$\overline{BELH}$

5. Write cycle timing (1 bus cycle per 8 clock)

(ALE wait, WR setup, Internal wait, CS recovery and WR recovery function are set to 1 cycle though Cycle expander function is not used.)


8 clock / 1 bus cycle
ALE wait period
WR set up period
Internal wait period
WR recovery period
CS recovery period
EXBCLK
CSx
ALE
A[23:16]
AD[15:0]
WR
BELL
BELH
t_{LL}
t_{CL}
t_{AL}
t_{LA}
Address
Address
t_{ACH}, t_{ACL}
t_{DW}
t_{WD}
Address
Data
Hi-Z
t_{LC}
t_{CAR}
t_{WW}

32.6.13 Debug Communication

32.6.13.1 AC Measurement Condition

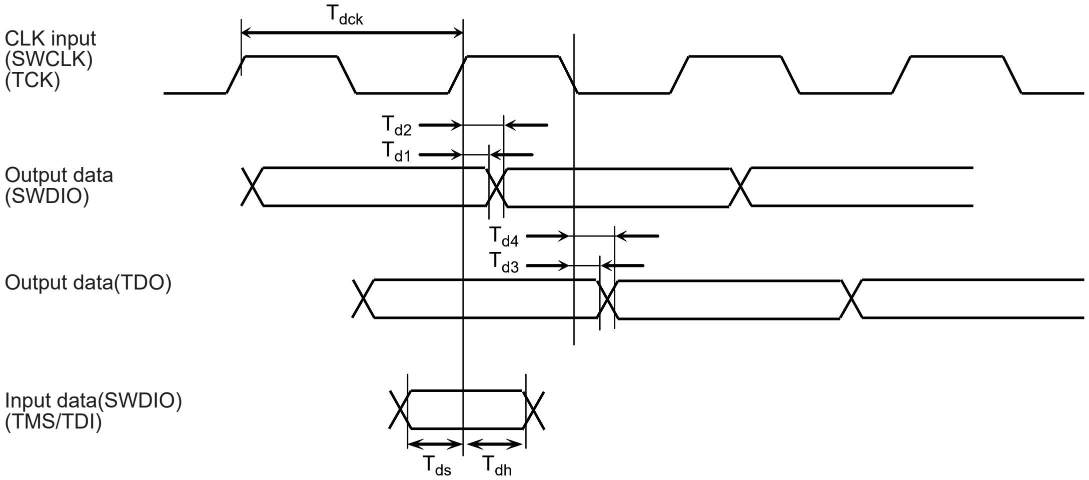
- Output levels: High = 0.8 × DVDD3A, Low = 0.2 × DVDD3A
- Input levels: Low = 0.8 × DVDD3A, Low = 0.2 × DVDD3A
- Load capacitance: CL = 30pF

32.6.13.2 SWD Interface

Parameter	Symbol	Min.	Max	Unit
CLK cycle	T_{dck}	100	–	ns
CLK rise → Output data hold	T_{d1}	4	–	
CLK rise → to output data valid	T_{d2}	–	30	
Input data valid → CLK rise	T_{ds}	20	–	
CLK rise → Input data hold	T_{dh}	15	–	

32.6.13.3 JTAG Interface

Parameter	Symbol	Min.	Max	Unit
CLK cycle	T_{dck}	100	–	ns
CLK fall→ Output data hold	T_{d3}	4	–	
CLK fall→ to output data valid	T_{d4}	–	50	
Input data valid → CLK rise	T_{ds}	20	–	
CLK rise → Input data hold	T_{dh}	15	–	


CLK input
(SWCLK)
(TCK)
T_{dck}
T_{d2}
T_{d1}
Output data
(SWDIO)
T_{d4}
T_{d3}
Output data(TDO)
Input data(SWDIO)
(TMS/TDI)
T_{ds}
T_{dh}

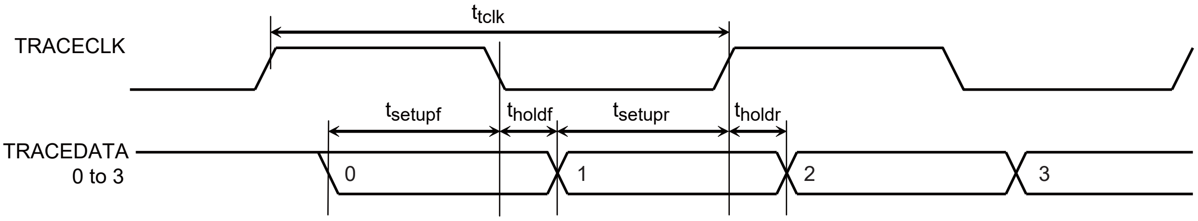
32.6.14 ETM Trace

32.6.14.1 AC Measurement Condition

- Output levels: High = 0.5 × DVDD3A, Low = 0.5 × AVDD3A
- Input levels: High = 0.8 × DVDD3A, Low = 0.2 × AVDD3A
- Load capacity: CL = 30pF

32.6.14.2 ETM Trace

Parameter	Symbol	Min.	Max	Unit
TRACECLK cycle	t_{tclk}	25	−	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	−	



32.6.15 On-chip Oscillator Characteristic

Parameter	Symbol	Condition	Min.	Typ.	Max	Unit
Oscillation frequency	IHOSC	Ta = -40 to 85°C	9.7	10.0	10.3	MHz

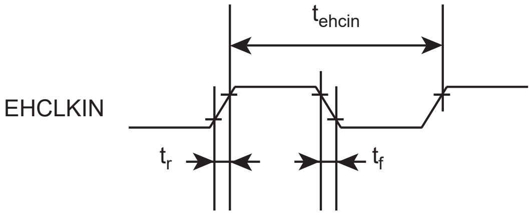
Note: Do not use an on-chip oscillator as a system clock (fsys) when high-accuracy oscillation frequency is required.

32.6.16 External Oscillator

Parameter	Symbol	Condition	Min.	Typ.	Max	Unit
High-frequency oscillation	EHOSC	Ta = -40 to 85°C	8	−	16	MHz

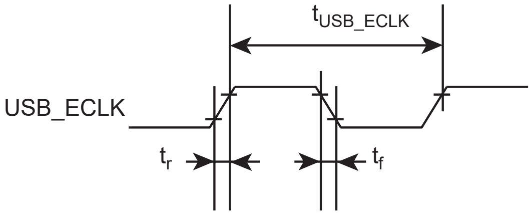
32.6.17 External Clock Input

Parameter	Symbol	Min.	Typ.	Max	Unit
External clock frequency	t_{ehcin}	8	−	16	MHz
External clock duty	−	−	50	−	%
External clock input rise time	t_r	−	−	10	ns
External clock input fall time	t_f	−	−	10	ns



32.6.18 USB External Clock Input

Parameter	Symbol	Min.	Typ.	Max	Unit
External clock frequency	t_{USB_ECLK}	8	−	48	MHz
External clock duty	−	−	50	−	%
External clock input rise time	t_r	−	−	4	ns
External clock input fall time	t_f	−	−	4	ns



32.6.19 Flash Characteristic

Parameter	Condition	Min.	Typ.	Max	Unit
Guaranteed number of Flash memory programming	Ta = 0 to 70°C	−	−	100	times

32.7 Recommended Oscillation Circuit

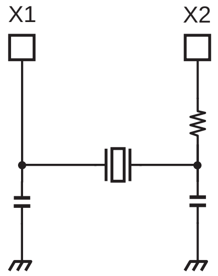


Figure 32-1 High-frequency oscillation connection

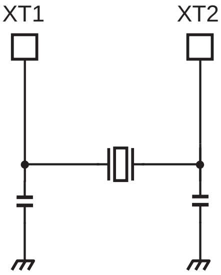


Figure 32-2 Low-frequency oscillation connection

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 TMPM367FDFG has been evaluated by the oscillator vender below. Please refer this information when selecting external parts

32.7.1 Ceramic Oscillator

The TMPM367FDFG recommends the high-frequency oscillator by Murata Manufacturing Co., Ltd.

Please refer to the Murata Website for details.

32.7.2 Crystal Oscillatior

The TMPM367FDFG recommends the low-frequency oscillator by KYOCERA Corporation.

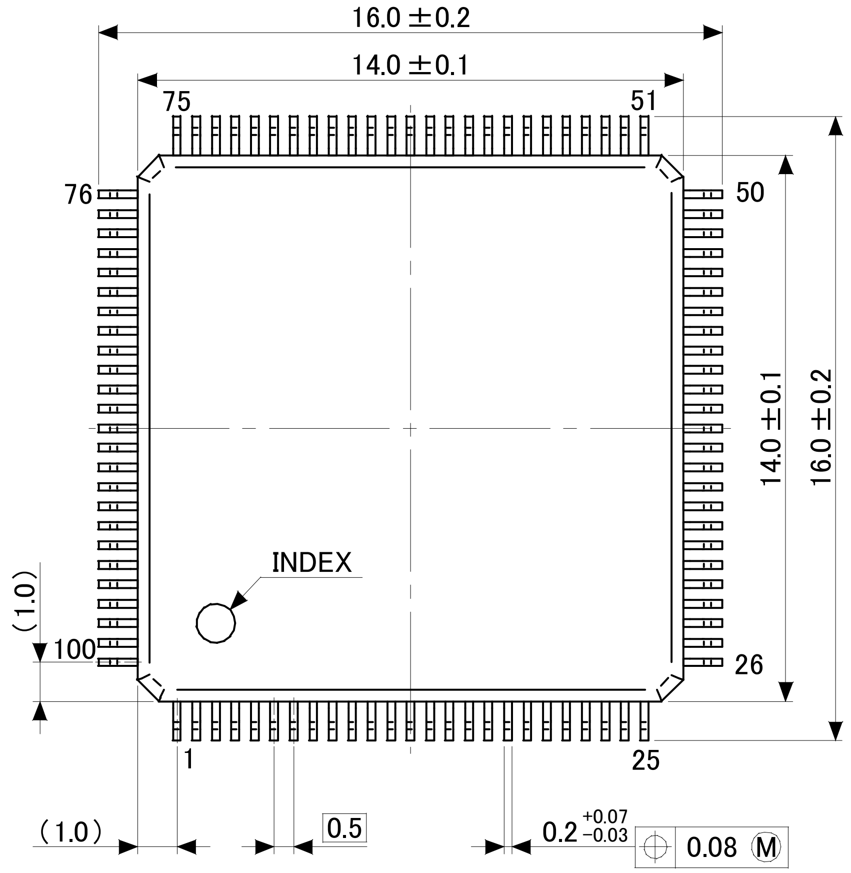
Please refer to the KYOCERA Website for details.

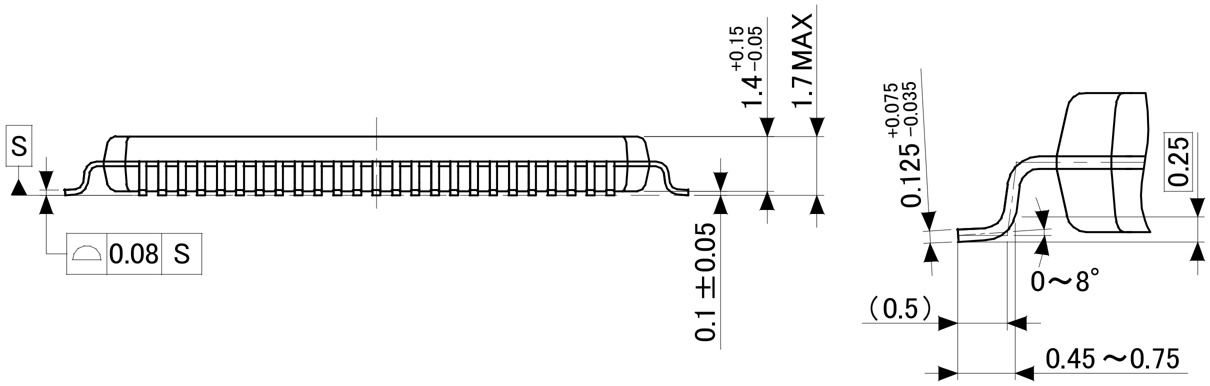
32.7.3 Precautions for designing printed circuit board

Be sure to design printed circuit board patterns that connect a crystal unit with other oscillation elements so that the length of such patterns become shortest possible to prevent deterioration of characteristics due to stray capacitances and wiring inductance. For multi-layer circuit boards, it is important not to wire the ground and other signal patterns right beneath the oscillation circuit.For more information, please refer to the URL of the oscillator vendor.

33. Package Dimensions

LQFP100-P-1414-0.50H "Unit : mm"


16.0 ±0.2
14.0 ±0.1
75
51
76
50
14.0 ±0.1
16.0 ±0.2
INDEX
(1.0)
100
26
1
25
(1.0)
0.5
0.2 +0.07 −0.03
⌖ 0.08 Ⓜ


1.4 +0.15 −0.05
1.7MAX
S
0.08 S
0.1 ±0.05
0.125 +0.075 −0.035
0.25
0~8°
(0.5)
0.45 ~0.75

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related 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, lifesaving and/or life supporting 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, and devices related to power plant. IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT. For details, please contact your TOSHIBA sales representative or contact us via our website.
- 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.**

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

https://toshiba.semicon-storage.com/