TOSHIBA

32 Bit RISC Microcontroller
TX00 Series

TMPM066/067/068FW

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

General precautions on the use of Toshiba MCUs

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

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

1. The MCUs' operation at power-on

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

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

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

2. Unused pins

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

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

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

3. Clock oscillation stability

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

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

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

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

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

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

Base Address = 0x0000_0000

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

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

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

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

1.2.2 SAMCR(Control register)

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

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

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

R / W	READ WRITE
R	READ
W	WRITE

c. Data description

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

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

d. Register descriptions

Registers are described as shown below.

- Register name <Bit Symbol>

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

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

- Register name [Bit]

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

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

Revision History

Date	Revision	Comment
2016/6/27	1	First Release
2018/9/21	2	Contents Revised

Table of Contents

General precautions on the use of Toshiba MCUs

TMPM066FWUG TMPM067FWQG TMPM068FWXBG

2. Product Information

3. Processor Core

4. Memory Map

5. Reset Operation

6. Clock/Mode control

7. Exceptions

8. µDMA Controller (µDMAC)

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

14. I2C Bus Interface

15. Toshiba Serial Peripheral Interface (TSPI)

16. USB Device Controller (USBD)

17. 10-bit Analog/Digital Converter (ADC)

24. Package Dimensions

CMOS 32-Bit Microcontroller

TMPM066FWUG TMPM067FWQG TMPM068FWXBG

The TMPM066/067/068FW is a 32-bit RISC microprocessor series with an Arm®Cortex®-M0 microprocessor core.

Features of the TMPM066/067/068FW are as follows.

1.1 Features

1. Arm®Cortex®-M0 microprocessor core
   a. Improved code efficiency has been realized through the use of Thumb®-2 instruction.
   b. Both high performance and low power consumption have been achieved.

      [High performance]

      - A 32-bit multiplication (32 x 32=32 bit) can be executed with one clock.

      [Low power consumptions]

      - 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. Endian: Little endian

3. On Chip program memory and data memory
   - On chip Flash ROM: 128 Kbyte
   - On chip RAM: 16 Kbyte

4. DMA controller (DMAC): 28 channels / 1 units
   - Transfer mode: Built-in memory, peripheral function and external memory.

5. Clock generator (CG)
   - External Clock input /external oscillation (8MHz to 16MHz)
   - on Chip PLL (12x ,Max)
   - Clock gear function: The high-speed clock can be divided into 1/1, 1/2, 1/4, 1/8, 1/16.

6. Standby mode
   - IDLE, STOP1

7. Interrupt source: The order of priority can be set to 4 levels.
   - Internal: 24factors
   - External: 6 factors

8. Input/output ports (PORT):

	TMPM066FWUG	TMPM067FWQG	TMPM068FWXBG
Input/Output pin	48	32	40
Input pin	-	-	-
Output pin	-	-	-

9. 16-bit timer (TMRB): 8 channels
   - 16-bit interval timer mode
   - 16-bit event counter mode
   - 16-bit PPG output (4 Phase Synchronous mode)
   - Input capture function

10. 16-bit timer (TMR16A): 2 channels

11. 16-bit timer with high resolution PPG output (TMRD):1unit
    - 4-channel synchronous PPG output
    - Duty can be settable in the units of 96MHz (10.4ns).

12. Watchdog timer (WDT): 1 channel

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

13. General-purpose serial interface (SIO/UART): 2 channels
    - Either UART mode or synchronous mode can be selected
    - Transmit FIFO: 4-stage 8-bit width, receive FIFO: 4-stage 8-bit width

14. I2C bus interface (I2C): 2 channels
    - Communication rate 100kbps / 400kbps / 1000kbps(Ch1 only)
    - Wake-up function by Slave Address match (Ch1 only)

15. Serial Peripheral interface (TSPI) : 1 channel
    - Supports 2 types of communications such as SPI mode and SIO mode
    - Data length is settable 8 through 32 bits in the units of 1 bit.

16. USB 2.0 device : 1 channel
    - Compliance Universal Serial Bus Specification Rev.2.0.
    - Supports 5 end points
    - Supports 4 transfer modes (Control/Interrupt /Bulk/ Isochronous).
    - Supports full communication speed (12Mbps) (dose not support Low Speed).

17. 10-bit AD converter (ADC):1 unit (8 channels)
    - Fixed channel/scan mode
    - Single/repeat mode
    - support Repeat conversion
    - AD monitoring

18. LVD/POR function: 1 unit

19. Maximum operating frequency: 24MHz

20. Debug Interface
    - SWD is supported.

21. Operating voltage range: 1.8 to 3.6V

22. Temperature range
    - −40°C to 85°C (except during Flash W/E)
    - 0°C to 70°C (during Flash W/E)

23. Package

Product Name	Package Type
TMPM066FWUG	LQFP64 (10mm × 10mm, 0.5mm pitch)
TMPM067FWQG	QFN48 (7mm × 7mm, 0.5mm pitch)
TMPM068FQXBG	VFBGA56 (5mm × 5mm, 0.5mm pitch)

1.2 Block Diagram

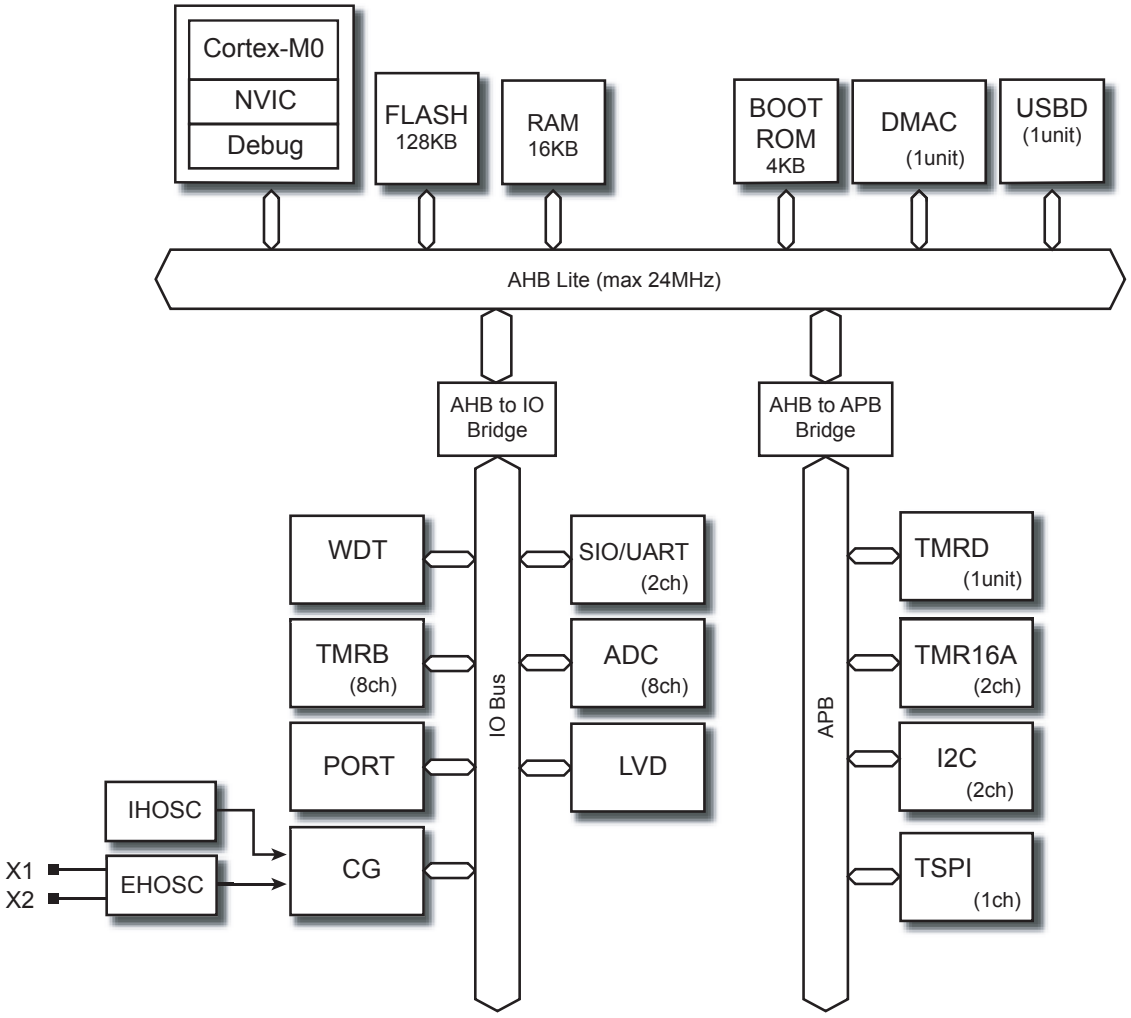


Figure 1-1 Block Diagram

1.3 Pin Layout (Top view)

Figure 1-2,Figure 1-3 and Figure 1-4 shows the pin layout of TMP066FWUG,TMPM067FWQG and TMPM068FWXBG as following.

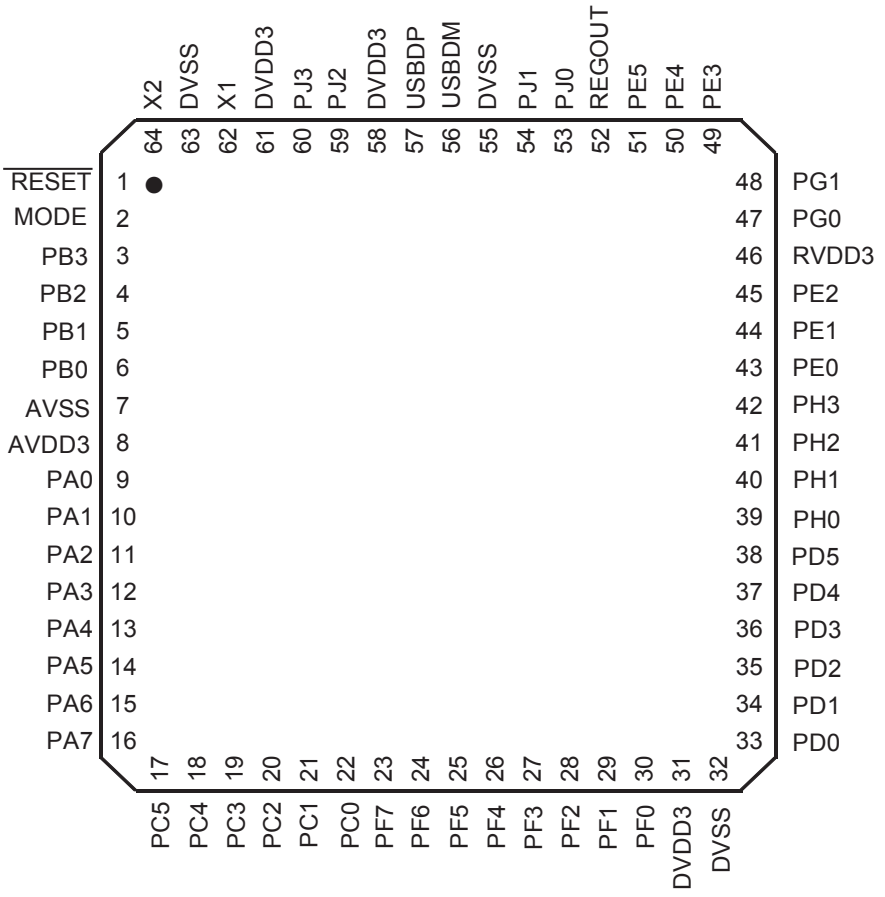


Figure 1-2 Pin Layout (LQFP64 TOP VIEW)

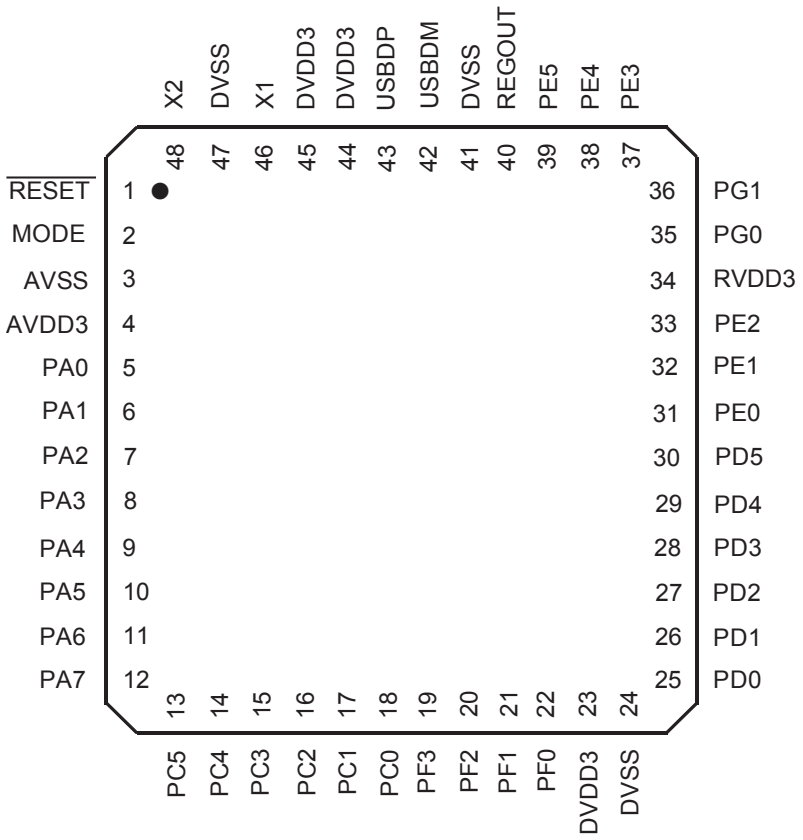


Figure 1-3 QFN48 TOP VIEW

	1	2	3	4	5	6	7	8	9
A	DVSS	X2	X1	DVDD3	USBDP	USBDM	DVSS	REGOUT	PE3
B	RESET_x	DVSS	MODE	DVDD3	PJ1	PJ0	PE5	PE4	PG1
C	PB0	PB1	AVSS					RVDD3	PG0
D	PA0	AVDD3						PE1	PE2
E	PA2	PA1						PE0	PH1
F	PA3	PA4						PD5	PH0
G	PA5	PA6						PD4	PD3
H	PA7	PC4	PC2	PF5	PF4	PF2	DVSS	PD2	PD1
J	PC5	PC3	PC1	PC0	PF3	PF1	PF0	DVDD3	PD0

Figure 1-4 BGA56 TOP VIEW

1.4 Pin names and Functions

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

1.4.1.1 Peripheral functions

Table 1-1 The number of pins and Pin names

Peripheral function	Pin name	Input or Output	Function
External interrupt	INTx	Input	External interrupt input pin x External interrupt input pin x has a noise filter (Filter width 30ns typ.)
16bit timer (TMRB)	TBxIN	Input	Input capture input pin
	TBxOUT	Output	output pin
16 bit timer (TMR16A)	T16AxOUT	Output	output pin
16bit timer (TMRD)	TDAxOUTx	Output	Output
Serial channel (SIO/UART)	SCxTXD	Output	Data output pin
	SCxRXD	Input	Data input pin
	SCxSCLK	I/O	Clock input/ output pin
Serial channel (TSPI)	TSPICS	I/O	CS pin (CSO: output, CSIN: input)
	TSPITXD	Output	Data output pin
	TSPIRXD	Input	Data input pin
	TSPICLK	I/O	Clock input/ output pin
I^2C bus interface (I^2C)	I2CxSDA	I/O	Data input / output pin
	I2CxSCL	I/O	Clock input/ output pin
Analog digital converter	AINx	Input	Analog input pin
USB interface (USBD)	USBDP	I/O	Data input/ output pin
	USBDM	I/O	Data input/ output pin

1.4.1.2 Debug function

Table 1-2 Pin name and functions

Pin name	Input or Output	Function
SWDIO	I/O	Serial wire data input and output pin
SWCLK	Input	Serial wire clock input pin

1.4.1.3 Control function

Table 1-3 Pin name and functions

Pin name	Input or Output	Function
X1	Input	Connected to a high-speed oscillator.
X2	Output	Connected to a high-speed oscillator.
MODE	Input	MODE pin MODE pin must be connected to GND.
$\overline{\text{RESET}}$	Input	Reset input pin
$\overline{\text{BOOT}}$	Input	BOOT mode control pin BOOT mode control pin is sampled at the rising edge of reset signal input pin. TMPM066/067/068FW changes Single Boot Mode when BOOT mode control pin is "Low". TMPM066/067/068FW changes Single Chip Mode when BOOT mode control pin is "High". Refer to "Flash memory" section for a detail.

1.4.1.4 Power supply pins

Table 1-4 Pin name and functions

Power supply pin name	Function
REGOUT	Pin connected with the capacitor(1μF) for the regulator.
RVDD3	Power supply pin for the regulator.
DVDD3	Power supply pin for the digital circuit.
DVSS	GND pin for the digital circuit.
AVDD3	Power supply pin for the analog circuit.
AVSS	GND pin for the analog circuit.

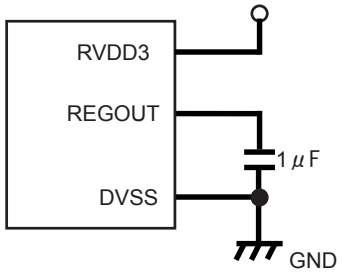


Figure 1-5 Capacitor connection of Regulator

(Must be placed on the shortest distance.)

1.4.2 Pin names and Function of TMPM066/067/068FW

1.4.2.1 The detail for pin names and function list

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

1. Function A

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

2. Function B

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

3. Pin specification

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

   - SMT/CMOS: Type of input gate

     SMT: Schmitt input

     CMOS: CMOS input

   - OD: Programmable open drain output support

     Yes: supported

     N/A: not supported

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

     PU: Programmable Pull-Up supported

     PD: Programmable Pull-Down supported

   - 10mA: 10mA current drive port

     Yes: supported

     N/A: not supported

1.4.2.2 PORT / Debug pin

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

PinNo.			PORT	Function A	Function B					Port Specification			
QFP 64	QFN 48	BGA 57			1	2	3	4	5	PU/PD	OD	SMT/ CMOS	10mA
PORTA													
9	5	D1	PA0	AIN0						PU/PD	Yes	SMT	N/A
10	6	E2	PA1	AIN1						PU/PD	Yes	SMT	N/A
11	7	E1	PA2	AIN2						PU/PD	Yes	SMT	N/A
12	8	F1	PA3	AIN3						PU/PD	Yes	SMT	N/A
13	9	F2	PA4	AIN4						PU/PD	Yes	SMT	N/A
14	10	G1	PA5	AIN5						PU/PD	Yes	SMT	N/A
15	11	G2	PA6	AIN6						PU/PD	Yes	SMT	N/A
16	12	H1	PA7	AIN7	TB1IN					PU/PD	Yes	SMT	N/A

PinNo.			PORT	Function A	Function B					Port Specification			
QFP 64	QFN 48	BGA 57			1	2	3	4	5	PU/PD	OD	SMT/ CMOS	10mA
PORTB													
6	-	C1	PB0							PU/PD	Yes	SMT	N/A
5	-	C2	PB1							PU/PD	Yes	SMT	N/A
4	-	-	PB2							PU/PD	Yes	SMT	N/A
3	-	-	PB3							PU/PD	Yes	SMT	N/A

PinNo.			PORT	Function A	Function B					Port Specification			
QFP 64	QFN 48	BGA 57			1	2	3	4	5	PU/PD	OD	SMT/ CMOS	10mA
PORTC													
22	18	J4	PC0		I2C0 SCL					PU/PD	Yes	SMT	N/A
21	17	J3	PC1		I2C0 SDA					PU/PD	Yes	SMT	N/A
20	16	H3	PC2		SC0TXD					PU/PD	Yes	SMT	Yes
19	15	J2	PC3		SC0RXD					PU/PD	Yes	SMT	Yes
18	14	H2	PC4		SC0SCK					PU/PD	Yes	SMT	N/A
17	13	J1	PC5	VBUS/ INT4	TB0IN					PU/PD	Yes	SMT	N/A

Note: PC5 : 5V input tolerant

PinNo.			PORT	Function A	Function B					Port Specification			
QFP 64	QFN 48	BGA 57			1	2	3	4	5	PU/PD	OD	SMT/ CMOS	10mA
PORTD													
33	25	J9	PD0		T16A0 OUT	TDA0 OUT0				PU/PD	Yes	SMT	N/A
34	26	H9	PD1		TB0 OUT	TDA0 OUT1				PU/PD	Yes	SMT	N/A
35	27	H8	PD2		TB1 OUT	TDA1 OUT0				PU/PD	Yes	SMT	N/A
36	28	G9	PD3		TB2 OUT	TDA1 OUT1				PU/PD	Yes	SMT	N/A
37	29	G8	PD4		TB3IN					PU/PD	Yes	SMT	N/A
38	30	F8	PD5	INT0						PU/PD	Yes	SMT	N/A

PinNo.			PORT	Function A	Function B					Port Specification			
QFP 64	QFN 48	BGA 57			1	2	3	4	5	PU/PD	OD	SMT/ CMOS	10mA
PORTE													
43	31	E8	PE0		SC1SCK					PU/PD	Yes	SMT	N/A
44	32	D8	PE1		SC1RXD	TB2IN				PU/PD	Yes	SMT	N/A
45	33	D9	PE2		SC1TXD					PU/PD	Yes	SMT	N/A
49	37	A9	PE3	$\overline{\text{BOOT}}$						PU/PD	Yes	SMT	N/A
50	38	B8	PE4		SWCLK					PU/PD	Yes	SMT	N/A
51	39	B7	PE5		SWDIO					PU/PD	Yes	SMT	N/A

PinNo.			PORT	Function A	Function B					Port Specification			
QFP 64	QFN 48	BGA 57			1	2	3	4	5	PU/PD	OD	SMT/ CMOS	10mA
PORTF													
30	22	J7	PF0	INT5	TSPI CSO	TSPI CSIN				PU/PD	Yes	SMT	Yes
29	21	J6	PF1	INT2	TSPI TXD					PU/PD	Yes	SMT	Yes
28	20	H6	PF2		TSPI RXD					PU/PD	Yes	SMT	Yes
27	19	J5	PF3		TSPI CLK					PU/PD	Yes	SMT	Yes
26	-	H5	PF4		TB3OUT					PU/PD	Yes	SMT	N/A
25	-	H4	PF5		TB4OUT					PU/PD	Yes	SMT	N/A
24	-	-	PF6							PU/PD	Yes	SMT	N/A
23	-	-	PF7							PU/PD	Yes	SMT	N/A

PinNo.			PORT	Function A	Function B					Port Specification			
QFP 64	QFN 48	BGA 57			1	2	3	4	5	PU/PD	OD	SMT/ CMOS	10mA
PORTG													
47	35	C9	PG0		I2C1 SCL					PU/PD	Yes	SMT	N/A
48	36	B9	PG1		I2C1 SDA					PU/PD	Yes	SMT	N/A

PinNo.			PORT	Function A	Function B					Port Specification			
QFP 64	QFN 48	BGA 57			1	2	3	4	5	PU/PD	OD	SMT/ CMOS	10mA
PORTH													
39	-	F9	PH0		TB4IN					PU/PD	Yes	SMT	N/A
40	-	E9	PH1		TB5IN					PU/PD	Yes	SMT	N/A
41	-	-	PH2		TB6IN					PU/PD	Yes	SMT	N/A
42	-	-	PH3		TB7IN					PU/PD	Yes	SMT	N/A

PinNo.			PORT	Function A	Function B					Port Specification			
QFP 64	QFN 48	BGA 57			1	2	3	4	5	PU/PD	OD	SMT/ CMOS	10mA
PORTJ													
53	-	B6	PJ0		TB5OUT					PU/PD	Yes	SMT	N/A
54	-	B5	PJ1		TB6OUT					PU/PD	Yes	SMT	N/A
59	-	-	PJ2							PU/PD	Yes	SMT	N/A
60	-	-	PJ3							PU/PD	Yes	SMT	N/A

1.4.2.3 USB & Control pin

Table 1-6 The number of pin and pin names

Pin No.			Control function Pin name
QFP64	QFN48	XBG57	
62	46	A3	X1
64	48	A2	X2
1	1	B1	$\overline{\text{RESET}}$
2	2	B3	MODE
49	37	A9	$\overline{\text{BOOT}}$
57	43	A5	USBDP
56	42	A6	USBDM

1.4.2.4 Power Supply pin

Table 1-7 The number of pin and pin names

Pin No.			Power supply Pin name
QFP64	QFN48	XBG57	
52	40	A8	REGOUT
46	34	C8	RVDD3
31,58,61	23,44,45	A4,B4,J8	DVDD3
32,55,63	24,41,47	A1,A7,B2,H7	DVSS
8	4	D2	AVDD3
7	3	C3	AVSS

2. Product Information

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

- "2.1.1 DMA Controller (DMAC)"
- "2.1.2 16-bit Timer/Event Counter (TMRB)"
- "2.1.3 16-bit Timer A (TMR16A)"
- "2.1.4 High Resolution 16-bit Timer (TMRD)"
- "2.1.5 Serial Channel (SIO/UART)"
- "2.1.6 I2C Bus (I2C)"
- "2.1.7 Toshiba Serial Peripheral Interface (TSPI)"
- "2.1.8 Analog/Digital Converter (ADC)"
- "2.1.9 USB Device (USBD)"
- "2.1.10 Debug Interface"

2.1 Information of Each Peripheral Function

2.1.1 DMA Controller (DMAC)

TMPM066/067/068FW incorporates 1 units of built-in DMA controller.

2.1.1.1 DMA Request table

DMA Request table are shows as follows.

Table 2-1 DMA Request table

Corresponding peripheral					
DMA ch No.	Burst	Single	DMA ch No.	Burst	Single
0	SIO/UART0 reception	−	16	TMRB ch6 Comparator match	−
1	SIO/UART0 transmission	−	17	TMRB ch7 Comparator match	−
2	SIO/UART1 reception	−	18	TMRB ch0 Capture input0	−
3	SIO/UART1 transmission	−	19	TMRB ch1 Capture input0	−
4	I2C ch0 reception	−	20	TMRB ch2 Capture input0	−
5	I2C ch0 transmission	−	21	TMRB ch3 Capture input0	−
6	I2C ch1 reception	−	22	TMRB ch4 Capture input0	−
7	I2C ch1 transmission	−	23	TMRB ch5 Capture input0	−
8	TSPI reception	−	24	TMRB ch6 Capture input0	−
9	TSPI transmission	−	25	TMRB ch7 Capture input0	−
10	TMRB ch0 Comparator match	−	26	−	−
11	TMRB ch1 Comparator match	−	27	−	−
12	TMRB ch2 Comparator match	−	28	−	−
13	TMRB ch3 Comparator match	−	29	−	−
14	TMRB ch4 Comparator match	−	30	Highest priority AD conversion completion	−
15	TMRB ch5 Comparator match	−	31	Normal AD conversion completion	−

2.1.2 16-bit Timer/Event Counter (TMRB)

TMPM066/067/068FW incorporates 8 channels of TMRB.

Table 2-2 Pin specifications

Channel	TBxOUT	TBxIN
TMRB0	PD1	PC5
TMRB1	PD2	PA7
TMRB2	PD3	PE1
TMRB3	PF4	PD4
TMRB4	PF5	PH0
TMRB5	PJ0	PH1
TMRB6	PJ1	PH2
TMRB7	-	PH3

Table 2-3 Synchronous start specifications

Master channel	Slave channel
TMRB0	TMRB1, TMRB2, TMRB3
TMRB4	TMRB5, TMRB6, TMRB7

Table 2-4 Capture trigger specifications

Trigger input channel	Trigger output
TMRB0 TMRB1 TMRB2	TB6OUT
TMRB3 TMRB4 TMRB5	TB7OUT

2.1.3 16-bit Timer A (TMR16A)

TMPM066/067/068FW incorporates 2 channels of TMR16A.

Table 2-5 Pin specifications

Channel	T16A0OUT
T16A0	PD0

2.1.4 High Resolution 16-bit Timer (TMRD)

TMPM066/067/068FW has one block of high resolution 16-bit timer.

Table 2-6 Pin specifications

channel	TMRD
TDAOUT0	PD0
TDAOUT1	PD1
TDAOUT2	PD2
TDAOUT3	PD3

2.1.4.1 Clock setting for TMRD

This device can be used the TMRD, when the PLL is enable and also the CGFSYSENA<IPENA18>, CGEXTENDO0<DCLKEN> are enabled.

Table 2-7 Support Clock of TMRD

fosc (MHz)	Number of PLL multiplication value	TMRDCLK (MHz)
8	12	96 (Max)
10	8	80
12	8	96 (Max)
16	6	96 (Max)
16 >	not used	not used

2.1.5 Serial Channel (SIO/UART)

TMPM066/067/068FW incorporates 2 channels of SIO.

Table 2-8 Pin specifications

Channel	SCxTXD	SCxRXD	SCxSCLK
SC0	PC2	PC3	PC4
SC1	PE2	PE1	PE0

Table 2-9 Transfer clock specifications

Clock input channel	Clock output
SC0	TB0OUT
SC1	TB1OUT

Note: TMPM066/067/068FW does not have the Handshake function (SCxCTS pin).

2.1.6 I2C Bus (I2C)

TMPM066/067/068FW incorporates 2 channels of I2C.

Table 2-10 Pin specifications

Channel	I2CxSDA	I2CxSCL	
I2C0	PC1	PC0	Wakeup function is supported.
I2C1	PG1	PG0	Wakeup function is not supported.

2.1.7 Toshiba Serial Peripheral Interface (TSPI)

TMPM066/067/068FW incorporates 1 channel of SPI.

Table 2-11 Pin specifications

TSPI Pin	Port
TSPICS0 (CSO/CSIN)	PF0
TSPITXD	PF1
TSPIRXD	PF2
TSPICLK	PF3

2.1.8 Analog/Digital Converter (ADC)

TMPM066/067/068FW incorporates 1 unit of ADC.

Table 2-12 Pin specifications

Analog input	AIN0 to 7
Ports	PA0 to 7

Note: TMPM066/067/068FW analog input pins are 8 channels of AIN0 to AIN7.

Table 2-13 Internal start-up trigger selection

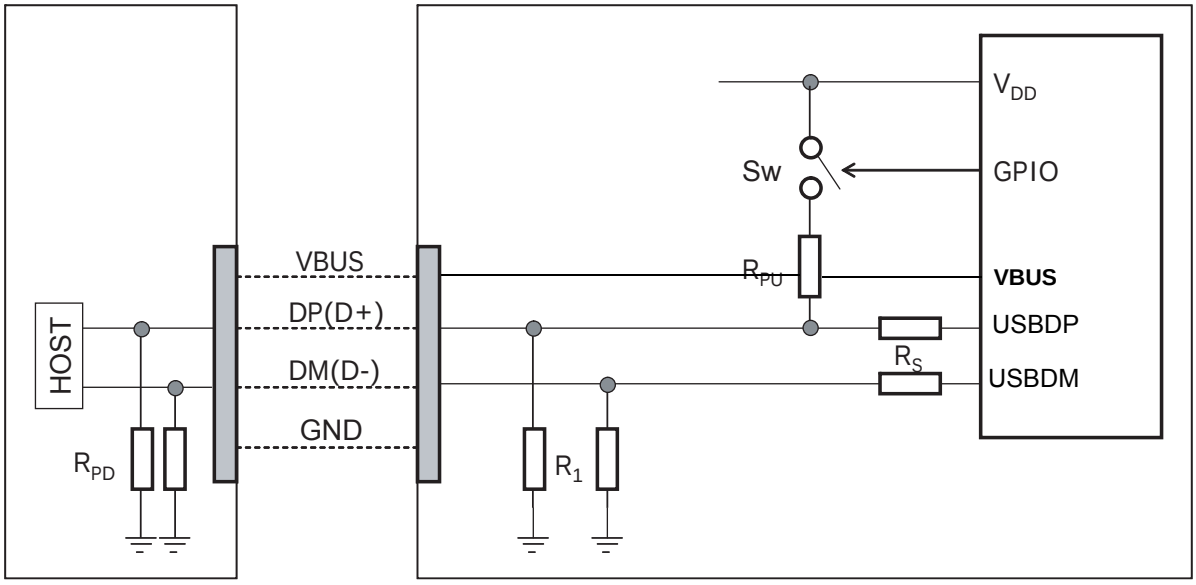
Type	Internal trigger
Normal AD conversion start-up trigger	A Compare register 0 match of TMRB ch0
Highest priority AD conversion start-up trigger	A Compare register 0 match of TMRB ch1

2.1.9 USB Device (USBD)

TMPM066/067/068FW supports 5 endpoints (EPs).

2.1.9.1 Reference Circuit

The example circuit is shown below when USB device circuit is used in TMPM066/067/068FW.



Note 1: On TMPM066/067/068FW, DP must be pull-ed up. The dumping resister should must be connected with DP and DM, too.

Note 2: Recommended value of resisters: Rpu= 1.5KΩ, Rs= 33Ω

Note 3: DP and DM are pulled-up by R1 if the Flow-through Current reducing (Optional). Recommended value of resister: R1= equal or more than 500KΩ.

Note 4: Rpu have on/off control by GPIO when Rpu and R1 are connected with TMPM066/067/068FW. Rpu must be disconnected when USB device circuit is not used.(Optional)

Note 5: For releasing USB suspend mode, it should be by the VBUS pin.(Recomended)

Note 6: We recommend the processing of Note 3 when USB is unused.

2.1.10 Debug Interface

TMPM066/067/068FW supports serial wire debug ports.

Table 2-14 Pin specifications

I/F	SWDIO	SWCLK
Serial wire	PE5	PE4

3. Processor Core

The TX00 series has a high-performance 32-bit processor core (the Arm Cortex-M0 processor core). For information on the operations of this processor core, please refer to the "Cortex-M0 Technical Reference Manual" issued by Arm Limited. This chapter describes the functions unique to the TX00 series that are not explained in that document.

3.1 Information on the processor core

The following table shows the revision of the processor core in the TMPM066/067/068FW.

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
TMPM066/067/068FW	r0p0-03

3.2 Configurable Options

The Cortex-M0 core has optional blocks. The following tables shows the configurable options in the TMPM066/067/068FW.

Configurable Options	Implementation
Interrpts	32
Data endiannes	Little-endian
SysTick timer	Present
Number of watchpoint comparators	2
Number of breakpoint comparators	4
Halting debug support	Present
Multiplier	Fast

Note: The fast multiplier provides a 32-bit × 32-bit multiply that yields the least-significant 32-bits.

3.3 Exceptions/ Interruptions

Exceptions and interruptions are described in the following section.

3.3.1 Number of Interrupt Inputs

The number of interrupt inputs can optionally be defined in the Cortex-M0 core.

TMPM066/067/068FW has 32 interrupt inputs.

3.3.2 SysTick

TMPM066/067/068FW has a SysTick timer which can generate SysTick exception.

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

3.3.3 SYSRESETREQ

The Cortex-M0 core outputs SYSRESETREQ signal when <SYSRESETREQ> bit of Application Interrupt and Reset Control Register are set.

TMPM066/067/068FW provides the same operation when SYSRESETREQ signal are output.

3.3.4 LOCKUP

When irreparable exception generates, the Cortex-M0 core outputs LOCKUP signal to show a serious error included in software.

TMPM066/067/068FW does not use this signal. To return from LOCKUP status, it is necessary to use non-maskable interruput (NMI) or reset.

3.4 Events

The Cortex-M0 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.

TMPM066/067/068FW does not use event output signals and event input signals. Please do not use SEV instruction and WFE instruction.

3.5 Power Management

The Cortex-M0 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.

TMPM066/067/068FW 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."

4. Memory Map

4.1 Memory map

The memory maps for the TMPM066/067/068FW are based on the Arm Cortex-M0 processor core memory map.

The internal ROM is mapped to the code of the Cortex-M0 core memory, the internal RAM is mapped to the SRAM region and the special function register (SFR) is mapped to the peripheral region respectively.

SRAM region and the special function register (SFR) is mapped to the peripheral region respectively.

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

TMPM066/067/068FW has bit-band feature equivalent to Cortex-M0 and the SRAM and SFR regions of TMPM066/067/068FW are all included into the bit-band region.(note: There is one exception area.)

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

For more information on each region, see the "Cortex-M0 Technical Reference Manual".

Note that access to regions indicated as "Fault" causes a hard fault. Do not access the vendor-specific region and the reserved region.

Figure 4-1 shows the memory map of the TMPM066/067/068FW.

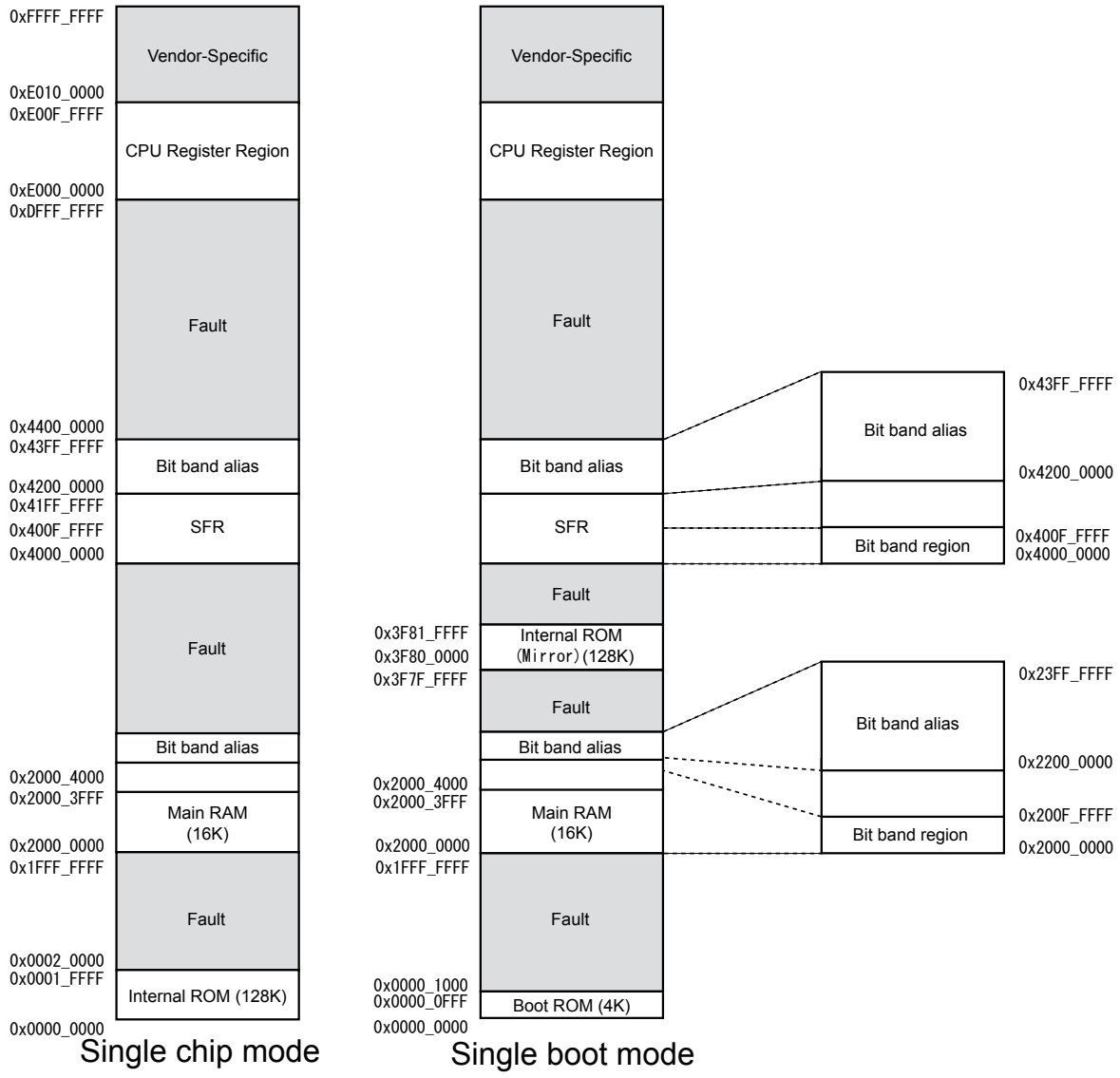


Figure 4-1 TMPM066/067/068FW Memory Map

4.2 Bus Matrix

The TMPM066/067/068FW contains three bus masters such as a CPU core, DMA controller and USB device.

Bus masters connect to slave ports (S0,S1,S2) of Bus Matrix. In the bus matrix, master ports (M0 to M9) connect to the 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.

4.2.1 Structure

4.2.1.1 Single chip mode, Single boot mode

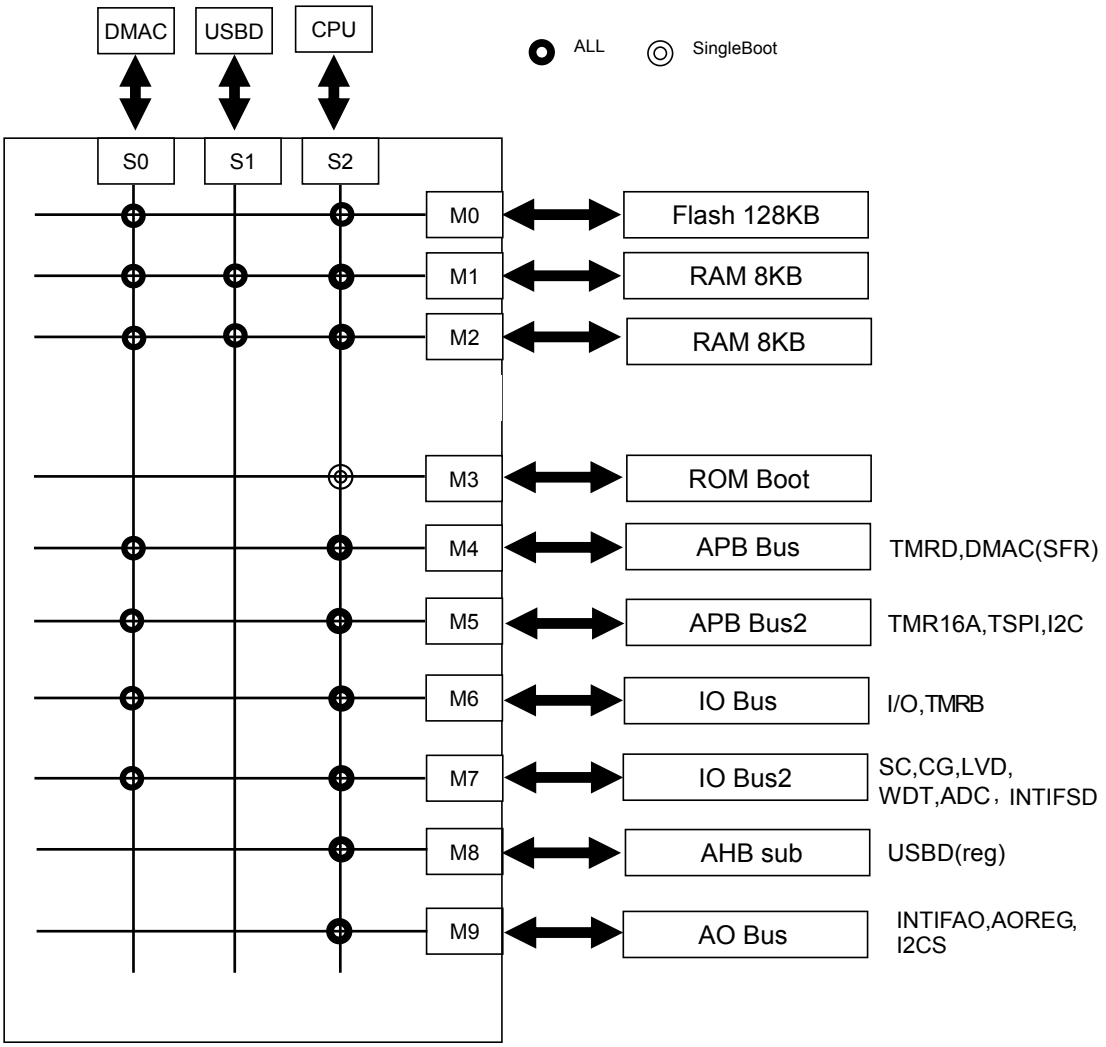


Figure 4-2 TMPM066/067/068FW

4.2.2 Connection table

4.2.2.1 Code area / SRAM area

(1) Single chip mode

Start Address			DMAC	USBD	Core
			S0	S1	S2
0x0000_0000	Flash ROM	M0	o	o	o
0x0002_0000	Fault	-	Fault	Fault	Fault
0x2000_0000	Main RAM0	M1	o	o	o
0x2000_2000	Main RAM1	M2	o	o	o
0x2400_0000	Fault	-	Fault	Fault	Fault

(2) Single boot mode

Start Address			DMAC	USBD	Core
			S0	S1	S2
0x0000_0000	BOOT ROM (mirror)	M3	Fault	Fault	o
0x0000_1000	Fault	-	Fault	Fault	Fault
0x2000_0000	Main RAM0	M1	o	o	o
0x2000_2000	Main RAM1	M2	o	o	o
0x2400_0000	Fault	-	Fault	Fault	Fault
0x3F80_0000	Flash ROM (mirror)	M0	o	Fault	o
0x3F82_0000	Fault	-	Fault	Fault	Fault

4.2.2.2 Peripheral area

Start Address			DMAC	USBD	Core
			S0	S1	S2
0x4000_0000	Fault	-	Fault	Fault	Fault
0x4000_8000	USBD	M8	-	-	o
0x4000_9000	Fault	-	Fault	Fault	Fault
0x4003_8000	AOSFR(INTIFAO, AOREG, I2CS) (note1)	M9	-	-	o
0x4003_8C00	Reserved	-	-	-	-
0x4004_C000	DMAC (uDMA)	M4	-	-	o
0x4005_8000	TMRD	M4	o	-	o
0x4008_D000	TMR16A	M5	o	-	o
0x4009_8000	TSPI	M5	o	-	o
0x4009_9000	Reserved	-	-	-	-
0x400A_0000	I2C	M4	o	o	o
0x400A_1000	Fault	-	Fault	Fault	Fault
0x400C_0000	PORT	M6	-	-	o
0x400C_0800	Reserved	-	-	-	-
0x400C_4000	TMRB(ch0-7)	M6	o	-	o
0x400C_4800	Fault	-	Fault	Fault	Fault
0x400E_1000	SIO/UART(ch0-1)	M7	o	-	o
0x400E_1500	Reserved	-	-	-	-
0x400F_2000	WDT	M7	-	-	o
0x400F_2100	Reserved	-	-	-	-
0x400F_3000	CG	M7	-	-	o
0x400F_3600	Reserved	-	-	-	-
0x400F_4000	LVD	M7	-	-	o
0x400F_4E00	INTIFSD	M7	o	-	o
0x400F_5000	Reserved	-	-	-	-
0x400F_C000	ADC	M7	-	-	o
0x400F_C100	Reserved	-	-	-	-
0x41FF_FF00	SFR(FLASH)	M7	-	-	o
0x4400_0000	Fault	-	Fault	Fault	Fault

Note 1: Byte access only is allowed. Bit-band access is impossible.

4.2.3 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
USBD	-	0x4000_8000
AOSFR	INTIFAO	0x4003_8000
	AOREG	0x4003_8400
	I2CS	0x4003_8800
DMA Controller (DMAC)	-	0x4004_C000
16-bit Timer (TMRD)	-	0x4005_8000
16-bit Timer (TMR16A)	ch0	0x4008_D000
	ch1	0x4008_E000
TSPI		0x4009_8000
I2C Serial bus interface (I2C)	ch0	0x400A_0000
	ch1	0x400A_1000
Input / Output port (PORT)	Port A	0x400C_0000
	Port B	0x400C_0100
	Port C	0x400C_0200
	Port D	0x400C_0300
	Port E	0x400C_0400
	Port F	0x400C_0500
	Port G	0x400C_0600
	Port H	0x400C_0700
	Port J	0x400C_0800
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
Serial Channel (SIO/UART)	ch0	0x400E_1000
	ch1	0x400E_1100
Watchdog Timer (WDT)		0x400F_2000
Clock/ Mode control (CG)		0x400F_3000
Low Voltage Detection Circuit (LVD)		0x400F_4000
INTIFSD		0x400F_4E00
Analog / Digital Converter (ADC)		0x400F_C000
SFR(FLASH)		0x41FF_FF00

5. Reset Operation

The following are sources of reset operation.

- Power On Reset
- Reset Pin(RESET)
- Low voltage detection circuit(LVD)
- Watch-dog timer(WDT)
- Application interrupt by CPU and signal from the reset register bit<SYSRESETREQ>

To recognize a source of reset, check RSTFLG register described in chapter of "Exception".

A reset by low voltage detection circuit is refer to the "Low voltage detection circuit".

A reset by WDT is refer to the chapter on the "Watch-dog timer".

A reset by <SYSRESETREQ> is referred to "Cortex-M0 Technical Reference Manual".

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

5.1 Cold Reset

5.1.1 Cold Reset by $\overline{\text{RESET}}$ pin

When turning-on power, $\overline{\text{RESET}}$ pin must be kept "Low".

When turning-on power, it is necessary to take a stable time of built-in regulator into consideration. In this product ,the internal regulator requires at least approximately 1.0ms to be stable. At cold reset, $\overline{\text{RESET}}$ pin must be kept "Low" for a duration of time sufficiently long enough for the internal regulator to be stable.

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

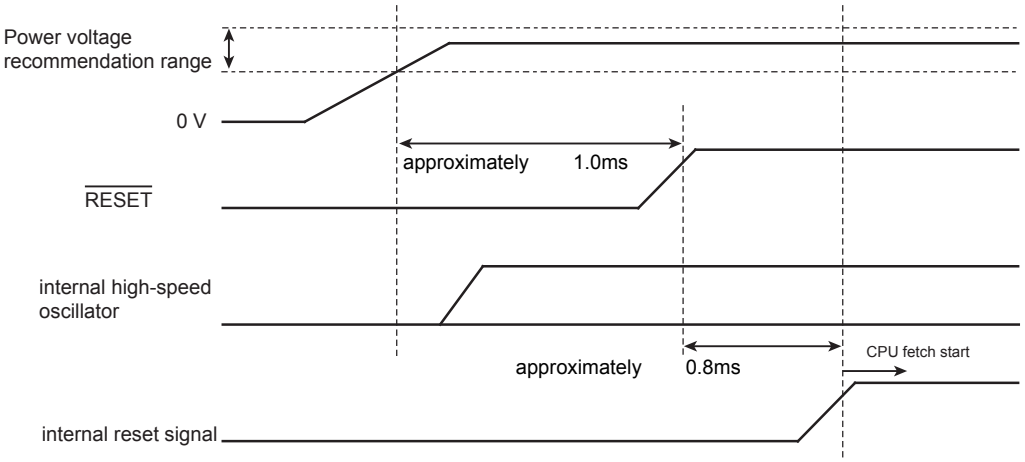


Figure 5-1 Cold reset Operation Sequence

Note: The above sequence is applied as well when restoring power.

5.1.2 Cold Reset with power-on-reset circuit

In case of Using power-on-reset circuit, there is a restriction on rising time of power line.

the power pin should start a power supply to reach the recommendation operation voltage range within 1.0ms.

Approximately 1.8ms after the power voltage becomes the recommendation operation voltage range, internal reset will be released.

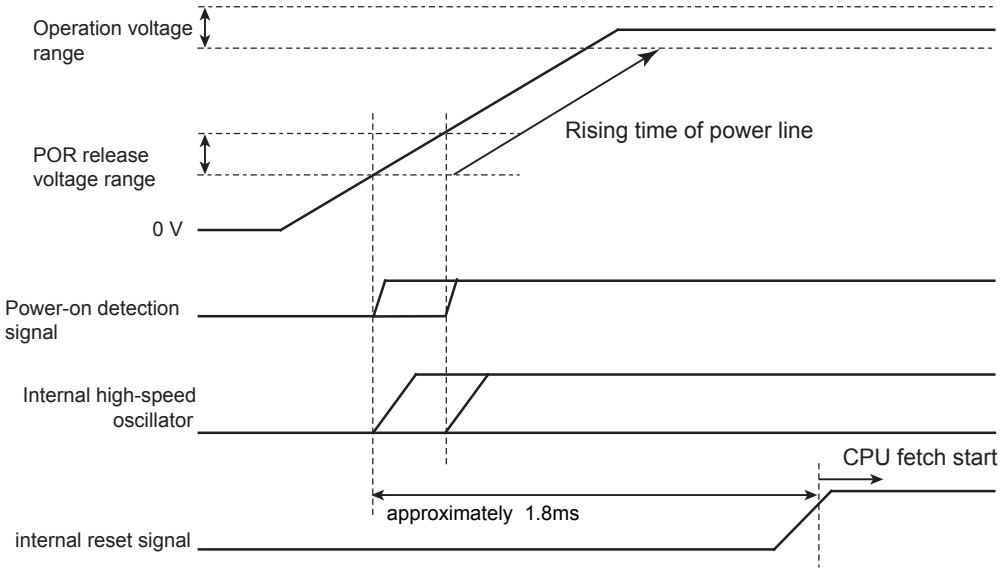


Figure 5-2 Cold reset with power-on-circuit

Note: The above sequence is applied as well when restoring power.

5.2 Warm Reset

To do reset TMPM066/067/068FW, the following conditions are required; power supply voltage is in the operational range ; $\overline{\text{RESET}}$ pin is kept "Low" at least for 12 internal high-speed clocks. Approximately 0.8ms after $\overline{\text{RESET}}$ pin becomes "High", internal reset will be released.

In case of WDT reset or <SYSRESETREQ>reset, internal reset will be released approimately 30 internal high-speed clocks after reset.

5.3 After reset

Most of the control register of the internal core and the peripheral function control register(SFR) are initialized by Warm reset.

System debug component registers(FPB, DWT, and ITM)of the internal core, 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 should be set if necessary.

6. Clock/Mode control

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

6.1 Features

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

In addition to NORMAL mode, the TMPM066/067/068FW can operate low power mode to reduce power consumption according to its usage conditions.

The clock/mode control block has the following functions:

- Controls the system clock
- Controls the prescaler clock
- Controls the PLL multiplication circuit

6.2 Registers

6.2.1 Register List

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

Clock/Mode control

Register name		Address (Base+)
Write Protect register	CGPROTECT	0x0000
Oscillation control register	CGOSCCR	0x0004
System clock control register	CGSYSCR	0x0008
Standby control register	CGSTBYCR	0x000C
PLL selection register of fsys	CGPLL0SEL	0x0020
Warm-up register of HOSC	CGWUPHCR	0x0030
fsys clock on/off register A	CGFSYSENA	0x0050
fsys clock on/off register B	CGFSYSENB	0x0054
ADC clock on/off register	CGSPCLKEN	0x005C
Optional function setting register	CGEXTENDO0	0x0060

6.2.2 CGPROTECT (Write 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	1	1	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PROTECT[7:0]	R/W	CG Register Write protection control (except this register) 0xC1: Register write enable Except 0xC1:Register write disable

6.2.3 CGOSCCR (Oscillation control register)

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

Bit	Bit Symbol	Type	Function
31-10	−	R	Read as "0".
9	OSCF	R	Status of Selected High-speed oscillator. 0:internal high-speed oscillator (IHOSC) 1:external high-speed oscillator (EHOSC)
8	OSCSEL	R/W	Select of fosc High-speed oscillator (Note1) 0: internal high-speed oscillator (IHOSC) 1: external high-speed oscillator (EHOSC)
7-3	−	R	Read as "0".
2-1	EOSCEN[1:0]	R/W	Select external OSC source (EHOSC) (note2) 00: no use 01: external oscillator (EHOSC) 10: external clock input 11: Reserved
0	IOSCEN	R/W	Internal High-Speed oscillator (IHOSC) 0: Stop 1: Oscillation

Note 1: When the setting is modified, confirm whether the written value is reflected to the CGOSCCR<OSCF> bit before starting the next operation.

Note 2: Set to "01" only when using External High-speed clcok.(oscillator).

6.2.4 CGSYSCR (System control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	PRCKST			
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	GEARST		
After reset	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	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
31-28	-	R	Read as "0".
27-24	PRCKST[3:0]	R	Status of Prescaler clock(φT0) 0000: fc 0100:fc/16 1000:fc/256 0001: fc/2 0101:fc/32 1001:fc/512 0010: fc/4 0110: fc/64 1010-1111:Reserved 0011: fc/8 0111:fc/128
23-19	-	R	Read as "0".
18-16	GEARST[2:0]	R	Status of gear selection of system clock.(fsys) 000: fc 100: fc/16 001: fc/2 101 - 111: Reserved 010: fc/4 011: fc/8
15-12	-	R	Read as "0".
11-8	PRCK[3:0]	R/W	Prescaler clock(φT0) 0000: fc 0100: fc/16 1000: fc/256 0001: fc/2 0101: fc/32 1001: fc/512 0010: fc/4 0110: fc/64 1010- 1111: Reserved 0011: fc/8 0111: fc/128 Specifies the prescaler clock to peripheral circuit.
7-3	-	R	Read as "0".
2-0	GEAR[2:0]	R/W	System clock (fsys) gear. 000: fc 100: fc/16 001: fc/2 101 - 111: Reserved 010: fc/4 011: fc/8

6.2.5 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	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	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	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1-0	STBY[1:0]	R/W	Low power consumption mode control. 00: IDLE 01: STOP1 10: Reserved 11: Reserved

Note: Access to the Reserved is prohibited.

6.2.6 CGPLL0SEL(PLL Selection Register of fsys)

	31	30	29	28	27	26	25	24
bit symbol	PLL0SET							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	PLL0SET							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	PLL0SET							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PLL0ST	PLL0SEL	PLL0ON
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	PLL0SET[23:0]	R/W	PLL multiplying value.
7-3	−	R	Read as "0".
2	PLL0ST	R	Status of selected clock in PLL 0: fosc 1: fPLL
1	PLL0SEL	R/W	Use of fsys clock 0: fosc 1: fPLL
0	PLL0ON	R/W	PLL (multiplying circuit) operation 0: Stop 1: Oscillation

6.2.7 CGWUPHCR(Warm-up register of HOSC)

	31	30	29	28	27	26	25	24
bit symbol	WUPT							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	WUPTT				-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	WUCLK
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	WUEF	WUON
After reset	0	0	0	0	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.(note3)
19-9	−	R	Read as "0".
8	WUCLK	R/W	Select High-Speed warm-up counter clock (note2) 0: Internal OSC (IHOSC) 1: External OC (EHOSC)
7-2	−	R	Read as "0".
1	WUEF	R	Status of warm-up timer(note1) 0: WUP finish 1: WUP active
0	WUON	W	Operation of warm-up timer 0: don't care 1: Warm-up timer start

Note 1: Do not modify the registers during the warm-up (WUEF=1'b1).Modify registers only when WUEF=1'b0.

Note 2: Use the internal oscillator to warm up when the MCU returns from STOP1. Do not use the external oscillator in this process.

Note 3: Warm-up time equation are shown below.

Warm-up time = (<WUPT[15:0]> + 0x10) × Clock cycle

(*** WUPT[3:0]=0x0 is fixed.)

(example)

・ When using IOSC 10MHz,<WUP[15:0]> = 0x2700

Warm-up time = (0x2700 + 0x10) × (1/10 MHz)

= 0x2710 × 100ns

= 10000 × 100ns

= 1.0 ms

・ When using EOSC 16MHz,<WUPT[15:0]> = 0x3E70.

Warm-up time = (0x3E70 + 0x10) × (1/16 MHz)

= 0x3E80 × 62.5 ns

= 16000 × 62.5ns

= 1.0 ms

6.2.8 CGFSYSENA (fsys Clock on/off register A)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	IPENA18	IPENA17	IPENA16
After reset	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
bit symbol	IPENA15	IPENA14	IPENA13	IPENA12	IPENA11	IPENA10	IPENA09	IPENA08
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	IPENA07	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-19	–	R/W	Write as "0".
18	IPENA18	R/W	Clock control for TMRD 0: Clock stop 1: Clock supply
17	IPENA17	R/W	Clock control for USBD 0: Clock stop 1: Clock supply
16	IPENA16	R/W	Clock control for ADC 0: Clock stop 1: Clock supply
15	IPENA15	R/W	Clock control for DMAC 0: Clock stop 1: Clock supply
14	IPENA14	R/W	Clock control for TSPI 0: Clock stop 1: Clock supply
13	IPENA13	R/W	Clock control for SIO ch0 0: Clock stop 1: Clock supply
12	IPENA12	R/W	Clock control for I2C ch0 0: Clock stop 1: Clock supply
11	IPENA11	R/W	Clock control for TMR16A 0: Clock stop 1: Clock supply
10	IPENA10	R/W	Clock control for TMRB(ch4 - 6) 0: Clock stop 1: Clock supply
9	IPENA09	R/W	Clock control for TMRB (ch0 - 3) 0: Clock stop 1: Clock supply
8	IPENA08	R/W	Clock control for PORT J 0: Clock stop 1: Clock supply
7	IPENA07	R/W	Clock control for PORT H. 0: Clock stop 1: Clock supply
6-0	-	R/W	Write as "0".

Note: Even if the initial value of a register is set to "0" (clock stops), every clock is supplied to the registers above mentioned during the reset operation.

6.2.9 CGFSYSENB (fsys Clock on/off register B)

	31	30	29	28	27	26	25	24
bit symbol	IPENB31	IPENB30	IPENB29	IPENB28	-	-	-	-
After reset	1	1	1	1	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
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	IPENB31	R/W	Clock control for I2C ch1 0: Clock stop 1: Clock supply
30	IPENB30	R/W	Clock control for WDT 0: Clock stop 1: Clock supply
29	IPENB29	R/W	Clock control for SIO ch1 0: Clock stop 1: Clock supply
28	IPENB28	R/W	Clock control for TMRB ch7 0: Clock stop 1: Clock supply
27-0	-	R/W	Write as "0".

Note: Even if the initial value of a register is set to "0" (clock stops), every clock is supplied to the registers above mentioned during the reset operation.

6.2.10 CGSPCLKEN (ADC Clock on/off register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	ADCKEN
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
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
31-17	-	R	Read as "0".
16	ADCKEN	R/W	Clock (fc) control for ADC 0: Clock stop 1: Clock supply
15-1	-	R	Read as "0".
0	-	R/W	Write as "0".

6.2.11 CGEXTENDO0 (Optional function 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	-	-	DCLKEN	EHCLKSEL	-	-	USBSEL	USBENA
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-6	−	R/W	Write as "0".
5	DCLKEN	R/W	fPLL Clock control for TMRD 0: Clock stop 1: Clock supply
4	EHCLKSEL	R/W	EHCLK selection （for fc/fsys=24MHx(Max)) 0: 1/1 Clock(8 to 24 MHz) 1: 1/2 Clock(48MHz)
3-2	−	R/W	Write as "0".
1	USBSEL	R/W	USB Clock selection 0: PLL multiplying (48MHz) 1: X1 input clock (external input 48MHz)
0	USBENA	R/W	USB clock control 0: USB Clock stop 1: USB Clock supply

6.3 Clock control

6.3.1 Clock Type

Each clock is defined as follows.

fosc	: Clock from the internal oscillator, or input from X1&X2 pin.
f_{PLL}	: Clock multiplied by PLL.
fc	: Clock specified by CGOSCCR<OSCSEL> (high-speed clock)
fsys	: Clock specified by CGSYSCR<GEAR[2:0]>.(system clock)
φT0	: Clock specified by CGSYSCR<PRCK[3:0]> (prescaler clock)

6.3.2 Initial Values after Reset

Reset operation initializes the clock configuration as follows.

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

6.3.3 Clock system Diagram

The clock system diagram shows as follows.

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

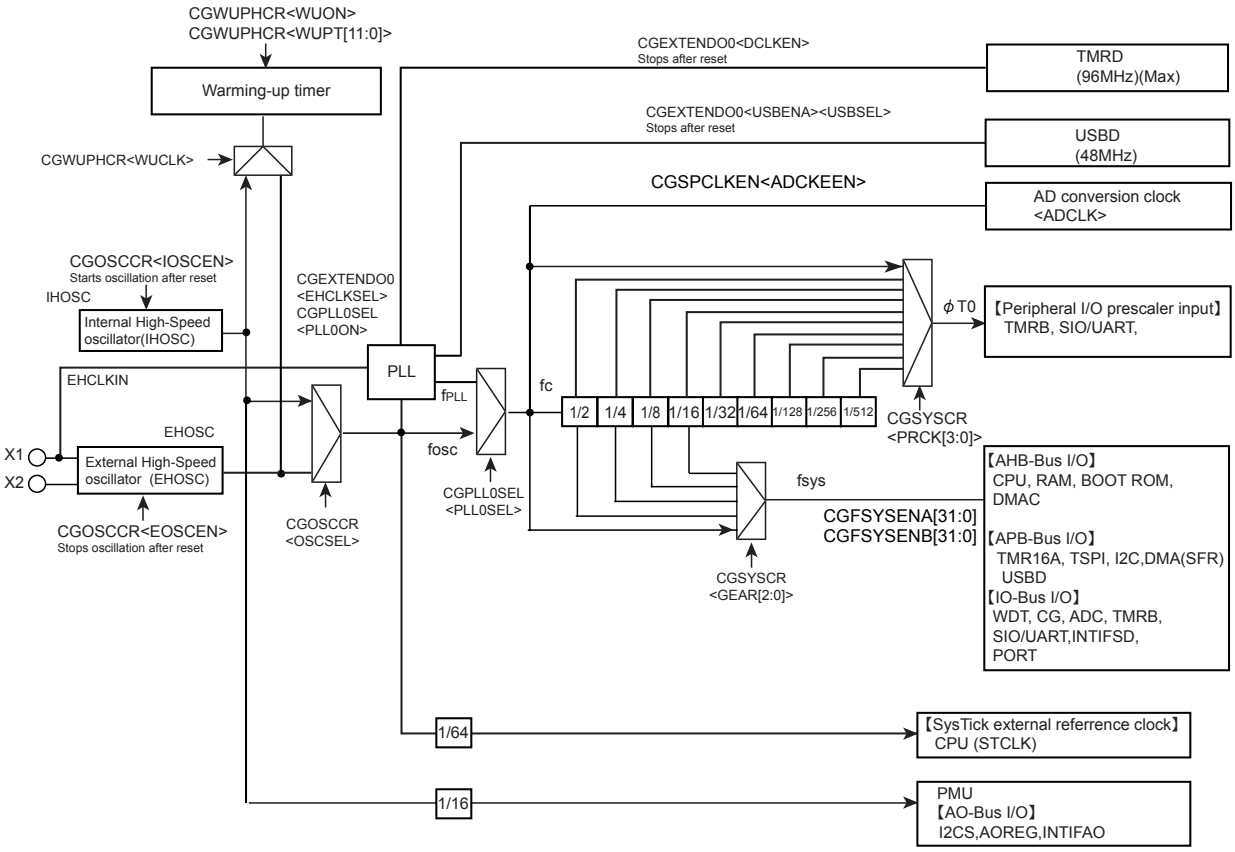


Figure 6-1 Clock Block Diagram

Note: the multiplying clock by PLL can not be made with the on-chip oscillator (IHOSC).

6.3.4 Clock Multiplication Circuit (PLL)

This circuit outputs the f_{PLL} clock (96MHz max) that is multiplied by 12/8/6 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.

6.3.4.1 Operation start

The PLL is disabled after reset.

To enable the PLL, firstly, specify "0" to CGPLL0SEL<PLL0ONT> before the multiplying value of CGPLL0SEL<PLL0SET> is set.

Secondly, after the initialization time of approximately 100μs has elapsed, specify "1" to <PLL0ON> to start the operation of the PLL. Finally, after the lockup time of 100μs has elapsed, specify "1" to GPLL0SEL<PLL0SEL> to use fPLL clock of which fosc is multiplied by 6,8, or 12.

Note that the certain time is necessary, including using the warm-up function, until the PLL operation becomes stable.

Setting value of <PLL0SET > are as following.

fOSC	Multiplying	<PLL0SET>
8MHz	12	0xC60B00
10MHz	8	0xC60700
12MHz	8	0xC60700
16MHz	6	0xC60500

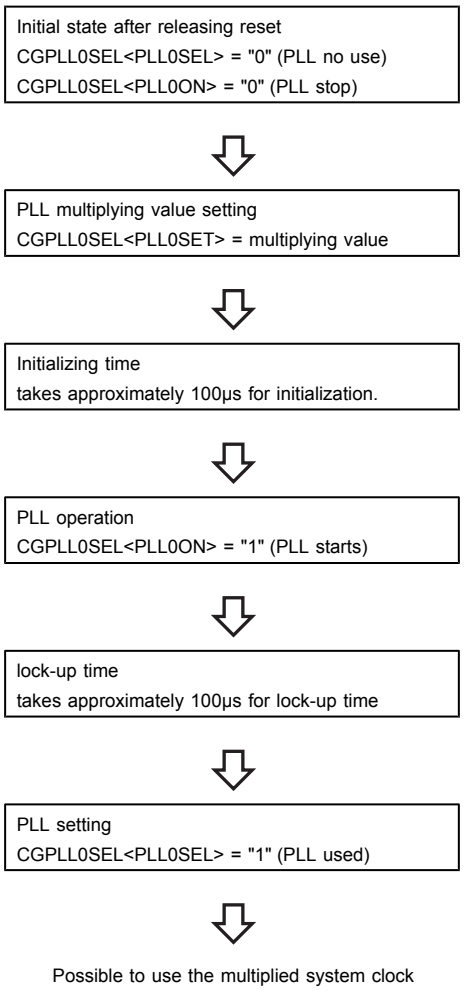
6.3.4.2 Changed the multiplication

The CGPLL0SEL<PLL0SEL> is first made "0" when the multiplying value is changed. Then, read CGPLL0SEL<PLL0ST>="0" to confirm whether the multiplication clock setting is changed to be "unused". After that, specify "0" to CGPLL0SEL<PLL0ON> to stop the PLL.

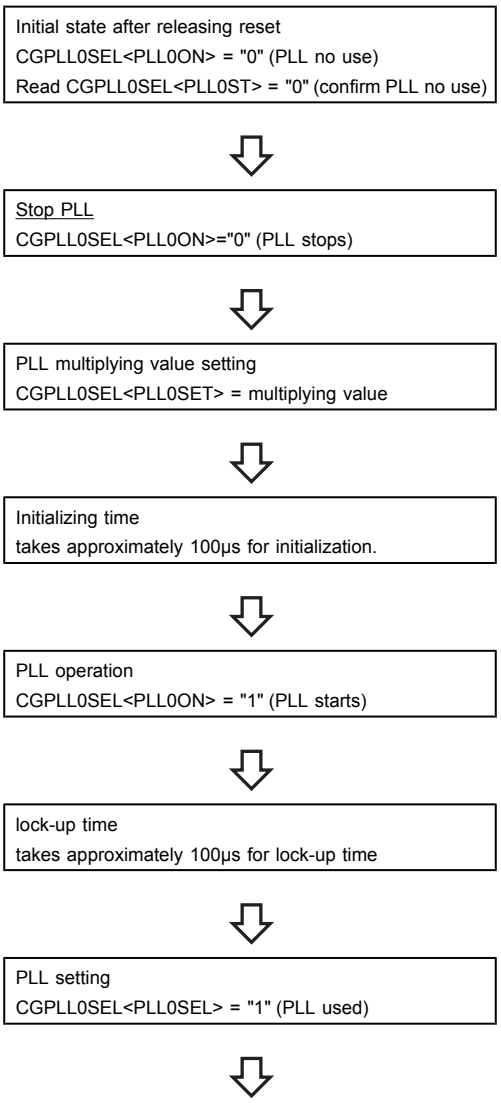
After that the multiplying CGPLL0SEL<PLL0SET> value is change, the CGPLL0SEL<PLL0ON> is set to "1" after approximately.

After the lockup time of approximately 100μs has elapsed, specify "1" to CGPLL0SEL<PLL0SEL>.

6.3.4.3 The sequence of PLL setting



6.3.4.4 The sequence of PLL setting (Changed multiplying value)



6.3.5 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.

When using internal high-speed oscillator (IHOSC), do not use it as system clock which high accuracy assurance is required. and without PLL too.

External High-speed clock (oscillator /clock input) can be used with PLL multiplying.

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 6-1 shows the example of the operation frequency by the setting of PLL and the clock gear.

Table 6-1 System clock frequency when PLL is used

External oscillator (MHz)	External clock input (MHz)	Internal Oscillator (MHz)	PLL multiplying	Max. operation frequency (fc) (MHz)	USBCLK	TMRD Clock	CGEXTENDO0 setting	
8	8	-	12	24	48	96	EHCLKSEL=0	-
10	10	-	8	20	-	80		USB can not use
-	-	10	-	10	-	-		USB,TMRD can not use
12	12	-	8	24	48	96		-
16	16	-	6	24	48	96		-
-	24	-	-	24	-	-		USB,TMRD can not use
-	48	-	-	24	48	-	EHCLKSEL=1	TMRD can not use

When CGSYSCR<GEAR>="000", fsys= fc/1.

6.3.6 Clock Supply Setting Function

This MCU has the clock supply function that performs on-off control of clock supply to unused peripheral functions; therefore the consumption current can be reduced. After reset, the clock is not supplied to peripheral functions with few exceptions.

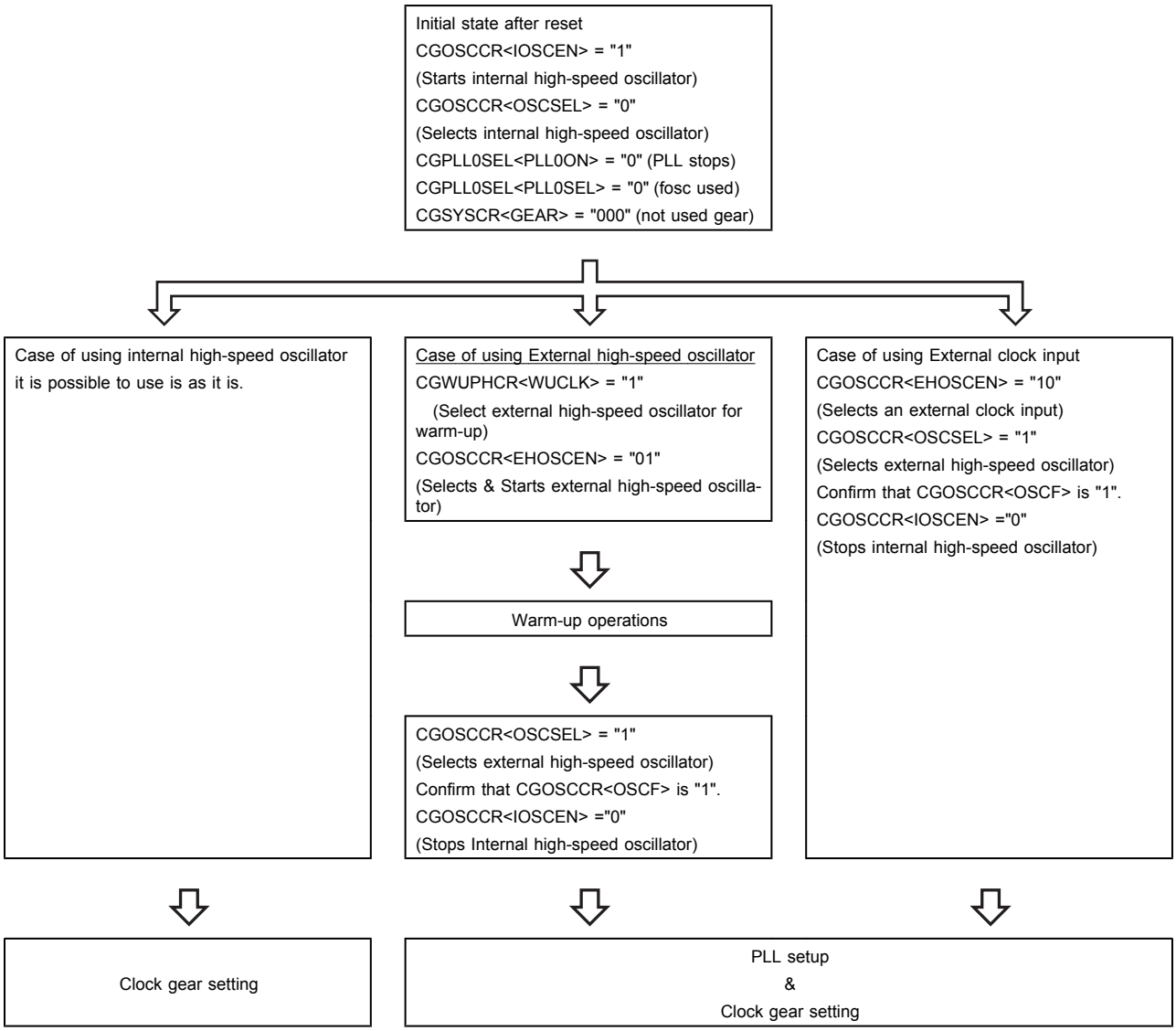
To supply the clock to the peripheral functions, specify "1" to the corresponding bits of CGFSYSENA and CGFSYSENB. For details of the registers, refer to the descriptions of each register.

6.3.7 Clock setting

The system clock can be selected by CGOSCCR. After the clock is selected, the PLL setting is done if necessary with CGPLL0SEL. And the clock gear is set with CGSYSCR.

The clock setup sequence is shown as follow.

Clock setup sequence



Note: When an internal high-speed oscillator is used as fsys, the usage of PLL function is prohibited.

6.4 Modes and Mode Transitions

This device has the NORMAL mode and the low power consumption mode (IDLE, STOP1) of Operation mode.

Depend on the mode transitions of each mode, this one can reduce power consumption.

6.4.1 Operation mode Transitions

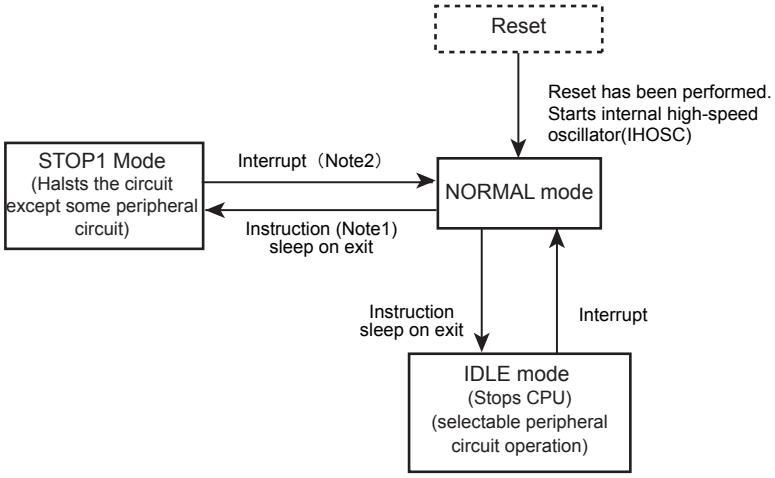


Figure 6-2 Mode Transition Diagram

Note 1: The warm-up is needed when returning form the STOP1 mode, The warm-up time is needed to set in NORMAL mode.

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

6.4.2 Transition to STOP1 mode

When transfer to STOP1 mode, it should be done the following processing.

1	Set WDMOD<WDTE>="0"	Disable WDT.
2	Set WDCR<WDCR>="0xB1"	Disable WDT. (Disable code writing.)
3		Wait for FLASH status until Ready, after that following sequence.
4	Read CGWUPHCR<WUEF>	Wait for end of Warming-up for HOSC
5	Set CGWUHPCR<WUCLK>="0"	Set Warm-up clock to IOSC
	Set CGWUPHCR<WUPT>= xxx	Set Warming-up time for IHOSC return from STOP1 mode.
6	Set CGSTBYCR<STBY>="01"	Select STOP1 mode.
7	Set CGPLL0SEL<PLL0SEL>="0"	Set PLL of fsys to fOSC (= PLL no USE).
8	Read CGPLL0SEL<PLL0ST>	Wait for PLL status of fsys until off state(fosc="0").
9	Set CGPLL0SEL<PLL0ON>="0"	Stop PLL of fsys
10	Set CGOSCCR<IOSCEN>="1"	Enable IOSC
11	Set CGOSCCR<OSCSEL>="0"	Set fosc to IOSC
12	Read CGOSCCR<OSCF>	Wait for fosc status until (IOSC)(="0").
13	Set CGOSCCR<EOSCEN>="00"	Set EHOSC off
14	Read CGOSCCR<EOSCEN>	Check the register status of above 13(="0")
15	WFI execution	Shift to STOP1 mode

6.5 Operation mode

6.5.1 NORMAL mode

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

It is shifted to the NORMAL mode after reset.

6.6 Low Power Consumption Modes

This device has two low power consumption modes: IDLE and STOP1. To shift to the low power consumption mode, specify the mode in the system control register CGSTBYCR<STBY[1:0]> and execute the WFI (Wait For Interrupt) instruction.

In this case, execute reset or generate the interrupt to release the mode.Releasing by the interrupt requires settings in advance. See the chapter "Exceptions" for details.

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

Note 2: This device does not support the low power consumption mode configured with the SLEEPDEEP bit in the Cortex-M0 core. Setting the <SLEEPDEEP> bit of the system control register is prohibited.

The features of IDLE, STOP1 mode are described as follows.

6.6.1 IDLE mode

Only the CPU is stopped in this mode.

Each peripheral function is enabled or disabled by the registers setting or peripheral or Clock Supply Setting Function.

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

6.6.2 STOP1 mode

Except the specific I2C block (Wake-up function block only) and the LVD, All the internal circuits including the internal oscillator are brought to a stop in STOP1 mode.

When releasing the 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 port register.Table 6-2 shows the pin status in the STOP1 mode.

Table 6-2 Pin States in the STOP1 mode

Function	Pin name	I/O	STOP1
Control pin	RESET_X,MODE	Input	o
Oscillator	X1	Input	×
	X2	Output	×
Port	SWCLK (Debug I/F setting, case of PxFRn<PxmFn>="1")	Input	Depends on PxIE[m]
	SWDIO (Debug I/F setting, case of PxFRn<PxmFn>="1")	Input	Depends on PxIE[m]
		Output	Enable when data is valid. Disabled when data is invalid
	INT0 to 5 (Interrupt setting, case of PxFRn<PxmFn>="1" and PxIE<PxmIE>="1")	Input	o
	If using other than listed above	Input	Depends on PxIE[m]
		Output	Depends on PxCR[m]

o : Valid input or output.
× : Invalid input or output

6.6.3 Low power Consumption Mode Setting

The low power consumption mode is specified by the setting of the standby control register CGSTBYCR<STBY[1:0]>.

Table 6-3 shows the mode setting in the <STBY[1:0]>.

Table 6-3 Low power consumption mode setting

Mode	CGSTBYCR <STBY[1:0]>
IDLE	00
STOP1	01

Note: Do not set any value other than those shown above in <STBY[1:0]>.

6.6.4 Operational Status in Each Mode

Table 6-4 show the operational status in each mode.

Table 6-4 Operational Status in Each Mode

Block	NORMAL Internal high-speed oscillator (IHOSC)	NORMAL external high-speed oscillator (EHOSC)	IDLE Internal high-speed oscillator (IHOSC)	IDLE external high-speed oscillator (EHOSC)	STOP1 (Note 1)
Processor core	o	o	−	−	−
DMAC	o	o	o	o	−
IO port	o	o	o	o	−
ADC	o	o	o	o	-
SIO/UART	o	o	o	o	−
I2C	o	o	o	o	−(Note2)
TSPI	o	o	o	o	-
USBD	o	o	o	o	-(Note3)
TMRB	o	o	o	o	−
TMR16A	o	o	o	o	−
TMRD	o	o	o	o	−
WDT	o	o	o(Note4)	o(Note4)	−
LVD	o	o	o	o	o
CG	o	o	o	o	−
External high-speed oscillator (EHOSC)	o	o	o	o	−
Internal high-speed oscillator (IHOSC)	o	o(Note 5)	o	o(Note 5)	−
RAM	o	o	o	o	−

o : Operation is available when in the target mode.

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

Note 1: Before entering STOP1 mode, it should be stopping peripheral functions of "−". and them move to that mode.
Note 2: I2C wake-up function only is available.
Note 3: Wake-up function only is available.
Note 4: Pay attention that the counter of watch dog timer function can not be cleared by CPU while in IDLE mode.
Note 5: After the reset operation is released or STOP1 mode is released, the clock is supplied from the internal oscillator.

6.6.5 Releasing the Low Power Consumption Mode

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

Details are shown in Table 6-5.

Table 6-5 Release Source in Each Mode

Low power consumption mode			IDLE	STOP1
release source	Interrupt	INT0,INT1,INT2,INT3,INT4,INT5	o	o
		INTRX0,INTTX0,INTRX1,INTTX1	o	×
		INTSPIRX,INTSPIYX,INTSPIERR	o	×
		INTI2C0, INTI2C1	o	×
		INTDMA	o	×
		INT16A0, INT16A1	o	×
		INTTB0 to 7	o	×
		INTI2CS	o	o
		INTTMRD	o	×
		INTUSBD	o	×
		INTUSBWKUP	o	×
		INTAD,INTADHP	o	×
	SysTick interrupt		o	×
	Non-Maskable Interrupt (INTWDT)		o	×
	Non-Maskable Interrupt (INTLVD)		o	o
	RESET (WDT)		o	×
	RESET (LVD)		o	o
	RESET (RESET_X pin)		o	o

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

× : Unavailable

Note 1: 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 2: When releasing by interrupting level mode from IDLE, STOP1 mode, hold the level until the interrupt handling starts. If the level is changed before that, the correct interrupt handling cannot be started.

- Release by interrupt request

  To release the low power consumption mode by an interrupt, the CPU must be set in advance to detect the interrupt. In addition to the setting in the CPU, the INTIF must be set to detect the interrupt to be used to release the STOP1 modes.

- Release by Non-Maskable Interrupt (NMI)

  There are two kinds of NMI sources: WDT interrupt (INTWDT), LVD interrupt (INTLVD).

  INTWDT can only be used in the IDLE mode.

- Release by reset

  Any low power consumption mode can be released by reset from the RESET pin,

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

  When STOP1 mode is released by reset operation, automatic warm-up is not performed; thus the reset signal must be kept valid until oscillation operation becomes stable.

- Release by SysTick interrupt

  SysTick interrupt can only be used in the IDLE mode.

Refer to "Interrupts" in the chapter of "Exceptions" for details.

6.6.6 Warm-up

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

In the mode transition from STOP1 to the NORMAL, the warm-up counter and the internal oscillator are activated automatically. And then the system clock output is started after the elapse of warm-up time.

It is necessary to set a warm-up time in the CGWUPHCR<WUPT[11:0]> before executing the instruction to enter the STOP1 mode.

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

Table 6-6 Warm-up setting in mode transition

Mode transition	Warm-up setting
NORMAL → IDLE	Not required
NORMAL → STOP1	Not required
IDLE → NORMAL	Not required
STOP1 → NORMAL	Auto-warm-up

6.6.7 Clock Operations in Mode Transition

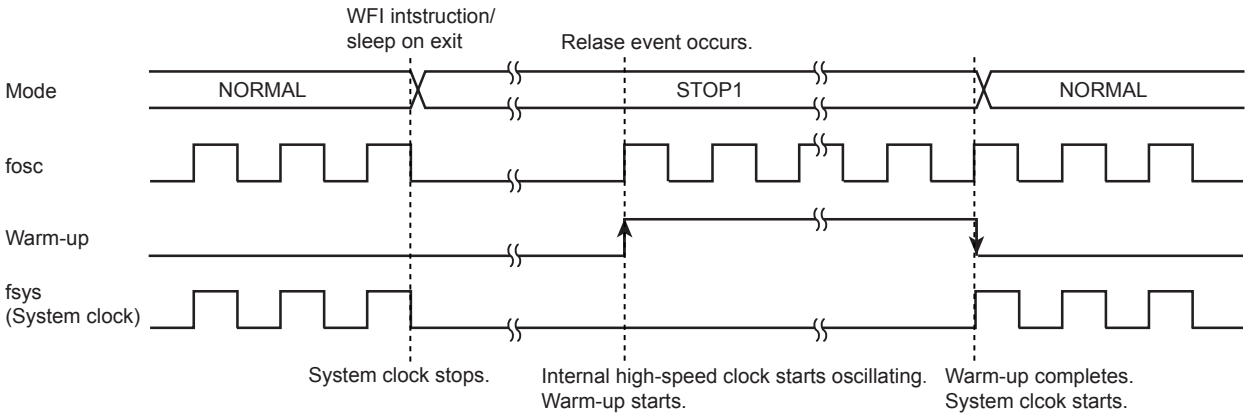
The clock operations in mode transition are described as follows.

6.6.7.1 Transition of operation modes: NORMAL → STOP1 → NORMAL

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

Before the MCU enters STOP1 mode, specify a warm-up time to WUPHCR<WUPT[11:0]>.

When releasing by reset is executed, automatic warm-up is not performed. Input a reset until the oscillator becomes stable.



7. Exceptions

This chapter describes features, types and handling of exceptions.

Exceptions have close relation to the CPU core. Refer to "Arm documentation set for the Arm Cortex-M series of processors" 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-M0

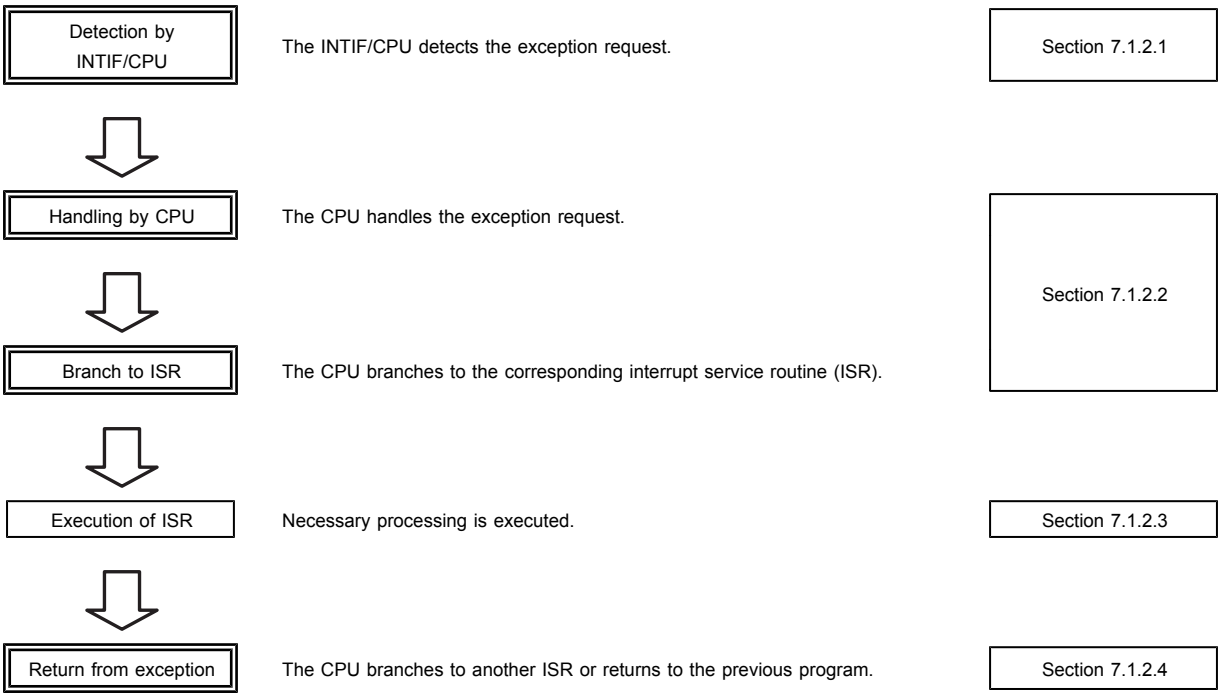
For detailed descriptions on each exception, refer to "Arm documentation set for the Arm Cortex-M series of processors".

- Reset
- Non-Maskable Interrupt (NMI)
- Hard Fault
- SVCall (Supervisor Call)
- 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 described later in this chapter.



7.1.2.1 Exception Request and Detection

(1) Exception occurrence

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

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

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

An interrupt request is generated from an external interrupt pin or peripheral function.For interrupts that are used for releasing a standby mode, relevant settings must be made in the INTIF. 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 7-1shows 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)	POR, Reset pin, WDT, LVD, SYSRESETREQ
2	Non-Maskable Interrupt	−2	WDT, LVD
3	Hard Fault	−1	Fault that cannot activate because a higher-priority fault is being handled or it is disabled
4 to 10	Reserved	−	
11	SVCall	Configurable	System service call with SVC instruction
12 to 13	Reserved	−	
14	PendSV	Configurable	Pending system service request
15	SysTick	Configurable	Notification from system timer
From 16	External interrupt	Configurable	External interrupt pin or peripheral function (Note2)

Note: External interrupts have different sources and numbers in each MCU. For details,"7.5.1.5 List of Interrupt Sources".

(3) Priority setting

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 of <PRI_n> is two bit, so the priority can be configured in the range from 0 to 3. The highest priority is "0". If multiple elements with the same priority exist, the smaller the number, the higher the priority becomes.

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

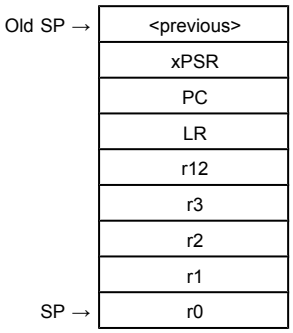
When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called "pre-emption".

(1) Stacking

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

1. Program Counter (xPSR)
2. Program Status register (PC)
3. r3 to r0
4. r12
5. 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.

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.

It should always be 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 to 0x28	Reserved	-	-
0x2C	SVCall	ISR address	Optional
0x30 to 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 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 Register to identify the source of a reset.

- External reset pin (RESET_X)

  A reset exception occurs when an external reset pin changes from "Low" to "High".

  (RSTFLG shows the power-on reset or Reset_x pin)
- Reset exception by POR

  The POR has a reset generating feature. For details, see the chapter on the POR.
- Reset exception by WDT

  The watchdog timer (WDT) has a reset generating feature. For details, see the chapter on the WDT.
- Reset exception by <SYSRESETREQ>

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

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

7.3 Non-Maskable Interrupts (NMI)

Non-maskable interrupts are generated from the following sources.

Refer to the INTFLAG0 Flag of INTIF (Interrupt IF) Register to identify the source of a non-maskable interrupt.

- Non-maskable interrupt by WDT

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

- Non-maskable interrupt by LVD

  The low voltage detection circuit (LVD) has a reset 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".

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 INTIF. Therefore, the settings of INTIF is required.

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 (A) that is not used to release standby are directly input to the CPU.(Route1)

The peripheral function interrupts (B)/(Route 2) used to release standby and interrupts from the external interrupt pin (Route 3) are input to the low-power consumption mode release logic of INTIF and are input to the CPU through the logic for releasing standby (Route 4 and 5).

The STOPxINTxxx<INTxxxEN> register determines to use interrupt requests (route 2 and route 3) for clearing low-power consumption mode (route 4 and 5).

When these interrupt requests are used to clear the low-power consumption mode, the CPU receives a "High" level interrupt request. (Route 2 to Route 4; Route 3 to Route 5).

The IDLEINTxxx is also as same as the STOPxINTxxx.

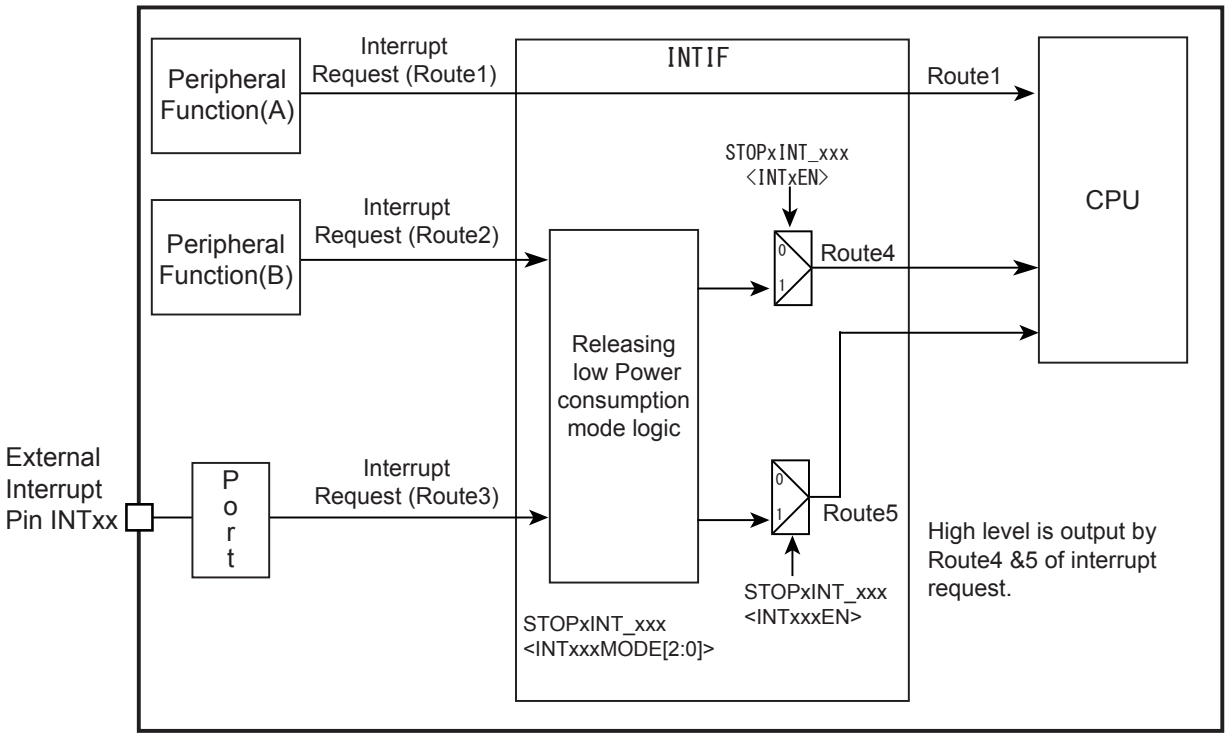


Figure 7-1 Interrupt Route

Interrupt request route 1

Interrupt request of peripheral functionÅF

Interrupt request route 2 (Interrupt for clearing low power consumption mode)

Interrupt request of peripheral functionÅF

Interrupt request route 3 (Interrupt for clearing low power consumption mode)

Interrupt request of External Interrupt PinÅF

The interrupt path through the low-power consumption mode release logic can also be used in NORMAL mode. Its setting is not different according to a mode.

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 INTIF. For these interrupt sources, appropriate settings must be made in the INTIF in advance.

When these interrupt requests are used for clearing the low-power consumption mode, the CPU receives a "High" level interrupt request. The interrupt requests, which cannot be used for clearing low-power consumption mode, are directly transmitted to the CPU.

For the external interrupt, be aware the Precautions of Next chapter.

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 1 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 7-2 shows the list of interrupt sources.

Table 7-2 List of Interrupt Sources

No	Offset Address	Interrupt Source		The active level to release the low power consumption mode	INTIF control register
0	0x40	INT0	External interrupt pin 0	↑↓ edge, L/H level, Both edge	INTCTL
1	0x44	INT1	External interrupt pin 1		
2	0x48	INT2	External interrupt pin 2		
3	0x4C	INT3	External interrupt pin 3		
4	0x50	INT4	External interrupt pin 4		
5	0x54	INT5	External interrupt pin 5		
6	0x58	INTRX0	UART reception		
7	0x5C	INTTX0	UART transmission		
8	0x60	INTRX1	UART reception		
9	0x64	INTTX1	UART transmission		
10	0x68	INTSPIRX	SPI reception		
11	0x6C	INTSPITX	SPI transmission		
12	0x70	INTSPIERR	SPI Error interrupt t		
13	0x74	INTI2C0	I2C Ch0 Interrupt		
14	0x78	INTI2C1	I2C Ch1 Interrupt		
15	0x7C	INTTDMA	DMAC interrupt		
16	0x80	INT16A0	16-bit TMRA0 match detection		
17	0x84	INT16A1	16-bit TMRA1 match detection		
18	0x88	INTTMRB0	16-bit TMRB (channel0)		
19	0x8C	INTTMRB1	16-bit TMRB (channel1)		
20	0x90	INTTMRB2	16-bit TMRB (channel2)		
21	0x94	INTTMRB3	16-bit TMRB (channel3)		
22	0x98	INTTMRB4	16-bit TMRB (channel4)		
23	0x9C	INTTMRB5	16-bit TMRB (channel5)		
24	0xA0	INTTMRB6	16-bit TMRB (channel6)		
25	0xA4	INTTMRB7	16-bit TMRB (channel7)		
26	0xA8	INTI2CS	I2C Wakeup Interrupt	↑ edge	INTCTL
27	0xAC	INTTMRD	TMRD Interrupt		
28	0xB0	INTUSBD	USBD Interrupt		
29	0xB4	INTUSBWKUP	USBD Wakeup Interrupt	↑ edge	INTCTL
30	0xB8	INTADHP	Highest priority AD conversion complete interrupt		
31	0xBC	INTAD	AD conversion complete interrupt		

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.

If an interrupt source is used for clearing a standby mode, setting the relevant INTIF register is also required. Enable the INTCTL<STOPxINT_xxx> bit and specify the active level in that register.

You must set the active level for interrupt requests from each peripheral function as shown in Table 7-2.

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

7.5.1.7 Precautions on Clearing Low-power Consumption Mode

To clear STOPx mode, the following two settings are necessary:

- Clearing low-power consumption mode (INTCTL<STOPxINT_xxx>)
- Enabling an interrupt (SETENA)

The operation returning to NORMAL mode from STOPx mode causes the suspended instruction to resume after the high-speed clock starts oscillation since the operation enters in the interrupt service routine.

	Interrupt mode Control register (STOPxINT_xxx)
	STOP1
INTx	Clear setting is required.
INTI2CS	
INTUSBWKUP	

7.5.1.8 Interrupt management numbers

This MCU provides the management numbers that indicate the presence or absence of the control register according to an interrupt factor, and indicate a register address. The usage of the management numbers and registers are shown in the table below:

Table 7-3 List of interrupt management numbers

Management No.	Usage	Interrupt control register (BUS)		Notes
000 to 015	These numbers are used if applicable under the following condition: an interrupt factor is a NMI.	AOBUS	According to an interrupt factor, the interrupt control register can be enabled/disabled or used to set a detection edge. For the address allocation of the management numbers, refer to the chapter on "INTIF (interrupt control) registers.	The registers of STOP2INT can be used.
016 to 031		IOBUS		The registers of STOP1INT or IDLEINT can be used.
032 to 095	These numbers are used if applicable under the following condition: -An interrupt factor is asynchronous (external interrupts or interrupts sent by the internal peripheral circuits, which operate on the clock except the system clock) -An interrupt factor is commonly used (OR).	AOBUS		The registers of STOP2INT can be used.
096 to 255		IOBUS		The registers of STOP1INT or IDLE INT can be used.
256 to 512	These numbers are used if applicable under the following condition: -An interrupt factor is synchronous with the system clock. For example, the interrupts sent from the internal peripheral circuits is applicable.	The register is not allocated.		

INTFLAG (Interrupt monitor flag register) are assigned by the Interrupt management numbers.

The table below shows the interrupt factors and management numbers implemented in this MCU.

Table 7-4 management numbers of TMPM066/067/068FW

No	Interrupt Source		No	Interrupt Source	
16	INTLVD	LVD interrupt Only lower than the setting voltage when voltage de-creasing.	102	INTDMA6	DMA ch6 transmission completion
17	INTLVD	LVD interrupt when returning from low voltage	103	INTDMA7	DMA ch7 transmission completion
18	INTWDT	WDT	104	INTDMA8	DMA ch8 transmission completion
32	INT0	External interrupt pin0	105	INTDMA9	DMA ch9 transmission completion
33	INT1	External interrupt pin1	106	INTDMA10	DMA ch10 transmission completion
34	INT2	External interrupt pin2	107	INTDMA11	DMA ch11 transmission completion
35	INT3	External interrupt pin3	108	INTDMA12	DMA ch12 transmission completion
36	INT4	External interrupt pin4	109	INTDMA13	DMA ch13 transmission completion
37	INT5	External interrupt pin5	110	INTDMA14	DMA ch14 transmission completion
38	INTI2CS	I2C(Wakeup) interrupt	111	INTDMA15	DMA ch15 transmission completion
39	INTUSBWKUP	USBD Wakeup	112	INTDMA16	DMA ch16 transmission completion
256	INTRX0	SIO reception	113	INTDMA17	DMA ch17 transmission completion
257	INTTX0	SIO transmission	114	INTDMA18	DMA ch18 transmission completion
258	INTRX1	SIO reception	115	INTDMA19	DMA ch19 transmission completion
259	INTXT1	SIO transmission	116	INTDMA20	DMA ch20 transmission completion
260	INTSPIRX	SPI reception	117	INTDMA21	DMA ch21 transmission completion
261	INTSPITX	SPI transmission	118	INTDMA22	DMA ch22 transmission completion
262	INTSPIERR	SPI Error	119	INTDMA23	DMA ch23 transmission completion
263	INTI2C0	I2C ch0	120	INTDMA24	DMA ch24 transmission completion
264	INTI2C0AL	I2C ch0 AL interrupt	121	INTDMA25	DMA ch25 transmission completion
265	INTI2C0BF	I2C ch0 Bus-Free	126	INTDMA30	DMA ch30 transmission completion
266	INTI2C0NACK	I2C ch0 NACK detection	127	INTDMA31	DMA ch31 transmission completion
267	INTI2C1	I2C ch1	128	INTDMAERR	DMA transmission error
268	INTI2C1AL	I2C ch1 AL interrupt	271	INT16A0	TMR16A ch0 match detection
269	INTI2C1BF	I2C ch1 Bus-Free	272	INT16A1	TMR16A ch1 match detection
270	INTI2C1NACK	I2C ch1 NACK detection	129	INTTB0	TMRB ch0 match detection/Over flow
96	INTDMA0	DMA ch0 transmission completion	130	INTTB0CAP0	TMRB ch0 input capture0
97	INTDMA1	DMA ch1 transmission completion	131	INTTB0CAP1	TMRB ch0 input capture1
98	INTDMA2	DMA ch2 transmission completion	132	INTTB1	TMRB ch1 match detection/Over flow
99	INTDMA3	DMA ch3 transmission completion	133	INTTB1CAP0	TMRB ch1 input capture0
100	INTDMA4	DMA ch4 transmission completion	134	INTTB1CAP1	TMRB ch1 input capture1
101	INTDMA5	DMA ch5 transmission completion	135	INTTB2	TMRB ch2 match detection/Over flow
136	INTTB2CAP0	TMRB ch2 input capture0	163	INTTMRD03	TMRD match detection 03
137	INTTB2CAP1	TMRB ch2 input capture1	164	INTTMRD04	TMRD match detection 04
138	INTTB3	TMRB ch3 match detection/Over flow	165	INTTMRD10	TMRD match detection 10
139	INTTB3CAP0	TMRB ch3 input capture0	166	INTTMRD11	TMRD match detection 11
140	INTTB3CAP1	TMRB ch3 input capture1	167	INTTMRD12	TMRD match detection 12
141	INTTB4	TMRB ch4 match detection/Over flow	168	INTTMRD13	TMRD match detection 13
142	INTTB4CAP0	TMRB ch4 input capture0	169	INTTMRD14	TMRD match detection 14
143	INTTB4CAP1	TMRB ch4 input capture1	273	INTUSBD	USB interrupt
144	INTTB5	TMRB ch5 match detection/Over flow	274	INTADHP	Highest priority AD conversion com-plete

Table 7-4 management numbers of TMPM066/067/068FW

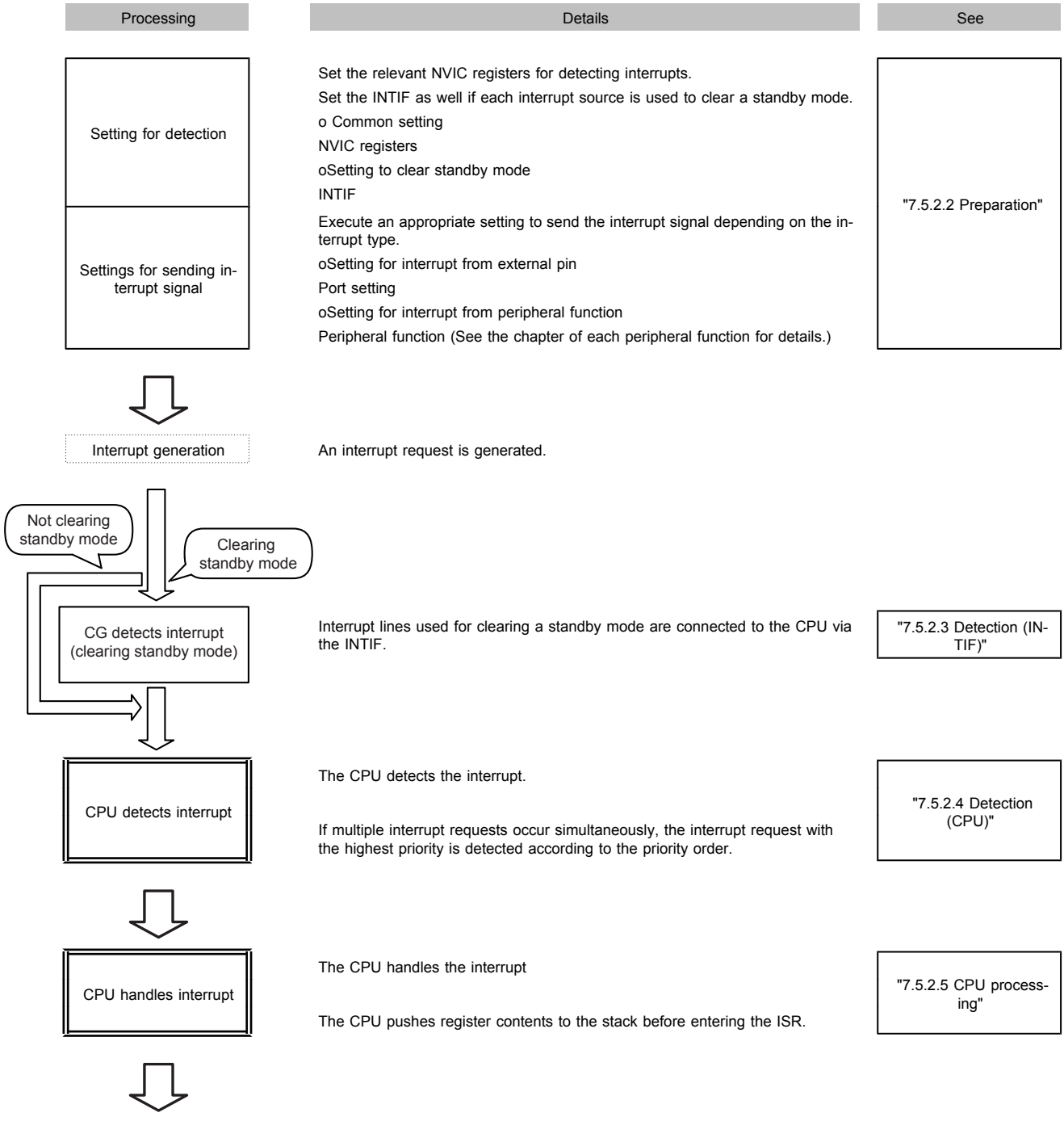
No	Interrupt Source		No	Interrupt Source	
145	INTTB5CAP0	TMRB ch5 input capture0	275	INTAD	AD conversion complete
146	INTTB5CAP1	TMRB ch5 input capture1			
147	INTTB6	TMRB ch6 match detection/Over flow			
148	INTTB6CAP0	TMRB ch6 input capture0			
149	INTTB6CAP1	TMRB ch6 input capture1			
150	INTTB7	TMRB ch7 match detection/Over flow			
151	INTTB7CAP0	TMRB ch7 input capture0			
152	INTTB7CAP1	TMRB ch7 input capture1			
160	INTTMRD00	TMRD match detection 00			
161	INTTMRD01	TMRD match detection 01			
162	INTTMRD02	TMRD match detection 02			

7.5.2 Interrupt Handling

7.5.2.1 Flowchart

The following shows how an interrupt is handled.

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



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

7.5.2.2 Preparation

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

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

To configure the INTIF, 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 INTIF 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 register setting
3. Pre configuration (1) (Interrupt from external pin)
4. Pre configuration (2) (Interrupt from peripheral function, Interrupt mask function)
5. Pre configuration (3) (Interrupt Set-Pending register)
6. configuring the INTIF
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)

(2) CPU register 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 two bits for assigning a priority level from 0 to 3. Priority level 0 is the highest priority level. If multiple sources have the same priority, the smallest-numbered interrupt source has the highest priority.

NVIC register		
<PRI_n>	←	"priority"

Note: "n" indicates the corresponding exceptions/interrupts.

(3) Pre configuration (1) (Interrupt from external pin)

Set "1" to the port function register of the corresponding pin for Interrupt from external pin. Setting PxIE[m] allows the pin to be used as the input port.

Port register		
PxIE<PxmIE>	←	"1"

Note: x: port number / m: corresponding bit.

Be careful not to enable interrupts that are not used.Also, refer to "7.5.1.4 Precautions when using external interrupt pins".

(4) Pre configuration (2) (Interrupt from peripheral function, Interrupt mask function setting)

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		
<SETPEND[m]>	←	"1"

Note: m: corresponding bit

(6) Configuring the INTIF

For an interrupt source to be used for exiting a standby mode, you need to set the active level and enable interrupts in the INTCTL register of the INTIF. The STOPxINT_xxx(IDELINT_xxx) 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 STOPxINT_xxx(IDLEINT_xxx) register.

See for each value.

INTIF register		
STOPxINT_xxx<INTxxxMODE>	←	Value corresponding to the interrupt to be used
STOPxINT_xxx<INTxxxEN>	←	"1"(Interrupt enabled)

Note: n: register number / xxx: 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 <CLRPEND[m]>	←	"1"
Interrupt Set-Enable <SETENA[m]>	←	"1"
Interrupt mask register		
PRIMASK	←	"0"

Note 1: m: corresponding bit

7.5.2.3 Detection (INTIF)

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

When the INTIF detects an interrupt request, it keeps sending the interrupt signal in "High" level to the CPU until the interrupt request is cleared in the Interrupt Control Register (INTCTL).

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 (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 xPSR, PC, LR, R12 and r3 to r0 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) Procedure during ISR

An ISR normally pushes register contents to the stack and handles an interrupt as required. The Cortex-M0 core automatically pushes the contents of xPSR, PC, LR, R12 and r3 to r0 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 INTCTL Register.

If an interrupt source is set as level-sensitive, an interrupt request continues to exist until it is cleared at its source.

Therefore, the interrupt source must be cleared.

In case of edge-sensitive, Clearing the interrupt source automatically clears the interrupt request signal from the INTIF.

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

7.6 Exception/Interrupt-Related Registers

The CPU's NVIC registers, INTIF registers and Interrupt masking function described in this chapter are shown below with their respective addresses.

7.6.1 Register List

NVIC registers — Base Address = 0xE000_E000

Register name	Address (Base +)
SysTick Control and Status Register	0x0010
SysTick Reload Value Register	0x0014
SysTick Current Value Register	0x0018
SysTick Calibration Value Register	0x001C
Interrupt Set-Enable Register	0x0100
Interrupt Clear-Enable Register	0x0180
Interrupt Set-Pending Register	0x0200
Interrupt Clear-Pending Register	0x0280
Interrupt Priority Register	0x0400 to 0x0430
Application Interrupt and Reset Control Register	0x0D0C
System Handler Priority Register	0x0D1C, 0x0D20
System Handler Control and State Register	0x0D24

peripheral function name: AOREG(note1) — Base Address= 0x4003_8400

Register name		Address (Base +)
Reset Flag Register	RSTFLG	0x0002
Reset Flag Register1	RSTFLG1	0x0003

Note 1: Byte access only is allowed. Bit-band access is impossible.

peripheral function name: INTIFAO — Base Address= 0x4003_8000

Register name		Address (Base +)
Interrupt Control Register(STOP2INT_032 to 096)	STOP2INT_xxx	0x0020

peripheral function name: INTIFSD — Base Address =0x400F_4E00

Register name		Address (Base +)
Interrupt Control Register (STOP1INT_016 to 031/ IDLEINT_061 to 031)	STOP1INT_xxx IDLEINT_xxx	0x0010
Interrupt Control Register (STOP1INT_096 to 255/ IDLEINT_096 to 255)	IDLEINT_096 to 255	0x0060
NMI Interrupt Monitor Flag (000 to 031)	INTFLAG0	0x0100
Interrupt Monitor Flag in AO area (032 to 063)	INTFLAG1	0x0104
Interrupt Monitor Flag in AO area (064 to 095)	INTFLAG2	0x0108
Interrupt Monitor Flag in SD area (096 to 127)	INTFLAG3	0x010C
Interrupt Monitor Flag in SD area (128 to 159)	INTFLAG4	0x0110
Interrupt Monitor Flag in SD area (160 to 191)	INTFLAG5	0x0114
Interrupt Monitor Flag in SD area (192 to 223)	INTFLAG6	0x0118
Interrupt Monitor Flag in SD area (224 to 255)	INTFLAG7	0x011C

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/64) 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.

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

Bit	Bit Symbol	Type	Function
31	NOREF	R	0: Reference clock provided 1: No reference clock
30	SKEW	R	0: Calibration value is 10 ms. 1: Calibration value is not 10 ms.
29-24	–	R	Read as 0.
23-0	TENMS	R	Calibration value

7.6.2.5 Interrupt Set-Enable Register 1

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

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

Note: <SETENA> and <CLRENA> cannot be set simultaneously. The latter setting is valid.

7.6.2.6 Interrupt Clear-Enable Register 1

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

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

Note:<SETENA> and <CLRENA> cannot be set simultaneously. The latter setting is valid.

7.6.2.7 Interrupt Set-Pending Register 1

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

Bit	Bit Symbol	Type	Function
31-0	SETPEND	R/W	Interrupt number [31:0] [Write] 1: Pend [Read] 0: Not pending 1: Pending

Note 1: <SETPEND> and <CLRPEND> cannot be set simultaneously. The latter setting is valid.

Note 2: When the external interrupt is used, the user needs special considrertion.When INTxxEN="0" (release for low-power consumption mode is unused), if IEx="0" is set, the CPU receives a "High" signal; therefore, an interrupt signal is considered to be exist.When the external interrupts are used, set IEx="1"(input is enable) to clear a suspended interrupt signal.

Note 3: 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.

7.6.2.8 Interrupt Clear-Pending Register 1

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

Bit	Bit Symbol	Type	Function
31-0	CLRPEND	R/W	Interrupt number [31:0] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

Note 1: <SETPEND> and <CLRPEND> cannot be set simultaneously. The latter setting is valid.

Note 2: When the external interrupt is used, the user needs special considrertion.When INTxxEN="0" (release for low-power consumption mode is unused), if IEx="0" is set, the CPU receives a "High" signal; therefore, an interrupt signal is considered to be exist.When the external interrupts are used, set IEx="1"(input is enable) to clear a suspended interrupt signal.

Note 3: 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

7.6.2.9 Interrupt Priority Register

The configuration of each interrupt is eight bit.

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

	31 24	23 16	15 8	7 0
0xE000_E400	PRI_3	PRI_2	PRI_1	PRI_0
0xE000_E404	PRI_7	PRI_6	PRI_5	PRI_4
0xE000_E408	PRI_11	PRI_10	PRI_9	PRI_8
0xE000_E40C	PRI_15	PRI_14	PRI_13	PRI_12
0xE000_E410	PRI_19	PRI_18	PRI_17	PRI_16
0xE000_E414	PRI_23	PRI_22	PRI_21	PRI_20
0xE000_E418	PRI_27	PRI_26	PRI_25	PRI_24
0xE000_E41C	PRI_31	PRI_30	PRI_29	PRI_28

Cortex-M0 core uses two 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
31-30	PRI_3	R/W	Priority of interrupt number 3
29-24	–	R	Read as 0,
23-22	PRI_2	R/W	Priority of interrupt number 2
21-16	–	R	Read as 0,
15-14	PRI_1	R/W	Priority of interrupt number 1
13-8	–	R	Read as 0,
7-6	PRI_0	R/W	Priority of interrupt number 0
5-0	–	R	Read as 0,

7.6.2.10 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	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	SYSRESET REQ	VECTCLR ACTIVE	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	VECTKEY(Written) / VECTKEYSTAT (Read)	R/W	Register key [Write] Writing to this register requires 0x5FA in the <VECTKEY> field. [Read] Read as 0xFA05.
15-3	−	R	read as "0".
2	SYSRESETREQ	R/W	System Reset Request 1=CPU outputs a SYSRESETREQ signal. (note1)
1	VECTCLRACTIVE	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 it the responsibility of the application to reinitialize the stack.
0	−	R	Read as "0".

Note 1: When SYSRESETREQ is output, reset is performed on this MCU. <SYSRESETREQ> is cleared by reset.

7.6.2.11 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_ED1C	PRI_11 (SVCall)	PRI_10	PRI_9	PRI_8
0xE000_ED20	PRI_15 (SysTick)	PRI_14 (PendSV)	PRI_13	PRI_12

Cortex-M0 core uses two 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_15		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	PRI_14		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	PRI_13		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PRI_12		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-30	PRI_15	R/W	Priority of SysTick
29-24	−	R	Read as "0".
23-22	PRI_14	R/W	Priority of PendSV.
21-16	−	R	Read as "0".
15-14	PRI_13	R/W	Reserved.
13-8	−	R	Read as "0".
7-6	PRI_12	R/W	Reserved.
5-0	−	R	Read as "0".

7.6.2.12 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	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SVCALL PENDED	-	-	-	-	-	-	-
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 "0".
15	SVCALLPENDED	R/W	SVCall 0: Not pending. 1: Pending.
14-0	−	R	Read as "0".

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 INTIF (Interrupt control) registers

7.6.3.1 INTCTL (interrupt control registers, for release from NMI, low-power consumption mode)

The interrupt control registers are allocated based on the factor. The table below shows the interrupt control registers of this MCU.

Management No	Address	INTIFAO Interrupt Control register (AOBUS)	INTIFSD Interrupt Control register (IOBUS)
000	X + 0x00		- - -
- - -	- - -		- - -
015	X + 0x0F		- - -
016	Y + 0x10		STOP1INT_016 or IDLEINT_016
- - -	- - -		- - -
031	Y + 01F		STOP1INT_031 or IDLEINT_031
032	X + 0x20	STOP2INT_032	- - -
- - -	- - -	- - -	- - -
095	X + 0x5F	STOP2INT_095	- - -
096	Y + 0x60		SOPT1INT_096 or IDLEINT_096
- - -	- - -		- - -
255	Y + 0xFF		STOP1INT_255 or IDLEINT_255

X = 0x4003_8000(AO bus area), Y = 0x400F_4E00 (IO bus area)

Here are the functions of STOP2INT_xxx. (xxx means the interrupt management number. xxx=000 to 015, 032 to 095).

STOP2INT_xxx

Bit No.	7	6	5	4	3	2	1	0
bit symbol	INTxxxNCLR	INTxxxPCLR	INTxxxNFLG	INTxxxPFLG	INTxxxMODE			INTxxxEN
Attribute	W	W	R	R	RW	RW	RW	RW
After reset	-	-	0	0	0	0	0	0

Bit Symbol	Function contents	When Read	When Write
INTxxxNCLR	Falling edge detection flag clear control	Read as "0"	0 :- 1: Clear Flag
INTxxxPCLR	Rising edge detection flag clear control	Read as "0".	0 : - 1 : Clear Flag
INTxxxNFLG	Fallingedge detection flag	0: The flag is not detected. 1: The flag is detected.	0: - 1: -
INTxxxPFLG	Rising edge detection flag	0: The flag is not detected. 1: The flag is detected.	0: - 1: -
INTxxxMODE	Detection level/edge selection	000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges 101: Reserved 11* : Reserved	
INTxxxEN	Interrupt control	0: Interrupt detection disable 1: Interrupt detection enable	

Note 1: The setting "Reserved" in INTxxxMODE means the state in which an interrupt is not accepted.

The function of STOP1INT_xxx register is as following.(xxx is management No of Interrupt. xxx = 016 to 031,096 to 255)

STOP1INT_xxx

Bit No.	7	6	5	4	3	2	1	0
bit symbol	INTxxxNCLR	INTxxxPCLR	INTxxxNFLG	INTxxxPFLG	INTxxxMODE			INTxxxEN
Attribute	W	W	R	R	RW	RW	RW	RW
After reset	-	-	0	0	0	0	0	0

Bit Symbol	Function contents	When Read	When Write
INTxxxNCLR	Falling edge detection flag clear control	Read as "0"	0 :- 1: Clear Flag
INTxxxPCLR	Rising edge detection flag clear control	Read as "0".	0 : - 1 : Clear Flag
INTxxxNFLG	Fallingedge detection flag	0: The flag is not detected. 1: The flag is detected.	0: - 1: -
INTxxxPFLG	Rising edge detection flag	0: The flag is not detected. 1: The flag is detected.	0: - 1: -
INTxxxMODE	Detection level/edge selection	000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges 101: Reserved 11* : Reserved	
INTxxxEN	Interrupt control	0: Interrupt detection disable 1: Interrupt detection enable	

Note 1: The setting "Reserved" in INTxxxMODE means the state in which an interrupt is not accepted.

The function of IDLEINT_xxx register is as following.(xxx is management No of Interrupt. xxx = 016 to 031,096 to 255)

IDLEINT_xxx

Bit No.	7	6	5	4	3	2	1	0
bit symbol	INTxxxNCLR	INTxxxPCLR	INTxxxNFLG	INTxxxPFLG	INTxxxMODE			INTxxxEN
Attribute	W	W	R	R	RW	RW	RW	RW
After reset	-	-	0	0	0	0	0	0

Bit Symbol	Function contents	When Read	When Write
INTxxxNCLR	Falling edge detection flag clear control	Read as "0"	0:- 1: Clear Flag
INTxxxPCLR	Rising edge detection flag clear control	Read as "0".	0: - 1: Clear Flag
INTxxxNFLG	Falling edge detection flag	0: The flag is not detected. 1: The flag is detected.	0: - 1: -
INTxxxPFLG	Rising edge detection flag	0: The flag is not detected. 1: The flag is detected.	0: - 1: -
INTxxxMODE	Detection level/edge selection	000: "Low" level 001: "High" level 010: Low pulse 011: High pulse 100: Reserved 101: Reserved 11*: Reserved	
INTxxxEN	Interrupt control	0: Interrupt detection disable 1: Interrupt detection enable	

Note 1: The setting "Reserved" in INTxxxMODE means the state in which an interrupt is not accepted.

Hear are the management numbers of STOP2INT in this MCU.

Management No	Address	Symbol	Interrupt
32	0x4003_8020	STOP2INT_032	External interrupt pin0
33	0x4003_8021	STOP2INT_033	External interrupt pin1
34	0x4003_8022	STOP2INT_034	External interrupt pin2
35	0x4003_8023	STOP2INT_035	External interrupt pin3
36	0x4003_8024	STOP2INT_036	External interrupt pin4
37	0x4003_8025	STOP2INT_037	External interrupt pin5
38	0x4003_8026	STOP2INT_038	I2C (Wake-up by address matched) interrupt
39	0x4003_8027	STOP2INT_039	USBD Wakeup interrupt.

The following table shows the Model and EN control related registers.

[STOP2INT_032]

Bit Symbol	Function contents	When Read	When Write
INT032MODE	Detection level/edge selection	000: "Low" level 001: "High" level 010: Low pulse 011: High pulse 100: Both edges 101: Reserved 11* : Reserved	
INT032EN	Interrupt control	0: Interrupt detection disable 1: Interrupt detection enable	

[STOP2INT_033]

Bit Symbol	Function contents	When Read	When Write
INT033MODE	Detection level/edge selection	000: "Low" level 001: "High" level 010: Low pulse 011: High pulse 100: Both edges 101: Reserved 11* : Reserved	
INT033EN	Interrupt control	0: Interrupt detection disable 1: Interrupt detection enable	

[STOP2INT_034]

Bit Symbol	Function contents	When Read	When Write
INT034MODE	Detection level/edge selection	000: "Low" level 001: "High" level 010: Low pulse 011: High pulse 100: Both edges 101: Reserved 11* : Reserved	
INT034EN	Interrupt control	0: Interrupt detection disable 1: Interrupt detection enable	

[STOP2INT_035]

Bit Symbol	Function contents	When Read	When Write
INT035MODE	Detection level/edge selection	000: "Low" level 001: "High" level 010: Low pulse 011: High pulse 100: Both edges 101: Reserved 11* : Reserved	
INT035EN	Interrupt control	0: Interrupt detection disable 1: Interrupt detection enable	

[STOP2INT_036]

Bit Symbol	Function contents	When Read	When Write
INT036MODE	Detection level/edge selection	000: "Low" level 001: "High" level 010: Low pulse 011: High pulse 100: Both edges 101: Reserved 11* : Reserved	
INT036EN	Interrupt control	0: Interrupt detection disable 1: Interrupt detection enable	

[STOP2INT_037]

Bit Symbol	Function contents	When Read	When Write
INT037MODE	Detection level/edge selection	000: "Low" level 001: "High" level 010: Low pulse 011: High pulse 100: Both edges 101: Reserved 11* : Reserved	
INT037EN	Interrupt control	0: Interrupt detection disable 1: Interrupt detection enable	

[STOP2INT_038]

Bit Symbol	Function contents	When Read	When Write
INT038MODE	Detection level/edge selection	000: "Low" level 001: "High" level 010: Low pulse 011: High pulse 100: Both edges 101: Reserved 11* : Reserved	
INT038EN	Interrupt control	0: Interrupt detection disable 1: Interrupt detection enable	

[STOP2INT_039]

Bit Symbol	Function contents	When Read	When Write
INT039MODE	Detection level/edge selection	000: "Low" level 001: "High" level 010: Low pulse 011: High pulse 100: Both edges 101: Reserved 11* : Reserved	
INT039EN	Interrupt control	0: Interrupt detection disable 1: Interrupt detection enable	

Hear are the management numbers of STOP1INT, IDLEINT in this MCU.

Management No	Address	Symbol	Interrupt
16	0x400F_4E10	STOP1INT_016	LVD interrupt (power-supply fall detection)
17	0x400F_4E11	STOP1INT_017	LVD interrupt (power-return detection)
18	0x400F_4E12	IDLEINT_018	WDT interrupt
96	0x400F_4E60	IDLEINT_096	DMA ch0 transmission completion
97	0x400F_4E61	IDLEINT_097	DMA ch1 transmission completion
98	0x400F_4E62	IDLEINT_098	DMA ch2 transmission completion
99	0x400F_4E63	IDLEINT_099	DMA ch3 transmission completion
100	0x400F_4E64	IDLEINT_100	DMA ch4 transmission completion
101	0x400F_4E65	IDLEINT_101	DMA ch5 transmission completion
102	0x400F_4E66	IDLEINT_102	DMA ch6 transmission completion
103	0x400F_4E67	IDLEINT_103	DMA ch7 transmission completion
104	0x400F_4E68	IDLEINT_104	DMA ch8 transmission completion
105	0x400F_4E69	IDLEINT_105	DMA ch9 transmission completion
106	0x400F_4E6A	IDLEINT_106	DMA ch10 transmission completion
107	0x400F_4E6B	IDLEINT_107	DMA ch11 transmission completion
108	0x400F_4E6C	IDLEINT_108	DMA ch12 transmission completion
109	0x400F_4E6D	IDLEINT_109	DMA ch13 transmission completion
110	0x400F_4E6E	IDLEINT_110	DMA ch14 transmission completion
111	0x400F_4E6F	IDLEINT_111	DMA ch15 transmission completion
112	0x400F_4E70	IDLEINT_112	DMA ch16 transmission completion
113	0x400F_4E71	IDLEINT_113	DMA ch17 transmission completion
114	0x400F_4E72	IDLEINT_114	DMA ch18 transmission completion
115	0x400F_4E73	IDLEINT_115	DMA ch19 transmission completion
116	0x400F_4E74	IDLEINT_116	DMA ch20 transmission completion
117	0x400F_4E75	IDLEINT_117	DMA ch21 transmission completion
118	0x400F_4E76	IDLEINT_118	DMA ch22 transmission completion
119	0x400F_4E77	IDLEINT_119	DMA ch23 transmission completion
120	0x400F_4E78	IDLEINT_120	DMA ch24 transmission completion
121	0x400F_4E79	IDLEINT_121	DMA ch25 transmission completion
122	0x400F_4E7A	IDLEINT_122	DMA ch26 transmission completion
123	0x400F_4E7B	IDLEINT_123	DMA ch27 transmission completion
124	0x400F_4E7C	IDLEINT_124	DMA ch28 transmission completion
125	0x400F_4E7D	IDLEINT_125	DMA ch29 transmission completion
126	0x400F_4E7E	IDLEINT_126	DMA ch30 transmission completion
127	0x400F_4E7F	IDLEINT_127	DMA ch31 transmission completion
128	0x400F_4E80	IDLEINT_128	DMA transmission error
129	0x400F_4E81	IDLEINT_129	TMRB ch0 match detection/Over flow
130	0x400F_4E82	IDLEINT_130	TMRB ch0 input capture0
131	0x400F_4E83	IDLEINT_131	TMRB ch0 input capture1
132	0x400F_4E84	IDLEINT_132	TMRB ch1 match detection/Over flow
133	0x400F_4E85	IDLEINT_133	TMRB ch1 input capture0
134	0x400F_4E86	IDLEINT_134	TMRB ch1 input capture1
135	0x400F_4E87	IDLEINT_135	TMRB ch2 match detection/Over flow
136	0x400F_4E88	IDLEINT_136	TMRB ch2 input capture0
137	0x400F_4E89	IDLEINT_137	TMRB ch2 input capture1
138	0x400F_4E8A	IDLEINT_138	TMRB ch3 match detection/Over flow
139	0x400F_4E8B	IDLEINT_139	TMRB ch3 input capture0
140	0x400F_4E8C	IDLEINT_140	TMRB ch3 input capture1
141	0x400F_4E8D	IDLEINT_141	TMRB ch4 match detection/Over flow
142	0x400F_4E8E	IDLEINT_142	TMRB ch4 input capture0

Management No	Address	Symbol	Interrupt
143	0x400F_4E8F	IDLEINT_143	TMRB ch4 input capture1
144	0x400F_4E90	IDLEINT_144	TMRB ch5 match detection/Over flow
145	0x400F_4E91	IDLEINT_145	TMRB ch5 input capture0
146	0x400F_4E92	IDLEINT_146	TMRB ch5 input capture1
147	0x400F_4E93	IDLEINT_147	TMRB ch6 match detection/Over flow
148	0x400F_4E94	IDLEINT_148	TMRB ch6 input capture0
149	0x400F_4E95	IDLEINT_149	TMRB ch6 input capture1
150	0x400F_4E96	IDLEINT_150	TMRB ch7 match detection/Over flow
151	0x400F_4E97	IDLEINT_151	TMRB ch7 input capture0
152	0x400F_4E98	IDLEINT_152	TMRB ch7 input capture1
153	0x400F_4E99	IDLEINT_153	No use
154	0x400F_4E9A	IDLEINT_154	
155	0x400F_4E9B	IDLEINT_155	
156	0x400F_4E9C	IDLEINT_156	
157	0x400F_4E9D	IDLEINT_157	
158	0x400F_4E9E	IDLEINT_158	
159	0x400F_4E9F	IDLEINT_159	
160	0x400F_4EA0	IDLEINT_160	TMRD match detection 00
161	0x400F_4EA1	IDLEINT_161	TMRD match detection 01
162	0x400F_4EA2	IDLEINT_162	TMRD match detection 02
163	0x400F_4EA3	IDLEINT_163	TMRD match detection 03
164	0x400F_4EA4	IDLEINT_164	TMRD match detection 04
165	0x400F_4EA5	IDLEINT_165	TMRD match detection 10
166	0x400F_4EA6	IDLEINT_166	TMRD match detection 11
167	0x400F_4EA7	IDLEINT_167	TMRD match detection 12
168	0x400F_4EA8	IDLEINT_168	TMRD match detection 13
169	0x400F_4EA9	IDLEINT_169	TMRD match detection 14

The following shows the interrupt control bit of STOP1INT_016 to 018, IDLEINT_096 to 152 and 160 to 169 in this device.

xxx = 016 to 018, 096 to 152, 160 to 169

Bit Symbol	Function contents	When Read	When Write
INTxxxNCLR	Falling edge detection flag clear control	Read as "0"	0 :- 1: Clear Flag
INTxxxPCLR	Rising edge detection flag clear control	Read as "0".	0 : - 1 : Clear Flag

Note: This device have no setting from bit0 to bit5 of each STOP1INT_xxx, IDLEINT_xxx register. the above two bits, it uses to clear the interrupt request.

To clear NMI request, it should be used the above register.

Note: Enable or Disable detection for Combined Interrupt request from the peripheral is set by a peripheral function

7.6.3.2 RSTFLG (Reset flag register)

The RSTFLG (reset flag register) and RSTFLG1 is the register to confirm the reset factor.

RSTFLG (reset flag)

Bit No.	7	6	5	4	3	2	1	0
bit symbol	-	-	LVDRSTF	-	PINRSTF	-	-	PORF
Attribute	RW	RW	RW	RW	RW	RW	RW	RW
After reset	0	0	0	0	0	0	0	1

Bit Symbol	Function contents	When Read	When Write
LVDRSTF	LVD Reset flag	0: No Reset 1: Reset from LVD	0: Clear reset flag 1: - (don't care)
PINRSTF	RESET pin flag	0: No Reset 1: Reset from RESET pin	0: Clear reset flag 1: - (don't care)
PORF	Power on reset	0: No Reset 1: Reset from Power On Reset	0: Clear reset flag 1: - (don't care)

RSTFLG1 (reset flag1)

Bit No.	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	WDTRSTF	-	SYSRSTF
Attribute	RW	RW	RW	RW	RW	RW	RW	RW
After reset	0	0	0	0	0	0	0	0

Bit Symbol	Function contents	When Read	When Write
WDTRSTF	WDT Reset flag	0: No Reset 1: Reset from WDT	0: Clear reset flag 1: - (don't care)
SYSRSTF	SYSRESETREQ RESET flag	0: No Reset 1: Reset from SYSRESETREQ(CPU)	0: Clear reset flag 1: - (don't care)

7.6.3.3 INTFLAG (Interrupt monitor flag register)

This MCU has interrupt flags to monitor the interrupt factor.

The interrupt flags are allocated to addresses according to the attribute of implemented interrupt as shown in the table below:

INTFLAG0 for NMI, INTFLAG1 and 2 in the AO area, INTFLAG3 to INTFLAG7 in the SD area are implemented in this MCU.

Address	Interrupt Flag (IOBUS)	remark	Interrupt Management Number
Y + 0x100	INTFLAG0	NMI Flag of AO/SD area (16 bits each)	0 to 31
Y + 0x104	INTFLAG1	Asynchronous interrupt of AO area	32 to 63
Y + 0x108	INTFLAG2		64 to 95
Y + 0x10C	INTFLAG3	Asynchronous interrupt of SD area.	96 to 127
Y + 0x110	INTFLAG4		128 to 159
Y + 0x114	INTFLAG5		160 to 191
Y + 0x118	INTFLAG6		192 to 223
-Y + 0x11C	INTFLAG7		224 to 255

Y = 0x400F_4E00 (SD Area, IO bus area)

The following table shows the details of INTFLAG0

INTFLAG0(NMI)

	31	30	29	28	27	26	25	24
bit symbol	INT31FLG	INT30FLG	INT29FLG	INT28FLG	INT27FLG	INT26FLG	INT25FLG	INT24FLG
After reset	-	-	-	-	-	-	-	-
	23	22	21	20	19	18	17	16
bit symbol	INT23FLG	INT22FLG	INT21FLG	INT20FLG	INT19FLG	INT18FLG	INT17FLG	INT16FLG
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	INT15FLG	INT14FLG	INT13FLG	INT12FLG	INT11FLG-	INT10FLG	INT9FLG-	INT8FLG
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	INT7FLG	INT6FLG	INT5FLG	INT4FLG	INT3FLG	INT2FLG	INT1FLG	INT0FLG
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-19	INT31FLG − INT19FLG	R	0: Interrupt is not detected (Both INTxNFLG and INTxPFLG is "0" 1: Interrupt is detected. (INTxNFLG or INTxPFLG is "1")
18	INT18FLG	R	WDT interrupt detection flag 0: Interrupt is not detected. 1: Interrupt is detected.
17	INT17FLG	R	LVD interrupt detection flag (When the supply voltage falls under or over the set detection voltage) 0: Interrupt is not detected. 1: Interrupt is detected.
16	INT16FLG	R	LVD interrupt detection flag (When the supply voltage falls under the set detection voltage) 0: Interrupt is not detected. 1: Interrupt is detected.
15-0	INT15FLG − INT0FLG	R	0: Interrupt is not detected (Both INTxNFLG and INTxPFLG is "0" 1: Interrupt is detected. (INTxNFLG or INTxPFLG is "1")

Here is the configuration of INTFLAG1."xxx" of INTxxxFLG means the interrupt management number described in the chapter "7.5.1.8 Interrupt management numbers".

INTFLAG1(AO)

	31	30	29	28	27	26	25	24
bit symbol	INT63FLG	INT62FLG	INT61FLG	INT60FLG	INT59FLG	INT58FLG	INT57FLG	INT56FLG
After reset	-	-	-	-	-	-	-	-
	23	22	21	20	19	18	17	16
bit symbol	INT55FLG	INT54FLG	INT53FLG	INT52FLG	INT51FLG	INT50FLG	INT49FLG	INT48FLG
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	INT47FLG	INT46FLG	INT45FLG	INT44FLG	INT43FLG	INT42FLG	INT41FLG	INT40FLG
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	INT39FLG	INT38FLG	INT37FLG	INT36FLG	INT35FLG	INT34FLG	INT33FLG	INT32FLG
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	INT63FLG - INT32FLG	R	0: Interrupt is not detected (Both INTxNFLG and INTxPFLG is "0" 1: Interrupt is detected. (INTxNFLG or INTxPFLG is "1")

Note: This MCU uses No.32 to 39 of INTFLAG1.

Here is the configuration of INTFLAG3 to INTFLAG5. "xxx" of INTxxxFLG means the interrupt management number described in the chapter "7.5.1.8 Interrupt management numbers".

This MCU uses NO.96 to 152 of INTFLAG3 and INTFLAG4, and No.160 to 169 of INTFLAG5

INTFLAG3(SD)

	31	30	29	28	27	26	25	24
bit symbol	INT127FLG	INT126FLG	INT125FLG	INT124FLG	INT123FLG	INT122FLG	INT121FLG	INT120FLG
After reset	-	-	-	-	-	-	-	-
	23	22	21	20	19	18	17	16
bit symbol	INT119FLG	INT118FLG	INT117FLG	INT116FLG	INT115FLG	INT114FLG	INT113FLG	INT112FLG
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	INT111FLG	INT110FLG	INT109FLG	INT108FLG	INT107FLG	INT106FLG	INT105FLG	INT104FLG
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	INT103FLG	INT102FLG	INT101FLG	INT100FLG	INT99FLG	INT98FLG	INT97FLG	INT96FLG
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	INT127FLG - INT96FLG	R	0: Interrupt is not detected (Both INTxNFLG and INTxPFLG is "0" 1: Interrupt is detected. (INTxNFLG or INTxPFLG is "1")

INTFLAG4(SD)

	31	30	29	28	27	26	25	24
bit symbol	INT159FLG	INT158FLG	INT157FLG	INT156FLG	INT155FLG	INT154FLG	INT153FLG	INT152FLG
After reset	-	-	-	-	-	-	-	-
	23	22	21	20	19	18	17	16
bit symbol	INT151FLG	INT150FLG	INT149FLG	INT148FLG	INT147FLG	INT146FLG	INT145FLG	INT144FLG
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	INT143FLG	INT142FLG	INT141FLG	INT140FLG	INT139FLG	INT138FLG	INT137FLG	INT136FLG
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	INT135FLG	INT134FLG	INT133FLG	INT132FLG	INT131FLG	INT130FLG	INT129FLG	INT128FLG
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	INT159FLG - INT128FLG	R	0: Interrupt is not detected (Both INTxNFLG and INTxPFLG is "0" 1: Interrupt is detected. (INTxNFLG or INTxPFLG is "1")

INTFLAG5(SD)

	31	30	29	28	27	26	25	24
bit symbol	INT191FLG	INT190FLG	INT189FLG	INT188FLG	INT187FLG	INT186FLG	INT185FLG	INT184FLG
After reset	-	-	-	-	-	-	-	-
	23	22	21	20	19	18	17	16
bit symbol	INT183FLG	INT182FLG	INT181FLG	INT180FLG	INT179FLG	INT178FLG	INT177FLG	INT176FLG
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	INT175FLG	INT174FLG	INT173FLG	INT172FLG	INT171FLG	INT170FLG	INT169FLG	INT168FLG
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	INT167FLG	INT166FLG	INT165FLG	INT164FLG	INT163FLG	INT162FLG	INT161FLG	INT160FLG
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	INT191FLG - INT160FLG	R	0: Interrupt is not detected (Both INTxNFLG and INTxPFLG is "0" 1: Interrupt is detected. (INTxNFLG or INTxPFLG is "1")

8. µDMA Controller (µDMAC)

8.1 Overview

8.1.1 Function List

The main functions per unit are shown as below:

For the information on the start trigger of peripheral functions, refer to the chapter on "Product Information."

Table 8-1 µDMA outline (Per unit)

Functions	Features		Descriptions
Channels	32 channels		-
Start trigger	Start by Hardware		DMA requests from peripheral functions
	Start by Software		Specified by the DMAxChnlSwRequest register
Priority	Between channels	ch0 (high priority) > … > ch31 (high priority) > ch0 (Normal priority) > … > ch31 (Normal priority)	High-priority can be configured by the DMAxChnl-PrioritySet register
Transfer data size	8/16/32 bits		-
The number of transfer	1 to 1024 times		-
Address	Transfer source address	Increment / fixed	Transfer source address and destination address can be selected from the increment setting or fixed setting.
	Transfer destination address	Increment / fixed	
Endian	Little-endian		-
Interrupt function	Transfer completion interrupt		Output for each channel
	Error interrupt		
Operation mode	Basic mode Automatic request mode Ping-pong mode Memory scatter/gather mode Peripheral scatter/gather mode		-

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.

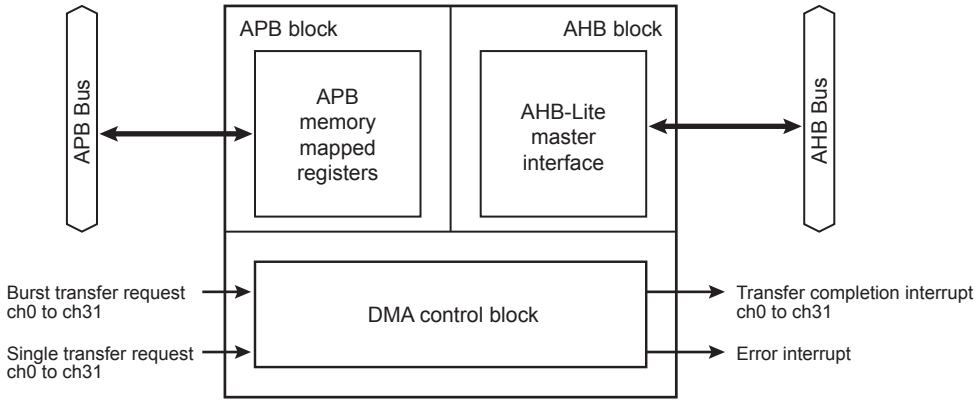


Figure 8-1 µDMA block diagram

8.3 Registers

8.3.1 Register List

The following table shows control registers and addresses:

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

Then name of peripheral: DMA

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
Channel software request status 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
Bus error clear register	DMAxErrClr	0x004C

Note: Access the registers in units of words (32 bits).

8.3.2 DMAxStatus (DMAC 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 "0".
28	-	R	Read as "1".
27-21	-	R	Read as "0".
20-16	-	R	Read as "1".
15-8	-	R	Read as "0".
7-4	-	R	Read as an undefined value.
3-1	-	R	Read as "0".
0	master_enable	R	DMA operation 0: Disabled 1: Enabled

8.3.3 DMAxCfg (DMAC 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 as "0".
0	master_ enable	W	DMA operation 0: Disabled 1: Enabled

Note: After DMAxCfg = 0x00000001, DMAxChnlReqMaskSet = 0xFFFFFFFF and DMAxChnlEnableSet = 0xFFFFFFFF are set, set "1" to the corresponding bit of DMAxChanlReqMaskClr to release masking of the channel to be used.

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 data base-pointer Specifies the base address of the primary data.
9-0	-	R	Read as "0".

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: A transfer request does not occur. 1: A transfer request occurs. 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 is disabled [Write] 1: Single-transfer is disabled. [Read] 0: Single-transfer is enabled. 1: Single-transfer is disabled. Each bit corresponds to the channels in the specified number. Writing "1" disables the single-transfer to the corresponding channel, and only burst transfer request becomes valid. Writing "0" has no meaning. Set the DMAxChnlUseburstClr register in order to cancel the disabled the single-transfer. By reading the bit, the channel state of the corresponding bit can be checked whether it is enabled or disabled. Bits are automatically set in the following conditions: ・ This bit is cleared to "0", 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 on the condition where the number of transfers 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 is enabled. 1: Enables the single-transfer. Each bit corresponds to the channels in the specified number. Writing "1" enables the single-transfer to the corresponding channel. Writing "0" has no meaning. To disable or confirm the 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 masking [Write] 1: Mask a DMA request [Read] 0: A DMA request is valid. 1: A DMA request is invalid. Each bit corresponds to the channels in the specified number. Writing "1" disables the single-transfer for the corresponding channel. Writing "0" has no meaning. To disable masking, configure the DMAxChnlReqMaskClr register. By reading the bit, the status of the DMA request setting can be checked whether it is enabled or disabled.

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 the specified number. Writing "1" disables the DMA request mask setting of the corresponding channel. Writing "0" has no meaning. Configure the DMAxChnlReqMaskSet register to enable and confirm the setting.

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 channel. [Read] 0: The corresponding bit is invalid. 1: The corresponding bit is valid. Each bit corresponds to the channels in the specified number. Writing "1" enables the corresponding channels. Writing "0" has no meaning. To disable the setting, configure the DMAxChnlEnableClr register. By reading the bit, the corresponding channel can be checked whether it is enabled or disabled. In the following conditions, the function automatically becomes invalid. ・DMA cycle ends. ・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 channel. Each bit corresponds to the channels in the specified number. Writing "1" disables the corresponding channel. Writing "0" has no meaning. Configure the DMAxChnlEnableSet register in order to enable and confirm the setting. In the following conditions, the function automatically becomes invalid. ・ DMA cycle ends. ・ 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 the specified number. Writing "1" specifies the data that is firstly used in the corresponding channel as "alternative data". Writing "0" has no meaning. To disable this bit, use the DMAxChnlEnableClr register. Only in basic mode, automatic request mode, and ping-pong mode the first data can be specified as alternative data. When this bit is read, data of the corresponding channel can be checked whether data is primary data or alternative data. In the following conditions, 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 the specified number. 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 setting. In the following conditions, the setting is automatically changed. ・ The primary data transfer in memory scatter/gather mode or peripheral scatter/gather mode is complete. ・ The primary data transfer in ping-pong mode is complete. ・ The alternative transfer in ping-pong mode, memory scatter/gather mode, or peripheral scatter/gather mode is complete.

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 the specified number. Writing "1" sets the priority of the corresponding channel high. Writing "0" has no meaning.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 the specified number. Writing "1" changes the priority of the corresponding channel to normal priority. Writing "0" has no meaning. 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 "0".
0	err_clr	R/W	Bus error [Write] 1: Clears a bus error. [Read] 0: No bus error 1: The state of a bus error A bus error occurrence can be confirmed by reading the bit. Writing "1" clears a bus error. Writing "0" has no meaning.

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 types 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 example of memory map of the channel control data.

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

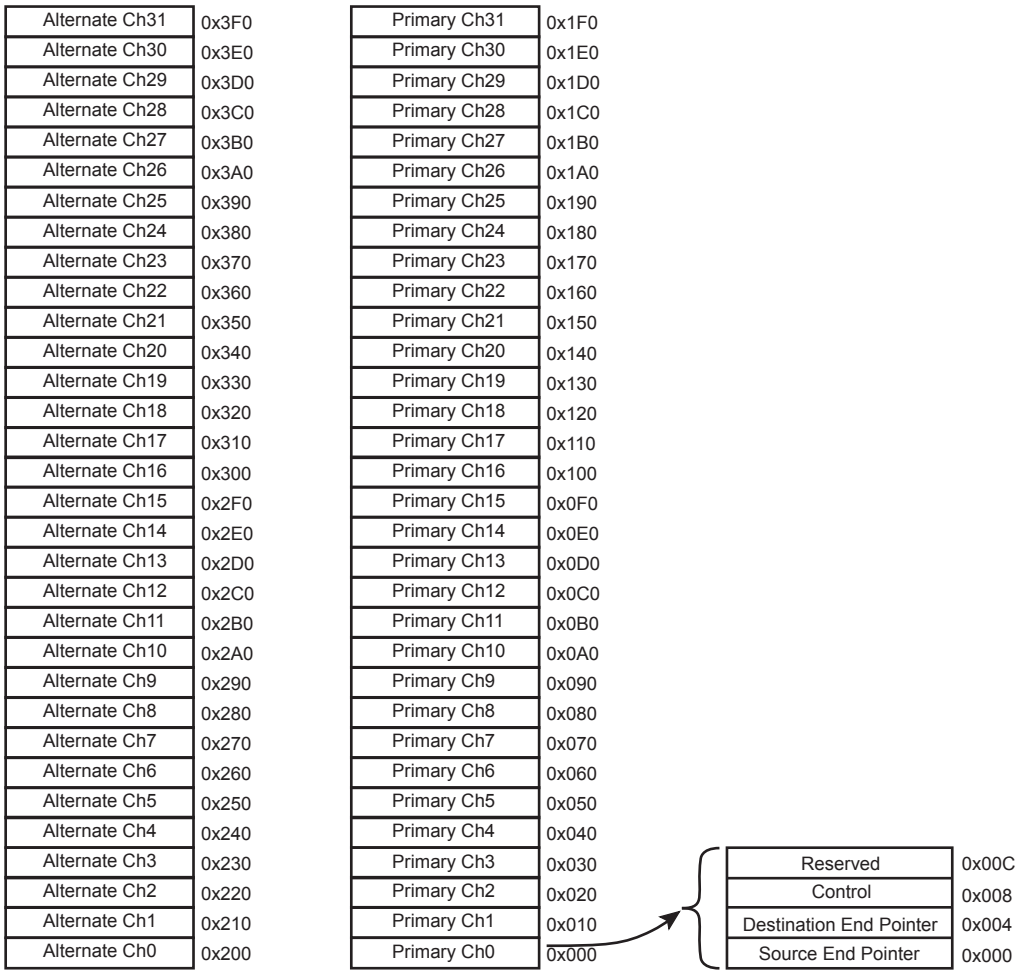


Figure 8-2 Memory map of the control data

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 8-2 shows the relationship between the number of channels and addresses.

Table 8-2 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

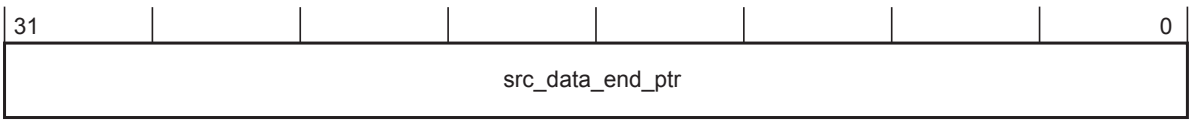
The channel control data contains the three kinds of data shown below:

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

Each data is described in the following sections:

8.4.2.1 Final Address of the Transfer Source Data

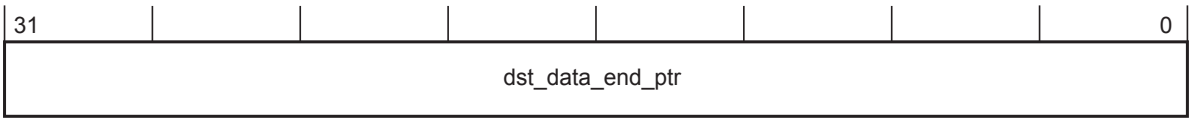
Specify the final address of the data to be transferred. The alignment of an address should be adjusted to a transfer data size. The DMA calculates the start address of the source address using this data.



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

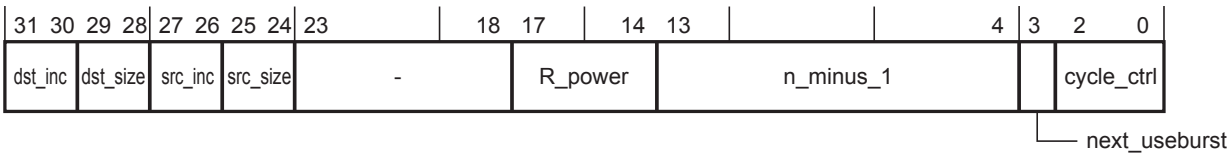
8.4.2.2 Final Address of the Transfer Destination Address

Specify the final address of the destination address. The alignment of an address should be adjusted to a transfer data size. The DMA calculates the start address of the destination address of the transfer destination address.



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

8.4.2.3 Control Data Setting



bit	Bit symbol	Function
[31:30]	dst_inc	Increments the transfer destination address (note 2) 00: 1 byte 01: 2 bytes 10: 4 bytes 11: No 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	Increments the transfer source address (note 2) 00: 1 byte 01: 2 bytes 10: 4 bytes 11: No 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".

bit	Bit symbol	Function
[17:14]	R_power	Arbitration 0000: After 1 transfer 0001: After 2 transfers 0010: After 4 transfers 0011: After 8 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: No arbitration A transfer request is checked after the specified number of transfers. If a higher-priority request exists, the controller arbitrates the DMA transfer.
[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. The 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 (1 byte)	01 (2 bytes)	10 (4 bytes)
00 (1byte)	o	-	-
01 (2bytes)	o	o	-
10 (4bytes)	o	o	o
No increment	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 invalid data is read either in ping-pong mode, memory scatter / gather mode or peripheral scatter / gather mode.

8.4.3.2 Basic Mode

In basic mode, data structure can be selected from primary data or alternative data.

A transfer is started 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 completion interrupt occurs.

8.4.3.3 Automatic Request Mode

In this mode, a single-transfer request stops the DMA transfer. The data structure can be selected from primary data or alternative data.

The DMA transfer is started by a transfer request.

In each transfer 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 completion interrupt occurs.

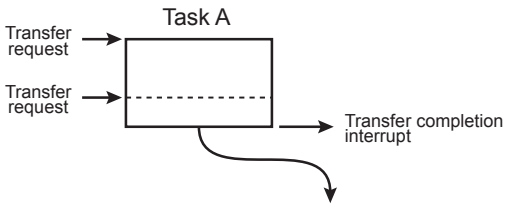
8.4.3.4 Ping-pong Mode

In ping-pong mode, a continuous DMA transfer that uses primary data and alternative data alternately is performed. If <cycle_ctrl> reads data specified to be invalid ("000"), or the channel is specified to be invalid, the transfer is stopped. Every time a DMA transfer (task) that uses primary data or alternative data is complete, a transfer completion interrupt occurs.

Preparation:

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

Task A: Primary data
<cycle_ctrl[2:0]> = "011"
(ping-pong mode)
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> = "0x005" (6 times)



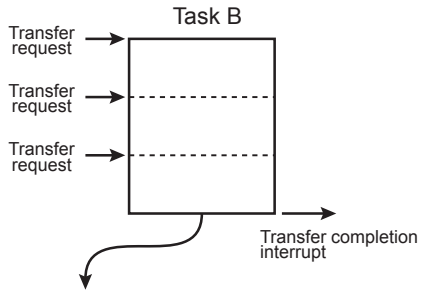
Receiving a transfer request, the DMA performs a transfer four times and performs arbitration.

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

The DMA generates a transfer completion interrupt request and performs an arbitration.

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

Task B: Alternative data
<cycle_ctrl[2:0]> = "011"
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> = "0x00B" (12 times)



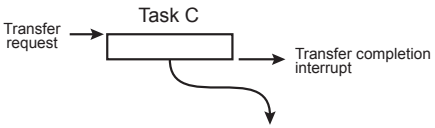
Receiving a transfer request, The DMA performs a transfer four times and performs arbitration.

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

The DMA generates a transfer completion interrupt request and performs an arbitration.

After completing Task B, alternative data for 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]> = "0x001" (2 times)

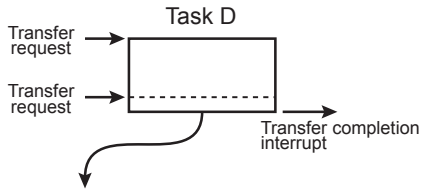


Receiving a transfer request, the DMA performs a transfer twice and performs arbitration.

The DMA generates a transfer completion interrupt request and performs an arbitration.

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

Task D: Alternative data
<cycle_ctrl[2:0]> = "011"
<R_power[4:0]> = "0010"
(4 times)
<n_minus_1[9:0]> = "0x004" (5 times)

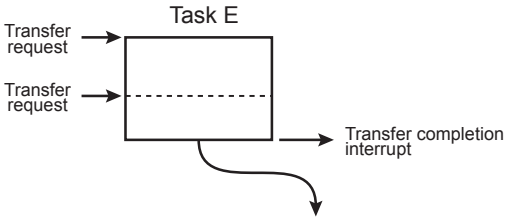


Receiving a transfer request, the DMA performs a transfer four times and performs arbitration.

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

The DMA generates a transfer completion 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]> = "0x006" (7 times)

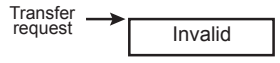


Receiving a transfer request, the DMA performs a transfer four times and performs arbitration.

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

The DMA generates a transfer completion interrupt request and performs an arbitration.

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



Even receiving a transfer request, the operation stops because <cycle_ctrl[2:0]> is set to invalid.

(The operation can be also stopped by setting the <cycle_ctrl[2:0]> of 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 [2:0]> 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 8-3 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 the transfers specified in the <n_minus_1> are complete, invalid data "000" is automatically set.

Preparation:

Prepare primary data. Set "100" to <cycle_ctrl> and set four task data 4 × 4 = 16 as the number of transfers <n_minus_1>.

Set alternative data for 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 DMAxCfg <master_enable> and DMAxChnlSet.

Copy A: Primary data

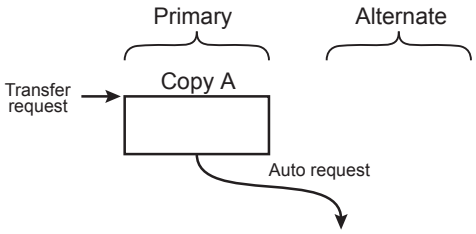
<cycle_ctrl[2:0]> = "100"

(Memory scatter / gather mode)

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

(4 times)

<n_minus_1[9:0]> = "0x00F" (16 times)



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

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

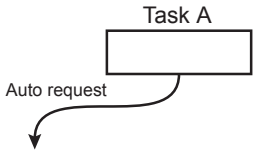
Task A: Alternative data

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

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

(4 times)

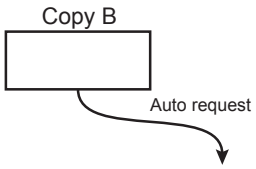
<n_minus_1[9:0]> = "0x002" (3 times)



The DMA performs Task A.

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

Copy B: Primary data



The DMA performs transfers for alternative data of Task B for four times.

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

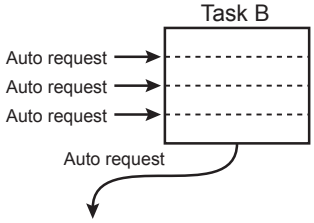
Task B: Alternative data

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

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

(2 times)

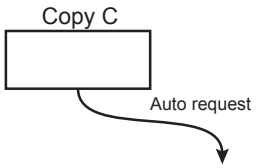
<n_minus_1[9:0]> = "0x007" (8 times)



The DMA performs Task B.

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

Copy C: Primary data



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

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

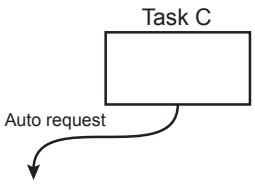
Task C: Alternative data

<cycle_ctrl[2:0> = "100"

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

(8 times)

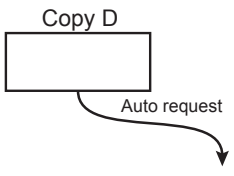
<n_minus_1[9:0]> = "0x004" (5 times)



The DMA performs Task C.

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

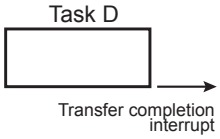
Copy D: Primary data



The DMA performs transfers for alternative data of Task D for four times. The 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_ctr[2:0]l> = "001"
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> =
"0x003" (4 times)



The DMA performs Task D.
Since <cycle_ctrl[2:0]> is set to the basic mode "001", the DMA generates a transfer completion 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 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 8-4 Fixed values in peripheral scatter / gather mode (Primary data)

Bit	Bit symbol	Setting value	Description
[31:30]	dst_inc	10	A 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	A 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.
[2:0]	cycle_ctrl	110	Specify peripheral scatter/gather mode (Primary data).

Note: If the transfers specified in the <n_minus_1> are complete, invalid data "000" is automatically set.

Preparation:

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

Set alternative data for 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 DMAxCfg <master_enable> and DMAxChnlEnableSet.

Copy A: Primary data

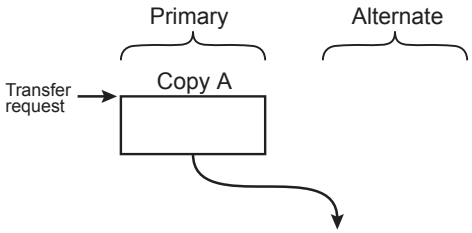
<cycle_ctrl[2:0]> = "110"

(Peripheral scatter / gather)

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

(4 times)

<n_minus_1[9:0]> = "0x00F" (16 times)



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

After completing the transfer, operation automatically moves onto Task A.

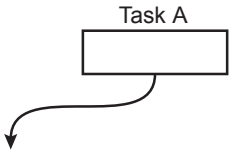
Task A: Alternative data

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

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

(4 times)

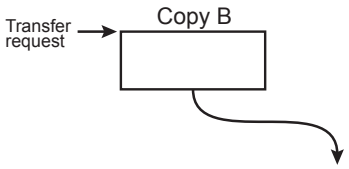
<n_minus_1[9:0]> = "0x002" (3 times)



The DMA performs 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



The DMA performs transfers for alternative data of Task B for four times.

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

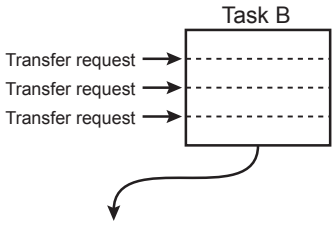
Task B: Alternative data

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

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

(2 times)

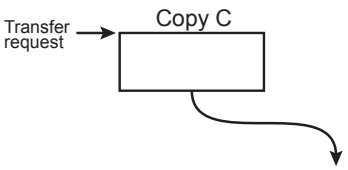
<n_minus_1[9:0]> = "0x007" (8 times)



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



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

After completing the transfer, operation automatically moves onto Task C.

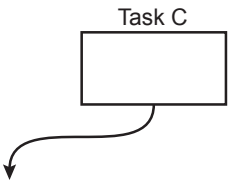
Task C: Alternative data

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

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

(8 times)

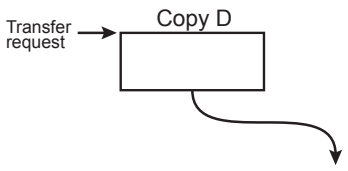
<n_minus_1[9:0]> = "0x004" (5 times)



The DMA performs 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.

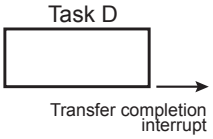
Copy D: Primary data



The DMA performs transfers for alternative data of Task D for four times. Also, The 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 onto Task D.

Task D: Alternative data
<cycle_ctrl[2:0]> = "001"
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> =
"0x003" (4 times)



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

8.5 Precautions

Extra caution should be exercised when a DMA transfer request is used in the following peripheral functions:

- Serial channel with 4-byte FIFO (SIO/UART)
- 16-bit timer/event counter (TMRB)
- Analog-to-digital converter (ADC)

8.5.1 SIO/UART, TMRB, ADC are Used

The following points should be considered:

- It is recommended to use the basic mode as a transfer mode.
- Set "after 1 transfer" as a DMA transfer rate.

  Specify "0000" as the arbitration rate <R_power> for the control data.
- Do not use the FIFO of the SIO/UART.

  Use the SIO/UART with the single-buffer or double-buffer setting.

A new request occurs after the DMA transfer is started, only one transfer is performed. Design the program to perform a DMA transfer surely.

In case that transfer will not be started, the following circumstances can be expected:

- A higher priority transfer request occurs in the same unit
- A transfer destination conflict occurs between other higher bus master and a sender.

As a guide, this µDMA controller takes 11 clocks on pre-/post-processing. It takes approximately 5 clocks for a data transfer between the peripheral functions and internal RAM.

9. Input/Output port

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

9.1 Registers

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

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

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

Register Name		Setting Value	
PxDATA	Data register	0 or 1	This register reads / writes port data.
PxCR	Output control register	0: Output Disable 1: Output enable	This register controls output.
PxFRn	Function register n	0: PORT 1: Function	This register sets the function. The assigned function can be enabled by setting "1". This register exists for the each function assigned to the port. In case of having some function, only one function can be enabled.
PxOD	Open-drain control register	0: CMOS 1: Open-drain	This register controls programmable open-drain outputs. Programmable open-drain outputs are set with PxOD.When output data is "1", output buffer is disabled and becomes a pseudo-open-drain output.
PxPUP	Pull-up control register	0: Pull-up Disable 1: Pull-up Enable	This register controls programmable pull-ups.
PxPDN	Pull-down control register	0: Pull-down Disable 1: Pull-down Enable	This register controls programmable pull-downs.
PxSEL	Input Voltage control register	0: 3V Input 1: 1.8V Input	This register controls the Input voltage range.
PxIE	Input control register	0: Input Disable 1: Input Enable	This register controls inputs. Some time is required after enabling PxIE until external data is reflected in PxDATA.

9.1.1 Register list

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

A bit without function is read as "0" and the writing a data to this bit is invalid

TMPM066FWUG(QFP64)		o	o	o	o	o
TMPM067FWQG(QFN48)		o	×	o	o	o
TMPM068FWXBG(BGA57)		o	o	o	o	o
Register name	Address (Base+)	Port A	Port B	Port C	Port D	Port E
Data register	0x0000	PADATA	PBDATA	PCDATA	PDDATA	PEDATA
Output control register	0x0004	PACR	PBCR	PCCR	PDCR	PECR
Function register 1	0x0008	PAFR1	-	PCFR1	PDFR1	PEFR1
Function register 2	0x000C	-	-	-	-	PEFR2
Open-drain control register	0x0028	PAOD	PBOD	PCOD	PDOD	PEOD
Pull-up control register	0x002C	PAPUP	PBPUP	PCPUP	PDPUP	PEPUP
Pull-down control register	0x0030	PAPDN	PBPDN	PCPDN	PDPDN	PEPDN
Input Voltage control register	0x0034	-	-	PCSEL	PDSEL	-
Input control register	0x0038	PAIE	PBIE	PCIE	PDIE	PEIE

TMPM066FWUG(QFP64)		o	o	o	o
TMPM067FWQG(QFN48)		o	o	×	×
TMPM068FWXBG(BGA57)		o	o	o	o
Register name	Address (Base+)	Port F	Port G	Port H (note1)	Port J (note1)
Data register	0x0000	PFDATA	PGDATA	PHDATA	PJDATA
Output control register	0x0004	PFCR	PGCR	PHCR	PJCR
Function register 1	0x0008	PFFR1	PGFR1	PHFR1	PJFR1
Function register 2	0x000C	PFFR2	-	-	-
Open-drain control register	0x0028	PFOD	PGOD	PHOD	PJOD
Pull-up control register	0x002C	PFPUP	PGPUP	PHPUP	PJPUP
Pull-down control register	0x0030	PFPDN	PGPDN	PHPDN	PJPDN
Input Voltage control register	0x0034	-	PGSEL	-	-
Input control register	0x0038	PFIE	PGIE	PHIE	PJIE

Note 1: To use the port H/J , it should be set clock supply enable bit in the CGFSYSENA register.

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

Note: For pin assignment of each product, refer to the "Table 1-5 Pin names and functions <Sorted by Port>".

9.1.2 Port function and setting list

The list of the function and setting register for each port is shown bellows.

- "Table 9-1 Port A Setting List"
- "Table 9-2 Port B Setting List"
- "Table 9-3 Port C Setting List"
- "Table 9-4 Port D Setting List"
- "Table 9-5 Port E Setting List"
- "Table 9-6 Port F Setting List"
- "Table 9-7 Port G Setting List"
- "Table 9-8 Port H Setting List"
- "Table 9-9 Port J Setting List"

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

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

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

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

9.1.2.1 PORT A

Table 9-1 Port A Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PADATA	PACR	PAFRn	PAOD	PAPUP	PAPDN	PAIE
PA0	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	AIN0	Input	FT5	0/1	0		0/1	0	0	0
PA1	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	AIN1	Input	FT5	0/1	0		0/1	0	0	0
PA2	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	AIN2	Input	FT5	0/1	0		0/1	0	0	0
PA3	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	AIN3	Input	FT5	0/1	0		0/1	0	0	0
PA4	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	AIN4	Input	FT5	0/1	0		0/1	0	0	0
PA5	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	AIN5	Input	FT5	0/1	0		0/1	0	0	0
	INT1	INput	FT4	0/1	0		0/1	0/1	0/1	1
PA6	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	AIN6	Input	FT5	0/1	0		0/1	0	0	0
	INT2	INPUT	FT4	0/1	0		0/1	0/1	0/1	1
PA7	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	AIN7	Input	FT5	0/1	0		0/1	0	0	0
	TBIN1	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1

9.1.2.2 PORT B

Table 9-2 Port B Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PBDATA	PBCR	PBFRn	PBOD	PBPUP	PBPDN	PBIE
PB0	After reset		FT1	0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
PB1	After reset		FT1	0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
PB2	After reset		FT1	0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
PB3	After reset		FT1	0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0

9.1.2.3 PORT C

Table 9-3 Port C Setting List

PORT	Reset status	Input/ Output	PORT Type	Control registers							
				PCDATA	PCCR	PCFRn	PCOD	PCPUP	PCPDN	PCSEL	PCIE
PC0	After reset			0	0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1		0
	I2C0SCL	I/O	FT1	0/1	1	PCFR1	1	0/1	0/1	0/1	1
PC1	After reset			0	0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1		0
	I2C0SDA	I/O	FT1	0/1	1	PCFR1	1	0/1	0/1	0/1	1
PC2	After reset			0	0	0	0	0	0		0
	Input Port	Input		0/1	0		0/1	0/1	0/1		1
	Output Port	Output		0/1	1		0/1	0/1	0/1		0
	SC0TXD	Output	FT1	0/1	1	PCFR1	0/1	0/1	0/1		0
PC3	After reset			0	0	0	0	0	0		0
	Input Port	Input		0/1	0		0/1	0/1	0/1		1
	Output Port	Output		0/1	1		0/1	0/1	0/1		0
	SC0RXD	Output	FT1	0/1	1	PCFR1	0/1	0/1	0/1		0
PC4	After reset			0	0	0	0	0	0		0
	Input Port	Input		0/1	0		0/1	0/1	0/1		1
	Output Port	Output		0/1	1		0/1	0/1	0/1		0
	SC0SCK	Output	FT1	0/1	1	PCFR1	0/1	0/1	0/1		0
PC5	After reset			0	0	0	0	0	0		0
	Input Port	Input		0/1	0		0/1	0/1	0/1		1
	Output Port	Output		0/1	1		0/1	0/1	0/1		0
	VBUS	Input		0/1	0		0/1	0/1	0/1		1
	INT4	Input	FT4	0/1	0		0/1	0/1	0/1		1

Table 9-3 Port C Setting List

PORT	Reset status	Input/ Output	PORT Type	Control registers							
				PCDATA	PCCR	PCFRn	PCOD	PCPUP	PCPDN	PCSEL	PCIE
	TB0IN	Input	FT1	0/1	0	PCFR1	0/1	0/1	0/1		1

9.1.2.4 PORT D

Table 9-4 Port D Setting List

PORT	Reset status	Input/ Output	PORT Type	Control register							
				PDDATA	PDCR	PDFRn	PDOD	PDPUP	PDPDN	PDSEL	PDIE
PD0	After reset			0	0	0	0	0	0		0
	Input Port	Input		0/1	0		0/1	0/1	0/1		1
	Output Port	Output		0/1	1		0/1	0/1	0/1		0
	T16A3OUT	Output		0/1	1	PDFR1	0/1	0/1	0/1		0
	TDA0OUT0	Output		0/1	1	PDFR2	0/1	0/1	0/1		0
PD1	After reset			0	0	0	0	0	0		0
	Input Port	Input		0/1	0		0/1	0/1	0/1		1
	Output Port	Output		0/1	1		0/1	0/1	0/1		0
	TB0OUT	Output		0/1	1	PDFR1	0/1	0/1	0/1		0
	TDA0OUT1	Output		0/1	1	PDFR2	0/1	0/1	0/1		0
PD2	After reset			0	0	0	0	0	0		0
	Input Port	Input		0/1	0		0/1	0/1	0/1		1
	Output Port	Output		0/1	1		0/1	0/1	0/1		0
	TB1OUT	Output		0/1	1	PDFR1	0/1	0/1	0/1		0
	TDA1OUT0	Output		0/1	1	PDFR2	0/1	0/1	0/1		0
PD3	After reset			0	0	0	0	0	0		0
	Input Port	Input		0/1	0		0/1	0/1	0/1		1
	Output Port	Output		0/1	1		0/1	0/1	0/1		0
	TB2OUT	Output		0/1	1	PDFR1	0/1	0/1	0/1		0
	TDA1OUT1	Output		0/1	1	PDFR2	0/1	0/1	0/1		0
PD4	After reset			0	0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1		0
	TB3IN	Input	FT1	0/1	0	PDFR1	0/1	0/1	0/1	0/1	1
PD5	After reset			0	0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1		0
	INT0	Input	FT4	0/1	0		0/1	0/1	0/1	0/1	

9.1.2.5 PORT E

Table 9-5 Port E Setting List

PORT	Reset status	Input/Output	PORT Type	Control register						
				PEDATA	PECR	PEFRn	PEOD	PEPUP	PEPDN	PEIE
PE0	After reset			0	0	0	0	0	0	0

Table 9-5 Port E Setting List

PORT	Reset status	Input/Output	PORT Type	Control register						
				PEDATA	PECR	PEFRn	PEOD	PEPUP	PEPDN	PEIE
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	SC1SCK	Input	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
PE1	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	SC1RXD	Input	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
	TB2IN	Input		0/1	0	PEFR2	0/1	0/1	0/1	1
PE2	After reset			0	1	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	SC1RXD	Input	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
PE3	After reset			0	1	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
PE4	After reset			0	0	PEFR1	0	0	1	1
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	SWCLK	Input	FT1	0/1	1	PEFR1	0/1	0/1	0/1	1
PE5	After reset			0	0	PEFR1	0	1	0	1
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	SWDIO	I/O	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1

Note: PE3 works as a BOOT function. It is enabled to be input and pulled-up while $\overline{\text{RESET}}$ pin is "Low". At the rising edge of the reset signal, if PE3 is "High", the device enters single chip mode and boots from the on-chip flash memory. If PE3 is "Low", the device enters single BOOT mode and boots from the internal BOOT program.

9.1.2.6 PORT F

Table 9-6 Port F Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PFDATA	PFCR	PFFRn	PFOD	PFPUP	PFPDN	PFIE
PF0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	INT5	Input	FT1	0/1	0		0/1	0/1	0/1	1
	TSPICSO	Output		0/1	1	PFFR1	0/1	0/1	0/1	0
	TSPICSIN	Input		0/1	0	PFFR2	0/1	0/1	0/1	1
PF1	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	INT3	Input	FT1	0/1	0		0/1	0/1	0/1	1
	TSPITXD	Output		0/1	1	PFFR1	0/1	0/1	0/1	0

Table 9-6 Port F Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PFDATA	PFCR	PFFRn	PFOD	PFPUP	PFPDN	PFIE
PF2	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	TSPIRXD	Input	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
PF3	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	TSPISCK	I/O	FT1	0/1	1	PFFR1	0/1	0/1	0/1	0
PF4	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	TB3OUT	Output	FT1	0/1	1	PFFR1	0/1	0/1	0/1	0
PF5	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	TB41OUT	Output	FT1	0/1	1	PFFR1	0/1	0/1	0/1	0
PF6	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
PF7	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0

9.1.2.7 PORT G

Table 9-7 Port G Setting List

PORT	Reset status	Input/ Output	PORT Type	Control registers							
				PGDATA	PGCR	PGFRn	PGOD	PGPUP	PGPDN	PGSEL	PGIE
PG0	After reset			0	0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	-	0
	I2C1SCL	I/O	FT1	0/1	1	PGFR1	0/1	0/1	0/1	0/1	1
PG1	After reset			0	0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	-	0
	I2C1SDA	I/O	FT1	0/1	1	PGFR1	0/1	0/1	0/1	0/1	1

9.1.2.8 PORT H

Table 9-8 Port H Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PGDATA	PGCR	PGFRn	PGOD	PGPUP	PGPDN	PGIE
PH0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	TB4IN	Input	FT1	0/1	0	PHFR1	0/1	0/1	0/1	1
PH1	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	TB5IN	Input	FT1	0/1	0	PHFR1	0/1	0/1	0/1	1
PH2	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	TB6IN	Input	FT1	0/1	0	PGFR1	0/1	0/1	0/1	1
PH3	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	TB7IN	Input	FT1	0/1	0	PHFR1	0/1	0/1	0/1	1

9.1.2.9 PORT J

Table 9-9 Port J Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PJDATA	PJCR	PJFRn	PJOD	PJPUP	PJPDN	PJIE
PJ0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	TB5OUT	Output	FT1	0/1	1	PJFR1	0/1	0/1	0/1	0
PJ1	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	TB6OUT	Output	FT1	0/1	1	PJFR1	0/1	0/1	0/1	0
PJ2	After reset		FT1	0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
PJ3	After reset		FT1	0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0

9.1.3 Block Diagrams of Ports

9.1.3.1 Port Type

The ports are classified as shown below. Please refer to the following pages for the block diagrams of each port type. A dotted box in the figure indicates the part of the equivalent circuit described in the "Block diagrams of ports".

Table 9-10 Function List

Type	Function Input /Output / Input/ Output	Pull-up	Pull-down	Analog
FT1	Input /Output	R	R	−
FT4	Input (INT)	R	R	−
FT5	Input (AIN)	R	R	o
FT6	Input ($\overline{\text{BOOT}}$)	EnR	R	−

int: interrupt input
−: no exist
o: Exist
R: Forced disable during reset
EnR: Forced enable during reset

9.1.3.2 Type FT1

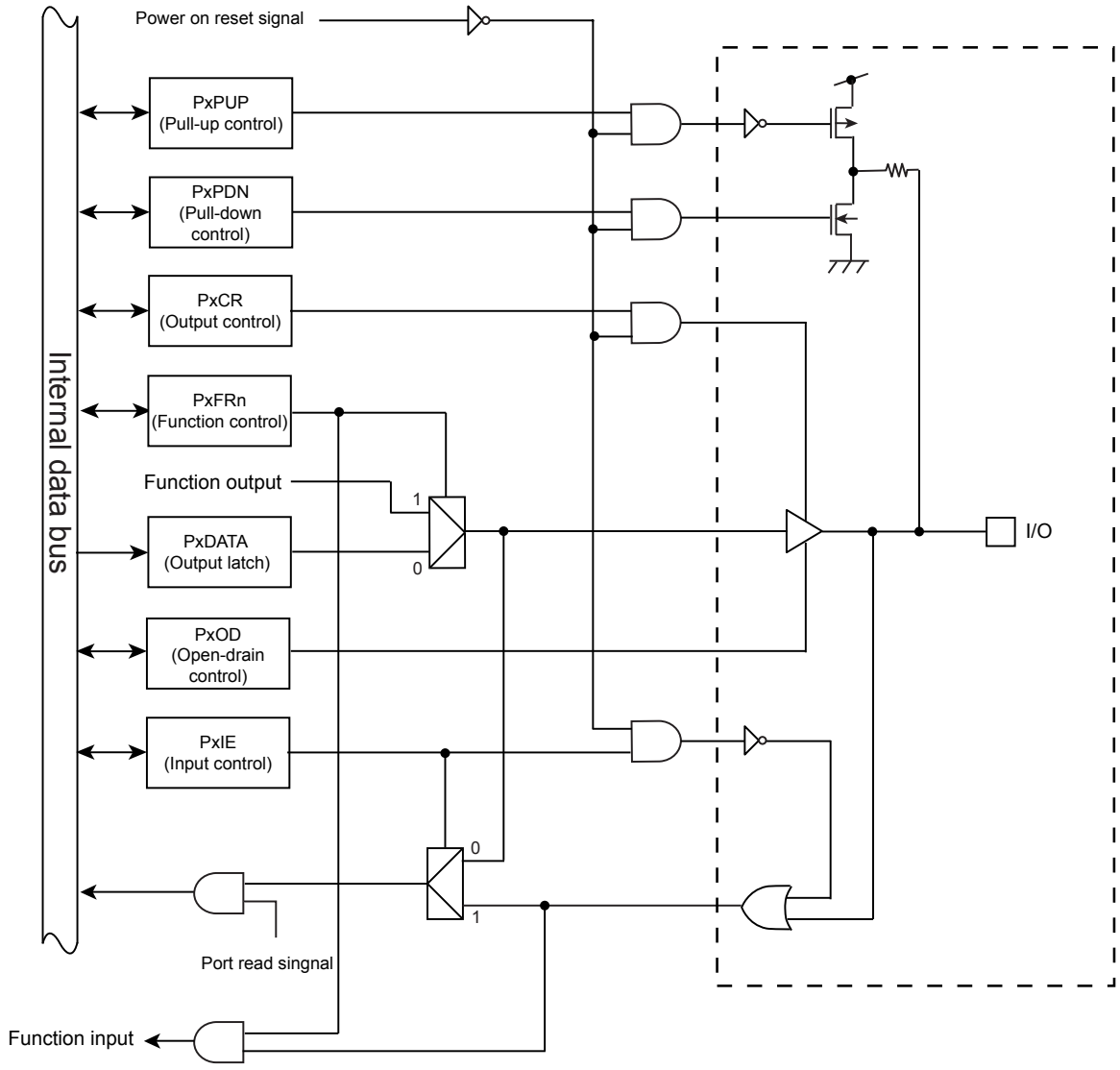


Figure 9-1 Port Type FT1

9.1.3.3 Type FT4

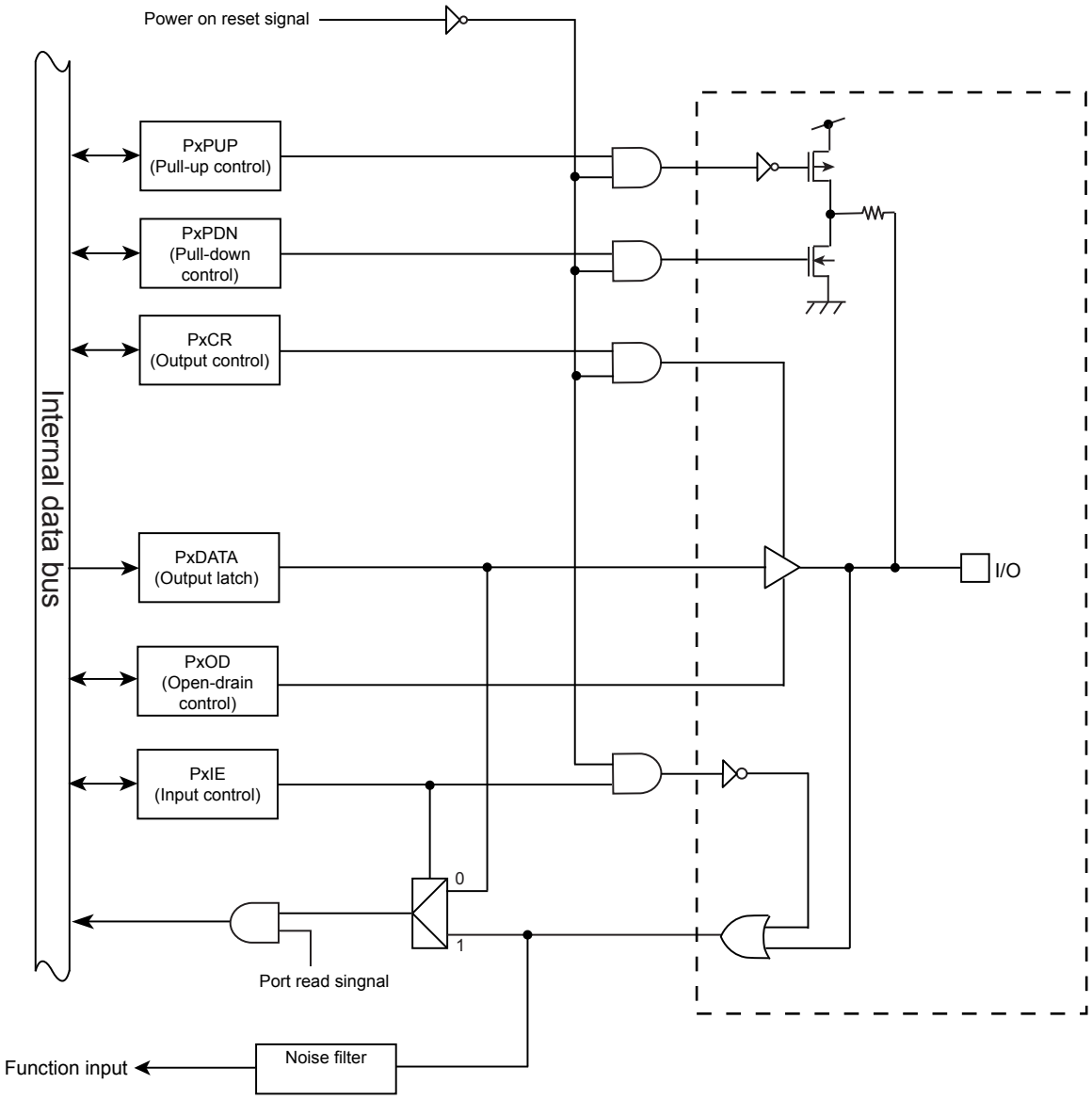


Figure 9-2 Port Type FT4

9.1.3.4 Type FT5

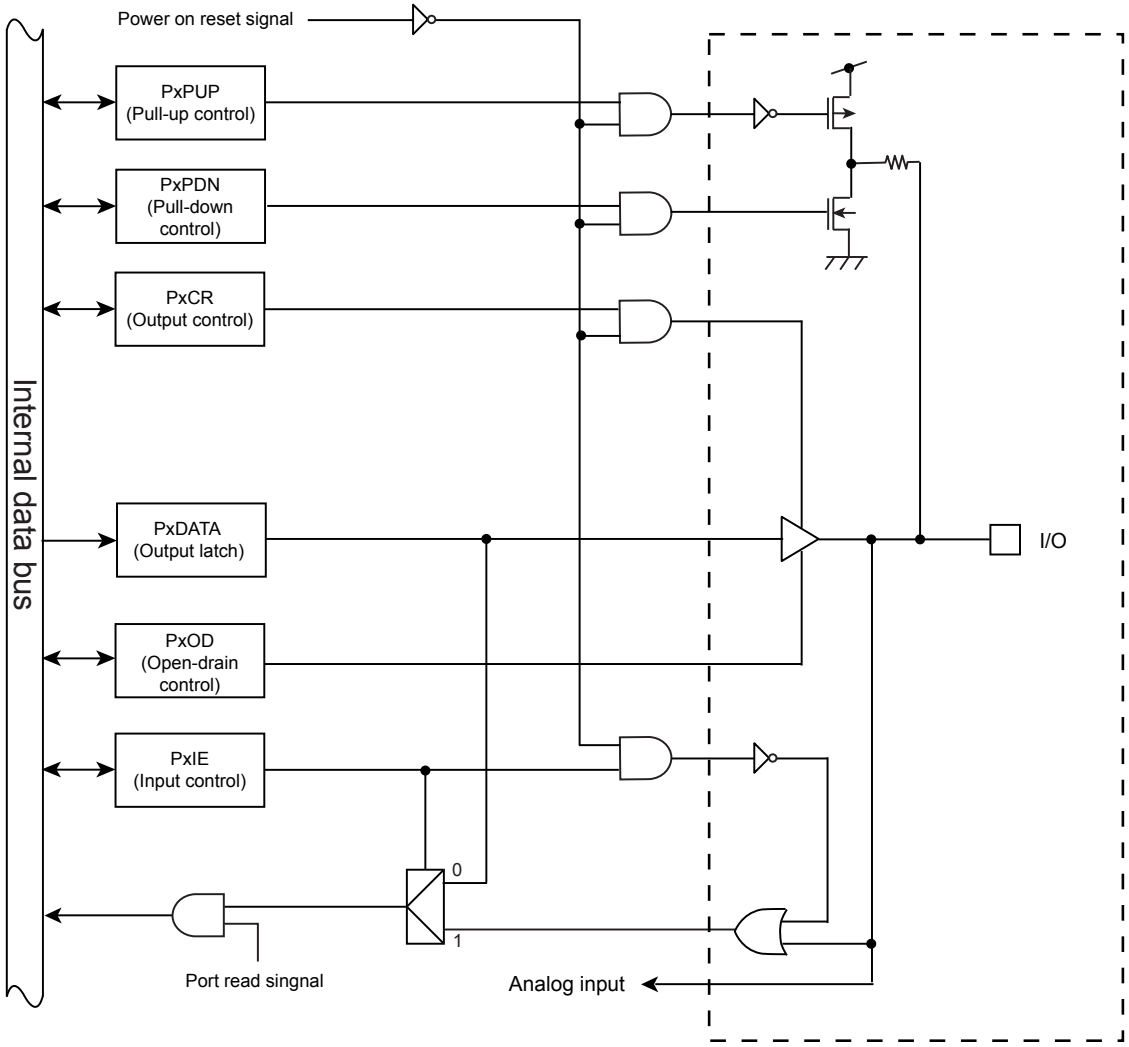


Figure 9-3 Port Type FT5

9.1.3.5 Type FT6

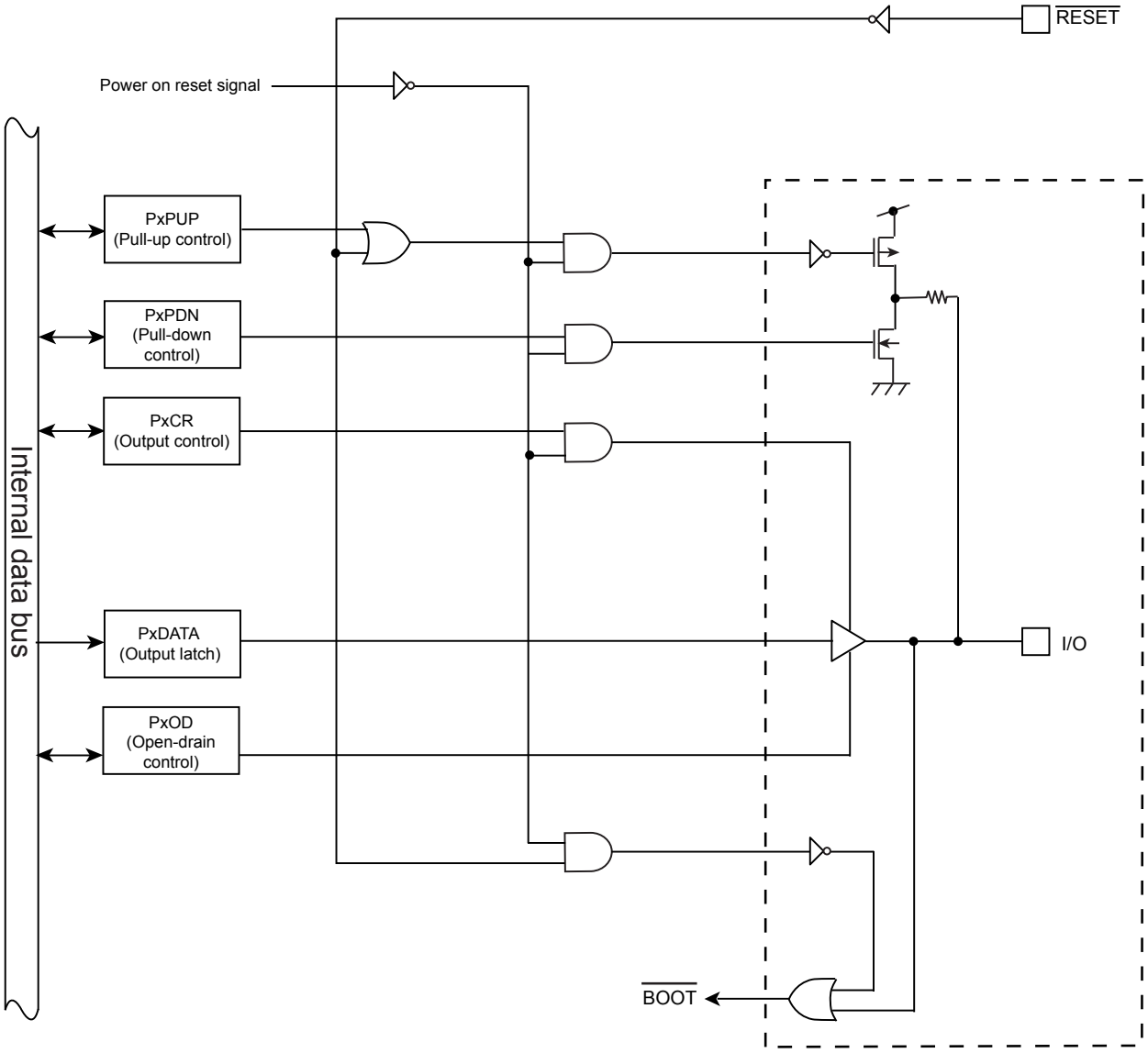


Figure 9-4 Port Type FT6

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

10.1 Outline

TMRBx has the operation modes shown below.

- Interval timer mode
- Event counter mode
- Programmable pulse generation (PPG) mode
- Programmable pulse generation (PPG) external trigger mode

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

- Frequency measurement
- Pulse width measurement

10.2 Block Diagram

TMRBx consists of a 16-bit up-counter, two 16-bit timer register (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 registers.

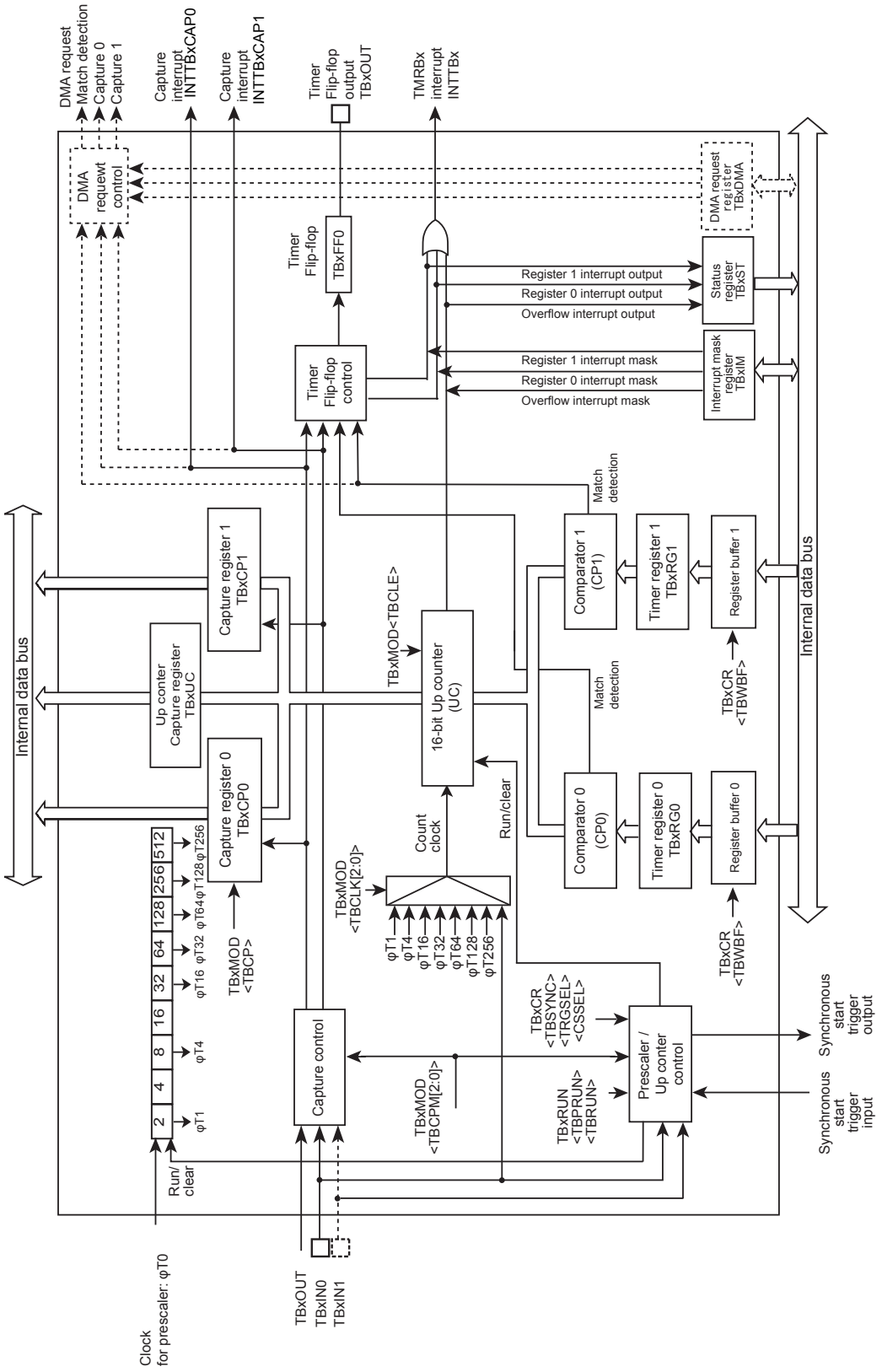


Figure 10-1 TMRBx Block Diagram

10.3 Registers

10.3.1 Register List

The table below shows control registers and their addresses.

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

Register name		Address (Base+)
Enable register	TBxEN	0x0000
RUN register	TBxRUN	0x0004
Control register	TBxCR	0x0008
Mode register	TBxMOD	0x000C
Flip-flop control register	TBxFFCR	0x0010
Status register	TBxST	0x0014
Interrupt mask register	TBxIM	0x0018
Up counter capture register	TBxUC	0x001C
Timer register 0	TBxRG0	0x0020
Timer register 1	TBxRG1	0x0024
Capture register 0	TBxCP0	0x0028
Capture register 1	TBxCP1	0x002C
DMA request enable register	TBxDMA	0x0030

Note: Do not modify the timer control register, timer mode register and timer flip-flop control register during timer operation. Users can modify them after stopping timer operation.

10.3.2 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: Disabled 1: Enabled Specifies the TMRBx operation. When the operation is disabled, no clock is supplied to the other registers in the TMRBx. This can reduce power consumption. (This disables reading from and writing to the other registers except TBxEN register.) To use the TMRB, enable the TMRBx operation (set to "1") before programming each register in the TMRBx. If the TMRBx operation is executed and then disabled, the settings will be maintained in each register.
6	TBHALT	R/W	Clock operation during debug HALT 0: run 1: stop Specifies the TMRBx clock setting to run or stop when the debug tool transits to HALT mode while in use.
5-0	−	R	Read as "0".

10.3.3 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

10.3.4 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	-	-	-	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: Disabled 1: Enabled
6	−	R/W	Write "0".
5	TBSYNC	R/W	Synchronous mode switching 0: individual (Each channel) 1: synchronous
4	−	R	Read as "0".
3-2	−	R/W	Write "0".
1	TRGSEL	R/W	Selects the edge when the external trigger is used. 0: rising 1: falling Selects the edge when counting is started by external triggers (TBxIN).
0	CSSEL	R/W	Selects the count start 0: starts by software 1: starts by external trigger

10.3.5 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	-	-	-	-	-	TBCPM		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	TBCP	-	-	TBCLE	TBCLK		
After reset	0	1	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-11	−	R	Read as "0".
10-8	TBCPM[2:0]	R/W	Sets capture timing by TBxIN0/1 and up-counter clearing timing. 000: Disabled 001: TBxIN0↑ TBxIN1↑ Captures a counter value on rising edge of TBxIN0 input into Capture register 0 (TBxCP0). Captures a counter value on rising edge of TBxIN1 input into Capture register 1 (TBxCP1). 010: TBxIN0↑ TBxIN0↓ Captures a counter value on rising edge of TBxIN0 input into Capture register 0 (TBxCP0). Captures a counter value on falling edge of TBxIN0 input into Capture register 1 (TBxCP1). 011: TBxFF0↑ TBxFF0↓ Captures a counter value on rising edge of TBxFF0 input into Capture register 0 (TBxCP0). Captures a counter value on falling edge of TBxFF0 input into Capture register 1 (TBxCP1). 100: Clears up-counter on TBxIN1↑ 101: Captures a counter value on TBxIN0↑ into Capture register 0(TBxCP0); clears up-counter on TBxIN1↑ If capture timing and up-counter clearing timing are same, capturing is performed first, and then up-counter is cleared. 110 to 111:Reserved
7	−	R/W	Write "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) captures a count value. Read as "1".
5-4	−	R	Read as "0".
3	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 up-counter matches with timer regsiter1 (TBxRG1).
2-0	TBCLK[2:0]	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: Do not make any changes of TBxMOD register while the TMRBx is running.

10.3.6 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 (reversed by software). 01: Set Sets TBxFF0 to "1". 10: Clear Clears TBxFF0 to "0". 11: Don't care This is always read as "11".

10.3.7 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 interrupt request flag 0: No overflow occurs 1: Overflow occurs When an up-counter is overflow, "1" is set.
1	INTTB1	R	Match (TBxRG1) interrupt request flag 0: No match is detected. 1: Detects a match with TBxRG1 When a match with the timer register 1 (TBxRG1) is detected, "1" is set.
0	INTTB0	R	Match(TBxRG0) interrupt request flag 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: Even if mask configuration by TBxIM register is valid, the status is set to TBxST register.

Note 2: When the interrupt mask configuration is disabled by the corresponding bit of TBxIM register, the interrupt is issued to the CPU.

Note 3: To clear the flag, read TBxST register.

10.3.8 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 request mask 0: Disable 1: Enable Sets the up-counter overflow interrupt to disable or enable.
1	TBIM1	R/W	Match (TBxRG1) interrupt request mask 0: Disable 1: Enable Sets the match interrupt request mask with the timer register 1 (TBxRG1) to enable or disable.
0	TBIM0	R/W	Match (TBxRG0) interrupt request mask 0: Disable 1: Enable Sets the match interrupt request mask with the Timer register 0 (TBxRG0) to enable or disable.

Note: Even if mask configuration by TBxIM register is valid, the status is set to TBxST register.

10.3.9 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 during the counter operation, the current value of up-counter will be captured.

10.3.10 TBxRG0 (Timer Register 0)

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

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

10.3.11 TBxRG1 (Timer Register 1)

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

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

10.3.12 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.

10.3.13 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.

10.3.14 TBxDMA(DMA request 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	-	-	-	-	-	TBDMAEN2	TBDMAEN1	TBDMAEN0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	TBDMAEN2	R/W	Selects DMA request: compare match with TBxRG1. (Note 1) 0: Disabled 1: Enabled
1	TBDMAEN1	R/W	Selects DMA request: input capture 1 0: Disabled 1: Enabled
0	TBDMAEN0	R/W	Selects DMA request: input capture 0 0: Disabled 1: Enabled

Note 1: When mask configuration by TBxIM<TBIM1> register is valid, DMA request does not occur even if a DMA request is enabled with <TBDMAEN2>.

Note 2: The assignment of DMA request factor differs by channel. Refer to "Product Information" Chapter

10.4 Description of Operation

10.4.1 Prescaler

TMRBx has 4-bit prescaler to generate the source clock for up-counter.

A input clock φT0 to the prescaler is either among fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16 or fperiph/32 selected by CGSYSCR<PRCK[2:0] in the CG circuit. The peripheral clock is either fgear (a clock selected by CGSYSCR<FPSEL> in the CG circuit) or fc (a clock before divided by the clock gear).

The operation or the stoppage of prescaler is set by TBxRUN<TBPRUN>. Writing "1" to the bit is to start counting; writing "0" to the bit is to stop counting.

10.4.2 Up-counter (UC)

UC is a 16-bit binary counter.

10.4.2.1 Source clock

The up-counter's source clock is specified by TBxMOD<TBCLK[2:0]>.

It can be selected from the prescaler output clock - φT1, φT4, φT16, φT32, φT64, φT128 and φT256 - or the external clock of the TBxIN pin.

10.4.2.2 Counter start / stop

There are software start, external trigger start and synchronous start to start the counter.

1. Software start

   If <TBRUN> is set to "1", the counter will start. If "0" is set to the <TBRUN>, the counter will stop and the up-counter will be cleared at the same time.

2. External trigger start

   In the external trigger mode, the counter will be started by external signals.

   If TBxCR<CSSEL> is set to "1", the external trigger start mode is set. At this time, if <TBRUN> is set to "1", the condition of the counter will be trigger wait. The counter will start on the rising/falling edge of TBxIN.

   TBxCR<TRGSEL> bit specifies the switching external trigger edges.

   <TRGSEL>="0": Rising edge of TBxIN is selected.

   <TRGSEL>="1": Falling edge of TBxIN is selected.

   If <TBRUN> is set to "0", the counter will stop and the up-counter will be cleared at the same time.

3. Synchronous start

   In the timer synchronous mode, synchronous start timers can be possible. If timer synchronous mode is used in the PPG output mode, motor drive application can be achieved.

   Depending on products, the combination of master channels and slave channels have already been determined. For the combination of master channels and slave channels of this product, refer to Chapter Product Information.

TBxCR<TBSYNC> bit specifies the switching of synchronous mode. If <TBSYNC> bit of a slave channel is set to "1", the counter will start/stop synchronously with the software or external trigger start of a master channel. TBxRUN<TBPRUN, TBRUN> bit of a slave channel is not required to set. <TBSYNC> bit of the master channel must be set to "0".

Note that if the external trigger counter mode and timer synchronous mode are both set, the timer synchronous mode gains a higher priority.

10.4.2.3 Counter Clear

The up-counter is cleared at the timings below:

1. When a match with TBxRG1 is detected

   By setting TBxMOD<TBCLE> = "1", up-counter is cleared if the comparator detects a match between UC and TBxRG1.

2. When up-counter stops

   Up-counter stops and is cleared when TBxRUN<TBRUN> = "0".

3. When TBxIN1 rises

   Up-counter is cleared on rising edge of TBxIN1 when TBxMOD<TBCPM[2:0]> is "100" or "101".

10.4.2.4 Up-Counter Overflow

If up-counter overflows, the INTTBx overflow interrupt is generated.

10.4.3 Timer Registers (TBxRG0, TBxRG1)

TBxRG0 and TBxRG1 are registers to compare with a value of up-counter. The two registers are built into each channel. If the comparator detects a match between a value set in timer register and a value in the up-counter, the comparator outputs the match detection signal.

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

Disabling or Enabling of double buffering is specified by TBxCR<TBWBF>. If <TBWBF> = 0, double buffering becomes disable; if <TBWBF> ="1", it becomes enable.

When double buffering is enabled, data is transferred from the register buffer to the timer register (TBxRG0/1) when up-counter is matched with TBxRG1.

When up-counter is stopped even if double buffering is enabled, the double buffering operates as a single buffer, and data can be written to the TBxRG0 and TBxRG1 directly.

10.4.4 Capture Control

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

Software can also capture the value of up-counter into capture registers. The value of up-counter are captured into the TBxCP0 in each time "0" is written to TBxMOD<TBCP>.

10.4.5 Capture Registers (TBxCP0, TBxCP1)

These registers capture the value of up-counter.

10.4.6 Up-Counter Capture Register (TBxUC)

If TBxUC register is read during the counter operation, the current value of up-counter will be captured and the value will be read. The value captured at the end is held while the counter is stopping.

10.4.7 Comparators (CP0, CP1)

These circuits compare up-counter and values set to TBxRG0/1 to detect a match. If a match is detected, INTTBx occurs.

10.4.8 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. Reversing is enabled or disabled by setting the TBxFFCR<TBC1T1, TBC0T1, TBC1T1, TBC1T0>.

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, program the corresponding port settings beforehand.

10.4.9 Capture Interrupt (INTTBxCAP0, INTTBxCAP1)

INTTBxCAP0 and INTTBxCAP1 can be generated at the timing of latching value from the up-counter into TBxCP0 and TBxCP1.

10.4.10 DMA Request

A DMA request is issued to the DMAC at the timing of match interrupts or capture interrupts generation. When DMA transfer is performed, set DMA request to be enabled by the corresponding bit of TBxDMA register

Note: Even if DMA request is enabled by TBxDMAREQ<TBDMAEN2>, if interrupt mask configuration has been set by TBxIM<TBIM1>, a DMA request does not occur.

10.5 Description of Operation for each mode

10.5.1 Interval Timer Mode

When interrupts is generated in constant intervals, set the interval time to the timer register (TBxRG1) to generate the INTTBx interrupt.

TBxEN<TBEN> = "1"	Enables TMRBx operation.
TBxRUN<TBPRUN><TBRUN> = "00"	Stops prescaler and counter.
Enable INTTBx interrupt	Set "1" to the bit corresponding to INTTBx interrupt to enable the interrupt.
TBxFFCR<TBC1T1><TBC0T1><TBE1T1><TBE0T1> = "0000"	Sets TBxFF0 not to invert the signal by detection of a match between TBxRG0 and TBxRG1, capturing TBxCP0 and TBxCP1.
TBxMOD<TBCPM[2:0]> = "000" TBxMOD<TBCP> = "1" TBxMOD<TBCLE> = "0" TBxMOD<TBCLK[2:0]> = "****" (*** = "001" to "111")	Changes to prescaler output clock as input clock. Specifies capture function to disable.
TBxRG1 = 0x****	Specifies a timer interval.
TBxRUN<TBPRUN><TBRUN> = "11"	Starts prescaler and counter.

Note:*; Optional value

10.5.2 Event Counter Mode

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

The up-counter counts up on the rising edge of TBxIN pin input. The value of up-counter can be captured by software. It is possible to read the count value by reading capture values.

TBxEN<TBEN> = "1"	Enables TMRBx operation.
TBxRUN<TBPRUN><TBRUN> = "00"	Stops prescaler and counter.
Allocate the corresponding port to .	
TBxFFCR<TBC1T1><TBC0T1><TBE1T1><TBE0T1> = "0000"	Sets TBxFF0 not to reverse the signal by detection of a match between TBxRG0 and TBxRG1, capturing TBxCP0 and TBxCP1.
TBxMOD<TBCPM[2:0]> = "000" TBxMOD<TBCP> = "1" TBxMOD<TBCLE> = "0" TBxMOD<TBCLK[2:0]> = "000"	Changes input clock to .
TBxRUN<TBPRUN><TBRUN> = "11"	Starts prescaler and counter.
TBxMOD<TBCP> = "0"	Captures a counter value by software.

Note:*; Optional value

10.5.3 Programmable Pulse Generation (PPG) Output Mode

Square wave can be output in any frequency and duty. The output pulse can be either low-active or high-active.

TBxFF0 is reversed when the up-counter matches the set value of TBxRG0 and TBxRG1. TBxFF0 can be output from TBxOUT pin.

Note that the set value of TBxRG0 and TBxRG1 must be satisfy the following requirement.

Set value of TBxRG0 < Set value of TBxRG1

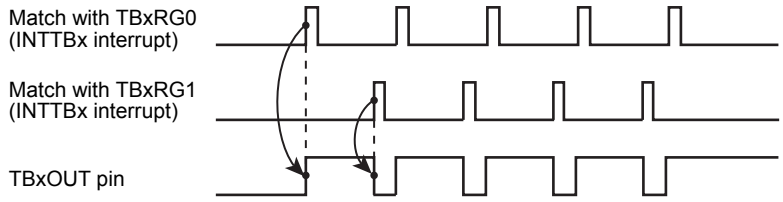


Figure 10-2 Example of programmable pulse generation output

In this mode, by enabling the double buffering, the value of register buffer 0 and 1 are shifted into TBxRG0 and TBxRG1 when UC matches the value of TBxRG1.

This makes possible to modify frequency and duty without a concern of a change timing of TBxRG0 and TBxRG1.

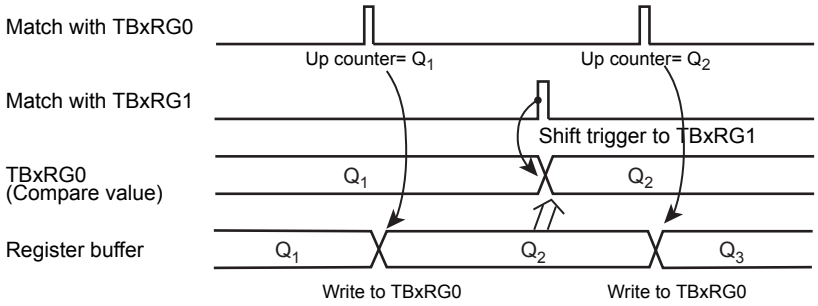


Figure 10-3 Register buffer operation

The block diagram of this mode is shown below.

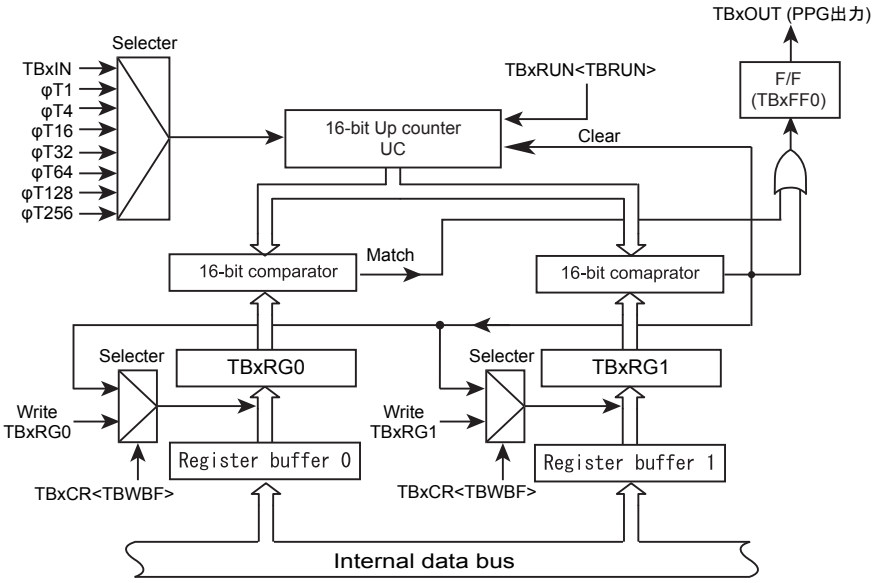


Figure 10-4 Block diagram of 16-bit PPG mode

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

TBxEN<TBEN> = "1"	Enables TMRBx operation.
TBxRUN<TBPRUN><TBRUN> = "00"	Stops prescaler and counter.
TBxCR<TBWBF> = "1"	Enables double buffer.
TBxRG0 = 0x****	Sets a duty ratio.
TBxRG1 = 0x****	Sets a cycle.
TBxFFCR<TBC1T1><TBC0T1><TBE1T1><TBE0T1> = "0011" TBxFFCR<TBFF0C[1:0]> = "10"	Sets TBxFF0 to invert the signal by detection of a match between TBxRG0 or TBxRG1 and the up-counter; sets TBxFF0 not to reverse the signal by capturing TBxCP0 or TBxCP1.Sets an initial value of TBxFF0 to "0".
TBxMOD<TBCPM[2:0]> = "000" TBxMOD<TBCP> = "1" TBxMOD<TBCLE> = "1" TBxMOD<TBCLK[2:0]> = "***" (*** = "001" to "111")	Changes to prescaler output clock as input clock. Specifies capture function to disable.
Allocate the corresponding port to TBxOUT.	
TBxRUN<TBPRUN><TBRUN> = "11"	Starts prescaler and counter.

Note: *; Optional value

10.5.4 Programmable Pulse Generation (PPG) External Trigger Output Mode

A PPG wave with a short delay time can be output when TMRBx is started by the external trigger count start mode in the PPG (Programmable Pulse Generation) output mode.

The example of one-shot pulse output (with delay) using external trigger count start is shown below.

To start TMRB in the external trigger count start mode, set "1" to TBxCR<CSSEL> and set "0" to TBxCR<TRGSEL> to count up TBxIN on the rising edge while 16-bit counter has been stopped.

TBxRG0 is set the delay time (d) from an external trigger signal. TBxRG1 is set the value (d)+(p) of which the delay time (d) is added to the width (p) of one-shot pulse.

To reverse TBxFF0 when the up-counter matches TBxRG0/1, set "1" to TBxFFCR<TBE1T1>, <TBE0T1>.

To start the up-counter, set "1" to TBxRUN<TBPRUN>, <TBRUN>.

At this time, if the external trigger pulse is input to TBxIN, the up-counter is started on the rising edge of external trigger pulse.

TBxFF0 is reversed when the up-counter counts up to (d). It matches TBxRG0 and then TBxFF0 becomes "High" level.

TBxFF0 is reversed when the up-counter counts up to (d)+(p). It matches TBxRG1 and then TBxFF0 becomes "Low" level.

To avoid changing the level of TBxFF0 by INTTBx that occurs when the up-counter matches TBxRG1, clear TBxFFCR<TBE1T1>, <TBE0T1> to "0" or stop the up-counter by TBxRUN<TBPRUN>, <TBRUN>.

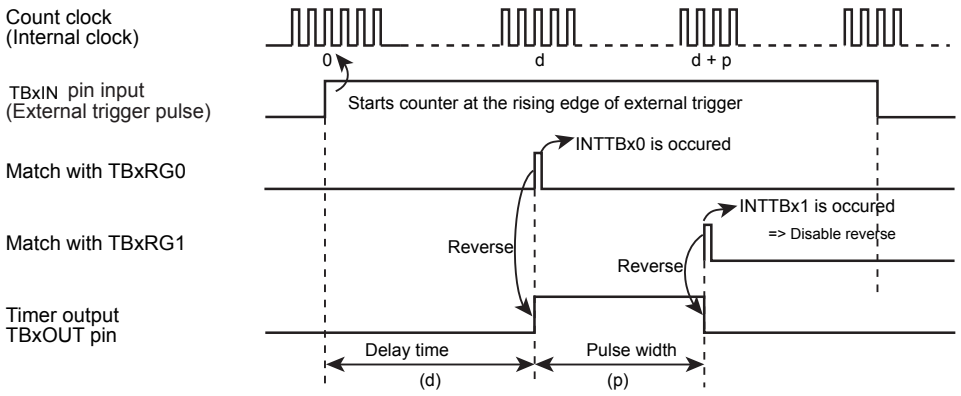


Figure 10-5 One-shot pulse output with delay using external trigger count start

The following shows the case that 2 ms width one-shot pulse is output by triggering TBxIN input on the rising edge after 3 ms has been elapsed. In this example, the source clock is φT1.

[Main process]	
Allocates a corresponding port to	
TBxEN<TBEN> = "1"	Enables TMRBx operation.
TBxRUN<TBPRUN><TBRUN> = "00"	Stops prescaler and counter.
TBxRG0 = 0x****	Sets a counter value. (3 ms/φT1
TBxRG1 = 0x****	Sets a cycle. ((3+2)ms/φT1)
TBxFFCR<TBC1T1><TBC0T1><TBE1T1><TBE0T1> = "0011" TBxFFCR<TBFF0C[1:0]> = "10"	Sets TBxFF0 to invert the signal by detection of a match between TBxRG0 or TBxRG1 and the up-counter; sets TBxFF0 not to reverse by capturing TBxCP0 or TBxCP1. Sets an initial value of TBxFF0 to "0".
TBxMOD<TBCPM[2:0]> = "000" TBxMOD<TBCP> = "1" TBxMOD<TBCLE> = "1" TBxMOD<TBCLK[2:0]> = "001"	Sets a source clock to φT1. Disables the capture function.
Allocates a corresponding port to TBxOUT.	
TBxIM<TBIMOF><TBIM1><TBIM0> = "101"	Masks interrupts except one caused by a match with TBxRG1.
Capture enable set register = 0x********	Permits INTTBx interrupt by setting the corresponding bit to "1".
TBxRUN<TBPRUN><TBRUN> = "11"	Starts prescaler and counter.
[Process in INTTBx interrupt service routine]	
TBxFFCR<TBE1T1><TBE0T1> = "00"	Disables TBxFF0 reverse trigger setting.
TBxRUN<TBPRUN><TBRUN> = "00"	Stops prescaler and counter.

Note: *; Optional value

10.6 Applications Using Capture Function

The capture function can be used in many applications.

The applications are shown below.

1. Frequency measurement
2. Pulse width measurement

10.6.1 Frequency Measurement

The following are examples of measuring a frequency of external clock.

In this section, TMRBm is used as 16-bit interval timer mode and TMRBn is used as 16-bit event counter mode.

To count the up-counter of TMRBn freely by an external clock, set TMnMOD<TBCLK> to "000" and set TBnRUN<TBE1T1><TBE0T1> to "11".

To reverse TBmFF0 when the up-counter of TMRBm matches TBmRG0 and TBmRG1, set TBmFFCR<TBE1T1> <TBE0T1> to "11".

To capture a value of the up-counter into TBnCP0 on the rising edge of TBmFF0 and to capture a value of the up-counter into TBnCP1 on the falling edge of TBmFF0, set TBnMOD<TBCPM[2:0]> to "011".

Set the number of counting external clocks to TBmRG0/1 and start TMRBm.

When a value the up-counter of TMRBm matches TBmRG0, TBmFF0 rises and a value of the up-counter of TMRBn is captured into TBnCP0. When the up-counter of TMRBm matches TBmRG1, TBmFF0 falls and a value of the up-counter of TMRBn is captured into TBnCP1.

A frequency can be measured from (TBnCP1 - TBnCP0) ÷ (TBmRG1 - TBmRG0) using INTTBm.

For example, the difference between TBmRG1 and TBmRG0 is 0.5 s and the difference between TBnCP1 and TBnCP0 is 100, the frequency is 200 Hz (100 ÷ 0.5 s = 200Hz)

Depending on the changing timing of TBmFF0, the result of TBnCP1 - TBnCP0 will be minus. Correct the value if TBnCP1 - TBnCP0 is minus.

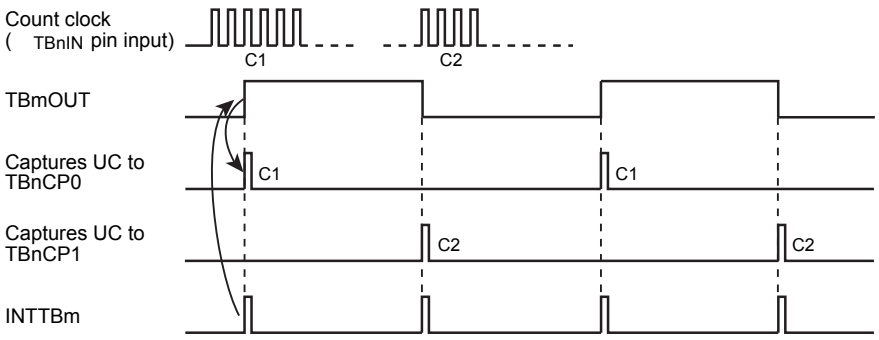


Figure 10-6 Frequency measurement

The following shows in the case that the measured pulse is input to TBnIN. In this example, the source clock is φT1.

[Main process] TBmFF0 capture setting	
Allocates a corresponding port to	
TBmEN<TBEN> = "1"	Enables TMRBm operation.
TBmRUN<TBPRUN><TBRUN> = "00"	Stops prescaler and counter.
TBnEN<TBEN> = "1"	Enables TMRBn operation.
TBnRUN<TBPRUN><TBRUN> = "00"	Stops prescaler and counter.
TBmCR<TBWBF> = "1"	Enables double buffer.
TBmRG0 = 0x****	Sets an external clock measurement time 1.
TBmRG1 = 0x****	Sets an external clock measurement time 2.
TBmFFCR<TBC1T1><TBC0T1><TBE1T1><TBE0T1> = "0011" TBmFFCR<TBFF0C[1:0]> = "10"	Sets TBmFF0 to invert the signal by detection of a match between TBmRG0 or TBmRG1 and the up-counter. Sets TBmFF0 not to reverse the signal by capturing TBmCP0 or TBmCP1. Sets an initial value of TBmFF0 to "0".
TBnMOD<TBCPM[2:0]> = "011" TBnMOD<TBCP> = "1" TBnMOD<TBCLE> = "0" TBnMOD<TBCLK[2:0]> = "000"	Captures a value of up-counter on rising/falling edge of TBmFF0. Disables up-counter clear. Sets an input clock to .
TBmIM<TBIMOF><TBIM1><TBIM0> = "101"	Masks interrupts except a match with TBmRG1.
Capture enable set register = 0x********	Sets "1" to the bit corresponding to INTTBm to enable the interrupt.
TBmRUN<TBPRUN><TBRUN> = "11"	Starts prescaler and counter.
TBnRUN<TBPRUN><TBRUN> = "11"	Starts prescaler and counter.
[Process in INTTBm interrupt service routine]	
TBmFFCR<TBE1T1><TBE0T1> = "00"	Disables TBmFF0 reverse trigger setting
Interrupt enable clear register = 0x********	Sets "1" to the bit corresponding to INTTBm to disable the interrupt.
Reads TBnCP0/1 and calculate a frequency.	

Note:m, n ; Optional channel number, *; Optional value

10.6.2 Pulse Width Measurement

"High" level width of the external pulse can be measured.

To capture a value of the up-counter into TBxCP0 on the rising edge of TBxIN and to capture a value of the up-counter into TBxCP1 on falling edge of TBxIN, set TBxMOD<TBCPM> to "010".

Enables INTTBxCAP1 interrupt.

Enables TMRBx operation.

If rising edge of external pulse is input to TBxIN, a value of the up-counter is captured into TBxCP0. If falling edge of external pulse is input to TBxIN, a value of the up-counter is captured into TBxCP1 and INTTBxCAP1 interrupt occurs.

The "High" level width of the external pulse can be calculated by multiplying the difference between TBxCP0 and TBxCP1 by the clock cycle of a prescaler output 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.

If a pulse width is beyond the maximum count time of the up-counter, make a correction.

In addition, "Low" level width of an external pulse can be measured.

To calculate the "Low" level width, enable INTTBxCAP0 interrupt and multiply the time difference between first C2 and second C1 in "Figure 10-7 Pulse width measurement" at the second time of INTTBxCAP0 by the cycle of the prescaler output clock.

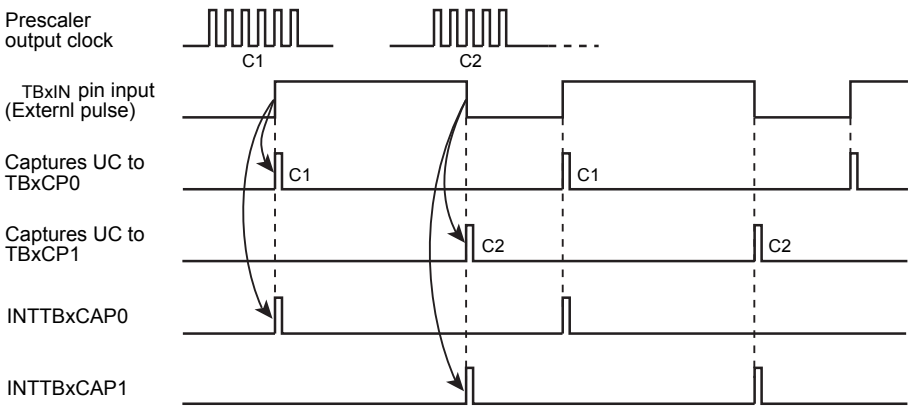


Figure 10-7 Pulse width measurement

The following describes the example of the measurement of "High" level width of the external pulse into TBxIN. In this example, the source clock is φT1.

[Main process] capture setting	
Allocates a corresponding port to .	
TBxEN<TBEN> = "1"	Enables TMRBx operation.
TBxRUN<TBPRUN><TBRUN> = "00"	Stops prescaler and counter.
TBxFFCR<TBC1T1><TBC0T1><TBE1T1><TBE0T1> = "0000" TBxFFCR<TBFF0C[1:0]> = "10"	Sets TBxFF0 not to invert the signal if TMRBx detects a match between TBxRG0 or TBxRG1 and the up-counter, or when capturing TBxCP0 or TBxCP1. Sets an initial value of TBxFF0 to "0".
TBxMOD<TBCPM[2:0]> = "010" TBxMOD<TBCP> = "1" TBxMOD<TBCLE> = "0" TBxMOD<TBCLK[2:0]> = "001"	Sets the source clock to φT1. Capture a value of up-counter on rising edge of pin into TBxCP0; captures a value of up-counter on falling edge of pin into TBxCP1.
Interrupt enable set register = 0x********	Set "1" to the bit corresponding to INTTBxCAP1 interrupt to enable the interrupt.
TBxRUN<TBPRUN><TBRUN> = "11"	Starts prescaler and counter.
[Process in INTTBxCAP1 interrupt service routine] Calculates "High" level width	
TBxFFCR<TBE1T1><TBE0T1> = "00"	Disables TBxFF0 reverse trigger setting
Interrupt enable clear register = 0x********	Set "1" to the bit corresponding to INTTBxCAP1 interrupt to disable the interrupt.
Reads a value of TBxRG0/1 and calculates "High" level width.	

Note: *; Optional value

11. 16-Bit Timer A (TMR16A)

11.1 Outline

TMR16A contains the following functions:

- Match interrupt
- Square waveform output
- Read capture

In this chapter, "x" indicates a channel number.

11.2 Block Diagram

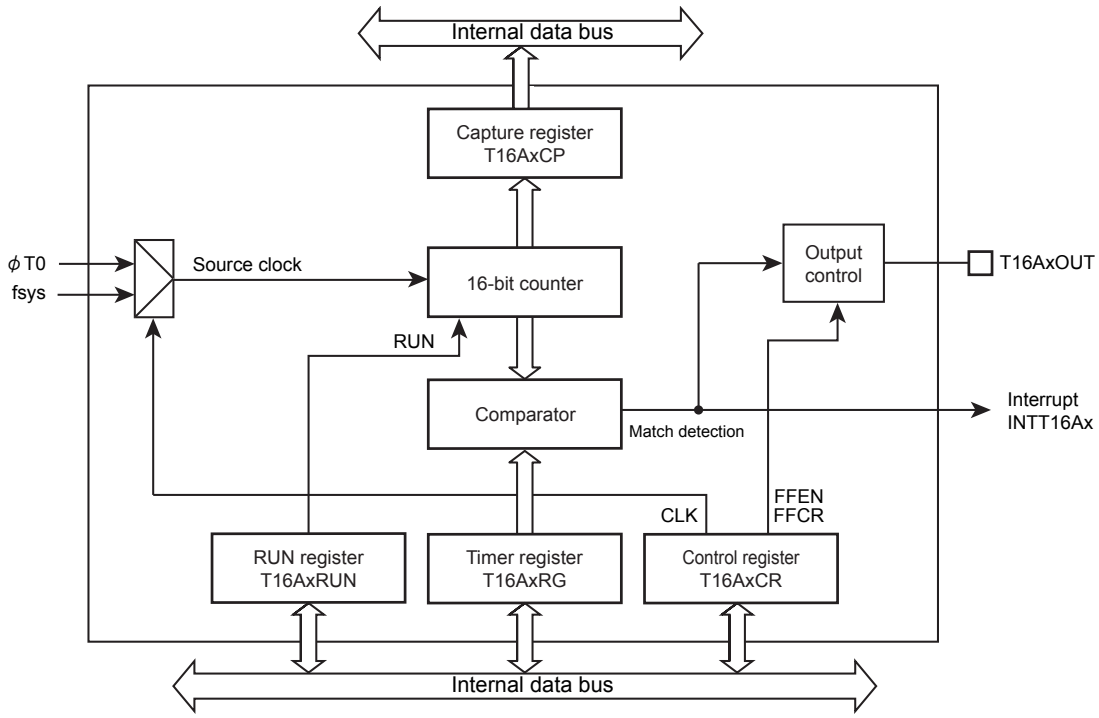


Figure 11-1 Block diagram of TMR16A

11.3 Registers

11.3.1 Register List

The table below shows control registers and their addresses.

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

Register name		Address(Base+)
Enable register	T16AxEN	0x0000
RUN register	T16AxRUN	0x0004
Control register	T16AxCR	0x0008
Timer register	T16AxRG	0x000C
Capture register	T16AxCP	0x0010

Note: When T16ARUN<RUN> is set to "1", do not modify T16AxEN, T16AxCR, T16AxRG and T16AxCP.

11.3.1.1 T16AxEN (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	-	-	-	-	-	-	HALT	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1	HALT	R/W	Operation during halt mode debug 0: Operating 1: Stop Specifies the operation during halt mode debug. Write "1" to the bit to stop the operation.
0	-	R/W	Reserved.

11.3.1.2 T16AxRUN (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	-	-	-	-	-	-	-	RUN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as "0".
0	RUN	R/W	Counter operation 0: Stop 1: Operating

11.3.1.3 T16AxCR (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	FFEN	-	FFCR		-	-	-	CLK
After reset	0	0	1	1	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15	−	R/W	Write to "0"
14-8	−	R	Read as "0".
7	FFEN	R/W	Inverse of T16AxOUT 0: Disabled 1: Enabled Write "1" to the bit to invert T16AxOUT when the counter matches with T16ARG.
6	−	R	Read as "0".
5-4	FFCR[1:0]	W	T16AxOUT control 00: Invert 01: Set 10: Clear 11: No operation Write a value to the bit to control T16AxOUT by software. Read as "11".
3-1	−	R	Read as "0".
0	CLK	R/W	Source clock 0: fsys 1: ΦT0 Specifies a source clock.

11.3.1.4 T16AxRG (Timer Register)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-0	RG[15:0]	R/W	Set a value to compare with a counter

Note: Do not set "0x0000".

11.3.1.5 T16AxCP (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	CP[15:8]							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CP[7:0]							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-0	CP[15:0]	R/W	Counter value [Read] Reads a current counter value. [Write] Sets a counter value. Since the counter is cleared only when the counter matches with T16AxRG<RG>, write "0x0000" to clear the register before starting the operation.

11.4 Operation Description

11.4.1 Timer Operation

1. Preparation

   Select a source clock with T16AxCR<CLK>. Write "0" to set fsys or write "1" to set ΦT0. Set a counter value to T16AxRG<RG[15:0]>.

2. Counter operation

   Before starting counter operation, set "0x0000" to T16AxCP<CP> to clear the counter.

   To start count-up, set "1" to T16AxRUN<RUN>. If the counter value matches with a value of T16AxRG<RG[15:0]>, it will be cleared to "0x0000" and continued to count-up.

3. Match detection interrupt generation

   If a counter value matches with a value of T16AxRG<RG[15:0]>, a match detection interrupt INTT16Ax will be output.

4. Stop

   To stop counts, set "0" to T16AxRUN<RUN>. The counter value is held. Then clear the counter before counting is started by setting "1" to <RUN>.

Note: Modification of T16AxCR, T16AxRG and T16AxCP must be performed while the counter is stopping (T16AxRUN<RUN> is set to "0").

11.4.2 T16AxOUT Control

T16AxOUT is modified by register setting or by matching the counter with T16AxRG.

An initial state of T16AxOUT is "0".

1. Control by software

   With T16AxCR<FFCR[1:0]> setting, T16AxOUT can be specified; "1" is to set, "0" is to clear, and also the inverted setting is possible.

   Modify T16AxCR while the counter stops (T16AxRUN<RUN> is "0").

2. Inverse due to matching counter

   Write "1" to T16ACR<FFEN> to invert T16AxOUT. When T16AxRG<RG[15:0]> matches with a counter value, T16AxOUT will invert. When the counter stops, a state of T16AxOUT will remain.

11.4.3 Read Capture

A current value of the counter can be captured by reading T16AxCP<[15:0]>.

11.4.4 Automatic Stop

With the setting of T16AxEN <HALT>, TMR16A automatically stops in the following conditions:

1.

2. Debug halt

With T16AxEN<HALT> setting, TMR16Ax operation during debug halt can be specified. If "0" is set, TMR16Ax automatically stops count-up when the transition to the debug halt mode occurs. If debug halt mode of the core is cancelled, count-up will restart.

12. High Resolution 16-bit Timer (TMRD ver.C)

12.1 Outline

TMRD consists of 16-bit program rectangular wave output (PPG).

1. 16-bit programmable rectangular pulse form output (PPG)

   This timer has the 1-bit modulation function that allows a higher pseudo-resolution when using PWM outputs.

   In 16-bit programmable rectangular pulse output mode, the following two modes are available.

   - PPG mode in which two units independently output programmed rectangular waves.
   - Interlock PPG mode is a mode that can change the phase relation between the pulse generated by one timer unit and the pulse generated by another timer unit in the range from -180 to +180 degree.

2. Synchronous cycle signal output function

   This timer can output signals that is synchronous with a timing of rectangular wave output (PPG).

As notational conventions, circuit elements and registers in TMRD are collectively described as "m", "n" or "**" in a product name.

In this section, "m", "n" and "**" lie within the following bounds in the table unless otherwise noted.

Table 12-1 Description of notation

Symbol	Value	Note
m	0 to 5	n=0
	0 to 4	n=1
n	0 to 1	-
**	n0 or n1	-

12.2 Configuration

12.2.1 Timer Unit

TMRD consists of 2 timer units such as TMR0 and TMR1.

TMDR0 consists of 16-bit counter (UC0), 6 comparators (CP0m, m=0 to 5) and 2 outputs channels (CH00 and CH01).

CH00 uses CP01, CP02 and wave generation circuit 00 (WG00) to output PPG output (PPG00). CH01 uses CP03, CP04 and WG01 to output PPG output (PPG01).

TMR1 consists of 16-bit counter (UC1), 5 comparators (CP1m, m=0 to 4) and 2 output channels (CH10 and CH11).

CH10 uses CP11, CP12 and WG10 to output PPG output (PPG10). CH11 uses CP13, CP14 and WG11 to output PPG output (PPG11).

TMR0 and TMR1 can operate in an interlocked manner. They operate on the cycle specified in CP05.

12.2.1.1 Block Diagram of Timer Unit

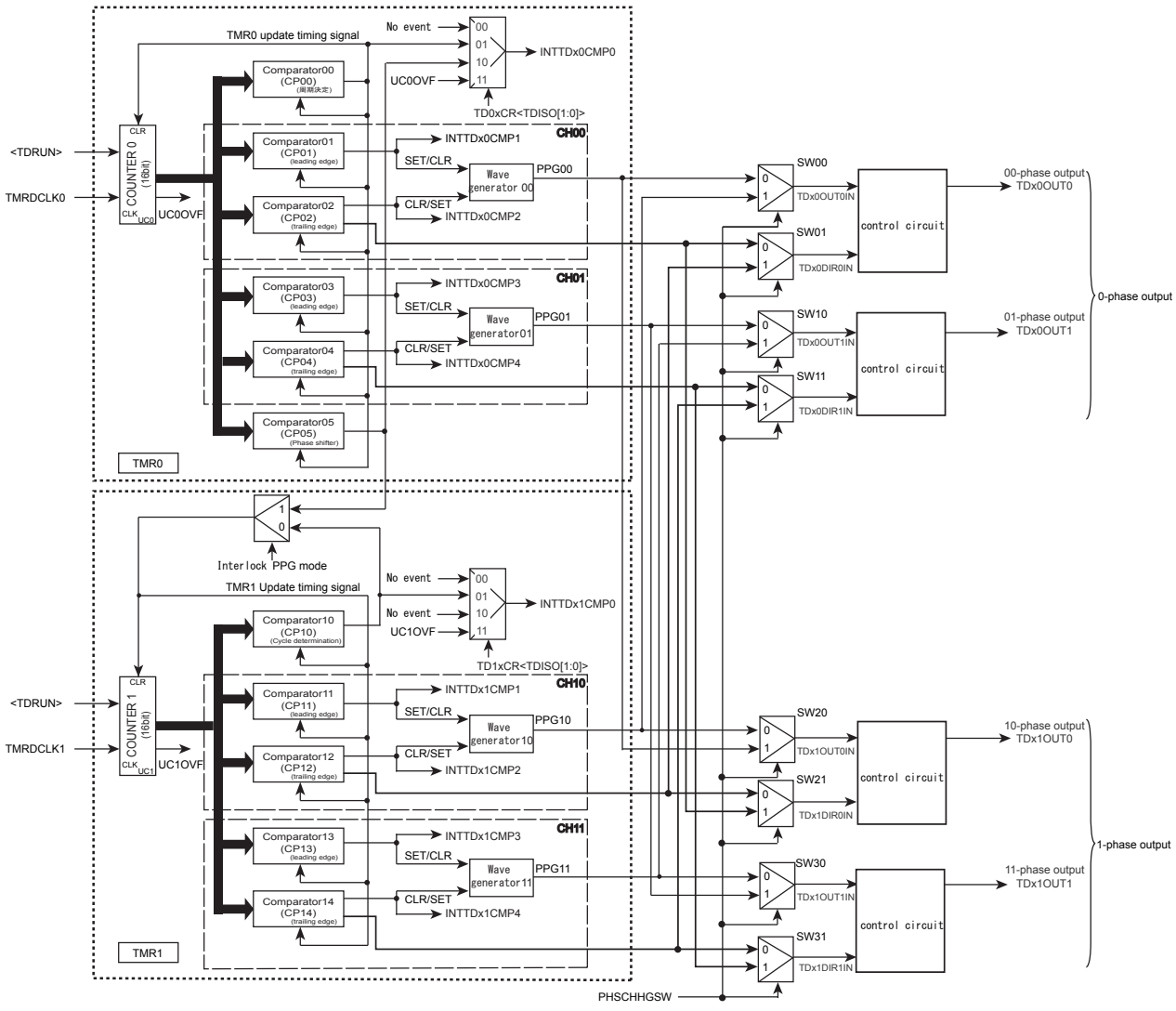


Figure 12-1 Block diagram of TMRD timer unit

12.3 Registers

12.3.1 Register List

12.3.1.1 Register List (TMRD)

The following are control registers and addresses.

For details of base addresses, refer to "Peripheral function base address list" of Chapter "Memory Map".

Register name		Address (Base+)
TMR0 timer RUN register	TD0RUN	0x0000
TMR0 timer control register	TD0CR	0x0004
TMR0 Timer mode register	TD0MOD	0x0008
TMR0 DMA request enable register	TD0DMA	0x000C
TMR0 timer Register 0	TD0RG0	0x0014
TMR0 timer register 1	TD0RG1	0x0018
TMR0 timer register 2	TD0RG2	0x001C
TMR0 timer register 3	TD0RG3	0x0020
TMR0 timer register 4	TD0RG4	0x0024
TMR0 timer register 5	TD0RG5	0x0028
TMR1 timer register 0	TD1RG0	0x002C
TMR1 timer register 1	TD1RG1	0x0030
TMR1 timer register 2	TD1RG2	0x0034
TMR1 timer register 3	TD1RG3	0x0038
TMR1 timer register 4	TD1RG4	0x003C
Update flag setting register	TDBCR	0x0040
Timer enable register	TDEN	0x0050
Timer config register	TDCONF	0x0054
Reserved	-	0x0060
Reserved	-	0x0064
Reserved	-	0x0068
Reserved	-	0x006C
TMR1 timer RUN register	TD1RUN	0x0100
TMR1 timer control register	TD1CR	0x0104
TMR1 Timer mode register	TD1MOD	0x0108
TMR1 DMA request enable register	TD1DMA	0x010C
TMR0 compare register 0	TD0CP0	0x0114
TMR0 compare register 1	TD0CP1	0x0118
TMR0 compare register 2	TD0CP2	0x011C
TMR0 compare register 3	TD0CP3	0x0120
TMR0 compare register 4	TD0CP4	0x0124
TMR0 compare register 5	TD0CP5	0x0128
TMR1 compare register 0	TD1CP0	0x012C
TMR1 compare register 1	TD1CP1	0x0130
TMR1 compare register 2	TD1CP2	0x0134
TMR1 compare register 3	TD1CP3	0x0138
TMR1 compare register 4	TD1CP4	0x013C
Reserved	-	0x0160
Reserved	-	0x0164
Reserved	-	0x0168
Reserved	-	0x016C

Note 1: Do not access the addresses that are noted as "Reserved".
Note 2: Read or write registers in the units of words.

(1) TDBCR (Update Flag 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	-	-	-	PHSCHG	TDSFT11	TDSFT10	TDSFT01	TDSFT00
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4	PHSCHG	R/W	Sets a relationship between 0-phase output and 1-phase output (advance/delay). 0: Delayed or same phase 1: Advanced or same phase Sets a relationship between 0-phase output and 1-phase output (advance/delay). The setting is reflected at the timing of boot by setting TD0RUN<TDRUN>=1. During operation, the setting is reflected at the timing of update triggered by <TDSFT00>.
3	TDSFT11	R/W	Update enable flag of TD1xCP3/TD1xCP4 (Updates of output channel CH11's comparator.) 0: Invalid 1: Update is enabled. This flag is an enable flag used to update a value of TD1RGm corresponding to TD1xCPm in TMR1. When data of compare register is updated, <TDSFT11> is cleared.
2	TDSFT10	R/W	Update enable flag of TD1xCP0/TD1xCP1/TD1xCP2 (Updates of output channel CH10's comparator and CP10) 0: Invalid 1: Update is enabled. This flag is an enable flag used to update a value of TD1RGm corresponding to TD1CPm in TMR1. When data of compare register is updated, <TDSFT10> is cleared.
1	TDSFT01	R/W	Update enable flag of TD0CP3/TD0CP4 (Updates of output channel CH01's comparator) 0: Invalid 1: Update is enabled. This flag is an enable flag used to update a value of TD0RGm corresponding to TD0CPm in TMR0. When data of compare register is updated, <TDSFT01> is cleared.
0	TDSFT00	R/W	Update enable flag of TD0CP0/TD0CP1/TD0CP2/TD0CP5 (Updates of output channel CH00's comparator and CP00) 0: Invalid 1: Update is enabled. This flag is an enable flag used to update a value of TD0RGm corresponding to TD0CPm in TMR0. When data of compare register is updated, <TDSFT00> is cleared.

Note 1: A status of update enable flag can be monitored by reading each register.

Note 2: <PHSCHG> is valid only in interlock PPG mode while TDCONF<TMRDMOD[2:0]>="111" is set. (In timer mode, interlock timer mode, PPG mode and interlock PPG mode, if TDCONF<TMRDMOD[2:0]>="110" is set, 0-phase and 1-phase outputs cannot be switched.)

(2) TDEN (Timer 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	TDEN1	TDEN0	TDHALT	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TDEN1	R/W	Clock supply operation to TMR1. 0: Stop (OFF) 1: Operation (ON) Sets on/off of the clock supply operation into TMR1.
6	TDEN0	R/W	Clock supply operation to TMR0. 0: Stop (OFF) 1: Operation (ON) Sets on/off of the clock supply operation into TMR0.
5	TDHALT	R/W	Operation setting during debug (up-counter at HALT) 0: Stop (Only up-counter stops.) 1: Operation (No up-counter stops.) Sets the operation when a HALT instruction occurs during debug.
4-0	−	R	Read as "0".

(3) TDCONF (Timer Config Register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	−	R/W	Reserved.
6	−	R/W	Reserved.
5-3	−	R	Read as "0".
2-0	TMRDMOD [2:0]	R/W	Operation mode selection Set the operation mode of TMR1 and TMR0.

<TMRDMOD[2:0]>	Operation mode of TMR0	Operation mode of TMR1
000 :	Reserved	Reserved
001 :	Reserved	PPG mode
010 :	PPG mode	Reserved
011 :	PPG mode	PPG mode
100 :	Reserved	
101 :	Prohibited	
110 :	Interlock PPG mode Update timing of CH00 in TMR0 is synchronous with CH10 in TMR1.	
111 :	Interlock PPG mode (Update timing of CH00 in TMR0 is synchronous with those of all channels of TMR1.)	

Note 1: When the timer mode or PPG mode (<TMRDMOD[2:0]>="000" to "100") is set, TMRDCLK0 and TMRDCLK1 can be set respectively.

Note 2: When the interlock mode (<TMRDMOD[2:0]>="110" to "111") is set, TMRDCLK0 and TMRDCLK1 cannot be set respectively. A value of TD1MOD<TCLK[3:0]> is ignored and a frequency of TMRDCLK1 becomes the same as those of TMRDCLK0.

(4) TDnRUN (TMRn Timer 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	-	-	-	-	-	-	-	TDRUN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	−	W	Always write "0".
0	TDRUN	W	TMRn operation 0: Stops TMRn operation (Stops the operation of COUNTERn (UCn) and initializes them to "0".) 1: Starts TMRn operation (Starts the operation (up-counting) of COUNTERn (UCn).) Controls the count operation of TMRn.

Note: In interlock PPG mode, the setting of TD1RUN<TDRUN> is invalid, and starts the interlocking operation with UC0.

(5) TDnCR (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	-	-	-	-	TDMDCYn1			TDMDPTn1
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TDMDCYn0			TDMDPTn0	-	TDRDE	TDISO	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	−	R	Read as "0".
11-9	TDMDCYn1 [2:0]	R/W	Select a cycle of 1-bit modulation in output channel CHn1. 000 : No 1-bit modulation function 001 : (A cycle defined with CPn0) × 2 010 : (A cycle defined with CPn0) × 4 011 : (A cycle defined with CPn0) × 8 100 : (A cycle defined with CPn0) × 16 Other than the above settings are prohibited. Selects a 1-bit modulation cycle in output channel CHn1 in the PPG and interlock PPG mode. (When n=0, PPG01 is set; when n=1, PPG11 is set.) A setting value to this register has no meaning in the timer mode or interlock timer mode. Note that when n=1, set the same value to TD1CR as TD0CR<TDMDCY00[2:0]> in the interlock PPG mode (TDCONF<TMRDMOD[2:0]>="111").
8	TDMDPTn1	R/W	Selects an update timing of TDnCP3 / TDnCP4 0: Every 1-bit modulation 1: A match detection with CPn0 Selects an update timing of DnCP3 or TDnCP4 value via timer registers in the PPG and interlock PPG mode (<TDRDE>=1). However, when <TDMDCYn1[2:0]>="000", the operation is the same as the operation in which "1" is set. Thus, a setting value to this register has no meaning. Note that when n=1, set the same value to TD1CR as TD0CR<TDMDPT00> in the interlock PPG mode (TDCONF<TMRDMOD[2:0]>="111"). A setting value to this register has no meaning in the timer mode or interlock timer mode. Depending on a value of TDnMOD0<TDCLE>, this register is updated at the following timings. TD0MOD<TDCLE>=0ÅFUpdated at which COUNTER0 overflows. TD0MOD<TDCLE>=1ÅFUpdated at which a match is detected in CP00.
7-5	TDMDCYn0 [2:0]	R/W	Selects a 1-bit modulation cycle in output channel CHn0. 000 : No 1-bit modulation function 001 : (A cycle defined with CPn0) × 2 010 : (A cycle defined with CPn0) × 4 011 : (A cycle defined with CPn0) × 8 100 : (A cycle defined with CPn0) × 16 Other than the above settings are prohibited. Selects a 1-bit modulation cycle in output channel CHn0 in the PPG and interlock PPG mode. (When n=0, PPG00 is set; when n=1, PPG10 is set.) Note that when n=1, set the same value to TD1CR as TD0CR<TDMDCY00[2:0]> in the interlock PPG mode (TDCONF<TMRDMOD[2:0]>="110" or "111"). A setting value to this register has no meaning in the timer mode or interlock timer mode.

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>4</td><td>TDMDPTn0</td><td>R/W</td><td>Select an update timing of TDnCP0/TDnCP1/TDnCP2 (when n=0, TD0CP5 is included.)
0: Each 1-bit modulation cycle
1: Match detection with CPn0
In the PPG and interlock PPG mode, when <TDRDE>=1 is set, update timings of TDnCP0,TDnCP1 and TDnCP2 (when n=0, TD0CP5 is included.) are selected.
However; when <TDMDCY00[2:0]>="000", the operation becomes the same as those when "1" is set, so that setting values to this register has no meaning.
In the interlock PPG mode (TDCONF<TMRDMOD[2:0]>="110" or "111") is set, set the same value to TD0CR<TDMDPT11> as those of TD0CR<TDMDPT00>.

A setting value to this register has no meaning in the timer mode or interlock timer mode. Values are updated at the following timing according to a value of TDnMOD0<TDCLE>.
TD0MOD<TDCLE>=0ÅFUpdated when COUNTER0 overflows.
TD0MOD<TDCLE>=1ÅFUpdated when a match in CP00 is detected.</td></tr>
<tr><td>3</td><td>−</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>2</td><td>TDRDE</td><td>R/W</td><td>Sets data writing path to the compare register (TDnCPm).
when n=1, a value specified in TD0CR<TDRDE> is selected in the interlock PPG mode.

0: Direct write by the CPU instruction
When a value is written to timer register (TDnRGm), this value is written to the corresponding compare register (TDnCPm) simultaneously.
At this time, TDBCR<TDSFTn0><TDSFTn1> is no need to set to "1" (update enable setting).
1: Writing via timer register (TDnRGm) in TMR0.
Sets TDBCR<TDSFTn0><TDSFTn1> to "1".
<table><tr><td>PPG mode/interlock PPG mode</td><td>A value of TDnMOD <TDCLE> has no meaning.</td><td>When a match is detected with CPn0, a value of timer registers (TDnRGm) is written to compare registers (TDnCPm).</td></tr></table></td></tr>
<tr><td>1-0</td><td>TDISO[1:0]</td><td>R/W</td><td>Interrupt event of INTTDnCMP0
00: No interrupt events
01: A match with CPn0
10: When n=0, a match with CP05 is detected; when n=1, there is no interrupt event.
11: When UCn overflows (In PPG mode, there is no interrupt event.)
IN Select an interrupt event of TTDnCMP0</td></tr>
</table>

(6) TDnMOD(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	TDIV1	TDIV0	-	TDCLE	TDCLK			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TDIV1	R/W	Sets the initial setting of PPGn1 on leading edge/trailing edge. (When n=0, PPG01 is set: when n=1, PPG11 is set.) 0: Rise when a match with CPn3 is detected and fall when a match with CPn4 is detected. 1: Fall when a match with CPn3 is detected, and rise when a match with CPn4 is detected. Selects a polarity of leading edge/trailing edge of output signal PPG01 in output channel CHn1.
6	TDIV0	R/W	Sets the initial setting of PPGn0 on leading edge/trailing edge. (When n=0, PPG00 is set; when n=1, PPG10 is set.) 0: Rise when a match with CPn1 is detected and fall when a match with CPn2 is detected. 1: Fall when a match with CPn1 is detected, and rise when a match with CPn2 is detected. Selects a polarity of leading edge/trailing edge of output signal PPG00 in output channel CHn0.
5	−	R	Read as "0".
4	TDCLE	R/W	UCn operation when a match with CPn0 is detected. 0: Operates as a free-running counter regardless of a match detection. 1: Initialized to "0" when a match is detected. Sets UCn operation when a match with CPn0 is detected.
3-0	TDCLK[3:0]	R/W	Selects a prescaler of TMRn 0000: ftmrd 1000: ftmrd/2 1001: ftmrd/4 1010: ftmrd/8 1011: ftmrd/16 Other than the above are prohibited. Selects a frequency of TMRDCLKn.

Note: A setting of <TDCLE>="0" is invalid in the PPG mode or interlock PPG mode. (It does not operate as a free-run counter.)

(7) TDnDMA (DMA Request 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	-	-	-	-	-	-	-	DMAEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4-1	−	R/W	Always write "0".
0	DMAEN	R/W	Set DMA request enable setting (INTTDnCMP0) 1: Enabled 0: Disabled Set DMA request to be enabled/disabled. (Issue DMA request by INTTDnCMP0 event.)

(8) TDnRG0 (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	TDRG0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TDRG0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-0	TDRG0[15:0]	R/W	For details of the setting range of the cycle, refer to "Table 12-4 Setting range of compare register in 16-bit programmable rectangular wave output (PPG)", and Chapter "Product Information.". 16-bit PPG :Sets a cycle of rectangular wave output (However, in interlock PPG mode, TD1CP0 is not used as a cycle setting register.)

(9) TDnRG1 (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	TDRG1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TDRG1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-0	TDRG1[15:0]	R/W	Timing setting 16-bit PPG : Sets the timing on leading edge of PPG00/PPG10 signal.

(10) TDnRG2 (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	-	-	-	-	TDRG2			
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TDRG2							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TDRG2				TDMDRT			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	−	R/W	Write as "0".
30-20	−	R	Read as "0".
19-4	TDRG2[15:0]	R/W	Timing setting 16-bit PPG : Sets a timing on trailing edge of PPG00/PPG10 in CHn0.
3-0	TDMDRT[3:0]	R/W	1-bit modulation rate setting Both in PPG mode and interlock PPG mode, sets 1-bit modulation rate for CHn0. Depending on a value of <TDMDCY00[2:0]><TDMDCY01[2:0]><TDMDCY10[2:0]><TDMDCY11[2:0]>, a valid value of this register will be different.

<TDMDCY**[2:0]>	Valid bit of <TDMDRT[3:0]>	Rate setting
000	: All bits are ignored.	None
001	:<TDMDRT[3]> is valid. Other bits settings are ignored.	0 to 1
010	:<TDMDRT[3:2]> is valid. Other bits settings are ignored.	0 to 3
011	:<TDMDRT[3:1]> is valid. Other bits settings are ignored.	0 to 7
100	:<TDMDRT[3:0]> is valid.	0 to 15

A rate is the number of rectangular waves that delays the timing of trailing edge by 1 clock of TMRDCLKn in 1-bit modulation cycles

(11) TDnRG3 (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	TDRG3							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TDRG3							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-0	TDRG3[15:0]	R/W	Timing setting 16-bit PPG : Sets the timing of leading edge of PG01/PPG11 signal

(12) TDnRG4 (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	-	-	-	-	TDRG4			
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TDRG4							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TDRG4				TDMDRT			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	−	R/W	Write as "0".
30-20	−	R	Read as "0".
19-4	TDRG4[15:0]	R/W	Timing setting 16-bit PPG : Sets the trailing edge Timing setting of PPG01/PPG11 output of CHn1.
3-0	TDMDRT[3:0]	R/W	1-bit modulation rate setting Both in PPG mode and interlock PPG mode, sets 1-bit modulation rate for CHn1. Depending on a value of <TDMDCY01[2:0]><TDMDCY11[2:0]>, a valid value of this register will be different.

<TDMDCY**[2:0]>	Valid bits of <TDMDRT[3:0]>	Rate setting
000	:All bits are ignored.	None
001	:<TDMDRT[3]> is valid but the other bit settings are ignored.	0 to 1
010	:<TDMDRT[3:2]> is valid but the other bit settings are ignored.	0 to 3
011	:<TDMDRT[3:1]> is valid but the other bit settings are ignored.	0 to 7
100	:<TDMDRT[3:0]> is valid.	0 to 15

A rate is the number of rectangular waves that delays the timing of trailing edge by 1 clock of TMRDCLKn in 1-bit modulation cycles.

(13) TD0RG5 (Timer 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	-	-	-	-	-	-	-	TDRG5
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TDRG5							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TDRG5							-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-17	−	R	Read as "0".
16-1	TDRG5[15:0]	R/W	Set timing setting and the amounts of phase shift 16-bit PPG : Set the amount of phase shift in the interlock PPG mode (In the PPG mode, this bit is invalid.)
0	−	R/W	Write as "0".

(14) TDnCP0 (Timer 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	CPRG0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CPRG0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-0	CPRG0[15:0]	R	For details of the setting range of the cycle, refer to "Table 12-4 Setting range of compare register in 16-bit programmable rectangular wave output (PPG)", and Chapter "Product Information.". In the interlock PPG mode, TD1CP0 is not used as cycle setting register.

Note: Compare register TDnCP0 has double-buffering structure with timer register TDnRG0. Timer registers are used to write the compare registers.

(15) TDnCP1 (Timer 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	CPRG1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CPRG1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-0	CPRG1[15:0]	R	Timing setting

Note: Compare register TDnCP1 has double-buffering structure with timer register TDnRG0. Timer registers are used to write the compare registers.

(16) TDnCP2 (Timer Compare 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	-	-	-	-	CPRG2			
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CPRG2							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CPRG2				CPMDRT			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-20	−	R	Read as "0".
19-4	CPRG2[15:0]	R	Timing setting register
3-0	CPMDRT[3:0]	R	1-bit modulation rate setting

Note: Compare register TDnCP2 has double-buffering structure with timer register TDnRG2. Timer registers are used to write the compare registers.

(17) TDnCP3 (Timer Compare 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	CPRG3							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CPRG3							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-17	−	R	Read as "0".
15-0	CPRG3[15:0]	R	Timing setting

Note: Compare register TDnCP3 has double-buffering structure with timer register TDnRG3. Timer registers are used to write the compare registers.

(18) TDnCP4 (Timer Compare 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	-	-	-	-	CPRG4			
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CPRG4							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CPRG4				CPMDRT			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-20	−	R	Read as "0".
19-4	CPRG4[15:0]	R	Timing setting
3-0	CPMDRT[3:0]	R	1-bit modulation rate setting

Note: Compare register TDnCP4 has double-buffering structure with timer register TDnRG4. Timer registers are used to write the compare registers.

(19) TD0CP5 (Timer Compare 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	-	-	-	-	-	-	-	CPRG5
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CPRG5							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CPRG5							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-17	−	R	Read as "0".
16-0	CPRG5[16:0]	R	Sets the timing and the amount of phase shift.

Note: Compare register TDnCP5 has double-buffering structure with timer register TDnRG5. Timer registers are used to write the compare registers.

12.4 Operation Description

12.4.1 Prescaler Clock

The prescaler clock TMRDCLKn operates the timer units. It selects a source clock for CG block by CGEXTENDO0 via PLL circuit. Clock setting register selects a prescaler clock for each timer units (TMR0/1) by TDnMOD<TDCLK[3:0]>.

Note that when interlock PPG mode is used, a value of TD1MOD<TDCLK[3:0]> is ignored and a frequency of TMRDCLK1 becomes a number defined in TD0MOD<TDCLK[3:0]>.

Note: A maximum operation frequency for timer units (TMRn) is 96MHz.

12.4.2 Timer Unit (TMR0, TMR1)

TMRD consists of 2 timer units (TMR0 and TMR1).

TMR0 consists of a 16-bit counter (UC0), 6 comparators (CP0m, m=0 to 5) and 2 output channels (CH00 and CH01).

TMR1 consists of a 16-bit counter (UC1), 5 comparators (CP1m, m=0 to 4) and 2 output channels (CH10 and CH11).

TMR0 and TMR1 can be interlocked at the cycle defined by CP05.

12.4.2.1 Counter (UCn)

UCn is a 16-bit binary counter that counts up at the clock (TMRDCLKn) output from clock setting block.

If "0" is set to TDnRUN<TDRUN>, the counter is reset and stops. If "1" is set to <TDRUN>, the counter starts.

If "0" is set to TDnMOD<TDCLE>, the counter operates as a free-running counter.

If "1" is set to <TDCLE>, the counter operates as one that is reset when a match with CPn0 is detected. However, UC1 is reset when a match with CP05 is detected in the interlock PPG mode, not with CP10.

12.4.2.2 Comparator (CPnm)

The timer unit 0 (TMR0) has 6 comparators (CP00 to CP05); The timer unit 1 (TMR1) has 5 comparators (CP10 to CP14).

The function of each comparator are as follows:

TMRD	Function
CP0	Determines a cycle of TMRn.
CP1,CP3,	Specifies a timing on leading edge of rectangular wave in PPG mode/interlock PPG mode.
CP2,CP4	Specifies a timing on trailing edge of rectangular wave in PPG mode/interlock PPG mode.
CP05	Specifies the relation of phases between TMR0 rectangular wave output and TMR1 rectangular wave output in the interlock PPG mode.

These comparators have timer registers (TDnRGm), compare registers (TDnCPm), write timing generation circuit and match detection circuit.

TDnRGm and TDnCPm have double-buffering structure. When data is written to TDnRGm, the same data is written to TDnCPm at the timing according to a value of TDnCR<TDRDE> shown as below.

TDnCR<TDRDE> = 0 :	When data is written to TDnRGm, the same value is written to TDnCPm simultaneously. However, TDnCPm is a read-only register; therefore direct writing specifying to this address cannot be performed. (An initial setting to TDnCPm can be changed freely.)
TDnCR<TDRDE> = 1 :	When update enable flag (TDBCR<TDSFT**>) is set to "1", a value of TDnRGm is written to TDnCPm at the specified timing. Details of update timing is described in Section Operation Mode.

Structures of each comparator are shown below.

(1) CP00, CP01, CP03, CP10, CP11, CP13

The comparators (CPnm) have timer registers (TDnRGm), compare registers (TDnCPm), write timing generation circuit n and match detection circuit nm. (n=0, 1, m=0, 1, 3)

TDnRGm	: 16-bit timer register
TDnCPm	: 16-bit compare register
Write timing generation circuit n	: Generate the timing to write a value of each timer register to each compare register.
Match detection circuit nm	: Detects a match between a counter output value of UCn and TDnCPm<CPRGm[15:0]>.

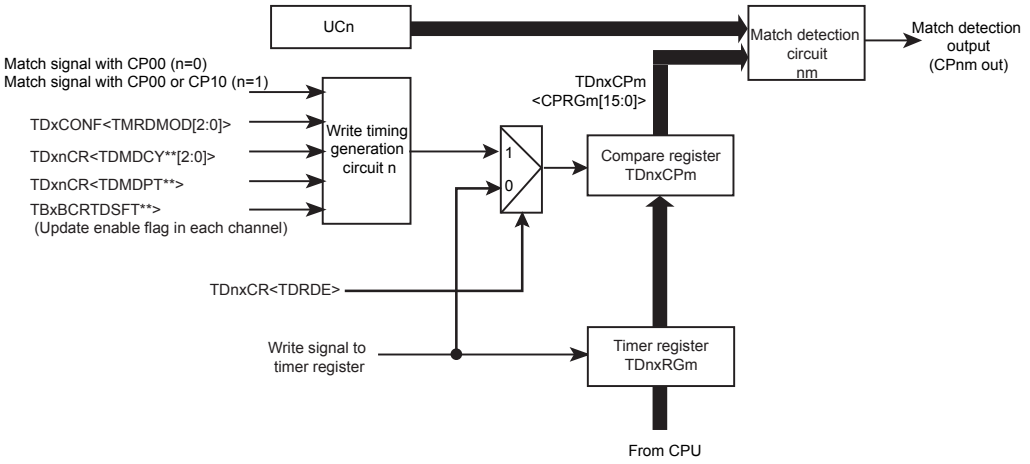


Figure 12-2 Configuration of comparators (CP00, CP01, CP03, CP05, CP10, CP11, CP13)

(2) CP02, CP04, CP12, CP14

The comparators (CPnm) have timer registers (TDnRGm), compare registers (TDnCPm), write timing generation circuit n, match detection circuit nm and 1-bit modulation setting circuit nm. (n=0,1, m=2,4)

The match detection circuit output nm is shifted by 1 clock depending on control output signal levels of 1-bit modulation setting circuit nm.

TDnRGm	:	16-bit timer register
TDnCPm	:	By setting 16-bit compare register (<CPRGm[15:0]>)<CPMDRT[3:0]>, this register operates as equivalent to 20-bit compare register.
Write timing generation circuit n	:	Generates the timing at which a value of each timer register is written to each compare register.
1-bit modulation setting circuit nm	:	Operates in the PPG/interlock PPG mode. Within1-bit modulation cycle, this circuit outputs control signals to the match detection circuit mn. The number of rectangular cycles (CP00/CP10) is specified using CDnCPm<CPMDRT[3:0]>. The timing is widen by 1 clock adding to a value specified with TDnCPm<CPRGm[15:0]>.
Match detection circuit nm	:	Generates the following 2 signals and outputs a match signal based on the control signal input from 1-bit modulation setting circuit nm. ・ A signal output at the timing of match between an output value of UCn and a value specified in TDnCPm<CPRGm[15:0]>. ・ A output signal shifted by 1 clock of TMRDCLKn This signal is shifted by 1clock of TMRDCLKn that is a match output signal between UCn and TDnCPm<CPRGm[15:0]>.

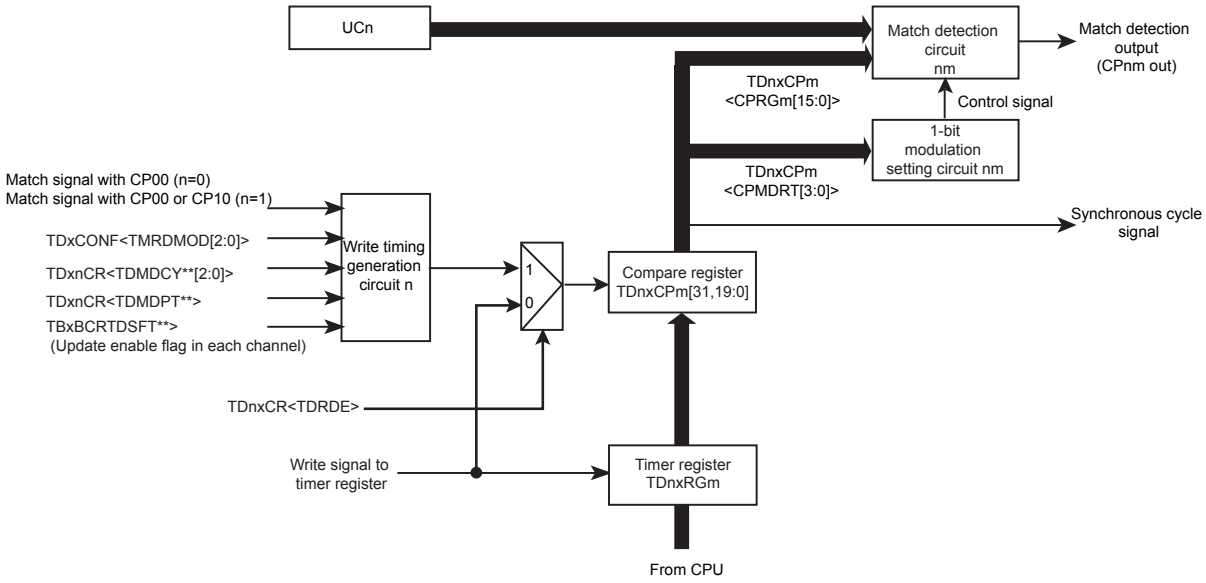


Figure 12-3 Configuration of comparators (CP02, CP04, CP12, CP14)

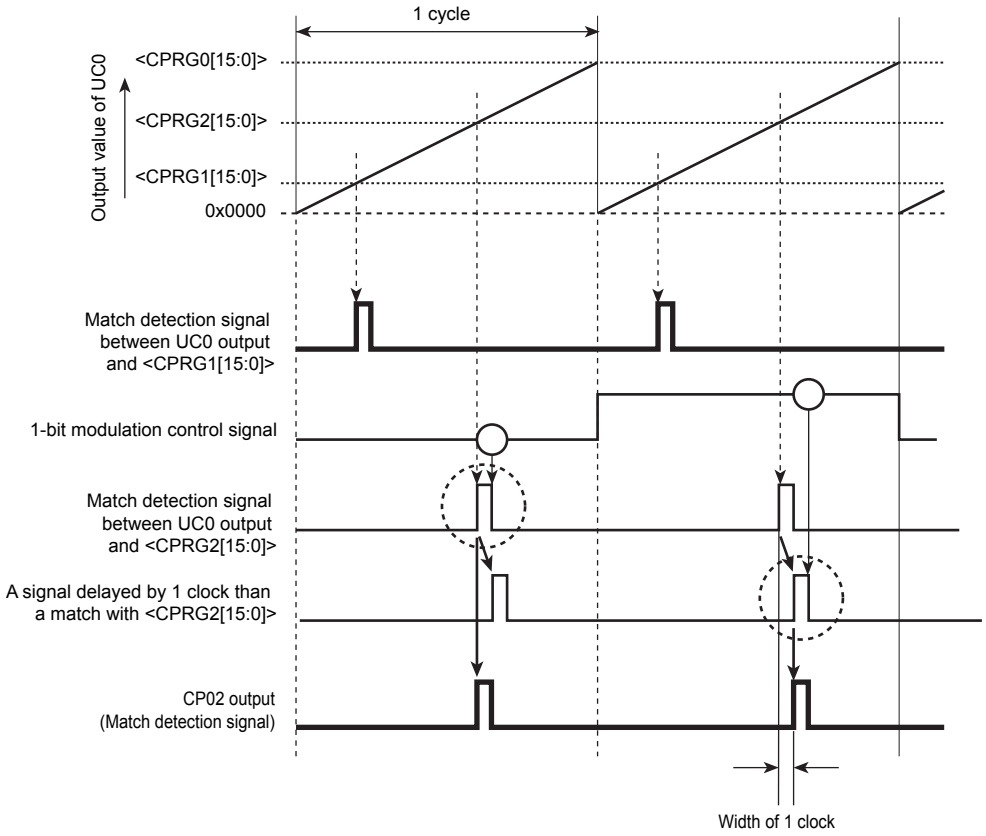


Figure 12-4 Variation of match detection signal by control output signals of 1-bit modulation setting circuit nm

(3) CP05

Comparators (CPnm) consists of timer registers (TDnRGm), compare registers (TDnCPm), write timing generation circuit (n) and match detection circuit (nm). (n=0, 1, m=0, 1, 3)

TDnRGm	: 16-bit timer register
TDnCPm	: 16-bit compare register
Write timing generation circuitn	: Generates the timing at which a value of each timer register is written to each compare register.
Match detection circuit nm	: Detects a match between an output value of UCn and TDnCPm<CPRGm [16:0]>.

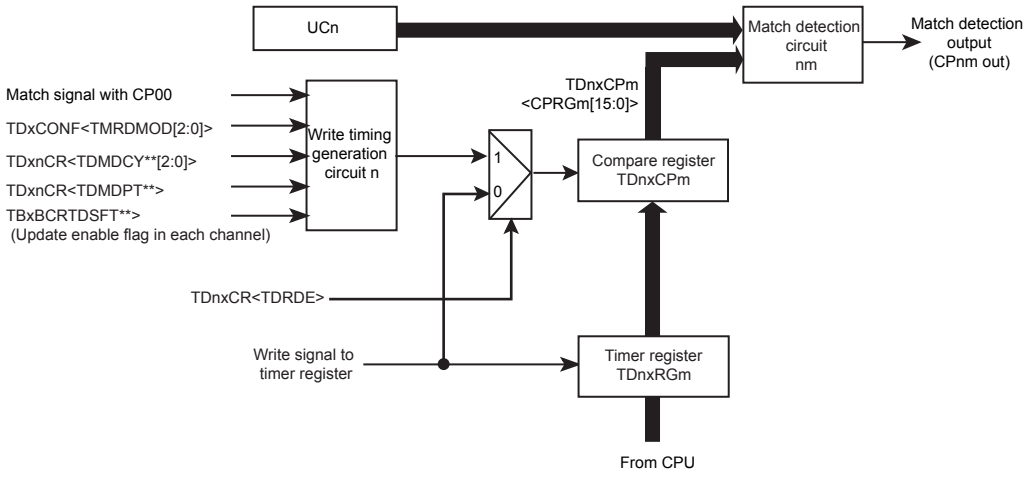


Figure 12-5 Configuration of comparator (CP05)

12.4.2.3 Output Channel (CHn0, CHn1)

Output channels (CHn0, CHn1) have 2 sets of comparator (CPn1 and CPn2; CPn3 and CPn4) and the wave generation circuits (WGn0, WGn1).

The wave generation circuits (WGn0, WGn1) generate the leading edge of CHn0 at the timing when a match with CPn1 is detected (When n=0, PPG00 is set; when n=1, PPG10 is set.) and the trailing edge of CHn0 is generated at the timing when a match with CPn2 is detected. The leading edge of CHn1 output is generated at the timing when a match with CPn3 is detected (When n=0, PPG01 is set; when n=1, PPG11 is set.) and the trailing edge of CHn1 is generated at the timing when a match with CPn4 is detected.

By using TDnMOD<TDIV1> and <TDIV0>, a polarity of leading/trailing edge of CHn0 and CHn1 can be set.

12.5 Each Operation Mode Description

An operation mode of TMRn is determined by the setting of TDCONF<TMRDMOD[2:0]>.

- 16-bit programmable rectangular wave output (PPG)
  - PPG mode
  - Interlock PPG mode

For a relationship between register setting values of TDCONF<TMRDMOD[2:0]> and each operation mode of timer units, refer to "(3) TDCONF (Timer Config Register)".

Each mode is described in the following sections.

12.5.1 16-bit Programmable Rectangular Pulse Output (PPG)

12.5.1.1 PPG mode

In this PPG mode, TMR0 and TMR1 operate independently and the frequency and duty for each timer can be set to output rectangular waves.

TMR0 and TMR1 have output pins from 2 systems. Each timer can output the same frequency rectangular wave.

	Output channel	Rectangular wave output pin
TMR0	CH00	PPG00
	CH01	PPG01
TMR1	CH10	PPG10
	CH11	PPG11

(1) Cycle of rectangular wave output (PPG)

In the TMR0, a cycle of rectangular wave output (PPG00/PPG01) can be defined with a value of TD0CP0<CPRG0[15:0]> in the compare register CP00.

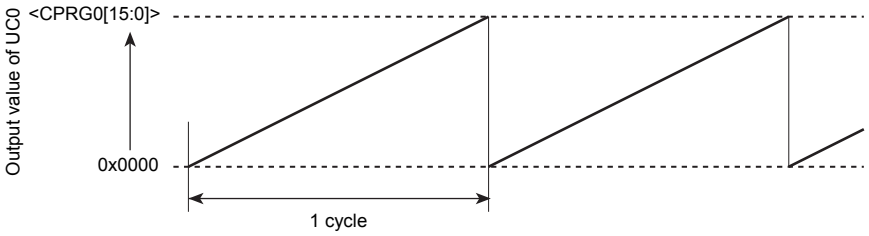


Figure 12-6 Cycle of rectangular wave in PPG mode

Also in the TMR1, a cycle of rectangular wave output (PPG10/PPG11) can be defined with a value of TD1CP0 <CPRG0[15:0]> in the compare register CP10.

(2) Duty of rectangular wave output (PPG)

In the TMR0, a leading edge of rectangular output (PPG00) is used to generate a rectangular wave when a match with CP01 is detected, and a trailing edge of those output is used to generate a rectangular wave when a match with CP02 is detected.

Same as the above mentioned, a leading edge of rectangular wave output (PPG01) is used to generate a rectangular wave when a match with CP03 is detected, and a trailing edge of those output is used to generate a rectangular wave when a match with CP04 is detected.

Same as TMR0, TMR1 uses a leading edge of rectangular wave output (PPG10/PPG11) to generate a rectangular wave when a match with CP11/CP13 is detected, and a trailing edge of those to generate a rectangular wave when a match with CP12/CP14 is detected.

: shows a timing of rectangular wave generation.

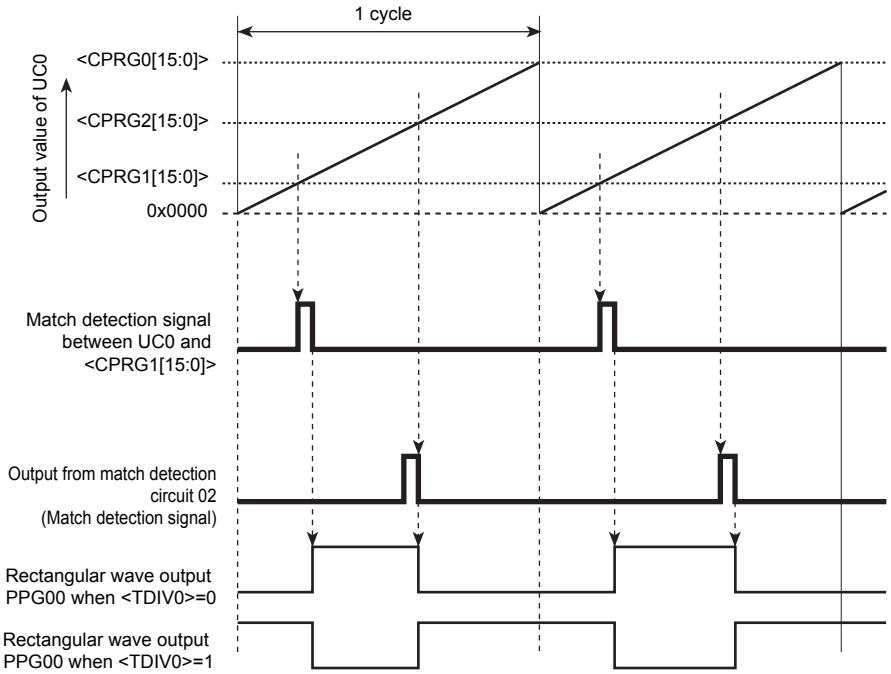


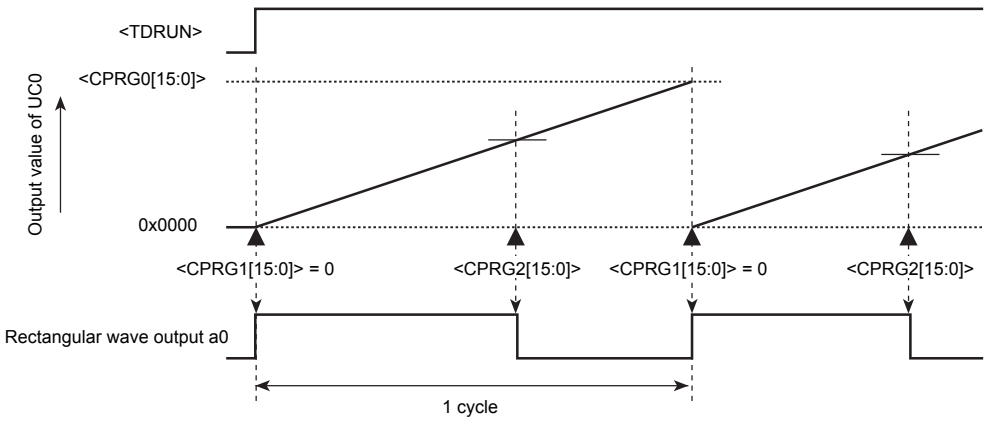
Figure 12-7 Rectangular wave output in PPG mode (Case of PPG00)

In the CP01, match detection circuit 01 outputs a match detection signal when a value of UC0 and <CPRG1[15:0]> are matched.

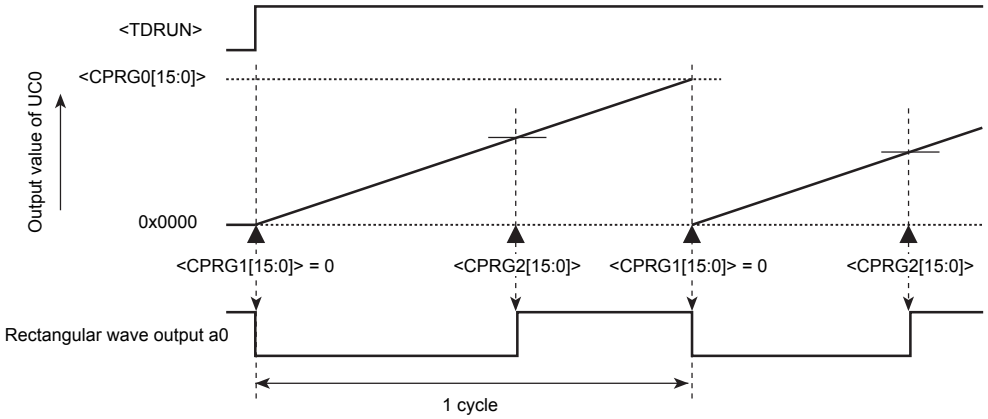
When TD0MOD<TDIV0>=0 is set, PPG00 is output on rising edge of CP01's leading edge; a match detection signal with CP02 is output on trailing edge. When TD0MOD<TDIV0>=1 is set, a match detection with CP01's leading edge is output on falling edge; a match detection with CP02 is output on rising edge.

In addition, "0x0000" can be set to CPn1 and CPn3, thus leading edge is output immediately at the start of TMR0 or TMR1.

In the setting of leading edge on rising edge/falling edge immediately at start, any rectangular wave can be output from the first cycle shown as Figure 12-8.



(1) Leading edge is specified on rising edge



(2) Leading edge is specified on falling edge

Figure 12-8 Timing chart at start (PPG00)

In the PPG mode, PHSCHGSW, which is a control signal of SW0/SW01/SW10/SW11/SW20/SW21/SW30/SW31, is fixed to "0". (Refer to "Figure 12-1 Block diagram of TMRD timer unit":)

(3) 1-bit modulation function

When PPG output is used as PWM output, 1-bit modulation function can be used to obtain pseudo-higher resolutions.

In the 1-bit modulation function, a duty wave is output widening by one clock of TMRDCLKn than the number of cycles defined in each 1-bit modulation cycle.

Thanks to this operation, an average resolution can be improved on each 1-bit modulation cycle.

A cycle of 1-bit is an integral multiple of cycle defined by CP00/CP10. This cycle can be set in each channel using <TDMDCY**[2:0]>.

If "000" is set to <TDMDCY**[2:0]>, "No 1-bit modulation" is set.

	Output channel	PWM cycle: N (specified in CP00/CP10)	1-bit modulation cycle (multiplied by 2/4/8/16)	Pulse width : N'
TMR0	CH00	<CPRG0[15:0]>	<TDMDCY00[2:0]>	TD0CP2<CPRG2[15:0]> − TD0CP1<CPRG1[15:0]>
	CH01	<CPRG0[15:0]>	<TDMDCY01[2:0]>	TD0CP4<CPRG4[15:0]> − TD0CP3<CPRG3[15:0]>
TMR1	CH10	<CPRG0[15:0]>	<TDMDCY10[2:0]>	TD1CP2<CPRG2[15:0]> − TD1CP1<CPRG1[15:0]>
	CH11	<CPRG0[15:0]>	<TDMDCY11[2:0]>	TD1CP4<CPRG4[15:0]> − TD1CP4<CPRG3[15:0]>

Also, using <CPMDRT[3:0]>, set a PWM cycle, whose duty wave is widen by 1 clock of TMRDCLKn.

Setting value of <TDMDCY**[2:0]>	Setting value of <CPMDRT[3:0]> [3]	[2]	[1]	[0]	Description
000	Invalid				No 1-bit modulation function
001	0	Invalid			No 1-bit modulation function
	1				Widen a duty wave by 1 clock to the 1st cycle.
010	0	0	Invalid		No 1-bit modulation function
	0	1			Widen a duty wave by 1 clock to the 1st cycle.
	1	0			Widen a duty wave by 1 clock to the 1st and 2nd cycles.
	1	1			Widen a duty wave by 1 clock to the 1st, 2nd and 3rd cycles.
011	0	0	0	Invalid	No 1-bit modulation function
	0	0	1		Widen a duty wave by 1 clock to the 1st cycle.
	0	1	0		Widen a duty wave by 1 clock to the 1st and 2nd cycles.
	. .				. .
	1	1	0	Invalid	Widen a duty wave by 1 clock to the 1st to 6th cycles.
	1	1	1		Widen a duty wave by 1 clock to the 1st to 7th cycles.
100	0	0	0	0	No 1-bit modulation function
	0	0	0	1	Widen a duty wave by 1 clock to the 1st cycle.
	0	0	1	0	Widen a duty wave by 1 clock to the 1st and 2nd cycles.
	. .				. .
	1	1	1	0	Widen a duty wave by 1 clock to the 1st to 14th cycles.
	1	1	1	1	Widen a duty wave by 1 clock to the 1st to 15th cycles.

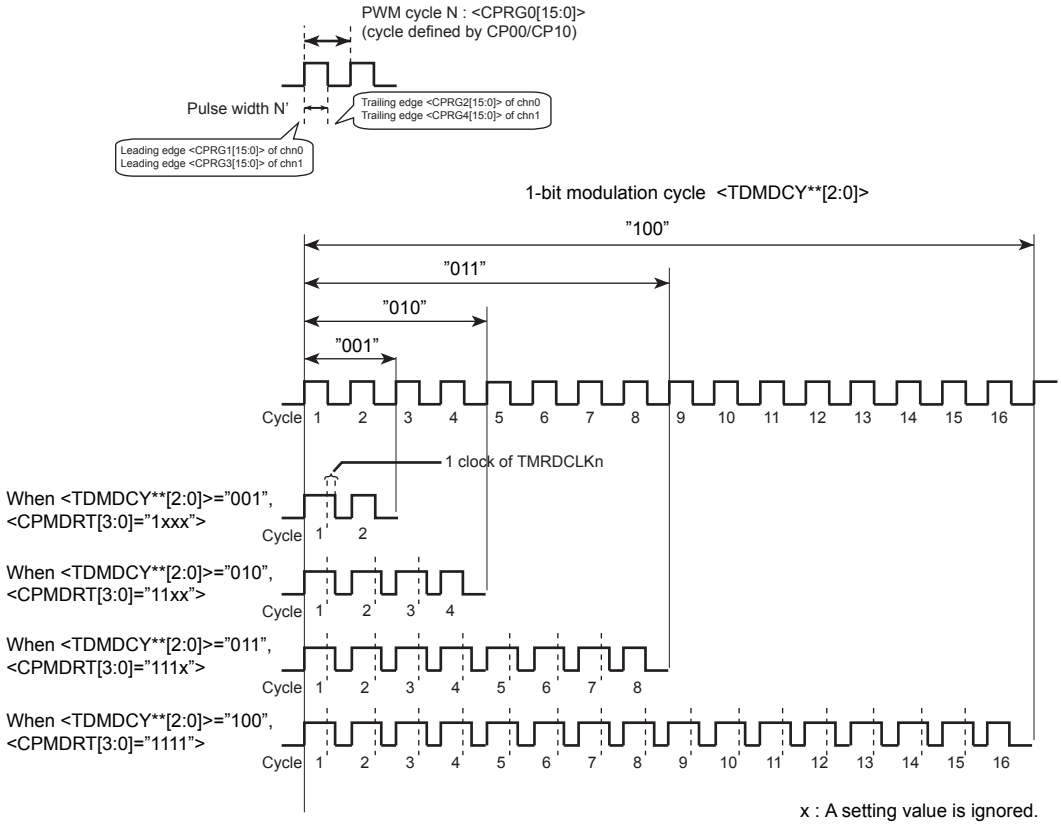


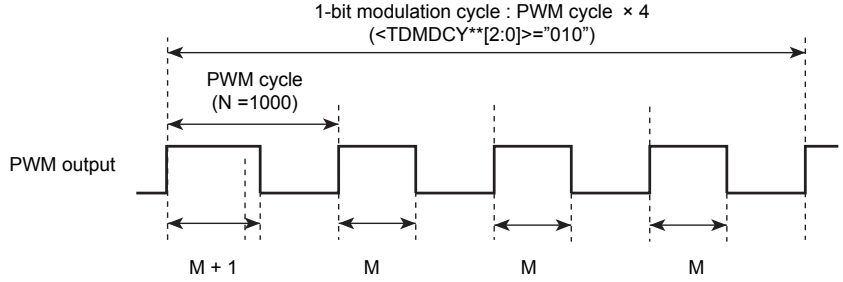
Figure 12-9 Schematic diagram of 1-bit modulation function

The following figure shows an example that a resolution becomes fourfold without any change of f_{PWM} (PWM frequency) when a 1-bit modulation cycle is the same as fourfold of PWM cycle.

To increase the resolution 4 times, the first PWM wave among 4 PWM waves in 1-bit modulation cycle is widen by 1 clock as duty wave.

Thus, a duty becomes (601 + 600 x 3) / 4 = 600.25 on average, and PWM output gives a 4 times of resolution. This example achieved a pulse width (Duty) of (601 + 600 x 3)/4 = 600.25 on average.

1-bit modulation cycle (Four cycles of PWM cycle)	: <TDMDCY**[2:0]> = "010"
Resolution of PWM cycle	: N = 1000
Pulse width setting	: M = 600
The number of PWM cycles to execute 1-bit modulation.	: <CPMDRT[3:0]> = "01xx" (x : don't care)



<CPMDRT[3:0]> = "01xx" : Add 1-bit modulation to the 1st cycle.

(4) Update timing in PPG mode

This item describes how to update a value of TDnCPm in each compare register.

The direct write mode (TDnCR<TDRDE>="0") is the same operation as those in the timer mode, so the description is omitted.

When TDnCR<TDRDE>="1" is set, an update timing is different depending on a value of TDnCR<TDMDCY**><TDMDPT**>.

Table 12-2 Comparators and TDnCR<TDMDCY**><TDMDPT**>

Timer unit	Registers	Comparators
TMR0	TD0CR<TDMDCY00[2:0]><TDMDPT00>	CP00/CP01/CP02/(CP05)
	TD0CR<TDMDCY01[2:0]><TDMDPT10>	CP03/CP04
TMR1	TD1CR<TDMDCY10[2:0]><TDMDPT10>	CP10/CP11/CP12
	TD1CR<TDMDCY11[2:0]><TDMDPT11>	CP13/CP14

The following describes each timer unit.

Update timings of all comparators are the same, thus only CPn0, CPn1 and CPn2 are described in this item.

(a) No 1-bit modulation function (<TDMDCY**[2:0]>="000")

When 1-bit modulation function is not used, values of CPn0 comparator are updated. In no 1-bit modulation function, TDnCP0, TDnCP1 and TDnCP2 of CPn1/CPn2 compare registers in CHn0 are also updated toTDnRG0, TDnRG1 and TDnRG2 respectively at the timing shown:

TDBCR<TDSFTn0> is an update enable flag corresponding to CPn0, CPn1 and CPn2. When this signal is "1", values of compare registers updated when a match signal of CPn0 (update timing signal of TMRDn) is set to "1".

TDBCR<TDSFTn0> is cleared at this update timing.

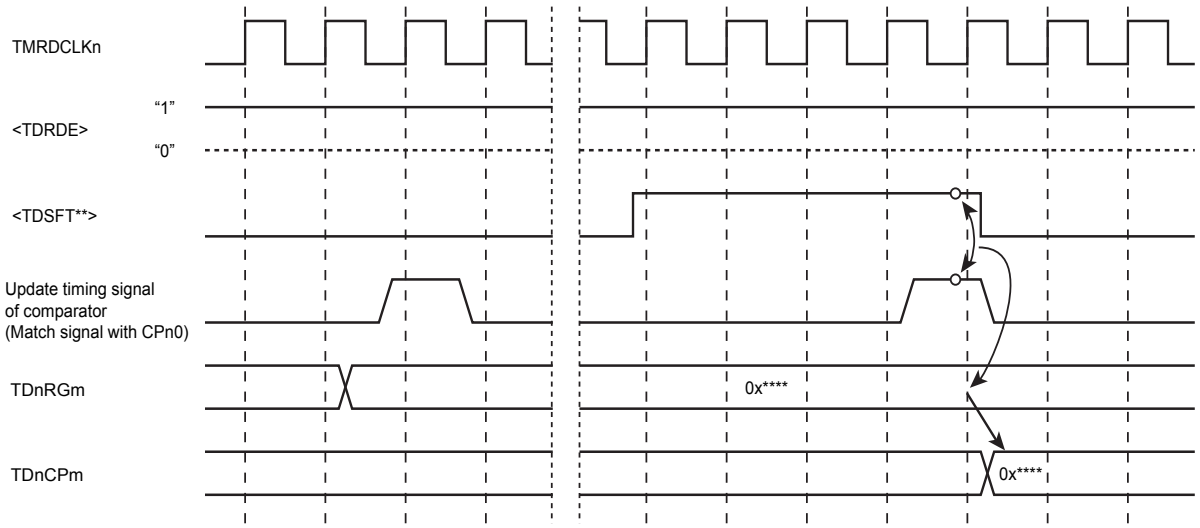


Figure 12-10 Update timing of CPn0,CPn1,CPn2 in PPG mode (<TDMDCY**[2:0]> = "000")

The relation between each update enable flag and corresponding comparator is as following.

- an update timing of the compare register TDnCPm:

  When TDnCR<TDRDE> is set to "0", values in TDnRGm and corresponding values in TDnCPm are updated simultaneously.

  :Figure 12-11 shows an update timing when TDnCR<TDRDE> is set to "1".

  When TDBCR<TDSFT**> is set to "1", a value in is updated at the timing when the comparator is updated to "1". TDBCR<TDSFT**> is cleared at the timing when when the comparator is updated to "1".

  In timer mode or interlock mode, update timings of TMR0 and TMR1 are as follows:

  - TMR0 : A timing when a match with CP00 is detected.
  - TMR1 : A timing when a match with CP10 is detected.

  Figure 12-11 shows a timing waveform when a value in TDnRGm is written to TDnCPm while TDnCR<TDRDE> is set to "1".

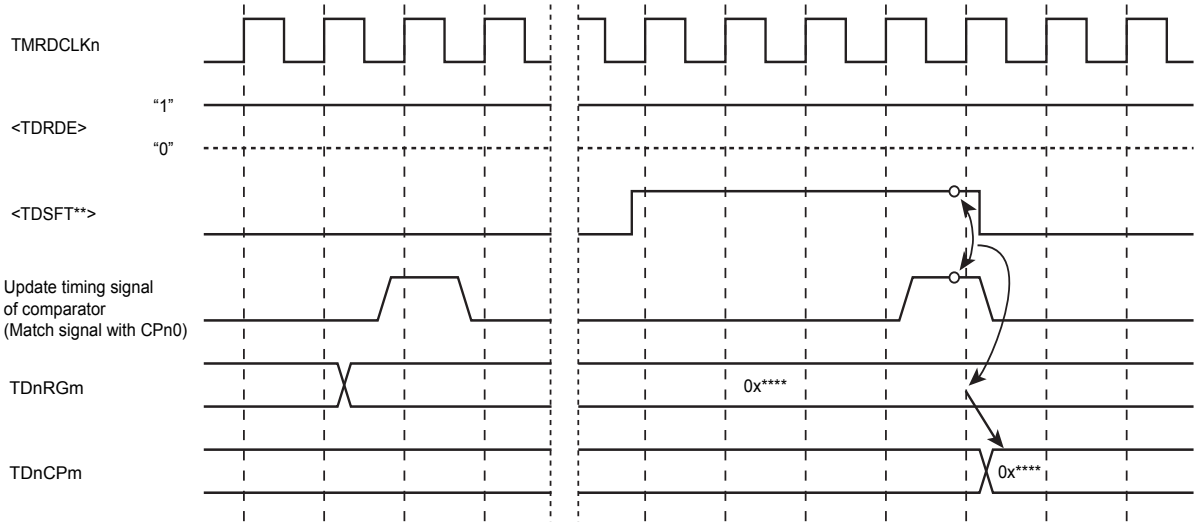


Figure 12-11 Write timing when a value in TDnRGm is written to TDnCPm (TDnCR<TDRDE>=1)

(b) With 1-bit modulation function (except <TDMDCY**[2:0]>="000")

When 1-bit modulation function is used, following two update timing can be chosen with the setting of TDnCR<TDMDPT**>.

1. Update timing when <TDMDPT**> = 0 is set.

   : shows the update timing when <TDMDCY**[2:0]>="010" is set.

   This figure shows that values of compare registers are updated at the timing when 1-bit modulation is complete while TDBCR<TDSFT**> is set to "1".

   is cleared to "0" at the timing when the setting values of compare registers are updated.

   When next values of TDnCP0, TDnCP1 and TDnCP2 are updated, a timing of leading/trailing edge of output signals (PPG00/PPG10) and a cycle of rectangular wave output of the output signal are changed according to values of TDnCP0, TDnCP1 and TDnCP2.

   In the PPG mode, if CH00 and CH01, or CH10 and CH11is synchronized (update at the same time), at the same time set the update enable flag <TDSFT00> and <TDSFT01>, or <TDSFT10> and <TDSFT11>.

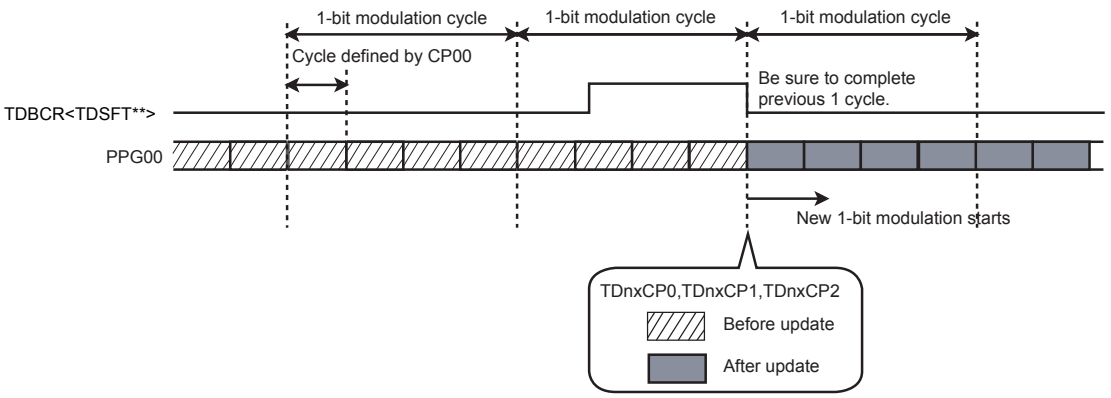


Figure 12-12 Update timing when 1-bit modulation is used in PPG mode (CPn0,CPn1,CPn2 : <TDMDCY**[2:0]>="010", <TDMDPT**>="0")

2. Update timing when TD0CR<TDMDPT00> = 1 is set.DPT00> = 1 is set.

   : shows that an example when TD0CR<TDMDCY00[2:0]> = "010" is set.

   This figure shows that a value of compare register is updated at the end of current 1-bit modulation cycle defined with CP00 is complete while TDBCR<TDSFT00> is set to "1". After an update, new 1-bit modulation cycle starts.

   TDBCR<TDSFT**> is cleared to "0" when a value of compare register is updated.

   When next values of TDnCP0, TDnCP1 and TDnCP2 are updated, a timing of leading/trailing edge of output signals (PPG00/PPG10) and a cycle of rectangular wave output of the output signal are changed according to this next values.

   In the PPG mode, if CH00 and CH01, or CH10 and CH11is synchronized (update at the same time), at the same time set the update enable flag <TDSFT00> and <TDSFT01>, or <TDSFT10> and <TDSFT11>.

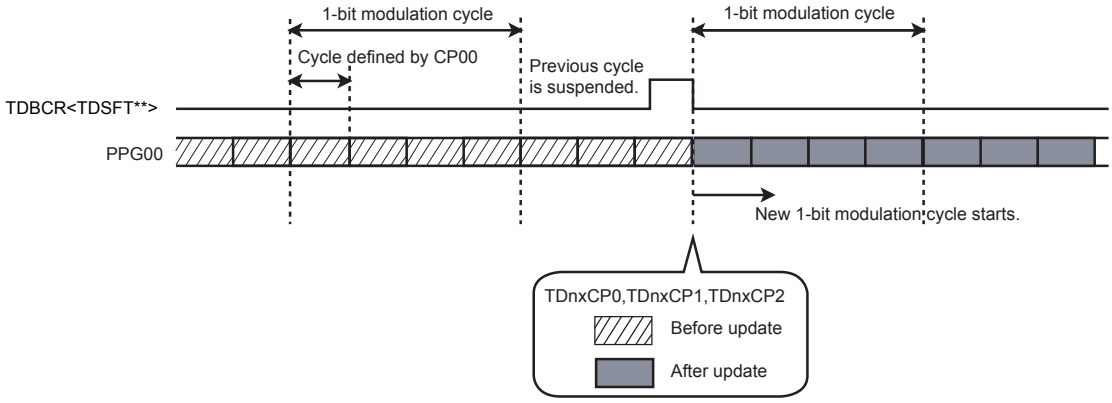


Figure 12-13 Update timing of 1-bit modulation in PPG mode

(5) Rectangular wave output (PPG) start timing

A rectangular wave output (PPG) is started when TDnRUN<TDRUN> is written to "1".

(6) Register setting procedure in PPG mode

(a) How to startup PPG mode

To startup PPG mode, set registers in the following procedures.

a. Supply clock selection

1. Select a supply clock for TMRD using CGEXTENDO0 in CG register.
2. For more information, refer to the setting of "CG" chapter.

b. Operation mode selection in each timer unit

1. Specify the timer unit in the PPG mode with TDCONF<TMRDMOD[2:0]>.

c. Select a supply clock in each timer unit (TMRn)

1. Set "1" to <TDENn> of TDEN that enables clock supply for each timer unit.ÅiNo need to set unnecessary timer unit to "1".
2. Select a clock for each timer unit using TDnMOD<TDCLK[3:0]>.
   At the same time, select either leading edge or trailing edge using <TDIVn>.

d.Set the operation mode in PPG and set an initial value setting of each timer register/compare register.

1. Set 1-bit modulation function, its cycle and update timing in each channel using TDnCR <TDMDCYn1[2:0]>, <TDMDPTn1>, <TDMDCYn0[2:0]> and <TDMDPTn0>.
   Set "0" to TDnCR<TDRDE> to change the write mode to compare registers to the direct write mode. At the same time, set an interrupt event of INTTDnCMP0 using TDnCR<TDISO[1:0]>.
2. Set a value to each timer register TDnRGm).
   In the direct write mode, an operation and update timing signal to set TDBCR<TDSFT**> to "1" is not required.

e. Startup PPG mode

1. Set "1" to TDnCR<TDRDE> to change the write path to compare registers to through timer registers. (PPG mode is one that a value of timer registers is written to the corresponding compare registers at the specified timing.)
2. Set TDnRUN<TDRUN> = "1" to startup

(b) How to update timer registers and compare registers

To update timer registers and compare registers, set registers in the following procedures.

1. Set any value to the timer register (TDnRGm) corresponding to the compare register to be updated.
2. After above setting, set "1" to TDBCR<TDSFT**> corresponding to a compare register to be updated.

By the above procedure, values in the timer register are set to the corresponding compare registers at specified timing.

Note: When an update timing is changed, change TDnCR<TDMDPTn1> and <TDMDPTn0> of corresponding compare registers.

Note: During PPG in operation, four registers such as TDnRGm, TDBCR, TDnRUN, TDnCR<TDMDPTn1><TDMDPTn0> can be modified (or written). The other registers must be modified while PPG stops

(c) How to stop rectangular wave output (PPG)

To stop rectangular wave output, set registers in the following procedures.

1. When rectangular wave outputs in each timer unit are stopped, set TDnRUN<TDRUN> register to = "0".

Note: During PPG in operation, four registers such as TDnRGm, TDBCR, TDnRUN, TDnCR<TDMDPTn1><TDMDPTn0> can be modified (or written). The other registers must be modified while PPG stops.

12.5.1.2 Interlock PPG Mode

In the interlock mode, timer units, TMR0 and TMR1, operate together through the phase shift function.

Adding to the normal PPG mode function, interlock mode can control the following operations:

1. Dynamically sets the phase relation between 0-phase output by TMR0 and 1-phase output by TMR1 using CP05 comparator within the range of -180° < θ < +180°.
2. Outputs synchronizing 3-channel or 4-channel rectangular wave output (PPG).

(1) Phase shifter function

Phase shifter function is one that shifts the relation between 0-phase of TMR0 and 1-phase of TMR1 using CP05.

In the interlock PPG mode, UC1's clearing timing and update timing of CP11 through CP14 are determined by a match timing with CP05 changing from CP10.

Therefore, a cycle of TMR1 is determined by a match with CP00 and rectangular wave output (PPG) is shifted by the amount of CP05.

Cycles of 0- and 1-phase are determined by a setting value of TD0CP0<CPRG0[15:0]> register in CP00, so that the amount (θ) of phase shift (delay) is determined by the following formula.

θ = 360° × (<CPRG5[16:1]> ÷ <CPRG0[15:0]>)

In the interlock PPG mode, the setting range of <CPRG5[16:1]> is 0x0000 ≤ CPRG5[16:1] ≤ CPRG0[15:0]÷ 2, so that the amount of phase shift can be set within the range of 0° ≤ θ < 180°.

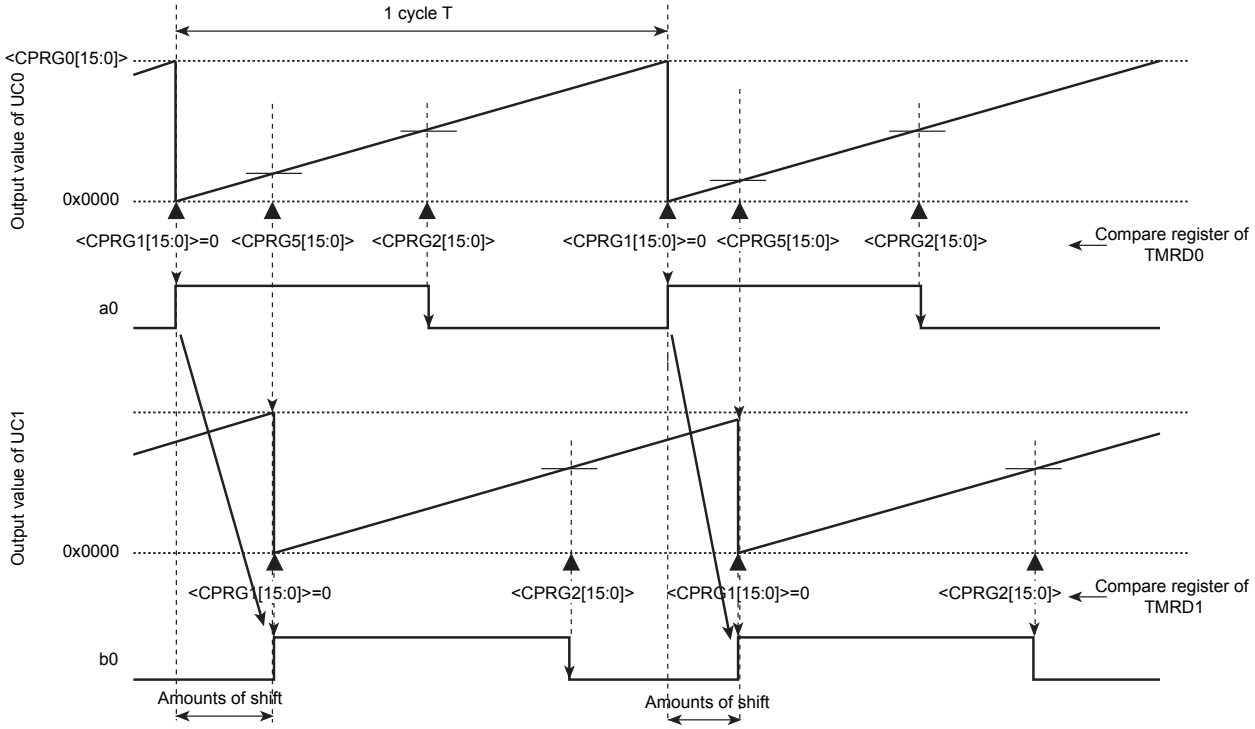


Figure 12-14 Example of phase relation between 0-phase (PPG00) and 1-phase (PPG10)

(2) Anti phase output

An anti phase output is set by TDBCR<PHSCHG>.

If <PHSCHG> is updated, the internal signal (PHSCHGSW) is changed and outputs of SW00 through SW31 are switched. As a results, 0-phase and 1-phase outputs are switched.

<PHSCHG> is ignored except when TDCONF<TMRDMOD[1:0]> is set to "111".

TDBCR<PHSCHG	Phase delay/advance	0-phase output	1-phase output
0	1-phase is delayed than 0-phase.	PPG00, PPG01	PPG10, PPG11
1	1-phase is advanced to 0-phase.	PPG10, PPG11	PPG00, PPG11

PHSCHGSW is changed according to a value of <PHSCHG> when TD0RUN<TDRUN> is set to "1", or when TDBCR<TDSFT00> is set to "1" and a timing when a match with CP00 is detected.

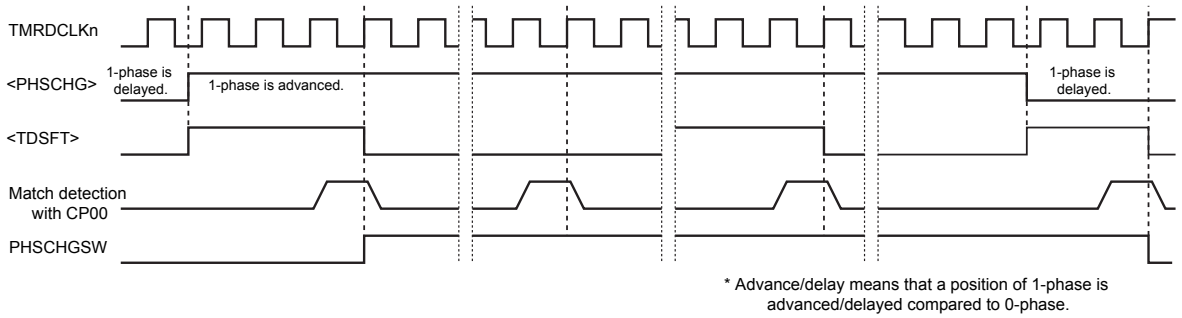


Figure 12-15 Switch timing of PHSCHGSW

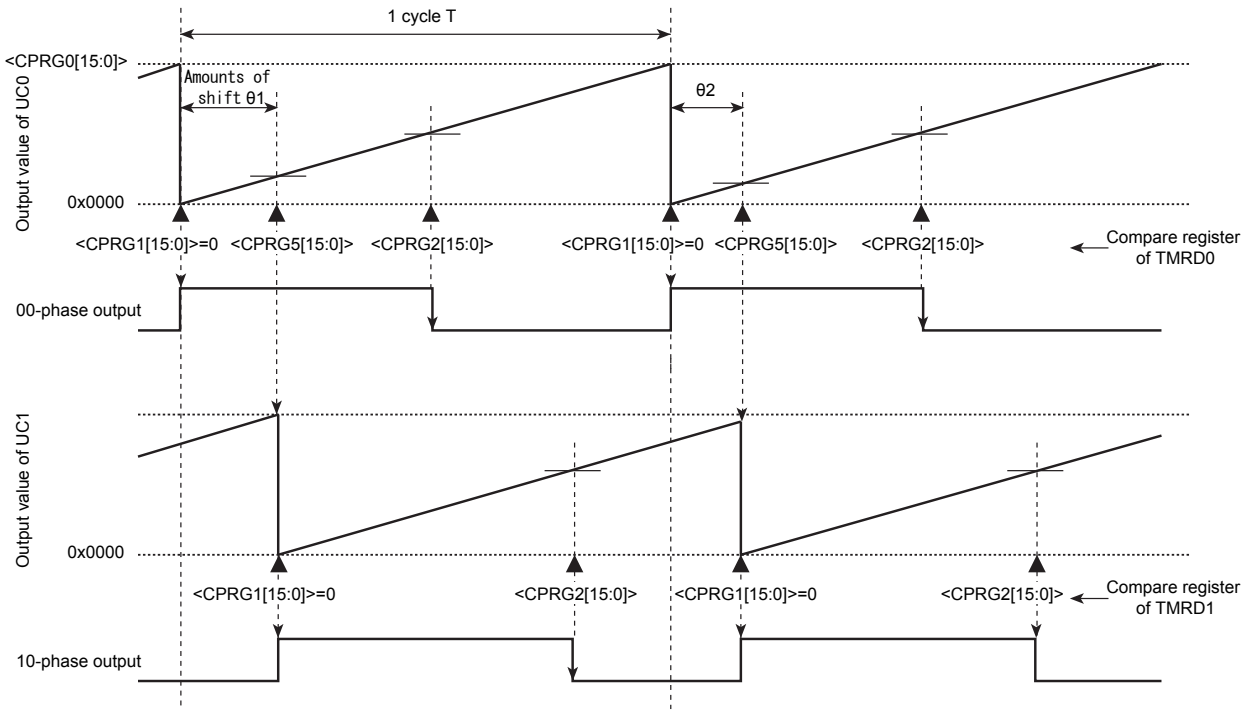


Figure 12-16 00-phase is advanced to 10-phase (00-phase output = Rectangular wave PPG00; 10-phase output = Rectangular wave PPG10; PHSCHGSW="0")

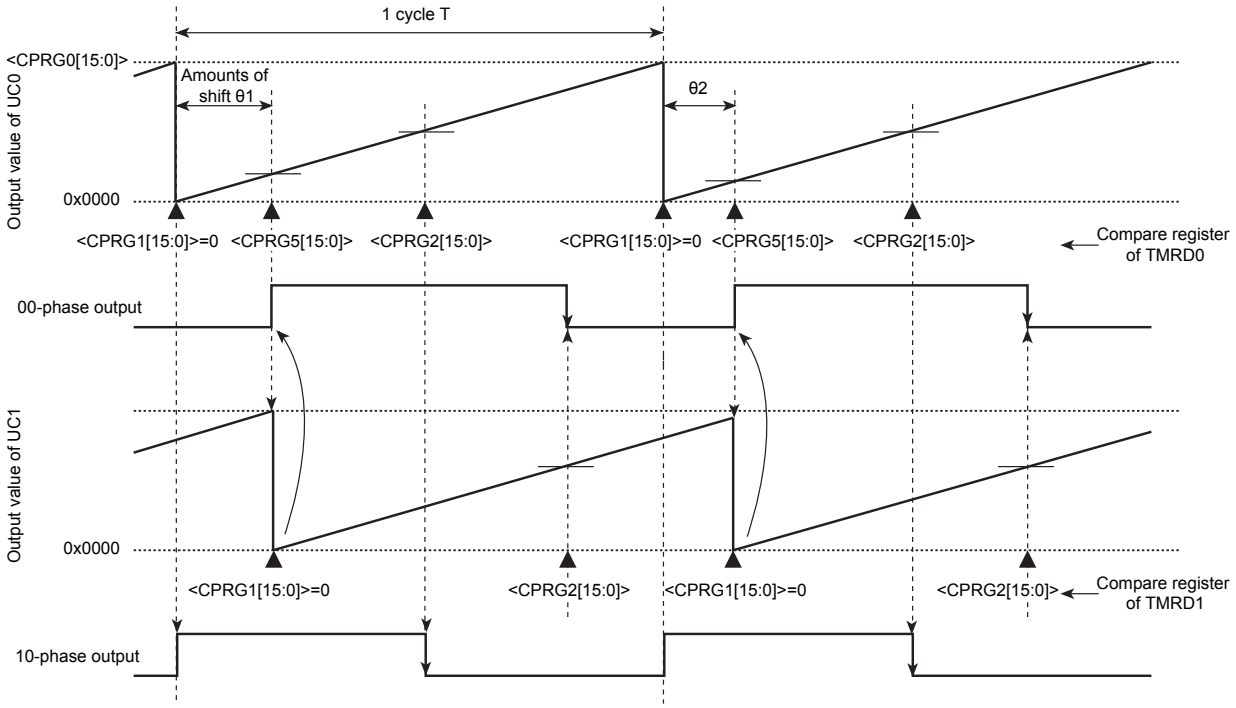


Figure 12-17 00-phase is delayed than 10-phase (00-phase output = Rectangular wave PPG00; 10-phase output = Rectangular wave PPG10; PHSCHGSW="1")

(3) How to set synchronous channel

In the interlock PPG mode, the number of synchronous output channels is determined by TDCONF<TMRDMOD[1:0]>.

When TDCONF<TMRDMOD[1:0]> is set to "110", CH00 and CH10 are synchronized; when TDCONF<TMRDMOD[1:0]> is set to "111", CH00, CH10 and CH11 are synchronized.

Note that, TDBCR<TDSFT**> of an output channel to be synchronized is set to "1" in advance.

Non synchronous channels can be operated independently of synchronous output channels.

TDCONF <TMRDMOD[1:0]>	TDBCR<TDSFT**>	TDnCR <TDMDCY**[2:0]>	Number of synchronous signals	Synchronous output channel
110	TDBCR<TDSFT00>="1" TDBCR<TDSFT10>="1"	TD0CR<TDMDCY00[2:0]>, Set the same value to TD1CR<TDMDCY10[2:0]>.	2	CH00, CH10
110	TDBCR<TDSFT00>="1" TDBCR<TDSFT01>="1" TDBCR<TDSFT10>="1"	TD0CR<TDMDCY00[2:0]>, TD0CR<TDMDCY01[2:0]>, Set the same value to TD1CR<TDMDCY10[2:0]>.	3	CH00, CH01, CH10
111	TDBCR<TDSFT00>="1" TDBCR<TDSFT10>="1" TDBCR<TDSFT11>="1"	TD0CR<TDMDCY00[2:0]>, TD1CR<TDMDCY10[2:0]>, Set the same value to TD1CR<TDMDCY11[2:0]>.	3	CH00, CH10, CH11
111	TDBCR<TDSFT**>="1"	TD0CR<TDMDCY00[2:0]>, TD0CR<TDMDCY01[2:0]>, TD1CR<TDMDCY10[2:0]>, Set the same value to TD1CR<TDMDCY11[2:0]>.	4	CH00, CH01, CH10, CH11

(4) Register update timing

A register update timing varies depending on the relation between a timing of update enable flag TDBCR<TDSFT**> being set to "1" and a timing of a match with CP05 being detected.

The following describes the case when TDCONF<TMRDMOD[1:0]> is set to "110".

When a value of TDCONF<TMRDMOD[1:0]> is set to "111", a value of comparator in output channel (CH11) is updated synchronizing with output channel (CH10).

If a value of compare register in output channel (CH11) is synchronized with CH01, set "1" to <TDSFT01> and <TDSFT00> simultaneously. A value of compare register in CH01 is updated at the same timing of CH00 updated.

a. A timing when update enable flag is set to "1" is delayed than a match with CP05.

If a match with CP00 is detected, TDBCR<TDSFT00> is cleared, at the same time, a values of CP00, CP05 and comparators (CP1 and CP2) of output channel (CH00) are updated.

The compare register of CH10 is updated when a match with CP05 is detected after setting update enable flag to "1". At the same time, <TDSFT10> is cleared and the compare register of CH10 is updated.

b. A timing when update enable flag is set to "1" is ahead of a match with CP05

If a match with CP00 is detected, TDBCR<TDSFT00> is cleared, at the same time, a values of CP00, CP05, comparators (CP1 and CP2) of output channel (CH00) are updated.

However, update of compare register of CH10 and clearing <TDSFT10> are not performed when a match with CP05 is detected after setting update enable flag to "1". These happen on next match with CP05.

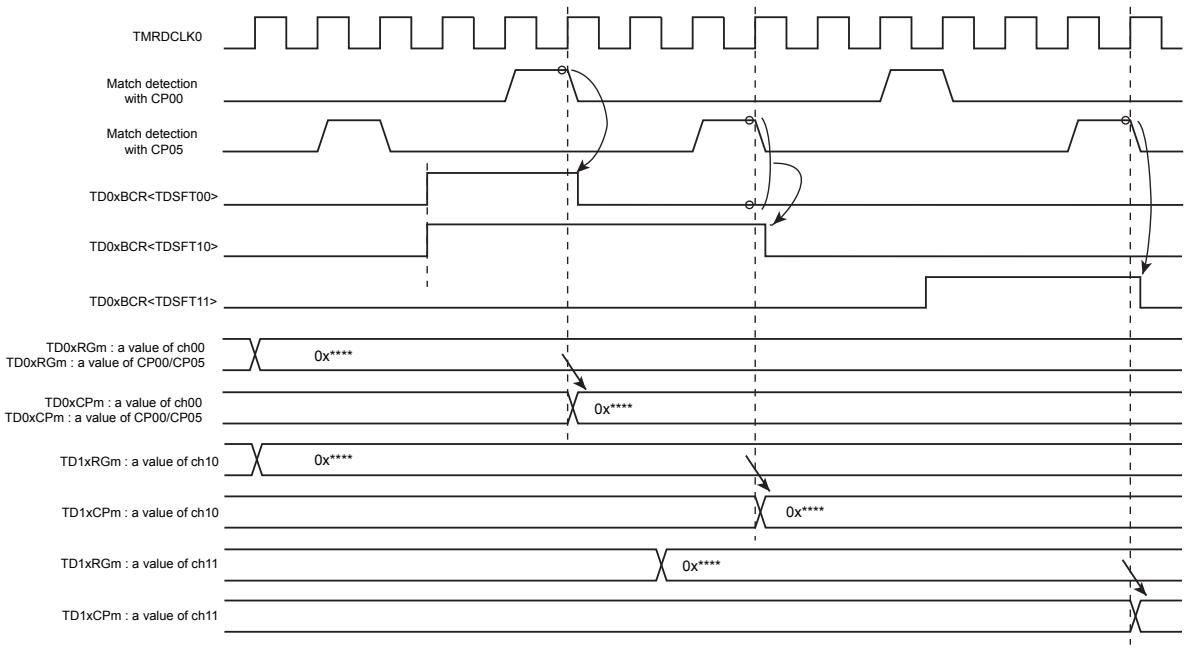


Figure 12-18 A timing to set update enable register is delayed than a match with CP05

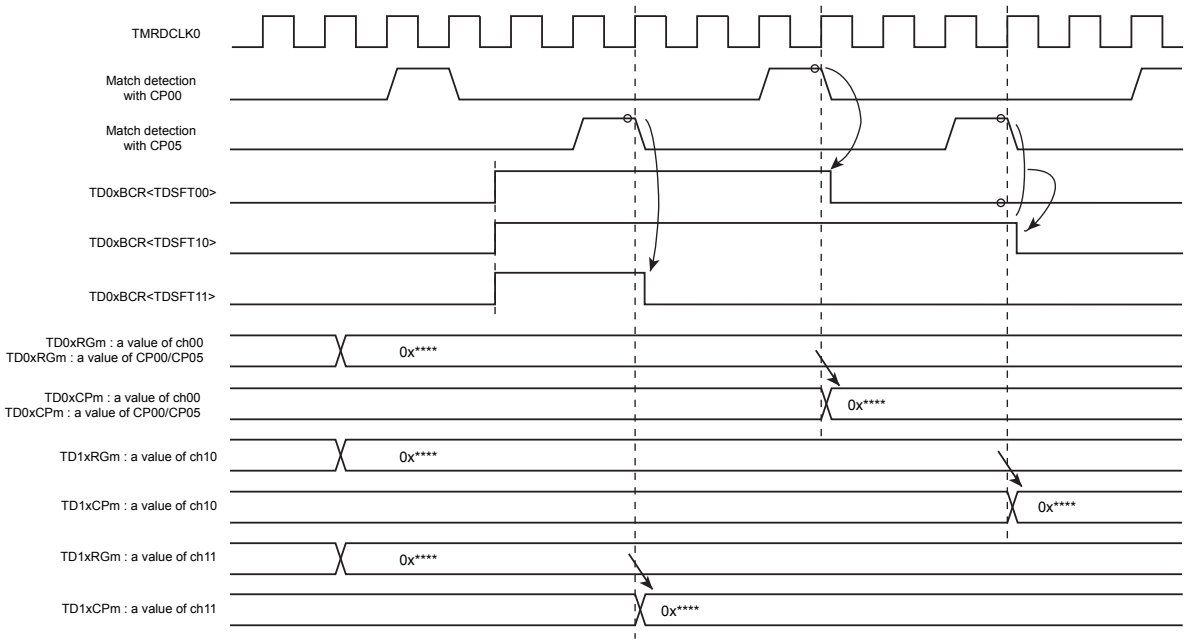


Figure 12-19 A timing to set update enable register is ahead of a match with CP05

(5) Overflow process of the UC1

Since the range of phase shift is $0° \leq \theta < 180°$, a time from a match with CP05 to next match is defined in $0.5T \leq T\theta < 1.5T$ where a cycle of 0-phase/1-phase is T.

In the interlock PPG mode, depending on a value of cycle T, the counter 1 (UC1) may overflow. Thus if a counter value of UC1 is exceeds 0xFFFF, UC1 stops counting-up until next match detection with CP05 is found and maintains the value of 0xFFFF.

(6) Register setting procedure in the interlock PPG mode

(a) Register priority in interlock PPG mode

In this interlock PPG mode, TMR0 serves as a master; TMR1 serves as a slave. Thus, the setting of TMR0 has a higher priority than those of TMR1. The target bit shows in:.

Table 12-3 Higher priority Bits in TMR0

TMR0
TD0MOD<TDCLK[3:0]>
TD0CR<TDRDE>
TD0RUN<TDRUN>

A setting of TMR0 becomes valid and those of TMR1 is ignored in these above registers

Note: Note that a setting value of registers in TMR1 is ignored; however the set value is reflected to the register. Therefore, the set value is immediately valid after the interlock PPG mode ends.

(b) How to startup interlock PPG mode

To startup interlock PPG mode, set registers in the following procedures.

a. Supply clock selection

1. Select a supply clock for TMRD using CGEXTENDO0 in CG register.
2. For more information, refer to the setting of "CG" chapter.

b. Operation mode selection in each timer unit

1. Specify the timer unit in the PPG mode with TDCONF<TMRDMOD[2:0]>.

c. Select a supply clock in each timer unit (TMRn)

1. Set "1" to <TDENn> of TDEN that enables clock supply for each timer unit.ÅiNo need to set unnecessary timer unit to "1".
2. Select a clock for each timer unit using TDnMOD<TDCLK[3:0]>.

   At the same time, select either leading edge or trailing edge using <TDIVn>.

d.Set an initial setting of each timer register and compare register.

1. Set "0" to TDnCR<TDRDE> to change the writing mode to the compare register to direct write mode. At the same time, set an interrupt event of INTTDnCMP0 using TDnCR<TDISO[1:0]>.
2. Set a value to each timer register TDnRGm).

   In the direct write mode, an operation and update timing signal to set TDBCR<TDSFT**> to "1" is not required.

e.Initial setting of phase relation (advance/delay) between 0-phase output and 1-phase output (TDCONF<TMRDMOD[2:0] >="111")

1. Set the relation (advance/delay) between 0-phase and 1-phase output to the TDBCR<PHSCHG> register at starting up. At this time, TDBCR<TDSFT**> is not required to set to "1".

f. Set the 1-bit modulation and update timing. Startup the interlock PPG mode.

1. Set to use or non-use the 1-bit modulation function in CH00 and CH01 with TD0CR<TDMDCY00[2:0]> and <TDMDCY01[2:0]>. Set its cycle with<TDMDPT00> and set the update timing of the compare register with <TDMDPT01>.

   At the same time, set to TD0CR<TDRDE> = "1" to change the writing path to the compare register through a time-rregister. (This is the mode where a value of timer register is written to the corresponding compare register at any timing.)
2. Set to use or non-use the 1-bit modulation function in ch10 and CH11 with the register TD1CR<TDMDCY10[2:0] > and <TDMDCY11[2:0]>. Set its cycle and update timing with <TDMDPT00> and <TDMDPT01>. For details of the setting are referred to the following explanation of registers.

   TD1CR<TDRDE> is not required to set "1".
3. Set the register TD0RUN<TDRUN>="1" to start the interlock PPG mode.

(c) How to update timer registers and compare registers

To update timer registers and compare registers, set the registers in the following preocedures.

1. Set any value to timer registers (TDnRGm) corresponding to compare registers to be updated.
2. After above setting, set "1" to TDBCR<TDSFT**> that corresponds to compare registers to be updated.
3. When TDCONF<TMRDMOD[2:0]>="111" is set, if the phase relation (advance/delay) between A-phase output and B-phase, change the setting value of TDBCR<PHSCHG> and set "!" to TDBCR<TDSFT00>.

   After above setting, if the phase relation (advance/delay) between 0-phase output and 1-phase output, change the setting value of TD0BCR<PHSCHG> and set "1" to TD0BCR<TDSFT>.

By above procedures, a value of timer registers are set to the corresponding compare registers at the specified timing.

4. After interlock PPG mode start-ups, an update timing can be changed using the setting of TDnCR<TDMDPT**> in the corresponding channel.

b. Stop rectangular wave outputs

1. When rectangular wave outputs in each timer unit are stopped, set TDnRUN<TDRUN> register to "0".

Note: During PPG in operation, four registers such as TDnRGm, TDBCR, TDnRUN, TDnCR<TDMDPTn1><TDMDPTn0> can be modified (or written). The other registers must be modified while PPG stops.

(d) How to stop rectangular wave outputs (PPG)

To stop rectangular wave outputs, set the registers in the following procedures.

1. To stop rectangular wave outputs in each timer unit, set TDnRUN<TDRUN>="0" register.

Note: During PPG in operation, four registers such as TDnRGm, TDBCR, TDnRUN, TDnCR<TDMDPTn1><TDMDPTn0> can be modified (or written). The other registers must be modified while PPG stops.

(7) Precaution on switching outputs in the interlock PPG mode

In the interlock mode (TDCONF<TMRDMOD[2:0]>="111"), a trailing edge of rectangular wave PPG10/PPG11 can be set to over the UC0 cycle. Therefore, extra caution should be exercised when a phase relation is changed to "advance to delay" or "delay to advance" (when switching outputs).

The following describes the case that a trailing edge in 1-phase is within the UC0 cycle; a trailing edge in 1-phase is over UC0 cycle; 0- and 1-phase output waves at switching outputs.

In this item, UC0 cycle means a time when a counter value returns to "0" from "0".

(a) In 1-phase, trailing edge is within the UC0 cycle

: shows an example that the phase relation between 00-phase and 10-phase is changed to being delayed from being advanced to 00PPG00-phase.

Output SW is changed at the timing when a match with CP00 is detected. A phase relation that 00-phase output = rectangular wave PPG00, 10-phase output = rectangular wave PPG10 are switched to 00-phase output = rectangular wave PPG10, 10-phase output = rectangular wave PPG00.

At this time, signal levels of rectangular waves (PPG00 and PPG10) at SW switching timing, so that an abnormal wave does not occur.

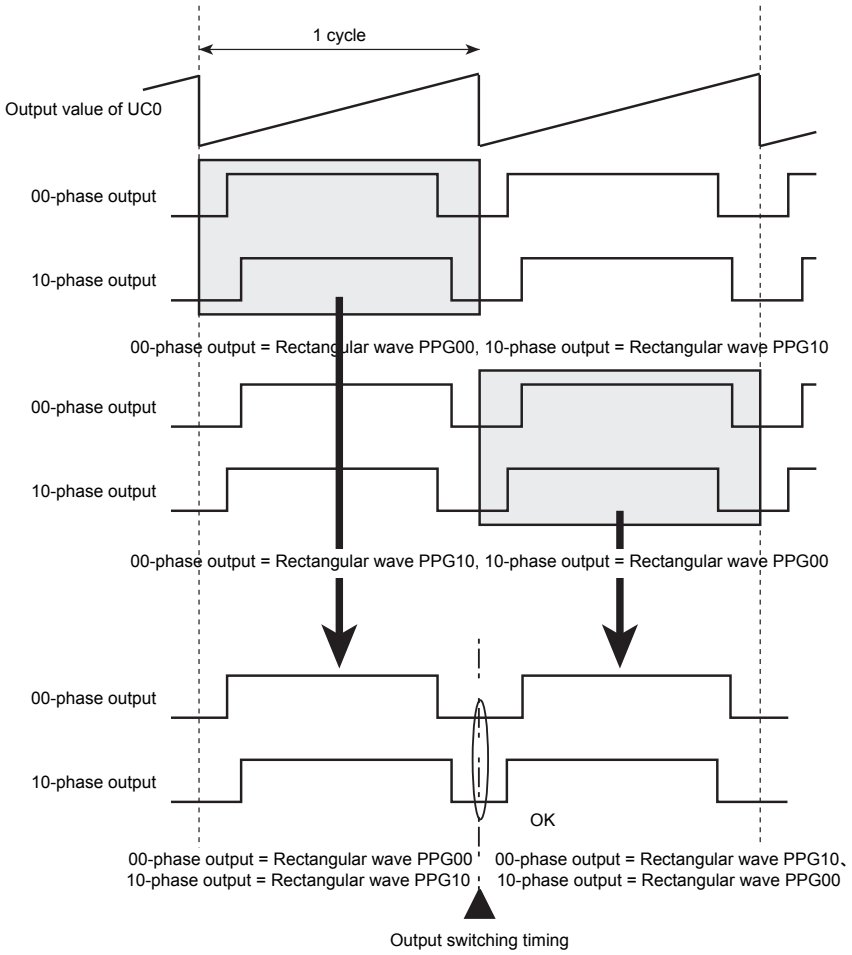


Figure 12-20 Waves before/after phase advance/delay (output SW) (A trailing edge of rectangular wave PPG10 is within UC0 cycle.)

If a phase relation is anti phase such as the case of being delayed to advanced, abnormal waves do not occur.

(b) In 1-phase, a trailing edge is over the UC0 cycle

: shows an example that the phase relation between 00PPG00-phase output (a trailing edge is over UC0 cycle) and 10-phase. In this figure, 00PPG00-phase is changed from being advanced to being delayed.

Output SW is changed at the timing when a match with CP00 is detected. The phase relation is changed as follows: PPG00-phase output = rectangular wave PPG00 or 10-phase output = rectangular wave PPG10 is changed to PPG00-phase output = rectangular wave PPG10 or 10-phase output = rectangular wave PPG00.

At this time, signal levels of rectangular wave PPG00 and PPG10 are different at switching output SWs, so that pulse waves occur.

If the phase relation is anti phase such as being delayed to being advanced, pulse waves occur.

If these pulse waves at switching the phase (advance/delay) can be a problem as a system, it is avoidable by going through the same phase once before changing the phase relations such as being advanced to being delayed and being delayed to being advanced.

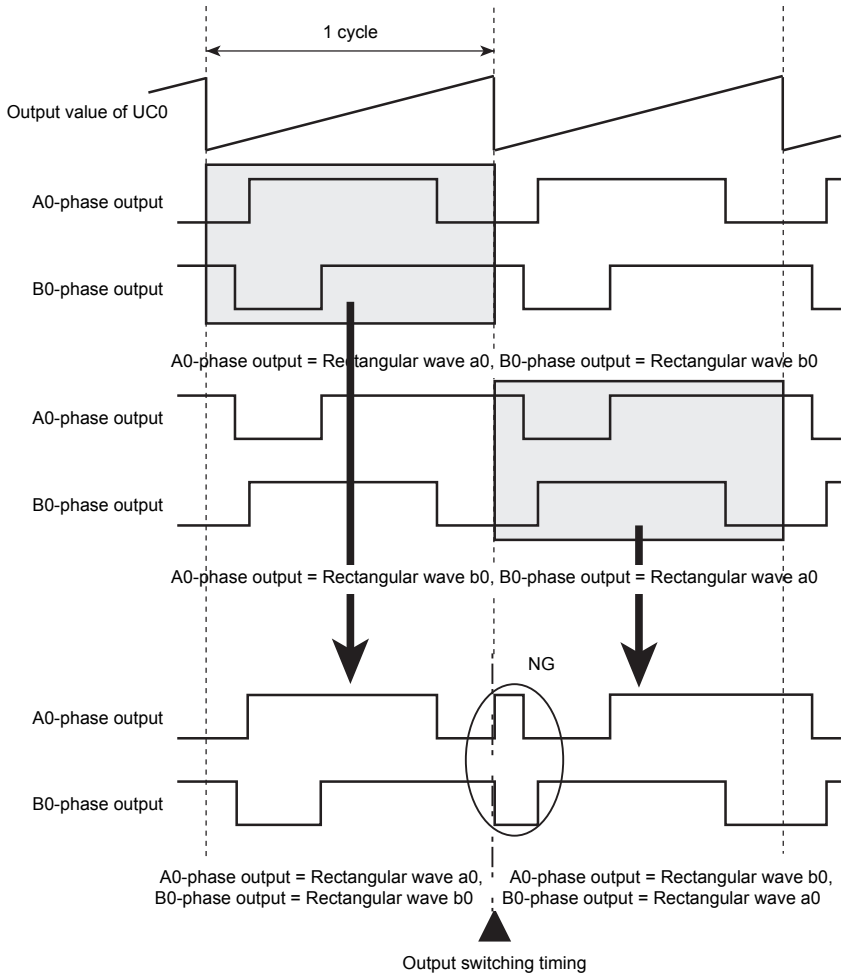


Figure 12-21 Waves before/after phase advance/delay (output SW) (A trailing edge of rectangular wave PPG10 is over the UC0 cycle.)

12.5.1.3 Setting Range of Compare Register

: a setting range of compare register in 16-bit programmable rectangular wave output (PPG).

Table 12-4 Setting range of compare register in 16-bit programmable rectangular wave output (PPG)

Timer unit	Register setting range			PPG	Inter-lock PPG
	Minimum value	Compare register	Maximum value		
TMRD0	(Note) ≤	TD0CP0<CPRG0[15:0]>	≤ 0xFFFF	A	A
	0x0000 ≤	TD0CP1<CPRG1[15:0]>	< TD0CP2<CPRG2[15:0]>	A	A
	TD0CP1<CPRG1[15:0]> <	TD0CP2<CPRG2[15:0]>	≤ TD0CP0<CPRG0[15:0]>	A	A
	0x0000 ≤	TD0CP3<CPRG3[15:0]>	< TD0CP4<CPRG4[15:0]>	A	A
	TD0CP3<CPRG3[15:0]> <	TD0CP4<CPRG4[15:0]>	≤ TD0CP0<CPRG0[15:0]>	A	A
	0x0000 ≤	TD0CP5<CPRG5[16:1]>	< TD0CP0<CPRG0[15:0]> ÷ 2	NA	A
TMRD1	(Note) ≤	TD1CP0<CPRG0[15:0]>	≤ 0xFFFF	A	NA
	0x0000 ≤	TD1CP1<CPRG1[15:0]>	< TD1CP2<CPRG2[15:0]>	A	A
	TD1CP1<CPRG1[15:0]> <	TD1CP2<CPRG2[15:0]>	≤ TD1CP0<CPRG0[15:0]>	A	A
	0x0000 ≤	TD1CP3<CPRG3[15:0]>	< TD1CP4<CPRG4[15:0]>	A	A
	TD1CP3<CPRG3[15:0]> <	TD1CP4<CPRG4[15:0]>	≤ TD1CP0<CPRG0[15:0]>	A	A

A : Available
NA : Not available

Note: The setting range of this product varies on the source clock for TMRD. For the setting range, refer to Chapter "Product Information."

A cycle Tn of the rectangular wave output can be given as the following equation where f_{CLKn} is a frequency of TMRDCLKn.

(1) PPG mode n = 0,1

$$Tn = (1/f_{CLKn}) \times TDnCP0<CPRG0[15:0]>$$

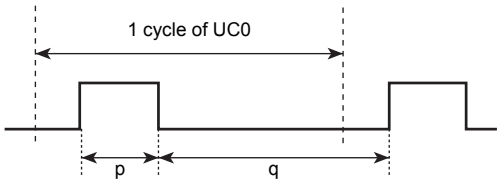
(2) Interlock PPG mode

$$T = (1/f_{CLK0}) \times TD0CP0<CPRG0[15:0]>$$

A duty of rectangular wave output can be given as the following equation where f_{CLK0} is a frequency of TMRDCLK0.

- PPG00

  p : q = {CPRG2[15:0] - CPRG1[15:0]} : {CPRG0[15:0] - CPRG2[15:0] + CPRG1[15:0]}



12.5.2 Each Operation Mode and Interrupts

: shows a relation between each operation mode and interrupt events.

INTTD0CMP0 sets an interrupt event according to a value of TD0CR<TDISO[1:0]>; INTTD1CMP0 sets an interrupt event according to a value of TD1CR<TDISO[1:0]>. Events of other interrupt requests are fixed.

Table 12-5 Each Operation Mode and Interrupt events

Timer unit	Interrupt request	Interrupt event		16-bit programmable rectangular wave output	
		TDnCR <TDISO[1:0]>		PPG mode	Interlock PPG mode
TMR0	INTTD0CMP0	00	No event	×	×
		01	A match with comparator 00(CP00) is detected.	o	o
		10	A match with comparator 05 (CP05) is detected.	o	o
		11	Counter 0 (UC0) overflows	×	×
	INTTD0CMP1	A match with comparator 01 (CP01) is detected.		o	o
	INTTD0CMP2	A match with comparator 02 (CP02) is detected.		o	o
	INTTD0CMP3	A match with comparator 03 (CP03) is detected.		o	o
	INTTD0CMP4	A match with comparator 04 (CP04) is detected.		o	o
TMR1	INTTD1CMP0	00	No event	×	×
		01	A match with comparator 10 (CP10) is detected.	o	×
		10	No event	×	×
		11	Counter 1 (UC1) overflows	×	×
	INTTD1CMP1	A match with comparator 11 (CP11) is detected.		o	o
	INTTD1CMP2	A match with comparator 12 (CP12) is detected.		o	o
	INTTD1CMP3	A match with comparator 13 (CP13) is detected.		o	o
	INTTD1CMP4	A match with comparator 14 (CP14) is detected.		o	o

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

13.1 Overview

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

- Synchronous communication mode (I/O interface mode)
- Asynchronous communication mode (UART mode)

Their features are given in the following.

- Transfer Clock
  - Dividing by the prescaler, from the peripheral clock (φT0) frequency into 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128.
  - Make it possible to divide from the prescaler output clock frequency into 1 to 16.
  - Make it possible to divide from the prescaler output clock frequency into N+m/16 (N=2 to 15, m=1 to 15). (only UART mode)
  - The usable system clock (fsys) (only UART mode).
- Buffer
  - The usable double buffer function.
  - Make it possible to clear the transmit buffer.
- FIFO

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

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

    Keep a "High" level or "Low" level
  - The last bit hold time of SCxTXD pin can be specified in clock input mode.
- UART Mode
  - Data length: 7 bits, 8bits, 9bits
  - Add parity bit (to be against 9bits data length)
  - Serial links to use wake-up function
  - Handshaking function with $\overline{\text{SCxCTS}}$ pin
  - Noise cancel for SCxRXD pin

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

13.2 Configuration

Serial channel block diagram and serial clock generator circuit diagram are shown in bellows.

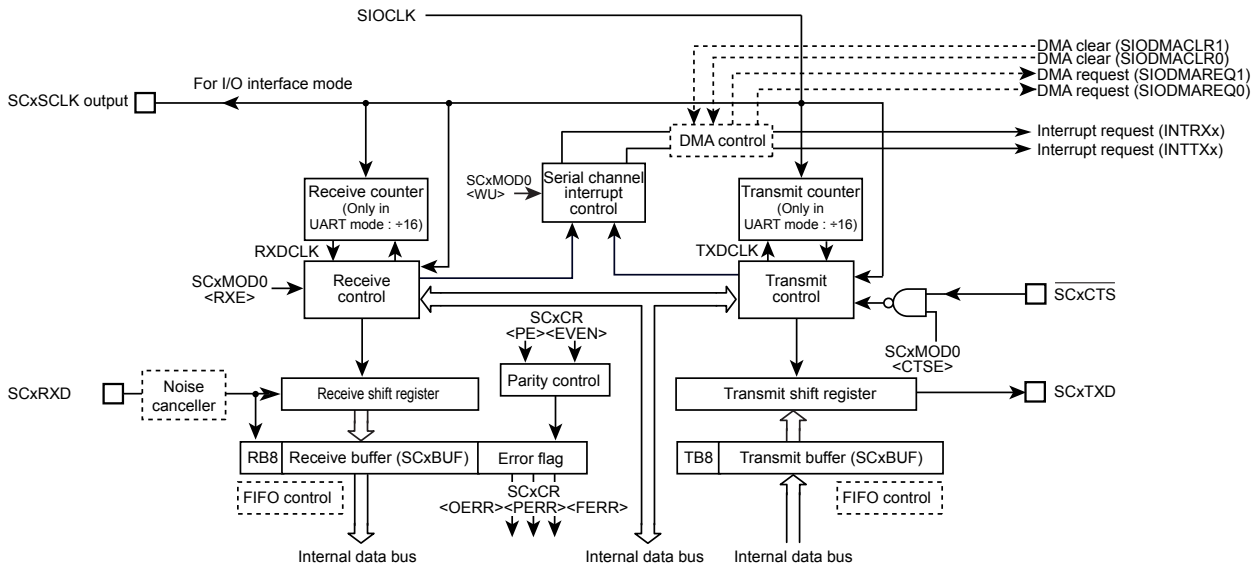


Figure 13-1 Serial Channel Block Diagram

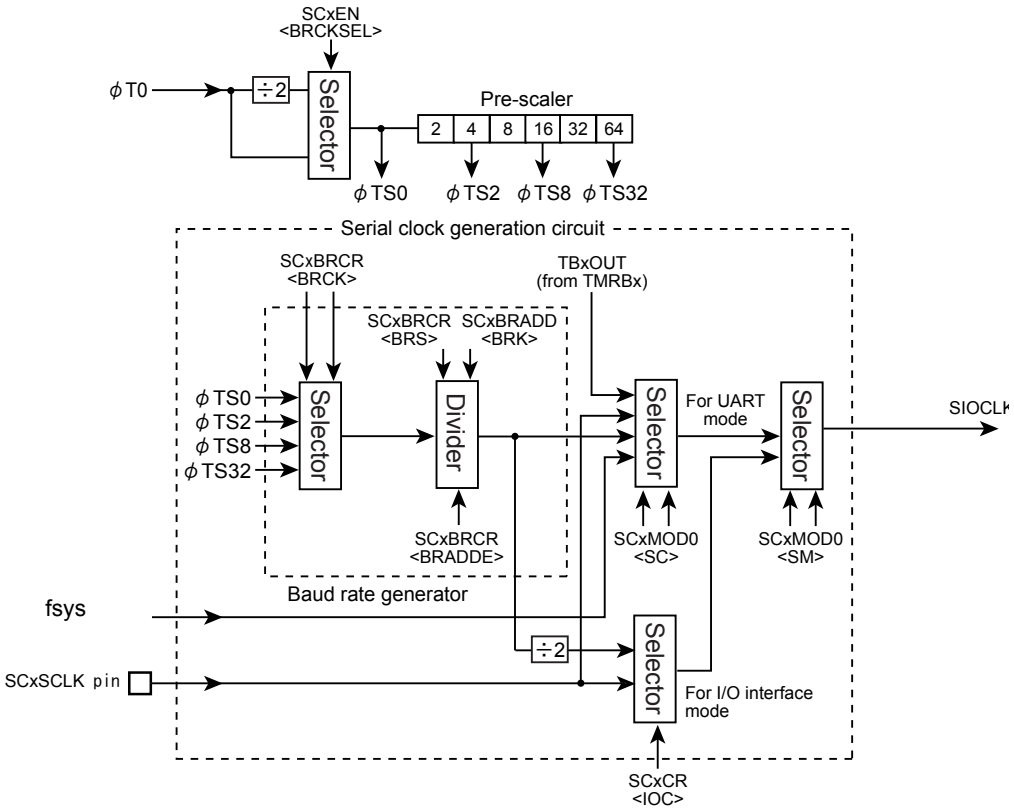


Figure 13-2 Serial clock generation circuit block diagram

13.3 Registers Description

13.3.1 Registers List

The table below shows control registers and their addresses.

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

Register name		Address (Base+)
Enable register	SCxEN	0x0000
Buffer register	SCxBUF	0x0004
Control register	SCxCR	0x0008
Mode control register 0	SCxMOD0	0x000C
Baud rate generator control register	SCxBRCR	0x0010
Baud rate generator control register 2	SCxBRADD	0x0014
Mode control register 1	SCxMOD1	0x0018
Mode control register 2	SCxMOD2	0x001C
Receive FIFO configuration register	SCxRFC	0x0020
Transmit FIFO configuration register	SCxTFC	0x0024
Receive FIFO status register	SCxRST	0x0028
Transmit FIFO status register	SCxTST	0x002C
FIFO configuration register	SCxFCNF	0x0030
DMA request enable register	SCxDMA	0x0034

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

13.3.2 SCxEN (Enable Register)

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

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

13.3.3 SCxBUF (Buffer Register)

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

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

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

13.3.4 SCxCR (Control Register)

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

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

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

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

13.3.5 SCxMOD0 (Mode Control Register 0)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TB8	R/W	Transmit data bit 8 (For only UART mode) Writes the 9th bit of transmit data in the 9-bit UART mode.
6	CTSE	R/W	Handshake function control (For only UART mode) 0: CTS disabled 1: CTS enabled Controls handshake function. Setting "1" enables handshake function using $\overline{\text{SCxCTS}}$ pin.
5	RXE	R/W	Receive control (Note1)(Note2) 0: Disabled 1: Enabled
4	WU	R/W	Wake-up function (For only UART mode) 0: Disabled 1: Enabled This function is available only at 9-bit UART mode. In other mode, this function has no meaning. When it is enabled, interrupt is occurred only when RB9 = "1" in a 9-bit UART mode.
3-2	SM[1:0]	R/W	Specifies transfer mode. 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode
1-0	SC[1:0]	R/W	Serial transfer clock (For only UART mode) 00: TMRB output 01: Baud rate generator 10: System clock (fsys) 11: External clock (SCxSCLK pin input) (For the I/O interface mode, the transfer clock in I/O interface mode is selected by SCxCR<IOC>.)

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

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

13.3.6 SCxMOD1 (Mode Control Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	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	−	R/W	Reserved
6-5	FDPX[1:0]	R/W	Transfer mode setting 00: Transfer prohibited 01: Half duplex (Receive) 10: Half duplex (Transmit) 11: Full duplex Configures the transfer mode in the I/O interface mode. And when FIFO is enabled, specify the configuration of FIFO. In UART mode, specify the only configuration of FIFO.
4	TXE	R/W	Transmit control (Note1)(Note2) 0 :Disabled 1: Enabled This bit enables transmission and is valid for all the transfer modes.
3-1	SINT[2:0]	R/W	Interval time of continuous transmission (For I/O interface mode) 000: None 001: 1 x SCLK cycle 010: 2 x SCLK cycle 011: 4 x SCLK cycle 100: 8 x SCLK cycle 101: 16 x SCLK cycle 110: 32 x SCLK cycle 111: 64 x SCLK cycle This parameter is valid only for the I/O interface mode when SCLK output mode is selected. In other modes, this parameter has no meaning. Specifies the interval time of continuous transmission when double buffering or FIFO is enabled in the I/O interface mode.
0	−	R/W	Write a "0".

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

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

13.3.7 SCxMOD2 (Mode Control Register 2)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TBEMP	R	Transmit buffer empty flag 0: Full 1: Empty If double buffering is disabled, this flag is insignificant. This flag shows that the transmit double buffers are empty. When data in the transmit double buffers is moved to the transmit shift register and the double buffers are empty, this bit is set to "1". Writing data again to the double buffers sets this bit to "0".
6	RBFLL	R	Receive buffer full flag 0: Empty 1: Full If double buffering is disabled, this flag is insignificant. This is a flag to show that the receive double buffers are full. When a receive operation is completed and received data is moved from the receive shift register to the receive double buffers, this bit changes to "1" .When reading the receive buffer ,this bit is cleared to "0".
5	TXRUN	R	In transmission flag 0: Stop 1: Operate This is a status flag to show that data transmission is in progress. <TXRUN> and <TBEMP> bits indicate the following status.
4	SBLEN	R/W	STOP bit length (for UART mode) 0: 1-bit 1: 2-bit This specifies the length of transmission stop bit in the UART mode. On the receive side, the decision is made using only a single bit regardless of the <SBLEN>.
3	DRCHG	R/W	Setting transfer direction 0: LSB first 1: MSB first Specifies the direction of data transfer. In the UART mode, set this bit to LSB first.
2	WBUF	R/W	Enable double-buffer 0: Disabled 1: Enabled This parameter enables or disables the transmit/receive double buffers to transmit (in both SCLK output/input modes) and receive (in SCLK output mode) data i n the I/O interface mode and to transmit in the UART mode. When receiving data in the I/O interface mode (in clock input mode) and UART mode, double buffering is enabled regardless of the <WBUF>.

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>1-0</td><td>SWRST[1:0]</td><td>R/W</td><td>Software reset
Overwriting "01" in place of "10" generates a software reset.
When a software reset is executed, the following bits are initialized and the transmit/receive circuit and FIFO become initial state (Note1)(Note2).
<table>
<tr><th>Register</th><th>Bit</th></tr>
<tr><td>SCxMOD0</td><td><RXE></td></tr>
<tr><td>SCxMOD1</td><td><TXE></td></tr>
<tr><td>SCxMOD2</td><td><TBEMP>, <RBFLL>, <TXRUN></td></tr>
<tr><td>SCxCR</td><td><OERR>, <PERR>, <FERR></td></tr>
</table>
</td></tr>
</table>

Note 1: While data transmission is in progress, any software reset operation must be executed twice in succession.

Note 2: A software reset requires 2 clocks-duration at the time between the end of recognition and the start of execution of software reset instruction.

13.3.8 SCxBRCR (Baud Rate Generator Control Register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	−	R/W	Write "0".
6	BRADDE	R/W	N + (16 − K)/16 divider function (Only for UART mode) 0: disabled 1: enabled
5-4	BRCK[1:0]	R/W	Select input clock to the baud rate generator. 00:φTS0 01:φTS2 10:φTS8 11:φTS32
3-0	BRS[3:0]	R/W	Division ratio "N" 0000: N = 16 0001: N = 1 0010: N = 2 ... 1111: N = 15

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

Note 2: The division ratio "1" of the baud rate generator can be specified only when the double buffering is used in the I/O interface mode.

13.3.9 SCxBRADD (Baud Rate Generator Control Register 2)

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

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

Table 13-1 lists the settings of baud rate generator division ratio.

Table 13-1 Setting division ratio

	<BRADDE> = "0"	<BRADDE> = "1" (Note1) (Only in the UART mode)
<BRS>	Specify "N"	
<BRK>	No setting required	Specify "K" (Note2)
Division ratio	Divide by N	$N + \frac{(16 - K)}{16}$ division.

Note 1: To use the "N + (16 - K)/16" division function, be sure to set <BRADDE> to "1" after setting the K value to <BRK>. The "N + (16 - K)/16" division function can only be used in the UART mode.

Note 2: Specifying "K = 0" is prohibited.

13.3.10 SCxFCNF (FIFO Configuration Register)

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

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

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

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

13.3.11 SCxRFC (Receive FIFO Configuration Register)

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

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[2:0]> is "000". And also the read pointer is initialized. Read as "0".
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]>) For the detail of interrupt condition, refer to "13.12.1.2 FIFO"
5-2	-	R	Read as "0".
1-0	RIL[1:0]	R/W	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 transfer mode (half duplex/full duplex) and enabling FIFO (SCxFCNF<CNFG> = "1")

13.3.12 SCxTFC (Transmit FIFO Configuration Register)

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

Bit	Bit Symbol	Type	Function
31-9	−	R	Read as "0".
8	TBCLR	W	Transmit buffer clear 0: Don't care 1: Clear When SCxTFC<TBCLR> is set to "1", the transmit buffer is cleared. Read as "0".
7	TFCS	W	Transmit FIFO clear (Note1) 0: Don't care 1: Clear When SCxTFC<TFCS> is set to "1", the transmit FIFO is cleared and SCxTST<TLVL[2:0]> is "000". And also the write pointer is initialized. Read as "0".
6	TFIS	R/W	Selects interrupt generation condition. 0: When FIFO fill level (SCxTST<TLVL[2:0]>) = Transmit FIFO fill level to generate transmit interrupt (<TIL[1:0]>) 1: When FIFO fill level (SCxTST<TLVL[2:0]>) ≤ Transmit FIFO fill level to generate transmit interrupt (<TIL[1:0]>) For the detail of interrupt condition, refer to "13.12.2.2 FIFO"
5-2	−	R	Read as "0".
1-0	TIL[1:0]	R/W	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 transfer mode (half duplex/full duplex) and enabling FIFO (SCxFCNF<CNFG> = "1").

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

13.3.13 SCxRST (Receive FIFO Status Register)

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

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

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

13.3.14 SCxTST (Transmit FIFO Status Register)

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

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

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

13.3.15 SCxDMA (DMA request 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	-	-	-	-	-	-	DMAEN1	DMAEN0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	DMAEN1	R/W	Enable DMA request. DMA request is generated by receive interrupt INTRX. 0: disable 1: enable
0	DMAEN0	R/W	Enable DMA request. DMA request is generated by transmit interrupt INTTX. 0: disable 1: enable

Note: When DMA request is generated during DMA transfer is being, it is not kept and nesting.

13.4 Operation in Each Mode

Table 13-2 shows the modes.

Table 13-2 Modes

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

The Mode 0 is a synchronous communication and can be used to extend I/O. This mode transmits and receives data in synchronization with SCLK clock. SCLK clock can be used for both input and output modes. The direction of data transfer can be selected from LSB first or MSB first. This mode is not allowed either to use parity bits or STOP bits.

The mode 1, mode 2 and mode 3 are asynchronous modes and the transfer directions can be selected as only the LSB first.

Parity bits can be added in the mode 1 and mode 2. The mode 3 has a wakeup function in which the master controller can start up slave controllers via the serial link (multi-controller system). STOP bit in transmission can be selected from 1 bit and 2 bits. The STOP bit length in reception is fixed to a one bit.

13.5 Data Format

13.5.1 Data Format List

Figure 13-3 shows data format.

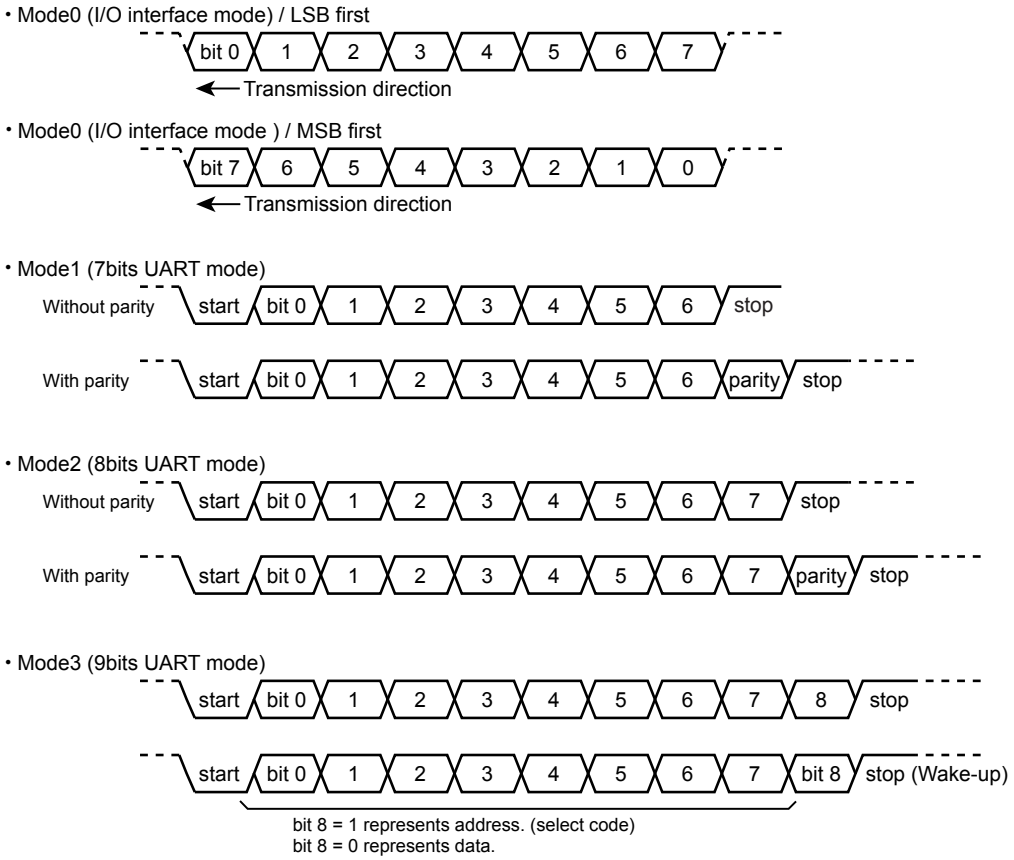


Figure 13-3 Data Format

13.5.2 Parity Control

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

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

13.5.2.1 Transmission

Upon data transmission, the parity control circuit automatically generates the parity with the data in the transmit buffer. The parity bit will be stored in SCxBUF<TB7> in the 7-bit UART mode and SCxMOD<TB8> in the 8-bit UART mode.

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

13.5.2.2 Receiption

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

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

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

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

13.5.3 STOP Bit Length

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

13.6 Clock Control

13.6.1 Prescaler

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

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

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

13.6.2 Serial Clock Generation Circuit

The serial clock generation circuit is a block to generate transmit and receive clocks (SIOCLK) and consists of the circuits in which clocks can be selected by the settings of the baud rates generator and modes.

13.6.2.1 Baud Rate Generator

The baud rate generator generates transmit and receive clocks to determine the serial channel transfer rate.

(1) Baud Rate Generator input clock

The input clock of the baud rate generator is selected from the prescaler outputs divided by 1, 4, 16 and 64.

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

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

(2) Baud Rate Generator output clock

The frequency division ratio of the output clock in the baud rate generator is set by SCxBRCR and SCxBRADD.

The following frequency divide ratios can be used; 1/N frequency division in the I/O interface mode, either 1/N or 1/(N + (16-K)/16) in the UART mode.

The table below shows the frequency division ratio which can be selected.

Mode	Divide Function Setting SCxBRCR<BRADDE>	Divide by N SCxBRCR<BRS[3:0]>	Divide by K SCxBRADD<BRK[3:0]>
I/O interface	Divide by N	1 to 16 (Note)	-
UART	Divide by N	1 to 16	-
	N + (16-K)/16 division	2 to 15	1 to 15

Note: 1/N (N=1) frequency division ratio can be used only when a double buffer is enabled.

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

- Divide by N

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

- N + (16-K)/16 division

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

13.6.2.2 Clock Selection Circuit

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

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

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

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

(1) Transfer Clock in I/O interface mode

Table 13-3 shows clock selection in I/O interface mode.

Table 13-3 Clock Selection in I/O Interface Mode

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

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

- If double buffer is used
  - SCLK cycle > 6/fsys
- If double buffer is not used
  - SCLK cycle > 8/fsys

(2) Transfer clock in the UART mode

Table 13-4 shows the clock selection in the UART mode. In the UART mode, selected clock is divided by 16 in the receive counter or the transmit counter before use.

Table 13-4 Clock Selection in UART Mode

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

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

- SCLK cycle > 2/fsys

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

Baud rates can be obtained by using the following formula.

Baud rate calculation

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by CGSYSCR<PRCK>}}{(\text{TBxRG1} \times 2) \times 2 \times 16}$$

In the case the timer prescaler clock ΦT1 (2divition ratio) is selected.

One clock cycle is a period that the timer flip-flop is inverted twice.

13.6.3 Transmit/Receive Buffer and FIFO

13.6.3.1 Configuration

Figure 13-4 shows the configuration of transmit buffer, receive buffer and FIFO.

Appropriate settings are required for using buffer and FIFO. The configuration may be predefined depending on the mode.

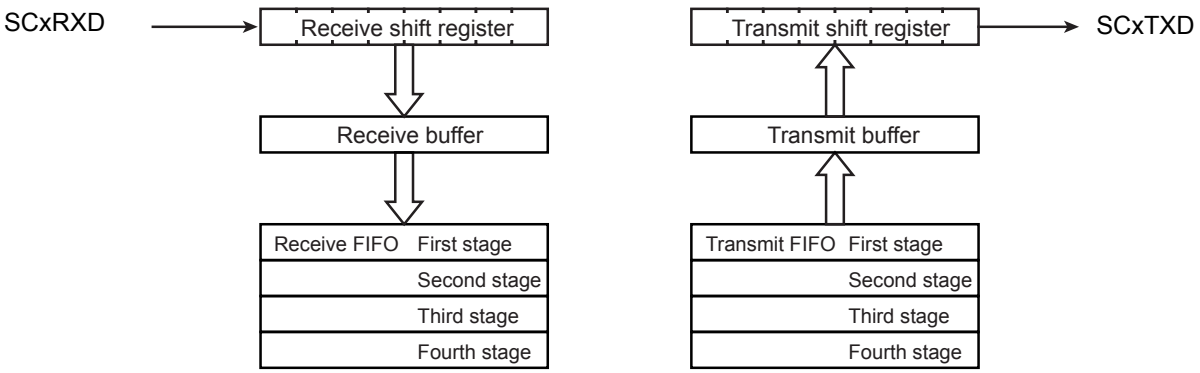


Figure 13-4 The Configuration of Buffer and FIFO

13.6.3.2 Transmit/Receive Buffer

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

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

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

Table 13-5 shows correlation between modes and buffers.

Table 13-5 Mode and buffer Composition

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

13.6.3.3 Initialize Transmit Buffer

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

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

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

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

13.6.3.4 FIFO

In addition to the double buffer function above described, 4-byte FIFO can be used.

To enable FIFO, enable the double buffer by setting SCxMOD2<WBUF> to "1" and SCxFCNF<CNFG> to "1". The FIFO buffer configuration is specified by SCxMOD1<FDPX[1:0]>.

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

Table 13-6 shows correction between modes and FIFO.

Table 13-6 Mode and FIFO Composition

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

13.7 Status Flag

The SCxMOD2 has two types of flag. This bit is significant only when the double buffer is enabled.

<RBFLL> is a flag to show that the receive buffer is full. When one frame of data is received and the data is moved from the receive shift register to the receive buffers, this bit changes to "1". When reading the receive buffer is read, this bit is cleared to "0".

<TBEMP> shows that the transmit buffer is empty. When data in the transmit buffers is moved to the transmit shift register, this bit is set to "1". When data is set to the transmit buffers, the bit is cleared to "0".

13.8 Error Flag

Three error flags are provided in the SCxCR. The meaning of the flags is changed depending on the modes. The table below shows the meanings in each mode.

These flags are cleared to "0" after reading the SCxCR.

<table>
<tr><th rowspan="2">Mode</th><th colspan="3">Flag</th></tr>
<tr><th><OERR></th><th><PERR></th><th><FERR></th></tr>
<tr><td>UART mode</td><td>Over-run error</td><td>Parity error</td><td>Framing error</td></tr>
<tr><td rowspan="2">I/O Interface mode
(Clock input mode)</td><td rowspan="2">Over-run error</td><td>Under-run error
(When a double buffer and FIFO are used)</td><td rowspan="2">Fixed to 0</td></tr>
<tr><td>Fixed to 0
(When a double buffer and FIFO are not used)</td></tr>
<tr><td>I/O Interface mode
(Clock output mode)</td><td>Undefined</td><td>Undefined</td><td>Fixed to 0</td></tr>
</table>

13.8.1 OERR Flag

In both UART and I/O interface modes, this bit is set to "1" when an error is generated by completing the reception of the next frame before the receive buffer has been read.

If the receive FIFO is enabled, the received data is automatically moved to the receive FIFO and no overrun error will be generated until the receive FIFO is full (or until the usable bytes are fully occupied).

In the I/O interface mode with clock output mode, the SCxSCLK pin output stops upon setting the flag.

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

13.8.2 PERR Flag

This flag indicates a parity error in the UART mode and an under-run error or completion of transmit in the I/O interface mode.

In the UART mode, <PERR> is set to "1" when the parity generated from the received data is different from the received parity bit.

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

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

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

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

13.8.3 FERR Flag

A framing error is generated if the corresponding stop bit is determined to be "0" by sampling the bit at around the center. Regardless of the stop bit length settings in the SCxMOD2<SBLEN>, the stop bit status is determined by only 1’st STOP bit.

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

13.9 Receive

13.9.1 Receive Counter

The receive counter is a 4-bit binary counter and is up-counted by SIOCLK.

In the UART mode, sixteen SIOCLK clock pulses are used in receiving a single data bit and the data symbol is sampled at the eighth pulse.

13.9.2 Receive Control Unit

13.9.2.1 I/O interface mode

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

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

13.9.2.2 UART Mode

The receive control unit has a start bit detection circuit, which is used to initiate receive operation when a normal start bit is detected.

13.9.3 Receive Operation

13.9.3.1 Receive Buffer

The received data is stored by 1 bit in the receive shift register. When a complete set of bits has been stored, the interrupt INTRXx is generated.

When the double buffer is enabled, the data is moved to the receive buffer (SCxBUF) and the receive buffer full flag (SCxMOD2<RBFLL>) is set to "1". The receive buffer full flag is cleared to "0" by reading the receive buffer. When the double buffer is disabled, the receive buffer full flag has no meaning.

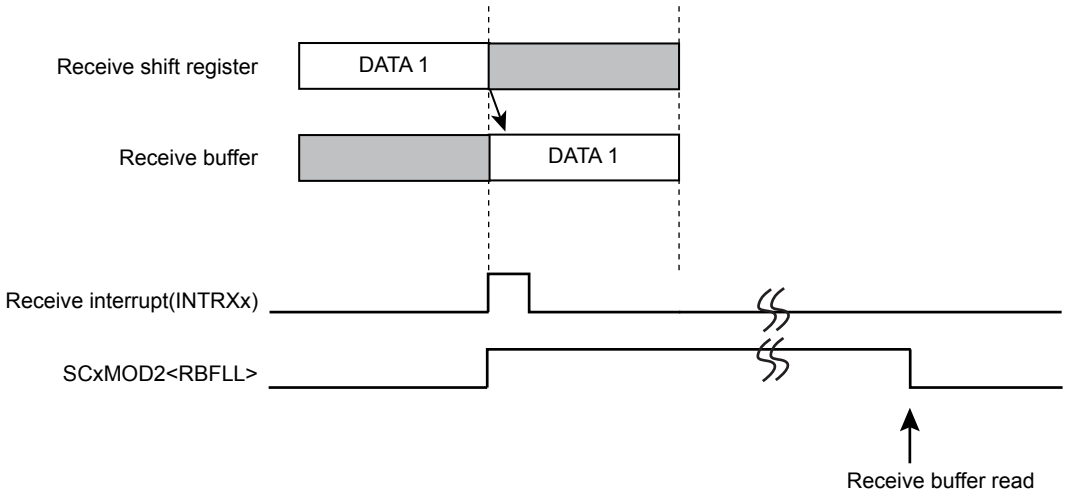


Figure 13-5 Receive Buffer Operation

13.9.3.2 Receive FIFO Operation

When FIFO is enabled, the received data is moved from receive buffer to receive FIFO and the receive buffer full flag is cleared immediately. An interrupt will be generated according to the SCxRFC<RIL[1:0]>.

Note: **When the data with parity bit are received in UART mode by using the FIFO, the parity error flag is shown the occurring the parity error in the received data.**

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

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

After setting of the above FIFO configuration, the data reception is started by writing "1" to the SCxMOD0<RXE>. When the data is stored all in the receive shift register, receive buffer and receive FIFO, SCxMOD0<RXE> is automatically cleared and the receive operations completed.

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

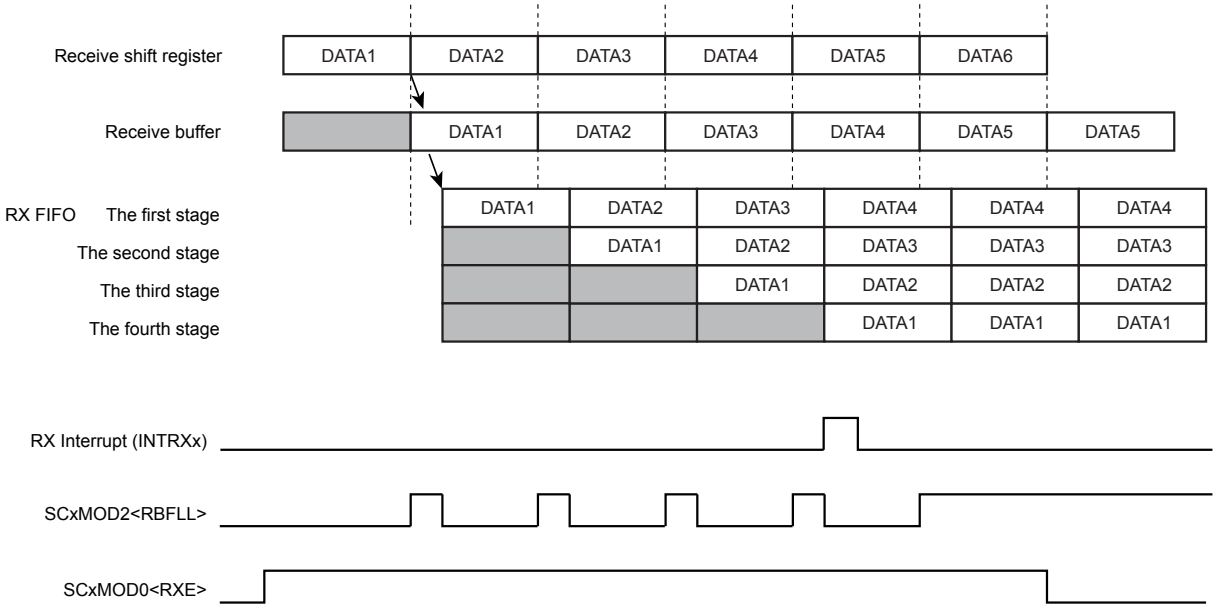


Figure 13-6 Receive FIFO Operation

13.9.3.3 I/O interface mode with clock output mode

In the I/O interface mode with clock output mode setting, clock stops when all received data is stored in the receive buffer and FIFO. So, in this mode, the over-run error flag has no meaning.

The timing of SCLK output stop and re-output depends on receive buffer and FIFO.

(1) Case of single buffer

Stop clock output after receiving a data. In this mode, I/O interface can transfer each data with the transfer device by hand-shake.

When the data in a buffer is read, clock output is restarted.

(2) Case of double buffer

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

When a data is read, clock output is restarted.

(3) Case of FIFO

Stop clock output after receiving the data into a shift register, received buffer and FIFO.

When one byte data is read, the data in the received buffer is transferred into FIFO and the data in the receive shift register is transferred into the received buffer and clock output restarts.

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

13.9.3.4 Read Received Data

In spite of enabling or disabling FIFO, read the received data from the receive buffer (SCxBUF).

When receive FIFO is disabled, the buffer full flag SCxMOD2<RBFLL> is cleared to "0" by this reading. The next data can be received in the receive shift register before reading a data from the receive buffer. The parity bit to be added in the 8-bit UART mode as well as the most significant bit in the 9-bit UART mode will be stored in SCxCR<RB8>.

When the receive FIFO is enabled, the 9-bit UART mode is prohibited because up to 8-bit data can be stored in receive FIFO. In the 8-bit UART mode, the parity bit is lost but parity error is determined and the result is stored in SCxCR<PERR>.

13.9.3.5 Wake-up Function

In the 9-bit UART mode, the slave controller can be operated in the wake-up mode by setting the wake-up function SCxMOD0 <WU> to "1". In this case, the interrupt INTRXx will be generated only when SCxCR <RB8> is set to "1".

13.9.3.6 Overrun Error

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

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

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

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

13.10 Transmit

13.10.1 Transmit Counter

The transmit counter is a 4-bit binary counter and is counted by SIOCLK as in the case of the receive counter. In UART mode, it generates a transmit clock (TXDCLK) on every 16th clock pulse.

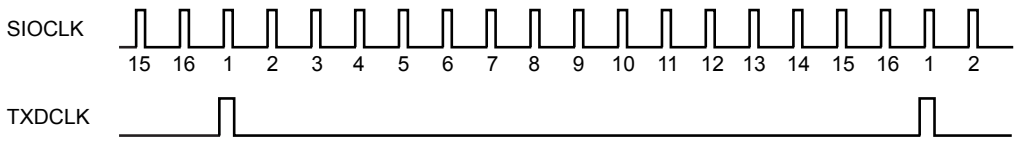


Figure 13-7 Generation of Transmission Clock in UART mode

13.10.2 Transmit Control

13.10.2.1 In I/O Interface Mode

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

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

13.10.2.2 In UART Mode

When the transmit data is written in the transmit buffer, data transmission is initiated on the rising edge of the next TXDCLK and the transmit shift clock signal is also generated.

13.10.3 Transmit Operation

13.10.3.1 Operation of Transmit Buffer

If double buffering is disabled, the CPU writes data only to transmit shift register and the transmit interrupt INTTXx is generated upon completion of data transmission.

If double buffering is enabled (including the case the transmit FIFO is enabled), data written to the transmit buffer is moved to the transmit shift register. The INTTXx interrupt is generated at the same time and the transmit buffer empty flag (SCxMOD2<TBEMP>) is set to "1". This flag indicates that the next transmit data can be written. When the next data is written to the transmit buffer, the <TBEMP> flag is cleared to "0".

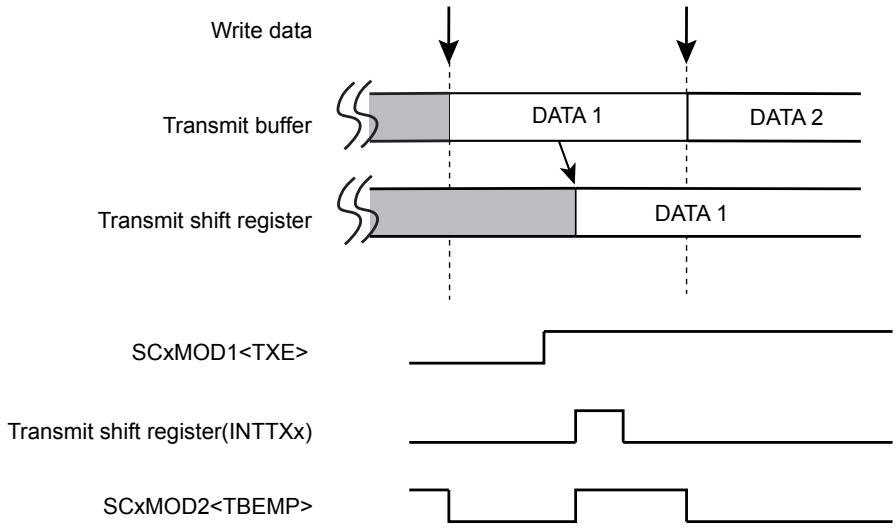


Figure 13-8 Operation of Transmit Buffer (Double-buffer is enabled)

13.10.3.2 Transmit FIFO Operation

When FIFO is enabled, the maximum 5-byte data can be stored using the transmit buffer and FIFO. Once transmission is enabled, data is transferred to the transmit shift register from the transmit buffer and start transmission. If data exists in the FIFO, the data is moved to the transmit buffer immediately, and the <TBEMP> flag is cleared to "0".

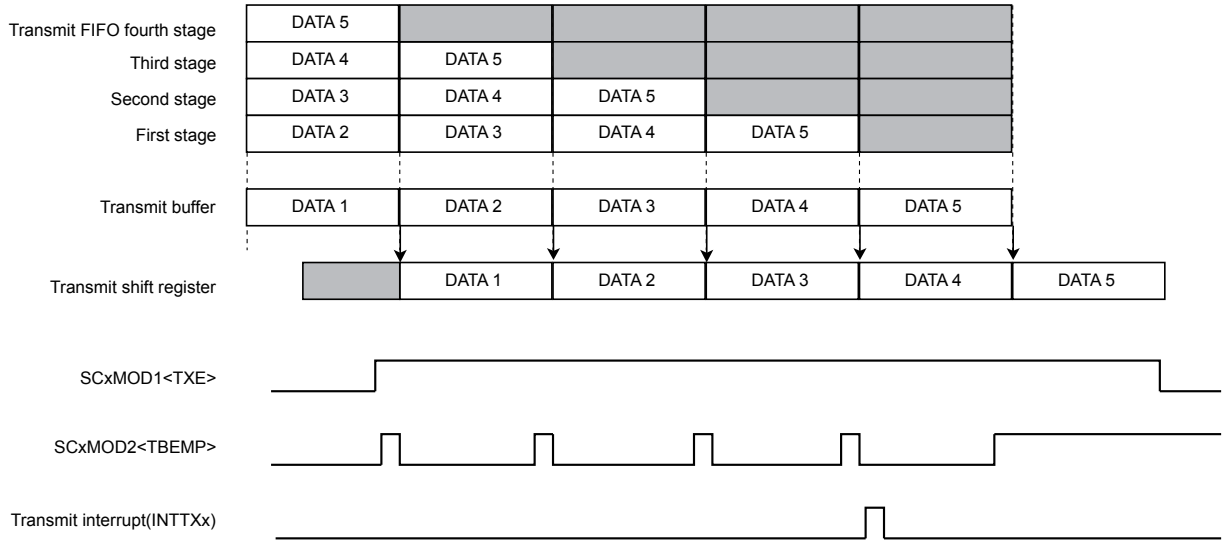
Note: To use Transmit FIFO buffer, Transmit FIFO must be cleared after setting the SIO transfer mode (half duplex/ full duplex) and enabling FIFO (SCxFCNF<CNFG>="1").

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

SCxMOD1<FDPX[1:0]> = "10"	:Transfer mode is set to half duplex.
SCxFCNF<RFST><TFIE><RFIE>	:Transmission is automatically disabled if FIFO becomes empty.
<RXTXCNT><CNFG> = "11011"	:The number of bytes to be used in the receive FIFO is the same as the interrupt :generation fill level.
SCxTFC<TIL[1:0]> = "00"	:Sets the interrupt generation fill level to "0".
SCxTFC<TFCS><TFIS> = "11"	:Clears receive FIFO and sets the condition of interrupt generation.
SCxFCNF<CNFG> = "1"	:Enable FIFO

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

Once above settings are configured, if the transmission is not set as auto disabled, the transmission should lasts writing transmit data.



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

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

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

(1) Single Buffer

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

(2) Double Buffer

The clock output stops upon completion of data transmission in the transmit shift register and the transmit buffer. The clock output resumes when the next data is written in the buffer.

(3) FIFO

The transmission of all data stored in the transmit shift register, transmit buffer and FIFO is completed, the SCLK output stops. The next data is written, clock output resumes.

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

13.10.3.4 Level of SCxTXD pin after the last bit is output in I/O interface mode

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

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

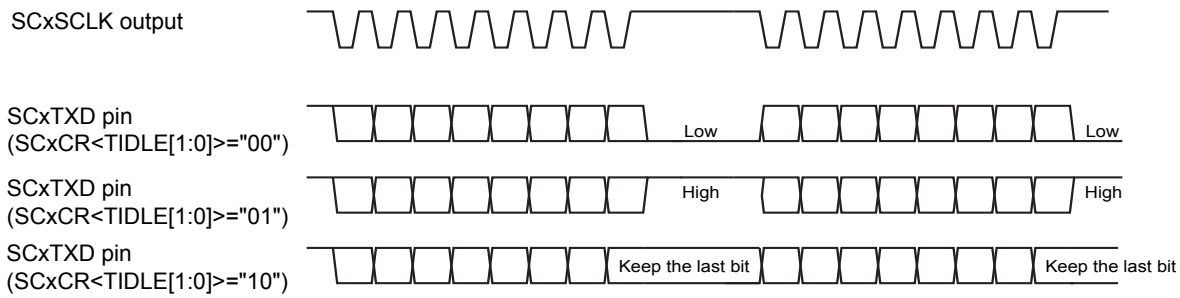


Figure 13-9 Level of SCxTXD pin After the last bit is output

13.10.3.5 Under-run error

In the I/O interface mode with clock input mode and if FIFO is empty and if no data is set in transmit buffer before the next frame clock input, which occurs upon completion of data transmission from transmit shift register, an under-run error occurs and SCxCR<PERR> is set to "1".

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

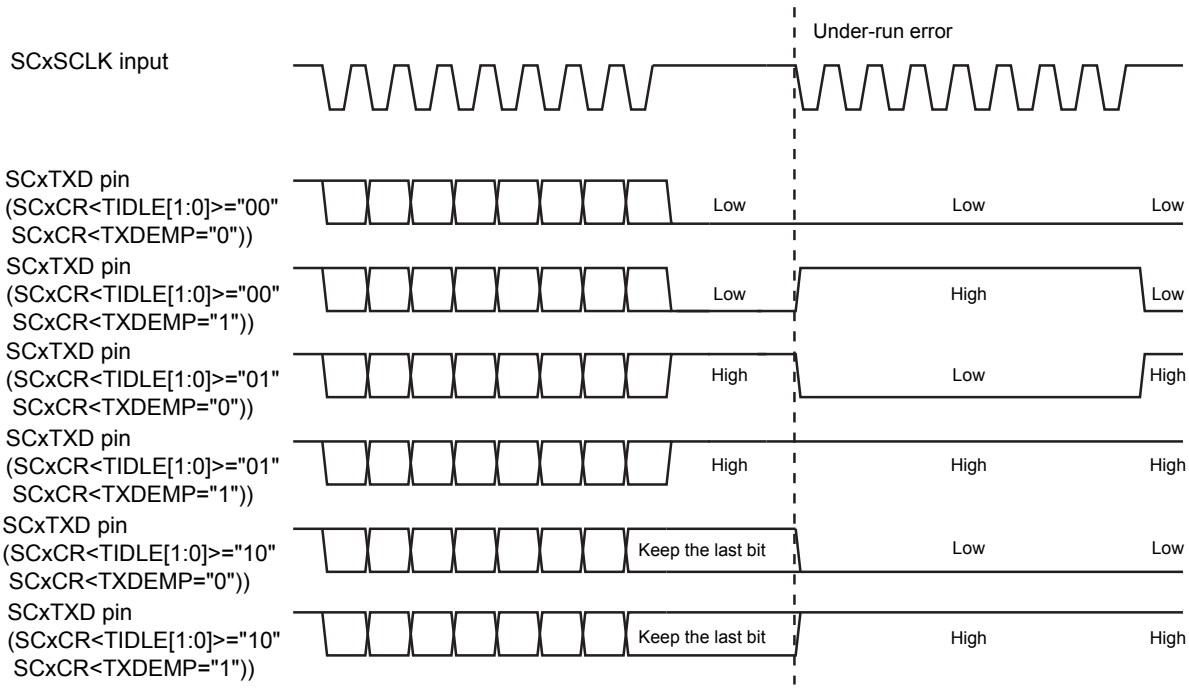


Figure 13-10 Level of SCxTXD pin when Under-run Error is Occurred

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

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

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

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

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

13.11 Handshake function

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

When the $\overline{\text{SCxCTS}}$ pin is set to "High" level, the current data transmission can be completed but the next data transmission is suspended until the $\overline{\text{SCxCTS}}$ pin returns to the "Low" level. The INTTXx interrupt is generated in the normal timing, the next transmit data is written in the transmit buffer, and it waits until it is ready to transmit data.

Note 1: If the $\overline{\text{CTS}}$ signal is set to "High" level during transmission, the next data transmission is suspended after the current transmission is completed.

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

Although no $\overline{\text{RTS}}$ pin is provided, a handshake control function can easily implemented by assigning one bit of the port for the $\overline{\text{RTS}}$ function. By setting the port to "High" level upon completion of data reception (in the receive interrupt routine), the transmit side can be requested to suspend data transmission.

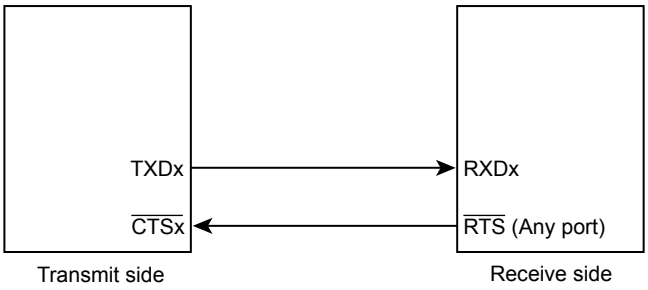


Figure 13-11 Handshake Function

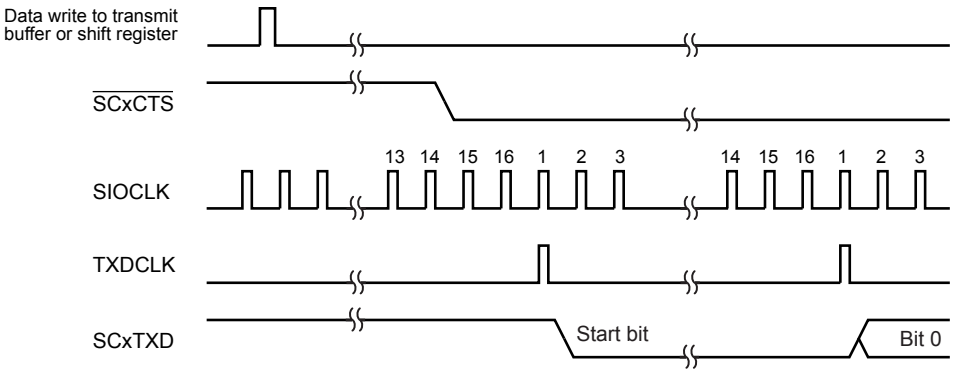


Figure 13-12 $\overline{\text{SCxCTS}}$ Signal timing

13.12 Interrupt/Error Generation Timing

13.12.1 Receive Interrupts

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

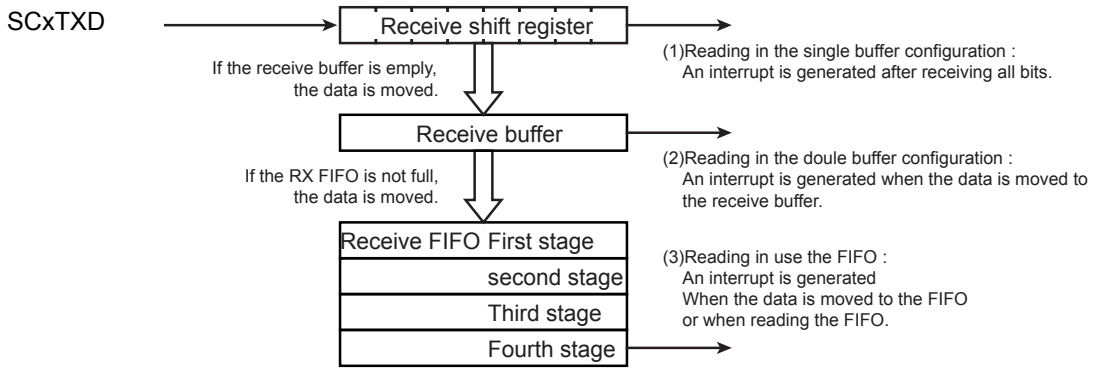


Figure 13-13 Receive Buffer/FIFO Configuration Diagram

13.12.1.1 Single Buffer / Double Buffer

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

Table 13-7 Receive Interrupt Conditions in use of Single Buffer / Double Buffer

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

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

13.12.1.2 FIFO

When the FIFO is used, a receive interrupt occurs on depending on the timing described in Table 13-8 and the condition specified with SCxRFC<RFIS>.

Table 13-8 Receive Interrupt Conditions in use of FIFO

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

13.12.2 Transmit interrupts

Figure 13-14 shows the data flow of transmit operation and the route of read.

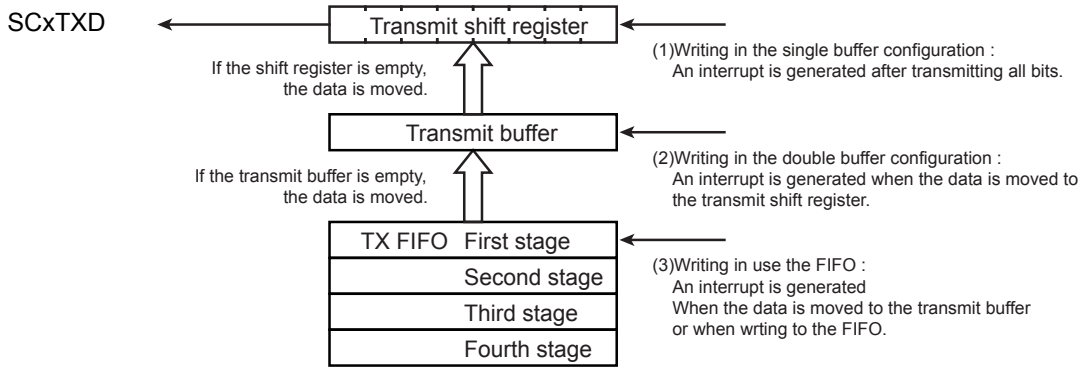


Figure 13-14 Transmit Buffer / FIFO Configuration Diagram

13.12.2.1 Singe Buffer / Double Buffer

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

Table 13-9 Transmit Interrupt conditions in use of Single Buffer/Double Buffer

Buffer Configurations	UART modes	IO interface modes
Single Buffer	Just before the stop bit is sent	Immediately after the raising / falling edge of the last SCxSCLK pin (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	When a data is moved from the transmit buffet to the transmit shift register. 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.	

13.12.2.2 FIFO

When the FIFO is used, a transmit interrupt occurs depending on the timing described in Table 13-10 and the condition specified with SCxTFC<TFIS>.

Table 13-10 Transmit Interrupt conditions in use of FIFO

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

13.12.3 Error Generation

13.12.3.1 UART Mode

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

13.12.3.2 I/O Interface Mode

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

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

13.13 DMA Request

DMA transfer can be started at the timing of interrupt request.

When you perform a DMA transfer, please set up the bit of a SCxDMA.

Please refer to the chapter of "product information" for the channel which can be used for a DMA request with this product.

Note 1: In case using DMA transfer by transmit or receive interrupt request, enabled DMA and set transmit and receive registers after generating software reset by SCxMOD<SWRST>.

Note 2: When the DMA transfer is used, the FIFO cannot be used.

Note 3: If transmission is performed using the DMA transfer with double-buffering, write two transmission data to the buffer, and then start up the DMA.

13.14 Software Reset

Software reset is generated by writing SCxMOD2<SWRST[1:0]> as "10" followed by "01".

As a result, SCxMOD0<RXE>, SCxMOD1<TXE>, SCxMOD2<TBEMP><RBFLL><TXRUN>, SCxCR <OERR><PERR><FERR> are initialized. And the receive circuit and the transmit circuit become initial state.

Other states are maintained.

13.15 Operation in Each Mode

13.15.1 Mode 0 (I/O interface mode)

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

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

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

13.15.1.1 Transmit

(1) Clock Output Mode

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

  Data is output from the SCxTXD pin and the clock is output from the SCxSCLK pin each time the CPU writes data to the transmit buffer. When all data is output, an interrupt (INTTXx) is generated.

- If the transmit double buffer is enabled (SCxMOD2<WBUF> = "1")

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

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

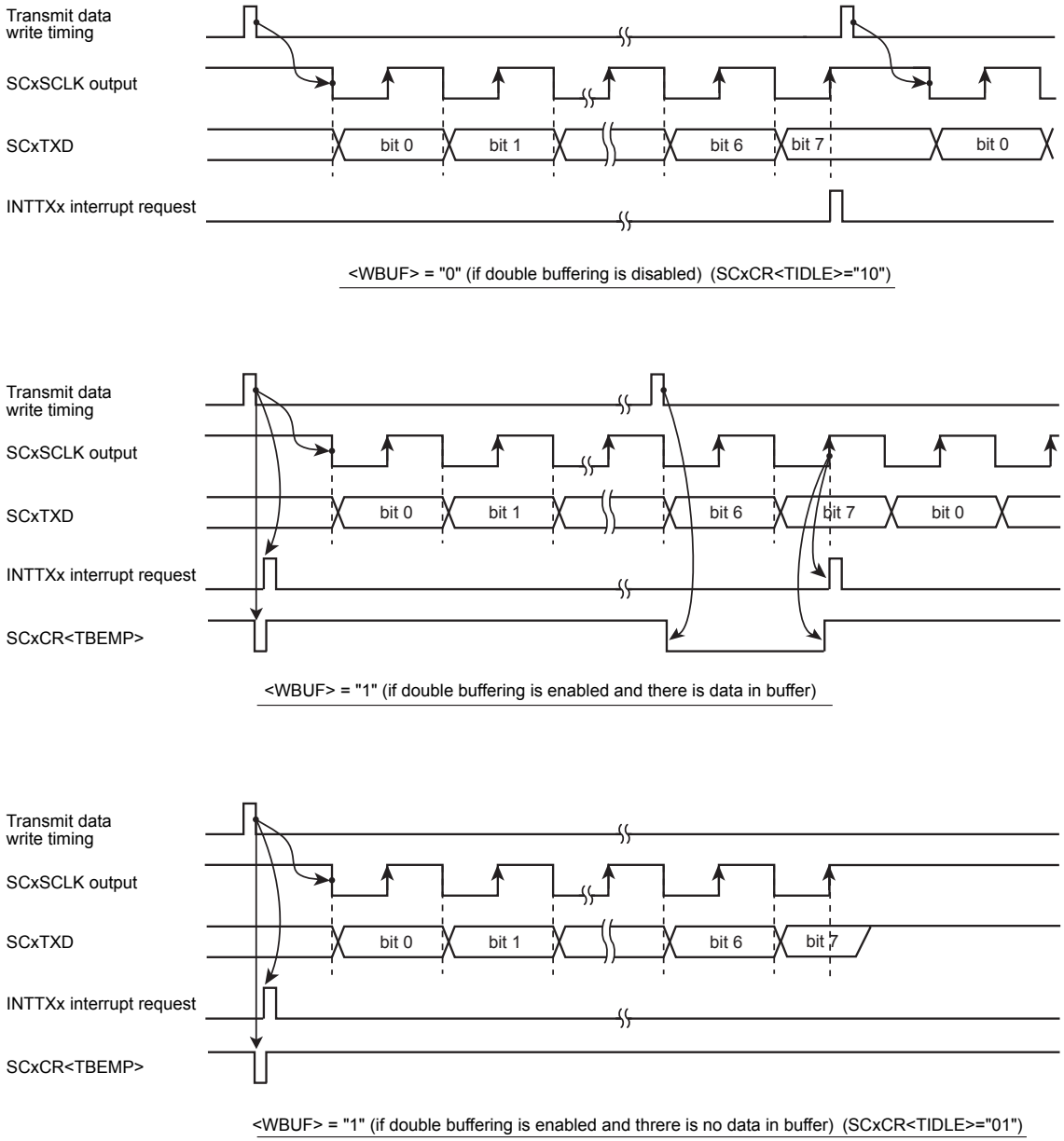


Figure 13-15 Transmit Operation in the I/O Interface Mode (Clock Output Mode)

(2) Clock Input Mode

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

  If the clock is input in the condition where data is written in the transmit buffer, 8-bit data is output from the SCxTXD pin. When all data is output, an interrupt INTTXx is generated. The next transmit data must be written before the timing of point "A" as shown in Figure 13-16.

- If double buffer is enabled (SCxMOD2<WBUF> = "1")

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

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

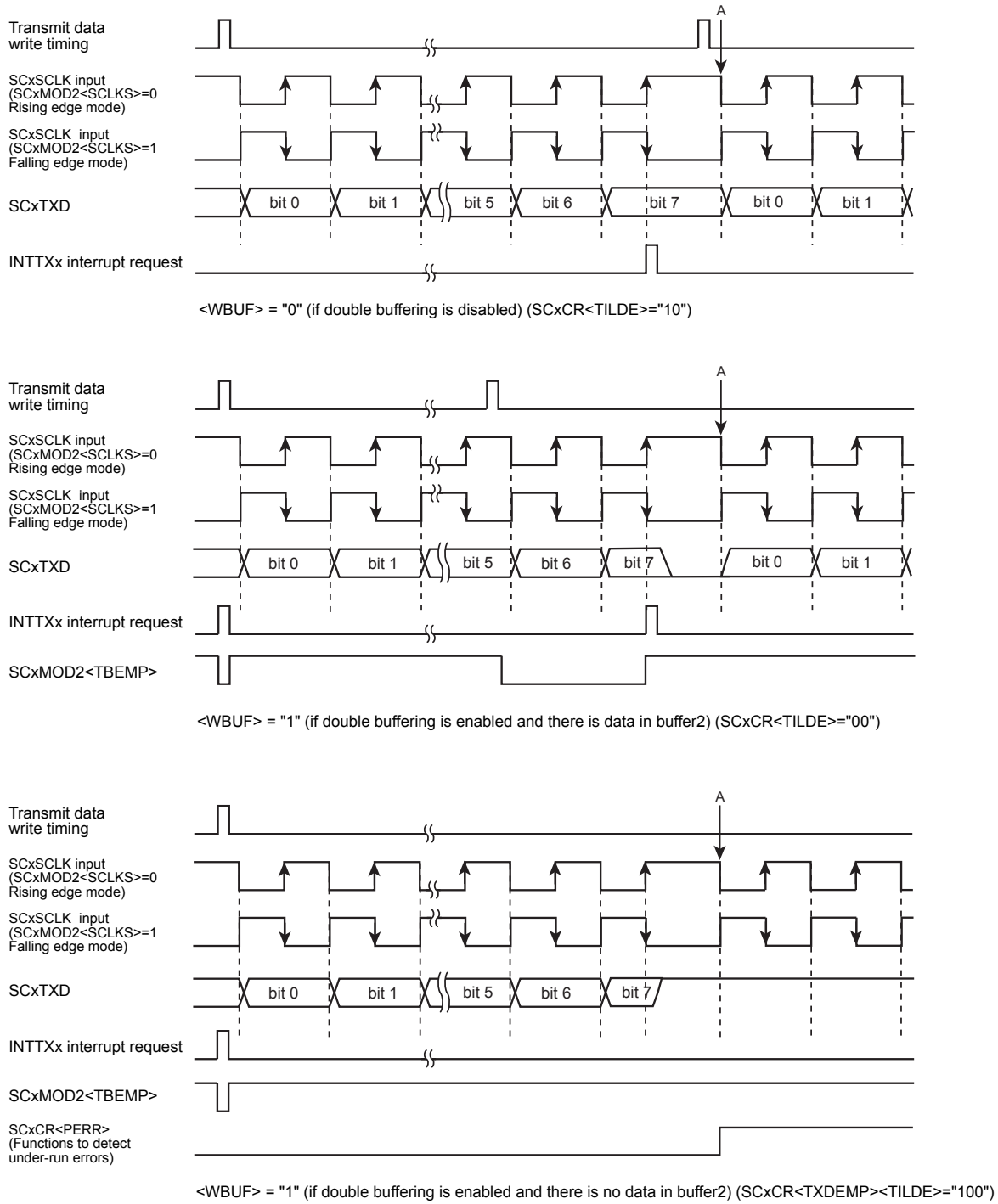


Figure 13-16 Transmit Operation in the I/O Interface Mode (Clock Input Mode)

13.15.1.2 Receive

(1) Clock Output Mode

The clock output starts by setting the receive enable bit SCxMOD0<RXE> to "1".

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

  A clock is output from the SCxSCLK pin and the next data is stored into the shift register each time the CPU reads received data. When all the 8 bits are received, the INTRXx interrupt is generated.

- If double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data stored in the shift register is moved to the receive buffer and the receive buffer can receive the next frame. A data is moved from the shift register to the receive buffer, SCxMOD2<RBFLL> is set to "1" and the INTRXx is generated.

  When a data is in the receive buffer, if the data is not read from the receive buffer before completing reception of the next 8 bits, the INTRXx interrupt is not generated and the clock output stops. In this state, reading data from the receive buffer allows data in the shift register to move to the receive buffer and thus the INTRXx interrupt is generated and data reception resumes.

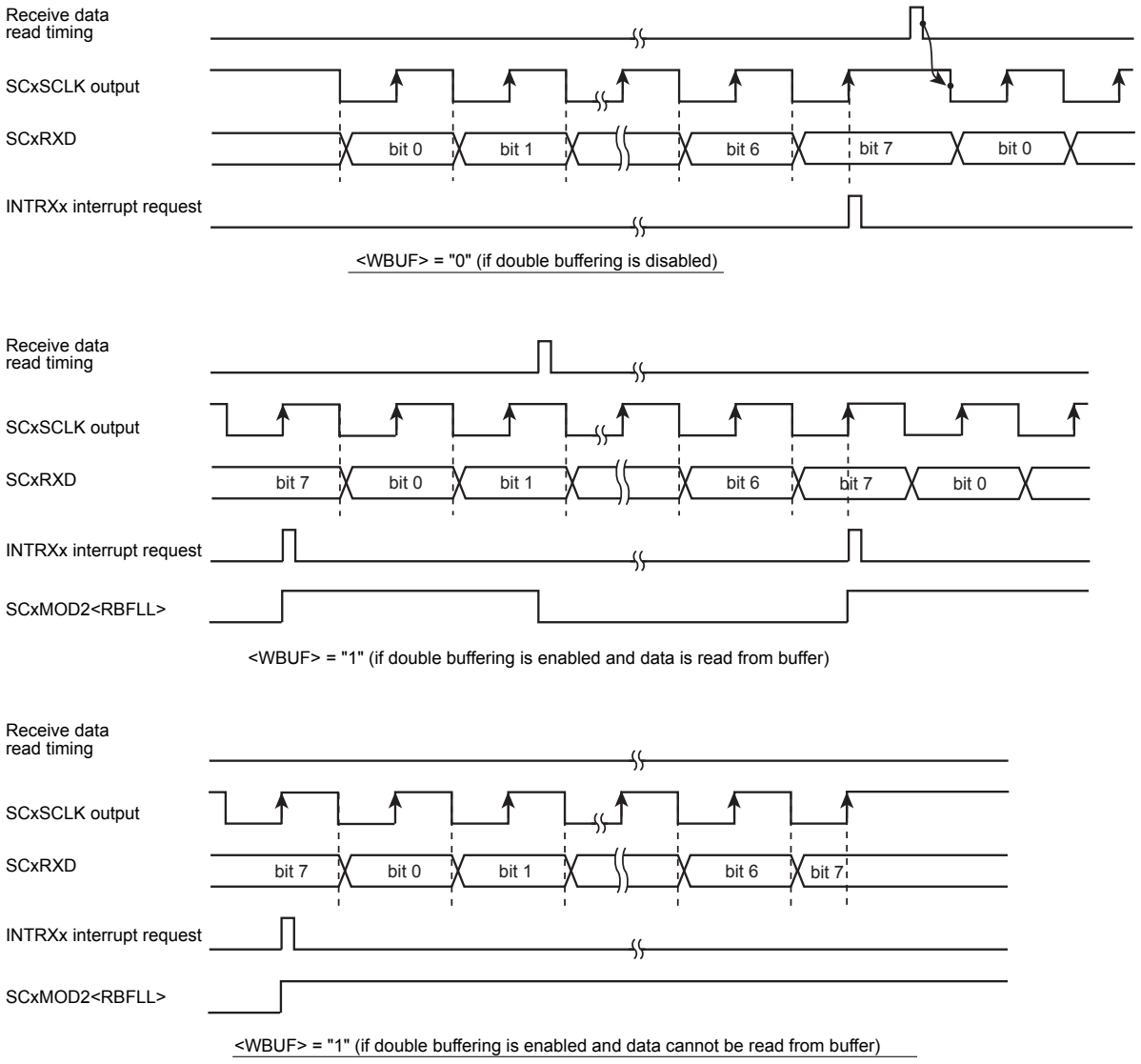


Figure 13-17 Receive Operation in the I/O Interface Mode (Clock Output Mode)

(2) clock input mode

In the clock input mode, receiving double buffering is always enabled, the received data can be moved to the receive buffer from the shift register, and the receive buffer can receive the next data successively.

The INTRXx receive interrupt is generated each time received data is moved to the receive buffer.

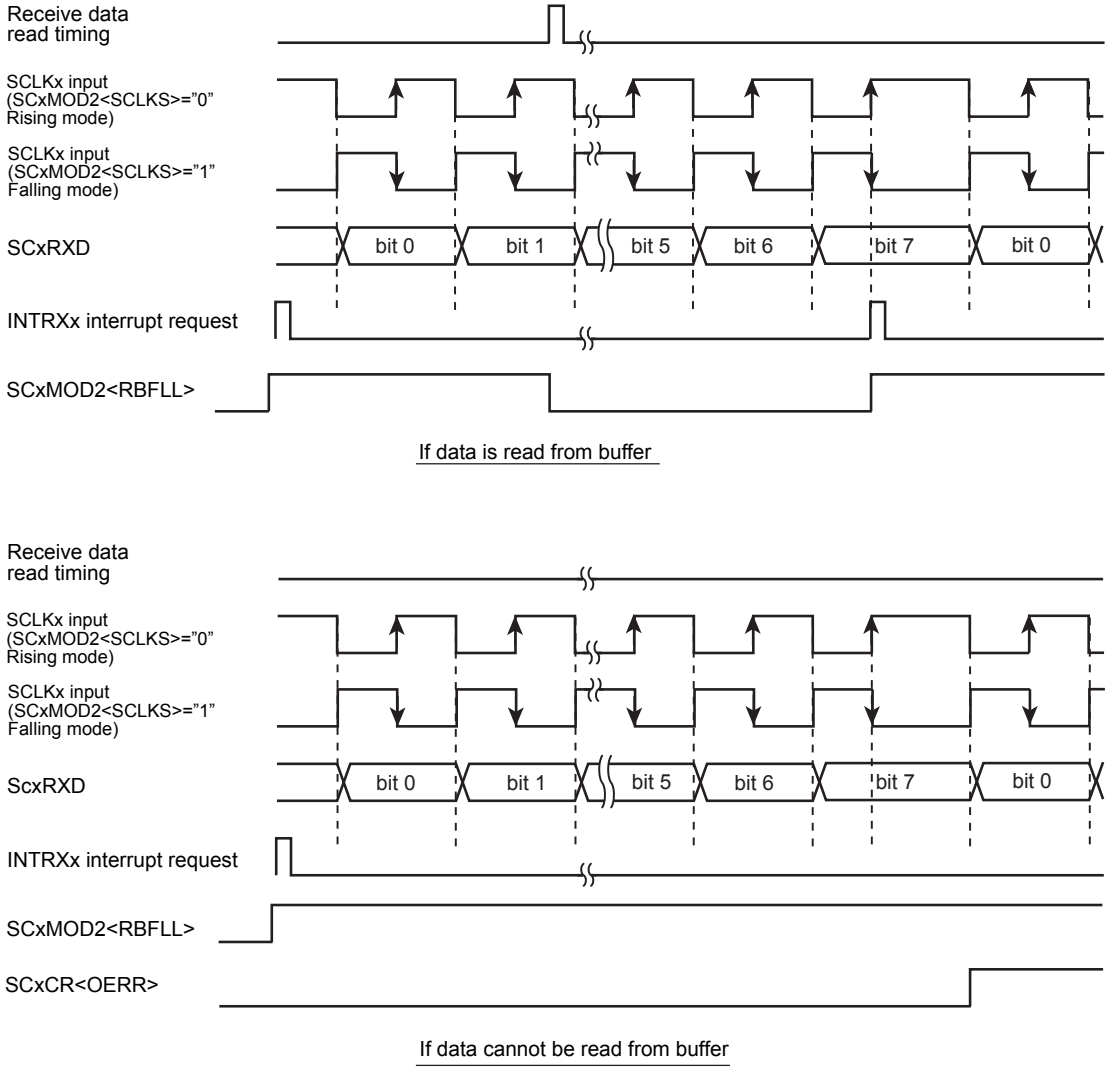


Figure 13-18 Receive Operation in the I/O Interface Mode (Clock Input Mode)

13.15.1.3 Transmit and Receive (Full-duplex)

(1) Clock Output Mode

- If double buffers are disabled (SCxMOD2<WBUF> = "0")

  Clock is output when the CPU writes data to the transmit buffer.

  Subsequently, a data is shifted into receive buffer and the INTRXx is generated. Concurrently, a data written to the transmit buffer is output from the SCxTXD pin, the INTTXx is generated when transmission of all data has been completed. Then, the clock output stops.

  The next round of data transmission and reception starts when the data is read from the receive buffer and the next transmit data is written to the transmit buffer by the CPU. The order of reading the receive buffer and writing to the transmit buffer can be freely determined. Data transmission is resumed only when both conditions are satisfied.

- If double buffers are enabled (SCxMOD2<WBUF> = "1")

  Clock is outputted when the CPU writes data to the transmit buffer.

  A data is shifted into the receive shift register, moved to the receive buffer, and the INTRXx is generated. While a data is received, a transmit data is output from the SCxTXD pin. When all data are sent out, the INTTXx is generated and the next data is moved from the transmit buffer to the transmit shift register.

  If the transmit buffer has no data to be moved to the transmit buffer (SCxMOD2<TBEMP> = "1") or when the receive buffer is full (SCxMOD2<RBFLL> = "1"), the clock output stops. When both conditions, receive data is read and transmit data is written, are satisfied, the clock output is resumed and the next round of data transmission and reception is started.

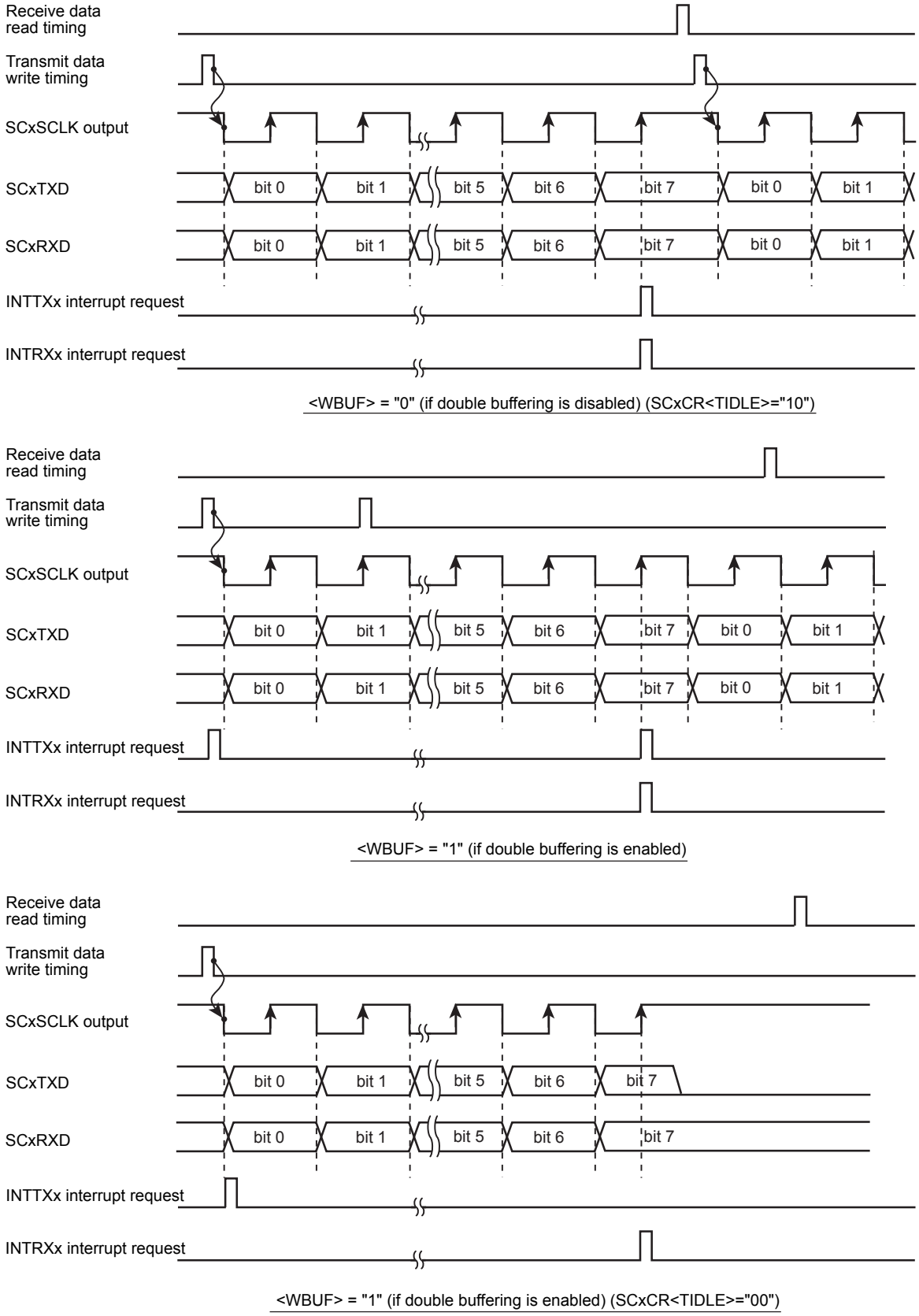


Figure 13-19 Transmit/Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) Clock Input Mode

- If double buffers are disabled. (SCxMOD2<WBUF> = "0")

  When receiving data, double buffer is always enabled regardless of the SCxMOD2 <WBUF> settings.

  A data written in the transmit buffer is outputted from the SCxTXD pin and a data is shifted into the receive buffer when the clock input becomes active. The INTTXx is generated upon completion of data transmission. The INTRXx is generated when the data is moved from shift register to receive buffer after completion of data reception.

  Note that transmit data must be written into the transmit buffer before the clock input for the next data (data must be written before the point A in Figure 13-20). Data must be read before completing reception of the next data.

- If double buffers are enabled. (SCxMOD2<WBUF> = "1")

  The INTTXx is generated at the timing the transmit buffer data is moved to the transmit shift register after completing data transmission from the transmit shift register. At the same time, data received is shifted to the shift register, it is moved to the receive buffer, and the INTRXx is generated.

  Note that transmit data must be written into the transmit buffer before the clock input for the next data (data must be written before the point A in Figure 13-20). Data must be read before completing reception of the next data.

  Upon the clock input for the next data, transmission from transmit shift register (in which data has been moved from transmit buffer) is started while receive data is shifted into receive shift register simultaneously.

  If data in receive buffer has not been read when the last bit of the data is received, an overrun error occurs.

  If there is no data written to transmit buffer when clock for the next data is input, an underrun error occurs. The level which is specified by SCxCR<TXDEMP> is output to SCxTXD pin.

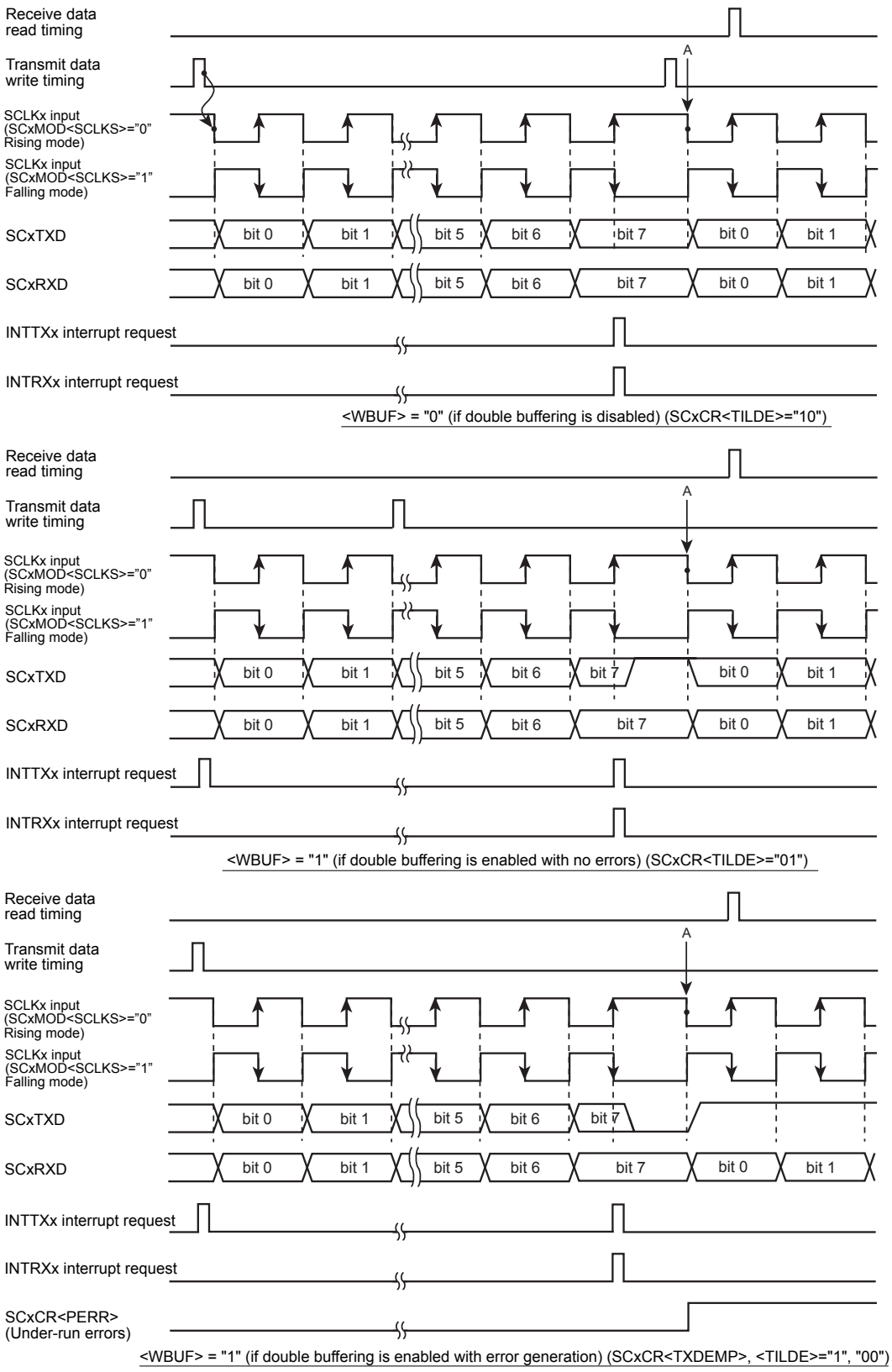


Figure 13-20 Transmit/Receive Operation in the I/O Interface Mode (Clock Input Mode)

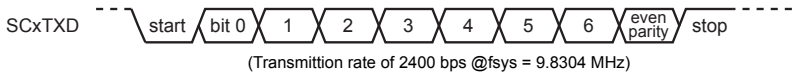
13.15.2 Mode 1 (7-bit UART mode)

The 7-bit UART mode is selected by setting SCxMOD<SM[1:0]> to "01".

In this mode, parity bits can be added to the transmit data stream; SCxCR<PE> controls the parity enable/disable setting.

When <PE> is set to "1" (enable), either even or odd parity may be selected using the SCxCR<EVEN>. The length of the stop bit can be specified using SCxMOD2<SBLEN>.

The following table shows the control register settings for transmitting in the following data format.



(Transmittion rate of 2400 bps @fsys = 9.8304 MHz)

Clocking conditions	system clock:	High-speed (fc)
	High-speed clock gear:	x 1 (fc)
	Prescaler clock:	fperiph/2 (fperiph = fsys)

		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	-	0	0	1	0	1	Set 7-bit UART mode
SCxCR	←	x	1	1	x	x	x	0	0	Even parity enabled
SCxBRCR	←	0	0	1	0	0	1	0	0	Set 2400bps
SCxBUF	←	*	*	*	*	*	*	*	*	Set transmit data

x: don't care - : no change

13.15.3 Mode 2 (8-bit UART mode)

The 8-bit UART mode is selected by setting SCxMOD0<SM[1:0]> to "10". In this mode, parity bits can be added and parity enable/disable is controlled using SCxCR<PE>. If <PE> = "1" (enabled), either even or odd parity can be selected using SCxCR<EVEN>.

The control register settings for receiving data in the following format are as follows:

(Reception rate of 9600 bps @fc = 9.8304 MHz)

Clocking conditions	System clock:	High-speed (fc)
	High-speed clock gear:	x 1 (fc)
	Prescaler clock:	fperiph/2 (fperiph = fsys)

		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	0	0	1	0	0	1	Set 8-bit UART mode
SCxCR	←	x	0	1	x	x	x	0	0	Odd parity enabled
SCxBRCR	←	0	0	0	1	0	1	0	0	Set 9600bps
SCxMOD0	←	-	-	1	-	-	-	-	-	Reception enabled

x: don't care - : no change

13.15.4 Mode 3 (9-bit UART mode)

The 9-bit UART mode is selected by setting SCxMOD0<SM[1:0]> to "11". In this mode, parity bits must be disabled (SCxCR<PE> = "0").

The most significant bit (9th bit) is written to SCxMOD0<TB8> for transmitting data. The data is stored in SCxCR<RB8> for receiving data.

When writing or reading data to/from the buffers, the most significant bit must be written or read first before writing or reading to/from SCxBUF.

The stop bit length can be specified using SCxMOD2<SBLEN>.

13.15.4.1 Wakeup function

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting SCxMOD0<WU> to "1".

In this case, the interrupt INTRXx will be generated only when SCxCR<RB8> is set to "1".

Note: The SCxTXD pin of the slave controller must be set to the open drain output mode using the PxOD.

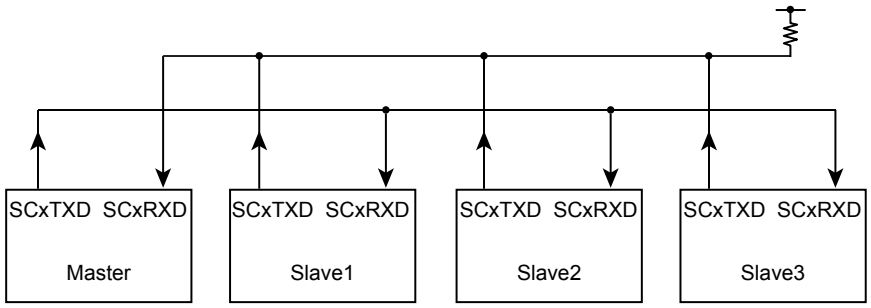


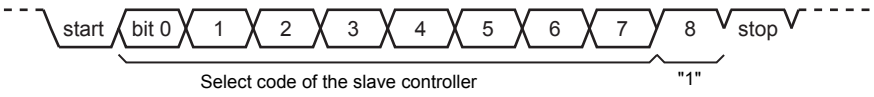
Figure 13-21 Serial Links to Use Wake-up Function

13.15.4.2 Protocol

1. Select the 9-bit UART mode for the master and slave controllers.

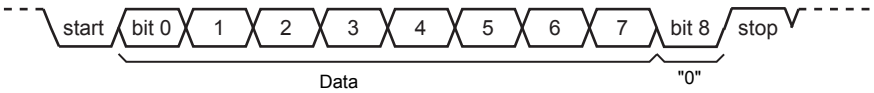
2. Set SCxMOD<WU> to "1" for the slave controllers to make them ready to receive data.

3. The master controller is to transmit a single frame of data that includes the slave controller select code (8 bits). In this, the most significant bit (bit 8) <TB8> must be set to "1".



4. Each slave controller receives the above data frame; if the code received matches with the controller's own select code, it clears the <WU> to "0".

5. The master controller transmits data to the designated slave controller (the controller of which SCxMOD<WU> is cleared to "0"). In this, the most significant bit (bit 8) <TB8> must be set to "0".



6. The slave controllers with the <WU> set to "1" ignore the receive data because the most significant bit (bit 8) <RB8> is set to "0" and thus no interrupt (INTRXx) is generated. Also, the slave controller with the <WU> set to "0" can transmit data to the master controller to inform that the data has been successfully received.

14. I2C Bus Interface

The TMPM066/067/068FW contains I2C Bus Interface with typical I2C bus standard.

This module also supports Toshiba's proprietary data format called " Free data format".

14.1 Features

The main features are as follows.

- Allows selection between master and slave.
- Allows selection between transmission and reception.
- Support multiple masters (arbitration, clock synchronization recognition).
- Baud rate (Support standard mode and fast mode, fast mode plus (FM+))
- Supports the addressing format of 7 bit only.
- Supports transfer data sizes of 1 to 8 bits.
- Provides one transfer (transmission or reception) complete interrupt (level sensitive).
- Enable or disable the interrupts.
- Arbitration lost detection interrupt (pulse): 1
- Bus free detection interrupt (pulse): 1
- NACK receive detection interrupt (pulse): 1
- Enable or disable the interrupts
- Monitor function for the I2C bus status
- Supports Toshiba's unique data format called "Free data format".
- Address match wake-up function (low-power consumption mode release function)

Table 14-1 I2C Bus Standard specifications

I2C bus feature	I2C Standard	TMPM066/067/068FW
STANDARD mode (up to 100KHz)	Required	Supported
FAST mode (up to 400KHz)	Required	Supported
FAST mode plus (up to 1MHz)	Option	Supported
Hs (High speed) mode (up to 3.4Mbps)	Required	Not Supported
7-bit addressing	Required	Supported
10-bit addressing	Required	Not Supported
START byte	Required	Supported
Noise canceller	Required	Supported (digital)
Slope control	Required	Not Supported
I/O at power off	Required	Supported
Schmidt (VIL/VHL)	VDD × 0.3 / VDD × 0.7	Not Supported
Output current at ¨VOL = 0.4V, VDD > 2V	3mA	Supported

14.2 Configuration

The configuration is shown in Figure 14-1.

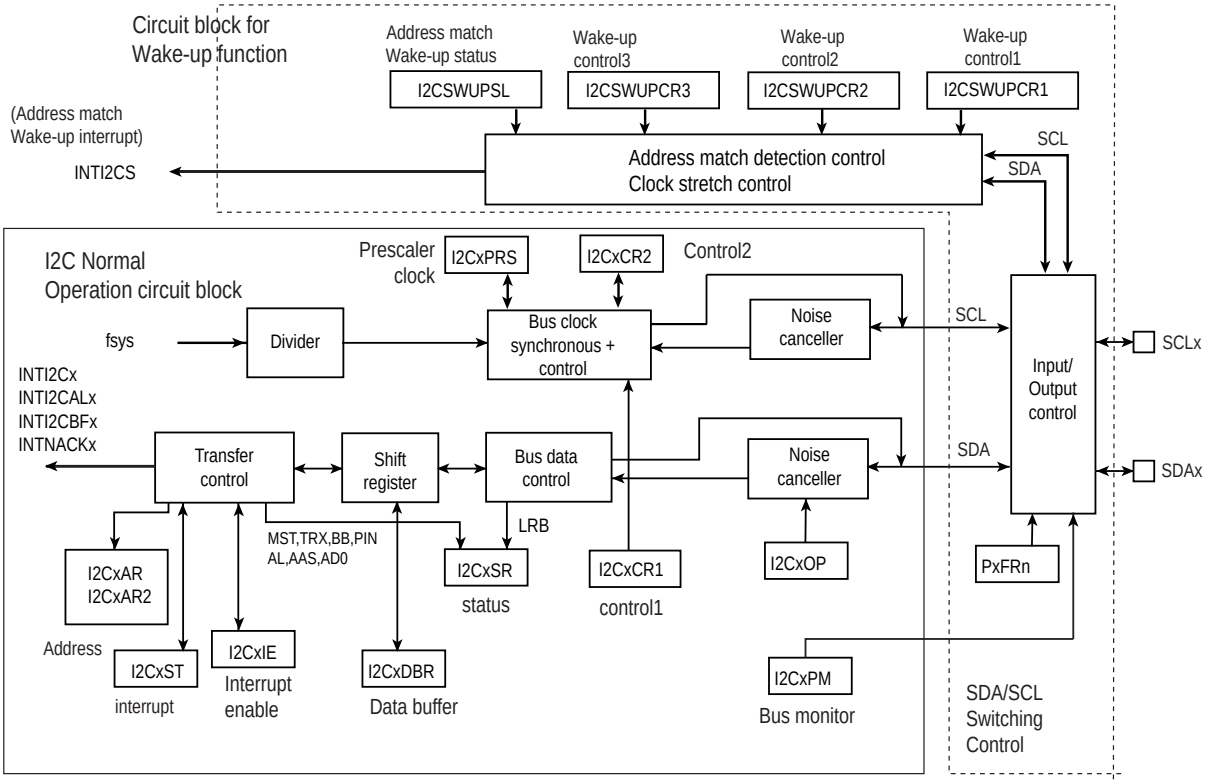


Figure 14-1 I2C Bus Block

14.2.1 I2C Bus mode

The I2C bus is connected to devices via the SDA and SCL pins and can communicate with multiple devices.

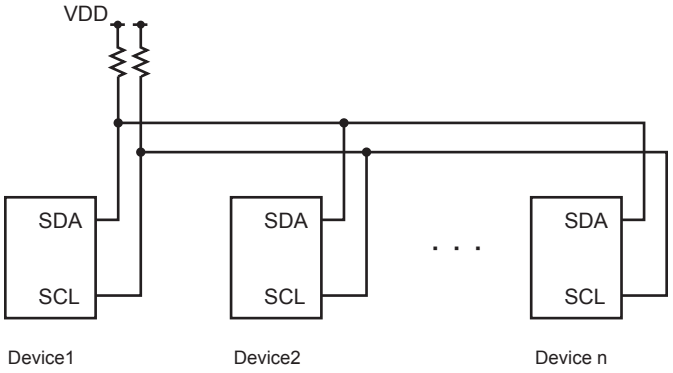


Figure 14-2

This module operates as a mater or slave device on the I2C bus.The master device drives the serial clock line (SCL) of the bus, send 8-bits address, and sends or receives data of 1 to 8bits.

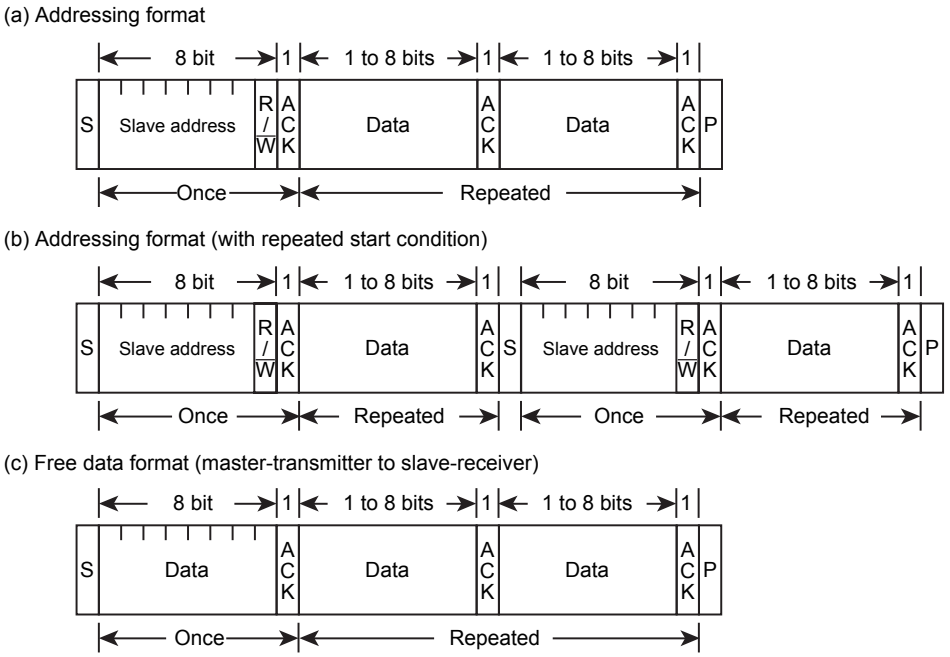
The slave device send a 8-bits addresses and sends or receives serial data of 1 to 8 bits in synchronization with serial clock on the bus.

The device that operates as a receiver can output an acknowledge signal after reception of serial data and the device the operates as a transmitter can receive that acknowledge signal, regardless of whether the device is a master or slave. The master device can output a clock for the acknowledge signal.

In multimaster mode in which multiple masters exist on the same bus, serial clock synchronization and arbitration lost to maintain consistency of serial data are supported.

14.2.1.1 I2C Bus Mode Data Format

Figure 14-3 shows the data formats used in the I2C bus mode.



Note) S : Start condition
R/$\overline{W}$: Direction bit
ACK : Acknowledge bit
P : Stop condition

Figure 14-3 I2C Bus Mode Data Formats

14.3 Register

14.3.1 Registers for each channel

The following registers table and address of the I2C bus interface.

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

I2C main function control register

Register name		Address(Base+)
Control register 1	I2CxCR1	0x0000
Data buffer register	I2CxDBR	0x0004
I2C bus1st address register	I2CxAR	0x0008
Control register 2	I2CxCR2 (writing)	0x000C
Status register	I2CxSR (reading)	
Prescaler Clock setting register	I2CxPRS	0x0010
Interrupt Enable Register	I2CxIE	0x0014
interrupt Register	I2CxST	0x0018
Optional Function Register	I2CxOP	0x001C
Monitor for the I2C Bus Line	I2CxPM	0x0020
I2C bus 2nd address register	I2CxAR2	0x0024

Wake-up function control register

Register name		Address(Base+)
I2C Wake-up Control register 1	I2CSWUPCR1	0x0000
I2C Wake-up Control register 2	I2CSWUPCR2	0x0001
I2C Wake-up Control register 3	I2CSWUPCR3	0x0002
I2C Wake-up status register	I2CSWUPSL	0x0003

Note: These register can be read or written by an only word access.

14.3.2 I2CxCR1(Control register 1)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-5	BC[2:0]	R/W	Select the number of bits per transfer (Note 1)

<BC[2:0]>	When <ACK> = 0		When <ACK> = 1	
	Number of clock cycles	Data length	Number of clock cycles	Data length
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

4	ACK	R/W	Master mode 0: Acknowledgement clock pulse is not generated. 1: Acknowledgement clock pulse is generated. Slave mode 0: Acknowledgement clock pulse is not counted. 1: Acknowledgement clock pulse is counted.
3	NOACK	R/W	Slave address match detection and general call detection. 0:the slave address match or general call detection are enabled. 1:the slave address match or general call detection are disabled. When I2CxAR<ALS>="1", this bit has no meaning.
2-0	SCK[2:0]	R/W	Select internal SCL output clock frequency (Note 2).

000	n = 0	100	n = 4
001	n = 1	101	n = 5
010	n = 2	110	n = 6
011	n = 3	111	n = 7

Note 1: Writing to this register must be done before a start condition is generated or after a stop condition is generated or between the instant when an address or data transfer interrupt occurs and the instant when the internal iterrupt is released. Do not write to this register during address or data transfer.

Note 2: For details on the SCL line clock frequency, refer to "14.4.3 Serial Clock".

Note 3: The initial value for selecting a frequency is <SCK[2:0]>=000 and is independent of the read initial value.

14.3.3 I2CxDBR (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)	Receive data The received data is stored in the LSB.
		W (Transmit)	Transmit data The transmission data must be written in to the register from the MSB (bit 7).

When the master needs to transmit a slave address, the transfer target address is written to I2CxDBR<DB [7:1]> and the transfer direction is specified in I2CxDBR<DB[0]> as follows:

<DB[0]>=0:Master(transmission) →Slave/reception

<DB[0]>=1:Master(reception) ←Slave/transmission

When all the bits in the I2CxDBR register are written as "0",a general call can be sent out on the bus.

In both transmission and reception modes, a write to the I2CxDBR register or read from the I2CxDBR register release the internal interrupt after the current transfer and initiates the next transfer.

I2CxDBR is provided as a transmit/receive buffer, it should be used as a dedicated transmit buffer in transmit mode and as a dedicated receive buffer in receive mode. This register in transmit mode and as a dedicated receive buffer in receive mode. this register should be accessed on a transfer -by - transfer basis.

Note: This register are initialized by Hardware RESET only. Software RESET can not initialize and are keeping the last data.

14.3.4 I2CxAR (I2Cbus 1st 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 1st slave address when the I2C 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 I2CxAR 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 I2CxAR to "0x00" in slave mode. (If I2CxAR is set to "0x00", it's recognized that the slave address matches the START byte ("0x01") of the I2C standard received in slave mode.)

14.3.5 I2CxCR2(Control register 2)

This register serves as I2CxSR 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	I2CM	-	SWRES	
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 INTI2CIx interrupt request 0: − 1: Clear interrupt request
3	I2CM	W	I2C operation control 0: Disable 1: Enable
2	-	R	Read as "0".
1-0	SWRES[1:0]	W	Software reset generation Write "10" followed by "01" to generate a reset. For detail, refer to "14.4.9 Software Reset".

Note 1: 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 mode.

Note 2: The I2CxCR2<I2CM> bit cannot be cleared to 0 to disable I2C operation while transfer operation is being performed. before clearing this bit, mask sure that transfer operation is completely stopped by reading the status register.

Note 3: Don't change the contents of the registers, except <SWRST[1:0]>, when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an interrupt request is generated for stopping the data transfer until it is released.

14.3.6 I2CxSR (Status Register)

This register serves as I2CxCR2 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	AD0	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	request monitor and SCL line monitor 0: request generated, SCL line is "L". 1: request not generated, SCL line is "Free".
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	AD0	R	General call detection 0: − 1:Detected
0	LRB	R	Last received bit monitor 0:Last received bit "0" 1:Last received bit "1"

14.3.7 I2CxPRS(Prescaler Clock setting register)

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

Bit	Bit Symbol	Type	Function
31-5	-	R	Read as "0".
4-0	PRSCK[4:0]	R/W	Prescaler clock frequency for generating the Serial clock 00000: P = divided by 32 00001: P = divided by 1 - - - - - - - - - - 11111: P = divided by 31

Note: Refer to Section "14.3.2 I2CxCR1(Control register 1)", "14.4.3 Serial Clock".

14.3.8 I2CxIE(Interrupt 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	-	SELPINCD	DMACTXR	DMACRXR	INTNACK	INTI2CBF	INTI2CAL	INTI2C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	-	R	Read as "0".
6	SELPINCD	R/W	Extended PIN open condition selection. 0: The PIN is not opened (=1) by reading DBR. 1: The PIN is opened (=1) by reading DBR.(If the DMA is used, use this setting.)
5	DMACTXR	R/W	Enable /disable a DMAC transmission request output. 0: Disable 1: Enable
4	DMACRXR	R/W	Enable /disable a DMAC reception request output. 0: Disable 1: Enable
3	INTNACK	R/W	INTNACK interrupt enable or disable setting.(note1) 0: Disable 1: Enable
2	INTI2CBF	R/W	INTI2CBF interrupt enable or disable setting.(note1) 0: Disable 1: Enable
1	INTI2CAL	R/W	iNTI2CAL interrupt enable or disable setting.(note1) 0: Disable 1: Enable
0	INTI2C	R/W	I2C interrupt enable or disable setting.(note1) 0: Disable 1: Enable

Note 1: These settings enable/disable an interrupt request from the I2C function. The interrupt setting of the CPU is required as well.

14.3.9 I2CxST(I2C Interrupt status register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	NACK	I2CBF	I2CAL	I2C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	-	R	Read as "0".
3	NACK	R	Indicates INTNACK interrupt status.(note1) 0: No interrupt 1: Interrupts occur
		W	Clear a INTNACK interrupt. 0: Invalid 1: Clear
2	I2CBF	R	Indicates INTI2CBF interrupt status.(note1) 0: No interrupt 1: Interrupts occur
		W	Clear a INTI2CBF interrupt. 0: Invalid 1: Clear
1	I2CAL	R	Indicates INTI2CAL interrupt status.(note1) 0: No interrupt 1: Interrupts occur
		W	Clear a INTI2CAL interrupt. 0: Invalid 1: Clear
0	I2C	R	Indicate INTI2C interrupt status.(note1) 0: No interrupt 1: interrupt occur.
		W	Clear a INTI2C interrupt. 0: Invalid 1: Clear

Note 1: These bits indicate the status of I2C interrupt generation regardless of setting of I2CxIE<IE>

14.3.10 I2CxOP(Expansion Function 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	-	SA2ST	SAST	NFSEL	RSTA	GCDI	SREN	MFACK
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	-	R	Read as "0".
6	SA2ST	R	Discriminates the received slave address.(Update when<AAS>=1) 0: Not matched <SA2[6:0]>. 1: Matched <SA2[6:0]>
5	SAST	R	Discriminates the received slave address.(Update when<AAS>=1) 0: Not matched <SA[6:0]>. 1: Matched <SA[6:0]>
4	NFSEL	R/W	Select Noise cancellation. 0: Normal use 1: When using the Address Match Wake-up function to release from Low power consumption mode.
3	RSTA	R	Detects a Repeated START flag. 0: Not detected. 1: Detected.
		W	0: Cleared. 1: No effect.
2	GCDI	R/W	Sets general-call detection.(Valid only when <NOACK>=0) 0: Detection is ON. 1: Detection is OFF.
1	SREN	R	Sets the repeated START output.(Valid only when the MCU in master mode.) 0: Repeated START has completed. 1: Repeated START is ongoing.
		W	0: Disable 1: Enable
0	MFACK	R/W	Selects an ACK output. 0: ACK output 1: NACK output (Note) This bit cannot be used in free-data format.

14.3.11 I2CxPM(Monitor for the I2C Bus Line register)

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

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1	SDA	R	Monitors the SDA pin. 0: Low level. 1: High level.
0	SCL	R	Monitors the SCL pin. 0: Low level. 1: High level.

14.3.12 I2CxAR2(I2C 2nd Slave 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	SA2							SA2EN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-1	SA2[6:0]	R/W	2nd Slave Address data.
0	SA2EN	R/W	Enable or disable for using the 2nd Slave Address. 0: No use 1: Used.

Note: Set "0" to I2CxCR1<NOACK> before using this register.

14.3.13 I2CSWUPCR1(I2C wake-up control register1)

	7	6	5	4	3	2	1	0
bit symbol	BUSY	SGCDI	ACK	I2RES	RW	-	GC	INTEND
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
7	BUSY	R	0: A stop condition is detected. 1: A start condition is detected.
6	SGCDI	R	0: General call detection On 1: General call detection off
5	ACK	R/W	0: ACK output (Output "0") 1: No ACK output (Output "1")
4	I2RES	R/W	I2CBUS reset status. (Note1,2,3) 0: reset release 1: reset done
3	RW	R	Monitor Transmit / Receive request from Master device. (Note4) 0: Slave Receive 1: Slave Transmit
2	-	R	Read as "0".
1	GC	R	Status of General call detection. 0: Not detected. 1: detected.
0	INTEND	R	Release the I2CS interrupt. 0: - 1: Release Interrupt

Note 1: When <I2RES>=1, I2CBUS is reset; however, the setting is not automatically returned to "0". When the reset state is released, set <I2RES>=0 prior to the reset.

Note 2: When the reset is performed with <I2RES>, all read registers of the I2C are initialized; however write data is not initialized.

Note 3: After the reset is released with <I2RES>, the circuit operations are performed along to the preset values. Therefore, set slave addresses or ACK signals when <I2RES>= 1.

Note 4: <RW> is "0" regardless of transmit/receive request from the master when the addresses does not match.

14.3.14 I2CSWUPCR2(I2C wake-up control register2)

	7	6	5	4	3	2	1	0
bit symbol	WUPSA1							-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
7-1	WUPSA1[6:0]	R/W	1st Slave address.
0	-	R	Read as "0".

14.3.15 I2CSWUPCR3(I2C wake-up control register3)

	7	6	5	4	3	2	1	0
bit symbol	WUPSA2							WUPSA2EB
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
7-1	WUPSA2[6:0]	R/W	2nd Slave address.
0	WUPSA2EN	R/W	Selection for the 2nd Slave address. 0: No use. 1: Used.

14.3.16 I2CSWUPSL(I2C status register)

	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	WUPSA2	WUPSA	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
7-3	-	R	Read as "0".
2	WUPSA2	R	0: Not matched 2nd Slave address. 1: Matched 2nd Slave address.
1	WUPSA	R	0: Not matched 1st Slave address. 1: Matched 1st Slave address.
0	-	R	Read as "0".

14.4 Function

14.4.1 Selection for a Slave Address Match Detection or General Call Detection

For a slave device, the I2CxCR1<NOACK> is done to enable or disable for a slave address match detection and general call detection in slave mode.

The I2CxCR1<NOACK> is done to enable or disable for a slave address match detection and general call detection in salve mode.

when I2CxCR1<NOACK>=0 and I2CxOP<GCDI>=1,it is enable for a slave address match detection and general call detection. If I2CxOP<GCDI>=0, General call detection is disable.

when I2CxCR1<NOACK>=1,it is disable for a slave address match detection and general call detection.

The slave device ignores a slave address and general calls sent from the master and returns non-acknowledgment. the INTI2Cx interrupt request are not generated.

In master mode, the bit of I2CxCR1<NOACK> is ignored and has no effect on operation.

14.4.2 Setting the Number of Bits per Transfer and the Acknowledgement Mode

1. The number of clocks for Data Transfer

   The number of clocks for data transfer is specified with I2CxCR1<BC[2:0]> and I2CxCR1<ACK>.

   Setting I2CxCR1<ACK> to "1" selects the acknowledge mode. In the acknowledgment mode, the Master device adds One clocks for the acknowledgment after sending data's clocks and then require the INTI2Cx interrupt request.

   By setting <ACK> to "0", the non-acknowledgment mode is activated. In the non-acknowledgment mode, the Master device require the INTI2Cx interrupt request after counting the Clocks for Data bits sending.

   And the Slave device also done same operation.

   However, the second byte is necessary to be controlled by software to generate an ACK signal on the contents of the second byte.

   (For example, the second byte is received in non-acknowledge mode; in receive interrupt routine, an ACK clock is output in a pseudo manner using 1-bit data output process.)

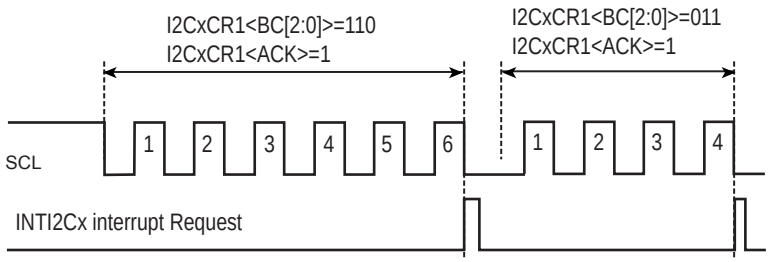


Figure 14-4 Number of clocks for Data Transfer with I2CxCR1<BC[2:0]>,I2CxCR1<ACK>

shows the relationship between the number of clocks of data transfer, I2CxCR1<BC[2:0]>, and I2CxCR1<ACK>.

Table 14-2 the number of clocks per transfer

<BC[2:0]>	acknowledge mode (I2CxCR1<ACK>)			
	0: No-ACK		1: ACK mode	
	Data length	Number of clock cycles	Data length	Number of clock cycles
000	8	8	8	9
001	1	1	1	2
010	2	2	2	3
011	3	3	3	4
100	4	4	4	5
101	5	5	5	6
110	6	6	6	7
111	7	7	7	8

<BC[2:0]> is cleared to "000" by a start condition.

Therefore, a slave address and the direction bit are always transferred in 8-bit length, or otherwise <BC[2:0]> maintains the specified value.

Note: Set I2CxCR1<ACK> before transmitting or receiving a slave address. If I2CxCR1<ACK> has been cleared, matching of slave addresses, and detection of a direction bit are not performed properly.

2. Acknowledgement signal output

In Acknowledgement mode, SDA pin is changing as follows during the clock cycle and generates acknowledgement signals.

- In master mode

  In transmitter mode, the SDA pin should be opened until the transmitter receives an ACK signal from the receiver as acknowledgement.

  In receive mode, the SDA pin should be pulled "Low" to generate an ACK signal until the receiver sends an ACK signal to the transmitter when I2CxOP<MFACK>=0. When I2CxOP<MFACK>=1, the SDA pin should be pulled "High".

- In Slave mode

  When the received slave address matches the slave address specified in I2CxAR<SA[6:0]>, or the general-call is received, the SDA pin is pulled "Low" to receive an acknowledge signal from the receiver until the receiver sends an ACK signal.

  In transmitter mode, data transfer after the received slave address matches the slave address or the general-call is received, the SDA should be opened to receive an acknowledge signal from the receiver.

In receiver mode, the SDA pin should be pulled "Low" to generate an acknowledge signal.

Table 14-3 Comparison between the status of SCL and SDA in acknowledge mode

Mode	Pin	Condition	Transmitter	Receiver
Master	SCL	-	Clocks are added for an acknowledge signal.	Clocks are added for an acknowledge signal.
	SDA	-	The SDA pin is opened to receive an acknowledge signal.	The SDA pin is pulled "Low" for an acknowledge signal. or SDA pin is pulled "High" for a Non-acknowledge signal.
Slave	SCL	-	Checking and counting the clocks for an acknowledge signal.	Checking and counting the clocks for an acknowledge signal.
	SDA	A slave address matches the preset slave address, or the receiver receives a general-call	-	the SDA pin is pulled "Low" for an acknowledge signal.
		A slave address matches the preset slave address, or the receiver receives a general-call	The SDA pin is opened to receive an acknowledge signal.	the SDA pin is pulled "Low" for an acknowledge signal.

14.4.3 Serial Clock

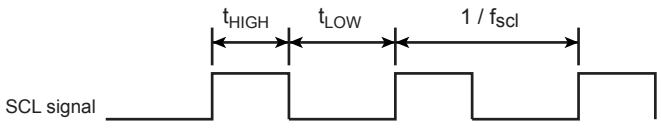
14.4.3.1 Clock source

In master mode, it is set the High and Low period of Serial clock with I2CxOP<NFSEL>, I2CxCR1<SCK[2:0]>.

<NFSEL>	<SCK[2:0]>	t_{HIGH} (i × T_{prsck})	t_{LOW} (j × T_{prsck})	Frequency (kHz)
		i	j	(note1)
0	000	8	12	800.0
	001	10	14	666.6
	010	14	18	500.0
	011	22	26	333.3
	100	38	42	200.0
	101	70	74	111.1
	110	134	138	58.8
	111	262	266	30.3

(Note1): In case of <PRSCK[4:0]> =00001, fsys=16MHz.(Tprsck=62.5ns)

(Note2): When using the Address Match Wake-up function to release from Low power consumption mode, use with <NFSEL>=1.



t_{LOW} = i × Tprsck
t_{HIGH} = j × Tprsck
fscl = 1 / (t_{LOW} + t_{HIGH})

Figure 14-5 I2C SCL output

Note: The tHIGH period may differ from the specified value if the rising edge becomes blunt depending on the combination of bus load capacitance and pull-up register. If the clock synchronization func-

tion for synchronizing clocks from multiple clocks is used, the actual clock period may differ from the specified setting.

In master mode, the hold time when a start conditions generated and the setup time when a stop condi-tion is generated are defined as the following.

Hold time:

I2CxOP<SREN>=0: t_{HIGH} [S]

I2CxOP<SREN>=1: 8Tprsck [S]

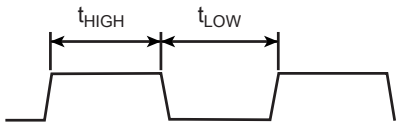
Setup time:

I2CxPRS<PRSCK>=1: t_{HIGH} [S]

I2CxPRS<PRSCK>≠1: t_{HIGH} - Tprsck [S]

When I2CxCR2<PIN> is set to "1" in slave mode, the time to the release of SCLx is defined as tLOW [S].

In both master and slave modes, the high level period must be 4×Tprsck [s] or longer and the low lev-el period must be 5×Tprsck [s] or longer for externally input serial clocks, regardless of the I2CxCR1<SCK [2:0]>. setting.



t_{LOW} >= (4 × Tprsck)
t_{HIGH} >= (5 × Tprsck)

Figure 14-6 SCLK Input

The serial clock rate to be output from the master is set through I2CxCR1<SCK[2:0]> and the I2CxPRS<PRSCK[4:0]>. The prescaler clock which is divided according to I2CxPRS<PRSCK[4:0]> is used as the reference clock for generating the serial clock. The prescaler clock is further divided according to I2CxCR1<SCK[2:0]> and used as the serial clock.

- <Serial transfer rate>

  The serial clock rate (f_{scl}) is determined by prescaler setting value "p" (I2CxPRS<PRSCK[4:0]>,p=1 to 32) and serial clock setting value "n" (I2CxCR1<SCK[2:0]>,n=0 to 7) based on the operating frequency (fsys) as follows:

  <NFSEL>=0:

$$\text{Serial clock rate : } f_{scl}\ (\text{kHz}) = \frac{\text{fsys(MHz)}}{p\times(2^{n+2}+16)} \times 1000$$

p:	prescaler setting	I2CxPRS<PRSCK[4:0]>,	1 to 32
n:	serial clock setting	I2CxCR1<SCK[2:0]>,	0 to 7

  The allowed range of prescaler setting value "p" (I2CxPRS<PRSCK[4:0]>) varies depending on the operating frequency (fsys) and must satisfy the following condition.

  <NFSEL>=0

$$50\text{ns} < \text{Prescaler clock width: Tprsck (ns)} \leq 150\text{ns}$$

Note: Setting the prescaler clock width out of this range is prohibited in both master and slave modes.

The serial clock rate may not be constant due to the clock synchronization function.

n: <SCK[2:0]>			<NFSEL>=0 p: <PRSCK[4:0]> = 00001 (divide by1) Ratio to fsys
0	0	0	20
0	0	1	24
0	1	0	32
0	1	1	48
1	0	0	80
1	0	1	144
1	1	0	272
1	1	1	528

Note: Writing to these bits must be done before a start condition is generated or after a stop condition is benerated.Writing to these bits during transfer will cause unexpected operation.

- <Prescaler clock width (=noise cancellation width)

  The prescaler clock width (Tprsck)(= noise cancellation width) is determined by prescaler setting value "p" (I2CxPRS<PRSCK[4:0]>,P=1 to 32) based on the operating frequency (fsys) as follows:

$$\text{Prescaler clock width : Tprsck (ns) (=Noise cancellation width)} = \frac{1}{\text{fsys(MHz)}} \times 1000 \times p$$

14.4.3.2 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.

I2C has a clock synchronization function to ensure proper transfer operation even when multiple master exist on a bus.

For example, the clock synchronization procedure for a bus with two masters is shown below.

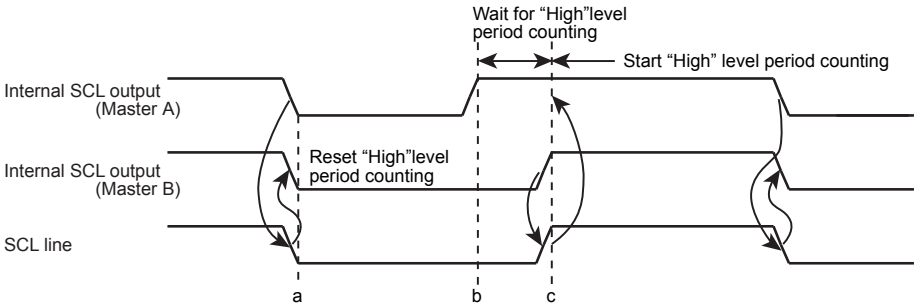


Figure 14-7 Example of Clock Synchronization

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.

14.4.4 Configuring the I2C as a Master or a Slave

Setting I2CxCR2<MST> to "1" configures the I2C to operate as a master device.

Setting I2CxCR2 <MST> to "0" configures the I2C as a slave device. I2CxCR2<MST> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

14.4.5 Configuring the I2C as a Transmitter or a Receiver

Setting I2CxCR2<TRX> to "1" configures the I2C as a transmitter. Setting <TRX> to "0" configures the I2C as a receiver.

At the slave mode, when data is transmitted in the addressing format.

If the value of the direction bit (R/$\overline{W}$) is "1", I2CxSR<TRX> is set to "1" by the hardware. If the bit is "0", I2CxSR<TRX> is set to "0".

As a master device, the I2C receives acknowledgement from a slave device. If the direction bit of "1" is transmitted, I2CxSR<TRX> is set to "0" by the hardware. If the direction bit is "0", <TRX> changes to "1". If the I2C does not receive acknowledgement, <TRX> retains the previous value.

Note: When I2CxCR1<NOACK>=1, the slave address detection and general call detection are disabled, and thus I2CxSR<TRX> remains unchanged.

Table 14-4 I2CxSR<TRX> operation

mode	direction bit	condition for state change	Changed <TRX> after state change
Slave mode	0	Received Slave address = I2CxAR<SA [6:0]>,I2CxAR2<SA2[6:0]>	0
	1		1
Master mode	0	ACK received	1
	1		0

When I2C is used in free data format, the salve address and direction bit are not recognized and bits immediately following a start condition are handled as data. Therefore, I2CxSR<TRX> is not changed by the hardware.

14.4.6 Generating Start and Stop Conditions

When I2CxSR<BB> is "0", writing "1" to I2CxCR2<MST, TRX, BB, PIN> causes the I2C 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. I2CxCR1<ACK> must be set to "1" in advance.

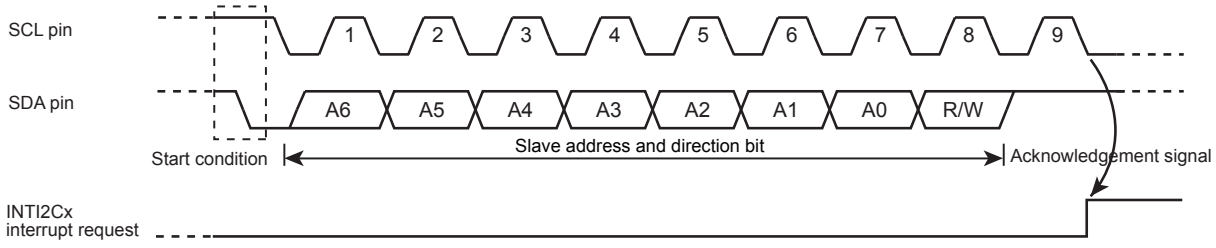


Figure 14-8 Generating the Start Condition and a Slave Address

When I2CxSR<BB> is "1", writing "1" to I2CxCR2<MST, TRX, PIN> and "0" to I2CxCR2<BB> causes the I2C 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.

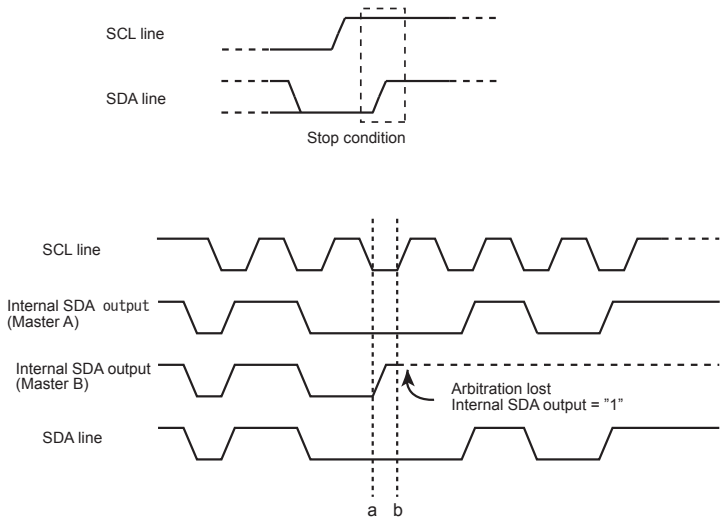


Figure 14-9 Generating the Stop Condition

I2CxSR<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).

The following table shows typical setting examples according to the I2CxSR state.

Although the I2CxCR2<MST>,<TRX>,<BB>,<PIN> bits are given independent functions, they are used in typical combinations, as shown below, according to the I2CxSR setting.

I2CxSR			I2CxCR2				Operation
<MST>	<BB>	<PIN>	<MST>	<TRX>	<BB>	<PIN>	
0	0	1	0	0	0	0	Wait for a start condition as a slave.
			1	1	1	1	Generate a start condition.
1	1	0	1	1	0	1	Generate a stop condition.

Note: When writing to these bits, do not change I2CxCR2<I2CM> by mistake.

14.4.7 Interrupt Service Request and Release

In master mode, a I2C bus interface request (INTI2Cx) is generated when the transfer of the number of clock cycles set by I2CxCR1<BC[2:0]> and I2CxCR1<ACK> is completed.

In slave mode, if the following conditions are established based on the above-mentioned conditions, INTI2Cx interrupt occurs.

- When I2CxCR1<NOACK> is "0",after output of the acknowledge signal which is generated when the received slave address matches the slave address set to I2CxAR<SA[6:0]>.
- When I2CxCR1<NOACK> is "0",after the acknowledge signal is generated when a general-call address is received.

When an interrupt request (INTI2Cx) is generated, I2CxCR2<PIN> is cleared to "0". While <PIN> is cleared to "0", the I2C pulls the SCL line to the "Low" level.

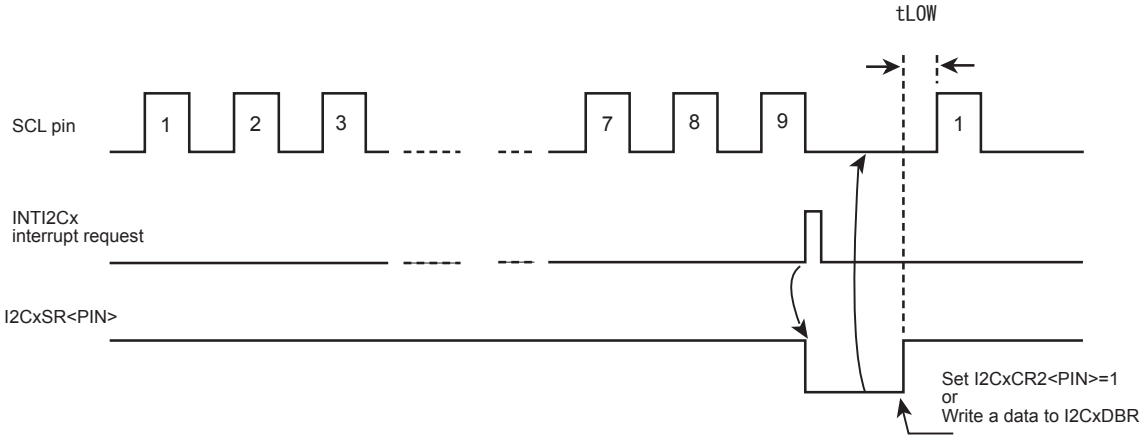


Figure 14-10 I2CxSR<PIN> and SCL

I2CxSR<PIN> is set to "1" when data is written to or read from I2CxDBR.

It takes a period of t_{LOW} for the SCL line to be released after I2CxSR<PIN> is set to "1". When the program writes "1" to I2CxSR<PIN>, it is set to "1". However, writing "0" does not clear this bit to "0".

Note: When arbitration lost occurs while a slave address and direction bit are transferred in the master mode, I2CxSR<PIN> is cleared to "0" and INTI2Cx occurs. This does not relate to whether a slave address matches I2CxAR<SA[6:0]>.

1. Bus free detection interrupt

   A bus free detection interrupt occurs when the I2CxSR<BB> flag of the device are changed (changed from BB=1 to BB=0).

   When a bus free detection interrupt is used, set "1" to I2CxIE<INTI2CBF>.

   Note that <BB> is used to detect a bus free status regardless of the value of the slave address. When a STOP condition is detected, the bus status changes to the bus free status from the bus busy status. Thus, the <BB> flag changes to the bus free status and the bus free interrupt occurs.

2. NACK detection interrupt

   When I2CxCR<ACK>=1 and I2CxIE<INTNACK>=1, if the master receives a NACK, a NACK reception detection interrupt occurs.

This operation is NACK detection in which this device acts as the transmitter in master mode or in slave mode.

An ACK detection interrupt occurs at the same timings INTI2Cx; therefore, check the interrupt with an I2CxST interrupt status flag.

14.4.8 I2C Bus mode

When I2CxCR2<I2CM> is set to "1". I2C bus mode is selected. ensure that the I2C bus interface pins are at "High" level before setting <I2CM> to "1". Also, ensure that the bus is free before switching the operating mode to the port mode.

Note: When I2CxCR2<I2CM> = 0, no value can be written to bits in the I2CxCR2 register other than I2CxCR2<I2CM> bit. Before setting I2CxCR2, write "1" into I2CxCR2<I2CM> to select the I2C bus mode.

14.4.9 Software Reset

If the I2C bus interface circuit locks up due to external noise, it can be initialized by using a software reset.

Writing sequentially "10" and "01" to I2CxCR2<SWRES[1:0]> causes software reset.

I2C is initialized after software reset; however, the I2CxCR2<I2CM> register and the I2CxDBR register are not initialized.

At this time, I2CxSR<LRB> becomes undefined. This flag state is the same as the pin state at which software reset has occurred.

14.4.10 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.

The I2C-bus arbitration takes place on the SDA line.

The arbitration procedure for two masters on a bus is shown below.

Up until the point a, Master A and Master B output the same data. At the point a, Master A outputs the "Low" level and Master B outputs the "High" level.

Then Master A pulls the SDA bus line to the "Low" level because the line has the wired-AND connection.

When the SCL line goes high at the point b, the slave device reads the SDA line data, i.e., data transmitted by Master A. At this time, data transmitted by Master B becomes invalid.

This condition of Master B is called "Arbitration Lost". Master B releases its SDA pin, so that it does not affect the data transfer initiated by another master. If two or more masters have transmitted exactly the same first data word, the arbitration procedure continues with the second data word.

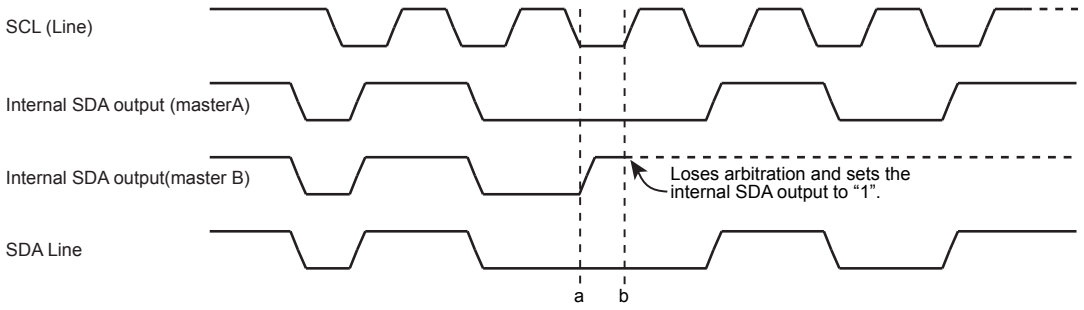


Figure 14-11 Arbitration Lost

If an arbitration lost interrupt is enabled, an AL flag is set at the point "b" in Figure 14-11. When subsequently data transfer is executed the number of preset clocks, an arbitration lost interrupt occurs.

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 I2CxSR<AL> is set to "1".

When I2CxSR<AL> is set to "1", I2CxSR<MST, TRX> are cleared to "0", causing the I2C to operate as a slave receiver.

Therefore, the serial bus interface circuit stops the clock output during data transfer after <AL> is set to "1".

I2CxSR<PIN> is cleared to "0" when data transfer is completed, the I2C pulls the SCL Line to the Low level.

An arbitration lost occurs during the transfer of the slave addresses and direction bit in Master B. In this case, Master B receives a slave address sent from other master devices, like other slave devices. Regardless of the match between the received slave address and I2CxAR<SA[6:0]>, <PIN> is cleared to "0" and then INTI2Cx is generated.

I2CxSR<AL> is cleared to "0" when data is written to or read from I2CxDBR or data is written to I2CxCR2.

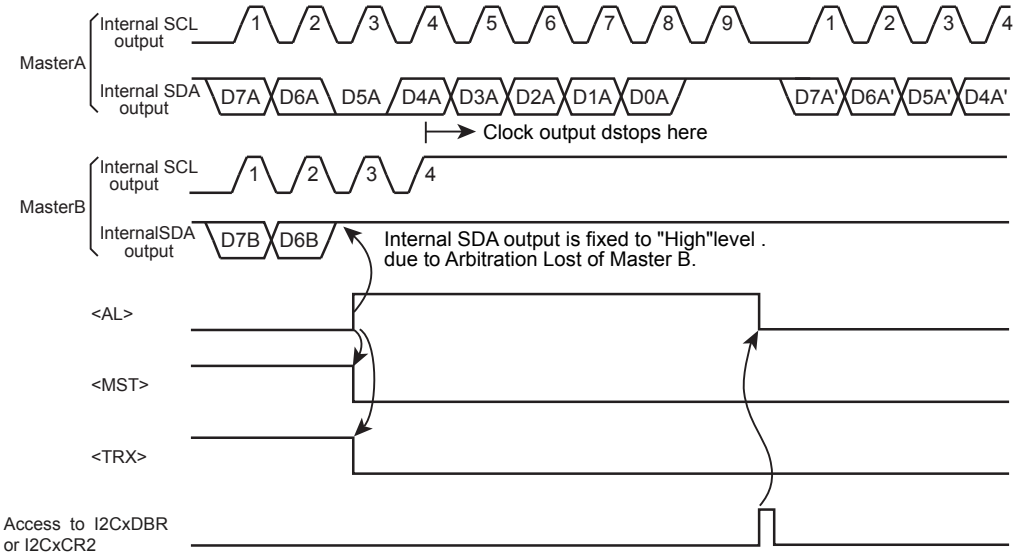


Figure 14-12 Example of Master B Lost Arbitration (D7A = D7B, D6A = D6B)

This MCU only supports normal arbitration lost function that conforms to I2C-bus specification. It does not support the arbitration lost function when a NACK is transferred. In this case, I2CxSR<LRB> should be checked in the interrupt service routine of the completion of transmission.

Note that multiple transmissions start almost simultaneously in a multi tasking fashion, the following cases is assumed:

(1) Arbitration is detected before the transmissions start, and they are cancelled.

(2) Arbitration is detected after the transmissions start.

(3) Arbitration is cancelled before the transmissions start. Arbitration cannot be detected.

For the device status, check the value of <AL>,<MST>

14.4.11 Slave Address Match Detection Monitor

Two slave addresses can be specified in this product. In I2C bus mode (I2CxAR<ALS> = 0),when Slave mode is selected, a match of slave addresses can be detected. Received slave addresses are compared with I2CxAR<SA[6:0]> and I2CxAR2<SA2[6:0]> sequentially. If the same address is used for I2CxAR<SA[6:0]> and I2CxAR2<SA2[6:0]>, only the comparison result with I2CxAR<SA[6:0]> is returned when the received slave address matches them. Note that if one slave address is used, use I2CxAR<SA[6:0]> only.

When I2CxCR1<NOACK> is cleared to "0", the address match detection is enabled. If I2CxSR<AAS> receives a general call or the slave address which is the same as the I2CxAR<SA[6:0]>, I2CxSR<AAS> is set to "1".

To use the second slave address, set the value to I2CxAR2<SA2[6:0]> and I2CxAR2<SA2EN>=1. Whether <SA[6:0]> or <SA2[6:0]>is used for match detection, this can be distinguish with I2CxOP<SA2ST> or <SAST>.

When "1" is set to I2CxCR1<NOACK>, address match detection is disabled. If this device receives a general call or the slave address which is the same as the I2CxAR<SA[6:0]> and the I2CxAR<SA2[6:0]>, I2CxSR<AAS> is not set to "1".

When Free data format mode(I2CxAR <ALS> is set to "1"), I2CxSR<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 I2CxDBR.

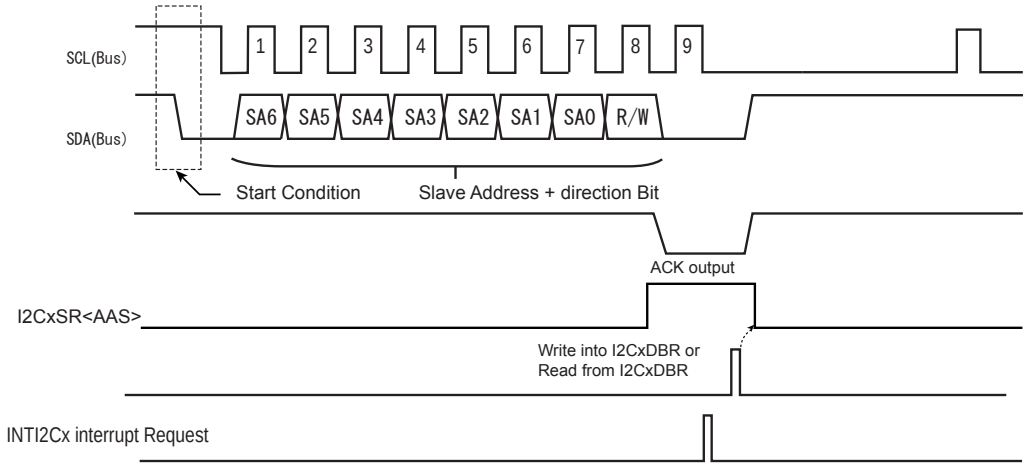


Figure 14-13 Slave address Match Detection Monitor

14.4.12 General-call Detection Monitor

In the I2C bus mode (I2CxAR<ALS> = 0), when Slave mode is selected, a slave address and general call can be detected.

When I2CxCR1<NOACK> = 0 and I2CxOP<GCDI>=0, if the device receives a general call (8 bits received immediately after a start condition are all zero), I2CxSR<AD0> is set to "1". (At this time, I2CxSR<AAS> is also set to "1".)

When I2CxCR1<NOACK>=1, a general call cannot be detected; therefore, if a general call is received, I2CxSR<AD0> remains "0". (At this time, I2CxSR<AAS> also remains "0".)

I2CxSR<AD0> is cleared to "0" when the start or stop condition is detected on the bus.

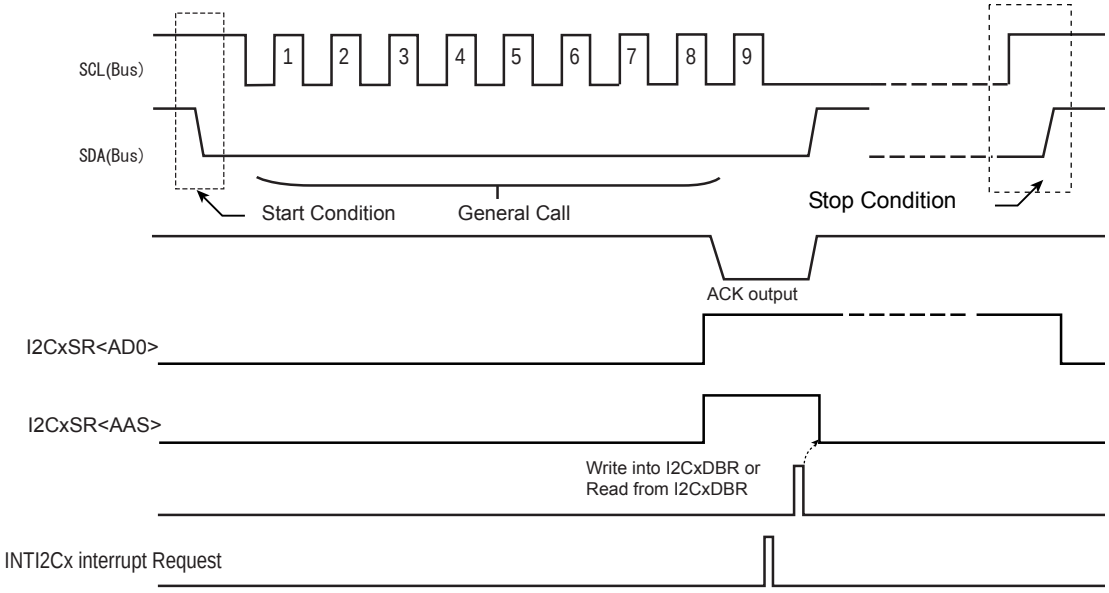


Figure 14-14 General-call Detection Monitor

14.4.13 Last Received Bit Monitor

I2CxSR<LRB> is always updated to the value of the SDA on the rising edge of the SCL Line.

In the acknowledgment mode, reading I2CxSR<LRB> immediately after generation of the INTI2Cx interrupt request causes ACK signal to be read.

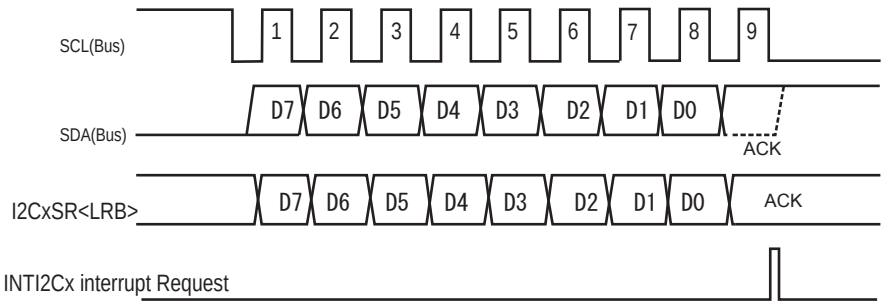


Figure 14-15 Last Received Bit Monitor

14.4.14 Setting of Slave Addressing and Address Recognition Mode

To use I2C bus mode, clear I2CxAR<ALS> to "0" and set a slave address to I2CxAR<SA[6:0]>.

If free data format that does not use a slave address is used, set "1" to I2CxAR<ALS>. Note that the I2C is used in free data format, a slave address and direction bit are not recognized. They are treated as data immediately after a start condition.

When the second slave address is used, set the value to I2CxAR2<SA2EN>. When <SA2EN>= 1, the second slave address is valid.

The received Slave addresses are compared with the first address (I2CxAR<SA[6:0]>) and the second address (I2CxAR2<SA2[6:0]>) sequentially.

The result of comparison is stored to I2CxIE<SAST> and <SA2ST>.

When the same address is used for the first address and the second address, only the detection result of the first address is returned when the received slave address matches them.

14.4.15 Noise Cancellation

The SCL pin and SDA pin have the noise cancelling function. For Normal use,Digital noise cancellation should be used and it is set with I2CxOP<NFSEL>=0.

To use the Address Match Wake-up function for releasing from the Low power consumption mode, set the Analog noise cancellation with I2CxOP<NFSEL>=1 before moving into that mode.

In digital noise cancellation, signals less than Tprsck (prescaler clock width) is eliminated as noise.

14.4.16 Repeated START Detection

In slave mode, when a repeated START is detected on the bus line, I2CxOP<RSTA> is set to "1". A repeated start is used for a master device to change the direction of transmission without terminating data transfer to a slave device.

Since <RSTA> is only initialized by a reset, clear the flag(<RSTA>=0) when stop condition occurs, bus free,.etc. (refer to Figure 14-16)

In case I2CxPRS<PRSCK[4:0]> ≠00001 in Master mode, <RSTA> is set to "1" after a start condition generation. Perform clear processing at the first INTI2Cx interrupt service routine, etc..

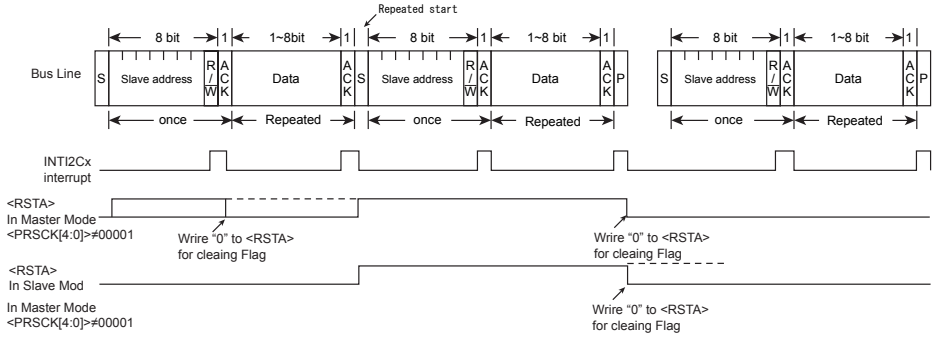


Figure 14-16 Repeated Start detection Flag

14.4.17 DMA request output Control

Data can be transferred using DMA in I2C mode. But one master device and one slave device are connected to a bus line and the number of transfer data is decided beforehand.

A request is output one clock after an INTI2Cx transmit/receive interrupt occurs; therefore, DMA sequential transfer between I2CxDBR and memory can be performed.

A request for DMA can be controlled with I2CxIE<DMARI2CTX> in transmission or with <DMARI2CRX> in reception respectively Enable or disable each register.

To transfer data using DMA, the number of bytes to be transferred should be preliminarily specified on the both transmission and reception.

If arbitration lost occurs during I2C transfer, the INTI2CALx interrupt and the INTI2Cx interrupt occurs, then the communication stops. A DMA request does not occur.

14.5 Control in the I2C Bus Mode

14.5.1 Data Transfer Procedure in the I2C Bus Mode

14.5.1.1 Device Initialization

After ensuring that the SDA and SCL pins are high (Bus free), set I2CxCR2<I2CM> to "1" for enable the I2C.

Next, write "1" to I2CxCR1<ACK]> and "0" to I2CxCR1<NOACK>.Writing "000" to I2CxCR1<BC [2:0]> at the time.

These setting enable acknowledge operation, slave address match detection and general call detection and set the data length to 8 bits. and set "tHIGH" and "tLOW" in I2CxCR1<SCK[2:0]>.

then, program I2CxAR by specifying a slave address at <SA[6:0]> and an address recognition mode at <ALS>. (<ALS> must be cleared to "0" when using the addressing format).

To configure the I2C Bus Interface as a slave receiver, ensure that the I2C bus interface pin is at "High" first.

Finally, write "0" to I2CxCR2<MST, TRX, BB>, "1" to I2CxCR2<PIN>, "00" to I2CxCR2<SWRES [1:0]> to configure the device as a slave receiver.

14.5.1.2 Generating the Start Condition and a Slave Address

Before generating the start condition or slave address, execute the following operations: check the bus free condition (I2CxSR<BB> = 0); set "1" to I2CxCR1<ACK>, and write data that contains a slave address of I2CxDBR and the direction bit. When "1" is written to I2CxCR2<TRX>, I2CxCR2<BB>, and I2CxCR2<PIN>, the slave address and direction bit are output to the I2C bus. Note that the time of tHIGH is required until the SCL pin falls after the start condition.

Subsequently, an INTI2Cx interrupt occurs on the falling edge of the 9th clock of SCL to clear I2CxSR<PIN> to "0".

At this time, SCL should be pulled "Low" level while I2CxSR<PIN> is "0". Only when the slave device returns an acknowledge signal, the direction of I2CxSR<TRX> is changed by hardware according to the direction bit at the timing when an INTI2Cx interrupt request occurs. If an acknowledge signal is not returned, I2CxSR<TRX> is not changed.

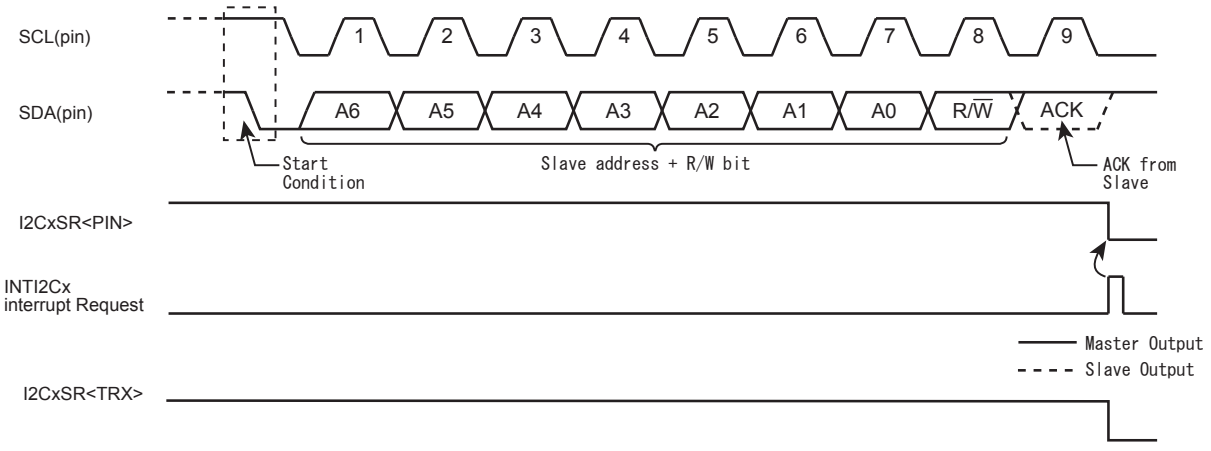


Figure 14-17 Generating the Start Condition and a Slave Address

14.5.1.3 Transferring a Data Word

At the end of a data word transfer, the INTI2Cx interrupt is generated to test I2CxSR<MST> to determine whether the I2C is in the master or slave mode

(1) Master mode (I2CxSR<MST> = 1)

Test <TRX> to determine whether the I2C is configured as a transmitter or a receiver.

(a) Transmitter mode (I2CxSR<TRX> = 1)

Check acknowledge from the receiver using the I2CxSR<LRB> flag.

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 I2CxDBR. If the data has different length, I2CxCR1<BC[2:0] > and I2CxCR1<ACK> are programmed and the transmit data is written into I2CxDBR.

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.

Note that even if the receiver requests the next data, a stop condition can be issued.

After the transfer is completed, the INTI2Cx interrupt request is generated, I2CxSR<PIN> is cleared to "0", and the SCL pin is pulled to the "Low" level.

To transmit more data words, test I2CxSR<LRB> again and repeat the above procedure.

I2CxSR <LRB> ="1" means the receiver does not request the data; therefore, the transmitter generates a stop condition (described later) to stop data transfer.

Note that even if the receiver requests the next data, data can be transmitted.

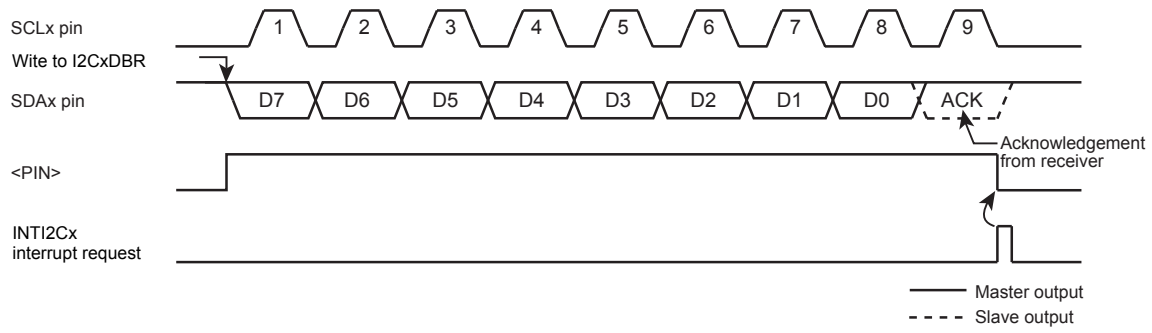


Figure 14-18 <BC[2:0]>= 000,<ACK>= 1 (Transmitter Mode)

(b) Receiver mode (I2CxSR<TRX> = 0)

If dummy data (0x00) is written to I2CxDBR, or "1" is set to I2CxCR2<PIN>, a 1-word transfer clock and acknowledge signal are output.

After an INTI2Cx interrupt indicating the completion of reception occurs, read receive data from I2CxDBR.

If the data has different length, set I2CxCR<BC[2:0]> again and set I2CxCR<ACK> to "1",and then Write the I2CxDBR to "0x00" or set the <PIN> to "1".

(Data that immediately after the slave address is transmitted is undefined.)

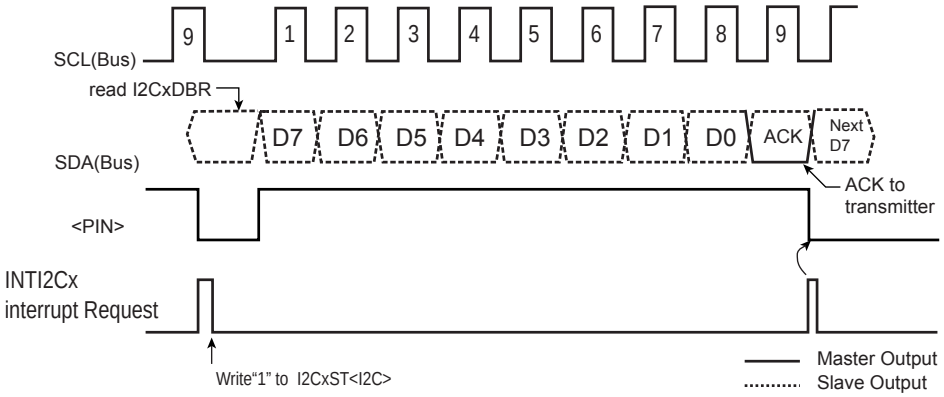


Figure 14-19 I2CxCR1<BC[2:0]>= 000,I2CxCR1<ACK>= 1 (Receiver Mode)

(c) The last word Received (I2CxSR<TRX>=0)

To determine the last word, perform the pseudo-communication without an acknowledge signal.

The process flows are described below:

To terminate the data transmission from the transmitter, <ACK> must be cleared to "0" immediately before reading the last data word.

Before Received the last data, following procedure is required to terminate the data transmission from the transmitter.

1. Read the received data from I2CxDBR
2. Set I2CxOP<MFACK> to 1 for the NACK transmission
3. Write the Dummy data (0x00) into I2CxDBR to set the I2CxCR2<PIN> as "1".

When "1" is set to I2CxCR2<PIN>, 1 - word transfer is started in NACK transmission. After 1-word transfer is complete, proceed with the following process:

1. Read the received data from I2CxDBR.
2. Generate the STOP condition, Terminates the data transmission.

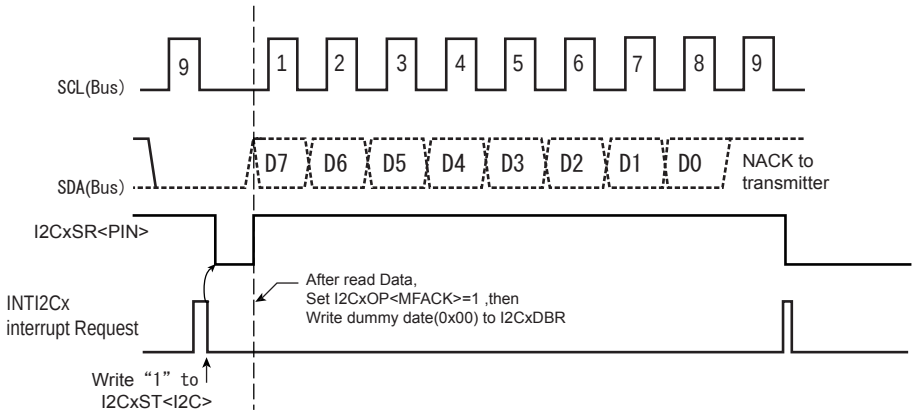


Figure 14-20 Terminating Data Transmission in the Master Receiver Mode

(2) Slave mode (I2CxSR<MST>=0)

In the slave mode, the normal slave mode processing or the processing as a result of Arbitration Lost is carried out.

In the slave mode, the I2C generates the INTI2Cx interrupt request on following status.

- when I2CxCR1<NOACK> is "0", the received slave address matches I2CxAR<SA[6:0]> address
- when I2CxCR1<NOACK> is "0", the I2C has received a general-call address
- when a data transfer has been completed in response to a received slave address matching or a general-call in master mode.
- If the I2C 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 INTI2Cx interrupt request is generated.

Table 14-5 shows the operation of interrupt requests and I2CxSR<PIN> after arbitration lost

Table 14-5 The operation of INTI2Cx interrupt requests and I2CxSR<PIN> after arbitration lost.

	In master mode, detecting the Arbitration Lost when sending the Slave address.	In Master transmitter mode, detecting the Arbitration Lost when sending the Data.
INTI2Cx Interrupt request	When a word transmission is completed, INTI2Cx interrupt is generated.	
I2CxSR<PIN>	Clear I2CxSR<PIN> to "0".	

The INTI2Cx interrupt request is generated, I2CxSR<PIN> is cleared to "0", and the SCL pin is pulled to the "Low" level.

When data is written to or read from I2CxDBR or when I2CxSR<PIN> is set to "1", the SCL pin is released after a period of tLOW.

I2CxSR<AL>,<TRX><AAS> and <AD0> are tested to determine the processing required.

"Table 14-6 Processing in Slave Mode" shows the slave mode states and required processing.

Table 14-6 Processing in Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	State	Processing
1	1	1	0	Arbitration Lost is detected while the slave address was being transmitted and the I2C 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 I2CxDBR.
	0	1	0	In the slave receiver mode, the I2C received a slave address with the direction bit "1" transmitted by the master.	
		0	0	In the slave transmitter mode, the I2C 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 I2CxDBR.
0	1	1	1/0	Arbitration Lost is detected while a slave address is being transmitted, and the I2C receives either a slave address with the direction bit "0" or a general-call address transmitted by another master.	Write the I2CxDBR (a dummy data "0x00" to set I2CxCR2<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.	The serial bus interface circuit is in slave mode. Therefore, clear the I2CxSR<AL> to "0". To set "1" to I2CxSR<PIN>, write dummy data (0x00) to I2CxDBR.
	0	1	1/0	In the slave receiver mode, the I2C received either a slave address with the direction bit "0" or a general-call address transmitted by the master.	Write the I2CxDBR (a dummy data "0x00" to set I2CxCR2<PIN> to 1, or write "1" to <PIN>.
		0	1/0	In the slave receiver mode, the I2C has completed a reception of a data word.	Set the number of bits in the data word to I2CxCR1<BC[2:0]> and read the received data from I2CxDBR, write dummy data (0x00) to I2CxDBR.

Note: In salve mode, if I2CxAR<SA[6:0]> is set to "0x00" a START byte (0x01) of the I2C bus standard is received, a slave address match is detected and I2CxSR<TRX> is set to "1". Not set I2CxAR<SA[6:0]> to "0x00".

14.5.1.4 Generating the Stop Condition

When I2CxSR<BB> is "1", writing "1" to I2CxCR2<MST, TRX, PIN> and "0" to <BB> causes the I2C 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 I2C waits until the SCL line is released.

After that, the SDA pin goes "High", and then causing the stop condition to be generated for a "t_{HIGH}".

The stop condition can be checked using the monitor pin I2CxSR<BB> for the bus

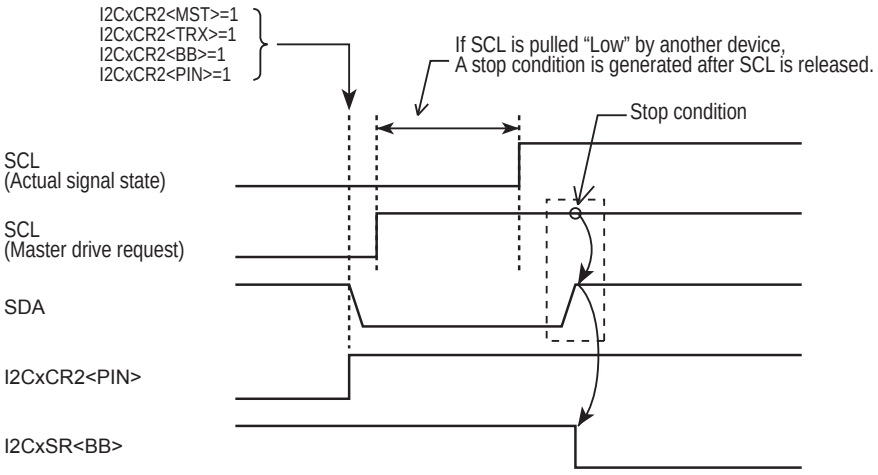


Figure 14-21 Generating the Stop Condition

14.5.1.5 Procedure of Repeated START

Repeated Start is used when a master device changes the data transfer direction without terminating the transfer to a slave device.

Repeated START is an operation that occurs after an ACK/NACK output caused by an INTI2Cx interrupt.

I2CxOPT<SREN> should be set until the clock stretch function is cleared after the occurrence of interrupts.

The procedure of generating a repeated Start in the master mode is described below.

Set I2CxOPT<SREN>=1 when I2CxSR<BB>=1, and write a slave address and direction bit to the data buffer.

After that, write "1" to I2CxCR2<MST>, <TRX>, <PIN>, and <BB> to output a start condition on the I2C bus.

Note: Do not set "0" to I2CxCR2<PIN> when I2CxOPT<SREN>=1

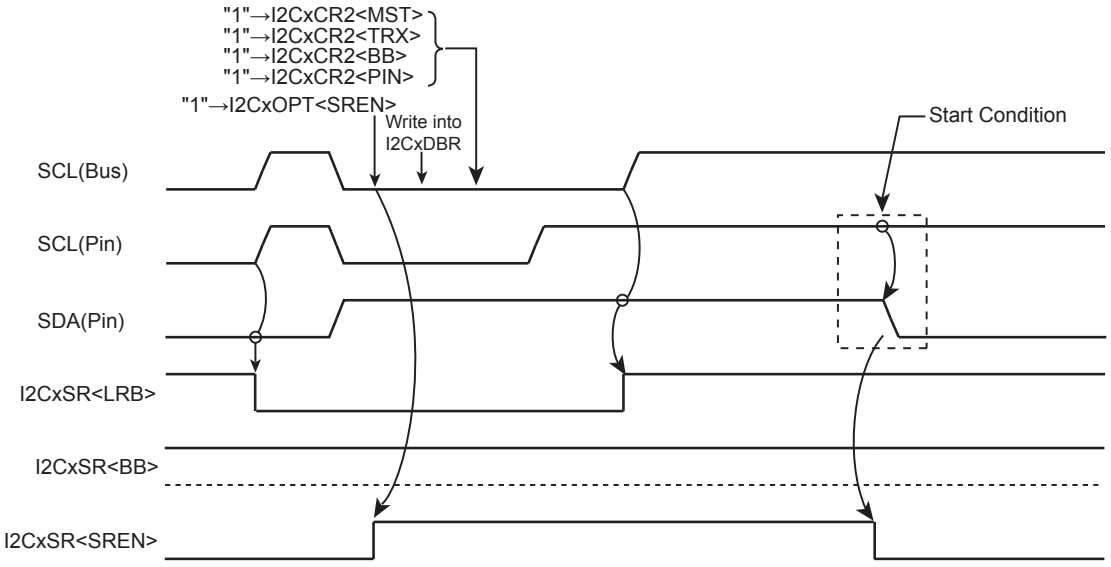


Figure 14-22 Timing Chart of Generating a Repeated Start

14.5.1.6 DMA Transfer using DMA

Data can be transferred using DMA in I2C mode. But one master device and one slave device are connected to a bus line and the number of transfer data is decided beforehand.

In I2C bus mode, when an INTI2Cx interrupt occurs, a DMA request occurs almost simultaneously (after 1 clock); therefore consecutive transfer can be executed between I2CxDBR (data buffer) and the memory using DMA transfer.

A DMA request for transmission and reception are separately executed.

Set enable or disable to the DMAC control register for transmission and reception before using the DMA.

To transfer data using DMA, the number of bytes to be transferred should be preliminarily specified on the both transmission and reception.

If arbitration lost occurs during I2C transfer, the INTI2CALx interrupt and the INTI2Cx interrupt occurs, then the communication stops. A DMA request does not occur.

To open the pins(I2CxSR<PIN>=1) by reading I2CxDBR in receiver mode, set "1" to I2CxIE<SELPINCD>.

The following shows an example of transfer procedure when transferring 8 bytes of data n times with I2C0,DMAC(ch0)

(1) How to transfer data in master mode

1. Generate start condition and slave address.
2. Check <LRB> and <TRX> in interrupt service routine of INTI2C0 after output slave address.
3. When <LRB> is "1", output a stop condition and complete the transferring because no slave Device which has the sent slave address is on the bus.
4. When <LRB> is "0", check <TRX> because slave device which has the sent slave address is on the bus. The proceeding is depended on <TRX>.

(a) When <TRX> is "1" (Transmitter mode)

a. Set DMA such as transfer bit width, burst size and the total transferring number, etc. Subtract 1 from the number of data to be transmitted. And enable to receive DMA request of INTI2C0 with I2C0IE<DMAI2CTX>.
b. Write the first data into I2C0DBR. Start transmit the first data by this writing
c. Start DMA transfer by the DMA request of INTI2C0 which is after the completed transferring. And transmit the 2nd data and following data.
d. When the specified the number of DMA transfer (N-1 times) ends, a DMA transfer end interrupt occurs. Change the settings to ignore a DMA request reception in this interrupt service routine. (Set the I2C0IE<DMARI2CTX> to "0".)
e. Generate stop condition to stop transmission at the interrupt service routine of INTI2C0.

(b) When <TRX> is "0" (Receiver mode)

a. Set DMA such as transfer bit width, burst size and the total transferring number, etc. Subtract 2 from the number of data to be transmitted. And enable to receive DMA request of INTI2C0 by I2C0IE<DMARI2CRX>.
b. Read dummy data from I2C0DBR. Start receive the first data by this reading.
c. Start DMA transfer by the DMA request of INTI2C0 which is after the completed receiving. And receive the 2nd data and following data.
d. When the specified the number of DMA transfer (N-2 times) ends, a DMA transfer end interrupt occurs. Change the settings to ignore a DMA request reception in this interrupt service routine. (Set the I2C0IE<DMARI2CRX> to "0").
e. Set "1" to <MFACK> to generate a NACK signal by an INTI2C0 interrupt that occurs at (n-1) data reception, and read then receive data from I2C0DBR.
f. Generate stop condition to stop transmission at the interrupt service routine of INTI2C0.

(2) How to transfer data in Slave mode

1. Receive start condition and slave address.
2. Check I2C0SR<TRX> in interrupt service routine of INTI2C0 after receive slave address
3. Check <TRX>. The proceeding is depended on <TRX>.

(a) When <TRX> is "1" (Transmitter mode)

a. Set DMA such as transfer bit width, burst size and the total transferring number, etc. Subtract 1 from the number of data to be transmitted. And enable to receive DMA request of INTI2C0 By I2C0IE<DMARI2CTX>.
b. Write the first data into I2C0DBR. This device can be received a clock from a master device and start transmit the first data by this writing.
c. Start DMA transfer by the DMA request of INTI2C0 which is after the completed transferring. And transmit the 2nd data and following data.
d. When the specified the number of DMA transfer (N-1 times) ends, a DMA transfer end interrupt occurs. Change the settings to ignore a DMA request reception in this interrupt service routine.(set I2C0IE<DMAI2CTX> to "0")
e. Wait stop condition from a master device without writing a data at the service routine of INTI2C0.

(b) When <TRX> is "0" (Receiver mode)

a. Set DMA such as transfer bit width, burst size and the total transferring number, etc. Subtract 2 from the number of data to be transmitted. And enable to receive DMA request of INTI2C0 By I2C0IE<DMARI2CRX>.
b. Read dummy data from I2C0DBR. Start receive the first data by this reading.
c. Start DMA transfer by the DMA request of INTI2C0 which is after the completed receiving. And receive the 2nd data and following data.
d. When the specified the number of DMA transfer (N-2 times) ends, a DMA transfer end interrupt occurs. Change the settings to ignore a DMA request reception in this interrupt service routine. (Set I2C0IE<DMARI2CRX to "0")
e. Set "1" to <MFACK> to generate a NACK signal by an INTI2C0 interrupt that occurs at (n-1) data reception, and read then receive data from I2C0DBR.
f. Wait stop condition from a master device without writing a data at the service routine of INTI2C0.

14.6 I2C Address Match Wake-Up Function

When the I2C block is used in slave mode, if the frame address on the I2C bus matches a self-address (slave address) in low-power consumption mode (STOP1), an interrupt (INTI2CS) occurs to clear low-power consumption mode.

14.6.1 Configuration

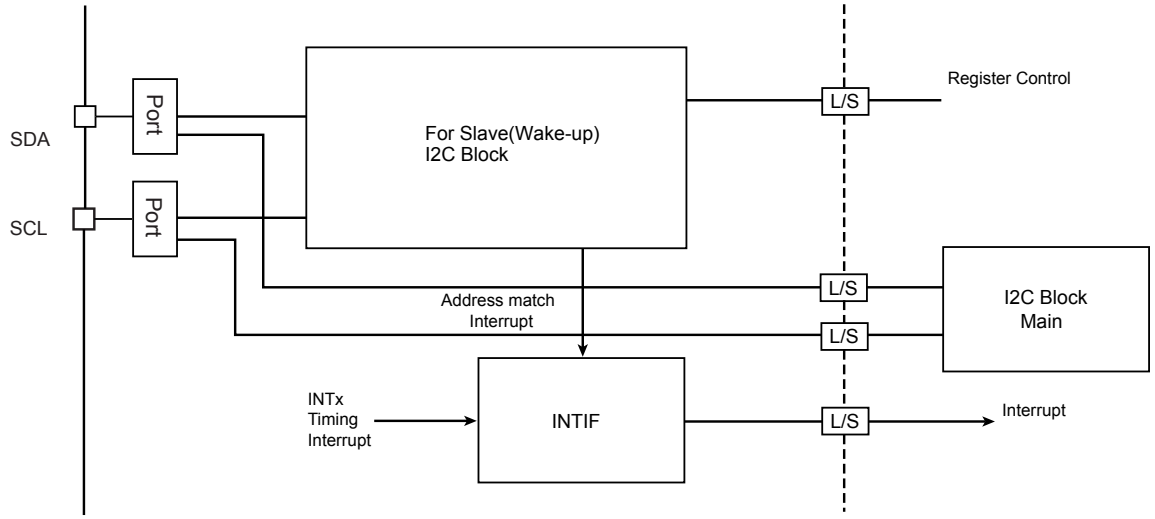


Figure 14-23 Wake-up Function Block

14.6.2 Description of Operation

14.6.2.1 Clock Stretch Function

After the MCU confirms the match of slave addresses in slave mode, it returns an ACK and then performs the clock stretch operation that pulls the SCL to "Low".

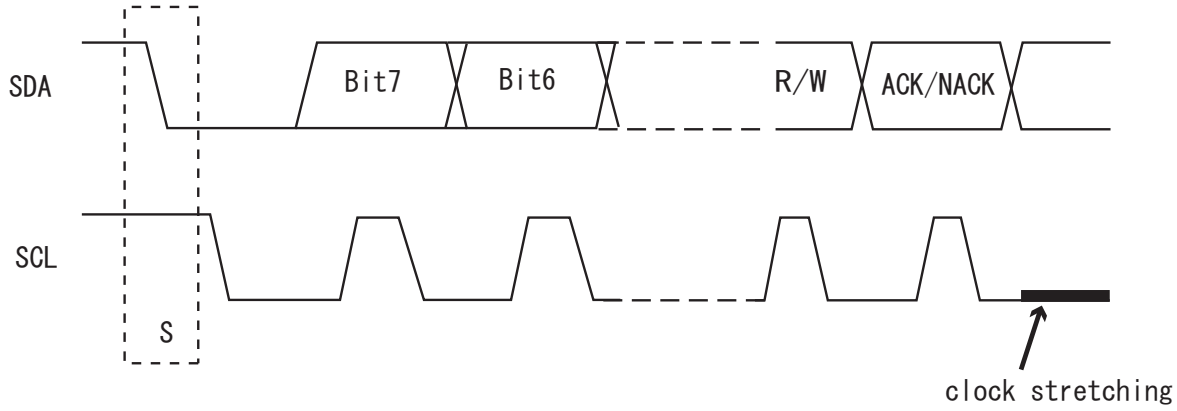


Figure 14-24 Clock stretch Function

Note: If the MCU enters low-power consumption mode during the clock stretch operation, the I2C bus maintains locked state, so that the address match wake-up function cannot be used.

14.6.2.2 Operation flow of the Address Match Wake-up Function

Figure 14-25 shows the operation flow described below:

1. When the MCU detects an address match, the MCU returns an ACK to generate INTI2CS.
2. Low-power consumption mode is released, and the MCU starts the clock stretch function.
3. In interrupt service routine after low-power consumption mode is released, the interrupt request is cleared with I2CSWUCR1<INTEND>.
4. The clock stretch function is cleared with I2CSWUPCR1<I2RES>.

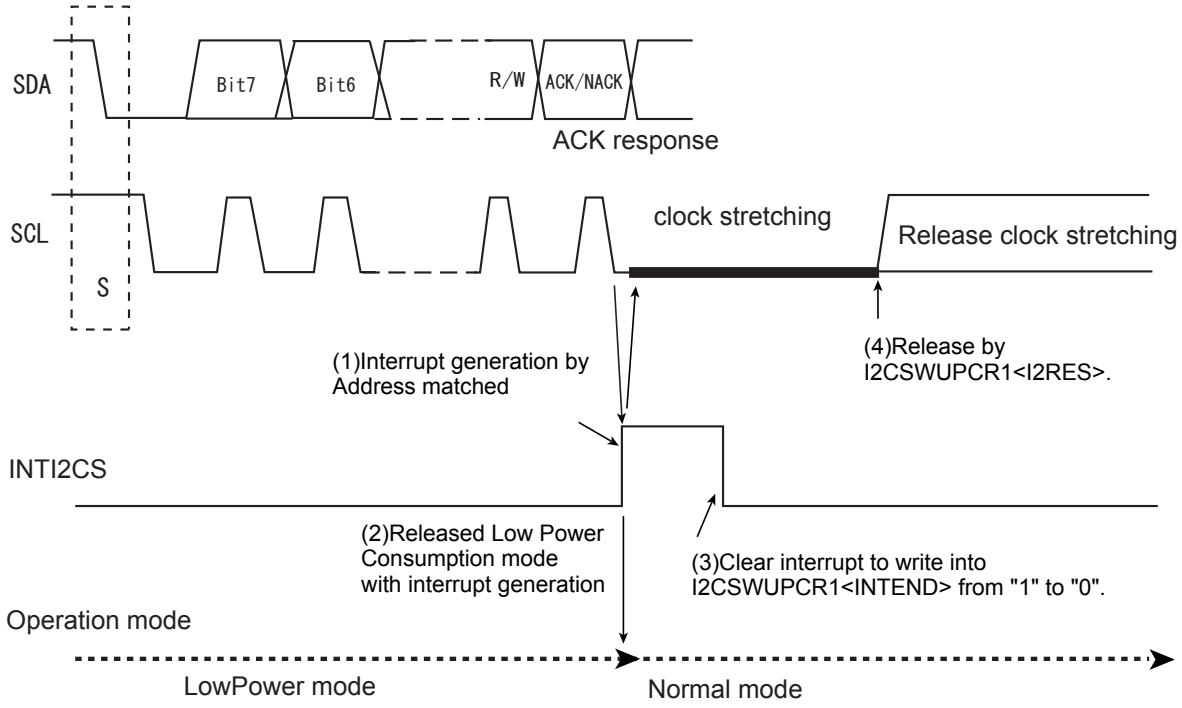


Figure 14-25 Address match Wake-up Function

Note 1: When the MCU performs the address match wake-up function, the I2C bus is locked until the MCU returns to the normal operation.

Note 2: When I2CSWUPCR1<INTEND> is changed to "0" from "1" in interrupt service routine, the I2C information is transmitted to the main block from the wake-up block.

Note that the detection conditions for slave addresses and general call must be the same when the address match wake-up function is used.

- I2CxAR<SA[6:0]>=I2CSWUPCR2<WUPSA1[6:0]>; same address, detection condition
- I2CxAR2<SA2[6:0]>=I2CSWUPCR3<WUPSA2[6:0]>; same address, detection condition
- Setting contents of I2CxCR1<ACK>, <NOACK> & I2CxOP<MFACK> is as same as I2CSWUPCR1<ACK>, <SGCDI>.

14.6.2.3 Timing

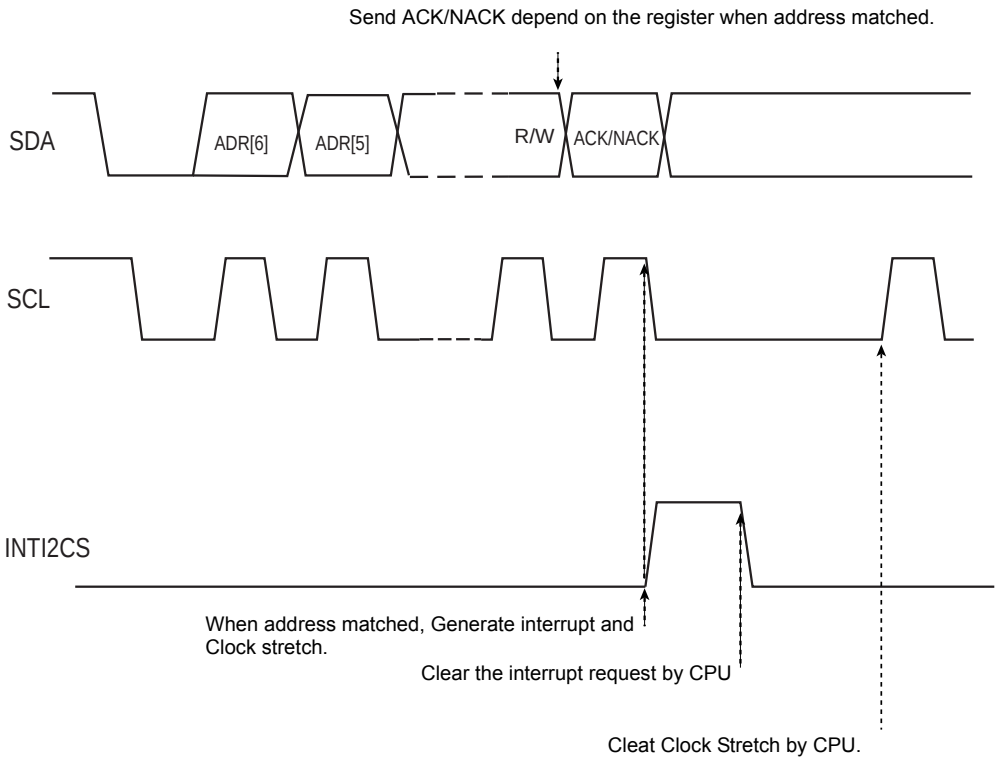
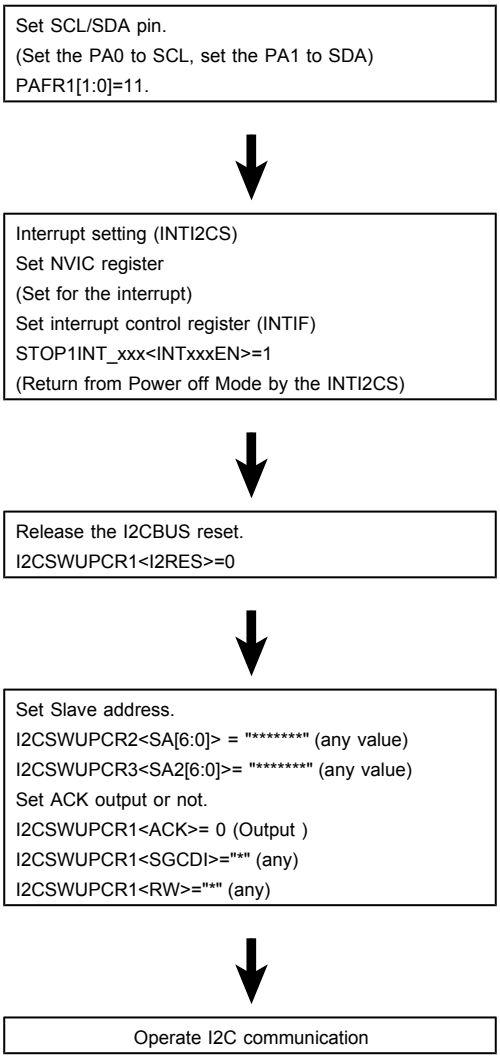


Figure 14-26 Address match Wake-up Function

Note: The master only sends an ACK signal. The device selects an ACK signal or NACK signal and then sends it.

14.6.2.4 Operation and setting flow (example)

<Initial setting> Wake-up block side setting to enter STOP1 mode.



Note: The register can be written during the reset operation caused by I2CSWPCR1<I2RES>=1 even when reset operation continues

The setting for the I2C main block should be completed before the MCU enters low-power consumption mode (STOP1). Communications after the address match wake-up function does not work properly unless the following registers are set.

I2CxCR1<BC[2:0]>,<ACK>,<NOACK>,<SCK[2:0]>

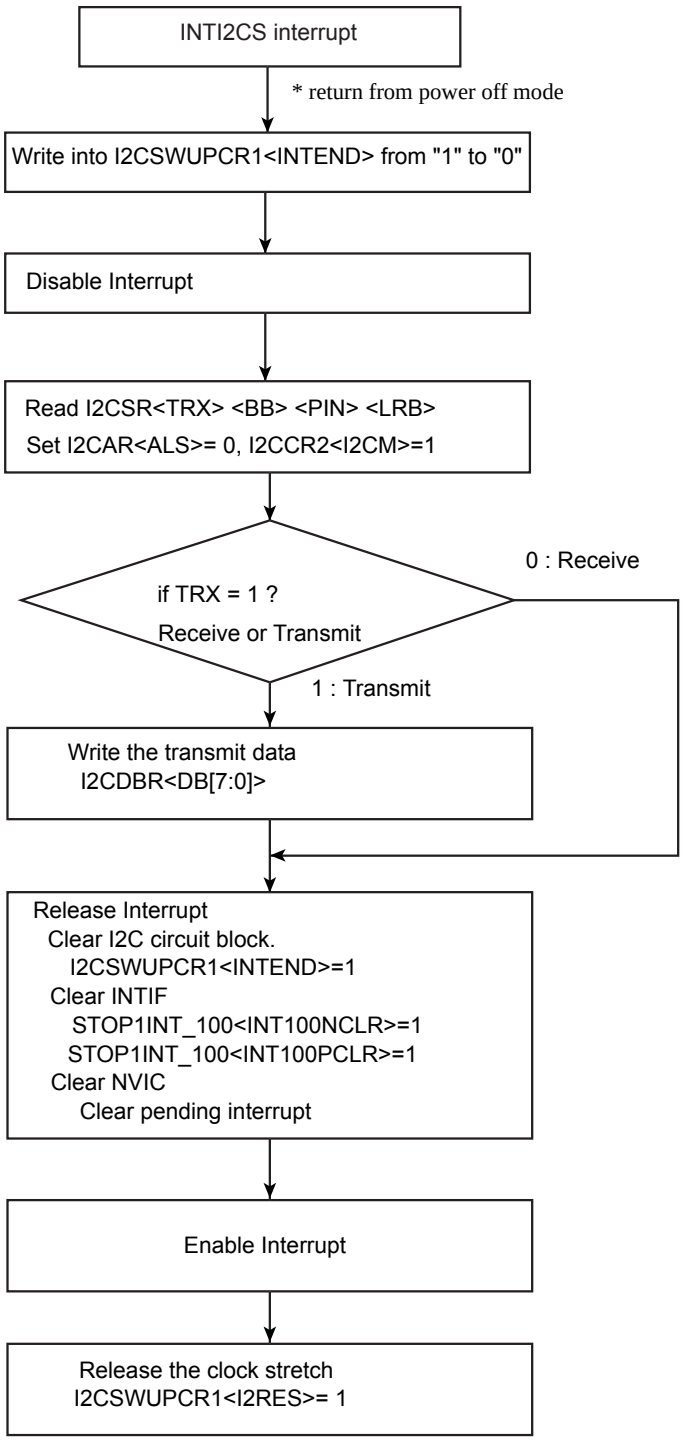
I2CxPRS<PRSCK[4:0]>

I2CxOP<NFSEL>,<MFACK>

I2CxAR2<SA2[6:0]>,<SA2EN> ; any value

I2CxIE ; set usage function

<When Generating Interrupt>



Notice of clearing the interrupt

Make sure to clear the circuit along the following order: I2C circuit, INTIF, and then NVIC. If the order is not correct, interrupt factors remain; therefore, the interrupts occur again.

15. Toshiba Serial Peripheral Interface (TSPI)

15.1 Outline

TSPI is a serial interface that can communicate in synchronous full-duplex communication. It has SPI and SIO modes, so that it can perform high-speed serial transfer between various peripheral devices.

TSPI has chip select signal pins (TSPIxCSO and TSPIxCSIN), a serial clock signal pin (TSPIxSCK) and serial transmit and receive data signal pins (TSPIxTXD and TSPIxRXD).

Data length can be arranged by 1 bit from 7 bits (with parity) to 32 bits (without parity).

Table 15-1 Outline

Parameter	SPI mode (Master)	SPI mode (Slave)	SIO mode (Master)	SIO mode (Slave)
Communication mode	Transmit and receive/transmit/receive			
Transfer mode	Single/burst (2 to 255 frames)			
Serial clock maximum frequency	fc/2	fc/2	fc/2	fc/2
TSPIxSCK	Output	Input	Output	Input
TSPIxCSO	Output	-	-	-
TSPIxCSIN	-	Input	Unused	Unused
Output level setting of TSPIxTXD at idle	High, Low, last bit data holding, Hi-z			
Output level setting of TSPIxTXD when underrun error occurs	-	High, Low	-	High, Low
TSPIxCSO polarity selection	Negative logic/positive logic	-	-	-
TSPIxCSIN polarity selection	-	Negative logic/positive logic	-	-
TSPIxSCK polarity selection	"L" at idle/"H" at idle			
Transfer order	MSB first/LSB first			
Data length	Settable by 1 bit 8 to 32 bits (no parity) 7 to 31 bits (parity exists)			
TSPIxCS valid→ TSPIxSCK valid time	Settable		-	-
Transfer completion→ TSPIxCS invalid time	Settable		-	-
Interval time between frames	Settable			
TSPIxCS invalid→ TSPIxCS valid time	Settable			
Adds parity	With/without			
Transmit FIFO empty detection	Exists			
Receive FIFO full detection	Exists			
Parity error detection	Exists			
Overrun error detection Underrun error detection	-	Exists	-	Exists
DMA request	Reached to fill level of receive FIFO Reached to fill level of transmit FIFO			

15.2 Block Diagram

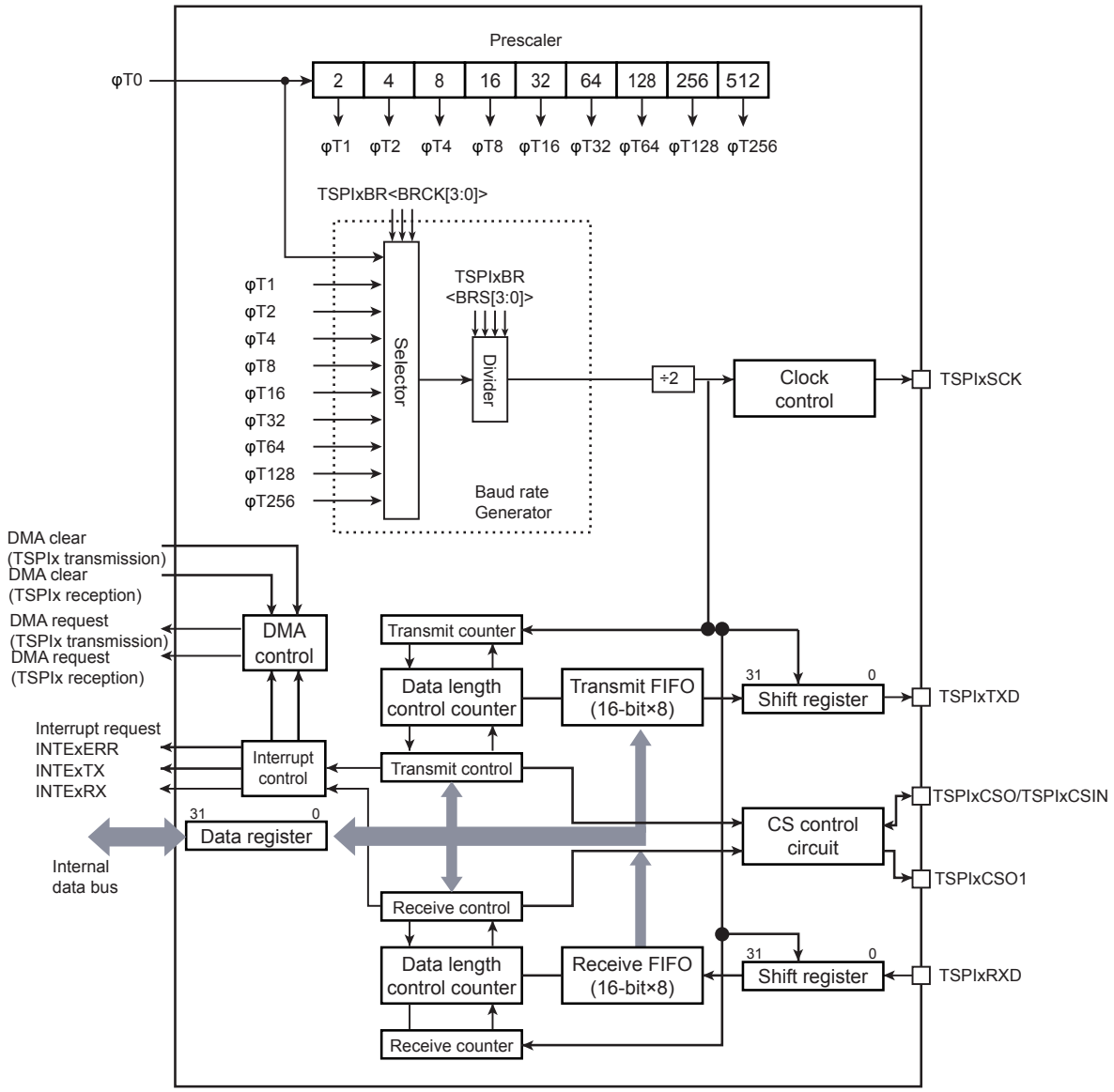


Figure 15-1 Block diagram of TSPIx

15.3 Registers

15.3.1 Register List

The following table lists the control registers and addresses.

For details of base address, refer to "Peripheral function base address map" in Chapter "Memory Map".

Register name (x= Channel number)		Address (Base+)
TSPI control register 0	TSPIxCR0	0x0000
TSPI control register 1	TSPIxCR1	0x0004
TSPI control register 2	TSPIxCR2	0x0008
TSPI control register 3	TSPIxCR3	0x000C
TSPI baud rate register	TSPIxBR	0x0010
TSPI format control register 0	TSPIxFMTR0	0x0014
TSPI format control register 1	TSPIxFMTR1	0x0018
Reserved	−	0x001C to 0x00FF
TSPI data register	TSPIxDR	0x0100
Reserved	−	0x0104 to 0x01FF
TSPI status register	TSPIxSR	0x0200
TSPI error flag register	TSPIxERR	0x0204
Reserved	−	0x0208 to 0x0FFF

Note 1: Do not access the register which is described as "Reserved" in the above table.

Note 2: Registers except TSPIxCR0<SWRST>, TSPIxCR1<TRXE>, TSPIxDR and TSPIxSR cannot be set when TSPIxSR<TSPISUE> is "1".

15.3.2 TSPIxCR0 (TSPI 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	SWRST		-	-	-	-	-	TSPIE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-6	SWRST[1:0]	W	TSPI software reset (Note) Software reset occurs by writing "10" → "01". If "10" → "01" is consecutively written to <SWRST[1:0]> bit, software reset occurs. Software reset initializes control registers, transmit/receive FIFO, transmit/receive shift registers and all internal circuits. However, <TSPIE> bit is not initialized.
5-1	−	R	Read as "0".
0	TSPIE	R/W	TSPI operation control 0: Stop 1: Operation <TSPIE> controls a whole operation of TSPI to start/stop (clock shutdown). When <TSPIE>=0(stop) is set, a clock is not fed into TSPI internally. Set <TSPIE>=1(operation) to start operation first. Then perform initialization and communications. <TSPIE> is not initialized by software reset.

Note: Completion of software reset takes two clocks after an instruction is executed.

15.3.3 TSPIxCR1 (TSPI 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	-	TRXE	TSPIMS	MSTR	TMMD		-	CSSEL
After reset	0	0	0	1	1	1	0	0
	7	6	5	4	3	2	1	0
bit symbol	FC							
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-15	−	R	Read as "0".
14	TRXE	R/W	Communication control (Note1)(Note2)(Note3) 0 : Communication stops 1 : Communication is enabled. Full duplex mode/transmission mode If valid data exists in the transmit FIFO or shift register, transmission starts. If valid data does not exist in the transmit FIFO or shift register, transmission does not start. To start communications, write data to transmit FIFO or write transmit data when communications are enabled. If this bit is set as disable during transmission, the transmission will stop after the ongoing frame will complete and the setting will become disable. Receive mode : Once this bit is enabled, reception immediately starts. If this bit is set to disable during reception, reception will stop after the ongoing frame is complete and the setting is becomes disable.
13	TSPIMS	R/W	Communication mode selection 0 : SPI mode 1 : SIO mode
12	MSTR	R/W	Master/slave selection 0 : Slave operation 1 : Master operation
11-10	TMMD[1:0]	R/W	Transfer mode selection 00 : Prohibited 01 : Transmit only 10 : Receive only 11 : Full-duplex mode (Transmit/receive) If only the transmit setting is done, a process circuit for TSPIxRXD stops. If only the reception setting is done, a process circuit for TSPIxTXD stops.
9	−	R	Read as "0".
8	CSSEL	R/W	Selection of TSPIxCSO 0 : TSPIxCSO is valid. 1 : Reserved.
7-0	FC[7:0]	R/W	Sets number of transfer frames 0 : Prohibited 1 : Single transfer 2 to 255 : Burst transfer

Note 1: <TRXE> is not cleared to "0" as long as CPU write "0" in single transfer. However, in the case of burst transfer, <TRXE> is automatically cleared to "0" after the specified numbers of burst transfers are complete. If burst transfer is executed again, check if TSPIxSR<TSPISUE> bit returns to "0" then write "1" to <TRXE>.

Note 2: In the slave operation, TSPI modify status flag TSPIxSR<TSPISUE> is not cleared to Åg0Åh, even if writes <TRXE> to "0" (disable communication) before actual communication (on master side) is not started after <TRXE> wrote to "1" (enable communication). When in spite of enabled communication on slave device, the communication on the master device is not started, please performs software reset by TSPIxCR0<SWRST> in a slave device, and re-set it up.

Note 3: In the slave operation, if <TRXE> is rewritten to "0" (communication prohibited) when data remains in the transmission buffer (FIFO) during transmission, transmission buffer (FIFO) should be cleared by TSPIxCR3<TFEMPCLR>, or should be performed software reset by TSPIxCR0<SWRST>, then re-set it up.

15.3.4 TSPIxCR2 (TSPI 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	TIDLE		TXDEMP	-	-	-	-	-
After reset	1	1	1	0	0	0	0	1
	15	14	13	12	11	10	9	8
bit symbol	TIL				RIL			
After reset	0	0	0	0	0	0	0	1
	7	6	5	4	3	2	1	0
bit symbol	INTTXFE	INTTXWE	INTRXFE	INTRXWE	-	INTERR	DMATE	DMARE
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-24	−	R	Read as "0".
23-22	TIDLE1:0]	R/W	Fixed output value function control when TSPIxTXD idles. 00 : Hi-z 01 : Last data in previous transfer 10 : Fixed to low 11 : Fixed to high
21	TXDEMP	R/W	Fixed output value function control when TSPIxTXD Andorrans. 0 : Fixed to low 1 : Fixed to high
20-17	−	R	Read as "0".
16	−	R/W	Write as "0".
15-12	TIL[3:0]	R/W	Transmit Fill level setting Transmit FIFO interrupt occurrence condition (Note)
11-8	RIL[3:0]	R/W	Receive Fill level setting Receive FIFO interrupt occurrence condition (Note)
7	INTTXFE	R/W	Transmit FIFO interrupt control 0 : Disabled 1 : Enabled
6	INTTXWE	R/W	Transmit completion interrupt control 0 : Disabled 1 : Enabled
5	INTRXFE	R/W	Receive FIFO interrupt control 0 : Disabled 1 : Enabled
4	INTRXWE	R/W	Receive completion interrupt control 0 : Disabled 1 : Enabled
3	−	R	Read as "0".
2	INTERR	R/W	Error interrupt control 0 : Disabled 1 : Enabled
1	DMATE	R/W	Transmit DMA control 0 : Disabled 1 : Enabled
0	DMARE	R/W	Receive DMA control 0 : Disabled 1 : Enabled

Note: Set the fill level within available values shown in "Table 15-5 Data format and settable fill level".

15.3.5 TSPIxCR3 (TSPI Control Register 3)

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

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	TFEMPCLR	W	Clears transmit buffer 0 : Invalid 1 : Clear By writing "1" to <TFEMPCLR>, the internal pointer of transmit FIFO and pointer of transmit shift register are initialized. Since contents of transmit FIFO and transmit shift register are not affected on the initialization, data remains the previous condition in which transmit buffer is cleared.
0	RFFLLCLR	W	Clears receive buffer 0 : Invalid 1 : Clear By writing "1" to <RFFLLCLR>, the internal pointer of receive FIFO is empty and the internal pointer of receive shift register is initialized. Since contents of transmit FIFO and transmit shift register are not affected on the initialization, data remains the previous condition in which transmit buffer is cleared.

15.3.6 TSPIxBR (TSPI 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	BRCK				BRS			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	BRCK[3:0]	R/W	Input clock selection for baud rate generator 0000 : φT0 0101 : φT16 0001 : φT1 0110 : φT32 0010 : φT2 0111 : φT64 0011 : φT4 1000 : φT128 0100 : φT8 1001 : φT256 1010 to 1111: Prohibited
3-0	BRS[3:0]	R/W	Sets a division ratio of baud rate generator "N". 0000 :16 0110 : 6 1100 : 12 0001 : 1 0111 : 7 1101 : 13 0010 : 2 1000 : 8 1110 : 14 0011 : 3 1001 : 9 1111 : 15 0100 : 4 1010 :10 0101 : 5 1011 :11

Table 15-2 Baud rate setting (sample)

Setting value of BRCK[3:0]	Setting value of BRS[3:0]	Division ratio	Serial clock frequency (MHz @ fc=24MHz)
0000	0000	32	0.75
0000	0001	2	12
0000	0010	4	6
0000	0011	6	4
0000	0100	8	3
0000	0101	10	2.4

15.3.7 TSPIxFMTR0 (TSPI Format Control Register 0)

	31	30	29	28	27	26	25	24
bit symbol	DIR	-	FL					
After reset	1	0	0	0	1	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	FINT				-	-	FW	
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	CKPOL	CSINT				-	CS0POL
After reset	1	1	0	0	0	1	0	0
	7	6	5	4	3	2	1	0
bit symbol	CSSCKDL				SCKCSDL			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	DIR	R/W	Transfer direction 0 : LSB first 1 : MSB first
30	−	R	Read as "0".
29-24	FL[5:0]	R/W	Sets a frame length. Sets a data length of one frame including a parity bit. 00_1000 : 8 bits 00_1001 : 9 bits : 01_1111 : 31 bits 10_0000 : 32 bits Other than the above is prohibited.
23-20	FINT[3:0]	R/W	Interval time between frames in the burst transfer. 0000: 0 0001 : 1 x TSPIxSCK cycles 0010 : 2 x TSPIxSCK cycles : 0111 : 14 x TSPIxSCK cycles 1111 : 15 x TSPIxSCK cycles
19-15	−	R	Read as "0".
14	CKPOL	R/W	Serial clock polarity 0 : TSPIxSCK level is "L" at idle. 1 : TSPIxSCK level is "H" at idle.
13-10	CSINT[3:0]	R/W	TSPIxCSO invalid→TSPIxCSO valid time (Minimum idle time) 0000: Prohibited 0001 : 1 x TSPIxSCK cycles 0010 : 2 x TSPIxSCK cycles : 0111 : 14 x TSPIxSCK cycles 1111 : 15 x TSPIxSCK cycles
9	-	R/W	Reserved.
8	CS0POL	R/W	Polarity of TSPIxCSO 0: Negative logic 1: Positive logic
7-4	CSSCKDL[3:0]	R/W	TSPIxCSO valid→TSPIxSCK valid time (Serial clock delay) Valid only in master mode.

Bit	Bit Symbol	Type	Function
			0000 : 1 x TSPIxSCK 0110 : 7 x TSPIxSCK 1100 : 13 x TSPIxSCK 0001 : 2 x TSPIxSCK 0111 : 8 x TSPxISCK 1101 : 14 x TSPIxSCK 0010 : 3 x TSPxSCK 1000 : 9 x TSPIxSCK 1110 : 15 x TSPIxSCK 0011 : 4 x TSPIxSCK 1001 : 10 x TSPIxSCK 1111 : 16 x TSPIxSCK 0100 : 5 x TSPIxSCK 1010 : 11 x TSPIxSCK 0101 : 6 x TSPIxSCK 1011 : 12 x TSPIxSCK
3-0	SCKCSDL[3:0]	R/W	Transfer is complete→TSPIxCSO invalid time (TSPIxCSO negate delay) Valid only in master mode. 0000 : 1 x TSPIxSCK 0110 : 7 x TSPIxSCK 1100 : 13 x TSPIxSCK 0001 : 2 x TSPIxSCK 0111 : 8 x TSPIxSCK 1101 : 14 x TSPIxSCK 0010 : 3 x TSPIxSCK 1000 : 9 x TSPIxSCK 1110 : 15 x TSPIxSCK 0011 : 4 x TSPIxSCK 1001 : 10 x TSPIxSCK 1111 : 16 x TSPIxSCK 0100 : 5 x TSPIxSCK 1010 : 11 x TSPIxSCK 0101 : 6 x TSPIxSCK 1011 : 12 x TSPIxSCK

Note: Whenever <TSPISUE> is "0", all data in FIFO are discarded if <FL[5:0]> is changed remaining data in FIFO.

15.3.8 TSPIxFMTR1 (TSPI Format 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	-	EHOLD			-	-	VPE	VPM
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-4	EHOLD[2:0]	R/W	Sets a last bit holding time of TSPIxTXD pin in SIO slave mode. 000: 2/fc 001: 4/fc 010: 8/fc 011: 16/fc 100: 32/fc 101: 64/fc 110: 128/fc 111: Reserved
3-2	−	R	Read as "0".
1	VPE	R/W	Parity function 0 : Disabled 1 : Enabled
0	VPM	R/W	Parity mode selection 1 : Odd parity 0 : Even parity

15.3.9 TSPIxDR (TSPI Data Register)

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

Bit	Bit Symbol	Type	Function
31-0	TSPIDR[31:0]	R/W	If this register is written, data is written to transmit FIFO; if this register is read, data is read from receive FIFO. [Write] : Access to transmit FIFO [Read]: Access to receive FIFO

Note 1: Do not write data to this register when transmit FIFO is full.

Note 2: Do not read data from this register when receive FIFO is empty.

15.3.10 TSPIxSR (TSPI Status Register)

	31	30	29	28	27	26	25	24
bit symbol	TSPISUE	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TXRUN	TXEND	INTTXWF	TFEMP	TLVL			
After reset	0	0	0	1	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RXRUN	RXEND	INTRXFF	RFFLL	RLVL			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	TSPISUE	R	TSPI modify status flag 0 : Setting is enabled to modify. 1 : Setting is disabled to modify. If <TSPISUE> is "0", TSPI is not transmitting or receiving, thus the register setting can be modified. <TSPISUE> is "0" in the following conditions: 1.Reset is input. 2.Software reset occurs. 3.In the single transfer mode, the time when current transferring frame is finished when TSPIxCR1<TRXE>=0 is set. 4.In the burst mode, the timing when specified number of transfers is finished. 5.The time when current transferring frame is finished when TSPIxCR1<TRXE>=0 is set during burst transfer. However, even if above conditions are satisfied, <TSPISUE> does not become "0" if transmit FIFO or receive shift register is full. To set <TSPISUE>=0, read receive FIFO and transfer a receive value in the receive shift register to receive FIFO.
30-24	−	R	Read as "0".
23	TXRUN	R	Transmit shift operation flag 0 : Stop 1 : Operation A status flag indicates the transmit shift operation is ongoing. Combination of <TXRUN> and <TFEMP> bits indicates the following status. <TXRUN> is set when data exists in the transmit shift register even if data does not exist in the transmit FIFO.
22	TXEND	R/W	Transmit completion flag [Read] 0: - 1 : Transmit is complete. [Write] 1 : Flag is cleared. A flag that is set at the time when transmission is complete. This flag is set at the last frame (TSPIxCSO is negated) in the single transfer or burst transfer. By writing "1" to this bit, it is cleared. If setting by transmission completion and clearing by writing "1" occur simultaneously, setting receives a higher priority.

<TXRUN>	<TFEMP>	Conditions
0	0	Stop or wait for next transmission
	1	Completed transmission and transmit FIFO is empty.
1	-	Under transmission

Bit	Bit Symbol	Type	Function
21	INTTXWF	R/W	Transmit FIFO interrupt flag [Read] 0: No interrupt 1: Interrupt occurs [Write] 0; Don't care 1 : Flag is cleared. This bit is set when remaining data in the transmit FIFO reaches to a TIL value from a fill level setting value (TIL)+1. This bit is cleared by writing "1".
20	TFEMP	R	Transmit FIFO empty flag 0 : Data exists in FIFO. 1 : Empty When transmit FIFO is empty, "1" is set. If transmit data is written to transmit FIFO, this bit is automatically cleared to "0".
19-16	TLVL[3:0]	R	Transmit FIFO fill level status Indicates the current value of transmit FIFO fill level (number of data). Stages of FIFO varies depending on the length of frame. Table 15-5 shows the display range.
15-8	−	R	Read as "0".
7	RXRUN	R	Receive operation flag 0 : Stop 1 : Operation A status flag indicates the receive shift operation is ongoing. Combination of <RXRUN> and <RFFLL> indicate the following status <RXRUN> / <RFFLL> / Conditions 0 / 0 / Stop or wait for next reception 0 / 1 / Receive FIFO is FULL and reception is complete. 1 / - / Receiving is ongoing.
6	RXEND	R/W	Receive completion flag [Read] 0: - 1 : Receiving is complete. A flag that is set at the time when reception is complete. This flag is set at the last frame (TSPIxCSO is negated) in the single transfer and burst transfer. [Write] 0 : Don't care 1 : Flag is cleared. This bit is cleared by writing "1". If setting by reception completion and clearing by writing "1" occur simultaneously, setting receives a higher priority.
5	INTRXFF	R/W	Receive FIFO interrupt flag [Read] 0: No interrupt 1: Interrupt occurs [Write] 0: Don't care 1 : Flag is cleared. This bit is set when remaining data in the receive FIFO reaches to a RIL value from a fill level setting value (RIL)-1. This bit is cleared by writing "1".
4	RFFLL	R	Receive FIFO full flag 0 : A space exists in FIFO 1 : Full Indicates receive FIFO is full. This bit is automatically cleared if data is read from receive FIFO.
3-0	RLVL[3:0]	R	Receive FIFO fill level status Indicates the current value of receive FIFO fill level (number of data). Stages of FIFO varies depending on the length of frame. Table 15-3 shows the display range on <TLVL>.

Table 15-3 Current value of Fill level depending on the range of <TLVL>/<RLVL>

Frame length	FIFO configuration		
	FIFO stages	Range of <RLVL> when receiving	Range of <TLVL> when transmitting
8 to 16bit	8 stages	0 to 8	0 to 8
17 to 32bit	4 stages	0 to 4	0 to 4

15.3.11 TSPIxERR (TSPI Error 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	-	-	-	-	UDRERR	OVRERR	PERR	-
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
31-12	−	R	Read as "0".
11	UDRERR	R/W	Underrun error flag [Read] 0 : No error 1 : Error exists [Write] 0: Don't care 1 : Flag is cleared. This bit is set when an underrun error occurs. This bit is cleared by writing "1". Do not clear it during transmission/reception.
10	OVRERR	R/W	Overrun error flag [Read] 0 : No error 1 : Error exists [Write] 0: Don't care 1 : Flag is cleared. This bit is set when an overrun error occurs. This bit is cleared by writing "1". Do not clear it during transmission/reception.
9	PERR	R/W	Parity error flag [Read] 0 : No error 1 : Error exists [Write] 0: Don't care 1 : Flag is cleared. This bit is set when an parity error occurs. This bit is cleared by writing "1". Do not clear it during transmission/reception.
8-0	−	R	Read as "0".

15.4 Operation Description

15.4.1 Data Format

If TSPIxFMTR0 is set, transfer directions (MSB/LSB first), a frame length and existence of parity are determined.

Note: When the parity function is valid, data length is 31 bits at maximum.

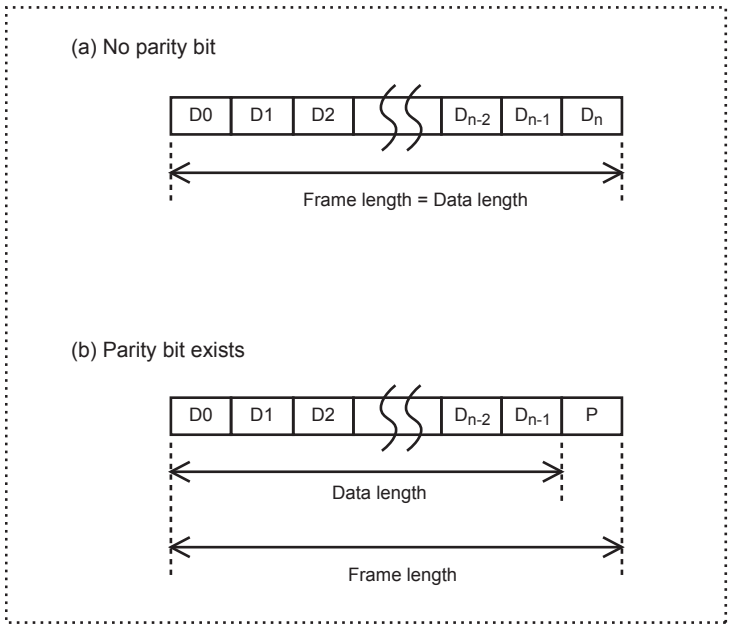


Figure 15-2 Data format schematic

15.4.1.1 Data Format without Parity

If a parity bit is not added, the length of frame must be the same as the length of data.

For example, if the length of data is 10-bit, set "10" to TSPIxFMTR0<FL[5:0]>.

If a parity bit is not added, data in the transmit FIFO is transferred remaining unchanged to the shift register.

(1) MSB First Transfer (32-bit data, without parity, 32-bit frame length)

Figure 15-3 shows a transmit/receive operation (without parity, MSB first, 32-bit data length).

In the transmission, data in the transmit FIFO is copied to D31 through D0 in the shift register. Transmit data copied to shift register is transferred sequentially from D31 through D0 on serial clock.

In the reception, receive data is stored in the D0 of the shift register. Shift operation repeats on serial clock. If the shift register stores 32-bit reception data, data is copied to receive FIFO.

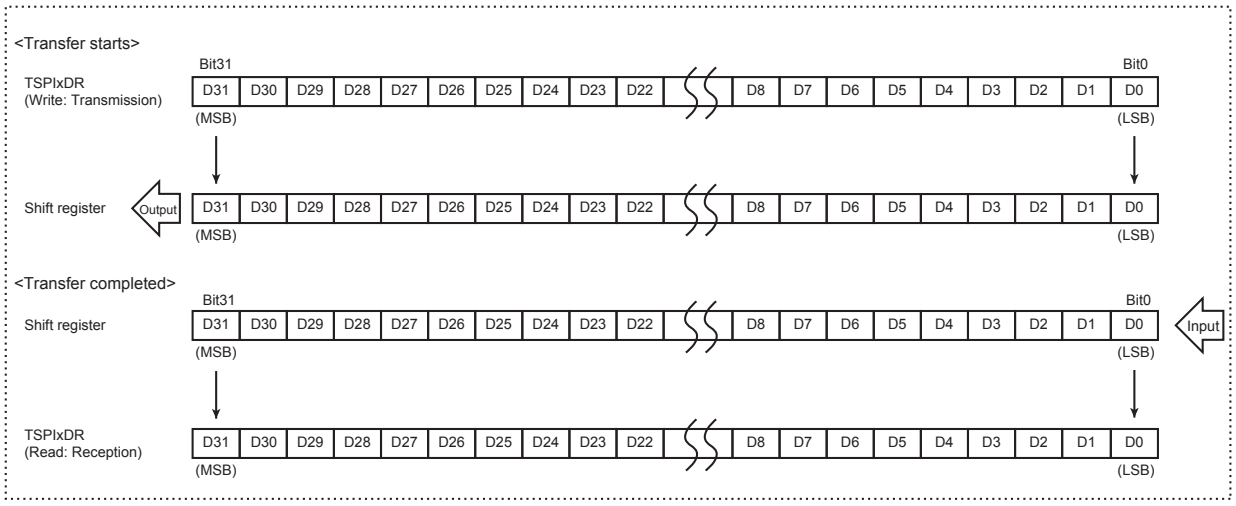


Figure 15-3 MSB first (32-bit data without a parity bit)

(2) MSB First Transfer (16-bit data without a parity bit, 16-bit data frame length)

Figure 15-4 shows a transmit/receive operation (without parity, MSB first, 32-bit data length).

In the transmission, data in the transmit FIFO is copied to D15 through D0 in the shift register. Transmit data copied to shift register is transferred sequentially from D15 through D0 on serial clock.

In the reception, receive data is stored in the D0 of the shift register. Shift operation repeats on serial clock. If the shift register stores 16-bit reception data, data is copied to receive FIFO.

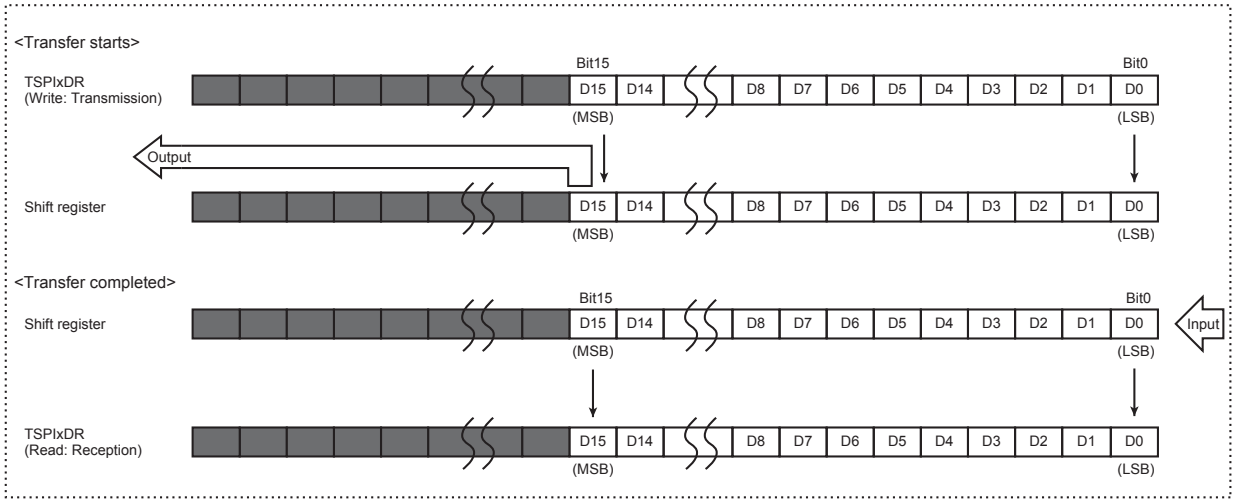


Figure 15-4 MSB first (16-bit data without a parity bit)

(3) LSB First Transfer (32-bit data without a parity bit, 32-bit frame length)

Figure 15-5 shows a 32-bit data length transmit/receive operation when parity function is disabled.

In the transmission, data in the transmit FIFO is sorted bit by bit when the data is copied to the shift register. Transmit data copied to shift register is transferred from D0 until reaching 32-bit shifted data on serial clock.

In the reception, receive data is stored in the D0 of the shift register. Shift operation repeats on serial clock. If the shift register stores 32-bit reception data, data is sorted bit by bit and copied to receive FIFO.

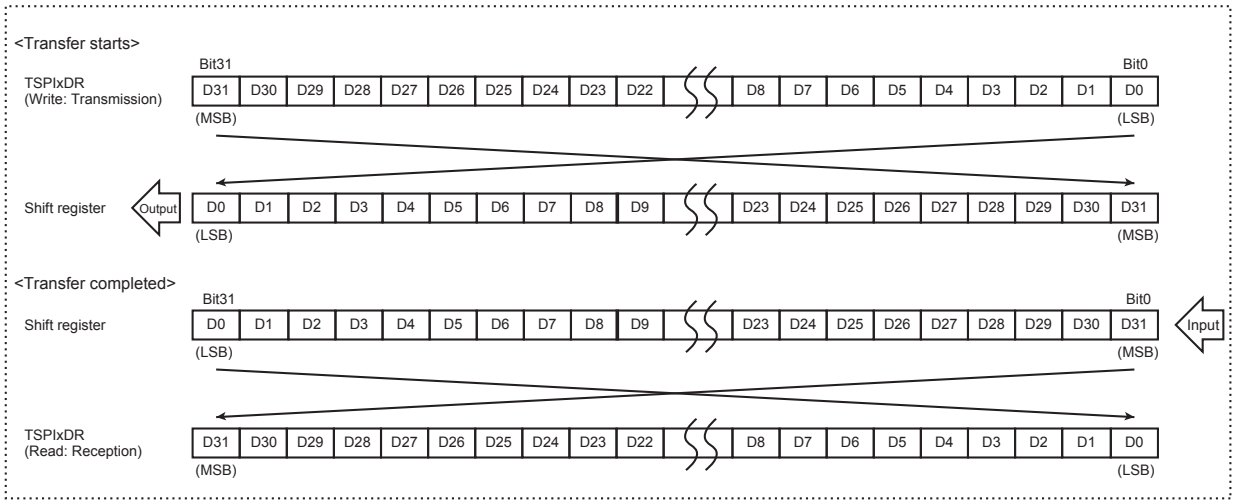


Figure 15-5 LSB first (32-bit data without a parity bit)

(4) LSB First Transfer (16-bit data without a parity bit, 16-bit frame length)

Figure 15-6 shows a 16-data length transmit/receive operation (without a parity bit, MSB first, 32-bit data length).

In the transmission, data in the transmit FIFO is sorted bit by bit when the data is copied to the shift register. Transmit data copied to shift register is transferred from D15 until reaching to 16-bit shifted data on serial clock.

In the reception, receive data is stored in the D0 of the shift register. Shift operation repeats on serial clock. If the shift register stores 32-bit reception data, data is sorted bit by bit and copied to receive FIFO.

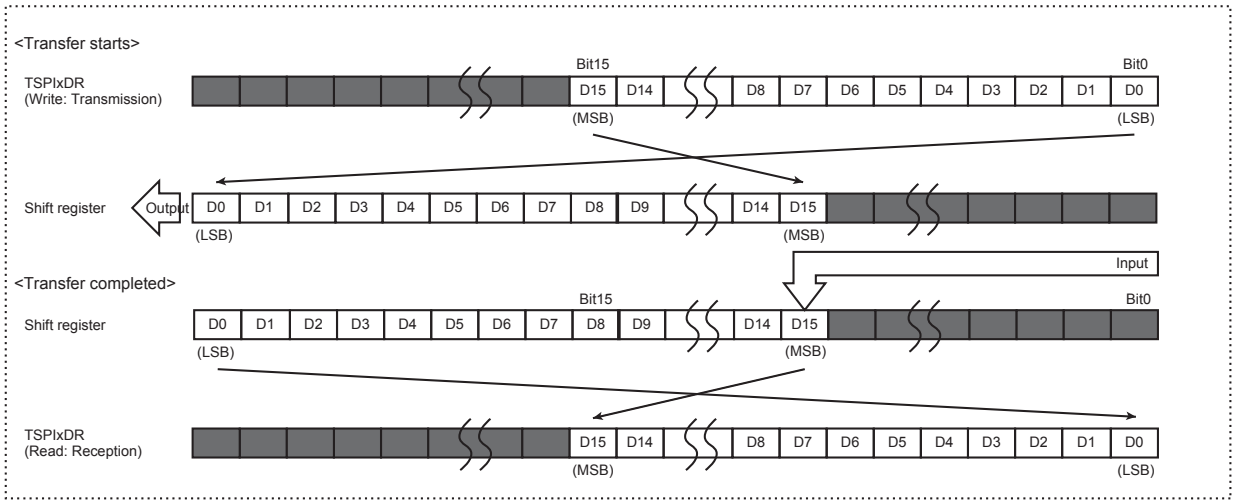


Figure 15-6 LSB first (16-bit data without a parity bit)

15.4.1.2 Data Format with a Parity Bit

If a parity bit is added, frame length is specified as a data length including a parity bit.

When data length is 10-bit, set "11" to TSPIxFMTR0<FL[5:0]>.

If a parity is used, a parity bit is automatically added to data in the transmit FIFO and the data is copied to shift register.

A parity bit is also automatically deleted from receive data in the shift register and the data is copied to receive FIFO.

(1) MSB first transfer (31 bits data with parity, 32-bit frame length.)

Figure 15-7 shows a 31-bit data length transmit/receive operation (with a parity bit, MSB first, 31-bit data length). A frame length is 32-bit data length including a parity bit.

In the transmission, data D30 through D0 in the transmit FIFO are copied to D31 through D1 in the shift register. At the same time, a parity is calculated using data D31 through D1 in the shift register. The result is stored in the D0 in the shift register.

Subsequently, transmit data in the shift register and parity data are sequentially transferred from D31 through D0 in the shift register on serial clock

In the reception, receive data is stored in the D0 of the shift register. Shift operation repeats on serial clock. If the shift register stores 32-bit reception data, data is copied to receive FIFO except a parity bit.

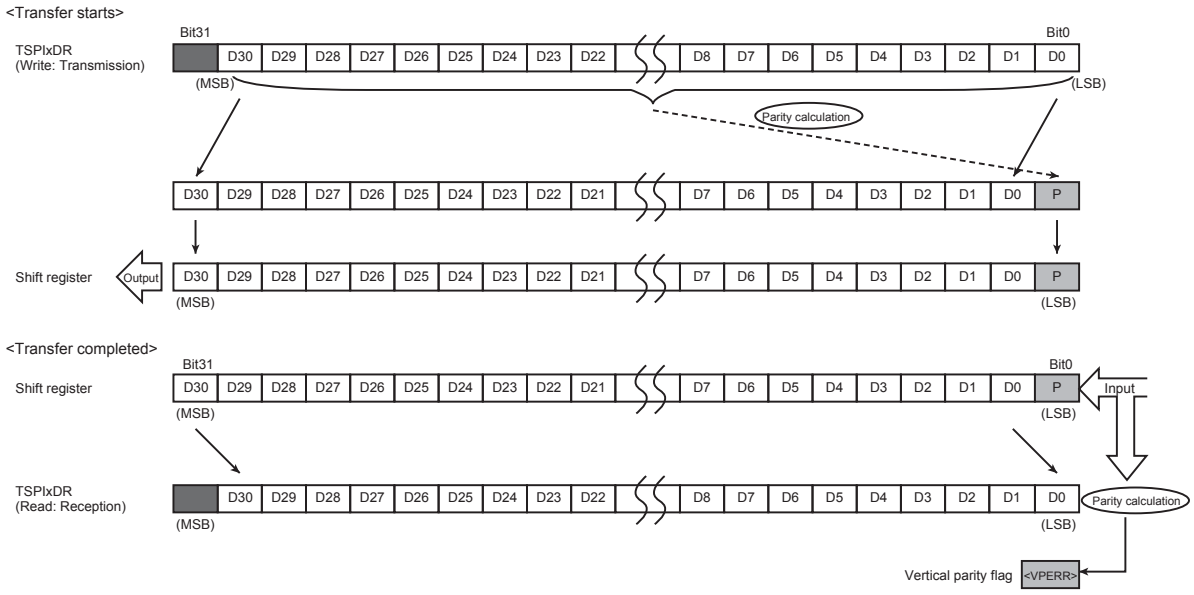


Figure 15-7 MSB first (31-bit data with a parity)

(2) MSB first transfer (15 bits data with parity, frame length is 16bit.)

Figure 15-8 shows a 15-bit data length transmit/receive operation (with a parity bit, MSB first, 15-bit data length). A frame length is 16-bit data length including a parity bit.

In the transmission, data D14 through D0 in the transmit FIFO is copied to D15 through D1 in the shift register. At the same time, a parity is calculated using data D14 through D0. The result is stored in the D0 in the shift register.

Subsequently, transmit data in the shift register and parity data are sequentially transferred from D31 through D0 in the shift register on serial clock

In the reception, receive data is stored in the D0 of the shift register. Shift operation repeats on serial clock. If the shift register stores 32-bit reception data, data is copied to receive FIFO except a parity bit.

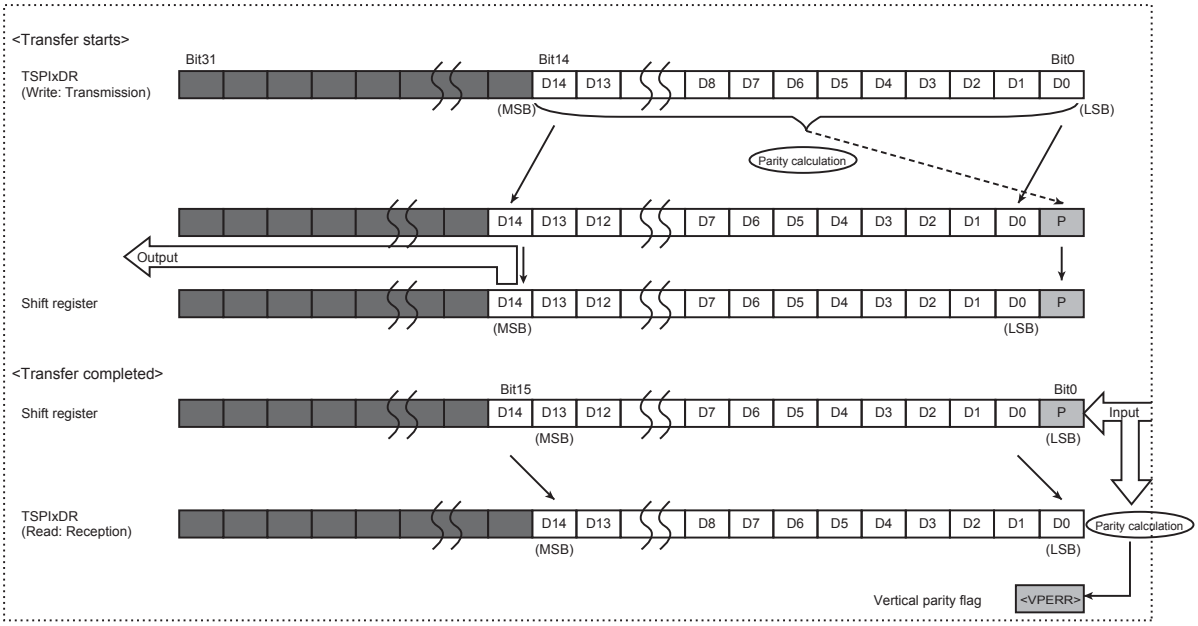


Figure 15-8 MSB first (15 bits data with parity)

(3) LSB first transfer (31 bits data with parity; frame length is 32bit)

Figure 15-9 shows a 31-bit data length transmit/receive operation (with a parity bit, LSB first, 31-bit data length).

In the transmission, data D30 through D0 in the transmit FIFO is sorted bit by bit and the data is copied to bit 31 through bit 1 in the shift register. At the same time, a parity is calculated using data D30 through D0. The result is stored in the D0 in the shift register.

Consequently, transmit data in the shift register and a parity data are sequentially transferred from D31 to D0 in the shift register on serial clock

In the reception, receive data is stored in the D0 of the shift register. Shift operation repeats on serial clock. If the shift register stores 31-bit reception data, only data excluding a parity bit is copied to receive FIFO.

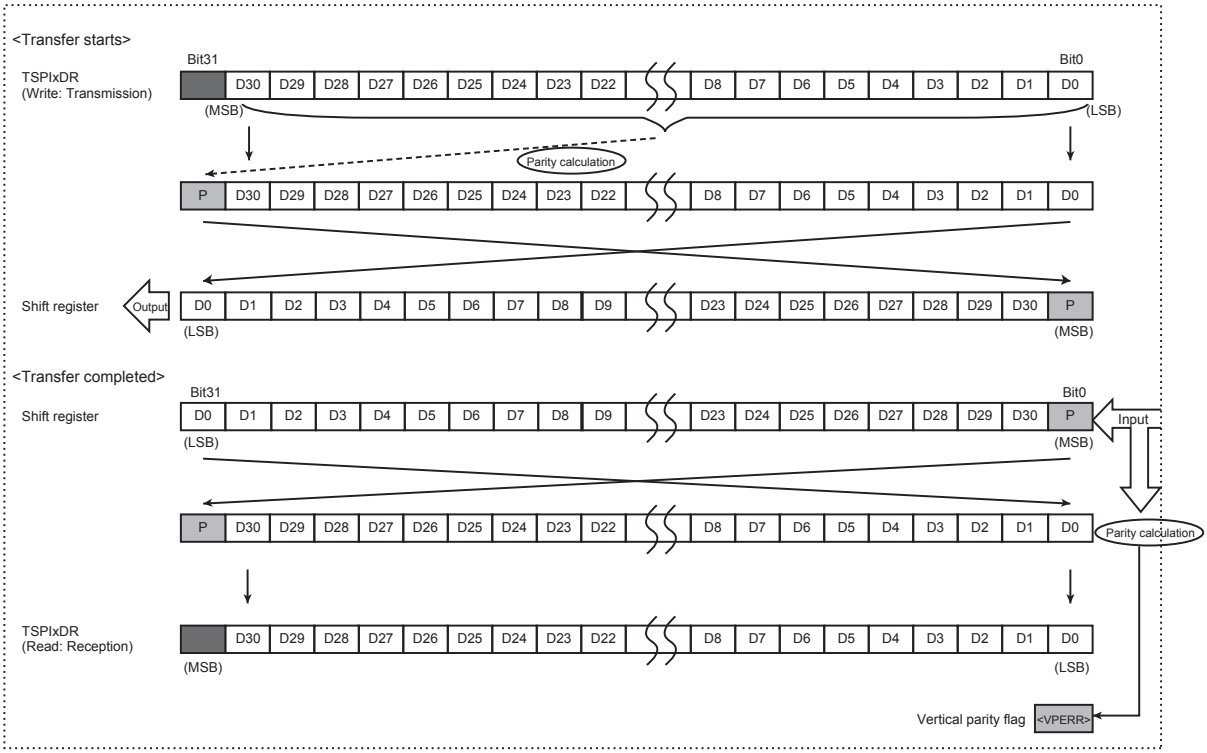


Figure 15-9 LSB first (31 bits data with parity)

(4) LSB first transfer (15-bit data with a parity bit, 16-bit frame length)

Figure 15-10 shows a 15-bit data length transmit/receive operation (with a parity bit, LSB first, 15-bit data length). A frame length is 16-bit data length including a parity bit.

In the transmission, data D14 through D0 in the transmit FIFO are sorted bit by bit and the data is copied to D17 from D31 in the shift register. At the same time, a parity is calculated using data D14 through D0. The result is stored in the D16 in the shift register.

Subsequently, transmit data in the shift register and parity data are sequentially transferred from D31 through D16 in the shift register on serial clock

In the reception, receive data is stored in the D0 of the shift register. Shift operation repeats on serial clock. If the shift register stores 16-bit reception data, only data excluding a parity bit is copied to receive FIFO.

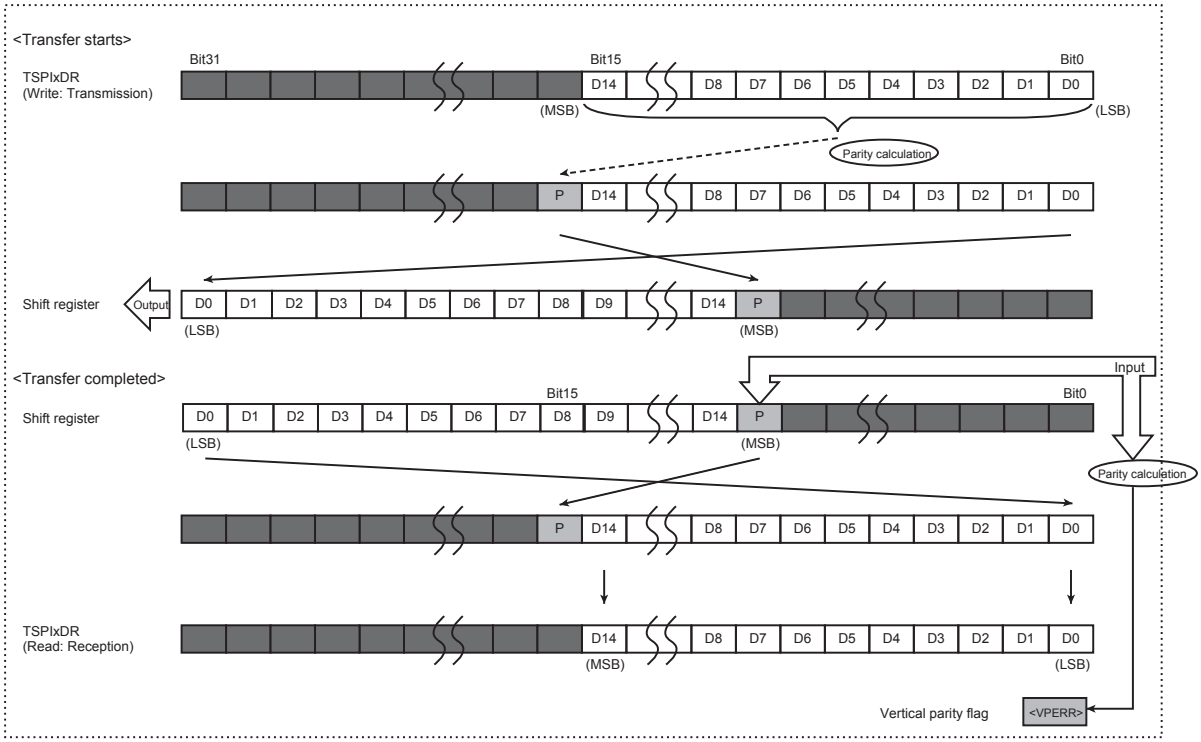


Figure 15-10 LSB first (15 bits data with parity)

15.4.2 Transfer Format

TSPI has one chip select signal outputs (TSPIxCSO) that enable communication with one type of external slave devices.

It also has one chip select signal (TSPIxCSIN) that enables communication with one type of master device.

A polarity of serial clock can be selected.

In the SPI mode, a polarity of TSPIxCSO and the generation timing can also be selected.

TSPIxTXD is selected in the idle mode, in which no communication is made.

15.4.2.1 Polarity of TSPIxCS Signal and Generation Timing

TSPIxCSO can select its polarity.

A logic of TSPIxCSO is set by TSPIxFMTR0<CS0POL>.

If TSPIxFMTR0<CS0POL> is set to "0", negative logic is selected.; if TSPIxFMTR0<CS0POL> is set to "1", positive logic is selected.

A generation timing of TSPIxCSO can be set. The following four timings are settable.

1. Serial clock delay

   t_A is a delay time from the time when TSPIxCSO is asserted until the transmit clock (TSPIxSCK) changes. To set a serial clock delay time, set TSPIxFMTR0<CSSCKDL>.

2. TSPIxCSO negate delay

   t_B is a delay time from the time when TSPIxCSO is negated after serial transfer completion. To set a TSPIxCSO negate delay time, set TSPIxFMTR0<SCKCSDL>.

3. Interval time between frames in the burst transfer

   t_C is an interval time between frames in the burst transfer. To set an interval time between frames, set TSPIxFMTR0<FINT>.

4. Minimum idle time

   t_D is a minimum wait time from the time when TSPIxCSO is negated and then until TSPIxCSO is asserted again. To set minimum idle time, set TSPIxFMTR0<CSINT>.

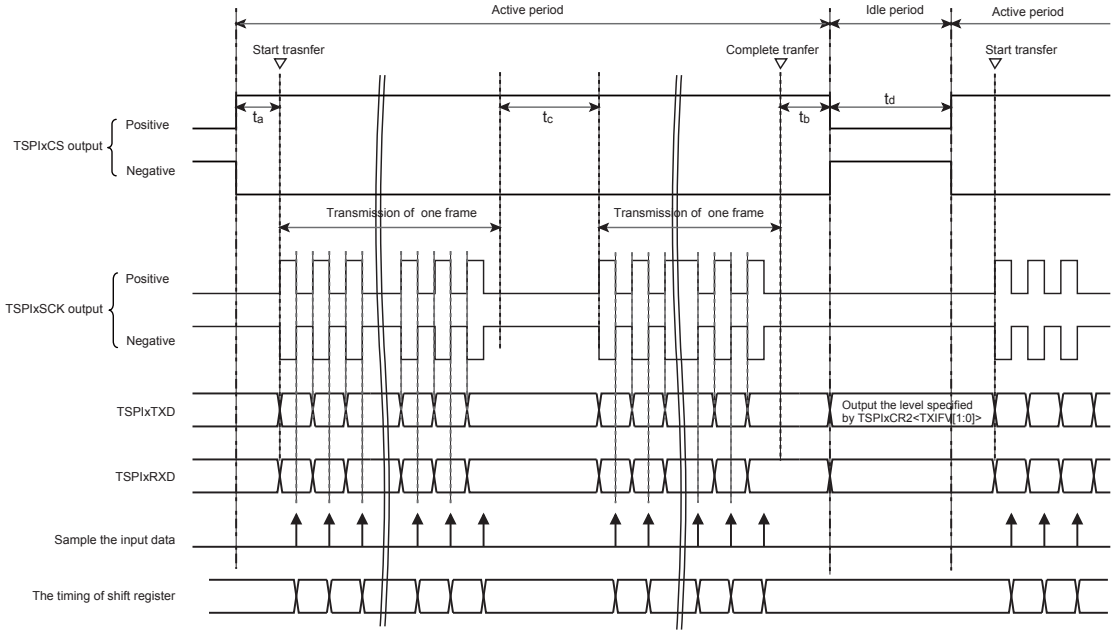


Figure 15-11 Transfer format and timing adjustment

15.4.2.2 Polarity of Clock

To select a polarity of clock, set TSPIxFMTR0<CKPOL>.

If TSPIxFMTR0<CKPOL>=0 is set, TSPIxSCK outputs Low during idle period and the first clock edge is a rising edge.

If TSPIxFMTR0<CKPOL>=1 is set, TSPIxSCK outputs High during idle period and the first clock edge is a falling edge.

15.4.2.3 TSPIxTXD Output during Idle

A level of TSPIxTXD output during idle period can be selected by TSPIxCR2<TXIFV[1:0]>.

Table 15-4 TSPIxTXD output during idle period

TSPIxCR2<TXIFV[1:0]>	Output
00	Hi-Z
01	Previously transferred data is held.
10	Low
11	High

After reset, if the setting that holds the last data of previous transfer is selected, TSPIxTXD outputs High.

TSPIxCR2<TXIFV[1:0]> = “10” : Example of outputting low at idle

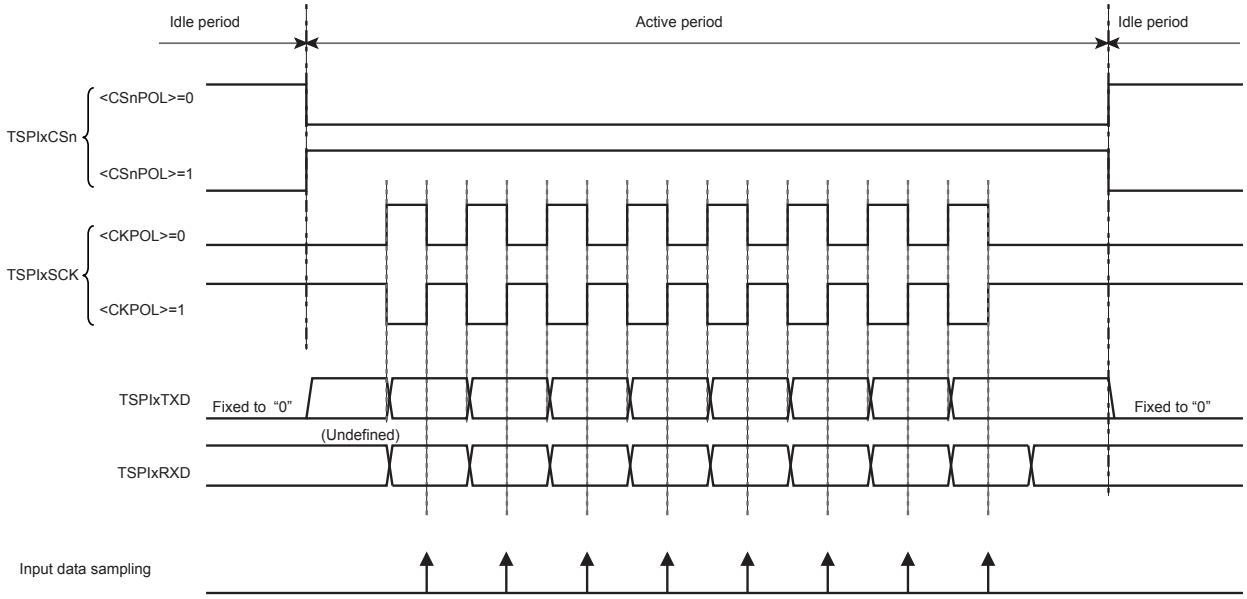


Figure 15-12 Idle period in SPI mode and the transmit pin status

TSPIxCR2<TXIFV[1:0]> = “10” : Example of outputting low at idle

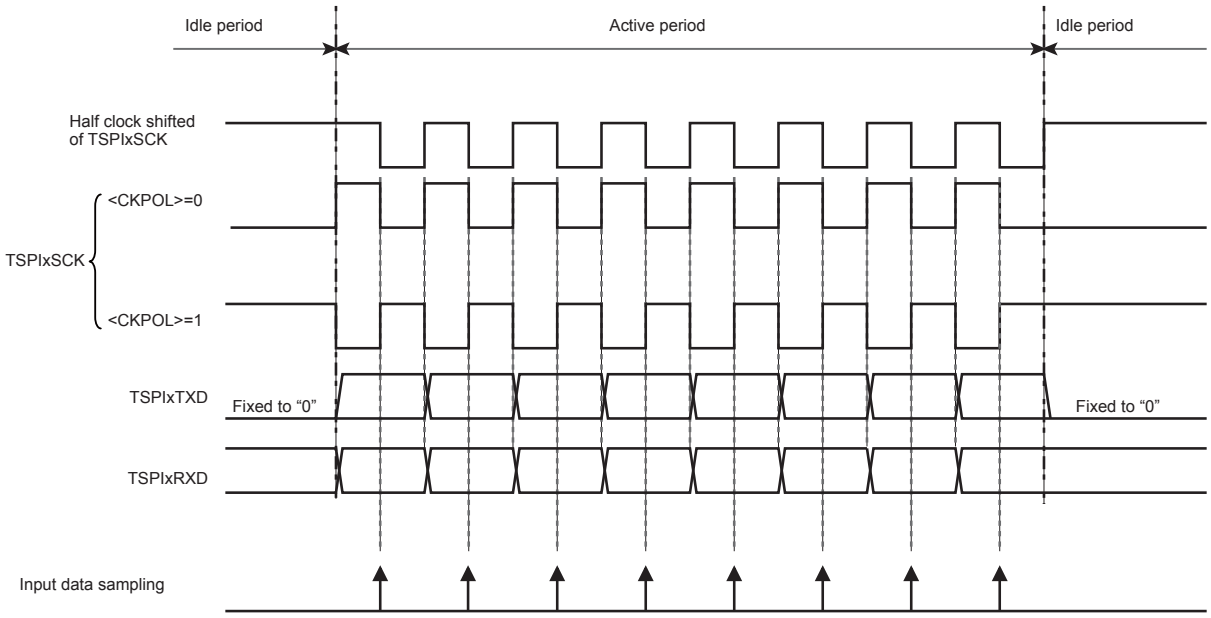


Figure 15-13 Idle period in SIO mode and the transmit pin status

15.4.3 Buffer Structure

A transmit buffer and receive buffer are independent respectively. Each buffer has a double-buffering structure consisting of FIFO and shift register.

There are transmit FIFO and receive FIFO. Each FIFO is 16-bit width and 8-stage. Settable FIFO level varies depending on the data length.

Shift register is 32-bit length.

Table 15-5 Data format and settable fill level

Data	Settable fill level	
	Transmit FIFO TSPIxCR2<TIL[3:0]>	Receive FIFO TSPIxCR2<RIL[3:0]>
7 to 16bit	0 to 7	1 to 8
17 to 32bit	0 to 3	1 to 4

Note: Set a value within a settable fill level. If a value outside of settable fill level is set, the operation is not guaranteed.

15.4.3.1 Data Length and FIFO Operation

Data register is 32-bit width. FIFO of TSPI adjusts data to 32-bit width for most efficient DMA transfer.

The following describes FIFO operation taking example of reception. Transmission is the same operation except data direction.

Received data are indicated as f0 and f1 in each frame. f0 indicates the first frame; f1 indicates the second frame.

Also, an upper byte and lower byte in one frame indicate as f1(H) and f1(L) respectively.

(1) Data length 7 to 16bit

If 7- to 16-bit data length are used, 1 stage of FIFO is used for storing 1 frame data.

FIFO is 8 stages that can store data up to 8 stage levels.

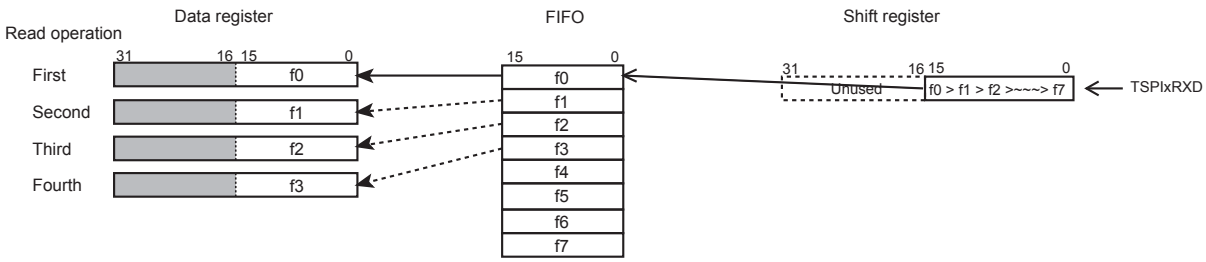


Figure 15-14 Operation in 7- to 16-bit data length

Input data to TSPIxRXD is captured by shift register.

When a certain frame length is transferred, if FIFO has space, received data in shift register is copied to FIFO. Data is stacked in FIFO in the order starting from f0, f1, f2, f3, f4, f5, f6 and f7.

If DMAC or CPU reads data register, a content of the stage in receive FIFO directed by receive FIFO pointer is read.

On the first read operation, the first stage (f0) of FIFO is copied to the lower 16 bits in the data register. Upper 16 bits in data register become undefined.

Receive FIFO pointer is incremented by one and directs the second stage of FIFO.

On the second read operation, f1is copied to lower 16 bits in the data register.

Receive FIFO pointer is incremented by one and directs the third stage of FIFO.

In the subsequent frames, receive pointer is incremented on reading operations, and each data is copied to lower 16 bits in the data register.

(2) Data length 17 to 32bit

If 17- to 32-bit data length are used, 2 stages of FIFO are used for 1 frame.

FIFO is 8 stages that can store four-frame data up to 4 stage levels.

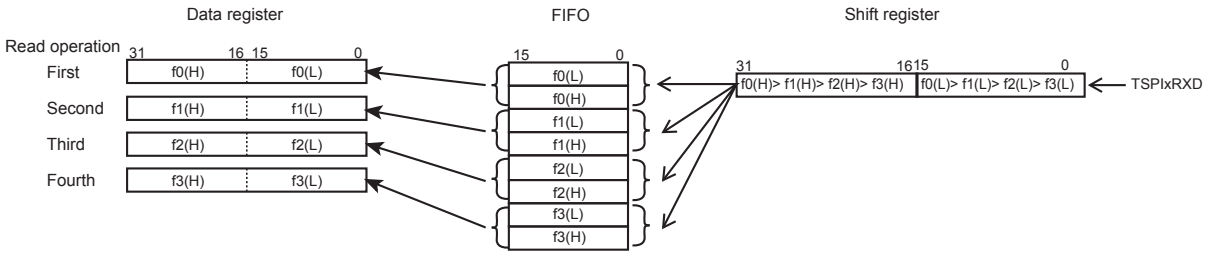


Figure 15-15 Operation in 17- to 32-bit data length

Input data to TSPIxRXD is captured in the shift register.

When a certain frame length is transferred, if FIFO has space, received data in shift register is copied to FIFO. Data is stacked in FIFO in the order starting from f0(L), f0(H). In the next frame, data is copied and stacked in the FIFO in the order starting from f0, f1, f2, f3, f4, f5, f6 and f7.

If DMAC or CPU reads data register, a content of the stage in receive FIFO directed by receive FIFO pointer is read.

On the first read operation, the first stage (f0) of FIFO is copied to the lower 16 bits in the data register. A content in the second stage f0(H) of FIFO is copied to upper 16 bits in data register.

Receive FIFO pointer is incremented by two and directs the 3rd stage of FIFO.

On the second read operation, f1 is read the same as f0. Receive FIFO pointer directs 5th stage of FIFO.

In the subsequent frames, receive pointer is incremented by two on reading operations, and fm(L) data is copied to lower 16 bits in the data register; fm(H) data is copied to upper 16 bits in the data register.

15.4.4 Interrupt Request

TSPI has three types of interrupts which are receive interrupts, transmit interrupts and error interrupts.

Each interrupt is output consisting of some signals related to interrupts. They are enabled/disabled respectively.

Table 15-6 Interrupt events and requests

Interrupt request	interrupt event	Enable register
Receive interrupt	Receive completion interrupt	TSPIxCR2<INTRXWE>
	Receive FIFO interrupt	TSPIxCR2<INTRXFE>
Transmit interrupt	Transmit completion interrupt	TSPIxCR2<INTTXWE>
	Transmit FIFO interrupt	TSPIxCR2<INTTXFE>
Error interrupt	Parity error interrupt	TSPIxCR2<INTERR>
	Overrun error interrupt	
	Underrun error interrupt	

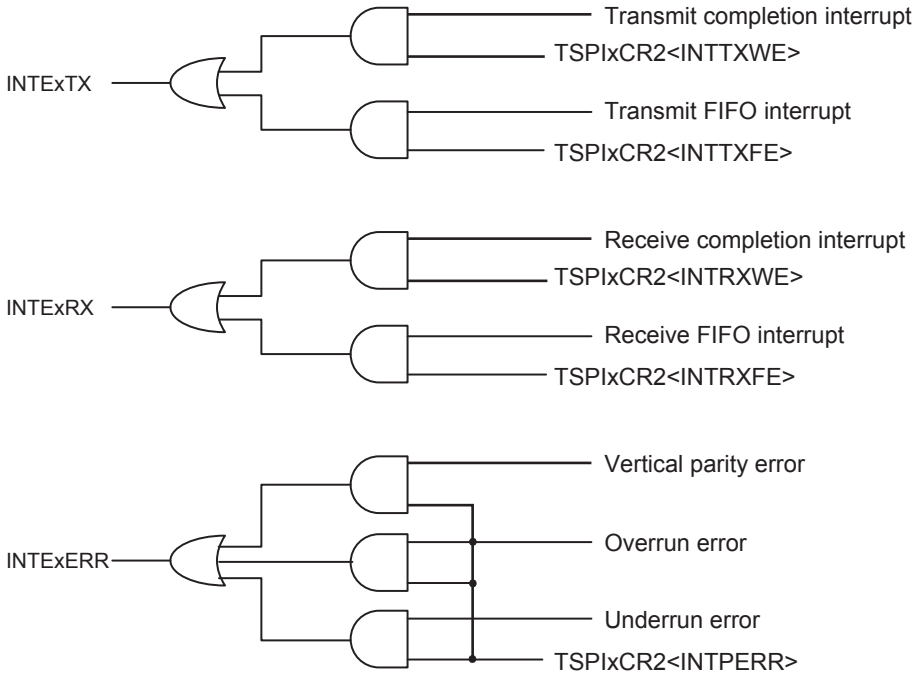


Figure 15-16 Interrupt events & requests

15.4.4.1 Transmit Completion Interrupt/Receive Completion Interrupt

Both in the single transfer and burst transfer, when a transmit completion interrupt occurs when TSPIxCSO is negated in transmission or full-duplex communications.

Both in the single transfer and burst transfer, when receive completion interrupt occurs when TSPIxCSO is negated in reception or full-duplex communications.

15.4.4.2 Transmit FIFO Interrupt/Receive FIFO Interrupt

A transmit FIFO interrupt occurs when the following conditions are met.

- TSPIxSR<TLVL[3:0]> is greater than transmit FIFO interrupt condition (Fill level) specified in TSPIxCR2<TIL[3:0]> by one.
- Data is transferred from transmit FIFO to transmit shift register. Fill level of transmit FIFO is decreased by one and the level is changed to the same value of transmit interrupt generation condition (Fill level).

A transmit FIFO interrupt occurs when the following conditions are met.

- TSPIxSR<RLVL[3:0]> is less than receive interrupt generation condition (Fill level) specified in TSPIxCR2<RIL[3:0]> by one.
- Data is transferred from receive shift register to receive FIFO. Fill level of receive FIFO is increased by one and the level is changed to the same value of receive interrupt generation condition (Fill level).

15.4.4.3 Error Interruption

(1) Parity Error Parity

A parity error interrupt is an interrupt that occurs when a parity error occurs.

If a parity is enabled, a parity is calculated using received data that is received 1 bit previous to the last data of the frame.

A received parity bit as the last bit of frame is compared. If they don’t match, a parity error interrupt occurs.

An interrupt generation timing is the timing at which receive frame data is stored in the receive FIFO.

(2) Overrun error interrupt and underrun error interrupt

An underrun and overrun errors occur in the slave mode.

An underrun error occurs when data does not exist in the transmit FIFO after data in the shift register is transferred completely, if next transfer clock is input.

An overrun error occurs when receive FIFO is full and receive shift register contains data, if next transfer clock is input.

Data in the frame where an overrun occurs is not received. Thus, contents of receive FIFO and receive shift register are not updated.

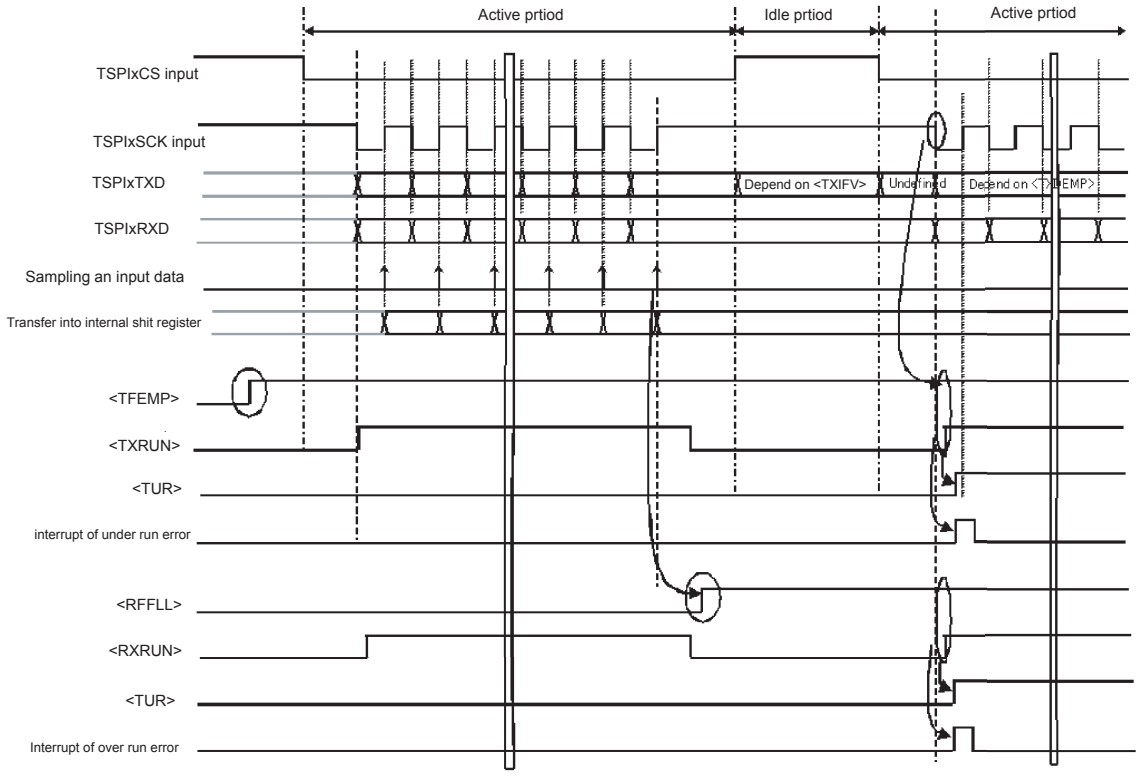


Figure 15-17 Overrun error and under run error

15.4.5 DMA Request

A transmit DMA request occurs when a value of TSPIxSR<TLVL[3:0]> indicating current value of fill level is equal or less than transmit interrupt generation condition (Fill level) specified in TSPIxCR2<TIL[3:0]>. If TSPIxSR<TLVL[3:0]> is still equal or less than the fill level after complication of DMA transfer, a transmit DMA request occurs again.

A receive DMA request occurs when a value of TSPIxSR<RLVL[3:0]> indicating current value of fill level is equal or greater than receive interrupt generation condition (Fill level) specified in TSPIxCR2<RIL[3:0]>. If TSPIxSR<RLVL[3:0]> is still equal or greater than TSPIxCR2<RIL[3:0]> after complication of DMA transfer, a receive DMA request occurs again.

15.4.6 Transfer Mode

A transfer mode consists of two modes; single and burst transfer. Single transfer can transfer one frame of data; burst transfer can transfer multiple frames of data.

Transfer mode is set by the frame specified in TSPIxCR1<FC[7:0]>.

15.4.6.1 Single Transfer

Single transfer is the mode that can transfer one frame.

In the SPI mode, if one frame transfer is complete, TSPIxCSO is surely negated.

15.4.6.2 Burst Transfer

Burst transfer is the transfer mode that can consecutively transfer multiple frames.

In the SPI mode, TSPIxCSO keeps asserted condition while specified frames are transferring.

15.4.7 SIO/SPI mode

SIO mode and SPI mode are in TSPI.

Transfer mode is specified by TSPIxCR1<TSPIMS>.

15.4.7.1 SPI mode

If TSPIxCR1<TSPIMS> is clear to "0", TSPI operates in SPI mode.

In SPI mode, one master device can be connected with one slave device via TSPIxCLK, TSPIxCSO, TSPIxCSIN, TSPIxTXD, TSPIxRXD.

15.4.7.2 SIO mode

If TSPIxCR1<TSPIMS> is clear to "1", TSPI operates in SIO mode.

In SIO mode, one master device can be connected with one slave device via TSPIxCLK, TSPIxTXD, TSPIxRXD.

15.4.8 Master / Slave

TSPI operates as Master (the device output clock) or Slave (the device input clock).

TSPI operates as Master if TSPIxCR1<MSTR> is "0".

TSPI operates as Slave if TSPIxCR1<MSTR> is "1".

15.4.8.1 Operation as Slave in SPI mode

TSPI receives a chip select signal input via TSPIxCSIN pin and operates synchronously with TSPIxCLK. If a chip select signal is invalid, TSPIxSCK is ignored.

15.4.8.2 Operation as Master in SPI mode

TSPI outputs a chip select signal from TSPIxCSO and outputs a clock from TSPIxSCK.

TSPI operates synchronously with TSPIxSCK.

15.4.8.3 Operation as Slave in SIO mode

TSPI receives a clock from TSPIxSCK and operates synchronously with TSPIxSCK.

15.4.8.4 Operation as Master in SIO mode

TSPI outputs a clock from TSPIxSCK and operates synchronously with TSPIxSCK.

15.5 TSPI Control

15.5.1 Reset

TSPI is initialized by reset or software reset.

To perform a software reset, consecutively write "10"→"01" to TSPIxCR0<SWRST[1:0]>.

Control register, transmit/receive FIFO, transmit/receive shift register and internal circuits are wholly initialized by software reset except TSPIxCR0<TSPIE>.

Note: Software reset operation needs two clocks after the instruction is issued.

Table 15-7 Initialized registers by software reset

Register name	Symbol name
TSPIxCR0	No registers
TSPIxCR1	<TRXE>
TSPIxCR2	<TIL><RIL><INTTXFE><INTTXWE><INTRXFE><INTRXWE><INTERR><DMATE><DMARE>
TSPIxCR3	No registers
TSPIxBR	No registers
TSPIxFMTR0	No registers
TSPIxFMTR1	No registers
TSPIxDR	No registers
TSPIxSR	<TSPISUE><TXRUN><TXEND><INTTXWF><TFEMP> <TLVL><RXRUN><RXEND><INTRXFF><RFFLL><RLVL>
TSPIxERR	<UDRERR><OVRERR><PERR>

15.5.2 Initial Setting of TSPI

To set TSPI, set "1" to TSPIxCR0<TSPIE>.

Subsequently, confirm if TSPIxSR<TSPISUE> is "0". Then set communication mode, transfer mode or transfer format, if necessary.

15.5.3 Start/Stop Transfer

There are two ways to start transfer in full duplex communication mode and transmit mode.

1. After TSPIxCR1<TRXE>=1 is set to enable the communication, write data to data register TSPIxDR.
2. After writing data to data register TSPIxDR, set TSPIxCR1<TRXE>=1 to enable the communication.

In the receive mode, set TSPIxCR1<TRXE>=1 to start reception immediately.

To stop transfer, set "0" to TSPIxCR1<TRXE>. Both in the single and burst transfer, a frame of transfer in progress continues operation until the frame is completely transferred.

If a transfer is stopping, TSPIxSCK, TSPIxCSO and TSPIxTXD are in the idle mode.

If a transfer is re-enabled after the transfer was stopped in the burst mode, the transfer re-starts its operation from the beginning of the burst transfer that was stopped

Table 15-8 shows a details of start/stop operation in the communication mode and transfer mode respectively.

Table 15-8 Transfer starting and stopping operation in each mode setting

communi-cation mode	Transfer mode	Communication start timing	Transfer stop timing
Full duplex	Single transfer	A transfer starts if TSPIxCR1<TRXE> bit is "1" and valid data exists in the transmit FIFO. Either condition can start transferring. When receive buffer (receive FIFO and receive shift register) is full, next frame cannot be transferred. When reading receive FIFO data, if data in the receive shift register is automatically transferred to FIFO, the shift register is determined as not full and a transfer is automatically re-started.	In transmission/reception, if this bit is set to stop, a transfer is stopped after a frame in progress is complete.
	Burst transfer	A transfer starts if TSPIxCR1<TRXE> bit is "1" and valid data exists in the transmit FIFO. Either condition can start transferring. In the burst transfer mode, if specified number of burst transfers are complete, TSPIxCR1<TRXE> bit returns to "0". If a burst transfer is performed again, set "1" to TSPIxCR1<TRXE> after the confirmation that TSPIxSR<TSPISUE> bit was returned to "0". If data in the transmit FIFO runs out during specified number of burst transfers, TSPIxCSO stays asserted. A transfer automatically re-starts at a point when valid data is written to the transmit FIFO. If a receive buffer (receive FIFO and receive shift register) is full, next frame cannot be transferred. At this time, TSPIxCSO stays asserted. When reading receive FIFO data, if data in the shift register is automatically transferred to FIFO, shift register is determined as not full and a transfer is automatically re-started.	In transmission/reception, if this bit is set to stop, a transfer is stopped after a frame in progress is complete.
Transmission mode	Single transfer	A transfer starts If TSPIxCR1<TRXE> bit is "1" and valid data exists in the FIFO. Either condition can start transferring.	In transmission/reception, if this bit is set to stop, a transfer is stopped after a frame in progress is complete.
	Burst transfer	A transfer starts if TSPIxCR1<TRXE> bit is "1" and valid data exists in the transmit FIFO. Either condition can start transferring. To perform burst transfer again, set "1" to <TRXE> after the confirmation that TSPIxSR<TSPISUE> bit was returned to "0". If data in the transmit FIFO runs out during specified number of burst transfers, TSPIxCSO stays asserted. A transfer automatically re-starts at a point when valid data is written to the transmit FIFO.	In transmission/reception, if this bit is set to stop, a transfer is stopped after a frame in progress is complete.
Receive mode	Single transfer	A reception starts if receive buffer (receive FIFO and receive shift register) is not full. If receive buffer is full, serial clock stops and next frame cannot be transferred. When reading receive FIFO data, if data in the shift register is automatically transferred to FIFO, the shift register is determined as not full and a transfer is automatically re-started.	In reception, if this bit is set to stop, a transfer is stopped after a frame in progress is complete.
	Burst transfer	A reception starts if receive buffer (receive FIFO and receive shift register) is not full. If receive buffer is full, next frame cannot be transferred. When reading receive FIFO data, if data in the shift register is automatically transferred to FIFO, the shift register is determined as not full and a transfer is automatically re-started. TSPIxCSO stays asserted until a transfer starts again. If burst transfer is performed again, set "1" to <TRXE> after TSPIxSR<TSPISUE> bit was returned to "0".	In reception, if this bit is set to stop, a transfer is stopped after a frame in progress is complete.

15.6 Communication Mode

15.6.1 Full Duplex Communication Mode

Figure 15-18 shows an operation example of full duplex communication in single transfer (32-bit frame length, no parity, one-stage of FIFO).

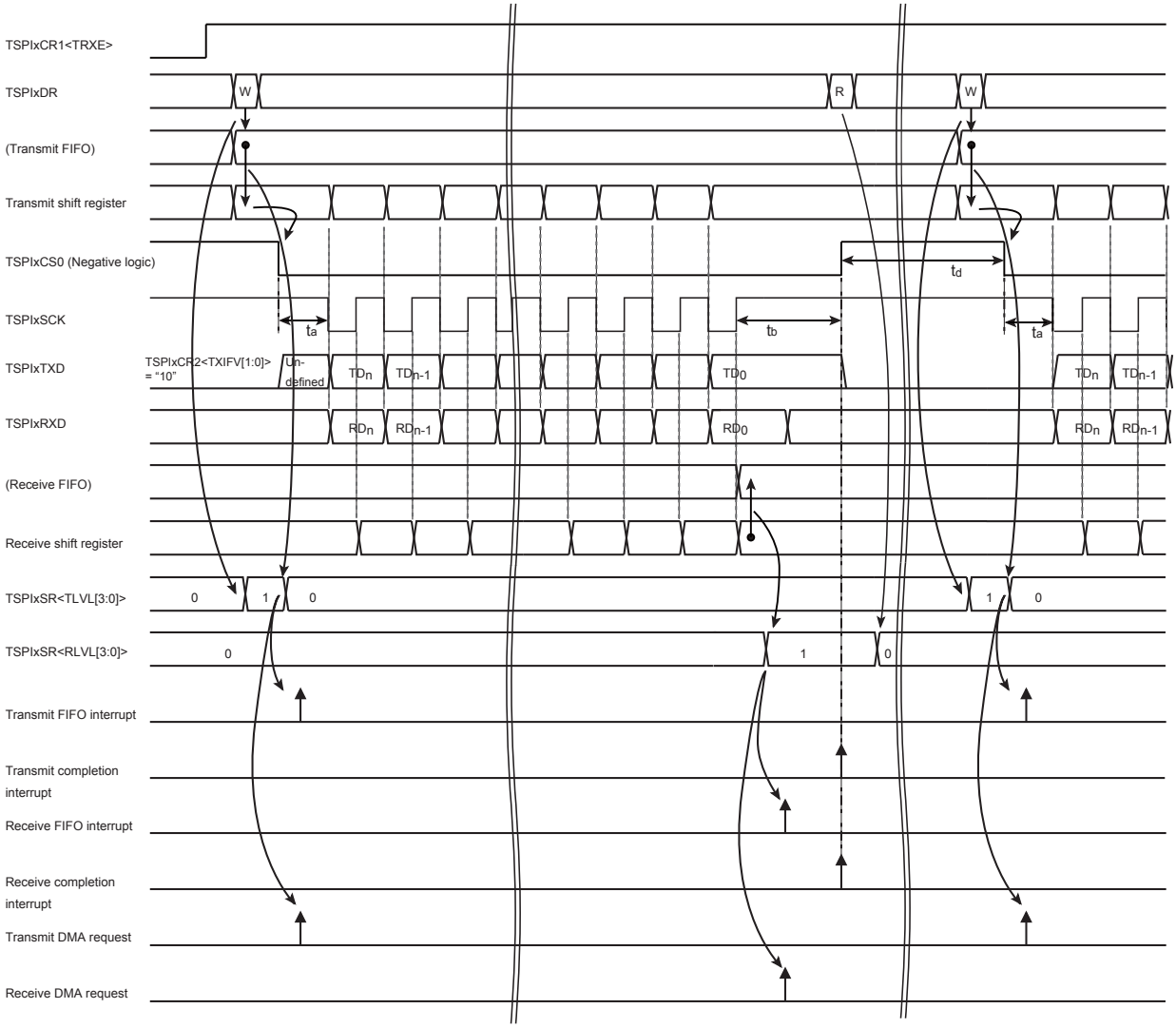


Figure 15-18 Operation example of full duplex communication

a. Write "1" to TSPIxCR1<TRXE> to enable communication
b. Write data to TSPIxDR
c. If data is written to TSPIxDR, the data is written to the stage of FIFO directed by internal transmit FIFO pointer.
d. Because one stage of data is buffered in transmit FIFO, TSPIxSR<TLVL> becomes "1".
e. Buffered data in transmit FIFO is copied to the shift register, so that TSPIxSR<TLVL> becomes "0". After a serial clock delay time (ta) specified by TSPIxFMTR0<CSSCKDL> has elapsed, TSPIxSCK starts outputting serial clock.
f. Because TSPIxSR<TLVL> changes to "0" from "1", a transmit FIFO interrupt (or transmit DMA request) occurs.

g. On the last rising edge of serial clock, all bits of receive data are captured by receive shift register and copied to receive FIFO. After a CS negate delay time (tb) specified by TSPIxFMTR0<SCKCSDL> has elapsed after the last rising edge of serial clock, TSPIxCSO is negated so that a transmit completion interrupt and receive completion interrupt occur.

h. Because one stage of receive FIFO is buffered, TSPIxSR<RLVL> becomes "1".

i. Because TSPIxSR<RLVL> changes to "1" from "0", a receive FIFO interrupt (or receive DMA request) occurs.

j. Until a minimum idle time (td) specified by TSPIxFMTR0<CSINT> has elapsed after TSPIxCSO is negated, serial transfer does not start and TSPIxCSO remains negated. After a minimum idle time (td) has elapsed, TSPIxCSO is asserted and serial transfer starts.

15.6.2 Transmit Mode

Figure 15-19 shows an operation example of single transfer (32-bit frame length, no parity, one-stage of FIFO) in the transmit mode.

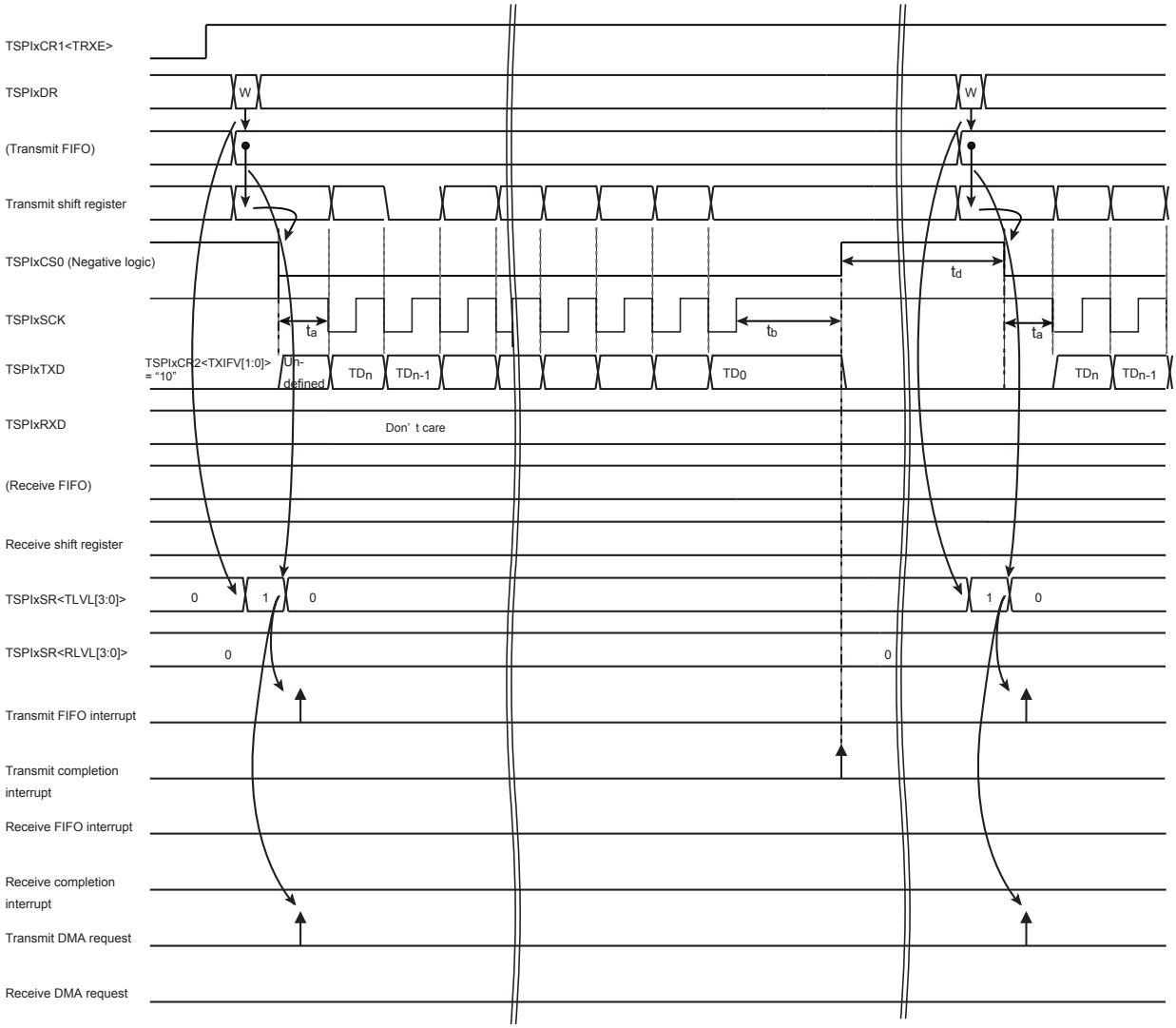


Figure 15-19 Operation example of transmit mode

a. Write "1" to TSPIxCR1<TRXE> to enable the communications.

b. Write data to TSPIxDR.

c. If data is written to TSPIxDR, data is written to a stage of FIFO directed by internal transmit FIFO pointer.

d. Because one stage of data is buffered to transmit FIFO, TSPIxSR<TLVL> becomes "1".

e. Because buffered data in transmit FIFO is copied to the shift register, TSPIxSR<TLVL> becomes "0". After a serial clock delay time (ta) specified by TSPIxFMTR0<CSSCKDL> has elapsed, TSPIxSCK starts outputting serial clock.

f. Because TSPIxSR<TLVL> changed to "0" from "1", a transmit FIFO interrupt (or transmit DMA request) occurs.

g. Until a minimum idle time (td) specified by TSPIxFMTR0<CSINT> has elapsed after TSPIxCSO is negated, serial transfer does not start and TSPIxCSO remains negated. After a minimum idle time (td) has elapsed, TSPIxCSO is asserted and serial transfer starts.

15.6.3 Receive Mode

Figure 15-20 shows an operation example of single transfer (32-bit length, no parity, one stage of FIFO) in receive mode.

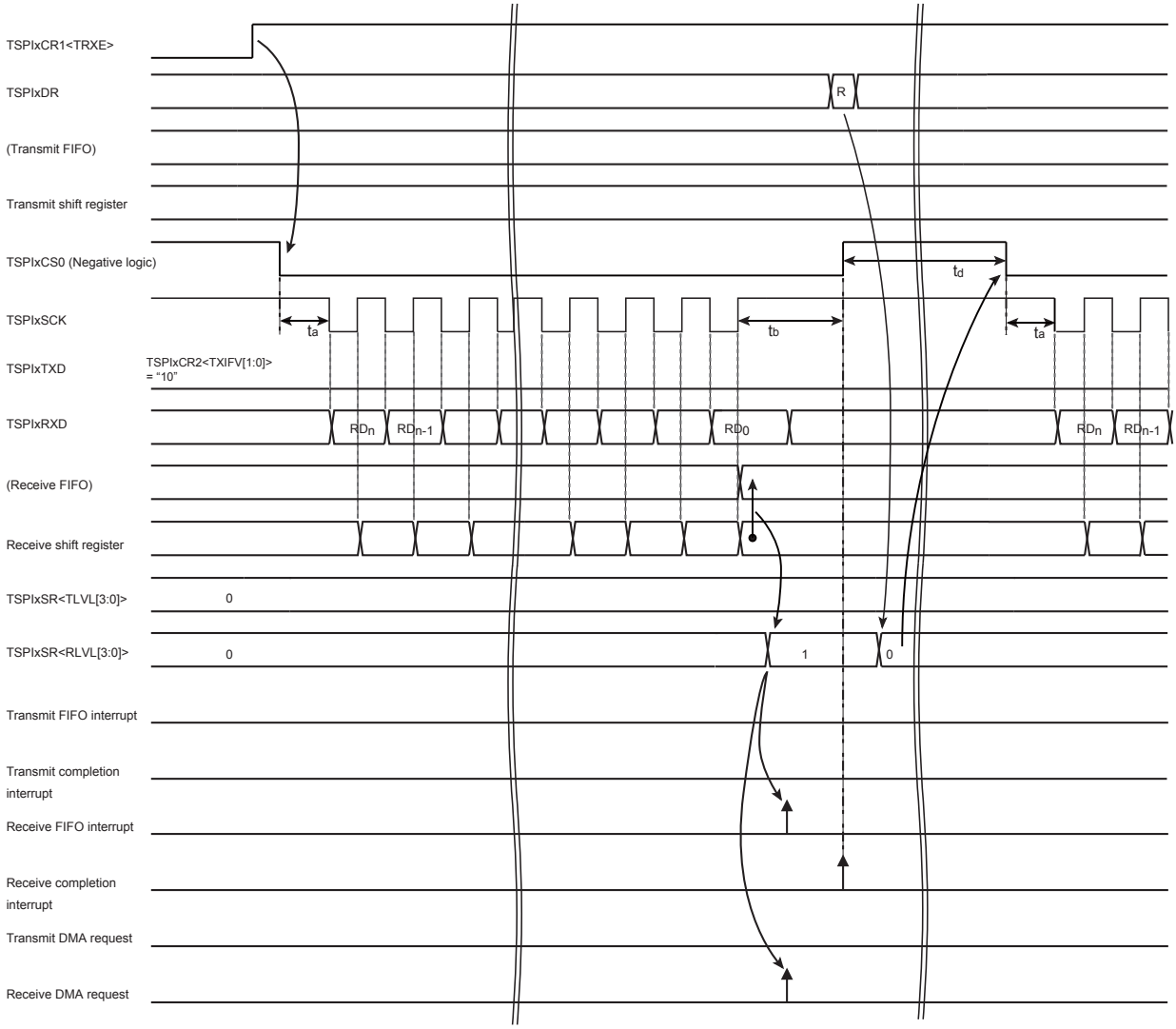


Figure 15-20 Operation example in receive mode

a. Write "1" to TSPIxCR1<TRXE> to enable the communications. Since receive FIFO is not full, TSPIxCSO is immediately asserted and serial clock transfer starts.
b. After a serial clock delay time (ta) specified by TSPIxFMTR0<CSSCKDL> has elapsed, serial clock starts outputting from TSPIxSCK.
c. On the last rising edge of serial clock, all bits of receive data are captured in the receive shift register and the data is copied to receive FIFO.
d. Because one stage data is buffered to receive FIFO, TSPIxSR<RLVL> becomes "1".
e. Because TSPIxSR<RLVL> changed to "1" from "0", a receive FIFO interrupt (or receive DMA request) occurs.
f. After a CS negated delay time (td) specified by TSPIxFMTR0<SCKCSDL> has elapsed after the last rising edge of serial clock, TSPIxCSO is negated and a receive completion interrupt occurs.
g. Until a minimum idle time (td) specified by TSPIxFMTR0<CSINT> has elapsed after TSPIxCSO is negated, serial transfer does not start and TSPIxCSO remains negated. After a minimum idle time (td) has elapsed, TSPIxCSO is asserted and serial transfer starts if receive FIFO is not full.

16. USB Device Controller (USBD)

This section describes about USB device controller.

An endpoint is described as EP in this section.

16.1 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 5 EPs.

Table 16-1 Endpoints

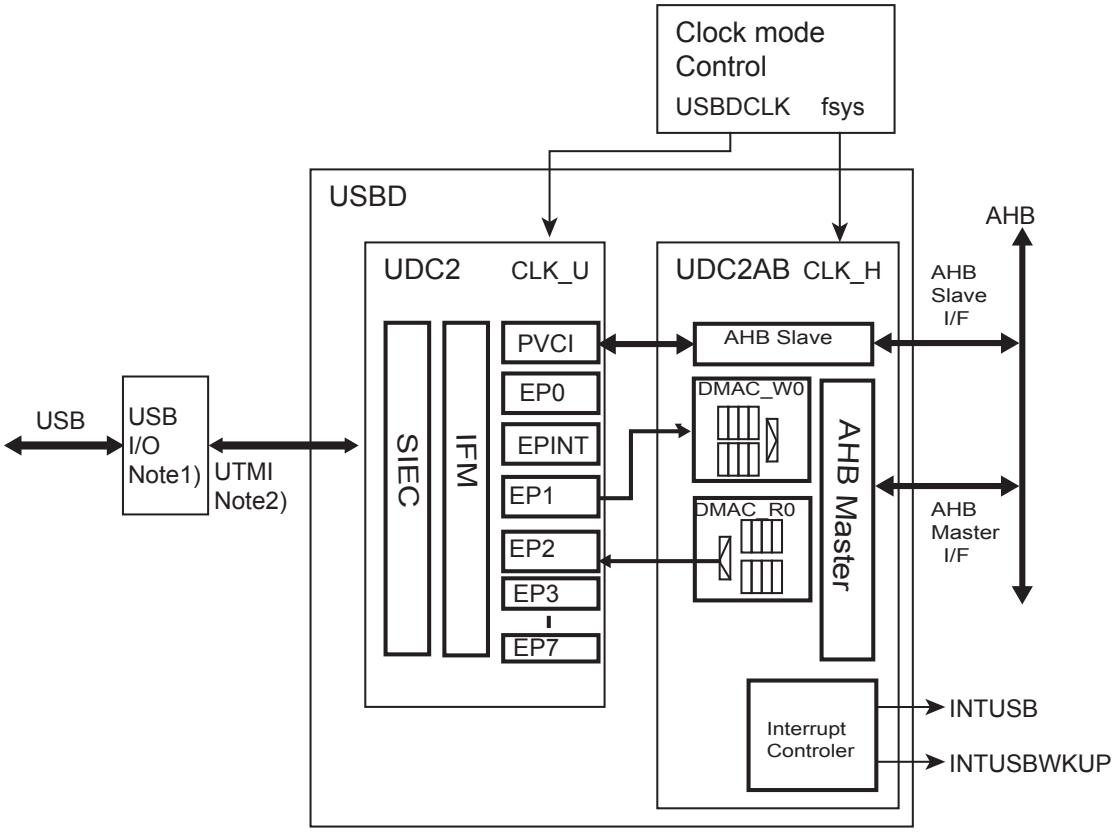
EP0:	Control	64byte × 1 FIFO
EP1:	Control / Interrupt / Bulk / Isochronous (IN)	64byte × 2 FIFO
EP2:	Control / Interrupt / Bulk / Isochronous (OUT)	64byte × 2 FIFO
EP3:	Control / Interrupt / Bulk / Isochronous (IN)	64byte × 2 FIFO
EP4:	Control / Interrupt / Bulk / Isochronous (OUT)	64byte × 2 FIFO

9. Supports Dual Packets Mode (except for EP 0)
10. Interrupt source signals to the interrupt controller: INTUSB, INTUSBWKUP

16.2 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, "16.2.1 AHB Bus Bridge (UDC2AB)" describes the configuration of the UDC2AB, and "16.2.2 Toshiba USB-Spec2.0 Device Controller (UDC2)" describes that of the UDC2.



Note1) TMPM066/067/068FW has USB I/O which can be used for Full Speed mode not Low speed mode.
The word "PHY" in this section should be read as USB I/O.

Figure 16-1 Block diagram of the USB device controller

16.2.1 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.

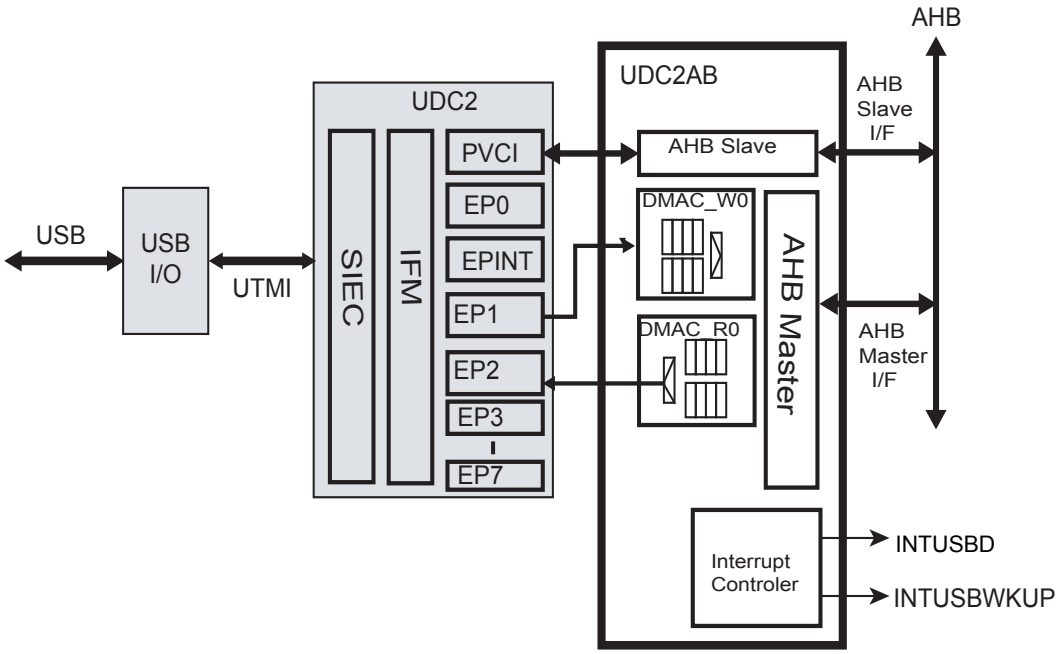


Figure 16-2 UDC2AB Block Diagram

16.2.1.1 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 16-2 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.

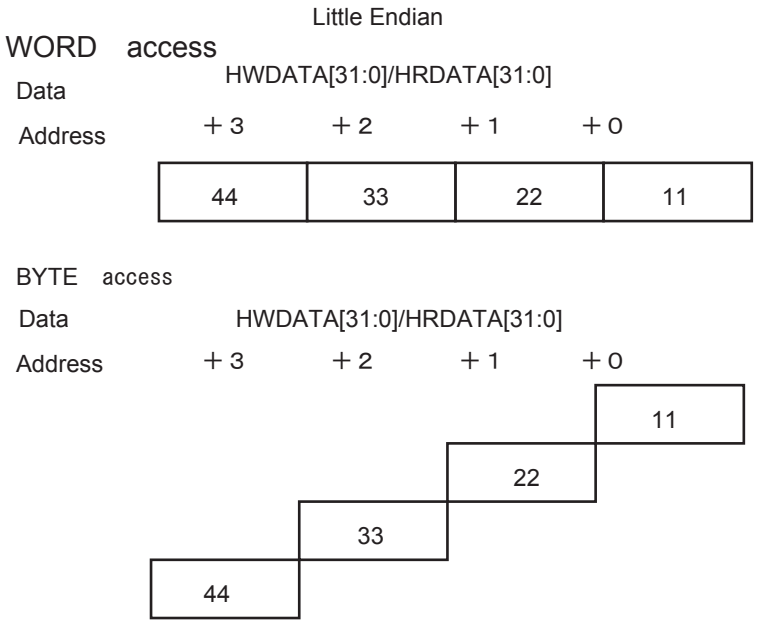


Figure 16-3 Image of Endian conversion in AHB Master function

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.

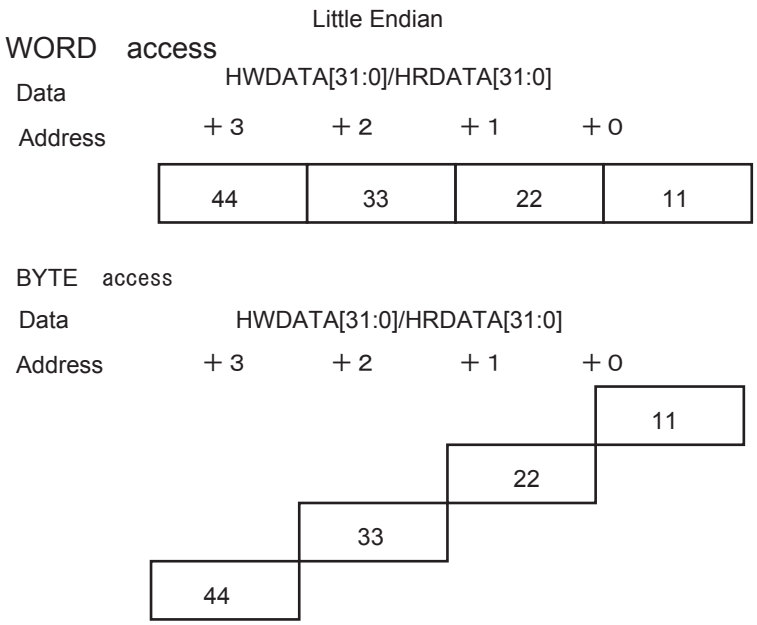


Figure 16-4 Image of Endian conversion in AHB Slave function

16.2.1.2 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).

16.2.1.3 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 TMPM066/067/068FW.

Since CLK_H is not provided during the fsys being stopped in the low-power consumption mode, INTUSB will not be generated.

Therefore, to detect the connection and disconnection of the VBUS, an interrupt to used needs to be selected from a External InterruptD generated by the VBUSD pin and INTUSB at the moment when CLK_H starts or stops.

Refer to "16.5.5.2 Sequence of USB Bus Power (VBUS) Connection/Disconnection" for more information.

USBDCLK provided from clock / mode 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 clock / mode control circuit using software.

16.2.2 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 4). 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.

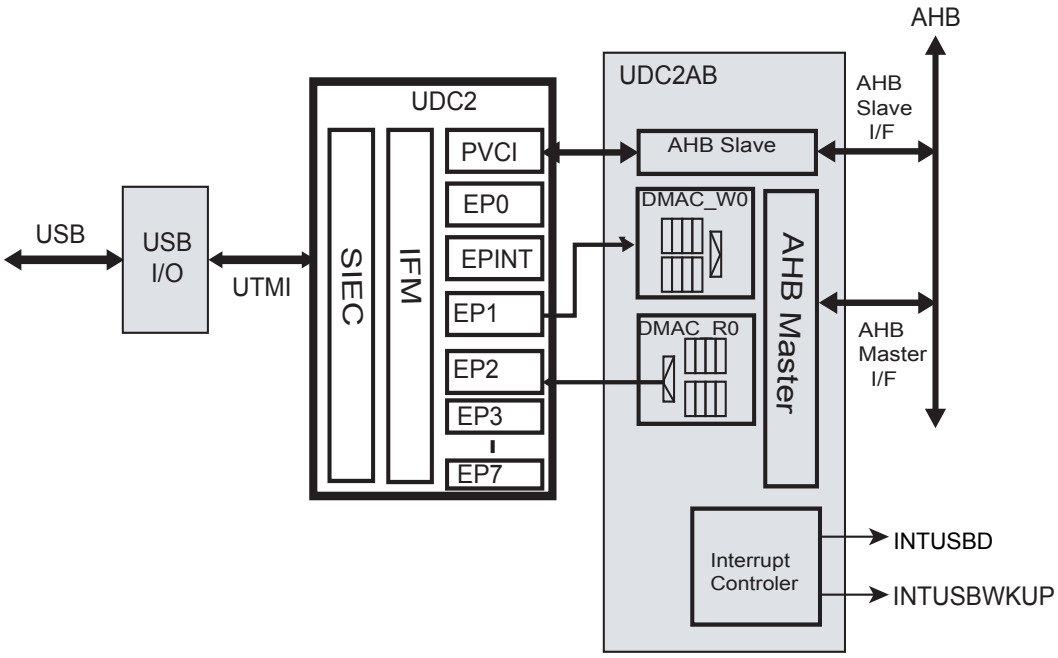


Figure 16-5 Block diagram of UDC2

16.2.2.1 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)

16.2.2.2 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.

16.2.2.3 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 USBDP 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 ex-

ists 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.

16.3 How to connect with the USB bus

The circuit below shows how to connect TMPM066/067/068FW with the USB bus.

Input the VBUS signal to VBUS pin to detect the connection of the USB power (VBUS).

The Pull-Up process using the pull-up resistor of the USBDP and the in-line damping resistor to the USBDM 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 DP and DM are unstable, we recommend to add a pull-down resistor to R_1.

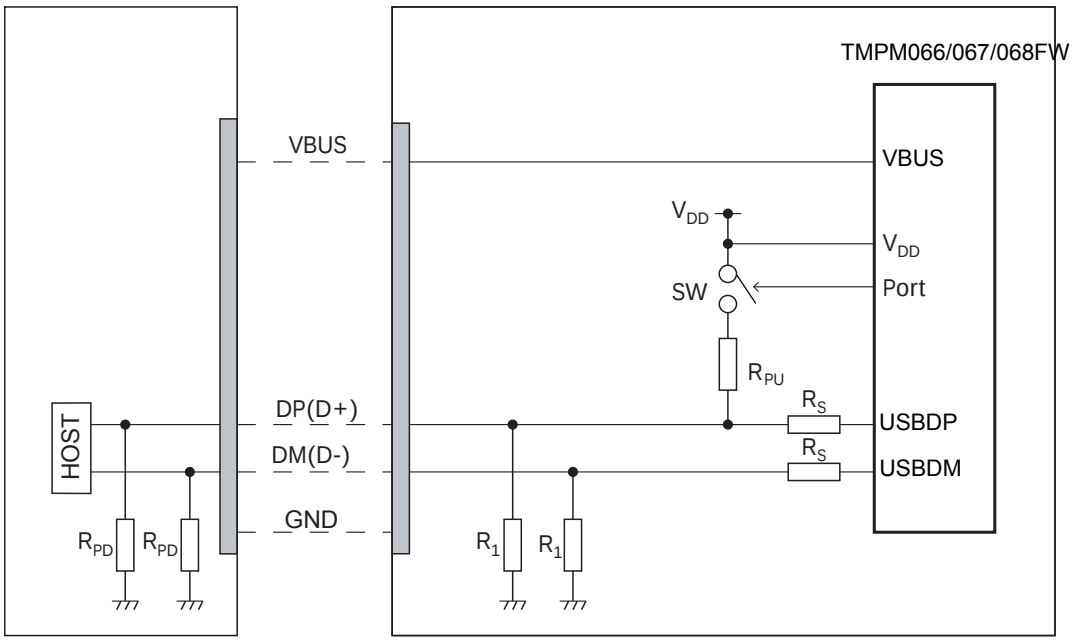


Figure 16-6 Connection example of the USB bus and TMPM066/067/068FW.

Note: R_1=500kΩ or higher (recommended value), R_S=33Ω (recommended value), R_{PU}=1.5kΩ (recommended value)

16.4 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.

16.4.1 UDC2AB Register

16.4.1.1 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.

16.4.1.2 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 UDFSMST-SET<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_re-set>. 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.

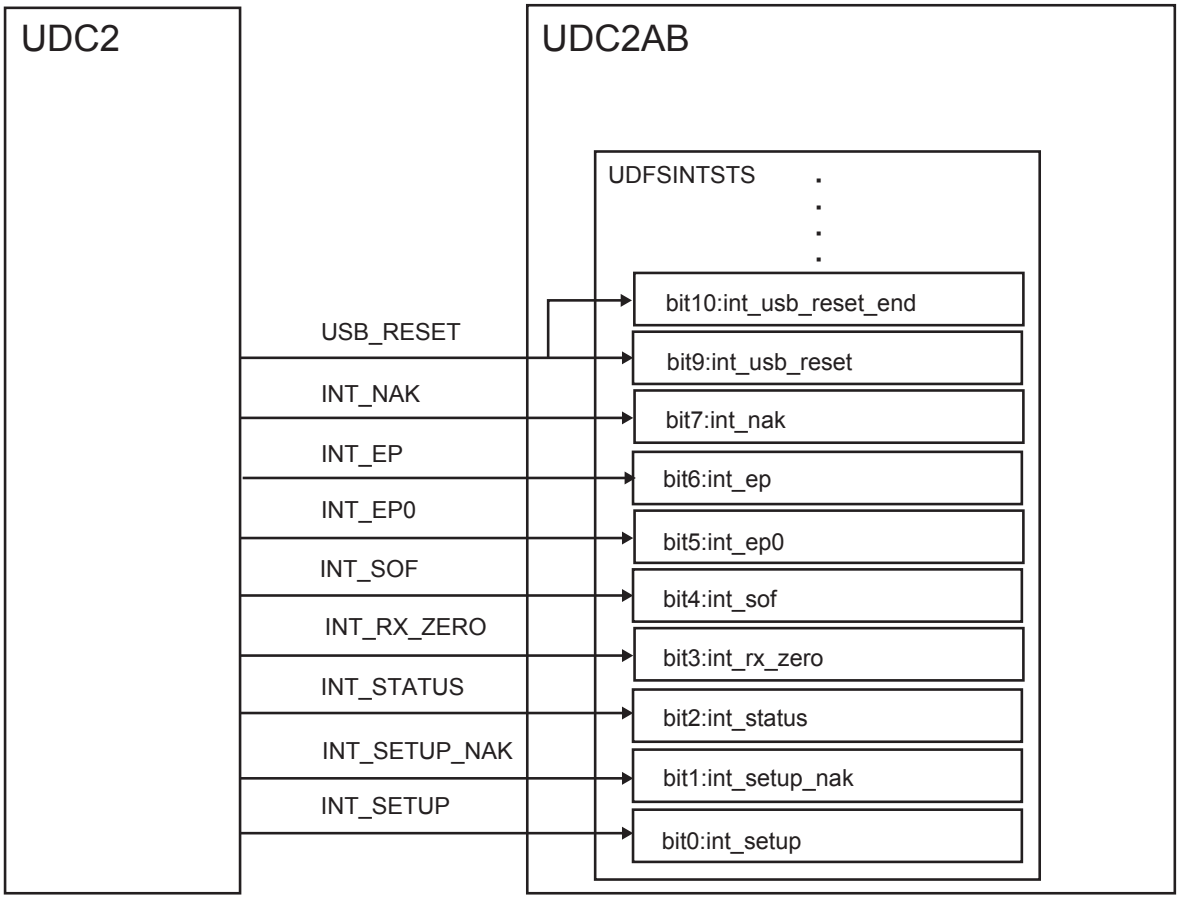


Figure 16-7 Connection between the flag output signals and interrupt bits.

16.4.1.3 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.

16.4.1.4 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

16.4.1.5 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"16.5.4.1 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.

16.4.1.6 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

16.4.1.7 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.

Note: Do not set "0" to the <dmardreg> and <dmardclr> at the same time.

Note: Before writing into UDC2 register, Make sure that the <dmardreg> and <dmardclr> is "0".

16.4.1.8 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.

16.4.1.9 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. (AHB error response will be occured.) 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 "16.4.1.1 UDC2AB Register list" for information about the register address of the UDC2. The offset addresses in the above register table (0x0200 to 0x0334) are reliant. Set it with the above udc2rdreq.
1-0	−	R	Read as undefined.

Note: Before writing into UDC2 register, Make sure that the <udc2rdreq> and <udc2drclr> is "0".

Note: Do not set "0" to the <udc2rdreq> and <udc2rdclr> at same time.

Note: When the <udc2rdreq>=1 or <udc2rdclr>=1, AHB error response will be occurred after writing into the UDC2 register.

16.4.1.10 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.

16.4.1.11 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.

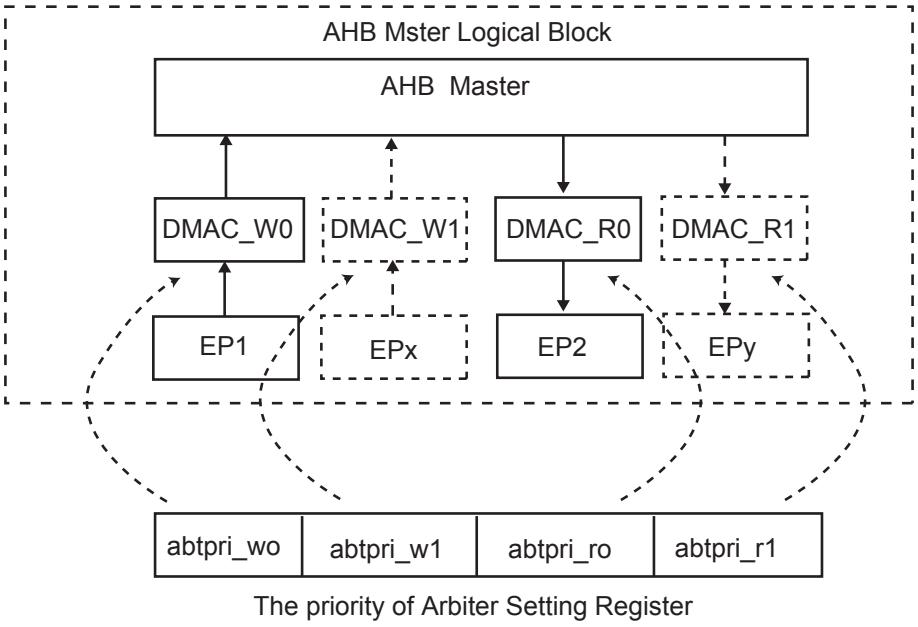


Figure 16-8 Relationship between DMAC and priority areas

16.4.1.12 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.

16.4.1.13 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.

16.4.1.14 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.

16.4.1.15 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.

16.4.1.16 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.

16.4.1.17 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.

16.4.1.18 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.

16.4.1.19 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.

16.4.1.20 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_re-mote_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 TMPM066/067/068FW 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 TMPM066/067/068FW to resuming from the low-power consumption mode using INTUSBWKUP. Refer to "16.5.7 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_remo-to_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 "16.5.7 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 "16.5.7 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 "16.5.1 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

16.4.1.21 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.

16.4.1.22 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.

16.4.2 UDC2 Register

16.4.2.1 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.

16.4.2.2 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 "16.4.2.1 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 clock/mode 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.

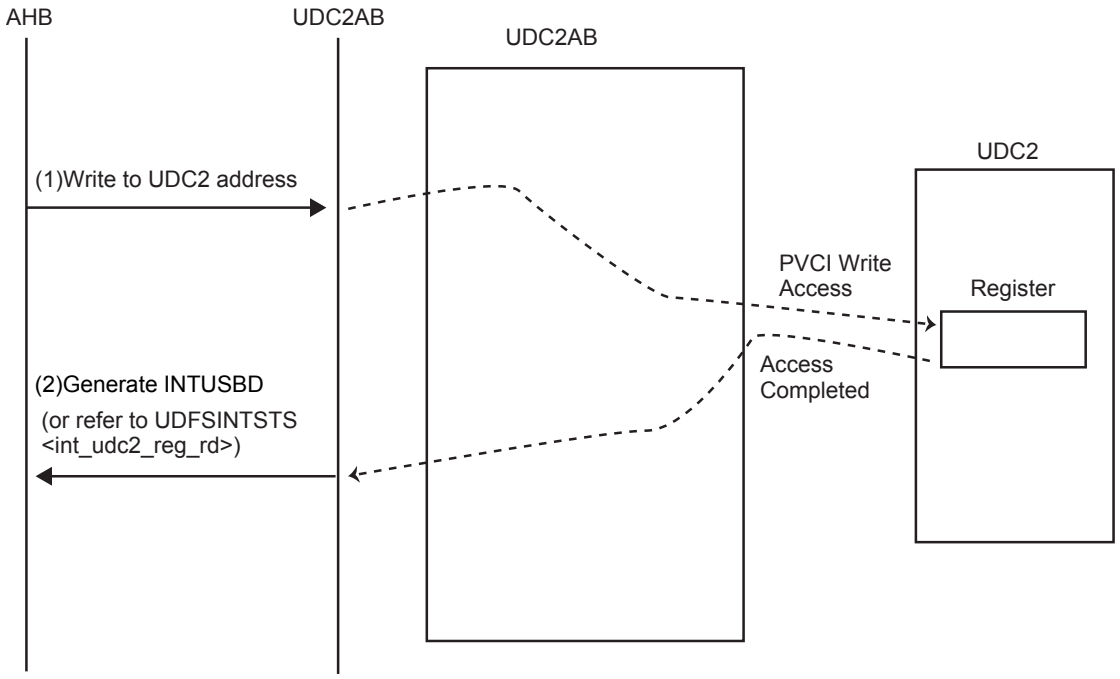


Figure 16-9 Write Access Flow Diagram of the UDC2 Register

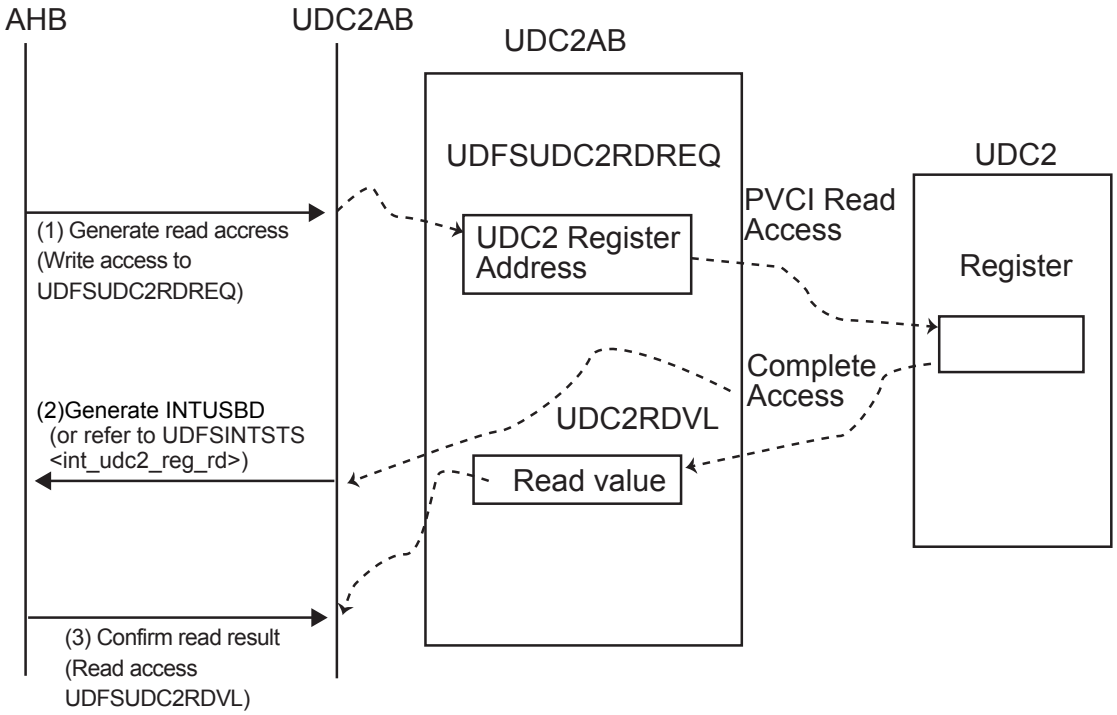


Figure 16-10 Read Access Flow Diagram of the UDC2 Register

16.4.2.3 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).

16.4.2.4 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.

16.4.2.5 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 "16.2.2.3 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

16.4.2.6 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

16.4.2.7 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).

16.4.2.8 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).

16.4.2.9 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).

16.4.2.10 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.

16.4.2.11 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.

16.4.2.12 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
UDFS2IN-TEPMSK	<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".

16.4.2.13 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.

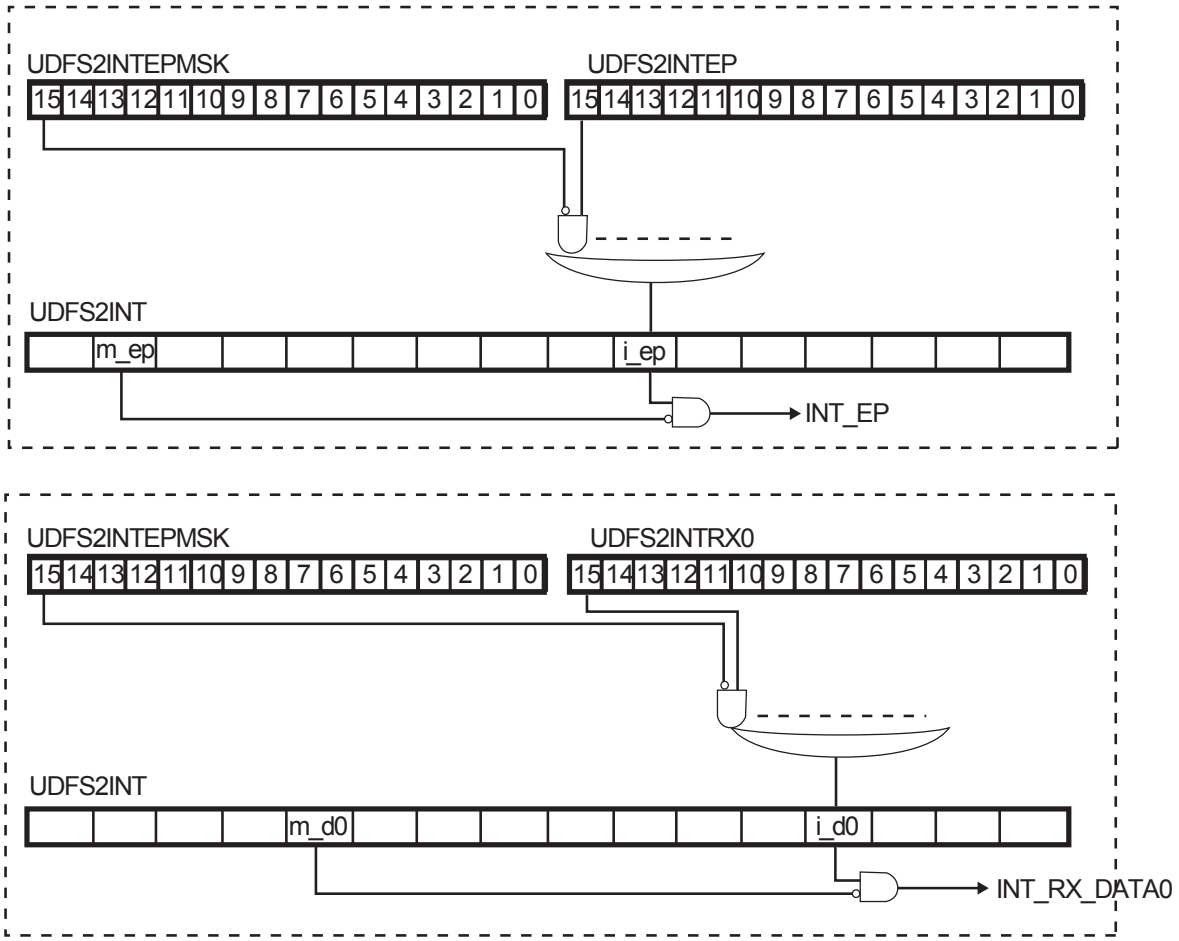


Figure 16-11 Interrupt Status and Mask Register

16.4.2.14 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.

16.4.2.15 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.

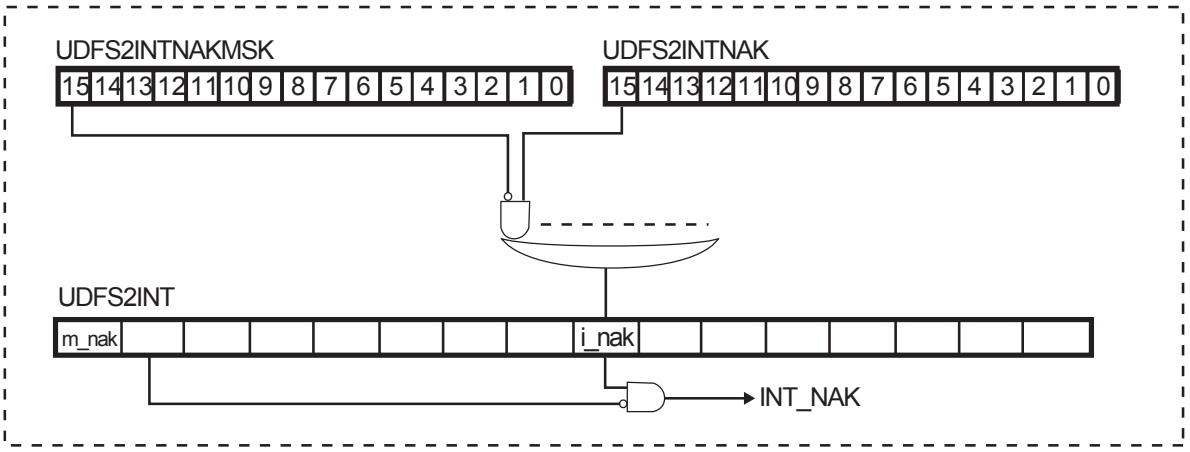


Figure 16-12 Interrupt and Status Register

16.4.2.16 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.

16.4.2.17 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.

16.4.2.18 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.

16.4.2.19 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 "16.7.1.1 Control-RD transfer" , "16.7.1.2 Control-WR transfer (without DATA-Stage)" and "16.7.1.3 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).

16.4.2.20 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 "16.9.2 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

16.4.2.21 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.

16.4.2.22 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

16.4.2.23 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

16.5 Description of UDC2AB operation

16.5.1 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 "16.4.1.1 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 "16.4.1.6 UDFSMSTSET(DMAC Setting Register)".

16.5.2 Interrupt Signals

There are two interrupt output signals of UDC2AB, INTUSB and INTUSBWKUP.

16.5.2.1 INTUSB Interrupt Signal

Interrupt output signal of INTUSB 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, INTUSB 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 INTUSB will not be asserted.

When the relevant bit of UDFSINTENB is set to "Enable" with UDFSINTSTS set, INTUSB will be asserted immediately after the setting is made.

Initial values for UDFSINTENB are all 0 (Disable).

Interrupt output signal of INTUSB will not be generated while CLK_H is stopped.

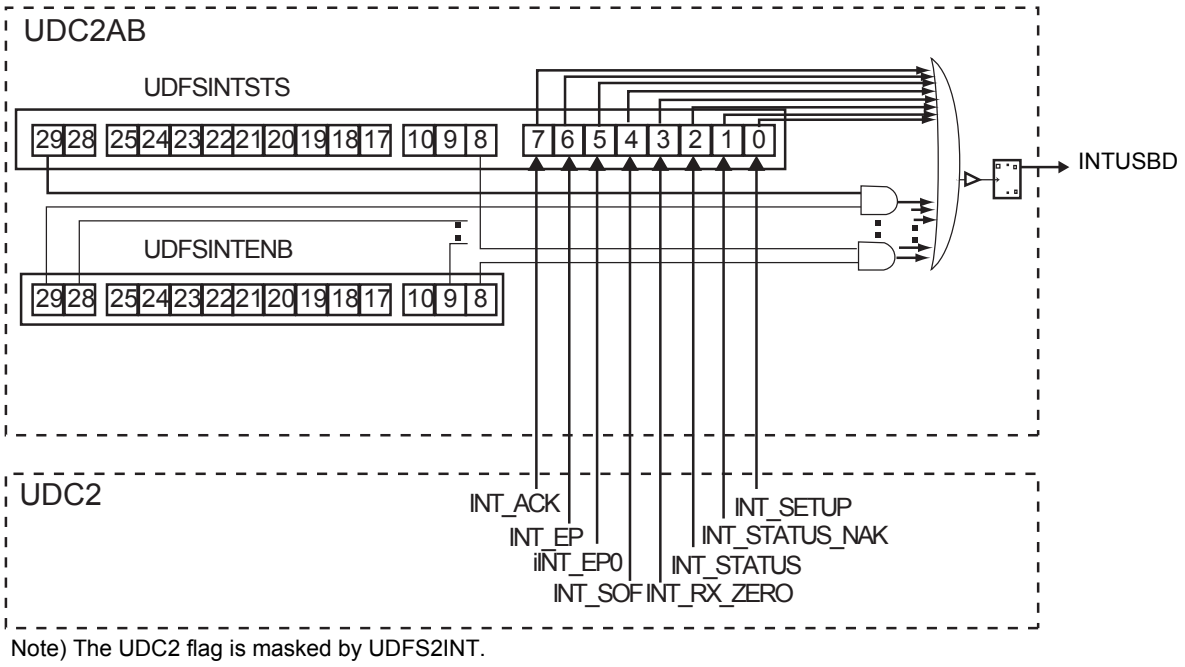


Figure 16-13 Relationship of INTUSB and registers

16.5.2.2 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.

16.5.3 Operation Sequence

The operation sequence of UDC2AB is as follows:

1. Hardware reset
2. Set the interrupt signal

   Configure the INTUSB interrupt, the INTUSBWKUP interrupt and the VBUS interrupt.
3. VBUS detection (connect) and reset

   Refer to "16.5.5.2 Sequence of USB Bus Power (VBUS) Connection/Disconnection" and "16.5.1 Reset" for details.
4. USB enumeration response

   Refer to "16.6 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 "16.5.4.1 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 "16.5.4.2 Master Write transfer" for details.
6. VBUS detection (disconnect)

   The USB bus power supply may be disconnected at any time.

   Refer to "16.5.5.2 Sequence of USB Bus Power (VBUS) Connection/Disconnection" for details.

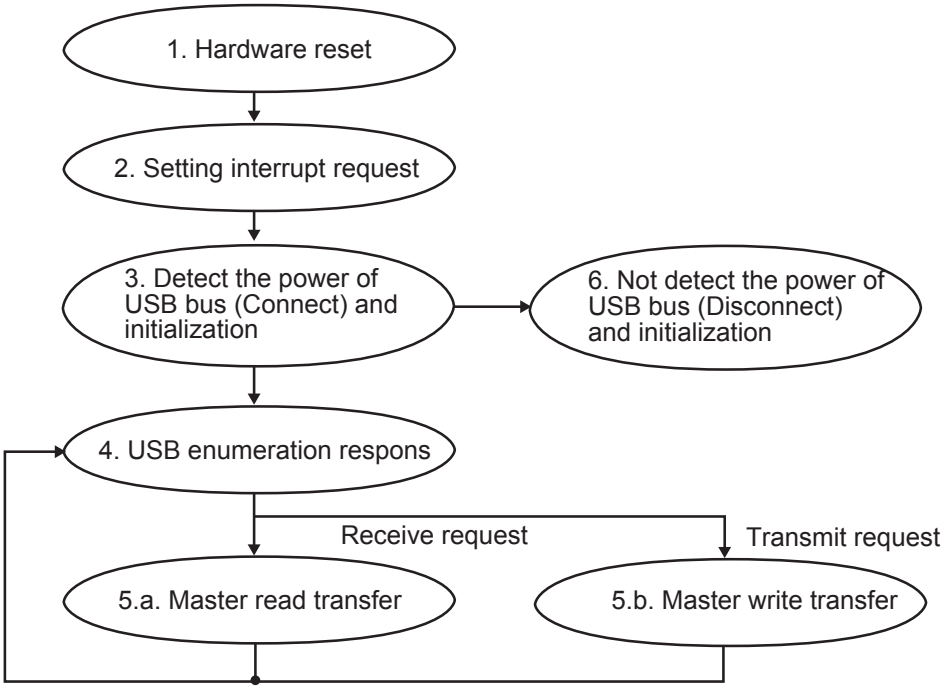


Figure 16-14 Operation Sequence

16.5.4 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."

16.5.4.1 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 "16.9.2 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.

16.5.4.2 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 16-15). 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.

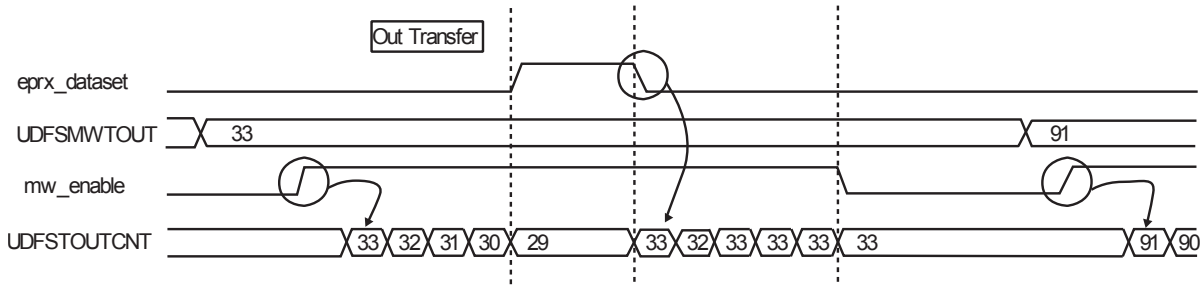


Figure 16-15 Example of MW timeout count

(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).

16.5.5 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.

16.5.5.1 Connection Diagram of Power Management Control Signal

Below is a connection diagram of signals related to power management control.

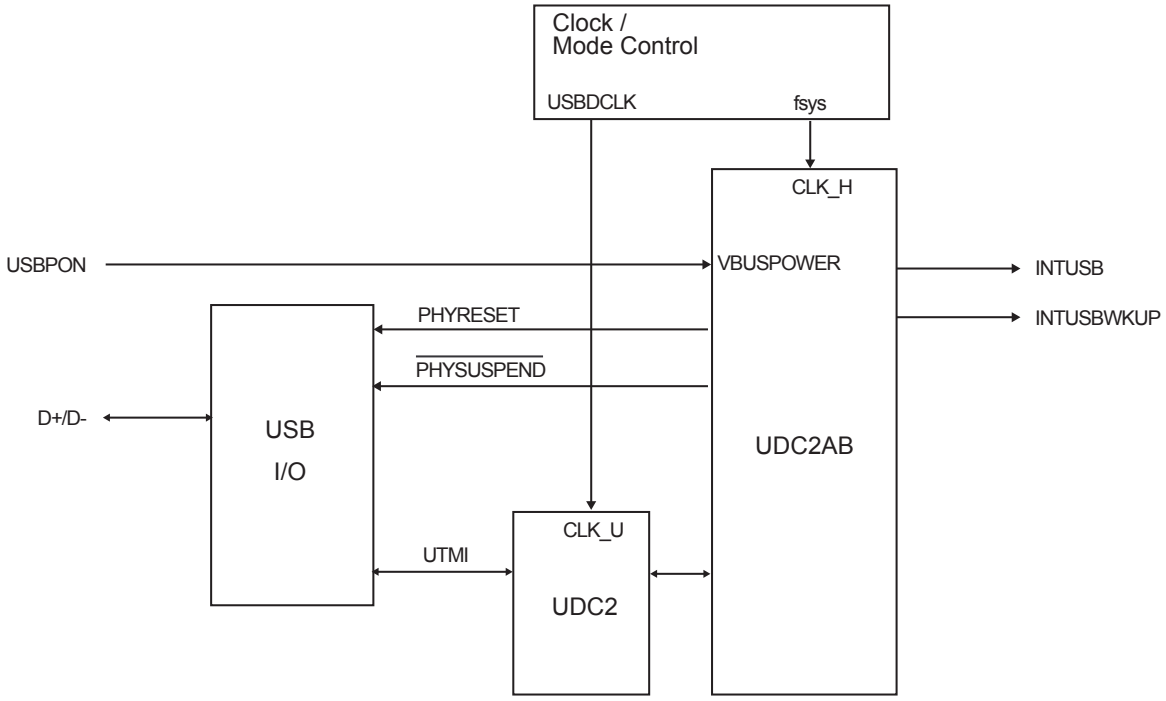


Figure 16-16 Connection Diagram of Power Management Control Signal

16.5.5.2 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 INTUSB(powerdetect) interrupt and UDFSPWCTL<pw_detect>. If UCLK_H is stopped, the USB power connection (VBUS) is detected using the External Dinterrupt 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 INTUSB (powerdetect) interrupt and UDFSPWCTL<pw_detect>. If CLK_H is stopped, the USB power (VBUS) disconnection is detected using the INTUSBWKUP interrupt.

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.

16.5.6 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.

16.5.7 Suspend / Resume

16.5.7.1 Shift to the suspended state

UDC2AB makes notification of detecting the suspended state of UDC2 by the INTUSB (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.

16.5.7.2 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.

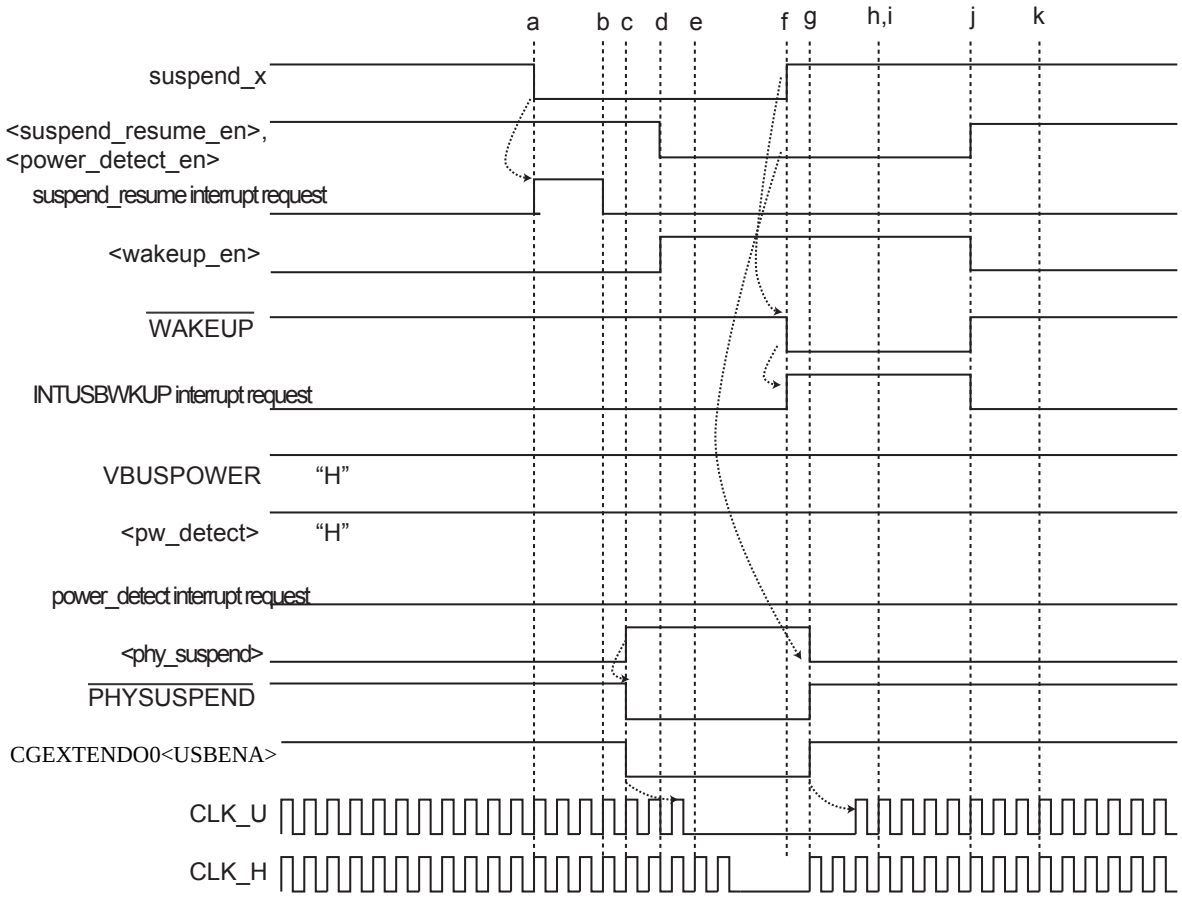


Figure 16-17 Signal operations when suspended and resumed (when CLK_H is stopped)

a. The suspend_x of the UDC2 is asserted to zero by detecting the suspend state on the USB bus, and the INTUSB(suspend_resume) interrupt occurs.
b. The service routine of the INTUSB(suspend_resume) interrupt clears the interrupt factor.
c. Set the UDFSPWCTL<phy_suspend> to "1". Setting the <phy_suspend> to "1" asserts the PHYSUSPEND output signal to "0".

   Zero clear the CGEXTENDO0<USBENA> of the clock/mode 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 INTUSB(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 CGEXTENDO0<USBENA> of the clock/mode 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 "16.5.7.3 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.

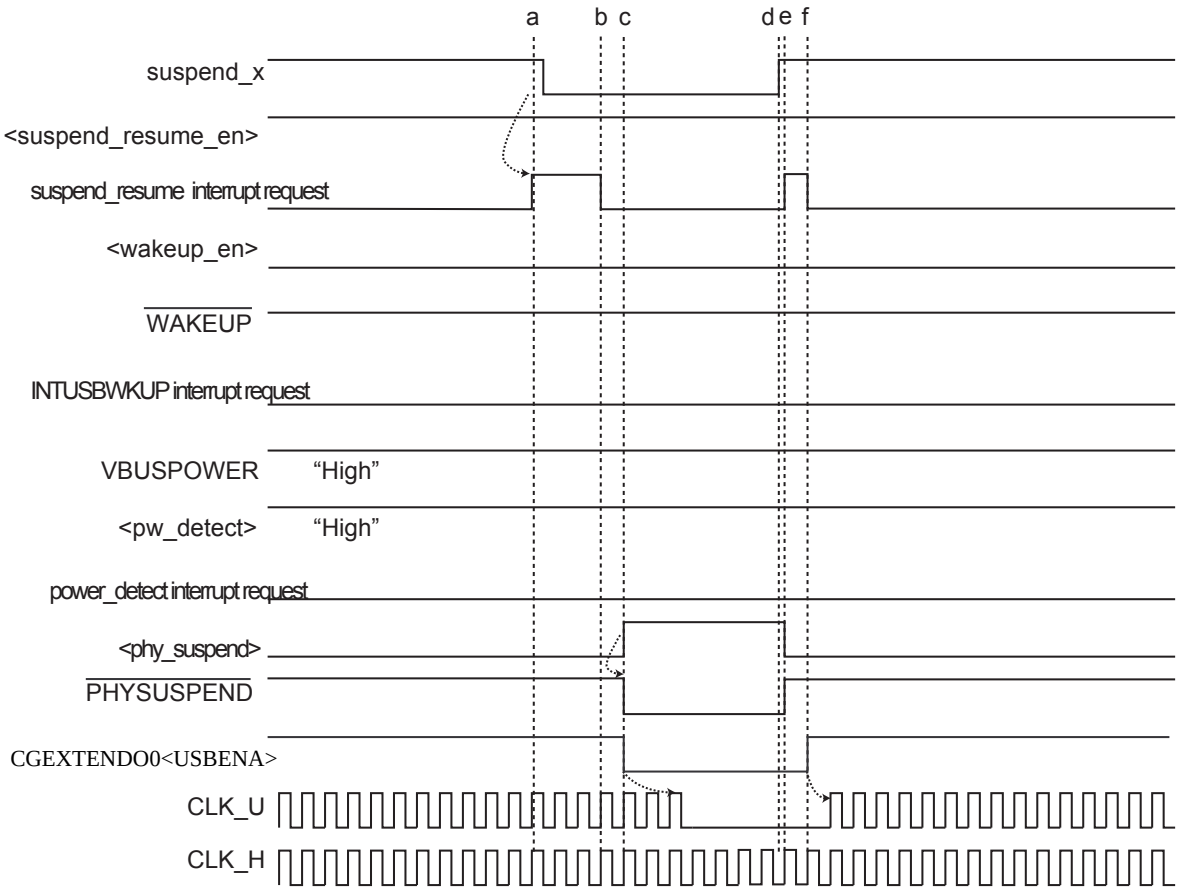


Figure 16-18 Operation of suspend/resume signals (to get the CLK_H work)

a. INTUSB(suspend_resume) interrupt occurs by detecting the suspended state on the USB bus.
b. Clears the interrupt source in the INTUSB(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 CGEXTENDO0<USBENA> of the clock/mode 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. INTUSB(suspend_resume) interrupt occurs.
f. Interrupt source is zero cleared in the service routine of the INTUSB(suspend_resume).

   Set CGEXTENDO0<USBENA> of the clock/mode 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.

16.5.7.3 Resuming from the suspend state (disconnect)

The procedures to resume from the suspended state (disconnection) and the signal change are shown as below.

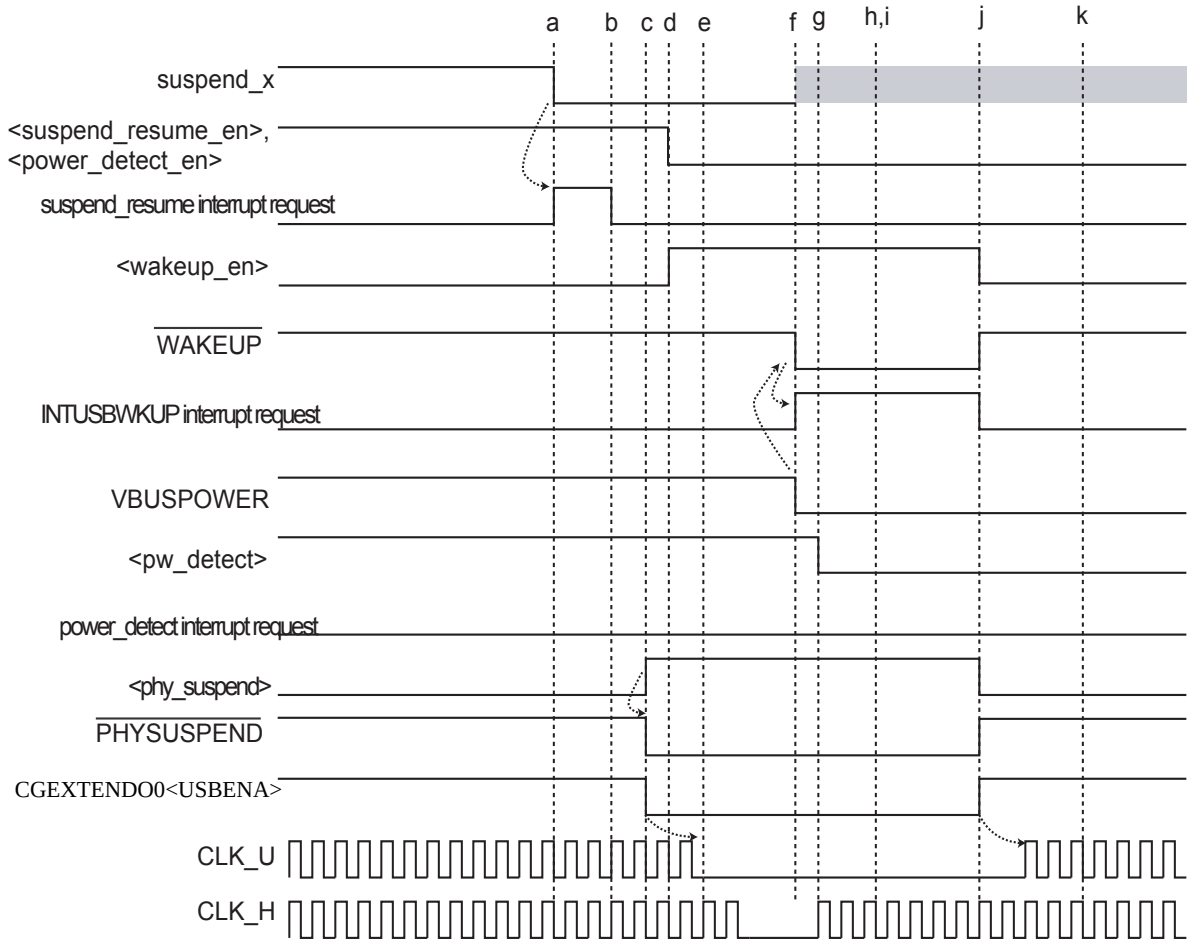


Figure 16-19 Operation of suspend/disconnect signals (to stop CLK_H)

a. 0 is asserted to the suspend_x of the UDC2 by detecting the suspend state on the USB bus and this generates the INTUSB(suspend_resume) interrupt.
b. Interrupt source is cleared by the service routine of the INTUSB(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 CGEXTENDO0<USBENA>of the clock/mode 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 INTUSB(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 "16.5.7.2 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 CGEXTENDO0<USBENA> of the clock/mode 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.

16.5.7.4 Remote wakeup from the suspended state

The procedure of remote wakeup from the suspended state and the signal change are shown below.

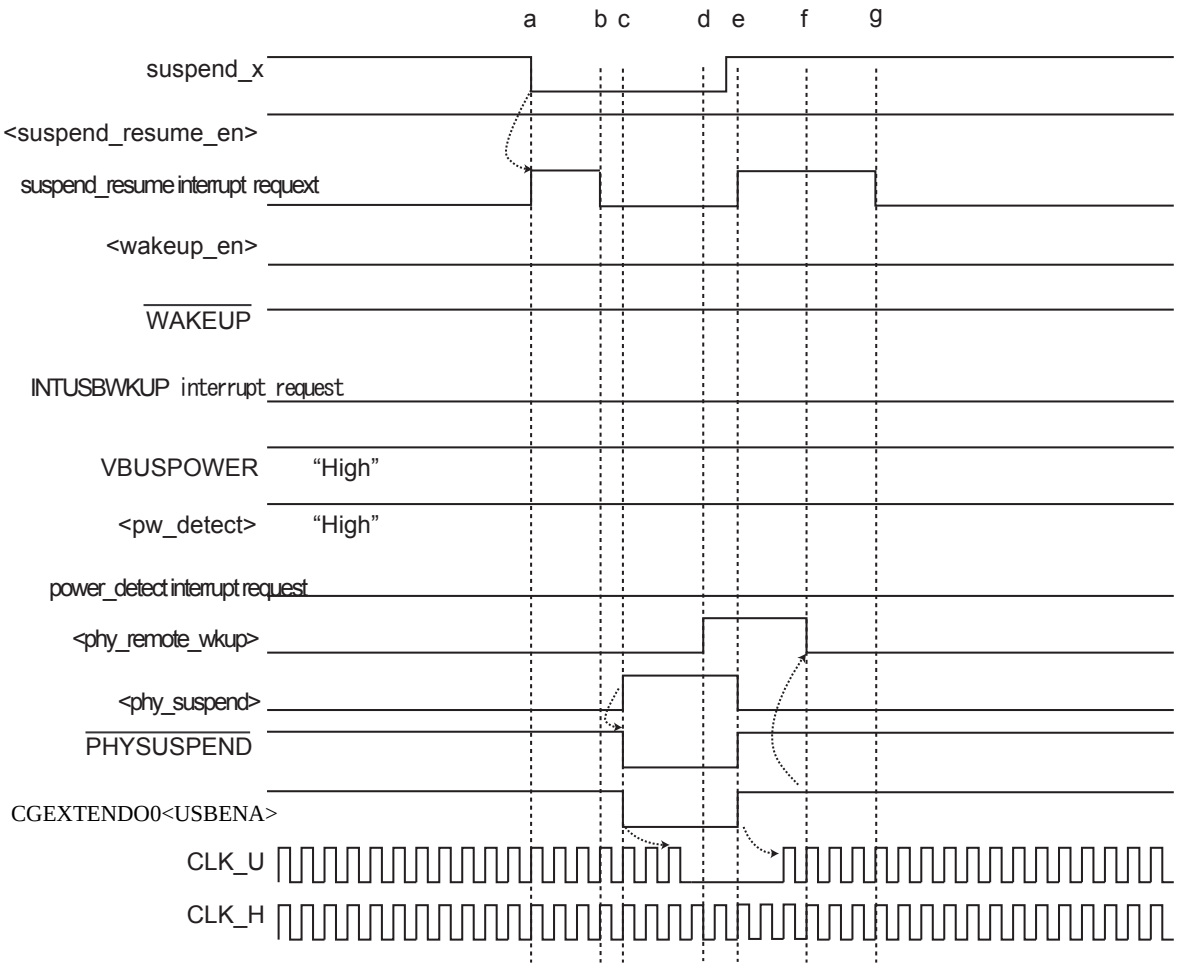


Figure 16-20 Operation of suspend/remote wakeup signals

a. suspend_x of the UDC2 is asserted to 0 by detecting the suspended state on the USB bus and the INTUSB(suspend_resume) interrupt occurs.

b. Clears the interrupt source in the service routine of the INTUSB(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 CGEXTENDO0<USBENA> of the clock/mode 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 INTUSB(suspend_resume) interrupt to occur and the PHYSUSPEND output signal to be deasserted to 1.
f. Set the CGEXTENDO0<USBENA> of the clock/mode 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.

16.6 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 USBDP 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 16-21 shows the "Device State Diagram".

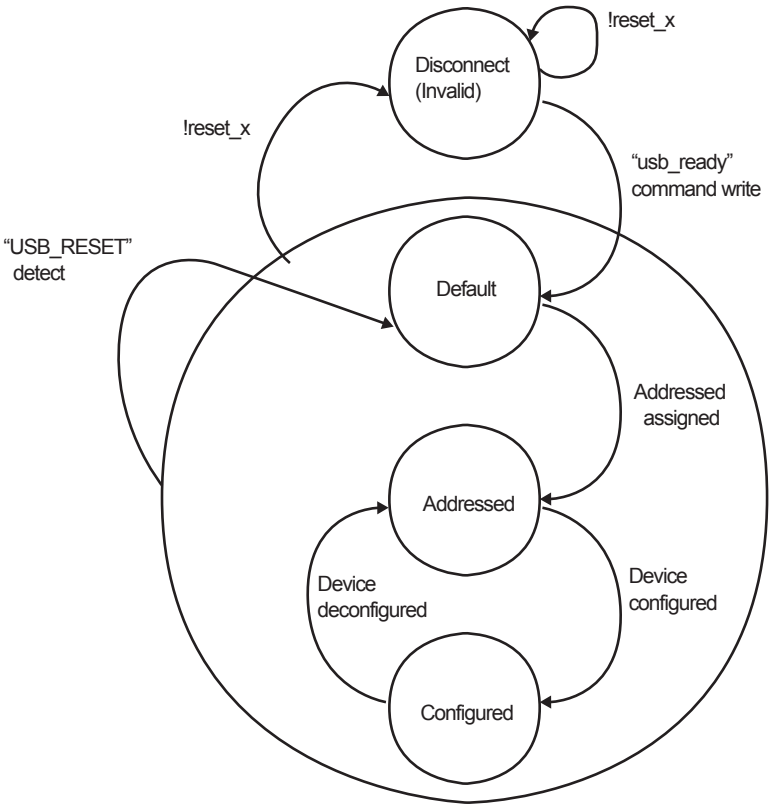


Figure 16-21 Device state diagram

16.7 Flow of Control in Transfer of EPs

16.7.1 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.

16.7.1.1 Control-RD transfer

The flow of control in Control-RD transfers is shown below.

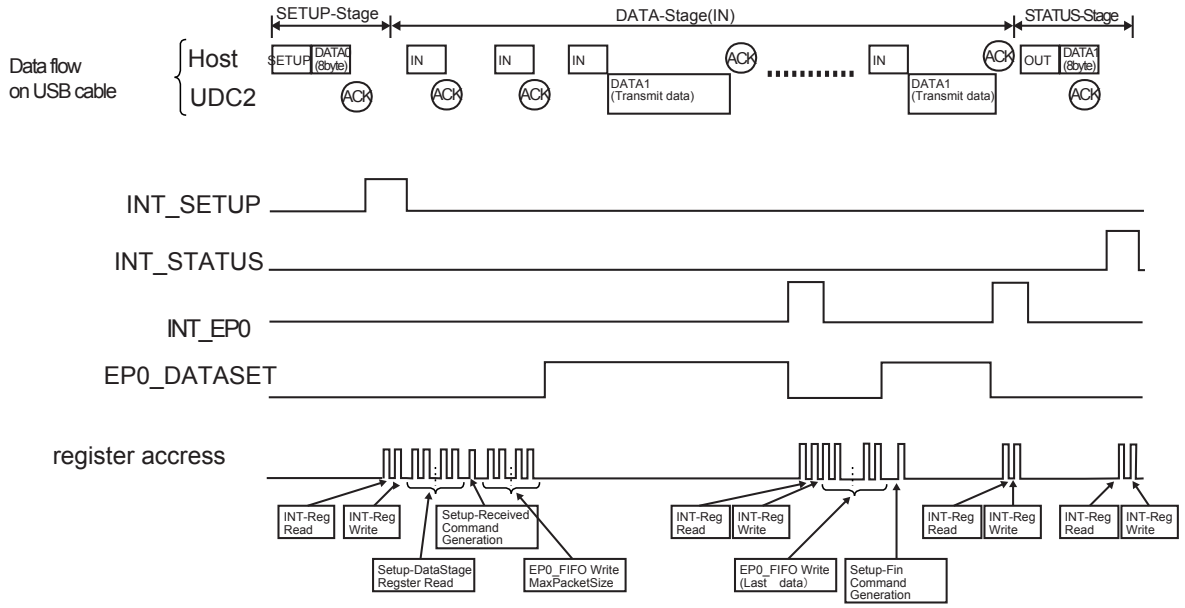


Figure 16-22 Flow of the control in Control-RD transfer

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.

16.7.1.2 Control-WR transfer (without DATA-Stage)

The flow of control in Control-WR transfer (without DATA-Stage) is shown below.

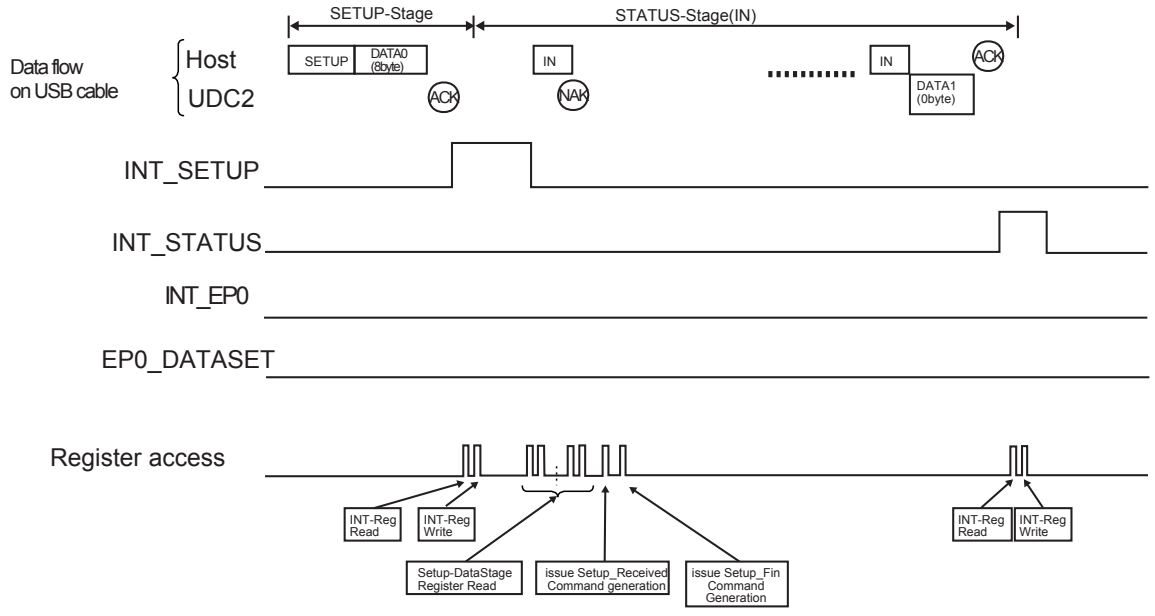


Figure 16-23 Flow of control in Control-WR transfer (without DATA-Stage)

(1) SETUP-Stage

Perform the same procedure described in "16.7.1.1 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 "16.7.1.1 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 "16.7.1.5 Processing when standard request".

16.7.1.3 Control-WR transfer (with DATA-Stage)

The flow of control in Control-WR transfer (with DATA-Stage) is shown below.

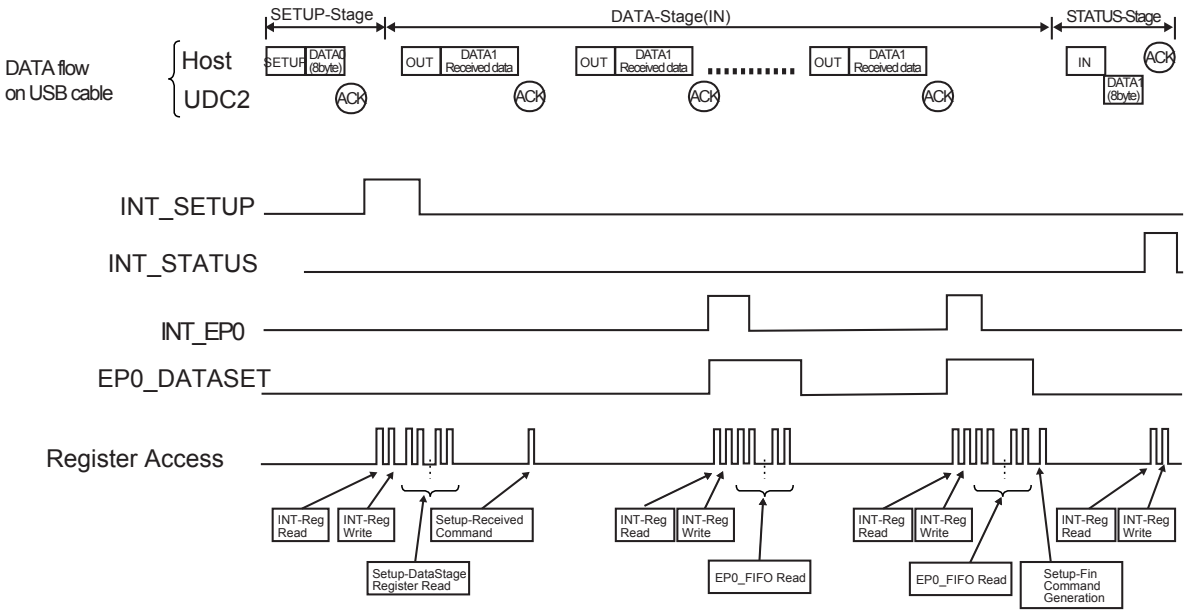


Figure 16-24 Flow of control in Control-WR transfers (with DATA-Stage)

(1) SETUP-Stage

To be processed in the same way of SETUP-Stage as described in"16.7.1.1 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 "16.7.1.1 Control-RD transfer".

16.7.1.4 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.)

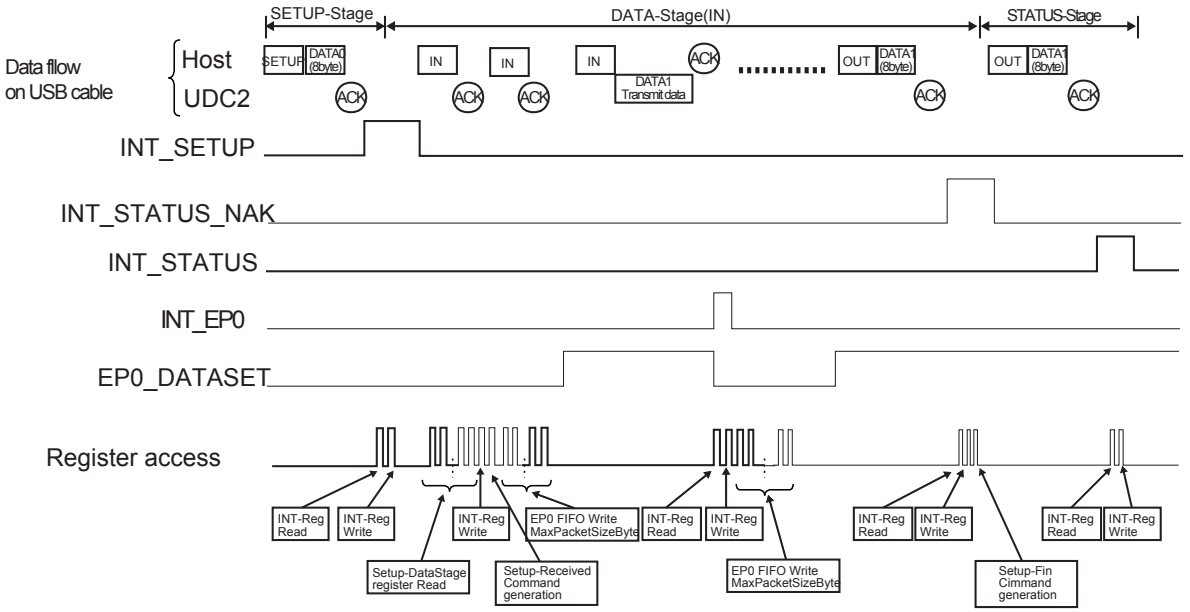


Figure 16-25 Example of using the INT_STATUS_NAK flag in Control-RD transfers

(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.

16.7.1.5 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 16.7.1.1 , 16.7.1.2 and 16.7.1.3.

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 "16.7.1.1 Control-RD transfer".

  Get Status Get Description Get Configuration

  Get Interface Get Frame
- Standard requests for "16.7.1.2 Control-WR transfer (without DATA-Stage)".

  Clear Feature Set Feature Set Address

  Set Configuration Set Interface
- Standard requests for "16.7.1.3 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 16-3) to UDFS2EP0FIFO.

<recipient> = Interface: Issue the EP-Stall command to EP0

<recipient> = EP : If wIndex=0(EP0), write the information on EP0 (Table 16-5) to UDFS2EP0FIFO. If wIndex≠0(EPx), issue the EP-Stall command to EP0.

- Configured state:

<recipient> = Device : Write the information on the device (Table 16-3) to UDFS2EP0FIFO.

<recipient> = Interface: If the interface specified by lwIndex, write the information on the interface (Table 16-4) to UDFS2EP0FIFO.

<recipient> = EP : If the EP specified by wIndex, write the information on the relevant EP(Table 16-5) to UDFS2EP0FIFO.

Table 16-3 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 16-4 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 16-5 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 "16.7.1.1 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 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.

- 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 Alternate Setting 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.

16.7.2 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.

16.8 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.

16.8.1 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 clock/mode control circuit.

16.8.2 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.

16.8.2.1 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.

16.8.2.2 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 clock/mode 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".

16.9 USB-Spec2.0 Device Controller Appendix

16.9.1 Appendix A System Power Management

In USB, operations related to the enumeration and power control signals (DP/DM) 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 DP/DM 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 DP/DM 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 USBDP/USBDM 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 DP/DM 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.

16.9.1.1 Connect / Disconnect Operations

(1) Connect Operation

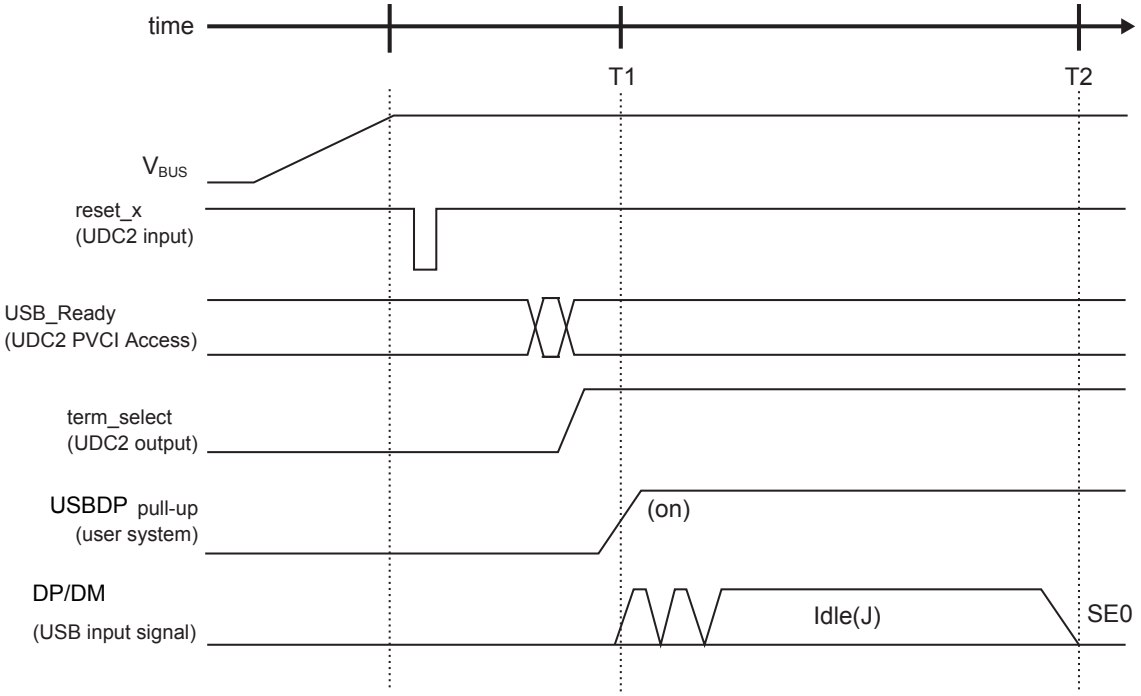


Figure 16-26 Connect operation timing

- 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 DP 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 DP 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.

16.9.1.2 Reset Operation

The "reset" here refers to the "Reset Signal" defined in the USB 2.0 Specification, not the system reset (reset_x) to UDC2.

(1) When Operation in FS Mode after Reset

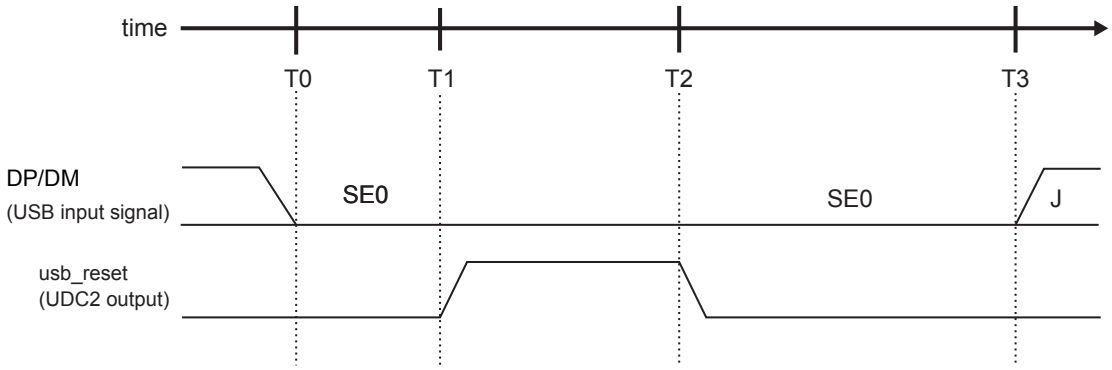


Figure 16-27 Reset Operation Timing

- 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 "16.4 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.

16.9.1.3 Suspend Operation

(1) Suspend operation

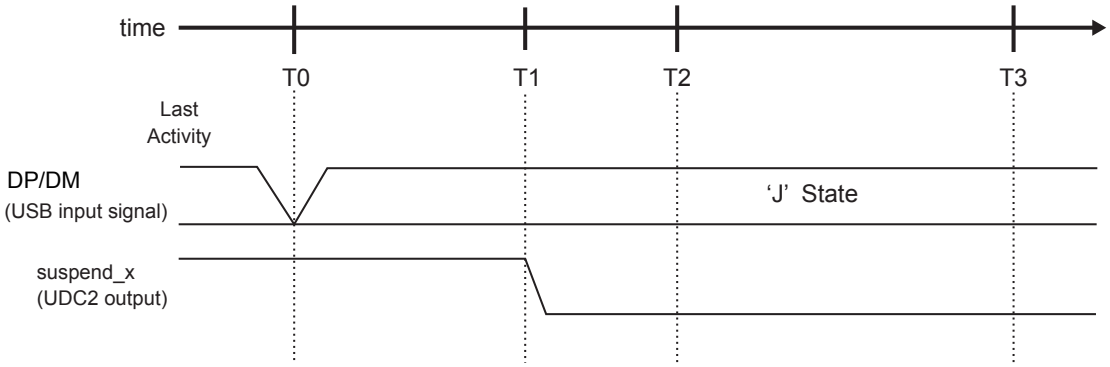


Figure 16-28 Suspend operation timing

- 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 Clock /mode 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.

16.9.1.4 Resume Operation

(1) Resume Operation by the Host

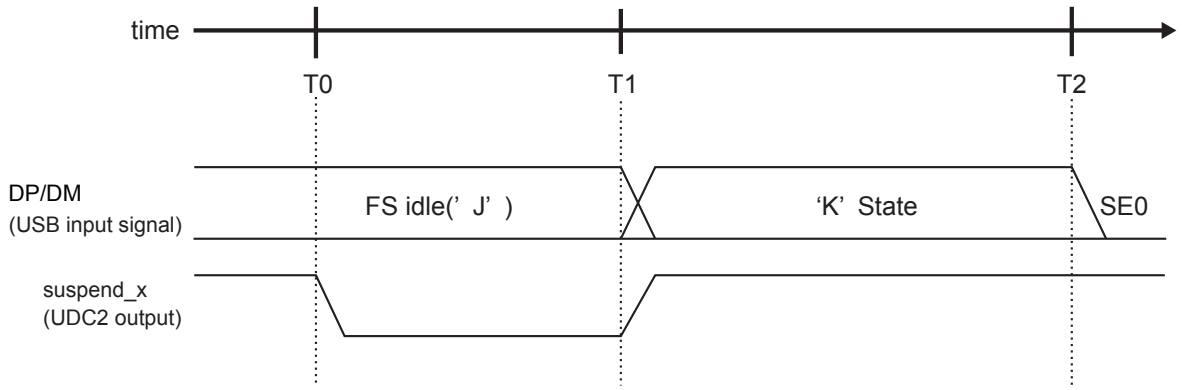


Figure 16-29 Resume operation timing by the host

- 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 clock/mode 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)

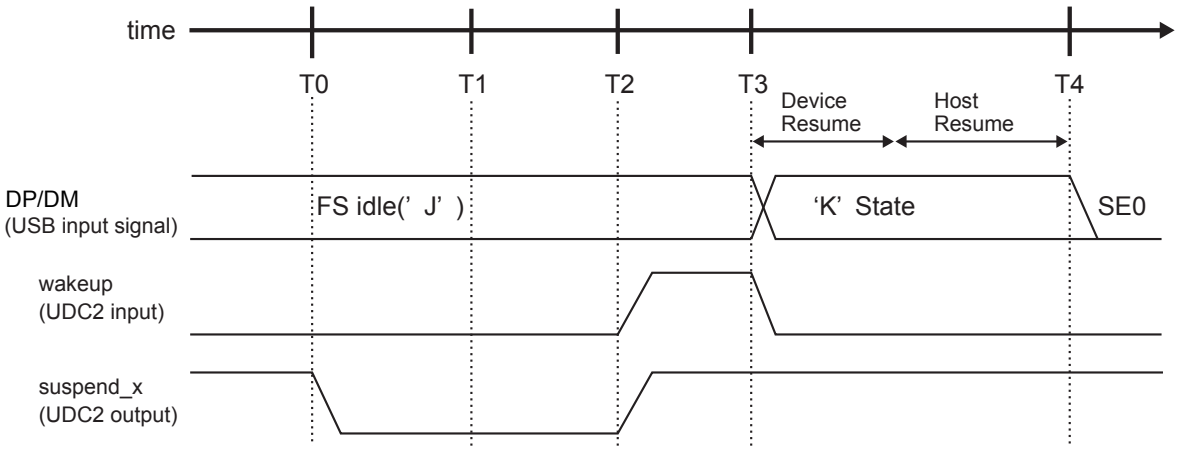


Figure 16-30 Remote wakeup operation timing

- 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 16.8 of the USB 2.0 Specification which offers detailed description.

16.9.2 Appendix B About Setting an Odd Number of Bytes as MaxPacketSize

16.9.2.1 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 16-6 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 16-6 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.

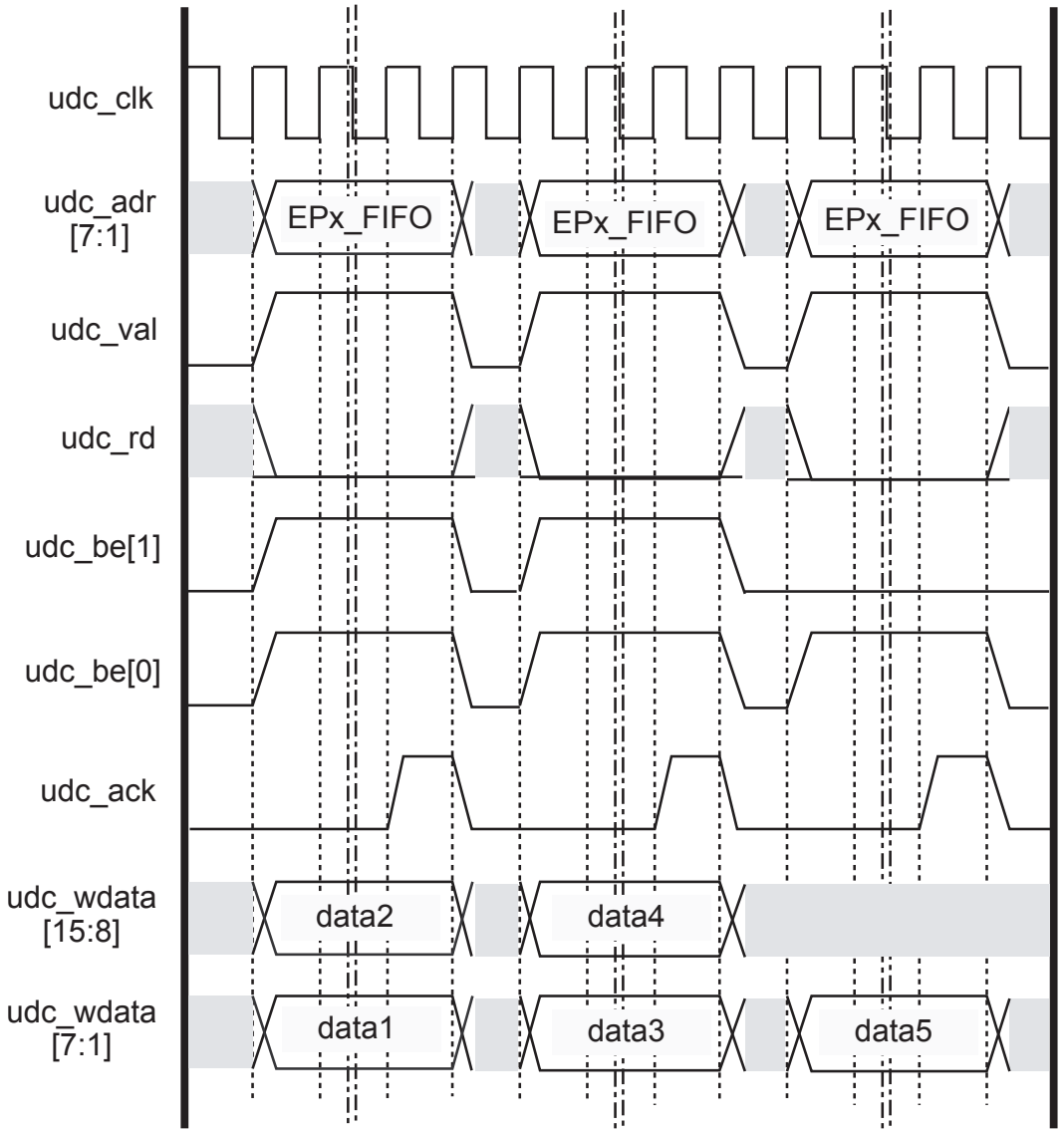


Figure 16-31 MPS write access with max_pkt = odd number (common bus access)

(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.

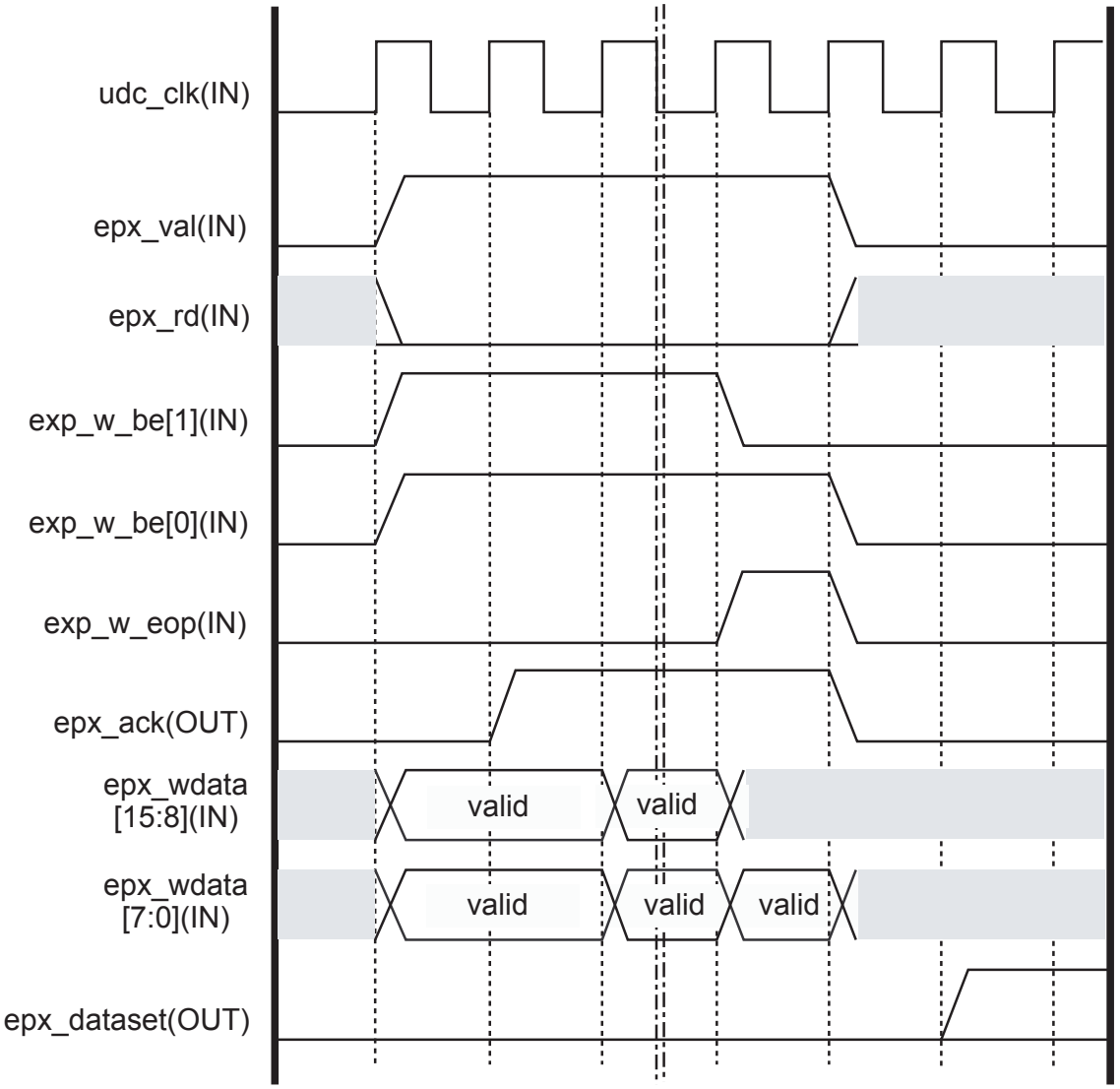


Figure 16-32 MPS (odd number) write access with max_pkt = even number (EP direct access)

16.9.3 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.

16.9.3.1 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.)

16.9.3.2 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).

17. 10-bit Analog/Digital Converter (ADC)

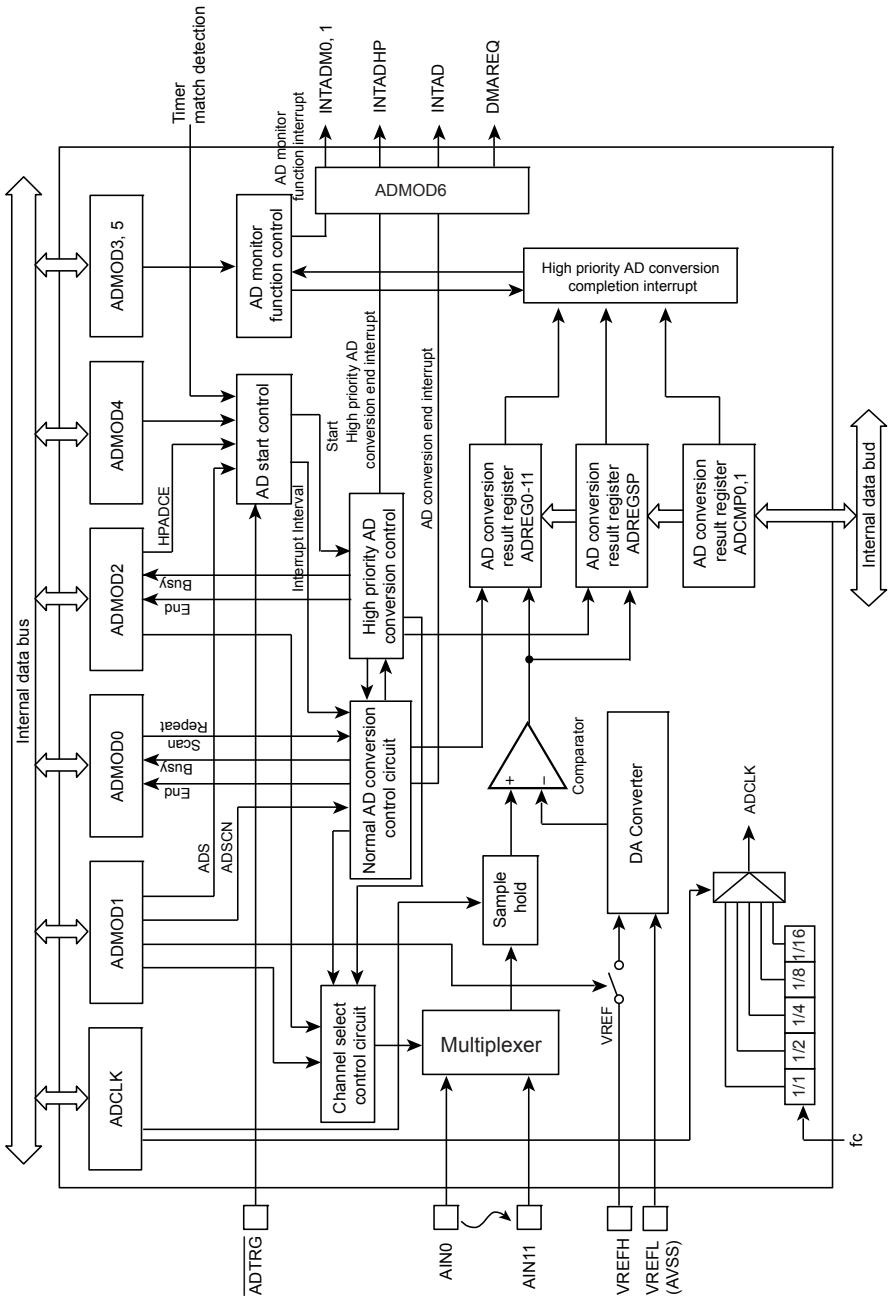
17.1 Outline

A 10-bit, sequential-conversion analog/digital converter (AD converter) is built into the TMPM066/067/068FW.

For details, refer to "Product information" to confirm usable channels and settings.

17.2 Configuration

Figure 17-1 shows the block diagram of this AD converter.



Note: VREFH and AVDD3 are shared. VREFL and AVSS are shared.

Figure 17-1 10-bit AD Converter Block Diagram

17.3 Registers

17.3.1 Register list

The control registers and addresses of the AD converter are as follows.

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

Register name		Address (Base+)
Conversion Clock Setting Register	ADCLK	0x0000
Mode Control Register 0	ADMOD0	0x0004
Mode Control Register 1	ADMOD1	0x0008
Mode Control Register 2	ADMOD2	0x000C
Mode Control Register 3	ADMOD3	0x0010
Mode Control Register 4	ADMOD4	0x0014
Mode Control Register 5	ADMOD5	0x0018
Mode Control Register 6	ADMOD6	0x001C
Conversion Result Register 0	ADREG0	0x0030
Conversion Result Register 1	ADREG1	0x0034
Conversion Result Register 2	ADREG2	0x0038
Conversion Result Register 3	ADREG3	0x003C
Conversion Result Register 4	ADREG4	0x0040
Conversion Result Register 5	ADREG5	0x0044
Conversion Result Register 6	ADREG6	0x0048
Conversion Result Register 7	ADREG7	0x004C
Conversion Result Register 8	ADREG8	0x0050
Conversion Result Register 9	ADREG9	0x0054
Conversion Result Register 10	ADREG10	0x0058
Conversion Result Register 11	ADREG11	0x005C
Conversion Result Register SP	ADREGSP	0x0060
Conversion Result Comparison Register 0	ADCMP0	0x0064
Conversion Result Comparison Register 1	ADCMP1	0x0068

17.3.2 ADCLK (Conversion Clock Setting Register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-4	ADCC[1:0]	R/W	Select the AD conversion clock count 00: 35.5 conversion clock 01: 42 conversion clock 10: 68 conversion clock 11: 81 conversion clock
5-4	-	R	Read as 0.
3-0	ADCLK[3:0]	R/W	Select the AD conversion clock (Note1) (Note2) 0000: fc 1000:fc/6 0001: fc/2 1001:fc/12 0010: fc/4 1010:fc/24 0011: fc/8 1011:fc/48 0100: fc/16 1100:fc/96 0101 - 0111: Reserved 0101--0111 & 1101 - 1111:Reserved

Note 1: Do not change the setting of the AD conversion clock during AD conversion.
Note 2: AD conversion clock setting by ADCLK is less than or equal to fsys (system clock).(ADCLK ≤ fsys)

A clock count is required to satisfy the condition that described below.

VREFH AVDD	Conversion time
2.7 to 3.6V	16.2 μs or longer
1.8 to 3.6V	32.4μs or longer

17.3.3 ADMOD0 (Mode Control Register 0)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	EOCFN	R	Normal AD conversion completion flag (note1) 0: Before or during conversion 1: Completion This bit is "0" cleared when it is read.
6	ADBFN	R	Normal AD conversion BUSY flag 0: Conversion stop 1: During conversion
5	−	R	Read as 0.
4-3	ITM[1:0]	R/W	Interrupt in fixed channel repeat conversion mode 00: Generate in interrupt once every single conversion 01: Generate interrupt once every 4 conversions 10: Generate interrupt once every 8 conversions 11: Setting prohibited It is valid only when it's specified in the fixed channel repeat mode (<REPEAT> ="1", <SCAN> = "0").
2	REPEAT	R/W	Specify repeat mode 0: Single conversion mode 1: Repeat conversion mode
1	SCAN	R/W	Specify scan mode 0: Fixed channel mode 1: Channel scan mode If channel scan mode is selected, set the channel number to ADMOD1<ADSCAN>.
0	ADS	W	Start AD conversion start 0: Don't care 1: Start conversion Conversion must be started after setting the mode. "0" is always read.

17.3.4 ADMOD1 (Mode Control Register 1)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	VREFON	I2AD	ADSCN		ADCH			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	VREFON	R/W	VREF application control (Note) 0: OFF 1: ON
6	I2AD	R/W	Specify operation mode in IDLE mode 0: Stop 1: Operation
5-4	ADSCN[1:0]	R/W	Specify operation mode in channel scan mode 00: 4-channel scan 01: 8-channel scan 10: 12-channel scan 11: Reserved Specify operation mode when channel scan mode is selected by ADMOD0<SCAN>. The conversion channel is selected by setting of <ADCH>. Refer to the below table.
3-0	ADCH[3:0]	R/W	Select analog input channel (Refer to the below table.)

Note: **Before starting AD conversion, write "1" to the <VREFON> bit, wait for 3μs during which time the internal reference voltage should stabilize, and then write "1" to the ADMOD0<ADS>.**

Selection of analog input channel

		<ADCH[3:0]>							
		0000	0001	0010	0011	0100	0101	0110	0111
ADMOD0 <SCAN>=0	Fixed channel	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7
ADMOD0 <SCAN>=1	<ADSCN>=00 4-channel scan	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN4	AIN4~ AIN5	AIN4~ AIN6	AIN4~ AIN7
	<ADSCN>=01 8-channel scan	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN0~ AIN4	AIN0~ AIN5	AIN0~ AIN6	AIN0~ AIN7
	<ADSCN>=10 12-channel scan	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN0~ AIN4	AIN0~ AIN5	AIN0~ AIN6	AIN0~ AIN7

		<ADCH[3:0]>							
		1000	1001	1010	1011	1100	1101	1110	1111
ADMOD0 <SCAN>=0	Fixed channel	AIN8	AIN9	AIN10	AIN11	AIN12	AIN13	AIN14	AIN15
ADMOD0 <SCAN>=1	<ADSCN>=00 4-channel scan	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN4	AIN4~ AIN5	AIN4~ AIN6	AIN4~ AIN7
	<ADSCN>=01 8-channel scan	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN0~ AIN4	AIN0~ AIN5	AIN0~ AIN6	AIN0~ AIN7
	<ADSCN>=10 12-channel scan	AIN0~ AIN8	AIN0~ AIN9	AIN0~ AIN10	AIN0~ AIN11	-	-	-	-

17.3.5 ADMOD2 (Mode Control Register 2)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	EOCFHP	R	Top-priority AD conversion completion flag (Note) 0: Before or during conversion 1: Completion
6	ADBFHP	R	Top-priority AD conversion BUSY flag 0: During conversion halts 1: During conversion
5	HPADCE	R/W	Activate top-priority conversion 0: Don't care 1: Start conversion "0" is always read.
4	-	R/W	Write "0".
3-0	HPADCH[3:0]	R/W	Select analog input channel when activating top-priority conversion. (See the table below)

Note: This bit is "0" cleared when it is read.

Selection of analog input channel

HPADCH[3:0]	0000	0001	0010	0011	0100	0101	0110	0111
Conversion channel	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7

HPADCH[3:0]	1000	1001	1010	1011	1100	1101	1110	1111
Conversion channel	AIN8	AIN9	AIN10	AIN11	AIN12	AIN13	AIN14	AIN15

17.3.6 ADMOD3 (Mode Control Register 3)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	-	-	ADOBIC0	ADREGS0				ADOBSV0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	−	R/W	Write "0".
6	−	R	Read as 0.
5	ADOBIC0	R/W	Set the AD monitor function interrupt 0 0: If the value of the conversion result is smaller than the comparison register 0, an interrupt is generated. 1: If the value of the conversion result is bigger than the comparison register 0, an interrupt is generated.
4-1	ADREGS0[3:0]	R/W	Select a target conversion result register when using the AD monitor function 0 (See the below table).
0	ADOBSV0	R/W	AD monitor function 0 0: Disable 1: Enable

<ADREGS0[3:0]>	Conversion result register to be compared	<ADREGS0[3:0]>	Conversion result register to be compared
0000	ADREG0	0100	ADREG4
0001	ADREG1	0101	ADREG5
0010	ADREG2	0110	ADREG6
0011	ADREG3	0111	ADREG7
-	-	1xxx	ADREGSP

17.3.7 ADMOD4 (Mode Control Register 4)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	HADHS	R/W	H/W source for activating top-priority AD conversion 0: External trigger 1: Match with timer register (Note 1)
6	HADHTG	R/W	H/W for activating top-priority AD conversion 0: Disable 1: Enable
5	ADHS	R/W	H/W source for activating normal AD conversion (note2) 0: External trigger 1: Match with timer register (Note 1)
4	ADHTG	R/W	HW for activating normal AD conversion 0: Disable 1: Enable
3-2	−	R	Read as 0.
1-0	ADRST[1:0]	W	Overwriting 10 with 01 allows ADC to be software reset.(note 3)

Note 1: For details, refer to "Product information" to confirm H/W source.

Note 2: The external trigger cannot be used for H/W activation of AD conversion when it is used for H/W activation of top priority AD conversion.

Note 3: A software reset initializes all the registers except for ADCLK<ADCLK>.

17.3.8 ADMOD5 (Mode Control Register 5)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	-	-	ADOBIC1	ADREGS1				ADOBSV1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-6	–	R	Read as 0.
5	ADOBIC1	R/W	Set the AD monitor function interrupt 1. 0: If the value of the conversion result is smaller than the comparison register 1, an interrupt is generated. 1: If the value of the conversion result is bigger than the comparison register 1, an interrupt is generated.
4-1	ADREGS1[3:0]	R/W	Select a target conversion result register when using the AD monitor function 1 (See the below table).
0	ADOBSV1	R/W	AD monitor function 1 0: Disable 1: Enable

<ADREGS1[3:0]>	Conversion result register to be compared	<ADREGS1[3:0]>	Conversion result register to be compared
0000	ADREG0	0100	ADREG4
0001	ADREG1	0101	ADREG5
0010	ADREG2	0110	ADREG6
0011	ADREG3	0111	ADREG7
–	–	1xxx	ADREGSP

17.3.9 ADMOD6 (Mode Control 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	-	-	-	-	ADM1DMA	ADM0DMA	ADHPDMA	ADDMA
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as 0.
3	ADM1DMA	R/W	Specify AD monitor function 1 DMA activation factor.(Triggered by INTADM1) 0: Disable 1: Enable
2	ADM0DMA	R/W	Specify AD monitor function 0 DMA activation factor.(Triggered by INTADM0) 0: Disable 1: Enable
1	ADHPDMA	R/W	Specify top-priority AD conversion DMA activation factor.(Triggered by INTADHP) 0: Disable 1: Enable
0	ADDMA	R/W	Specify normal AD conversion DMA activation factor.(Triggered by INTAD) 0: Disable 1: Enable

17.3.10 ADREGn (Conversion Result Register n: n = 0 to 11)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADRn							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADR0		-	-	-	-	OVRn	ADRnRF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADRn[9:0]	R	AD conversion result Conversion result is stored. For information about the correlation between the conversion channel and the conversion result register, refer to the Table 17-2 and Table 17-3, "17.4.5.7 Interrupt generation timings and AD conversion result storage register".
5-2	−	R	Read as 0.
1	OVRn	R	Overrun flag 0: Not generated 1: Generated If the conversion result is overwritten before reading <ADR0>, "1" is set. This bit is "0" cleared when it is read.
0	ADRnRF	R	AD conversion result storage flag 0: Conversion result is not stored 1: Conversion result is stored. If a conversion result is stored, "1" is set. This bit is "0" cleared when the conversion result is read.

Note: Access to this register must be a half word or a word access.

17.3.11 ADREGSP (AD Conversion Result Register SP)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADRSP[9:0]	R	AD conversion result. Top-priority AD conversion result is stored
5-2	−	R	Read as 0
1	OVRSP	R	Overrun flag 0: Not generated 1: Generated If a conversion result is overwritten before reading <ADRSP>, "1" is set. This bit is "0" cleared when it is read.
0	ADRSPRF	R	AD conversion result storage flag 0: Conversion result is not stored. 1: Conversion result is stored. If a conversion result is stored, "1" is set. This bit is "0" cleared when the conversion result is read.

Note: Access to this register must be a half word or a word access.

17.3.12 ADCMP0 (AD Conversion Result Comparison Register 0)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADCOM0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADCOM0		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADCOM0[9:0]	R/W	When AD monitor function 0 is enabled, it sets a value to be compared with the value of the conversion result register specified by ADMOD3<ADREGS0>.
5-0	−	R	Read as 0.

Note: **To write values into this register, the AD monitor function 0 must be disabled (ADMOD3<ADBSV0> ="0").**

17.3.13 ADCMP1 (AD Conversion Result Comparison Register 1)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADCOM1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADCOM1		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADCOM1[9:0]	R/W	When AD monitor function 1 is enabled, it sets a value to be compared with the value of the conversion result register specified by ADMODt<ADREGS1>.
5-0	−	R	Read as 0.

Note: **To write values into this register, the AD monitor function 1 must be disabled (ADMOD5<ADBSV1>="0").**

17.4 Description of Operations

17.4.1 Analog Reference Voltage

The "High" level of the analog reference voltage shall be applied to the VRFEH pin, and the "Low" shall be applied to the VREFL pin.

To start AD conversion, make sure that you first write "1" to the <VREFON> bit, wait for 3 μs during which time the internal reference voltage should stabilize, and then write "1" to the ADMOD0<ADS> bit.

If you do not use ADC function, write "0" to the ADMOD1<DACON>. The consumption current of the analog circuit is reduced.

Note: In TMPM066/067/068FW VREFH and AVDD3 are shared.Also VREFL and AVSS are shared.

17.4.2 AD Conversion Mode

Two types of AD conversion are supported: normal AD conversion and top-priority AD conversion.

For normal AD conversion, the following four operation modes are supported.

17.4.2.1 Normal AD conversion

For normal AD conversion, the following four operation modes are supported and the operation mode is selected with the ADMOD0 <REPEAT> <SCAN>.

- Fixed channel single conversion mode
- Channel scan single conversion mode
- Fixed channel repeat conversion mode
- Channel scan repeat conversion mode

For channel scan mode, the following three modes are supported and the operation mode is selected with the ADMOD1<ADSCN>.

- 4-channel scan mode
- 8-channel scan mode
- 12-channel scan mode

(1) Fixed channel single conversion mode

If ADMOD0<REPEAT, SCAN> is set to "00", "AD conversion is performed in the fixed channel single conversion mode.

In this mode, AD conversion is performed once for one channel selected. After AD conversion is completed, ADMOD0<EOCFN> is set to "1", ADMOD0<ADBFN> is cleared to "0", and the AD conversion completion interrupt request (INTAD) is generated. <EOCFN> is cleared to "0" upon read.

(2) Channel scan single conversion mode

If ADMOD0 <REPEAT, SCAN> is set to "01," AD conversion is performed in the channel scan single conversion mode.

In this mode, AD conversion is performed once for each scan channel selected. After AD scan conversion is completed, ADMOD0<EOCFN> is set to "1", ADMOD0<ADBFN> is cleared to "0", and the conversion completion interrupt request (INTAD) is generated. <EOCFN> is cleared to "0" upon read.

(3) Fixed channel repeat conversion mode

If ADMOD0<REPEAT, SCAN> is set to "10", AD conversion is performed in fixed channel repeat conversation mode.

In this mode, AD conversion is performed repeatedly for one channel selected. After AD conversion is completed, ADMOD0<EOCFN> is set to "1". ADMOD0<ADBFN> is not cleared to "0". It remains at "1". The timing with which the conversion completion interrupt request (INTAD) is generated can be selected by setting ADMOD0<ITM> to an appropriate setting. <EOCFN> is set with the same timing as this interrupt INTAD is generated.

<EOCFN> is cleared to "0" upon read.

(4) Channel scan repeat conversion mode

If ADMOD0<REPEAT, SCAN> is set to "11", AD conversion is performed in the channel scan repeat conversion mode.

In this mode, AD conversion is performed repeatedly for a scan channel selected. Each time one AD scan conversion is completed, ADMOD0<EOCFN> is set to "1", and the conversion completion interrupt request (INTAD) is generated. ADMOD0<ADBFN> is not cleared to "0". It remains at "1". <EOCFN> is cleared to "0" upon read.

17.4.2.2 Top-priority AD conversion

By interrupting ongoing normal AD conversion, top-priority AD conversion can be performed.

The fixed-channel single conversion is automatically selected, irrespective of the ADMOD0 <REPEAT, SCAN> setting. When conditions to start operation are met, a conversion is performed just once for a channel designated by ADMOD2<HPADCH>. When conversion is completed, the top-priority AD conversion completion interrupt (INTADHP) is generated, and ADMOD2<EOCFHP> showing the completion of AD conversion is set to "1". <ADBFHP> returns to "0". EOCFHP flag is cleared to "0" upon read.

Top-priority AD conversion activated while top-priority AD conversion is under way is ignored.

17.4.3 AD Monitor Function

There are two channels of AD monitor function.

If ADMOD3<ADOBSV0> and ADMOD5<ADOBSV1> are set to "1", the AD monitor function is enabled. If the value of the conversion result register specified by ADMOD3 <ADREGS0> and ADMOD5 <ADREGS1> becomes larger or smaller ("Larger" or "Smaller" to be designated by ADMOD3 <ADOBIC0> and ADMOD5 <ADBIC1>) than the value of a comparison register, the AD monitor function interrupt (INTADM0, INTADM1) is generated. This comparison operation is performed each time a result is stored in a corresponding conversion result register.

If the conversion result register assigned to perform the AD monitor function is continuously used without reading the conversion result, the conversion result is overwritten. The conversion result storage flag <ADRxRF> and the overrun flag <OVRx> remain being set.

17.4.4 Selecting the Input Channel

How the input channel is selected is different depending on AD converter operation mode to be used.

1. Normal AD conversion mode
   - If the analog input channel is used in a fixed state (ADMOD0<SCAN> = "0")

     One channel is selected from analog input pins by setting ADMOD1<ADCH> to an appropriate setting.
   - If the analog input channel is used in a scan state (ADMOD0<SCAN> = "1")

     One scan mode is selected from the scan modes by setting ADMOD1 <ADCH> and <ADSCN> to an appropriate setting.

2. Top-priority AD conversion mode

   One channel is selected from analog input pins by setting ADMOD2<HPADCH> to an appropriate setting.

17.4.5 AD Conversion Details

17.4.5.1 Starting AD Conversion

Normal AD conversion is activated by setting ADMOD0<ADS> to "1". Top-priority AD conversion is activated by setting ADMOD2<HPADCE> to "1".

Four operation modes are made available to normal AD conversion. In performing normal AD conversion, one of these operation modes must be selected by setting ADMOD0<REPEAT,SCAN> to an appropriate setting. For top-priority AD conversion, only one operation mode can be used: fixed channel single conversion mode.

Normal AD conversion can be activated using the H/W activation source selected by ADMOD4 <ADHS>, and top-priority AD conversion can be activated using the HW activation source selected by ADMOD4 <HADHS>. If bits of <ADHS> and <HADHS> are "0", normal and top-priority AD conversions are activated in response to the input of a falling edge through the $\overline{\text{ADTRG}}$ pin. If these bits are "1", conversion is activated in response to match detection of timer.

To permit H/W activation, set ADMOD4 <ADHTG> to "1" for normal AD conversion and set ADMOD4 <HADHTG> to "1" for top-priority AD conversion.

Software activation is still valid even after H/W activation has been permitted.

Note 1: Some products don't provide the $\overline{\text{ADTRG}}$ pin.

Note 2: When an external trigger is used for the HW activation source of a top-priority AD conversion, an external trigger cannot be set for activating normal AD conversion H/W.

Note 3: For details, refer to "Product information" to confirm usable match detection of timer.

17.4.5.2 AD Conversion

When normal AD conversion starts, the AD conversion Busy flag (ADMOD0<ADBFN>) showing that AD conversion is under way is set to "1".

When top-priority AD conversion starts, the top-priority AD conversion Busy flag (ADMOD2 <ADBFHP>) showing that AD conversion is underway is set to "1".

At that time, the value of the Busy flag ADMOD0<ADBFN> for normal AD conversion before the start of top-priority AD conversions are retained. The value of the conversion completion flag ADMOD0<EOCFN> for normal AD conversion before the start of top-priority AD conversion can also be retained.

Note: Normal AD conversion must not be activated when top-priority AD conversion is under way.

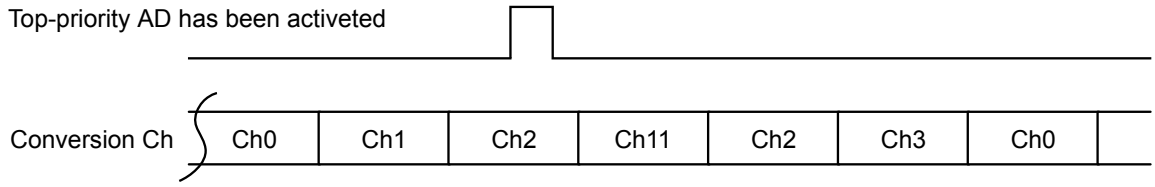
17.4.5.3 Top-priority AD conversion during normal AD conversion

If top-priority AD conversion has been activated during normal AD conversion, ongoing normal AD conversion is suspended, and restarts normal AD conversion after top-priority AD conversion is completed.

If ADMOD2<HPADCE> is set to "1" during normal AD conversion, ongoing normal AD conversion is suspended, and the top-priority AD conversion starts; specifically, AD conversion (fixed-channel single conversion) is executed for a channel designated by ADMOD2<HPADCH>. After the result of this top-priority AD conversion is stored in the storage register ADREGSP, normal AD conversion is resumed.

If H/W activation of top-priority AD conversion is authorized during normal AD conversion, ongoing AD conversion is discontinued when requirements for activation using a H/W activation resource are met, and top-priority AD conversion (fixed-channel single conversion) starts for a channel designated by ADMOD2<HPADCH>. After the result of this top-priority AD conversion is stored in the storage register ADREGSP, normal AD conversion is resumed.

For example, if channel repeat conversion is activated for channels AIN0 through AIN3 and if <HPADCE> is set to "1" during AIN2 conversion, AIN2 conversion is suspended, and conversion is performed for a channel designated by <HPADCH> (AIN11 in the case shown below). After the result of conversion is stored in ADREGSP, channel repeat conversion is resumed, starting from AIN2.



17.4.5.4 Stopping Repeat Conversion Mode

To stop the AD conversion operation in the repeat conversion mode (fixed-channel repeat conversion mode or channel scan conversion mode), write "0" to ADMOD0<REPEAT>. When ongoing AD conversion is completed, the repeat conversion mode terminates, and ADMOD0<ADBFN> is set to "0".

17.4.5.5 Reactivating normal AD conversion

To reactivate normal AD conversion while the conversion is underway, a software reset (ADMOD3 <ADRST>) must be performed before starting AD conversion. The H/W activation method must not be used to reactivate normal AD conversion.

17.4.5.6 Conversion completion

(1) Normal AD conversion completion

When normal AD conversion is completed, the AD conversion completion interrupt (INTAD) is generated. The result of AD conversion is stored in the storage register, and two registers change: the register ADMOD0 <EOCFN> which indicates the completion of AD conversion and the register ADMOD0 <ADBFN>.

For details, refer to Table 17-2 and Table 17-3 to confirm storage register corresponding to the conversion mode.

Interrupt requests, flag changes are as shown below.

- Fixed-channel single conversion mode

  After AD conversion completed, ADMOD0<EOCFN> is set to "1", ADMOD0<ADBFN> is cleared to "0", and the interrupt request is generated.

- Channel scan single conversion mode

  After the channel scan conversion is completed, ADMOD0<EOCFN> is set to "1", ADMOD0<ADBFN> is set to "0", and the interrupt request INTAD is generated.

- Fixed-channel repeat conversion mode

  ADMOD0<ADBFN> is not cleared to "0". It remains at "1". The timing with which the interrupt request INTAD is generated can be selected by setting ADMOD0<ITM> to an appropriate setting. ADMOD0<EOCFN> is set with the same timing as this interrupt INTAD is generated.

- Channel scan repeat conversion mode

  Each time one AD scan conversion is completed, ADMOD0 <EOCF> is set to "1" and interrupt request INTAD is generated. ADMOD0<ADBFN> is not cleared to "0". It remains at "1".

(2) Top-priority AD conversion completion

After the AD conversion is completed, the top-priority AD conversion completion interrupt (INTADHP) is generated, and ADMOD2<EOCFHP> which indicates the completion of top-priority AD conversion is set to "1".

AD conversion results are stored in the AD conversion result register SP.

(3) Data polling

To confirm the completion of AD conversion without using interrupts, data polling can be used. When AD conversion is completed, ADMOD0<EOCFN> is set to "1". To confirm the completion of AD conversion and to obtain the results, poll this bit.

AD conversion result storage register must be read by half word or word access. If <OVRx> = “0” and <ADRxRF> = "1", a correct conversion result has been obtained.

17.4.5.7 Interrupt generation timings and AD conversion result storage register

Table 17-1 shows a relation in the following three items: AD conversion modes, interrupt generation timings and flag operations. Table 17-2 and Table 17-3 shows a relation between analog channel inputs and AD conversion result registers.

Table 17-1 Relations in conversion modes, interrupt generation timings and flag operations

Conversion mode		Scan/repeat mode setting			Interrupt generation timing	ADMOD0<EOCFN>/ ADMOD2<EOCFHP> set timing (Note)	ADMOD0	ADMOD2
		ADMOD0 <REPEAT>	ADMOD0 <SCAIN>	ADMOD0 <ITM[1:0]>			<ADBFN> (After the interrupt is generated)	<ADBFHP>
Normal conversion	Fixed-channel single conversion	0	0	–	After generation is completed.	After conversion is completed.	0	–
	Fixed-channel repeat conversion	1	0	00	Each time one conversion is completed.	After one conversion is completed.	1	–
				01	Each time four conversion is completed.	After four conversions are completed.	1	–
				10	Each time eight conversion is completed.	After eight conversions are completed.	1	–
	Channel scan single conversion	0	1	–	After scan conversion is completed.	After scan conversion is completed.	0	–
	Channel scan repeat conversion	1	1	–	After one scan conversion is completed.	After one scan conversion is completed.	1	–
Top-priority conversion		–	–	–	After completion is completed.	Conversion completion	–	0

Note: ADMOD0<EOCFN> and ADMOD2<EOCFHP> are cleared upon read.

Table 17-2 Result registers (Fixed-channel repeat conversion mode)

<ITM[1:0]>	Result register
00 Generate in interrupt once every single conversion	ADREG0
01 Generate interrupt once every 4 conversions	ADREG0 to ADREG3
10 Generate interrupt once every 8 conversions	ADREG0 to ADREG7

Table 17-3 Result registers (Except for fixed-channel repot conversion mode)

ADMOD1 <ADCH[3:0]>	ADMOD0 <SCAN>=0		ADMOD0 <SCAN>=1					
	Fixed channel		<ADSCN>=00 4-channel scan		<ADSCN>=00 8-channel scan		<ADSCN>=00 12-channel scan	
	Conversion channel	Result register	Conversion channel	Result register	Conversion channel	Result register	Conversion channel	Result register
0000	AIN0	ADREG0	AIN0	ADREG0	AIN0	ADREG0	AIN0	ADREG0
0001	AIN1	ADREG1	AIN0 to AIN1	ADREG0 to ADREG1	AIN0 to AIN1	ADREG0 to ADREG1	AIN0 to AIN1	ADREG0 to ADREG1
0010	AIN2	ADREG2	AIN0 to AIN2	ADREG0 to ADREG2	AIN0 to AIN2	ADREG0 to ADREG2	AIN0 to AIN2	ADREG0 to ADREG2
0011	AIN3	ADREG3	AIN0 to AIN3	ADREG0 to ADREG3	AIN0 to AIN3	ADREG0 to ADREG3	AIN0 to AIN3	ADREG0 to ADREG3
0100	AIN4	ADREG4	AIN4	ADREG4	AIN0 to AIN4	ADREG0 to ADREG4	AIN0 to AIN4	ADREG0 to ADREG4
0101	AIN5	ADREG5	AIN4 to AIN5	ADREG4 to ADREG5	AIN0 to AIN5	ADREG0 to ADREG5	AIN0 to AIN5	ADREG0 to ADREG5
0110	AIN6	ADREG6	AIN4 to AIN6	ADREG4 to ADREG6	AIN0 to AIN6	ADREG0 to ADREG6	AIN0 to AIN6	ADREG0 to ADREG6
0111	AIN7	ADREG7	AIN4 to AIN7	ADREG4 to ADREG7	AIN0 to AIN7	ADREG0 to ADREG7	AIN0 to AIN7	ADREG0 to ADREG7
1000	AIN8	ADREG0	AIN8	ADREG0	AIN8	ADREG0	AIN0 to AIN8	ADREG0 to ADREG8
1001	AIN9	ADREG1	AIN8 to AIN9	ADREG0 to ADREG1	AIN8 to AIN9	ADREG0 to ADREG1	AIN0 to AIN9	ADREG0 to ADREG9
1010	AIN10	ADREG2	AIN8 to AIN10	ADREG0 to ADREG2	AIN8 to AIN10	ADREG0 to ADREG2	AIN0 to AIN10	ADREG0 to ADREG10
1011	AIN11	ADREG3	AIN8 to AIN11	ADREG0 to ADREG3	AIN8 to AIN11	ADREG0 to ADREG3	AIN0 to AIN11	ADREG0 to ADREG11
1100	AIN12	ADREG4	AIN12	ADREG4	AIN8 to AIN12	ADREG0 to ADREG4	-	-
1101	AIN13	ADREG5	AIN12 to AIN13	ADREG4 to ADREG5	AIN8 to AIN13	ADREG0 to ADREG5	-	-
1110	AIN14	ADREG6	AIN12 to AIN14	ADREG4 to ADREG6	AIN8 to AIN14	ADREG0 to ADREG6	-	-
1111	AIN15	ADREG7	AIN12 to AIN15	ADREG4 to ADREG7	AIN8 to AIN15	ADREG0 to ADREG7	-	-

17.4.5.8 DMA Request

A DMA request is issued to the DMAC at the timing of AD conversion completion interrupts or AD monitor function interrupts generation.

When DMA transfer is performed, set DMA request to be enabled by the corresponding bit of ADMOD6 register

17.4.5.9 Cautions

Cautions
The result value of AD conversion may vary depending on the fluctuation of the supply voltage, or may be affected by noise.
When using analog input pins and ports alternately, do not read and write ports during conversion because the conversion accuracy may be reduced. Also the conversion accuracy may be reduced if the output ports current fluctuate during AD conversion.
Please take counteractive measures with the program such as averaging the AD conversion results.

18. Low Voltage detection circuit (LVD)

The low voltage detection circuit generates reset or an interrupt (INTLVD) by detecting a decreasing voltage.

Note: INTLVD is a factor of non-maskable interrupts (NMI).

18.1 Configuration

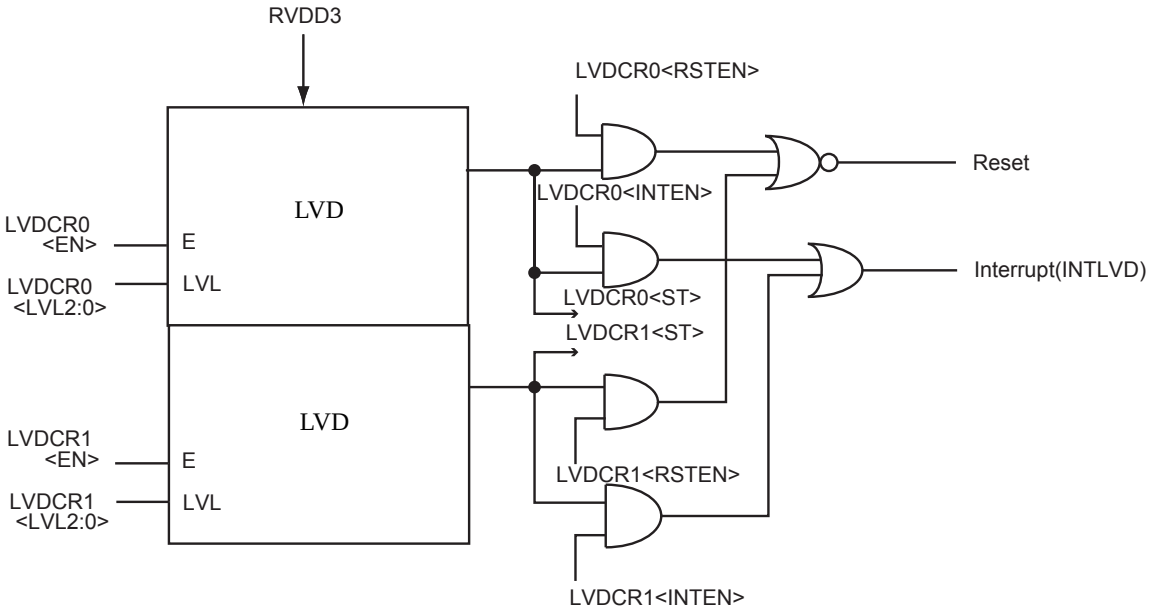


Figure 18-1 LVD Block Diagram

18.2 Registers

The following table lists the control registers and addresses.

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

18.2.1 Register list

Register name		Address (Base+)
LVD detection control 0	LVDCR0	0x0000
LVD detection control 1	LVDCR1	0x0004

18.2.2 LVDCR0 (LVD detection 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	ST	RSTEN	INTEN	INTSEL	LVL			EN
after reset	0	1	0	0	1	0	0	1

Bit	Bit Symbol	Type	Function
31 - 8	–	R	Read as "0"
7	ST	R	LVL voltage detection status 0: Power-supply voltage is the same as detection voltage or higher. 1: Power-supply voltage is the same as detection voltage or lower.
6	RSTEN	R/W	Controls RESET output 0: Disable 1: Enable
5	INTEN	R/W	Controls INTLVD output 0: Disable 1: Enable
4	INTSEL	R	Interrupt generation condition. 0: Only lower than the setting voltage when voltage decreasing. 1: Both lower and upper than the setting voltage when voltage decreasing. this bit can be used with <RSTEN>="0" and <INTEN>="1".
3 - 1	LVL[2:0]	R/W	3V Power supply detection voltage 0 000: Reserved 001: Reserved 010: Reserved 011: Reserved 100: 2.0 ± 0.2V 101: 2.1 ± 0.2V 110: Reserved 111: Reserved
0	EN	R/W	Voltage detection operation 0: Disable 1: Enable

Note 1: In this Device, it is prohibited to enable both <RSTEN> and <INTEN>.

Note 2: LVDCRn is initialized by power-on reset and a terminal reset.

Note 3: There is not hysteresis between the detection of LVD and release voltage. Chattering may occur during detection, depending on the slope of power supply.

18.2.3 LVDCR1 (LVD detection 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	ST	RSTEN	INTEN	INTSEL	LVL			EN
after reset	0	0	0	0	0	1	1	0

Bit	Bit Symbol	Type	Function
31 - 8	–	R	Read as "0"
7	ST	R	LVL voltage detection status 0: Power-supply voltage is the same as detection voltage or higher. 1: Power-supply voltage is the same as detection voltage or lower.
6	RSTEN	R/W	Controls RESET output 0: Disable 1: Enable
5	INTEN	R/W	Controls INTLVD output 0: Disable 1: Enable
4	INTSEL	R	Interrupt generation condition. 0: Only lower than the setting voltage when voltage decreasing. 1: Both lower and upper than the setting voltage when voltage decreasing. this bit can be used with <RSTEN>="0" and <INTEN>="1".
3 - 1	LVL[2:0]	R/W	3V Power supply detection voltage 1 000: 2.2 ± 0.2V 001: 2.3 ± 0.2V 010: 2.4 ± 0.2V 011: 2.5 ± 0.2V 100: 2.6 ± 0.2V 101: 2.7 ± 0.2V 110: 2.8 ± 0.2V 111: 2.9 ± 0.2V
0	EN	R/W	Voltage detection operation 0: Disable 1: Enable

Note 1: In this Device, it is prohibited to enable both <RSTEN> and <INTEN>.

Note 2: LVDCRn is initialized by power-on reset and a terminal reset.

Note 3: There is not hysteresis between the detection of LVD and release voltage. Chattering may occur during detection, depending on the slope of power supply.

18.3 Operation

18.3.1 Selecting detection voltage and enabling voltage detection operation

Voltage detection is enabled when voltage to be detected is selected by setting the register LVDCRn<LVL [2:0]> and "1" is set to the LVDCRn<EN>.

18.3.2 Reset by Detecting a supply voltage

Reset occurs when "1" is set to the LVDCRx<RSTEN> and the supply voltage falls under the set detection voltage.

It needs approximately 100 μs to detect voltage reduction and generate reset. If the period that the supply voltage falls under the detected voltage is short, reset may not occur.

18.3.3 Interrupt by Detecting a supply voltage

A interrupt (INTLVD) occurs When "0" is set to LVDCEn<RSTEN> and "1" is set to LVDCRn<INTEN> if the supply voltage falls under or over the set detection voltage level.

The Interrupt condition can be set by LVDCRn<INTSEL>.

An interrupt occurs when "0" is set to the LVDCRx<INTSEL> and the supply voltage falls under the set detection voltage.

An interrupt occurs when "1" is set to the LVDCRx<INTSEL> and the supply voltage falls under or over the set detection voltage.

It needs approximately 100 μs to detect voltage reduction and generate a interrupt. If the period that the supply voltage falls under the detected voltage is short, an interrupt may not occur.

18.3.4 Detecting Status

Reading the LVDCRn<SR> can be confirmed the Detecting status of low voltage detection.

When the LVDCRn<ST> is "0", the voltage is the same as detection voltage or higher.

When the LVDCRn<ST> is "1", the voltage is the same as detection voltage or lower.

In the Interrupt service routine (ISR) of low voltage detection (LVD), to read the status of LVDCRn<ST> after detection is known stably the shift of the voltage

19. Watchdog Timer (WDT)

The watchdog timer (WDT) is for detecting malfunctions (runaway) 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".

19.1 Configuration

Figure 19-1 shows the block diagram of the watchdog timer.

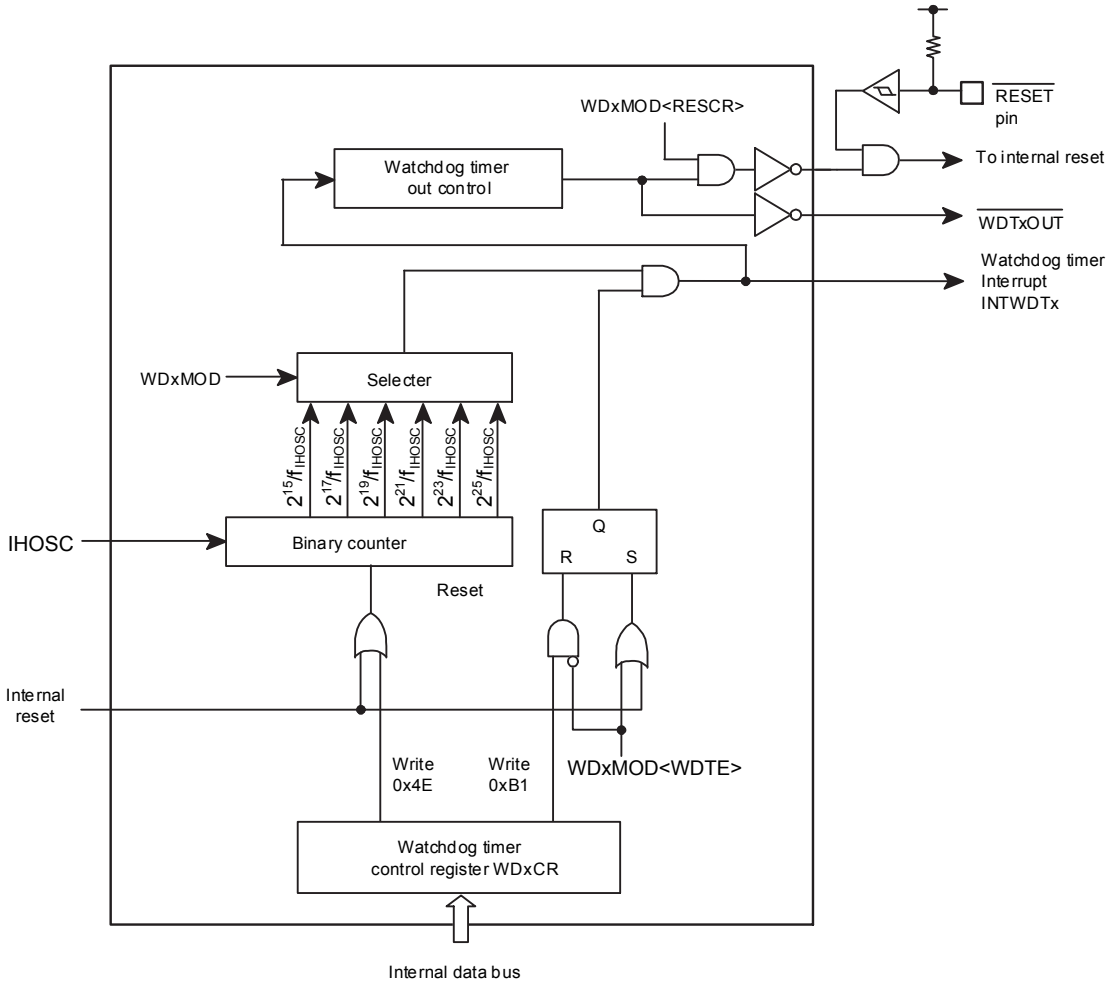


Figure 19-1 Block Diagram of the watchdog Timer

19.2 Register

19.2.1 Register List

The table below shows control registers and their addresses.

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

Register name		Address (Base+)
Watchdog Timer Mode Register	WDMOD	0x0000
Watchdog Timer Control Register	WDCR	0x0004
Watchdog Flag Register	WDFLG	0x0008

19.2.2 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			-	-	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 To disable the watchdog timer to protect from the error writing by the malfunction, first <WDTE> is set to "0", and then the disable code (0xB1) must be written to WDCR. To change the status of the watchdog timer from "disable" to "enable", set <WDTE> to "1".
6-4	WDTP[2:0]	R/W	Selects WDT detection time 000: $2^{15}/f_{SYS}$ 100: $2^{23}/f_{SYS}$ 001: $2^{17}/f_{SYS}$ 101: $2^{25}/f_{SYS}$ 010: $2^{19}/f_{SYS}$ 110: Reserved 011: $2^{21}/f_{SYS}$ 111: Reserved
3	−	R	Read as "0"
2	−	R/W	Reserved.
1	RESCR	R/W	Operation after detecting malfunction 0: INTWDT interrupt request is generated. Note) 1: Reset
0	−	R/W	Write "0".

Note: INTWDT interrupt is a factor of the non-mask interrupt.

19.2.3 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	0	0	0	0	0	0	0	0

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

19.2.4 WDFLG (Watchdog 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	-	-	-	-	-	-	-	FLG
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	FLG	R	Flag for writing to registers 0: Enable 1: Disable When writing to WDMOD or WDCR, confirm "0" of this bit.

19.3 Description of Operation

19.3.1 Basic Operation

The watchdog timer is consists of the binary counter that works 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) is generated, and the watchdog timer out pin ($\overline{\text{WDTOUT}}$) outputs "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 is generated. If the binary counter is not cleared, the non-maskable interrupt is generated by INTWDT. Thus CPU detect malfunction (runaway), 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.

19.3.2 Operation Mode and Status

The watchdog timer begins operation immediately after a reset is released. If not using the watchdog timer, it should be disabled.

The watchdog timer can not be used at the high-speed frequency clock is stopped. Before transition to below operation modes, the watchdog timer should be disabled.

- STOP1 mode

Also, the binary counter is automatically stopped during debug mode.

19.3.3 Operation when malfunction (runaway) is detected.

19.3.3.1 INTWDT interrupt generation

Figure 19-2 shows the case that INTWDT interrupt is generated (WDMOD<RESCR>="0").

When an overflow of the binary counter occurs, INTWDT interrupt is generated. It is a factor of non-maskable interrupt (NMI). Thus CPU detects non-maskable interrupt and perform the countermeasure program.

When INTWDT interrupt is generated, 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.

In addition, Clear the interrupt request to the IDLExxxINT register in interrupt processing, etc.

For detail, refer to the Exception chapter.

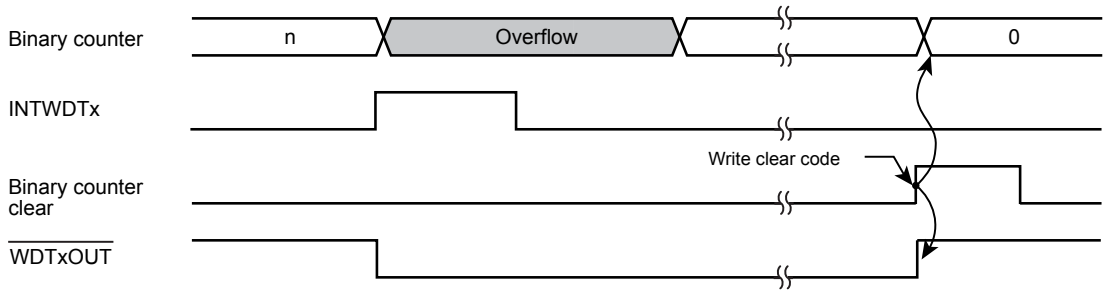


Figure 19-2 INTWDT interrupt generation

19.3.3.2 Internal Reset Generation

Figure 19-3 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.

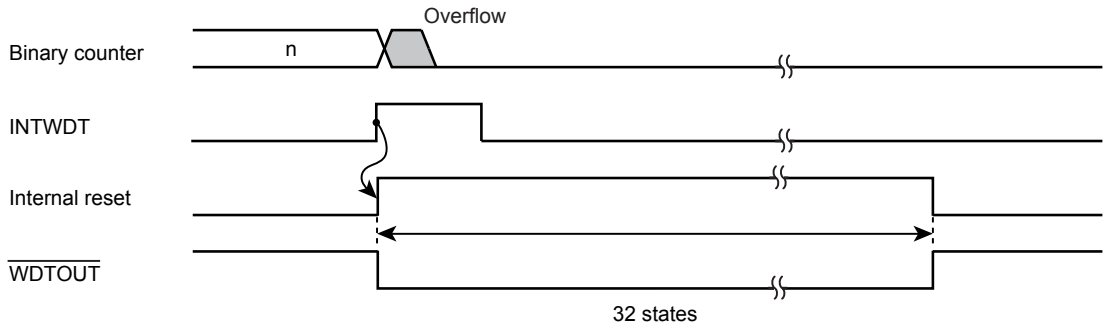


Figure 19-3 Internal reset generation

19.4 Control of the watchdog timer

19.4.1 Register access

When writing to WDMOD and WDCR, confirm whether WDFLG<FLG> is "0".

However, the consecutive writing to WDMOD and WDCR is available for the case of the disable control. Confirming WDFLG<FLG> is required only before writing to WDMOD.

19.4.2 Disable control

By writing the disable code (0xB1) to WDCR after setting WDMOD<WDTE> to "0", the watchdog timer can be disabled and the binary counter can be cleared.

19.4.3 Enable control

Set WDMOD<WDTE> to "1".

19.4.4 Watchdog timer clearing control

Writing the clear code (0x4E) to WDCR clears the binary counter and it restarts counting.

19.4.5 Detection time of watchdog timer

Set WDMOD<WDTP[2:0]> depend on the detection time.

For example, in the case that $2^{21}/f_{SYS}$ is used, set "011" to WDMOD<WDTP[2:0]>.

20. Flash Memory Operation

This section describes the hardware configuration and operation of Flash memory. In this section, "1-word" means 32 bits.

20.1 Features

20.1.1 Memory Size and Configuration

Table 20-1 and Figure 20-1 show a built-in memory size and configuration of TMPM066/067/068FW.

Table 20-1 Memory size and configuration

Memory size	Block configuration				# of words per page	# of pages	Write time		Erase time	
	128 KB	64 KB	32 KB	16 KB			1 page	Total area	Block erase	Chip erase
128 KB	–	–	4	–	32	1024	1.25ms	1.28 sec	0.1sec	0.2 sec

Note: The above values are theoretical values not including data transfer time. The write time per chip depends on the write method used by a user.

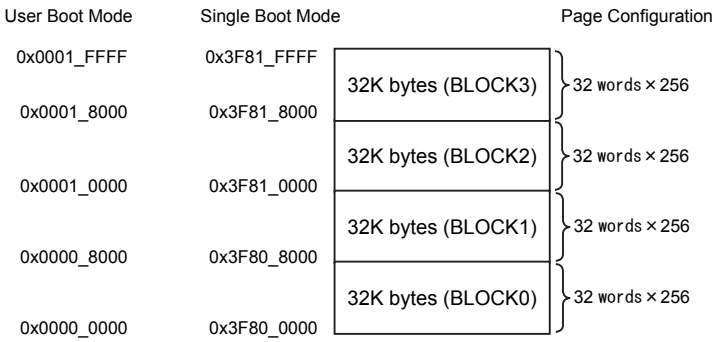


Figure 20-1 Block configuration

Flash memory configuration units ares described as "block" and "page".

- Page

  One page is 32 words. Same address [31:7] is used in a page. First address of the group is [6:0] = 0 and the last address of the group is [6:0] = 0x7F.

- Block

  One block is 32KB and flash memory is consists of four blocks.

Write operation is performed per page. The write time per page is 1.25ms. (Typ.)

Erase is performed per block (auto block erase command use) or performed on entire flash memory (use of auto chip erase command). Erase time varies on commands. If auto block command is used, the erase time will be 0.1 sec per block (Typ.). If the auto chip erase command is used to erase entire area, the time will be 0.2 sec (Typ.).

In addition, the protect function can be used per block. For detail of the protect function, refer to "20.1.5 Protect/Security Function".

20.1.2 Function

Flash memory built-in this device is generally compliant with the JEDEC standards except for some specific functions. Therefore, if a user is currently using a flash memory as an external memory, it is easy to implement the functions into this device. Furthermore, to provide easy write or erase operation, this product contains a dedicated circuit to perform write or chip erase automatically.

JEDEC compliant functions	Modified, added, or deleted functions
・Automatic programming ・Automatic chip erase ・Automatic block erase ・Data polling/toggle bit	<Modified> Block write/erase protect (only software protection is supported) <Deleted> Erase resume - suspend function

20.1.3 Operation Mode

20.1.3.1 Mode Description

This device provides the single chip mode and single boot mode. The single chip mode contains the normal mode and user boot mode. Figure 20-2 shows the mode transition.

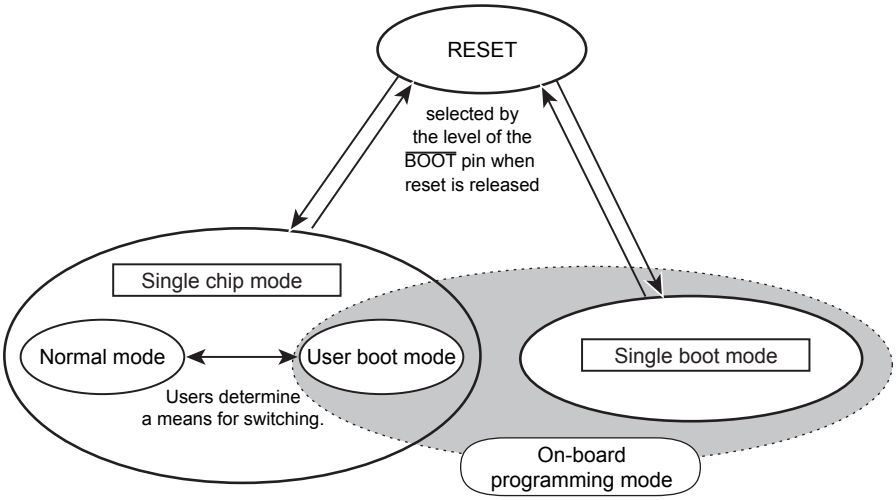


Figure 20-2 Mode transition

(1) Single chip mode

The single chip mode is a mode where the device can boot-up from Flash memory after reset. The mode contains two sub-modes in below.

- Normal mode

  The mode where user application program is executed.
- User boot mode

  The mode where flash memory is re-programmed on the user's set.

Users can switch the normal mode to user boot mode freely. For example, a user can set if PA0 of port A is "1", the mode is the normal mode. If PA0 of port A is "0", the mode is the user boot mode. The user must prepare a routine program in the application program to determine the switching.

(2) Single boot mode

The mode where flash memory can boot-up from the built-in BOOT ROM (Mask ROM) after reset.

The BOOT ROM contains the algorithm that can rewrite Flash memory via serial port of this device on the user's set. With connecting the serial port to external host, data transfer is performed in above-mentioned protocol and re-programmed Flash memory.

(3) On-board programming mode

The user boot mode and single boot mode are the modes where flash memory can be re-programmable on the user's set. These two modes are called "on-board programming mode".

20.1.3.2 Mode Determination

Either the single chip or single boot operation mode can be selected by the level of the $\overline{\text{BOOT}}$ pin when reset is released.

Table 20-2 Operation mode setting

Operation mode	Pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$
Single chip mode	0 → 1	1
Single boot mode	0 → 1	0

20.1.4 Memory Map

Figure 20-3 shows a comparison of the memory map in the single chip mode and single boot mode. In the single boot mode, built-in Flash memory is mapped to 0x3F80_0000 and subsequent addresses, and the built-in BOOT ROM is mapped to 0x0000_0000 through 0x0000_0FFF.

Flash memory and RAM addresses are shown below.

FLASH size	RAM size	FLASH address	RAM address
128 KB	16 KB	0x0000_0000 to 0x0001_FFFF(single chip mode) 0x3F80_0000 to 0x3F81_FFFF(single boot mode)	0x2000_0000 to 0x2000_3FFF

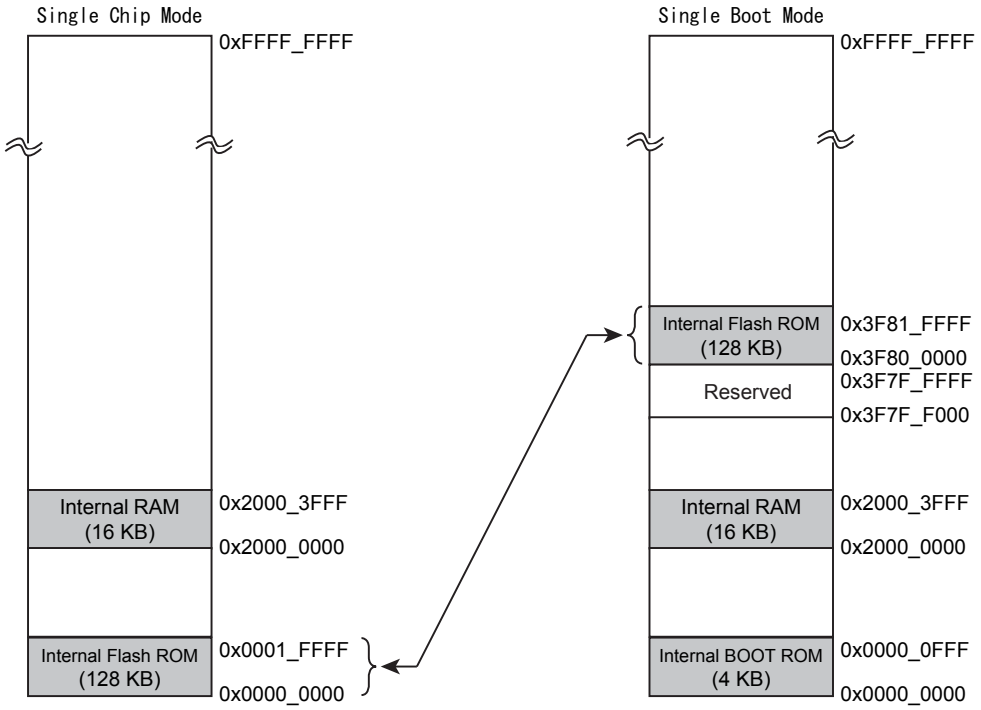


Figure 20-3 Comparison of memory map

20.1.5 Protect/Security Function

This device has the protect and security functions for Flash memory.

1. Protect function

   The write/erase operation can be inhibited per block.

2. Protect bit Mask function

   Temporarily release the protect function

3. Security function

   The read operation from a flash writer can be inhibited.

   Usage restrictions on debug functions

20.1.5.1 Protect Function

This function inhibits the write/erase operation per block.

To enable the protect function, a protect bit corresponding to a block is set to "1" using the protect bit program command. If a protect bit is set to "0" using the protect bit erase command, a block protect can be cancelled. The protect bit can be monitored with FCPSRA<BLK[31:0]>.

A program of protect bit can be programmed by 1-bit unit and can be erased by 4-bit unit. For detail of programming/erasing of protect bits, refer to "20.2.5 Command Description".

20.1.5.2 Protect Bit Mask Function

TMPM066/067/068FW can temporarily release the protect function by masking the protect bits. To enable the protect Bit Mask function, set the corresponding bit of FCPMRA<BLKM[3:0]> to "0".

Note that when the system reset occurs, FCPMRA<BLKM[3:0]> is set to "1".

The contents of the protect bit are maintained in the non-maskable state.

FCPMRA<BLKM[3:0]> should be written as follows:

Note: Use a 32-bit transfer instruction when the following writing operations, item1 and 2.

1. Write the specified code (0xa74a9d23) to FCPMRA
2. Write data within 16 clocks after the operation of item 1.

Note 1: When the Protect Bit Mask function is enabled, Automatic Chip Erase can not be used.

20.1.5.3 Security Function

Table 20-3 shows operations when the security function is enabled.

Table 20-3 Operations when the security function is enabled.

Item	Description
Read flash memory	CPU can read flash memory.
Debug port	JTAG, serial wire or trace communication is disabled.
Command execution to Flash memory	Command write to flash memory is not accepted. If a user tries to erase a protect bit, chip erase is executed and all protect bits are erased. (note)When using Protect bit Mask Function, Block Erase and Page Program command only can be used.

The security function is enabled under the following conditions;

1. FCSECBIT<SECBIT> is set to "1".
2. All protect bits (FCPSRA<BLK[3:0]>) are set to "1".

FCSECBIT<SECBIT> is set to "1" by the PowerOnReset. Rewriting of FCSECBIT <SECBIT> is described in below.

Note: Use a 32-bit transfer instruction when the following writing operations, item1 and 2.

1. Write the specified code (0xa74a9d23) to FCSECBIT
2. Write data within 16 clocks after the operation of item 1.

20.1.6 Register

20.1.6.1 Register List

The following table shows control registers and addresses.

For details of base address, refer to "Address lists of peripheral functions" of "Memory Map" Chapter.

Register name		Address(Base+)
Security bit register	FCSECBIT	0x0010
Flash status register	FCSR	0x0020
Flash protect status register A	FCPSRA	0x0030
Flash protect Mask register A	FCPMRA	0x0038

20.1.6.2 FCSR (Flash status register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	RDY_BSY	R	Ready/Busy (Note 1) 0: Busy (during auto operation) 1: Ready (auto operation ends) This bit is a function bit to monitor flash memory from CPU. While flash memory is in auto operation, this bit outputs "0" to indicate that flash memory is busy. Once auto operation is finished, this bit becomes ready state and outputs "1". Then next command is accepted. If a result of auto operation is failed, this bit outputs "0" continuously. The bit returns to "1" by hardware reset.

Note 1: Make sure that flash memory is ready before commands are issued. If a command is issued during busy, not only the command is not sent but also subsequent commands may not be accepted. In that case, use hardware reset to return. Hardware reset needs 0.5 µs or more reset period regardless of system clock. At this time, it takes approximately 2 ms until enabling to read after reset.

20.1.6.3 FCSECBIT (Security bit register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	SECBIT	R/W	Security bit 0: Security function setting is disabled. 1: Security function setting is enabled.

Note: This register is initialized by PowerOnReset.

20.1.6.4 FCPSRA (Flash protect status register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	BLK3	BLK2	BLK1	BLK0
After reset	0	0	0	0	(Note 1)	(Note 1)	(Note 1)	(Note 1)

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3-0	BLK3- BLK0	R	Protection status of Block3 to 0 0: Not protected 1: Protected Protect bit values correspond to protect status of each block. If corresponding bit indicates "1", corresponding block is in the protection status. A block in the protection status cannot be re-programmable.

Note 1: A value will correspond to the protection status.

20.1.6.5 FCPMRA (Flash protect mask register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	BLKM3	BLKM2	BLKM1	BLKM0
After reset	0	0	0	0	1	1	1	1

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3-0	BLKM3 to BLKM0	R/W	Masks protect bits of block3 through block 0. 0: Release the protect function (Protect bit is mask.) 1: - (Protect bit is not masked.) When the corresponding bit is "0", this block is released. When the corresponding bit is "1", the corresponding bit is set to the protect state (each bit of FCPSRA) and this block in the Flash memory becomes protect state.

Note 1: This register is initialized by system reset.

Note 2: Do not modify FCPMRA<BLKM[3:0]> while data is being written/erased to/from the Flash memory.

Note 3: When FCPMRA <BLKM[3:0]> is modified, read the register again to check whether the Flash is re-written. Then access the Flash memory.

20.2 Detail of Flash Memory

In on-board programming, the CPU executes commands for reprogramming or erasing Flash memory. This reprogramming/erase control program should be prepared by the user beforehand. Since Flash memory content cannot be read while Flash memory is being written or erased, it is necessary to run the reprogram/erase control program on the built-in RAM. Do not generate interrupt/fault except reset to avoid abnormal program termination.

20.2.1 Function

Flash memory is generally compliant with the JEDEC standards except for some specific functions. However; a method of address designation of operation command is different from standard commands.

If write/erase operation is executed, commands are input to flash memory using 32-bit (1-word) store instruction command. After command input, write or erase operation is automatically executed in inside.

Table 20-4 Flash memory function

Main function	Description
Automatic page program	Writes data automatically.
Automatic chip erase	Erases the entire area of Flash memory automatically.
Automatic block erase	Erases a selected block automatically.
Write/erase protect	The write or erase operation can be individually inhibited for each block.

Note: Check the FCSR<RDY_BSY> to make sure each command sequence end such as Flash writing, Flash Erase, Protection bit program, Protection bit Erase. and then hold for 200 μs or more before reading data from Flash memory or starting instruction fetch.

20.2.2 Operation Mode of Flash Memory

Flash memory provides mainly two types of operation modes;

- The mode to read memory data (Read mode)
- The mode to erase or rewrite memory data automatically (Automatic operation mode)

After power-on, after rest or after automatic operation mode is finished normally, Flash memory becomes read mode. Instruction stored in Flash memory or data read is executed in the read mode.

If commands is input during the read mode, the operation mode becomes the automatic operation. If the command process is normally finished, the operation mode returns to the read mode except the ID-Read command. During the automatic operation, data read and instruction execution stored in Flash memory cannot be performed.

If command process is abnormally finished then the operation mode should forcibly return to read mode. In this case, use the read command, read/reset command or hardware reset.

20.2.3 Hardware Reset

A hardware reset means a PowerOnReset or warm reset to use returning to the read mode when the automatic programming/erase operation is forcibly cancelled, or automatic operation abnormally ends.

If the hardware reset occurs during the automatic operation, Flash memory stops the automatic operation and returns to the read mode. If a hardware reset is generated during Flash memory automatic program/erase operation, the hardware reset needs 0.5 μs or more reset period regardless of system clock. At this time, it takes approximately 2 ms until enabling to read after reset. Note that if a hardware reset occurs during the automatic operation, data write operation is not executed properly. Set write operation again.

For detail of the reset operation, refer to "Reset". After a given reset input, CPU will read the reset vector data and then starts the routine after reset.

20.2.4 How to Execute Command

The command execution is performed by writing command sequences to Flash memory with a store instruction. Flash memory executes each automatic operation command according to the combination of input addresses and data. For detail of the command execution, refer to "20.2.5 Command Description".

An execution of store instruction to the Flash memory is called "bus write cycle". Each command consists of some bus write cycles. In Flash memory, when address and data of bus write cycle are performed in the specified order, the automatic command operation is performed. When the cycle is performed in non-specified order, Flash memory stops command execution and returns to the read mode.

If you cancel the command during the command sequence or input a different command sequence, execute the read command or read/reset command. Then Flash memory stops command execution and returns to the read mode. The read command and read/reset command are called "software reset".

When write command sequence ends, the automatic operation starts and FCSR<RDY_BSY> is set to "0". When the automatic operation normally ends, FCSR<RDY_BSY> = "1" is set and Flash memory returns to the read mode.

New command sequences are not accepted during the automatic operation. If you want to stop the command operation, use a hardware reset. In case that the automatic operation abnormally ends (FCSR<RDY_BSY> remains "0"), Flash memory remains locked and will not return to the read mode. To returns to the read mode, use a hardware reset. If the hardware reset stops the command operation, commands are not normally executed.

Notes on the command execution;

1. To recognize command, command sequencer need to be in the read mode before command starting. Confirm FCSR<RDY_BSY> = 1 is set prior to the first bus write cycle of each command. Consecutively, it is recommended that the read command is executed.
2. Execute each command sequence from outside of Flash memory.
3. Execute sequentially each bus write cycle by data transfer instruction in one-word (32-bit).
4. Do not access Flash memory during the each command sequence. Do not generate any interrupt or fault except reset.
5. Upon issuing a command, if any address or data is incorrectly written, make sure to return to the read mode by using software reset.

20.2.5 Command Description

This section explains each command content. For detail of specific command sequences, refer to "20.2.6 Command Sequence".

20.2.5.1 Automatic Page Program

(1) Operation Description

The automatic page program writes data per page. When the program writes data to multiple pages, a page command need to be executed in page by page. Writing across pages is not possible.

Writing to Flash memory means that data cell of "1" becomes data of "0". It is not possible to become data cell of "1" from data of "0". To become data cell of "1" from "0", the erase operation is required.

The automatic page program is allowed only once to each page already erased. Either data cell of "1" or "0" cannot be written data twice or more. If rewriting to a page that has already been written once, the automatic page program is needed to be set again after the automatic block erase or automatic chip erase command is executed.

Note 1: Page program execution to the same page twice or more without erasing operation may damage the device.

Note 2: Writing to the protected block is not possible.

(2) How to Set

The 1st to 3rd bus write cycles indicate the automatic page program command.

In the 4th bus write cycle, the first address and data of the page are written. On and after 5th bus cycle, one page data will be written sequentially. Data is written in one-word unit (32-bit).

If a part of the page is written, set "0xFFFFFFFF" as data, which means not required to write, for entire one page.

No automatic verify operation is performed internally in the device. So, be sure to read the data programmed to confirm that it has been correctly written.

If the automatic page program is abnormally terminated, that page has been failed to write. It is recommended not to use the device or not to use the block including the failed address.

20.2.5.2 Automatic Chip Erase

(1) Operation Description

The automatic chip erase is executed to the memory cell of all addresses. If protected blocks are contained, these blocks will not be erased. If all blocks are protected, the automatic chip erase operation will not performed and will return to the read mode after a command sequence is input.

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic chip erase command. After the command sequence is input, the automatic chip erase operation starts.

No automatic verify operation is performed internally in the device. So, be sure to read the data to confirm that it has been correctly erased.

20.2.5.3 Automatic Block Erase

(1) Operation Description

The automatic erase command performs erase operation to the specified block. If the specified block is protected, erase operation is not executed.

(2) How to Set

The 1st to 5th bus write cycles indicate the automatic block erase command. In the 6th bus write cycle, the block to be erased is specified. After the command sequence is input, the automatic block erase operation starts.

No automatic verify operation is performed internally in the device. So, be sure to read the data to confirm that it has been correctly erased.

20.2.5.4 Automatic Protect Bit Program

(1) Operation Description

The automatic protect bit program writes "1" to a protect bit at a time. To set "0" to a protect bit, use the automatic protect bit erase command.

For detail of the protect function, refer to "20.1.5 Protect/Security Function".

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic protect bit program command. In the 7th bus write cycle, the protect bit to be written is specified. After the command sequence is input, the automatic protect bit program starts. Check whether write operation is normally terminated with FCPSRA<BLK[3:0]>.

20.2.5.5 Auto Protect Bit Erase

(1) Operation Description

The automatic protect bit erase command operation depends on the security status. For detail of security status, refer to "20.1.5 Protect/Security Function".

- Non-security status

  Clear the specified protect bit to "0". Protect bit erase is performed in 4-bit unit.
- Security status

  Erase all protect bits after all addresses of Flash memory are erased.

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic protect bit erase command. In the 7th bus write cycle, the protect bit to be erased is specified. After the command sequence is input, the automatic protect bit erase operation starts.

In the non-security status, specified protect bit is erased. Check whether erase operation is normally terminated with FCPSRA<BLK[3:0]>.

In the security status, all addresses and all protect of Flash memory bits are erased. Confirm if data and protect bits are erased normally. If necessary, execute the automatic protect bit erase, automatic chip erase or automatic block erase.

All cases are the same as other commands, FCSR<RDY_BSY> becomes "0" during the automatic protect bit erase command operation. After the operation is complete, FCSR<RDY_BSY> becomes "1" and Flash memory will return to the read mode. To abort the operation, a hardware reset is required.

20.2.5.6 ID-Read

(1) Operation Description

The ID-Read command can read information including Flash memory type and three types of codes such as a maker code, device code and macro code.

(2) How to Set

The 1st to 3rd bus write cycles indicate the ID-Read command. In the 4th bus write cycle, the code to be read is specified. After the 4th bus write cycle, read operation in the arbitrary flash area acquires codes.

The ID-Read can be executed successively. The 4th bus write cycle and reading ID value can be executed repeatedly.

The ID-Read command does not automatically return to the read mode. To return to the read mode, execute the read command, read/reset command or hardware reset.

20.2.5.7 Read Command and Read/reset Command (Software Reset)

(1) Operation Description

A command to return Flash memory to the read mode.

When the ID-Read command is executed, macro stops at the current status without automatically return to the read mode. To return to the read mode from this situation, use the read command or read/reset command. It is also used to cancel the command when commands are input to the middle.

(2) How to Set

The 1st bus cycle indicates the read command. The 1st to 3rd bus write cycles indicate the read/reset command. After either command sequence is executed, Flash memory returns to the read mode.

20.2.6 Command Sequence

20.2.6.1 Command Sequence List

Table 20-5 shows addresses and data of bus write cycle in each command.

All command cycles except the 5th bus cycle of ID-Read command are bus write cycles. A bus write cycle is performed by 32-bit (1-word) data transfer instruction. (Following table shows only lower 8 bits of data.)

For detail of addresses, refer to Table 20-6. Use below values to "command" described in a column of Addr[15:9] in the Table 20-6.

Note 1) Always set to "0" to the address bit [1:0].

Note 2) Set below values to the address bit [19] according to Flash memory size.
Memory size is 1MB or less : Always set to "0"
Memory size is over 1MB : If bus write to 1MB area or less, the bit is set to "0".
If bus write to over 1MB area, the bit is set to "1".

Table 20-5 Command Sequence

Command	1st bus cycle	2nd bus cycle	3rd bus cycle	4th bus cycle	5th bus cycle	6th bus cycle	7th bus cycle
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	Data	Data	Data	Data	Data	Data	Data
Read	0xXX	–	–	–	–	–	–
	0xF0	–	–	–	–	–	–
Read/reset	0xX55X	0xXAAX	0xX55X	–	–	–	–
	0xAA	0x55	0xF0	–	–	–	–
ID-Read	0xX55X	0xXAAX	0xX55X	IA	0xXX	–	–
	0xAA	0x55	0x90	0x00	ID	–	–
Automatic page program	0xX55X	0xXAAX	0xX55X	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic chip erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	–
	0xAA	0x55	0x80	0xAA	0x55	0x10	–
Automatic block erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	BA	–
	0xAA	0x55	0x80	0xAA	0x55	0x30	–
Automatic protect bit program	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
Automatic protect bit erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	0xXX
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

Supplementary explanation

- IA: ID Address
- ID: ID data
- PA: Program page address
- PD: Program data (32-bit data)

  After the 4th bus cycle, input data in the order of the addresses per page
- BA: Block address (see Table 20-7)

- PBA: Protect bit address (see Table 20-8)

20.2.6.2 Address Bit Configuration in the Bus Cycle

Table 20-6 is used in conjunction with "Table 20-5 Command Sequence".

Set the address setting according to the normal bus write cycle address configuration from the first bus cycle.

Table 20-6 Address bit configuration in the bus write cycle

Address	Addr [31:15]	Addr [14]	Addr [13:12]	Addr [11:9]	Addr [8:7]	Addr [6:4]	Addr [3:0]

Normal Command	Normal bus write cycle address configuration				
	Flash area	"0" is recommended.		Command	Addr[1:0] = "0" (fixed) Other bits = "0" (recommended)
ID-READ	IA: ID address (Setting of the 4th bus write cycle address for ID-READ)				
	Flash area	"0" is recom-mended.	ID Address		Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)
Block erase	BA: Block address (Setting of the 6th bus write cycle address for block erase)				
	Block address (Table 20-7)	Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)			
Automatic page pro-gram	PA: Program page address (Setting of the 4th bus write cycle address for page program)				
	Page address				Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)
Protect bit program	PBA: Protect bit address (Setting of the 7th bus write cycle address for protect bit program)				
	Flash area	Fix to "0"		Protect bit selection (Table 20-8)	Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)

20.2.6.3 Block Address (BA)

Table 20-7 shows block addresses. Specify any address included in the block to be erased in the 6th bus write cycle of the automatic block erase command.

Table 20-7 Block address

Block	Address (User boot mode)	Address (Single boot mode)	Size (Kbyte)
2	0x0001_8000 to 0x0001_FFFF	0x3F81_8000 to 0x3F81_FFFF	32
3	0x0001_0000 to 0x0001_7FFF	0x3F81_0000 to 0x3F81_7FFF	32
1	0x0000_8000 to 0x0000_FFFF	0x3F80_8000 to 0x3F80_FFFF	32
0	0x0000_0000 to 0x0000_7FFF	0x3F80_0000 to 0x3F80_7FFF	32

20.2.6.4 How to Specify Protect Bit (PBA)

The protect bit is specified in 1-bit unit in programming and in 4-bit unit in erasing.

Table 20-8 shows a protect bit selection table of the automatic protect bit program. The column of address example indicates an address described in upper side is used in the use boot mode and the lower side is used in the single boot mode.

Four protect bits are erased by the automatic protect bit erase command in all.

Table 20-8 Protect bit program address

Block	Protect bit	Address of 7th bus write cycle			Address example [31:0]
		Address [14:9]	Address [8]	Address [7]	
Block0	<BLK[0]>	Fix to "0"	0	0	0x0000_0000 0x3F80_0000
Block1	<BLK[1]>		0	1	0x0000_0080 0x3F80_0080
Block2	<BLK[2]>		1	0	0x0000_0100 0x3F80_0100
Block3	<BLK[3]>		1	1	0x0000_0180 0x3F80_0180

20.2.6.5 ID-Read Code (IA, ID)

Table 20-9 shows how to specify a code and the content using ID-Read command.

The column of address example indicates an address described in the upper side is used in the use boot mode and the lower side is used in the single boot mode

Table 20-9 ID-Read Command codes and contents

Code	ID[7:0]	IA[13:12]	Address Example [31:0]
Manufacture code	0x98	0b00	0x0000_0000 0x3F80_0000
Device code	0x5A	0b01	0x0000_1000 0x3F80_1000
-	Reserved	0b10	-
Macro code	0x33	0b11	0x0000_3000 0x3F80_3000

20.2.6.6 Example of Command Sequence

(1) use boot mode

Command		Bus cycle						
		1	2	3	4	5	6	7
Read	Address	0x0000_0000	–	–	–	–	–	–
	Data	0x0000_00F0	–	–	–	–	–	–
Read/reset	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	–	–	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_00F0	–	–	–	–
ID-Read	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	IA	0x0000_0000	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	–	–
Automatic page program	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	PA	In the following cycles, write addresses and data successively per page.		
	Data	0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
Automatic chip erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	–
Automatic block erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	BA	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	–
Automatic protect bit program	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	PBA
	Data	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A
Automatic protect bit erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550
	Data	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A

(2) Data single boot mode

Command		Bus cycle						
		1	2	3	4	5	6	7
Read	Address	0x3F80_0000	–	–	–	–	–	–
	Data	0x0000_00F0	–	–	–	–	–	–
Read/reset	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	–	–	–	–
	Data	0x0000_00AA	0x3F80_0055	0x3F80_00F0	–	–	–	–
ID-Read	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	IA	0x0000_0000	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	–	–
Automatic page program	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	PA	In the following cycles, write addresses and data successively per page.		
	Data	0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
Automatic chip erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	–
Automatic block erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	BA	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	–
Automatic protect bit program	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	PBA
	Data	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A
Automatic protect bit erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550
	Data	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A

20.2.7 Flowchart

20.2.7.1 Automatic Program

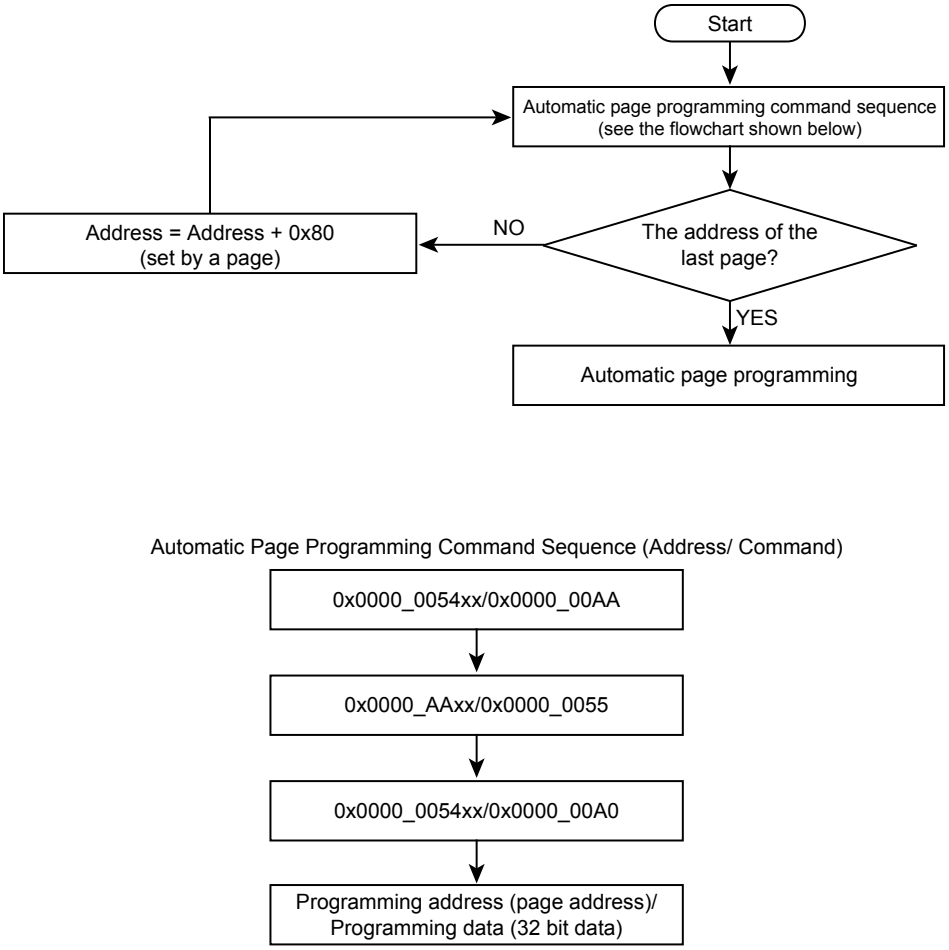


Figure 20-4 Flowchart of automatic program

20.2.7.2 Automatic Erase

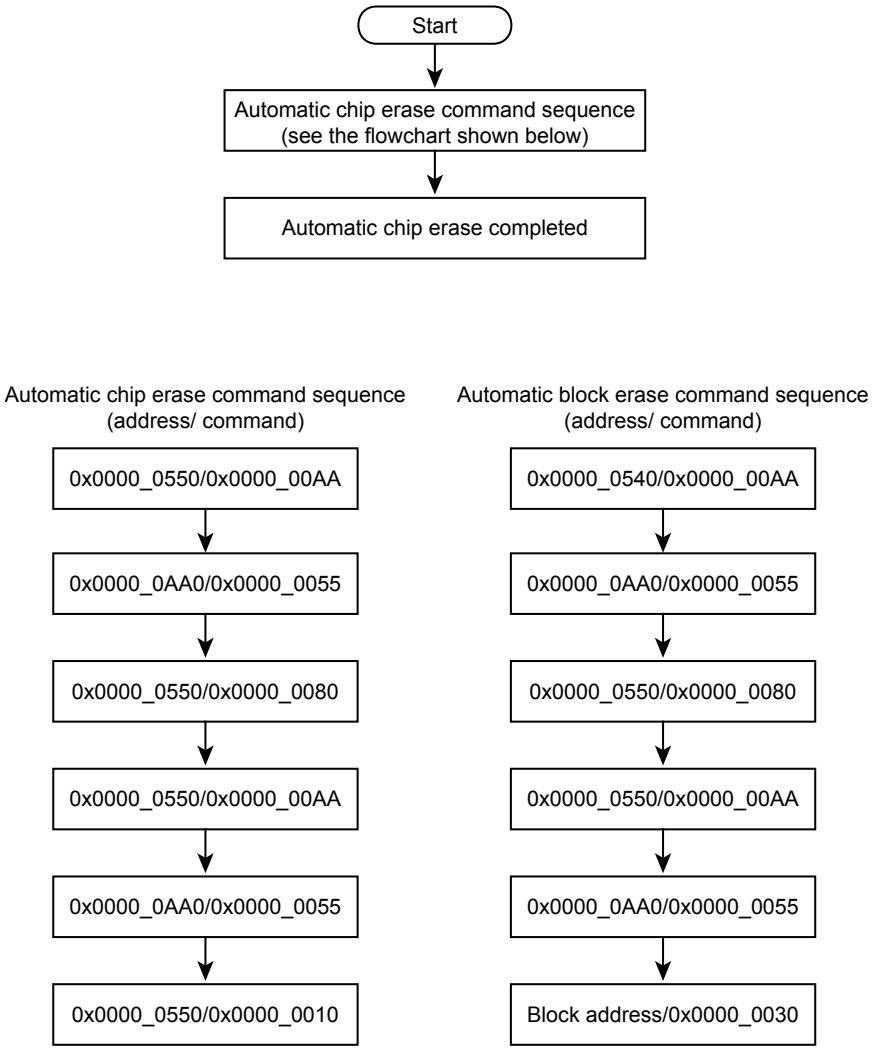


Figure 20-5 Flowchart of automatic erase

20.3 How to Reprogram Flash using Single Boot Mode

The single boot mode utilizes a program contained in built-in BOOT ROM for reprogramming Flash memory. In this mode, BOOT ROM is mapped to the area containing interrupt vector tables and Flash memory is mapped to another address area other than BOOT ROM area.

In the boot mode, Flash memory is reprogrammed using serial command/data transfer. With connecting serial channel (SIO/UART) of this device to the external host, a reprogramming program is copied from the external host to the built-in RAM. A reprogramming routine in the RAM is executed to reprogram Flash memory. For details of communication with host, follow the protocol described later.

Even in the single boot mode, do not generate interrupt/fault except reset to avoid abnormal program termination.

To secure the contents of Flash memory in the single chip mode (normal operation mode), once re-programming is complete, it is recommended to protect relevant flash blocks against accidental erasure during subsequent single chip operations.

20.3.1 Mode Setting

In order to execute the on-board programming, this device is booted-up in the single boot mode. Below setting is for the single boot mode setting.

$\overline{BOOT} = 0$
$\overline{RESET} = 0 \rightarrow 1$

While $\overline{BOOT}$ pin is set to the above in advance, set $\overline{RESET}$ pin to "0". Then release $\overline{RESET}$ pin, the device will boot-up in the single boot mode.

20.3.2 Interface Specification

This section describes SIO/UART communication format in the single boot mode. The serial operation supports both UART (asynchronous communication) and I/O interface modes. In order to execute the on-board programming, set the communication format of the programming controller as well.

- UART communication

  Communication channel: channel

  Serial transfer mode: UART (asynchronous), half-duplex, LSB first

  Data length: 8-bit

  Parity bit: None

  STOP bit: 1-bit

  Baud rate: Arbitrary baud rate

- I/O interface mode

  Communication channel: channel

  Serial transfer mode: I/O interface, full-duplex, LSB first

  Synchronous signal (SCLK): Input mode, rising edge setting

  Handshaking signal: PE4 (output mode)

  Baud rate: Arbitrary baud rate

The boot program operates the clock/mode control block setting as an initial condition. For detail of the initial setting of the clock, refer to "Clock/Mode control".

As explained in the "20.3.5.1 Serial Operation Mode Determination", a baud rate is determined by the 16-bit timer (TMRB). When determining the baud rate, communication is executed by 1/16 of a desired baud rate. Therefore, the communication baud rate must be within the measurable range. The timer count clock operates at ΦT1 (fc/2).

A handshaking pin of I/O interface mode outputs "Low" waiting in receive state and outputs "High" in transmission state. Check the handshaking pin before communications and must follow the communication protocol.

Table 20-10 shows the pins used in the boot program. Other than these pins are not used by the boot program.

Table 20-10 Pin connection

Pin		Interface	
		UART	I/O interface mode
Mode setting pin	$\overline{\text{BOOT}}$	o	o
Reset pin	$\overline{\text{RESET}}$	o	o
Communication pin	TXD (PE2)	o	o
	RXD (PE1)	o	o
	SCLK (PE0)	×	o (Input mode)
	PE4	×	o (Output mode)

o:used ×:unused

20.3.3 Restrictions on Internal Memories

Note that the single boot mode places restrictions on the built-in RAM and built-in flash memory as shown in Table 20-11.

Table 20-11 Restrictions on the memories in the single boot mode

Memory	Restrictions
Internal RAM	Boot program uses the memory as a work area through 0x2000_0000 to 0x2000_03FF. Store the program 0x2000_0400 through the end address of RAM. The start address of the program must be even address.
Internal flash memory	The following addresses are assigned for storing software ID information and passowerds. Storing program in below addresses is not recommendable. 0x3F80_FFF0 to 0x3F80_FFFF

Note: If a password is erased data (0xFF), it is difficult to protect data secure due to an easy-to-guess password. Even if the single boot mode is not used, it is recommended to set a unique value as a password.

20.3.4 Operation Command

The boot program provides the following operation commands.

Table 20-12 Operation command data

Operation command data	Operation mode
0x10	RAM transfer
0x40	Flash memory chip erase and protect bit erase

20.3.4.1 RAM Transfer

The RAM transfer is to store data from the controller to the built-in RAM. When the transfer is complete normally, a user program starts. User program can use the memory address of 0x2000_0400 or later except 0x2000_0000 to 0x2000_03FF for the boot program. CPU will start execution from RAM store start address. The start address must be even address.

This RAM transfer function enables user-specific on-board programming control. In order to execute the on-board programming by a user program, use Flash memory command sequence explained in 20.2.6.

20.3.4.2 Flash Memory Chip Erase and Protect Bit Erase

Flash memory chip erase and protect bit erase commands erase the entire blocks of Flash memory and write/erase protects of all blocks regardless of write/erase protect or security status.

20.3.5 Common Operation regardless of Command

This section describes common operation under the boot program execution.

20.3.5.1 Serial Operation Mode Determination

When the controller communicates via UART, set the 1st byte to 0x86 at the desired baud rate. When the controller communicate via I/O interface mode, set the 1st byte to 0x30 at 1/16 of the desired baud rate. Figure 20-6 shows waveforms in each case.

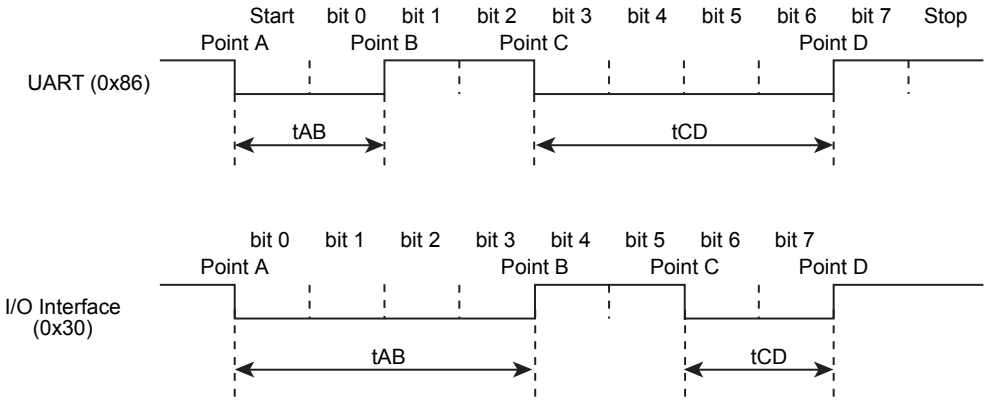


Figure 20-6 Serial operation mode determination data

Figure 20-7 shows a flowchart of boot program. Using 16-bit timer (TMRB) with the time of tAB, tAC and tAD, the 1st byte of serial operation mode determination data (0x86, 0x30) after reset is provided. In Figure 20-7, the CPU monitors level of the receive pin, and obtains a timer value at the moment when the receive pin's level is changed. Consequently, the timer values of tAB, tAC and tAD have a margin of error. In addition, note that if the transfer goes at a high baud rate, the CPU may not be able to determine the level of receive pin. In particular, I/O Interface tends to generate this problem since its baud rate is generally much higher than those of UART. To avoid this, the controller should send data at 1/16 of the desired baud rate in the I/O interface mode.

The flowchart in Figure 20-8 shows the serial operation mode is determined that the time length of the receive pin is long or short. If the length is tAB ≤ tCD, the serial operation mode is determined as UART mode. The time of tAD is used whether the automatic baud rate setting is enable or not. If the length is tAB > tCD, the serial operation mode is determined as I/O Interface mode. Note that timer values of

tAB, tAC and tAD have a margin of error. If the baud rate is high and operation frequency is low, each timer value becomes small. This may generates unexpected determination occurs. (To prevent this problem, re-set UART within the programming routine.)

For example, When UART mode is utilized, the controller should allow for a time-out period where the time is expected to receive an echo-back (0x86) from the target board. The controller should give up the communication if it fails to get that echo-back within the allowed time.

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 where the time is expected to receive an echo-back (0x86) from the target board. The controller should give up the communication if it fails to get that echo-back within the allowed time. 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, it is not necessary that the first byte is 0x30 as long as tAB >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 > tCD is established and SIO is selected by the resolution of the operation mode determination, the second byte code is 0x30 even though the transmitted code on the first byte is not 0x30 (The first byte code to determine I/O interface mode is described as 0x30).

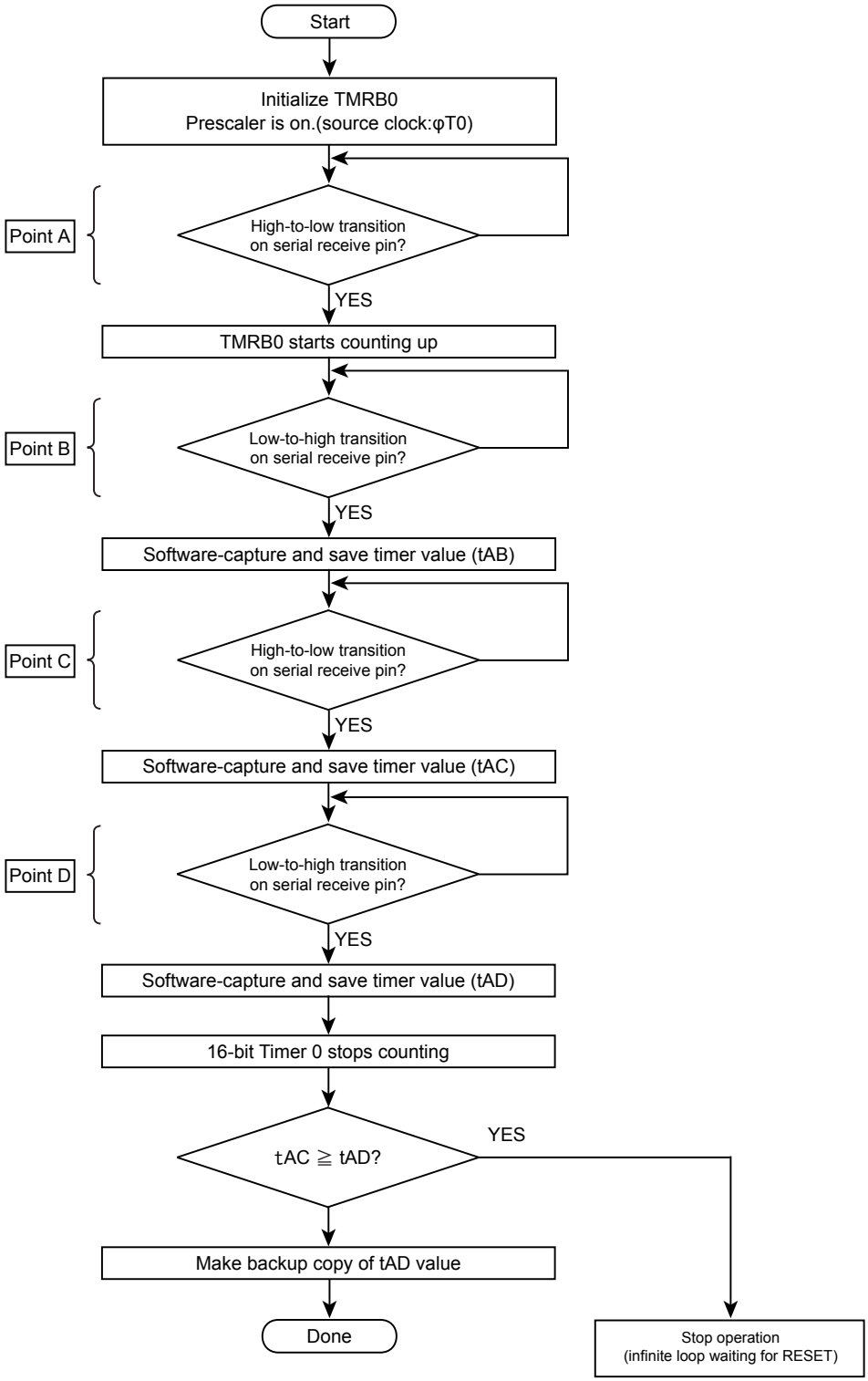


Figure 20-7 Serial operation mode receive flowchart

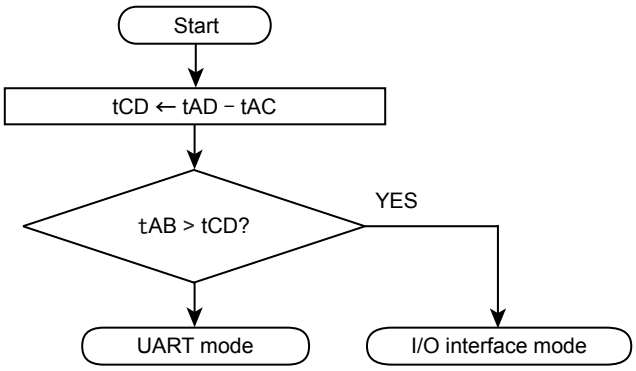


Figure 20-8 Serial operation mode determination flowchart

20.3.5.2 Acknowledge Response Data

The boot program represents processing states in specific codes and sends them to the controller. Table 20-13 to Table 20-16 show the values of acknowledge responses to each receive data.

In Table 20-14 to Table 20-16, the upper four bits of the acknowledge response are equal to those of the operation command data. The 3rd bit indicates a receive error. The 0th bit indicates an invalid operation command error, a checksum error or a password error. The 1st bit and 2nd bit are always "0". Receive error checking is not performed in I/O Interface mode.

Table 20-13 ACK response to the serial operation determination data

Transmit data	Description
0x86	Determined that UART communication is possible. (Note)
0x30	Determined that I/O interface communication is possible.

Note: When the serial operation is determined as UART, if the baud rate setting is determined as unacceptable, the boot program aborts without sending back any response.

Table 20-14 ACK response to the operation command data

Transmit data	Description
0x?8 (Note)	A receive error occurs in the operation command data
0x?1 (Note)	An undefined operation command data is received normally.
0x10	Determined as a RAM transfer command
0x40	Determined as a flash memory chip erase command

Note: The upper 4 bits of the ACK response data are the same as those of the previous command data.

Table 20-15 ACK response to the CHECK SUM data

Transmit data	Description
0xN8 (Note)	A receive error occurs.
0xN1 (Note)	A CHECK SUM or a password error occurs.
0xN0 (Note)	The CHECK SUM value is correct.

Note: The upper 4 bits of the ACK response data are the same as those of the operation command data.

Table 20-16 ACK response to Flash memory chip erase and protect bit erase operation

Transmit data	Description
0x54	Determined as a erase enable command
0x4F	Erase command is complete.
0x4C	Erase command is abnormally terminated.

Note: Even when an erase command is performed normally, a Negative acknowledge may be returned by ACK response. Check the FCSR<RDY_BSY> to make sure the command sequence end, and then hold for 200 μ s or more, after that reconfirm the erase status.

20.3.5.3 Password Determination

The boot program use the below area to determine whether a password is required or use as a password.

Area	Address
Password requirement determination	0x3F81_FFF0 (1byte)
Password area	0x3F81_FFF4 to 0x3F81_FFFF (12byte)

The RAM Transfer command performs a password verification regardless of necessity judging data. Flash memory chip erase or protect bit erase command performs a password verification only when necessity judging is determined as "required".

Password requirement setting	Data
Need password	Other than 0xFF
No password	0xFF

If a password is set to 0xFF (erased data), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.

(1) Password verification using RAM transfer command

If all these address locations contain the same bytes of data other than 0xFF, this condition is determined as a password area error as shown in Figure 20-9. In this case, the boot program returns an error acknowledge (0x11) in response to the 17th byte of checksum value regardless of the password verification.

The boot program verifies 5th byte to 16th byte of receive data (password data). A password error occurs if all 12 bytes do not match. If the password error is determined, an ACK response data to the 17th of CHECK SUM data is a password error.

The password verification is performed even if the security function is enabled.

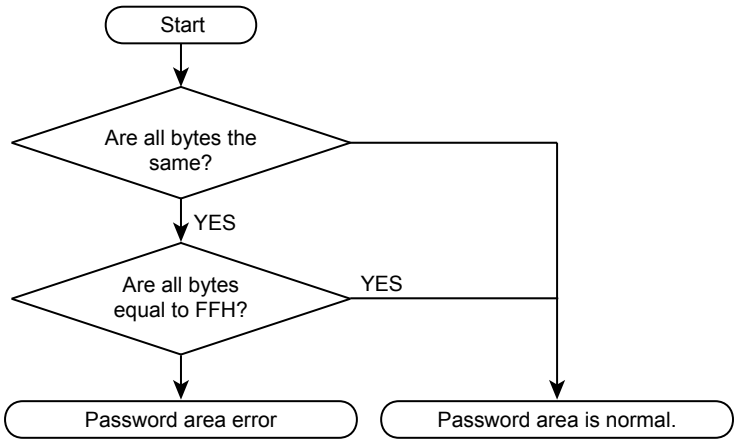


Figure 20-9 Password area check flowchart

(2) Password verification to Flash memory chip erase and protect bit erase command

When a password is enable in the erase password necessity determination area as shown in Figure 20-10 and the passwords are identical data, a password area error occurs. If a password area error is determined, an ACK response to the 17th byte of CHECK SUM sends 0x41 regardless of the password verification.

The boot program verifies 5th byte to 16th byte of receive data (password data). A password error occurs if all 12 bytes do not match. If the password error is determined, an ACK response data to the 17th of CHECK SUM data is a password error.

The password verification is performed even if the security function is enabled.

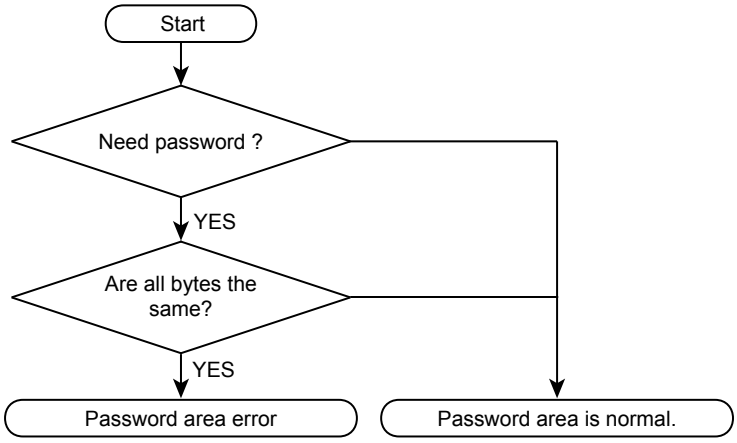


Figure 20-10 Password area check flowchart

20.3.5.4 CHECK SUM Calculation

The checksum is calculated by 8-bit addition to transmit data, dropping the carries, and taking the two's complement of the total sum. The controller must perform the same checksum operation in transmitting checksum bytes.

Example of CHECK SUM

To calculate the checksum for a series of 0xE5 and 0xF6, perform 8-bit addition.

0xE5 + 0xF6 = 0x1DB

Take the two's complement of the sum to the lower 8-bit, and that is a checksum value. So the boot program sends 0x25 to the controller.

0 − 0xDB = 0x25

20.3.6 Transfer Format at RAM Transfer

This section shows a RAM transfer command format. Transfer directions in the table are indicated as follows:

Transfer direction (C→T): Controller to TMPM066/067/068FW

Transfer direction (C←T): TMPM066/067/068FW to Controller

Number of transfer bytes	Transfer direction	Transfer data	Description
1	C→T	Serial operation mode and baud rate setting	Sends data to determine the serial operation mode. For detail of mode determination, refer to "20.3.5.1 Serial Operation Mode Determination".
		[UART mode] 0x86	Sends 0x86. If UART mode is determined, the program determines whether a baud setting is possible. If not, the program stops and communication is shutdown.
		[I/O interface mode] 0x30	Sends 0x30 at 1/6 of desired baud rate.The 2nd byte is also sent at 1/6 of desired baud rate. From the 3rd byte or later, you can send the data at a desirable baud rate.
2	C←T	ACK response to serial operation mode	The 2nd byte of transmit data is a ACK response data to the 1st byte that corresponds to the serial operation setting mode data. If the setting is possible, sets SIO/UART. A receive enable timing is set before transmit buffer is written to the data.
		[UART mode] Normal state: 0x86	If the setting is determined to be possible, sends 0x86. If not, the operation aborts without sending back any response. When the controller finished to send the 1st byte of data, requires a time-out time (5 seconds). If data (0x86) is not normally received within a time-out time, communication is not possible.
		[I/O interface mode] Normal state:0x30	Writes data (0x30) to the transmit buffer and waits for SCLK0 clock. When the controller finished to send the 1st byte of data and several ms (idle time) later, outputs SCLK clock. At this time, the baud rate is set to 1/16 of desired baud rate. If receive data is 0x30, communications are possible. After the 3rd byte, sets a desired baud rate to communicate.
3	C→T	Operation command data (0x10)	Sends RAM transfer command data (0x10).
4	C←T	ACK response to operation command Normal state: 0x10 Abnormal state: 0xX1 Communication error: 0xX8	ACK response data to the operation command. First, checks if 3rd byte of receive data has errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command.) Note that in the I/O interface, receive error check is not performed. Then, if the 3rd byte of receive data corresponds to either operation command data in Table 20-12, receive data is echoed back. In the case of RAM transfer, 0x10 is echoed back and the transfer data branches to the RAM transfer service routine. If the data does not correspond to the command in Table 20-12, sends a ACK response data 0xX1 that means operation command errors, and waits for next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command data.)
5 to 16	C→T	Password data (12-byte) 0x3F810_FFF4 to 0x3F810_FFFF	Checks data in the password area. For detail of password area checking, refer to "20.3.5.3 Password Determination". Compares 5th to 16th byte of receive data with 0x3F810_FFF0 to 0x3F810_FFFF of data of Flash memory. If the data does not match the address, a password error flag is set.
17	C→T	5th to 16th byte of CHECK SUM values	Send 5th to 16th byte of CHECK SUM values. For detail of CHECK SUM calculation, refer to 20.3.5.4.

Number of transfer bytes	Transfer direction	Transfer data	Description
18	C←T	ACK response to CHECK SUM value Normal state: 0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 5th to 17th byte of receive data have errors.(UART mode only) If receive errors exist, sends a ACK response data 0x18 that means abnormal communications and waits for a next operation command (3rd byte). Then checks 17th byte of CHECK SUM data. If errors exist, sends 0x11 and waits for a next operation command (3rd byte). Finally, checks the result of password verification. If a password error exists, sends a ACK response data 0x11 that means a password error and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
19	C→T	RAM store start Address 31 to 24	Sends a start address of block transfer for RAM store. The 19th byte corresponds to 31st to 24th bit of address.The 22nd byte corresponds to 7th to 0th bit of address. Specify the address to the address 0x2000_0400 through the last address of RAM. The address must be even address.
20	C→T	RAM store start Address 23 to 16	
21	C→T	RAM store start Address 15 to 8	
22	C→T	RAM store start Address 7 to 0	
23	C→T	Number of RAM store bytes 15 to 8	Set the number of bytes to perform block transfer. The 23rd byte corresponds to the15th bit to 8th bit of transfer bytes. The 24th byte corresponds to 7th bit to 0th bit of transfer bytes. Specify the data to be stored in the address from 0x2000_0400 through the last address of RAM.
24	C→T	Number of RAM store bytes 7 to 0	
25	C→T	19th to 24th byte of CHECK SUM value	Send 19th byte to 24th byte of CHECK SUM values
26	C←T	ACK response to CHECK SUM value Normal state: 0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 19th byte to 25th byte of receive data have errors.(UART mode only) If receive errors exist, sends a ACK response data 0x18 that means abnormal communications and waits for a next operation command (3rd byte). Then checks 25th byte of CHECK SUM data. If errors exist, sends 0x11 and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
27 to m	C→T	RAM stored data	Sends same bytes of data specified in 23th bytes to 24 byte for RAM stored data.
m+1	C→T	27 to m byte of CHECK SUM value	Sends 27th byte to m byte of CHECK SUM value
m+2	C←T	ACK response to CHECK SUM value Normal state:0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 27th byte to m+1 byte of receive data have errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Then checks m+1 byte of CHECK SUM data, if errors exist, sends 0x11 and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
–	–	–	If m + 2nd byte of ACK response data is normal ACK response data, the transfer data branches to the address specified in 19th byte to 22 byte.

20.3.7 Transfer Format of Flash memory Chip Erase and Protect Bit Erase

This section shows a transfer format of Flash memory chip erase and protect bit erase commands. Transfer directions in the table are indicated as follows:

Transfer direction (C→T): Controller to TMPM066/067/068FW

Transfer direction (C←T): TMPM066/067/068FW to Controller

Number of transfer bytes	Transfer direction	Transfer data	Description
1	C→T	Serial operation mode and baud rate setting	Sends data to determine the serial operation mode. For detail of mode determination, refer to"20.3.5.1 Serial Operation Mode Determination".
		[UART mode] 0x86	Sends 0x86. If UART mode is determined, checks if baud rate setting can be done. If not, operation stops communications.
		[I/O interface mode] 0x30	Sends 0x30 at 1/16 of desired baud rate. Same as the 1st byte, sends the 2nd byte at 1/16 of desired baud rate. Desired baud rate can be used after 3rd byte.
2	C←T	ACK response to serial operation mode	The 2nd byte of transmit data is a ACK response data to the 1st byte that corresponds to the serial operation setting mode data. If the setting is possible, sets SIO/UART. A receive enable timing is set before transmit buffer is written to the data.
		[UART mode] Normal state: 0x86	If the setting is determined to be possible, sends 0x86. If not, the operation aborts without sending back any response. When the controller finished to send the 1st byte of data, requires a time-out time (5 seconds). If data (0x86) is not normally received within a time-out time, communication is not possible.
		[I/O interface mode] Normal state:0x30	Writes data (0x30) to the transmit buffer and waits for SCLK0 clock. When the controller finished to send the 1st byte of data and several ms (idle time) later, outputs SCLK clock. At this time, the baud rate is set to 1/16 of desired baud rate. If receive data is 0x30, communications are possible. After the 3rd byte, sets a desired baud rate to communicate.
3	C→T	Operation command data (0x40)	Sends Flash memory chip erase and protect bit erase command data (0x40).
4	C←T	ACK response to the operation command Normal state: 0x40 Abnormal state: 0xX1 Communication error: 0xX8	ACK response data to the operation command. First, checks if 3rd byte of receive data has errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command data.) Note that in the I/O interface, receive error check is not performed. Then, if the 3rd byte of receive data corresponds to either operation command data in Table 20-12, receive data is echoed back. If the data does not correspond to the command in Table 20-12, sends a ACK response data 0xX1 that means operation command errors, and waits for next operation command. (3rd byte) Upper 4 bits of transmit data are undefined. (Upper 4 bits of immediate before operation command data are used.)
5 to 16	C→T	Password data (12-byte) 0x3F810_FFF4 to 0x3F810_FFFF	If password necessity is set to "none", this data is dummy data. If password necessity is set to "necessary", checks data in the password area. For a method of password area data checking, refer to "20.3.5.3 Password Determination". Compares 5th to 16th byte of receive data with 0x3F810_FFF0 to 0x3F810_FFFF of data of Flash memory in order. If the data does not match, a password error flag is set.
17	C→T	5th to 16th byte CHECK SUM value	Sends 5th byte to 16 byte of CHECK SUM value. For a method of CHECK SUM calculation, refer to"20.3.5.4 CHECK SUM Calculation".

Number of transfer bytes	Transfer direction	Transfer data	Description
18	C←T	ACK response to the CHECK SUM value Normal state: 0x40 Abnormal state: 0x41 Communication error: 0x48	If password necessity is set to "none", sends a normal ACK response data 0x40. If password necessity is set to "necessary", first checks if receive errors exist in the 5th byte to 17th byte receive data. (UART mode only) If a receive error exists, sends a ACK response data 0x48 that means abnormal communications and waits for next operation command. (3rd byte) Then checks 17th byte of CHECK SUM data. If error occurs, sends 0x41 and waits for a next operation command (3rd byte) Finally, checks the result of password verification. If a password error exists, sends a ACK response data 0x41that means a password error and waits for a next operation command (3rd byte) If all procedure normally ends, sends a normal ACK response data 0x40.
19	C→T	Erase enable command data (0x54)	Sends an enable command data (0x54).
20	C←T	ACK response to the erase enable command Normal state: 0x54 Abnormal state: 0xX1 Communication error: 0x58	First, checks if 19th byte of receive data has errors. If receive errors exist, sends a ACK response data (bit 3) 0x58 that means abnormal communication and waits for next operation command (3rd byte). Then, if 19th byte of receive data corresponds to the erase enable command, receive data is echoed back (normal ACK response data). In this case, 0x54 is echoed back and the transfer data branches into Flash memory chip erase process routine. If the data does not correspond to the erase enable command, sends a ACK response data (bit 0) 0xX1 and waits for next operation command. Upper 4 bits of transmit data are undefined. (Upper 4 bits of immediate before operation command data are used.)
21	C→T	ACK response to the erase command (note1) Normal state: 0x4F Abnormal state: 0x4C	If the operation is normally complete, the end code (0x4F) is returned. If erase error occurs, an error code (0x4C) is returned.
–	–	–	Waits for a next operation command.

Note 1: Even when an erase command is performed normally, a Negative acknowledge may be returned by ACK response. Check the FCSR<RDY_BSY> to make sure the command sequence end, and then hold for 200 μ s or more, after that reconfirm the erase status.

20.3.8 Boot Program Whole Flowchart

This section shows a boot program whole flowchart.

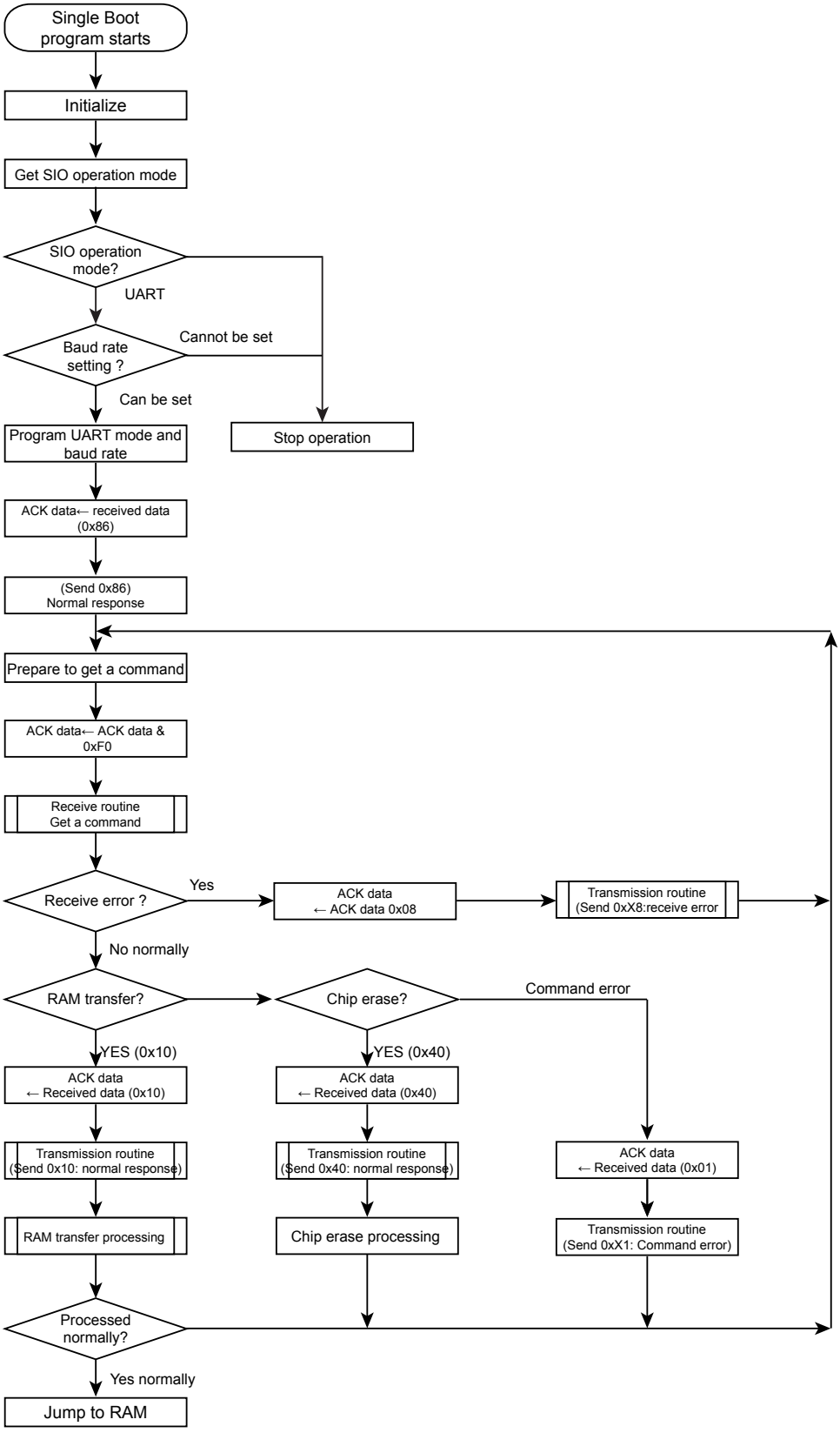


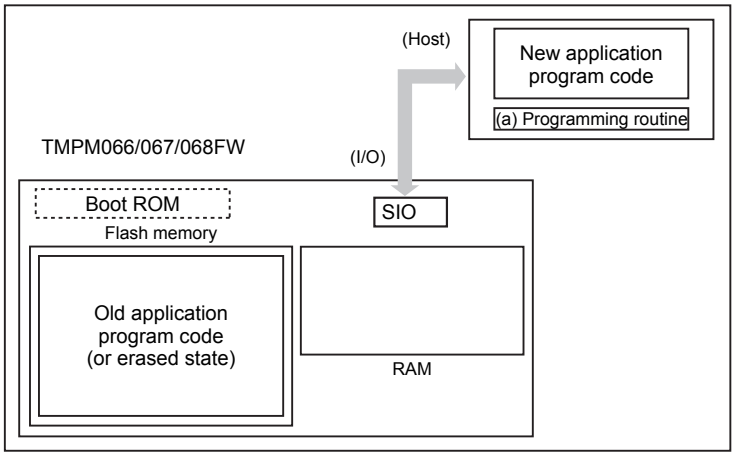
Figure 20-11 Boot program whole flowchart

20.3.9 Reprogramming Procedure of Flash using reprogramming algorithm in the on-chip BOOT ROM

This section describes the reprogramming procedure of the flash using reprogramming algorithm in the on-chip boot ROM.

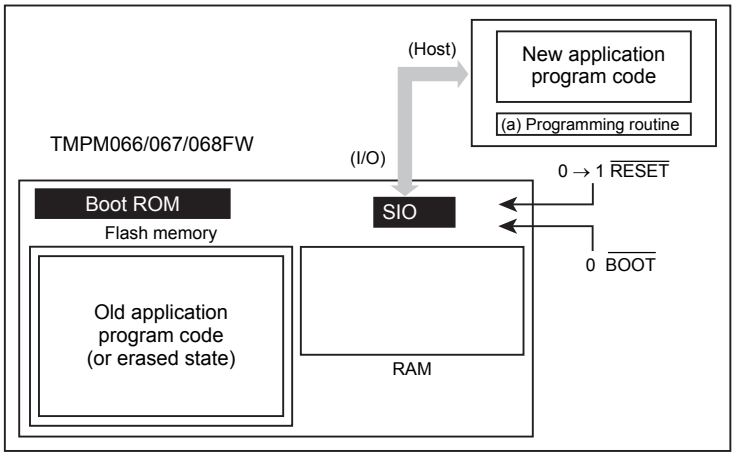
20.3.9.1 Step-1

The condition of Flash memory does not care whether a user program made of former versions has been written or erased. Since a programming routine and programming data are transferred via the SIO (SIO), the SIO must be connected to an external host. A programming routine (a) is prepared on the host.



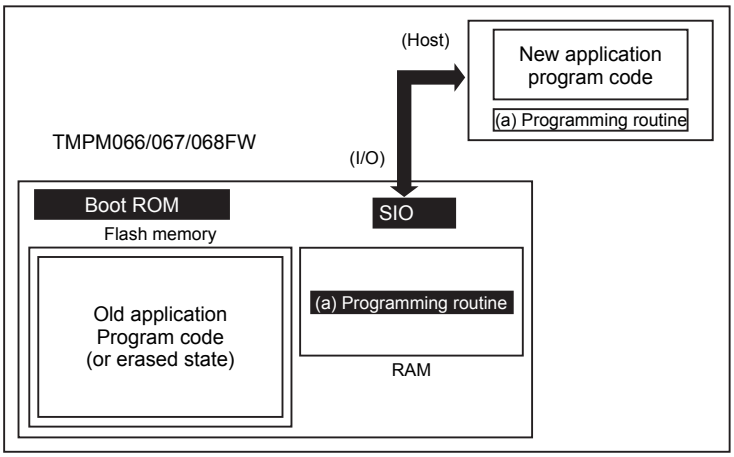
20.3.9.2 Step-2

Release the reset by pin condition setting in the boot mode and boot-up the BOOT ROM. According to the procedure of boot mode, transfer the programming routine (a) via SIO from the source (host). A password verification with the password in the user application program is performed. (If Flash memory is erased, an erase data (0xFF) is dealt with a password.)



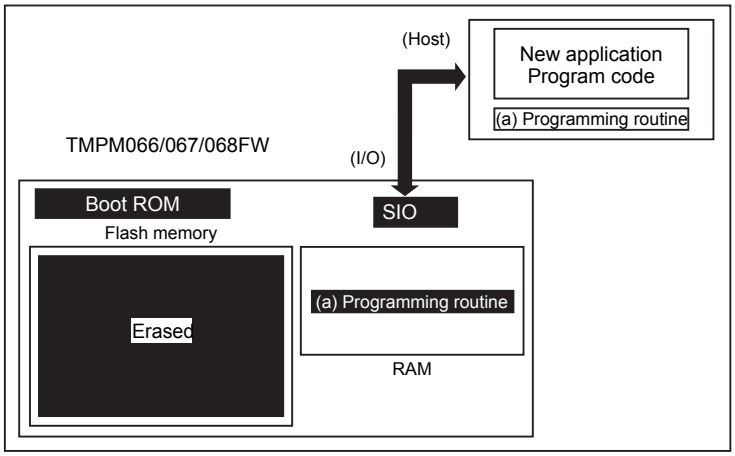
20.3.9.3 Step-3

If the password verification is complete, the boot program transfer a programming routine (a) from the host into the on-chip RAM. The programming routine must be stored in the range from 0x2000_0400 to the end address of RAM.



20.3.9.4 Step-4

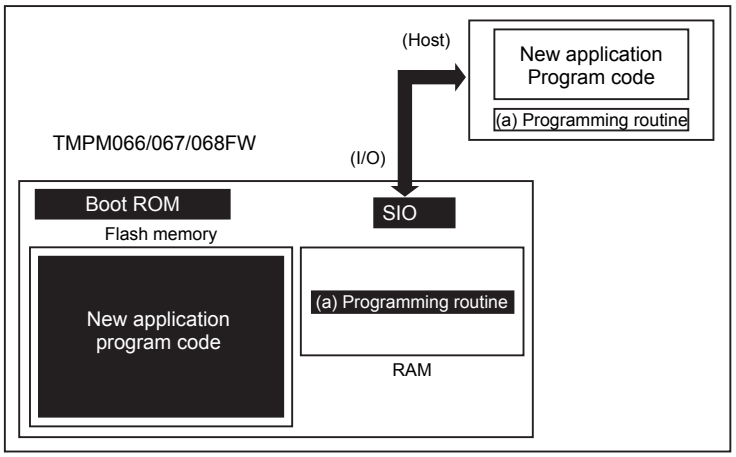
The boot program jumps to the programming routine (a) in the on-chip RAM to erase the flash block containing old application program codes. The Block Erase or Chip Erase command is be used.



20.3.9.5 Step-5

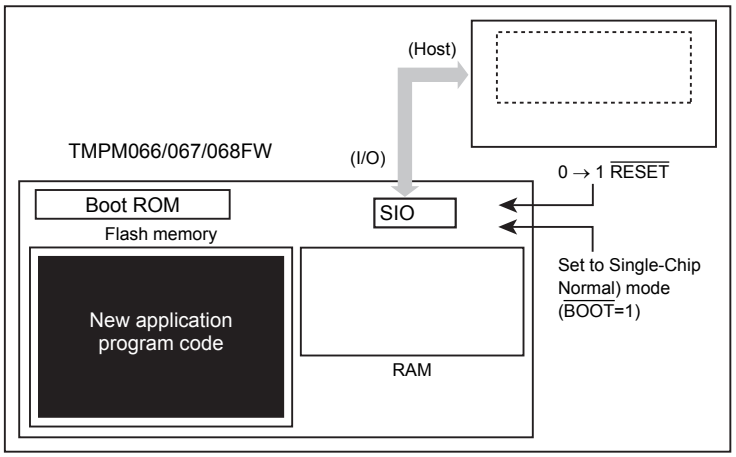
The boot program executes the programming routine (a) to download new application program codes from the host and programs it into the erased flash area. When the programming is complete, the writing or erase protection of that flash area in the user's program must be set.

In the example below, new program code comes from the same host via the same SIO channel as for the programming routine. However, once the programming routine has begun to execute, it is free to change the transfer path and the source of the transfer. Create a hardware board and programming routine to suit your particular needs.



20.3.9.6 Step-6

When programming of Flash memory is complete, power off the board and disconnect the cable leading from the host to the target board. Turn on the power again so that the device re-boots in the single-chip (Normal) mode to execute the new program.



20.4 Programming in the User Boot Mode

A user Boot mode is to use flash memory programming routine defined by users. It is used when the data transfer buses for flash memory program code on the user application is different from the serial I/O. It operates in the single chip mode; therefore, a switch from normal mode in which user application is activated in the use boot mode to the user boot mode for programming flash is required. Specifically, add a mode judgment routine to the reset service routine in the user application program.

The condition to switch the modes needs to be set according to the user's system setup condition. Also, a flash memory programming routine that the user uniquely makes up needs to be set in the new application. This routine is used for programming after being switched to the user boot mode. The data in built-in Flash memory cannot be read out during erase/reprogramming mode. Thus, reprogramming routine must be take place while it is stored in the area outside of Flash memory area. Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental reprogramming. Be sure not to generate interrupt/fault except reset to avoid abnormal termination during the user boot mode.

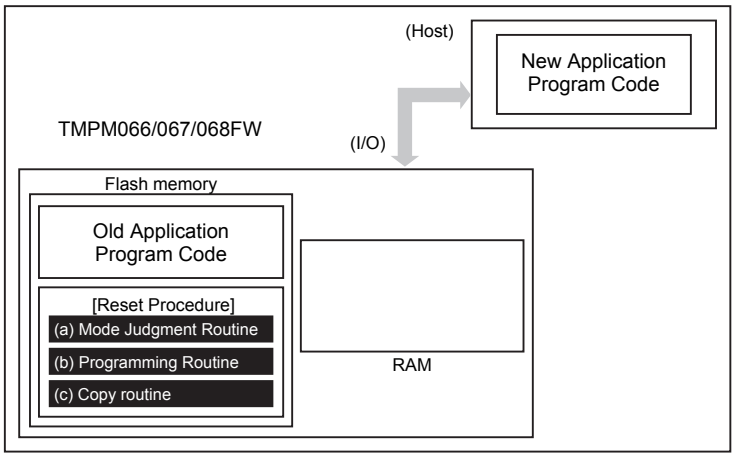
Taking examples from two cases such as the method that reprogramming routine stored in Flash memory (1-A) and transferred from the external device (1-B), the following section explains the procedure. For a detail of the program/erase to Flash memory, refer to "20.2 Detail of Flash Memory".

20.4.1 (1-A) Procedure that a Programming Routine Stored in Flash memory

20.4.1.1 Step-1

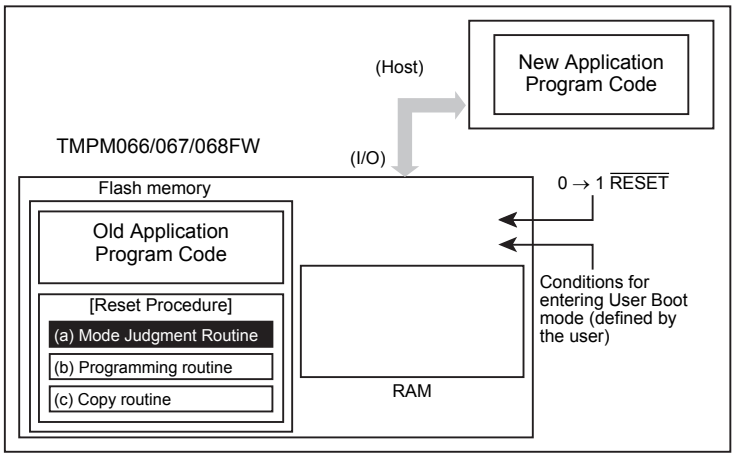
A user determines the conditions (e.g., pin status) to enter the user boot mode and the I/O bus to be used to transfer data. Then suitable circuit design and program are created. Before installing the device on a printed circuit board, write the following three program routines into an arbitrary flash block using programming equipment such as a flash writer.

(a) Mode determination routine:	A program to determine to switch to user boot mode or not
(b) Flash programming routine:	A program to download new program from the host controller and re-program Flash memory
(c) Copy routine:	A program to copy the data described in (a) to the built-in RAM or external memory device



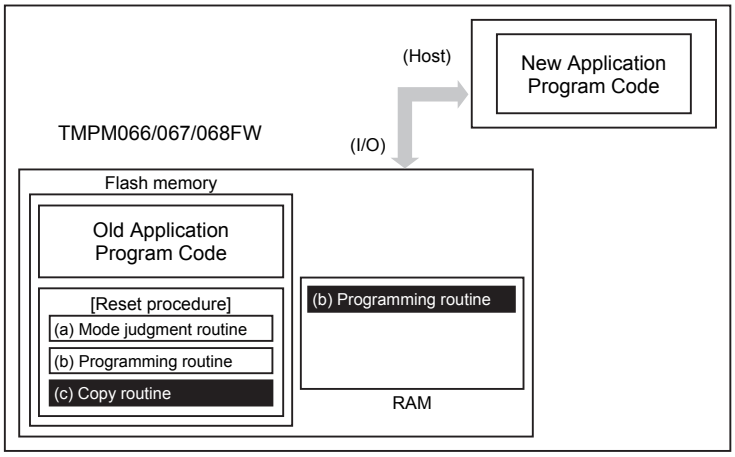
20.4.1.2 Step-2

This section explains the case that a programming routine storied in the reset routine. First, the reset routine determines to enter the user boot mode. If mode switching conditions are met, the device enters the user boot mode to reprogram data.



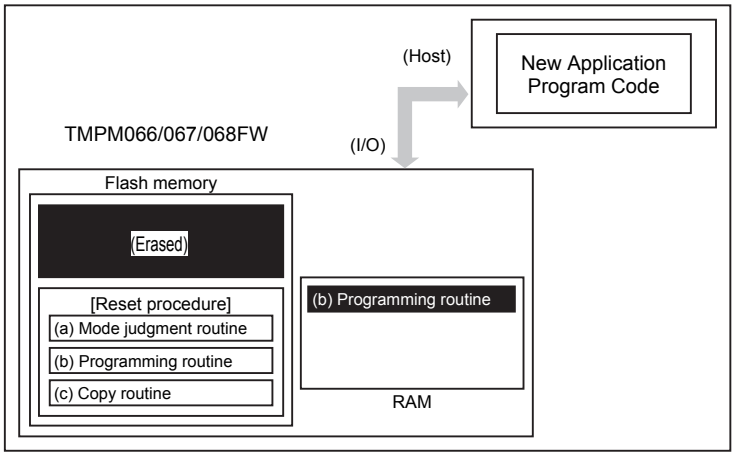
20.4.1.3 Step-3

Once the device enters the user boot mode, execute the copy routine (C) to download the flash programming routine (b) from the host controller to the built-in RAM.



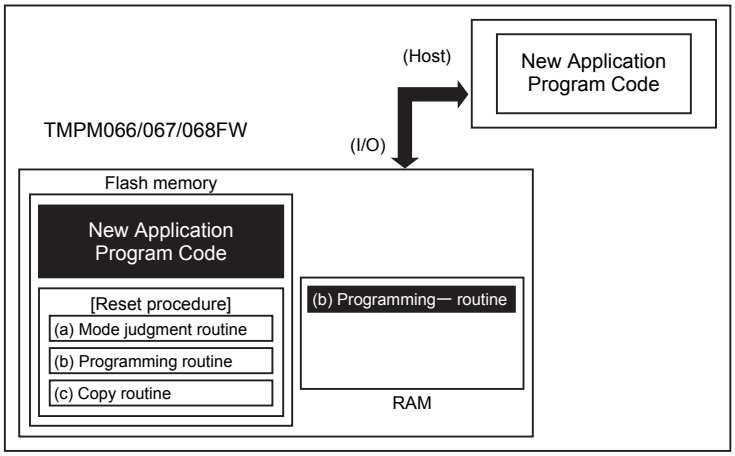
20.4.1.4 Step-4

Jump to the reprogramming routine in the built-in RAM to release the write/erase protection for the old application program, and to erase a flash in block unit.



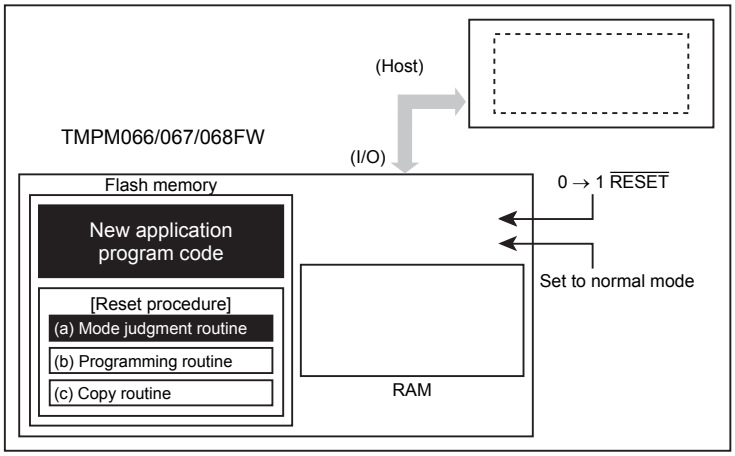
20.4.1.5 Step-5

Continue to execute the flash programming routine to download new program data from the host controller and program it into the erased flash block. When the programming is complete, the write/erase protection of that flash block in the user program area must be set.



20.4.1.6 Step-6

Set $\overline{\text{RESET}}$ to "0". Upon reset, Flash memory is set to the normal mode. After reset, the CPU will start along with the new application program.



20.4.2 (1-B) Procedure that a Programming Routine is transferred from External Host

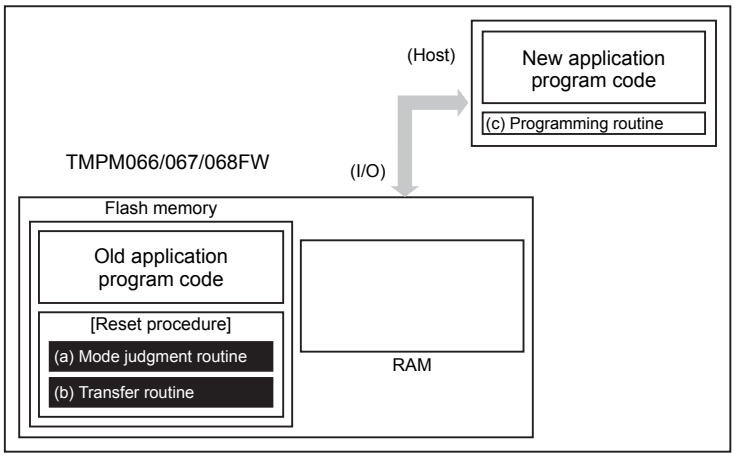
20.4.2.1 Step-1

A user determines the conditions (e.g., pin status) to enter the user boot mode and the I/O bus to be used to transfer data. Then suitable circuit design and program are created. Before installing the device on a printed circuit board, write the following two program routines into an arbitrary flash block using programming equipment such as a flash writer.

(a) Mode determination routine:	A program to determine to switch to reprogramming operation
(b) Transfer routine:	A program to obtain a reprogramming program from the external device.

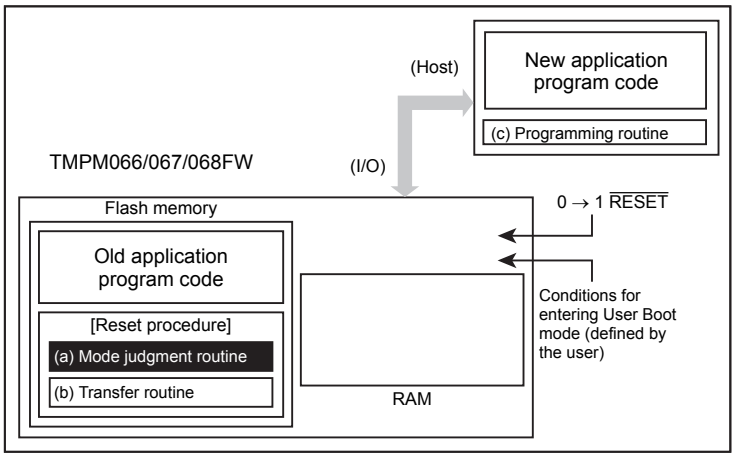
In addition, prepare a reprogramming routine shown below must be stored on the host controller.

(c) Reprogramming routine:	A program to reprogram data



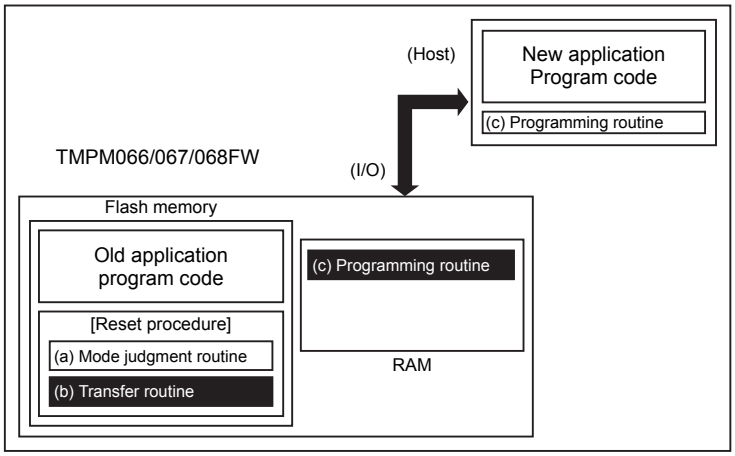
20.4.2.2 Step-2

This section explains the case that a programming routine storied in the reset routine. First, the reset routine determines to enter the user boot mode. If mode switching conditions are met, the device enters the user boot mode to reprogram data.



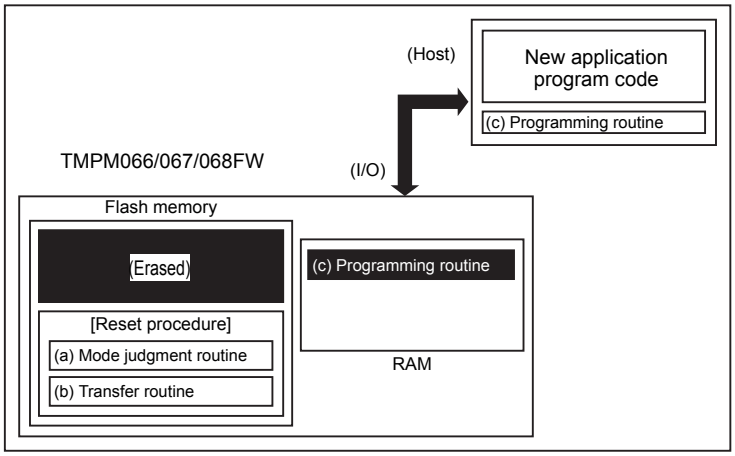
20.4.2.3 Step-3

Once the device enters the user boot mode, execute the transfer routine (b) to download the programming routine (c) from the host controller to the built-in RAM.



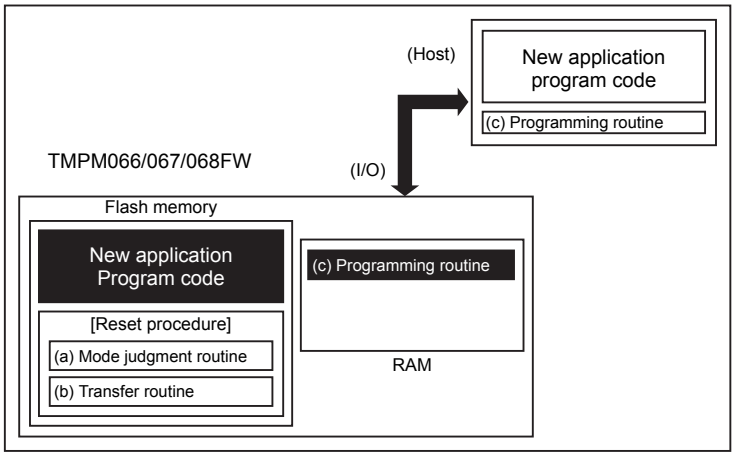
20.4.2.4 Step-4

Jump to the reprogramming routine in the built-in RAM to release the write/erase protection for the old application program, and to erase a flash in block unit.



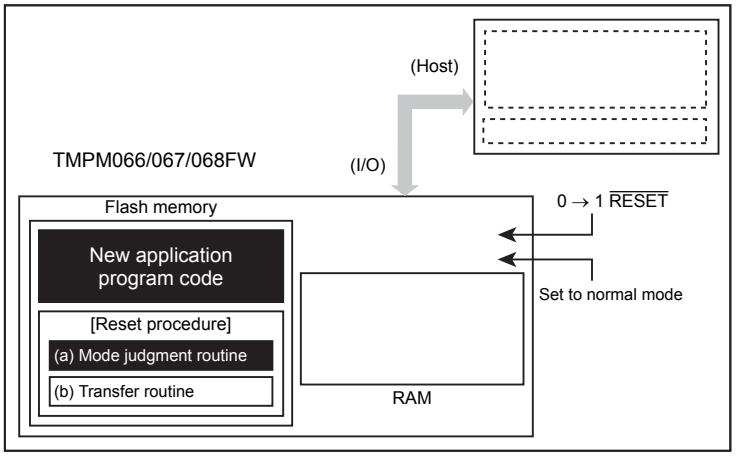
20.4.2.5 Step-5

Continue to execute the flash programming routine (c) to download new program data from the host controller and program it into the erased flash block. When the programming is complete, the write/erase protection of that flash block in the user program area must be set.



20.4.2.6 Step-6

Set $\overline{\text{RESET}}$ to "0". Upon reset, Flash memory is set to the normal mode. After reset, the CPU will start along with the new application program.



21. Debug Interface

21.1 Specification Overview

TMPM066/067/068FW contains the Serial Wire JTAG Debug Port (SWJ-DP) unit for interfacing with the debugging tools .

For details about SWJ-DP, refer to "Arm documentations set for the Cortex-M0".

21.2 SWJ-DP

SWJ-DP supports the Serial Wire Debug Port (SWCLK, SWDIO) .

Pin name	Function	Description	I/O
SWDIO	SW	Serial Wire Data Input/Output	I/O
SWCLK	SW	Serial Wire Clock	Input

21.3 Peripheral Functions in Halt Mode

When the Cortex-M0 core enters in the halt mode, the watchdog-timer (WDT) automatically stops. It is selectable that 16-bit Timer(TMRB and TMR16A) continue or stop counting.Other peripheral functions continue to operate.

21.4 Connection with a Debug Tool

21.4.1 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 mode is prohibited.

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

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

After releasing reset, the particular pins of the debug interface pins are initialized as the debug interface pins. The other debug interface pins should be changed to the debug interface pins if needed.

If the debug interface pins are used as the general I/O port, please prepare the way to change the general I/O port to the debug interface pins beforehand.

Table 21-1 Example Table of using debug interface pins

	Debug interface pins	
	SWCLK	SWDIO
SW	o	o

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

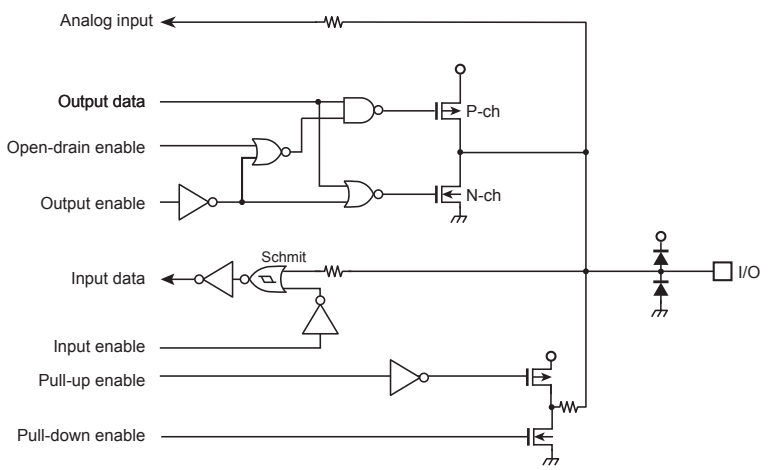
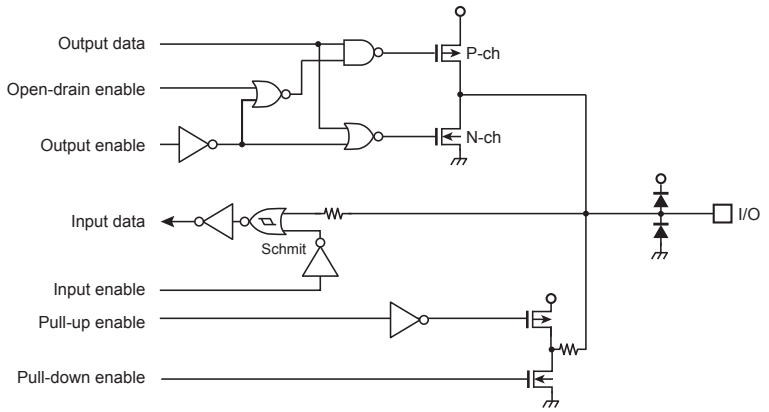
22. Port Section Equivalent Circuit Schematic

Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

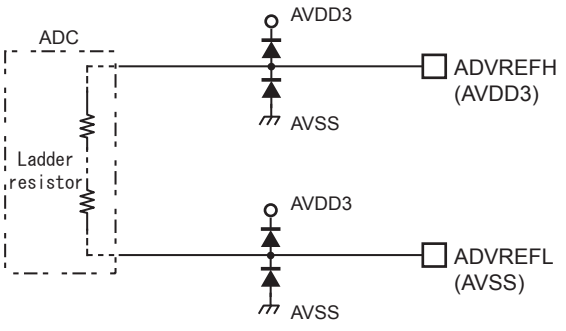
The input protection resistance ranges from several tens of Ω to several hundreds of Ω. Damping resistors X2 are shown with a typical value.

Note: Resistors without values in the figure show input protection resistors.

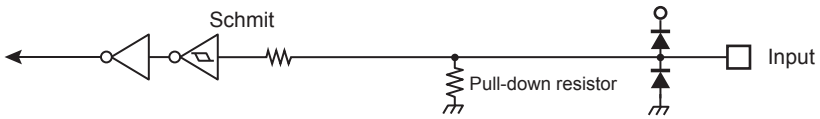
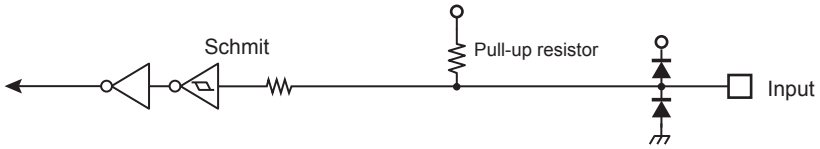
22.1 PORT pin

PA0 to PA7	
PB0 to PB3, PC0 to PC5, PD0 to PD5, PE0 to PE5, PF0 to PF73, PG0 to PG1, PH0 to PH3, PJ0 to PJ3	

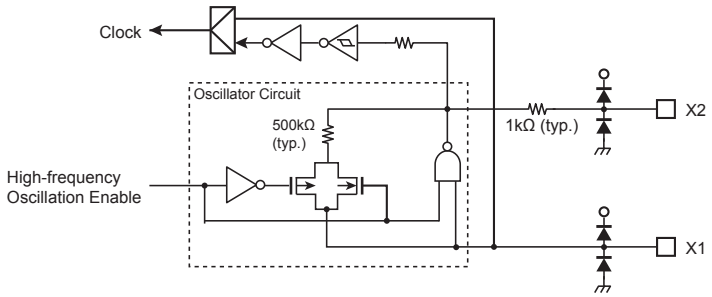
22.2 Analog pin

VREFH VREFL	

22.3 Control pin

MODE	 Mode pin must be connected to GND.
$\overline{\text{RESET}}$	

22.4 Clock pin

X1, X2	

23. Electrical Characteristics

23.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		DVDD3	−0.3 to 3.9	V
		RVDD3	−0.3 to 3.9	
		AVDD3	−0.3 to 3.9	
input voltage	Digital input pins	V_{IN1}	−0.3 to DVDD3 + 0.3	V
	Analog input pins	V_{IN2}	−0.3 to AVDD3 + 0.3	
Low-level Output current	Per pin (except the following)	I_{OL1}	5	mA
	PC2,PC3,PG6,PG7	I_{OL2}	20	
	Total	ΣI_{OL}	75	
High-level output current	Per pin (except the following)	I_{OH1}	−5	
	PC2,PC3,PG6,PG7	I_{OH2}	−20	
	Total	ΣI_{OH}	-75	
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		T_{OPR}	−40 to 85	°C

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 blow up and/or burning.**

23.2 DC Electrical Characteristics (1/2)

DVSS = RVSS = AVSS = 0V
Ta = −40 to 85 °C

Parameter		Symbol	Condition	Min	Typ.	Max	Unit
Supply voltage	DVDD3 RVDD3 AVDD3	DVDD3 RVDD3 AVDD3	f_{OSC} = 8 to 16 MHz fsys = 1 to 24 MHz	1.8	−	3.6	V
Low-level Input voltage	PB0 to 3,PC0 to 5,PD0 to 5, PE0 to 5,PF0 to PF7,PG0,1, PH0 to 3,PJ 0 to 3(note1)	V_{IL1}	Schmitt	−0.3	−	0.2 DVDD3	V
	PA0 to 7	V_{IL2}				0.2 AVDD3	
	X1, MODE, $\overline{RESET}$	V_{IL3}				0.2 DVDD3	
High-level Input voltage	PB0 to 3,PC0 to 5,PD0 to 5, PE0 to 5,PF0 to PF7,PG0,1, PH0 to 3,PJ 0 to 3(note1)	V_{IH1}	Schmitt	0.8 DVDD3	−	DVDD3+0.3	V
	PA0 to 7	V_{IH2}		0.8 AVDD3		AVDD3+0.3	
	X1, MODE, $\overline{RESET}$	V_{IH3}		0.8 DVDD3		DVDD3+0.3	
Low-level output voltage	PAx,PBx,PCx,PDx,PEx, PF4 to 7,PGx,PH4 to 7x,PJx (note1)	V_{OL1}	I_{OL1} = 2 mA 1.8 ≤ DVDD3 ≤ 3.6V	−	−	0.4	V
	PF0,PF1,PF2,PF3	V_{OL2}	I_{OL2} = 10 mA 1.8 ≤ DVDD3 ≤ 3.6V			0.4	
High-level output voltage	PAx,PBx,PCx,PDx,PEx, PF4 to 7,PGx,PH4 to 7x,PJx (note1)	V_{OH1}	I_{OH1} = −2 mA 1.8 ≤ DVDD3 ≤ 3.6V	DVDD3-0.4	−	DVDD3	V
	PF0,PF1,PF2,PF3	V_{OH2}	I_{OH2} = −10 mA 1.8 ≤ DVDD3 ≤ 2.7V	DVDD3-0.4	−	DVDD3	

Note 1: There is a terminal which is not mounted with a product. Refer to the pin list for details.

DVSS = RVSS = AVSS = 0V
Ta = −40 to 85 °C

Parameter		Symbol	Condition	Min	Typ. (Note1)	Max	Unit
Input leakage current	−	I_{LI1}	0.0 ≤ V_{IN} ≤ DVDD3 0.0 ≤ V_{IN} ≤ AVDD3	−	0.02	±5	μA
	PC0, PC1,PG0,PG1(when power off)	I_{LI2}	0.0 ≤ V_{IN} ≤ DVDD3 0 ≤ DVDD3 ≤ 0.2	−	−	±1.8	
Output leakage current		I_{LO}	0.2 ≤ V_{IN} ≤ DVDD3 -0.2 0.2 ≤ V_{IN} ≤ AVDD3 -0.2	−	0.05	±10	
Schmitt trigger input width		VTH1	2.7 V ≤ DVDD3 ≤ 3.6 V	0.1 DVDD3	−	−	V
		VTH2	1.8 V ≤ DVDD3 ≤ 2.7 V	0.1 DVDD3	−		
Pull-up resistor at Reset		RRST	1.8 V ≤ DVDD3 ≤ 3.6 V	25	50	75	kΩ
Programmable pull-up/pull-down resistor		PKH1	2.7 V ≤ DVDD3 ≤ 3.6 V	25	50	75	kΩ
		PKH2	1.8 V ≤ DVDD3 < 2.7 V	25	50	120	
Power supply variation rate in operation range		VRS	RVDD3 = DVDD3	−	−	10	mV/μs
		VFS		−	−	-10	
Pin capacitance (Except power supply pins)		C_{IO}	fc = 1 MHz	−	−	10	pF
Low-level output current		I_{OL1}	Per pin: PAx,PBx,PCx,PDx,PEx, PF4to7,PGx,PHx,PJx, 1.8 V ≤ DVDD3 ≤ 3.6 V	−	−	2	mA
		I_{OL2}	Per pin: PF0,PF1,PF2,PF3 1.8 V ≤ DVDD3 ≤ 3.6 V	−	−	10	mA
		ΣI_{OL1}	Per port:PA	−	−	10	mA
		ΣI_{OL2}	Per area: PCx,PFx	−	−	32	mA
		ΣI_{OL3}	Per area: PBx,PDx,PEx,PGx,PHx, PJx	−	−	32	mA
High-level output current		I_{OH1}	Per pin: PAx,PBx,PCx,PDx,PEx, PF4to7,PGx,PHx,PJx, 1.8 V ≤ DVDD3 ≤ 3.6 V	−	−	-2	mA
		I_{OH2}	Per pin: PF0,PF1,PF2,PF3 1.8 V ≤ DVDD3 ≤ 3.6 V	−	−	-10	mA
		ΣI_{OH1}	Per port :PA	−	−	-10	mA
		ΣI_{OH2}	Per area: PCx,PFx,	−	−	-32	mA
		ΣI_{OH3}	Per area: PBx,PDx,PEx,PGx,PHx, PJx	−	−	-32	mA
output current		ΣI_O	total, all ports	−	−	± 60	mA

Note 1: Ta = 25 °C, DVDD3 = RVDD3 = AVDD3 = 3.3 V, unless otherwise noted.
Note 2: The same voltage must be supplied to DVDD3, RVDD3, AVDD3.

23.3 DC Electrical Characteristics (2/2)

Ta = −40 to 85 °C

Parameter		Symbol	condition			Min	Typ. (Note)	Max	Unit
			Operation Voltage	High-speed oscillation	Operation condition				
NORMAL		I_{DD}	DVDD3 = RVDD3 = AVDD3 = 3.6V	Enabled	All peripheral function included with CPU	−	−	32	mA
	NORMAL1		DVDD3 = RVDD3 = AVDD3 = 3.3V	Refer toTable 23-1 regrading to the operation condition		−	28	30	
	NORMAL2					−	13	14	
IDLE						−	7	8	mA
STOP1				Disabled	Refer toTable 23-1 regrading to the operation condition	−	50	320	μA

Note: Ta = 25 °C, DVDD3 = RVDD3 = AVDD3 = 3.3 V, unless otherwise noted.

Table 23-1 I_{DD}Measurement Condition (Pin condition, Oscillator)

			NORMAL1	NORMAL2	IDLE	STOP1
Pin condition	DVDD3 = RVDD3 = AVDD3		3.3V			
	X1, X2 pin		Connected to the high-speed oscillator (10MHz)			
	Input pin		Fixed			
	Output pin		Open			
Operation condition (Oscillator)	System clock (fsys)		24MHz			Disabled
	External high-speed oscillator (EHOSC)		Enabled			Disabled
	Internal high-speed oscillator (IHOSC)		Disabled			
	PLL for fsys		Enabled (6 multiplying, fosc=16MHz)			Disabled
	CPU		Running			Disabled
	peripheral function	USB	Enabled & Run	Disabled	Disabled	Disabled
		others	Enabled & Run	Enabled & Run	Disabled	Disabled

23.4 10-bit AD Converter electrical Characteristics

AVSS = DVSS = 0V, Ta = -40 to 85 °C

Parameter		Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage (+)		AVDD	−	1.8	2.7	3.6	V
Analog input voltage		VAIN	−	AVSS	−	AVDD	V
Power supply current of analog reference voltage	AD conversion	IREF	DVSS = AVSS	−	0.45	0.7	mA
	Non-AD conversion			−	15	50	μA
INL error		−	Conversion time ≥ 16.2 μs AVDD=2.7 to 3.6V	−	4.0	6.0	LSB
DNL error				−	4.0	6.0	
Zero-scale error				−	4.0	6.0	
Full-scale error				−	4.0	6.0	
Total error				−	4.0	6.0	
INL error		−	Conversion time ≥ 32.4 μs AVDD=1.8 to 3.6V	−	4.0	6.0	
DNL error				−	4.0	6.0	
Zero-scale error				−	4.0	6.0	
Full-scale error				−	4.0	6.0	
Total error				−	4.0	6.0	

Note 1: 1LSB = (AVDD − AVSS)/1024 [V]

Note 2: This characteristics is shown in operating only ADC.

23.5 AC Electrical Characteristics

23.5.1 Serial channel (SIO/UART)

23.5.1.1 AC measurement Condition

The AC characteristics data of this chapter is measured under the following conditions.

- Output levels: High = 0.8 × DVDD3, Low = 0.2 × DVDD3
- Input levels: High = 0.8 × DVDD3, Low = 0.2 × DVDD3
- Load capacity: CL = 30pF

23.5.1.2 AC Electrical Characteristics (I/O Interface Mode)

In the table below, the letter x represents the SIO operation clock cycle time which is identical to the fsys cycle time.

(1) SCLK input mode

[Input]

DVDD3=2.7 to3.6V

Parameter	Symbol	Equation		fsys = 24 MHz		Unit
		Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	4x	−	166.7	−	ns
SCLK Clocked Low width (input)	t_{SCL}	4x	−	166.7	−	
SCLK cycle	t_{SCY}	t_{SCH} + t_{SCL}	−	333.3	−	
Valid Data input ← SCLK rise/ fall (Note1)	t_{SRD}	30	−	30	−	
SCLK rise / fall (Note1) → Input Data hold	t_{HSR}	x + 30	−	71.7	−	

[Output]

DVDD3=2.7 to3.6V

Parameter	Sym-bol	Equation		fsys = 24 MHz		Unit
		Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	4x	−	170 (Note3)	−	ns
SCLK Clocked Low width (input)	t_{SCL}	4x	−	170 (Note3)	−	
SCLK cycle	t_{SCY}	t_{SCH} + t_{SCL}	−	340	−	
Output Data ← SCLK rise or fall (Note1)	t_{OSS}	t_{SCY}/2 − 3x− 45	−	0 (Note2)	−	
SCLK rise or fall → Output Data hold (Note1)	t_{OHS}	t_{SCY}/2	−	170	−	

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 mode.

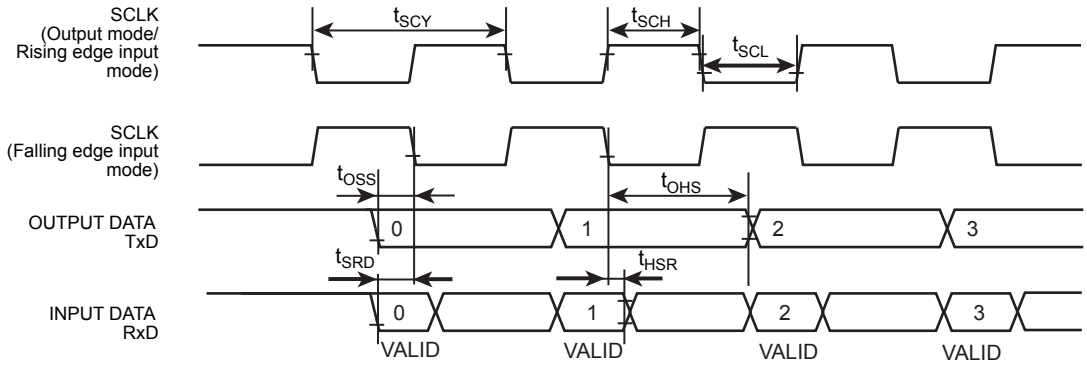
(2) SCLK Output mode

DVDD3=2.7 to3.6V

Parameter	Symbol	Equation		fsys = 24 MHz		Unit
		Min	Max	Min	Max	
SCLK cycle (programmable)	t_{SCY}	4x	–	166.7	–	ns
Output Data ← SCLK rise	t_{OSS}	t_{SCY}/2 - 30	–	53.3	–	
SCLK rise → Output Data hold	t_{OHS}	t_{SCY}/2 - 30	–	53.3	–	
Valid Data Input ← SCLK rise	t_{SRD}	45	–	45	–	
SCLK rise → Input Data hold	t_{HSR}	0	–	0	–	

DVDD3=1.8 to2.7V

Parameter	Symbol	Equation		fsys = 24 MHz		Unit
		Min	Max	Min	Max	
SCLK cycle (programmable)	t_{SCY}	4x	–	170	–	ns
Output Data ← SCLK rise	t_{OSS}	t_{SCY}/2 - 30	–	55	–	
SCLK rise → Output Data hold	t_{OHS}	t_{SCY}/2 - 70	–	15	–	
Valid Data Input ← SCLK rise	t_{SRD}	70	–	90	–	
SCLK rise → Input Data hold	t_{HSR}	0	–	0	–	



23.5.2 I2C interface (I2C)

23.5.2.1 AC measurement Condition

The AC characteristics data of this chapter is measured under the following conditions unless otherwise noted.

- Output levels: High = 0.8 × DVDD3, Low = 0.2 × DVDD3
- Input levels: High = 0.8 × DVDD3, Low = 0.2 × DVDD3
- Load capacity: CL = 30pF

23.5.2.2 AC Electrical Characteristics

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 I2CxCR1.

p denotes the value of p programmed into the PRSCK (dividing clock select) field in the I2CxPRS.

DVDD3=2.7V to 3.6V

Parameter	Symbol	Standard Mode		Fast Mode		Fast Mode +		Unit
		Min	Max	Min	Max	Min	Max	
SCL Clock frequency	f_{SCL}	0	100	0	400	0	1000	kHz
Hold timer for START condition	$t_{HD;\ STA}$	4.0	−	0.6	−	0.26	−	μs
SCL Low width (Input)(Note1)	t_{LOW}	4.7	−	1.3	−	0.5	−	
SCL High width (Input)(Note2)	t_{HIGH}	4.0	−	0.6	−	0.26	−	
Setup time for a repeated START condition	$t_{SU;\ STA}$	4.7	−	0.6	−	0.26	−	
Data hold time (Input) (Note 3, 4)	$t_{HD;\ DAT}$	0.0	−	0.0	−	0.0	−	
Data setup time	$t_{SU;\ DAT}$	250	−	100	−	50	−	ns
Setup time for a STOP condition	$t_{SU;\ STO}$	4.0	−	0.6	−	0.26	−	μs
Bus free time between stop condition an start condition(note5)	t_{BUF}	4.7	−	1.3	−	0.5	−	

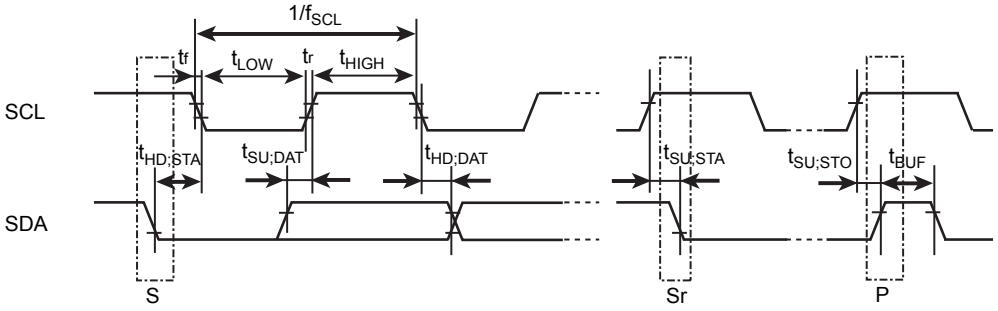
Note 1: SCL clock Low width (output): P × (2^{n-1} + 10)/X (I2CxOP<NFSEL>=0)

Note 2: SCL clock High width (output): P × (2^{n-1} + 6)/X (I2CxOP<NFSEL>=0) 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. (P: depend on I2CxPRS<PRSCK[4:0]>, n:depend on I2CxCR1<SCK[2:0]>)

Note 3: The output data hold time is equal to 4 cycle of Prescaler clock (Tprsck) from the edge 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 I2C 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



S: Start condition
Sr: Repeated-start condition
P: Stop condition

23.5.3 Toshiba serial peripheral interface (TSPI)

23.5.3.1 AC Measurement Condition

The AC characteristics data of this chapter is measured under the following conditions.

- Output levels: High = 0.8 × DVDD3, Low = 0.2 × DVDD3
- Input levels: High = 0.8 × DVDD3, Low = 0.2 × DVDD3
- Load capacity: CL = 30pF TSPIxSCK, TSPIxTXD0 to 3), 30pF (TSPIxCS)

[Slave mode]

DVDD3= 2.7 to 3.6V

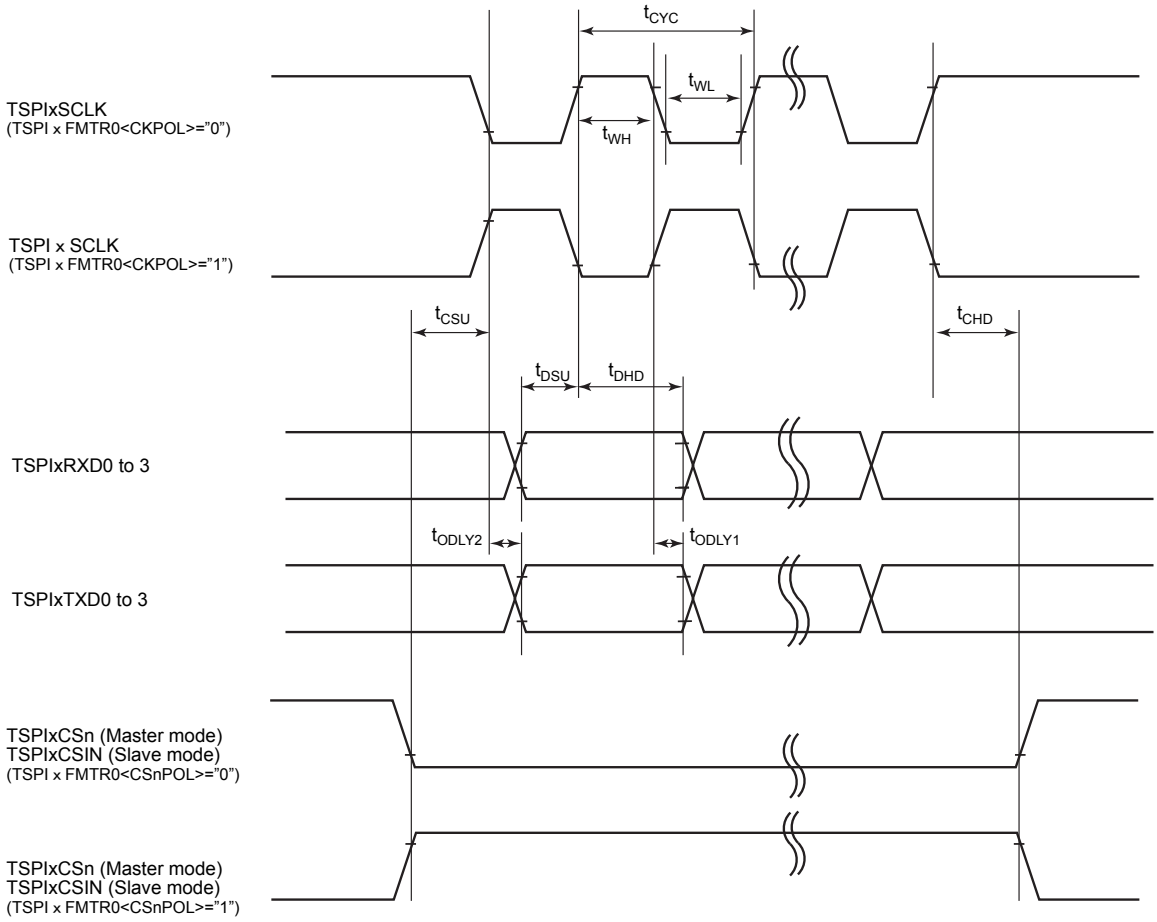
Parameter	Symbol	Equation		Unit
		Min.	Max.	
TSPIxSCLK input frequency	f_{CYC}	−	12	MHz
TSPIxSCLK input cycle	t_{CYC}	83.3	−	ns
TSPIxSCLK Low level input pulse width	t_{WL}	28.7	−	
TSPIxSCLK High level input pulse width	t_{WH}	28.7	−	
TSPIxCSIN valid → TSPIxSCLK fall/rise time	t_{CSU}	63.3	−	
TSPIxSCLK rise/fall → TSPICSIN invalid time	t_{CHD}	5	−	
TSPIxRXD input valid ← TSPISCLK rise/fall data setup time	t_{DSU}	7	−	
TSPIxSCLK fall/rise → TSPIRXD input data hold time	t_{DHD}	10	−	
TSPIxSCLK rise/fall → TSPITXD output data hold time	t_{ODLY1}	0	−	
TSPIxSCLK rise/fall → TSPITXD output data delay	t_{ODLY2}	−	25	

[Master mode]

DVDD3 = 2.7V to 3.6V

Parameter	Symbol	Equation		t_{CYC}=83.3ns	Unit
		Min.	Max.		
TSPIxSCLK input frequency	f_{CYC}	−	12	12	MHz
TSPIxSCLK input cycle	t_{CYC}	83.3	−	83.3	ns
TSPIxSCLK Low level input pulse width	t_{WL}	(t_{CYC}/2) - 13	−	28.7	
TSPIxSCLK High level input pulse width	t_{WH}	(t_{CYC}/2) - 13	−	28.7	
TSPIxCSn valid → TSPIxSCLK fall/rise time	t_{CSU}	t_{CYC}x k1 - 20	−	63.3	
TSPIxSCLK rise/fall → TSPICSn hold time	t_{CHD}	t_{CYC}x k2+0.5 - 20	−	63.3	
TSPIxRXD input valid ← TSPISCLK rise/fall data setup time	t_{DSU}	5	−	5	
TSPIxSCLK fall/rise → TSPIRXD input data hold time	t_{DHD}	fsys +0	−	41.7	
TSPIxSCLK fall/rise → TSPITXD output data hold time	t_{ODLY1}	-10	−	-10	
TSPIxSCLK fall/rise → TSPITXD output data delay	t_{ODLY2}	−	13	13	

Note: The value of k1 is depend on TSPIxFMTR0<CSSCKDL[3:0]> and the value of k2 is depend on TSPIxFMTR0<SCKCSDL[3:0]>.

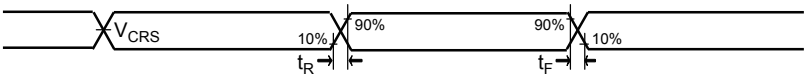


23.5.4 USB Timing (Full-speed)

DVDD3 = 3.0V to 3.45V

Parameter	Symbol	Min	Max	Unit
USB-DP,DM Supply rise time	t_R	4	20	ns
USB-DP,DM Supply fall time	t_F	4	20	
Data Line crossover voltage	V_{CRS}	1.3	2.0	V

USB-DP, DM



23.5.5 16-bit Timer / Event counter (TMRB)

23.5.5.1 Event Counter

(1) AC Measurement Condition

The AC characteristics data of this chapter is measured under the following conditions.

- Input levels: High = 0.8 × DVDD3, Low = 0.2 × DVDD3
- Load capacity: CL = 30pF

(2) AC Electrical Characteristics

In the table below, the letter x represents the TMRB operation clock cycle time. TMRB use the clock cycle as same as one of fsys. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		fsys = 24 MHz		Unit
		Min	Max	Min	Max	
Clock low pulse width	t_{VCKL}	2x + 100	−	183.3	−	ns
Clock high pulse width	t_{VCKH}	2x + 100	−	183.3	−	

23.5.5.2 Capture

(1) AC Measurement Condition

The AC characteristics data of this chapter is measured under the following conditions.

- Input levels: High = 0.8 × DVDD3, Low = 0.2 × DVDD3
- Load capacity: CL = 30pF

(2) AC Electrical Characteristics´

In the table below, the letter x represents the TMRB operation clock cycle time.TMRB use the clock cycle as same as one of fsys. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		fsys = 24 MHz		Unit
		Min	Max	Min	Max	
Low pulse width	t_{CPL}	2x + 100	−	183.3	−	ns
High pulse width	t_{CPH}	2x + 100	−	183.3	−	

23.5.6 High Resolution 16-bit Timer (TMRD Ver.C) PPG output

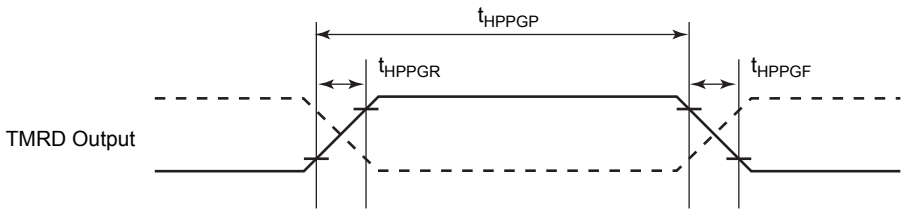
23.5.6.1 AC Measurement Condition

The AC characteristics data of this chapter is measured under the following conditions.

- Output levels: High = 0.8 × DVDD3, Low = 0.2 × DVDD3
- Load capacity: CL = 30pF

DVDD3 = 2.7V to 3.6V

Parameter	Symbol	TMRDCLK = 96 MHz		Unit
		Min.	Max.	
Minimum pulse width of PPG output	t_{HPPGP}	104.2	−	ns
Rising time of PPG output	t_{HPPGR}	−	15	
Falling time of PPG output	t_{HPPGF}	−	15	



23.5.7 External Interrupt

23.5.7.1 AC Measurement Condition

The AC characteristics data of this chapter is measured under the following conditions.

- Input levels: High = 0.8 × DVDD3, Low = 0.2 × DVDD3
- Load capacity: CL = 30pF

23.5.7.2 AC Electrical Characteristics

In the table below, the letter x represents the fsys cycle time.

1. Except STOP1 release interrupt

Parameter	Symbol	Equation		fsys = 24 MHz		Unit
		Min	Max	Min	Max	
low-level pulse width	t_{INTAL}	x + 100	−	141.7	−	ns
High-level pulse width	t_{INTAH}	x + 100	−	141.7	−	

2. STOP1 release interrupt

Parameter	Symbol	Equation		fsys = 24 MHz		Unit
		Min	Max	Min	Max	
low-level pulse width	t_{INTBL}	500	−	500	−	ns
High-level pulse width	t_{INTBH}	500	−	500	−	

23.5.8 Debug Communication

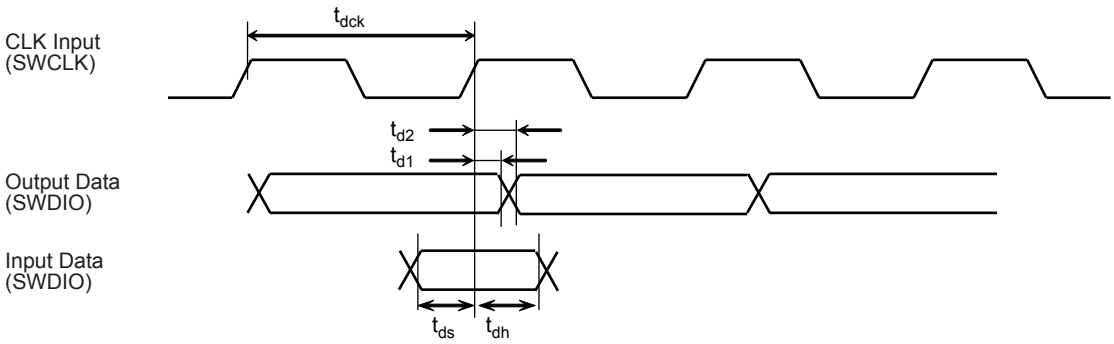
23.5.8.1 AC Measurement Condition

- Output levels: High = 0.8 × DVDD3, Low = 0.2 × DVDD3
- Input levels: Low = 0.8 × DVDD3, Low = 0.2 × DVDD3
- Load capacity: CL = 30pF (SWDIO)

23.5.8.2 SWD interface

DVDD3=1.8V to 3.6V

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	-	



23.5.9 On-chip Oscillator Characteristic

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Oscillation frequency	IHOSC	Ta = -40 to 85°C	9.0	10	11	MHz

Note: Do not use an on-chip oscillator as a system clock (fsys) when high-accuracy oscillation frequency is required.

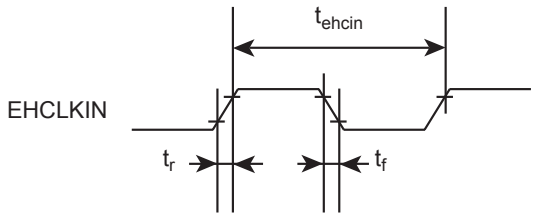
23.5.10 External Oscillator

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
High-frequency oscillation	EHOSC	Ta = -40 to 85°C	8	−	16	MHz

23.5.11 External Clock Input

Parameter	Symbol	Min	Typ.	Max	Unit
External clock frequency(note1)	t_{ehcin}	8	–	to16 and 48 only	MHz
External clock duty	–	45	–	55	%
External clock input rise time	t_r	–	–	10	ns
External clock input fall time	t_f	–	–	10	ns

Note 1: input clock frequency: 8MHz ≤ Clock frequency ≤ 16MHz, Clock frequency = 48MHz.



23.5.12 Flash Characteristic

Parameter	Condition	Min	Typ.	Max	Unit
Guaranteed number of Flash memory programming	DVDD3 = RVDD3 = AVDD3 = 1.8 V to 3.6 V Ta = 0 to 70°C	–	–	100	times

23.5.13 Noise Filter Characteristic

Parameter	Condition	Min	Typ.	Max	Unit
The width of noise filter	–	15	30	60	ns

23.6 Recommended Oscillation Circuit

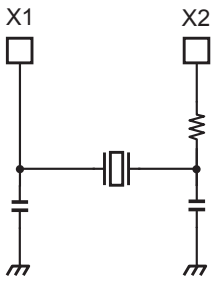


Figure 23-1 High-frequency oscillation connection

Note: To obtain a stable oscillation, load capacity and the position of the oscillator must be configured properly. Since these factors are strongly affected by substrate patterns, please evaluate oscillation stability using the substrate you use.

The TMPM066/067/068FW has been evaluated by the oscillator vendor below. Please refer this information when selecting external parts.

23.6.1 Ceramic Oscillator

The TMPM066/067/068FW has been evaluated by the the high-frequency ceramic oscillator by Murata Manufacturing Co., Ltd.

Please refer to the Murata Website for details.

23.6.2 Crystal Oscillator

The TMPM066/067/068FW has been evaluated by the low-frequency crystal oscillator by KYOCERA Corporation.

Please refer to the KYOCERA Website for details.

23.6.3 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.

23.7 Handling Precaution

23.7.1 Voltage level of power supply at power-on

The rising of power supply in power-on must be less than the value in below table.

TMPM066/067/068FW has some power supply pin. They must be supplied the power at same time.

Power supply pin =DVDD3, AVDD3, RVDD3
Ta = -40 to 85 °C

Parameter	condition	Min	Typ.	Max	Unit
The rising of power supply in power-on	0V→1.8V to 3.6V	-	-	10	mV/μs

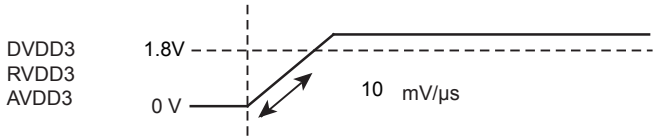


Figure 23-2 Voltage level of power supply at power-on

23.7.2 Voltage drop during operations

When the supply voltage drop occurs during operations, if the supply voltage is below the operation voltage (Brownout), turn on power again.

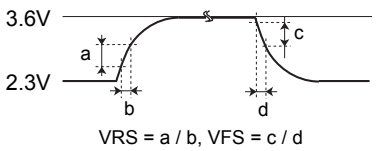
23.7.3 Power supply variation rate in operation range

Malfunction may be caused by voltage fluctuation also in a operation range. In this case, turn on power again.

Power supply pin = DVDD3, AVDD3, RVDD3
Ta = −40 to 85 °C

Parameter	Symbol	Condition	Min	Typ. (Note1)	Max	Unit
Power supply variation rate in operation range	VRS	RVDD3 = DVDD3	−	−	2.25	mV/μs
	VFS		−	−	-1.8	

Note 1: VRS (Rising), VFS (Falling) should be measured at a strict level against a characteristics.



VRS = a / b, VFS = c / d

23.7.4 Operating Voltage on Startup

The Operating lower Voltage of TMPM066/067/068FW is 2.3V, however, the detection voltage of LVD is set to 2.0V ±0.2 on first Startup, Over 2.2V Operating Voltage(DVDD3(=RVDD3)) is required on that time.

After startup, TMPM066/067/068FW can operate from 3.6V to the 1.8V lower Voltage when the LVD has been disabled by a program.

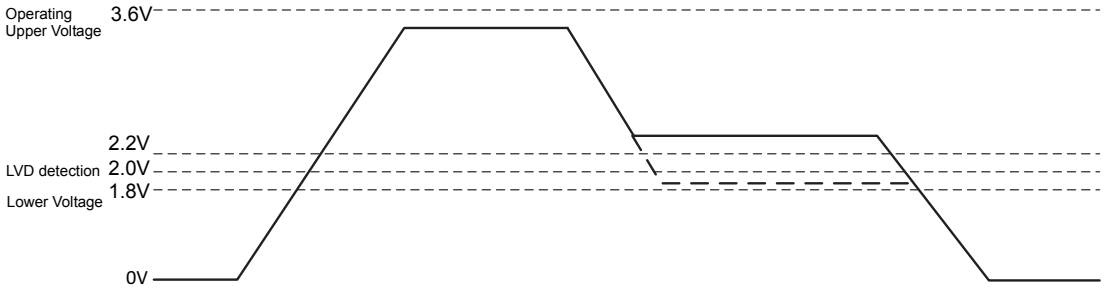


Figure 23-3

24. Package Dimensions

Type : LQFP64-P-1010-0.50E

Unit: mm

Dimensions

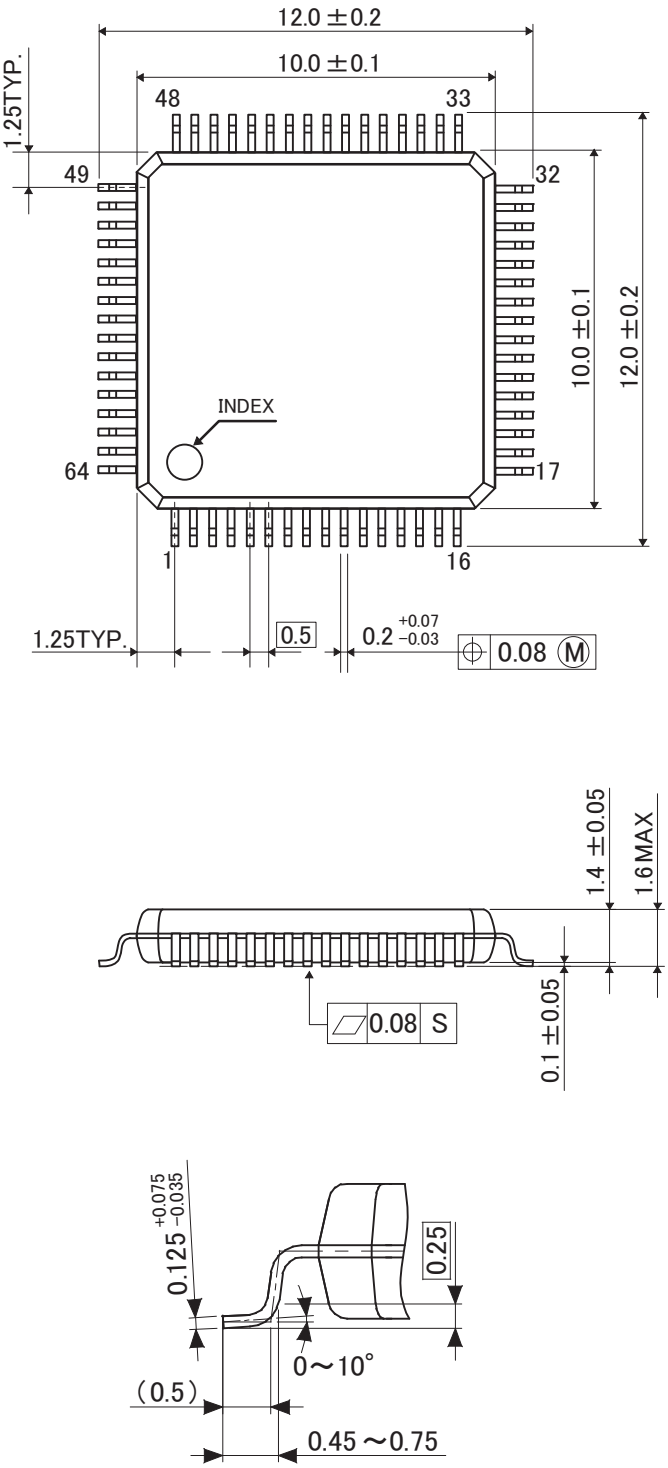


Figure 24-1 TMPM066FWUG

Type : QFN48-P-0707-0.50

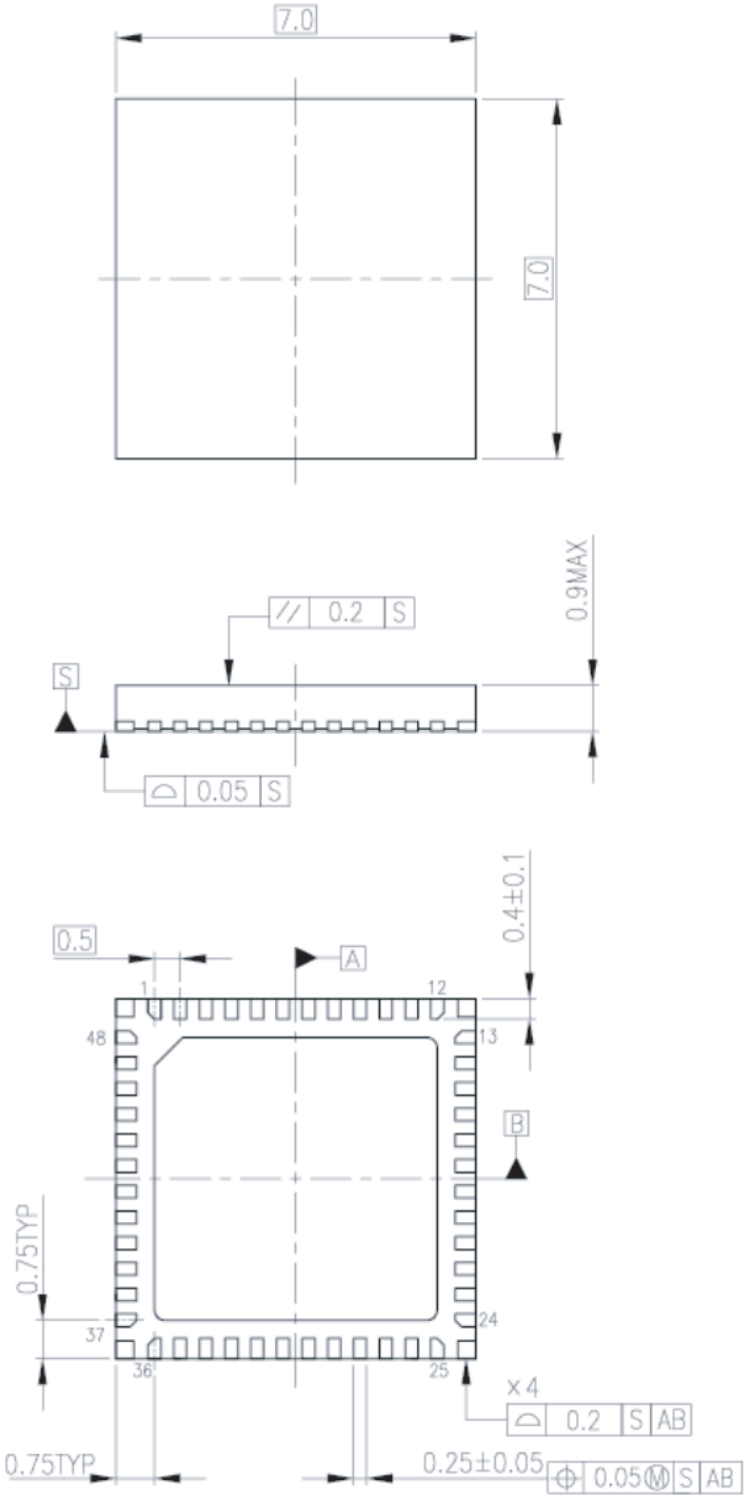


Figure 24-2 TMPM067FWQG

Type : P-VFBGA57-0505-0.50-001

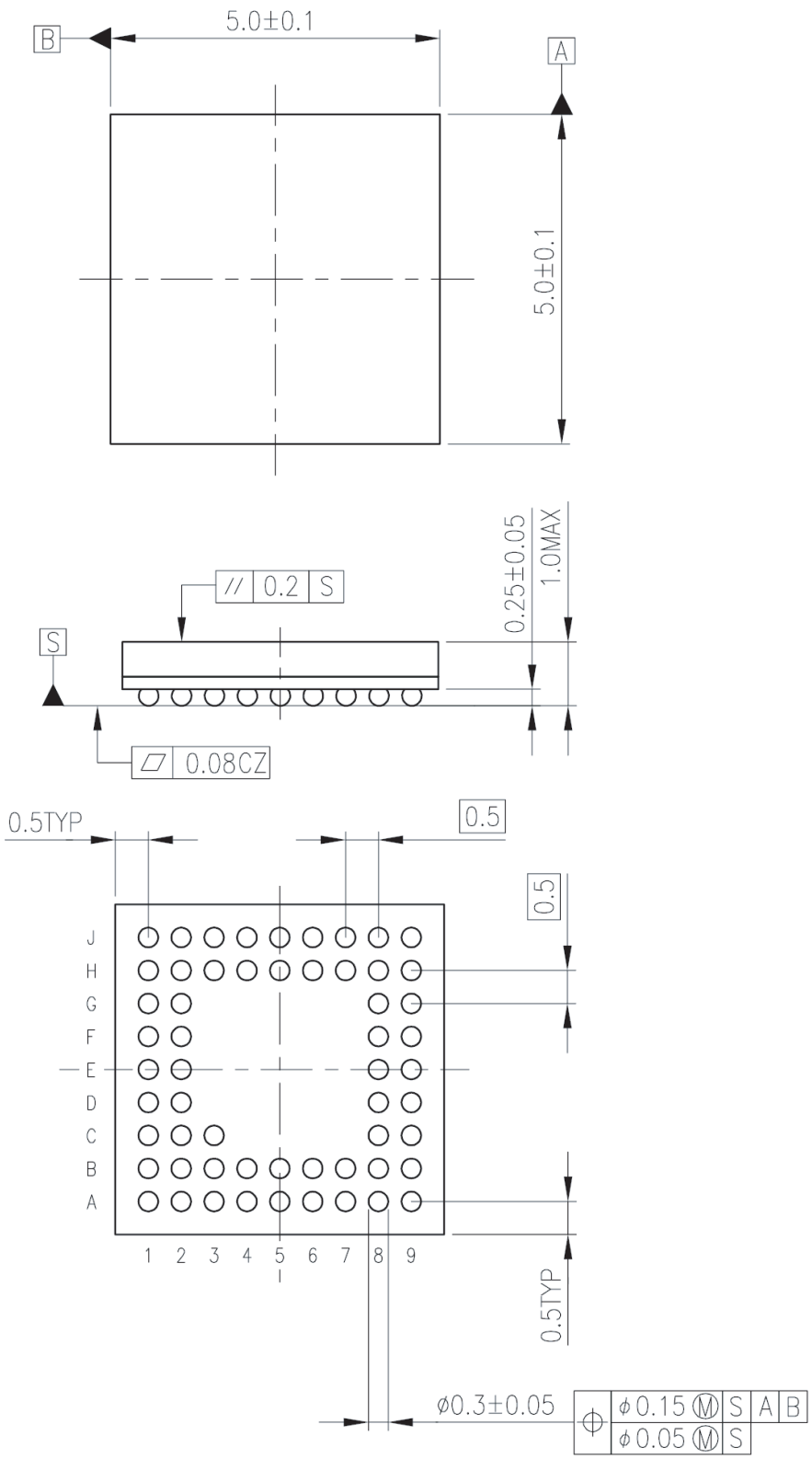


Figure 24-3 TMPM068FWXBG

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.**

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for ARM Microcontrollers - MCU ***category:***

Click to view products by Toshiba ***manufacturer:***

Other Similar products are found below :

R7FS3A77C2A01CLK#AC1 CP8363AT MB96F119RBPMC-GSE1 MB9BF122LPMC1-G-JNE2 MB9BF122LPMC-G-JNE2 MB9BF128SAPMC-GE2 MB9BF218TBGL-GE1 MB9BF529TBGL-GE1 26-21/R6C-AT1V2B/CT 5962-8506403MQA MB9AF342MAPMC-G-JNE2 MB96F001YBPMC1-GSE1 MB9BF121KPMC-G-JNE2 VA10800-D000003PCA CP8547AT CY9AF156NPMC-G-JNE2 MB9BF104NAPMC-G-JNE1 CY8C4724FNI-S402T ADUCM410BCBZ-RL7 ADUCM410BBCZ-RL7 GD32f303RGT6 NHS3152UK/A1Z MK26FN2M0CAC18R EFM32TG230F32-D-QFN64 EFM32TG232F32-D-QFP64 EFM32TG825F32-D-BGA48 MB9AFB44NBBGL-GE1 MB9BF304RBPMC-G-JNE2 MB9BF416RPMC-G-JNE2 MB9AF155MABGL-GE1 MB9BF306RBPMC-G-JNE2 MB9BF618TBGL-GE1 MK20DX64VFT5 MK50DX128CMC7 MK51DN256CMD10 MK51DX128CMC7 MK53DX256CMD10 MKL25Z32VFT4 MKL25Z64VFT4 LPC1754FBD80 STM32F030K6T6TR STM32L073VBT6 AT91M42800A-33AU AT91SAM7L64-CU ATSAM3N0AA-MU ATSAM3N0CA-CU ATSAM3SD8BA-MU ATSAM4LC2BA-UUR ATSAM4LC4BA-MU ATSAM4LS2AA-MU