TOSHIBA

32 Bit RISC Microcontroller

TX03 Series

TMPM341FDXBG/FYXBG

TOSHIBA CORPORATION
Semiconductor & Storage Products Company

Important Notices

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

1 Serial bus interface

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

1.1 Description

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

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

1.2 Condition

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

1.3 Workaround

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

1.4 How to Recover from These Phenomena

Perform recovery process by software.

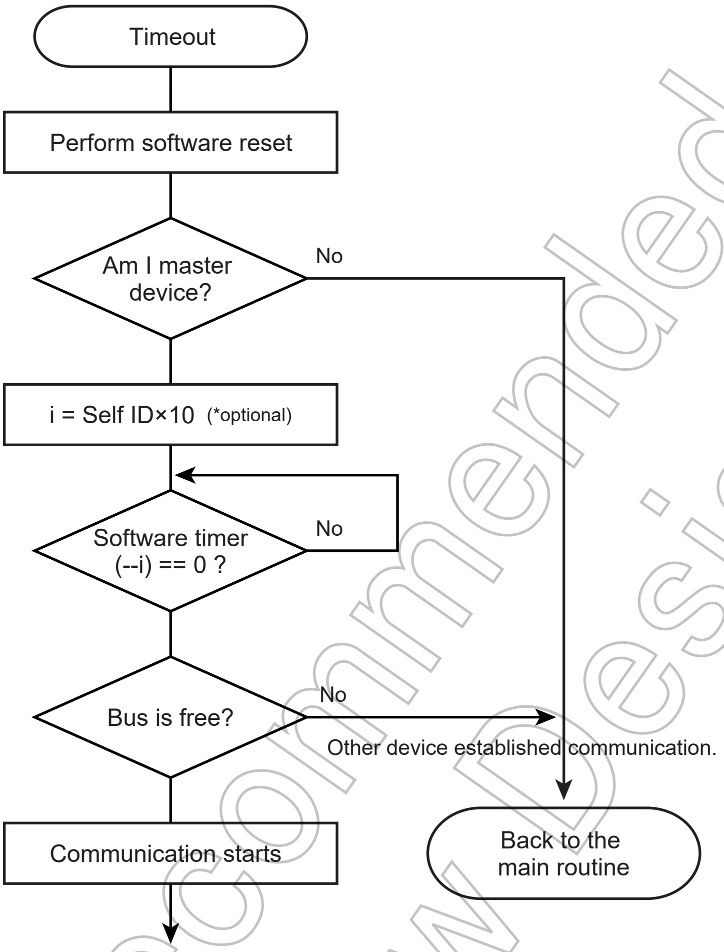
By using a timer, add timeout process to check whether communication is in a lock-up state.

An example of recovery process:

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

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

Example: Recovery process after a timeout is detected.



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

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

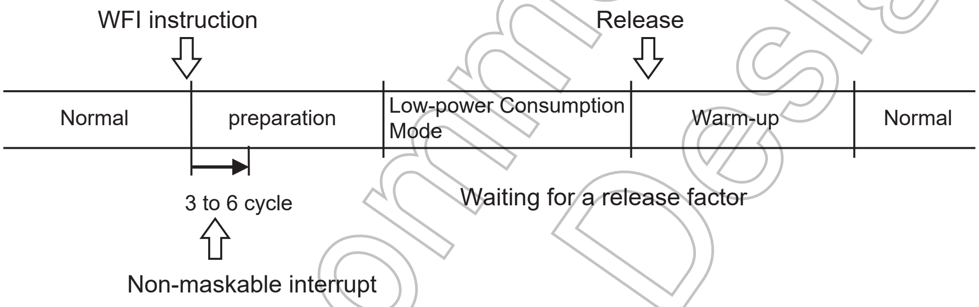
- STOP1
- STOP2

2.1 Description

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

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

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



2.2 Conditions

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

2.3 Workaround

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

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

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

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

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

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

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

a. SFR table of each IP as an example

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

Base Address = 0x0000_0000

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

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

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

b. SFR(register)

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

1.2.2 SAMCR(Control register)

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

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

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

R / W	READ WRITE
R	READ
W	WRITE

c. Data descriptopn

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

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

d. Register descriptoption

Registers are described as shown below.

- Register name <Bit Symbol>

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

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

- Register name [Bit]

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

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

Revision History

Date	Revision	Comment
2010/10/07	Tentative 1	First Release
2011/04/22	Tentative 2	Contents Revised
2011/10/15	1	First Release
2013/06/11	2	Contents Revised
2023/07/21	3	Contents Revised

Table of Contents

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

TMPM341FDXBG/FYXBG

2. Processor Core

3. Endian

4. Memory Map

5. External bus interface (EBIF)

6. DMA Controller(DMAC)

10. Clock/Mode control

11. Exceptions

13. 16-bit Timer/Event Counters(TMRB)

17. Synchronous Serial Port (SSP)

21. Flash Memory Operation

22. ROM protection

23. Debug Interface

27. Package Dimensions

TMPM341FDXBG/FYXBG

The TMPM341FDXBG/FYXBG is a 32-bit RISC microprocessor series with an ARM Cortex™-M3 microprocessor core.

Product Name	ROM (FLASH)	RAM	Package
TMPM341FDXBG	512 Kbyte	32 Kbyte	P-TFBGA113-0606-0.50A4
TMPM341FYXBG	256 Kbyte		

Features of the TMPM341FDXBG/FYXBG are as follows:

1.1 Features

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

   b. Both high performance and low power consumption have been achieved.

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

      [Low power consumption]
      - Optimized design using a low power consumption library
      - Standby function that stops the operation of the micro controller core

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

2. On chip program memory and data memory
   a. TMPM341FDXBG
      - On chip RAM : 32 Kbyte
      - On chip Flash ROM 512 Kbyte

   b. TMPM341FYXBG
      - On chip SRAM : 32 Kbyte
      - On chip Flash ROM 256 Kbyte

3. External bus interface (EBIF)
   - Up to 16Mbytes access area (Program / Data)
   - External data bus (Separate / Multiplex): 8/16bit bus width

- Chip select / Wait controller: 2 channels

4. DMA controller (DMAC): 4 channels

   Transfer can support on chip Memory / Peripheral I/O / External memory

5. 16-bit timer / event counter (TMRB): 10 channels
   - 16-bit interval timer mode
   - 16-bit event counter mode
   - 16-bit PPG output (can start 4-channels synchronously)
   - Input capture function

6. 2-phase pulse input counter (PHCNT): 4 channels

7. High resolution 16-bit timer / PPG output (TMRD): 2 channels

   Minimum resolution: 6.25ns

8. Watchdog timer (WDT): 1 channel

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

9. Oscillation Frequency Detector (OFD): 1 channel

   Monitoring for external high frequency oscillator

10. Serial channel (SIO/UART): 5 channels

    Either UART mode or I/O interface mode can be selected (4byte FIFO equipped)

11. Serial bus interface (I2C/SIO): 2 channels

    Either I2C bus mode or clock-synchronous 8-bit SIO mode can be selected.

12. Synchronous serial port (SSP): 1 channel
    - Communication protocol that includes SPI: 3 types (SPI/SSI/Microwire)
    - Baud rate: Master mode: 20Mbps (max.), Slave mode: 4.5Mbps (max.)

13. 12-bit AD converter (ADC): 15 channels
    - Start up with 16-bit timer / Start up with an external trigger input
    - Fixed channel / Channel scan mode
    - Single / repeat mode
    - AD monitoring: 2channels
    - Conversion time: 1µsec. (@fsys = 40MHz)

14. 10-bit DA converter (DAC): 2 channels
    - Built-in cutoff function of AVREFH (reduce the power consumption)
    - Settling time of input variation / output variation: 100µs

15. Interrupt source
    - Internal 73 factors: The order of precedence can be set over 7 levels (except the watchdog timer interrupt).

- External 12 factors: The order of precedence can be set over 7 levels.

16. Non-maskable interrupt (NMI)

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

17. Input/ output ports (PORT): 86 pins

    I/O pin: 85pins (11 5V tolerant input included)

    Output pin: 1 pin

18. Low power consumption mode

    IDLE, STOP1, STOP2

19. Clock generator (CG)
    - On chip PLL (8 times or 16 times selectable)
    - Clock gear function: The high-speed clock can be divided into 1/1, 1/2, 1/4, 1/8 or 1/16.

20. Endian

    Little-Endian or Big-Endian can be selectable

21. Debug interface

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

22. JTAG interface

    Boundary scan

23. Maximum operating frequency: 54 MHz

24. Operating voltage range
    - DVDD3A = 2.7 V to 3.6 V
    - DVDD3B = 1.65 V to 3.6 V
    - AVDD3 = 2.7 V to 3.6 V

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

26. Package

    P-TFBGA113-0606-0.50A4 (6mm × 6mm, 0.5mm pitch)

1.2 Block Diagram

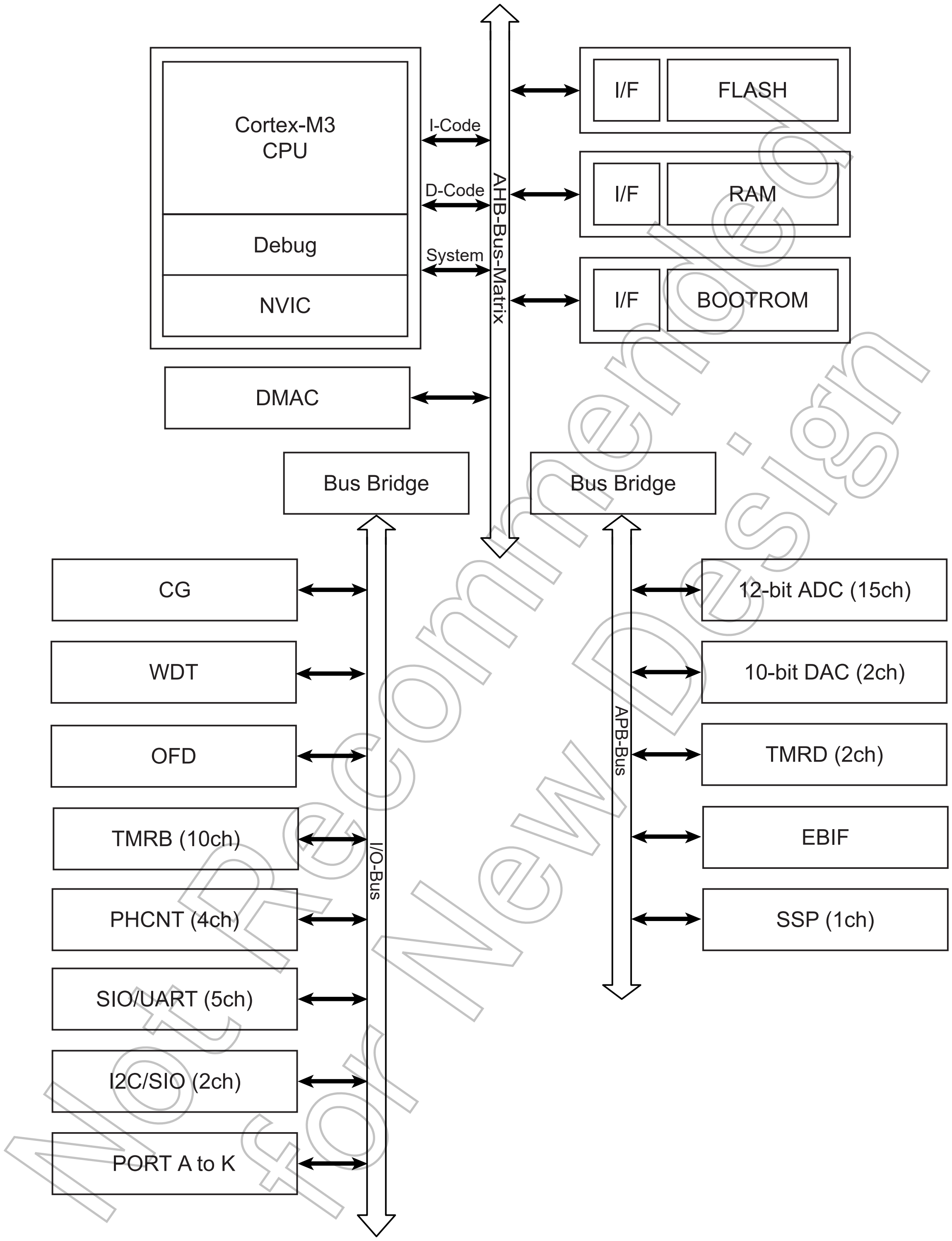


Figure 1-1 TMPM341FDXBG/FYXBGBlock Diagram

1.3 Pin Layout (Top view)

Figure 1-2 shows the pin layout of TMPM341FDXBG/FYXBG.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
E1	E2	E3	E4	E5	—	—	E8	E9	E10	E11
F1	F2	F3	F4	—	—	—	F8	F9	F10	F11
G1	G2	G3	G4	—	—	—	G8	G9	G10	G11
H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11
J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11
K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11
L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11

Figure 1-2 Pin Layout (BGA113)

1.4 Pin names and Functions

Table 1-1 and Table 1-2 sort the input and output pins of the TMPM341FDXBG/FYXBG by pin or port.

1.4.1 Sorted by Pin

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

Type	Pin No.	Pin Name	Input/ Out-put	Function
Control	A1	ENDIAN	I	Little Endian operation: Fixed DVSSA Big Endian operation: Fixed DVDD3A
Function	A2	PK4 AIN12	I/O I	I/O port Analog input
Function	A3	PJ7 AIN07 INT9 TB0IN1	I/O I I I	I/O port Analog input External interrupt pin Inputting 16-bit timer / event counter capture trigger
Function	A4	PJ3 AIN03 PHC1IN1	I/O I I	I/O port Analog input Inputting the capture trigger of two-phase pulse counter
PS	A5	DVDD3A	−	Power supply pin
Function	A6	$\overline{\text{NMI}}$	I	Non-maskable interrupt (note) With a noise filter (about 30ns (typical value))
Clock	A7	X1/EHCLKIN	I	Connected to a high-speed oscillator/ External clock input pin
PS	A8	DVSSC	−	GND pin
Clock	A9	X2	O	Connected to a high-speed oscillator.
Function/ Debug	A10	PI5 TCK/SWCLK	I/O I	I/O port Debug pin
Function/ Debug	A11	PH5 TRACEDA-TA3	I/O O	I/O port Debug pin
Function	B1	PK6 AIN14	I/O I	I/O port Analog input
Function	B2	PK5 AIN13	I/O I	I/O port Analog input
Function	B3	PK0 AIN08 TB1IN0	I/O I I	I/O port Analog input Inputting 16-bit timer / event counter capture trigger
Function	B4	PJ4 AIN04 PHC2IN0	I/O I I	I/O port Analog input Inputting the capture trigger of two-phase pulse counter
Function	B5	PJ0 AIN00 PHC0IN0	I/O I I	I/O port Analog input Inputting the capture trigger of two-phase pulse counter
Function	B6	$\overline{\text{RESET}}$	I	Reset input pin (note) With a pull-up and a noise filter (about 30ns (typical value))

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

Type	Pin No.	Pin Name	Input/ Out-put	Function
Function	B7	PH3 PHC3IN0 TB4OUT	I/O I O	I/O port Inputting the capture trigger of two-phase pulse counter 16-bit timer / event counter output
PS	B8	RVSS	−	GND pin for internal regulator
PS	B9	RVDD3	−	Power supply pin for internal regulator
Function/ Debug	B10	PI4 TDI	I/O I	I/O port Debug pin
Function/ Debug	B11	PH6 TRACEDA-TA2	I/O O	I/O port Debug pin
PS	C1	AVDD3	I	Power supply pin for AD and DA converters. (note) AVDD3 must be connected to power supply even if A/D and D/A converters are not used.
PS	C2	AVREFH	I	Power supply pin for AD and DA converters. (note) AVREFH must be connected to power supply even if A/D and D/A converters are not used.
Function	C3	PK1 AIN09 INTA TB1IN1	I/O I I I	I/O port Analog input External interrupt pin Inputting 16-bit timer / event counter capture trigger
Function	C4	PJ5 AIN05 PHC2IN1	I/O I I	I/O port Analog input Inputting the capture trigger of two-phase pulse counter
Function	C5	PJ1 AIN01 PHC0IN1	I/O I I	I/O port Analog input Inputting the capture trigger of two-phase pulse counter
Control	C6	MODE	I	Mode pin (note) MODE pin must be connected to GND.
Function	C7	PH4 PHC3IN1 TB5OUT	I/O I O	I/O port Inputting the capture trigger of two-phase pulse counter 16-bit timer / event counter output
Function	C8	PH0 TXD4	I/O O	I/O port with 5V tolerant input (note) Serial channel sending serial data
Function/ Debug	C9	PI7 TDO/SWV	I/O O	I/O port Debug pin
Function/ Debug	C10	PI6 TMS/SWDIO	I/O I/O	I/O port Debug pin
PS	C11	DVDD3A	−	Power supply pin
PS	D1	AVSS	I	GND pin for AD or DA converters. (note) AVSS must be connected to GND even if A/D and D/A converters are not used.
PS	D2	AVREFL	I	Power supply pin for AD and DA converters. (note) AVREFL must be connected to GND even if A/D and D/A converters are not used.

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

Type	Pin No.	Pin Name	Input/ Out- put	Function
Function	D3	PK2 AIN10 TB6IN0	I/O I I	I/O port Analog input Inputting 16-bit timer / event counter capture trigger
Function	D4	PJ6 AIN06 TB0IN0	I/O I I	I/O port Analog input Inputting 16-bit timer / event counter capture trigger
Function	D5	PJ2 AIN02 PHC1IN0	I/O I I	I/O port Analog input Inputting the capture trigger of two-phase pulse counter
Function	D6	PG7 INT1 TB9IN1	I/O I I	I/O port with 5V tolerant input (note) External interrupt pin Inputting 16-bit timer / event counter capture trigger
Function	D7	PG6 SCLK3 TB9IN0 $\overline{\text{CTS3}}$	I/O I/O I I	I/O port with 5V tolerant input (note) Serial channel clock pin Inputting 16-bit timer / event counter capture trigger Handshake input pin
Function	D8	PH1 RXD4	I/O I	I/O port with 5V tolerant input (note) Serial channel receiving serial data
Function/ Debug	D9	PI2 TRACECLK	I/O O	I/O port Debug pin
Function/ Debug	D10	PI3 $\overline{\text{TRST}}$	I/O I	I/O port Debug pin
PS	D11	DVSSA	−	GND pin
PS	E1	DVSSB	−	GND pin
PS	E2	DVDD3B	−	Power supply pin
Function	E3	PK3 AIN11 INTB TB6IN1	I/O I I I	I/O port Analog input External interrupt pin Inputting 16-bit timer / event counter capture trigger
Function	E4	DA0	O	Analog output
Function	E5	DA1	O	Analog output
Function	E8	PH2 SCLK4 $\overline{\text{CTS4}}$	I/O I/O I	I/O port with 5V tolerant input (note) Serial channel clock pin Handshake input pin
Function	E9	PG5 RXD3 TB8IN1	I/O I I	I/O port with 5V tolerant input (note) Serial channel receiving serial data Inputting 16-bit timer / event counter capture trigger
Function/ Debug	E10	PI1 TRACEDA-TA0	I/O O	I/O port Debug pin
PS	E11	DVSSB	−	GND pin

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

Type	Pin No.	Pin Name	Input/ Out- put	Function
Function	F1	PA0 D0/AD0	I/O I/O	I/O port Data bus/Address data bus
Function/ Control	F2	PF0 $\overline{\text{BOOT}}$ TB6OUT	O I O	I/O port Setting a single boot mode: TMPM341FDXBG/FYXBG goes into single boot mode by sampling "Low" at the rising edge of a $\overline{\text{RESET}}$ pin. 16-bit timer / event counter output
Function	F3	PF1 $\overline{\text{RD}}$	I/O O	I/O port Read strobe signal
Control	F4	INTLV	I	Interleave control pin (Refer to Clock/Mode control chapter) If fc > 40MHz, connect with pull-up. If fc ≤ 40MHz, connect with pull-down.
Function	F8	PG4 TXD3 TB8IN0	I/O O I	I/O port with 5V tolerant input (note) Serial channel sending serial data Inputting 16-bit timer / event counter capture trigger
Function	F9	PG3 INT0	I/O I	I/O port with 5V tolerant input (note) External interrupt pin
Function/ Debug	F10	PI0 TRACEDATA1	I/O O	I/O port Debug pin
PS	F11	DVDD3B	−	Power supply pin
Function	G1	PA1 D1/AD1	I/O I/O	I/O port Data bus/Address data bus
Function	G2	PA2 D2/AD2	I/O I/O	I/O port Data bus/Address data bus
Function	G3	PF3 $\overline{\text{BELL}}$	I/O O	I/O port Byte enable signal as an external 8-bit memory access
Function	G4	PF2 $\overline{\text{WR}}$	I/O O	I/O port Write strobe signal
Function	G8	PG2 SCK0 INT8	I/O I/O I	I/O port with 5V tolerant input (note) Clock in SIO mode External interrupt pin
Function	G9	PG1 SCL0/SI0 TB7IN1	I/O I/O I	I/O port with 5V tolerant input (note) Clock in I2C mode / Data in SIO mode Inputting 16-bit timer / event counter capture trigger
Function	G10	PD7 A15 SPFSS SCOUT	I/O O I/O O	I/O port Address bus SSP FSS input/ output System clock output
Function	G11	PD6 A14 SPCLK	I/O O I/O	I/O port Address bus SSP clock input/ output
Function	H1	PA3 D3/AD3	I/O I/O	I/O port Data bus/Address data bus

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

Type	Pin No.	Pin Name	Input/ Out- put	Function
Function	H2	PA4 D4/AD4	I/O I/O	I/O port Data bus/Address data bus
Function	H3	PF4 $\overline{\text{BELH}}$ INT6 TB5IN0	I/O O I I	I/O port Byte enable signal as an external 16-bit memory access External interrupt pin Inputting 16-bit timer / event counter capture trigger
Function	H4	PF5 $\overline{\text{CS1}}$ INT7 TB5IN1	I/O O I I	I/O port Chip select output External interrupt pin Inputting 16-bit timer / event counter capture trigger
Function	H5	PC1 A1 RXD1 TB2IN1	I/O O I I	I/O port Address bus Serial channel receiving serial data Inputting 16-bit timer / event counter capture trigger
Function	H6	PC3 A3 INT2 TB1OUT	I/O O I O	I/O port Address bus External interrupt pin 16-bit timer / event counter output
Function	H7	PC6 A6 SCLK2 TB4IN0 $\overline{\text{CTS2}}$	I/O O I/O I I	I/O port Address bus Serial channel clock pin Inputting 16-bit timer / event counter capture trigger Handshake input pin
Function	H8	PC7 A7 INT3 TB4IN1	I/O O I I	I/O port Address bus External interrupt pin Inputting 16-bit timer / event counter capture trigger
Function	H9	PG0 SDA0/SO0 TB7IN0	I/O I/O I	I/O port with 5V tolerant input (note) Data in I2C mode / Data in SIO mode Inputting 16-bit timer / event counter capture trigger
Function	H10	PD5 A13 SPDI	I/O O I	I/O port Address bus SSP DI input
Function	H11	PD4 A12 SPDO	I/O O O	I/O port Address bus SSP DO output
Function	J1	PA5 D5/AD5	I/O I/O	I/O port Data bus/Address data bus
Function	J2	PA6 D6/AD6	I/O I/O	I/O port Data bus/Address data bus
Function	J3	PF6 $\overline{\text{CS0}}$	I/O O	I/O port Chip select output

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

Type	Pin No.	Pin Name	Input/ Out- put	Function
Function	J4	PF7 ALE	I/O O	I/O port Address latch enable (output disable for noise-reduction can be selectable)
Function	J5	PC0 A0 TXD1 TB2IN0	I/O O O I	I/O port Address bus Serial channel sending serial data Inputting 16-bit timer / event counter capture trigger
Function	J6	PC2 A2 SCLK1 TB0OUT $\overline{\text{CTS1}}$	I/O O I/O O I	I/O port Address bus Serial channel clock pin 16-bit timer / event counter output Handshake input pin
Function	J7	PC5 A5 RXD2 TB3IN1	I/O O I I	I/O port Address bus Serial channel receiving serial data Inputting 16-bit timer / event counter capture trigger
Function	J8	PE0 TXD0 A16	I/O O O	I/O port Serial channel sending serial data Address bus
Function	J9	PE2 SCLK0 A18 TB2OUT $\overline{\text{CTS0}}$	I/O I/O O O I	I/O port Serial channel clock pin Address bus 16-bit timer / event counter output Handshake input pin
Function	J10	PD3 A11 INT4 $\overline{\text{ADTRG}}$	I/O O I I	I/O port Address bus External interrupt pin ADC trigger input
Function	J11	PD2 A10 SCK1 TB9OUT	I/O O I/O O	I/O port Address bus Clock in SIO mode 16-bit timer / event counter output
Function	K1	PA7 D7/AD7	I/O I/O	I/O port Data bus/Address data bus
Function	K2	PB1 D9/AD9 A1	I/O I/O O	I/O port Data bus/Address data bus Address bus
Function	K3	PB3 D11/AD11 A3	I/O I/O O	I/O port Data bus/Address data bus Address bus
Function	K4	PB5 D13/AD13 A5	I/O I/O O	I/O port Data bus/Address data bus Address bus

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

Type	Pin No.	Pin Name	Input/ Out- put	Function
Function	K5	PB7 D15/AD15 A7	I/O I/O O	I/O port Data bus/Address data bus Address bus
PS	K6	DVDD3B	−	Power supply pin
Function	K7	PC4 A4 TXD2 TB3IN0	I/O O O I	I/O port Address bus Serial channel sending serial data Inputting 16-bit timer / event counter capture trigger
Function	K8	PE1 RXD0 A17	I/O I O	I/O port Serial channel receiving serial data Address bus
Function	K9	PE3 INT5 A19 TB3OUT	I/O I O O	I/O port External interrupt pin Address bus 16-bit timer / event counter output
Function	K10	PD1 A9 SCL1/SI1 TB8OUT	I/O O I/O O	I/O port Address bus Clock in I2C mode / Data in SIO mode 16-bit timer / event counter output
Function	K11	PD0 A8 SDA1/SO1 TB7OUT	I/O O I/O O	I/O port Address bus Data in I2C mode / Data in SIO mode 16-bit timer / event counter output
Control	L1	BSC	I	JTAG Boundary scan control pin (note) BSC must be connected to GND even if boundary scan is not used.
Function	L2	PB0 D8/AD8 A0	I/O I/O O	I/O port Data bus/Address data bus Address bus
Function	L3	PB2 D10/AD10 A2	I/O I/O O	I/O port Data bus/Address data bus Address bus
Function	L4	PB4 D12/AD12 A4	I/O I/O O	I/O port Data bus/Address data bus Address bus
Function	L5	PB6 D14/AD14 A6	I/O I/O O	I/O port Data bus/Address data bus Address bus
PS	L6	DVSSB	−	GND pin
Function	L7	PE4 A20 TD0OUT0	I/O O O	I/O port Address bus Timer D output

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

Type	Pin No.	Pin Name	Input/ Out- put	Function
Function	L8	PE5 A21 TD0OUT1	I/O O O	I/O port Address bus Timer D output
Function	L9	PE6 A22 TD1OUT0	I/O O O	I/O port Address bus Timer D output
Function	L10	PE7 A23 TD1OUT1	I/O O O	I/O port Address bus Timer D output
Control	L11	FTEST3	−	TEST pin (note) TEST pin must be left OPEN.

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

1.4.2 Sorted by Port

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

PORT	Type	Pin No.	Pin Name	Input/ Out- put	Function
PORT A	Function	F1	PA0 D0/AD0	I/O I/O	I/O port Data bus/Address data bus
PORT A	Function	G1	PA1 D1/AD1	I/O I/O	I/O port Data bus/Address data bus
PORT A	Function	G2	PA2 D2/AD2	I/O I/O	I/O port Data bus/Address data bus
PORT A	Function	H1	PA3 D3/AD3	I/O I/O	I/O port Data bus/Address data bus
PORT A	Function	H2	PA4 D4/AD4	I/O I/O	I/O port Data bus/Address data bus
PORT A	Function	J1	PA5 D5/AD5	I/O I/O	I/O port Data bus/Address data bus
PORT A	Function	J2	PA6 D6/AD6	I/O I/O	I/O port Data bus/Address data bus
PORT A	Function	K1	PA7 D7/AD7	I/O I/O	I/O port Data bus/Address data bus
PORT B	Function	L2	PB0 D8/AD8 A0	I/O I/O O	I/O port Data bus/Address data bus Address bus
PORT B	Function	K2	PB1 D9/AD9 A1	I/O I/O O	I/O port Data bus/Address data bus Address bus
PORT B	Function	L3	PB2 D10/AD10 A2	I/O I/O O	I/O port Data bus/Address data bus Address bus
PORT B	Function	K3	PB3 D11/AD11 A3	I/O I/O O	I/O port Data bus/Address data bus Address bus
PORT B	Function	L4	PB4 D12/AD12 A4	I/O I/O O	I/O port Data bus/Address data bus Address bus
PORT B	Function	K4	PB5 D13/AD13 A5	I/O I/O O	I/O port Data bus/Address data bus Address bus
PORT B	Function	L5	PB6 D14/AD14 A6	I/O I/O O	I/O port Data bus/Address data bus Address bus
PORT B	Function	K5	PB7 D15/AD15 A7	I/O I/O O	I/O port Data bus/Address data bus Address bus

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

PORT	Type	Pin No.	Pin Name	Input/ Out- put	Function
PORT C	Function	J5	PC0 A0 TXD1 TB2IN0	I/O O O I	I/O port Address bus Serial channel sending serial data Inputting 16-bit timer / event counter capture trigger
PORT C	Function	H5	PC1 A1 RXD1 TB2IN1	I/O O I I	I/O port Address bus Serial channel receiving serial data Inputting 16-bit timer / event counter capture trigger
PORT C	Function	J6	PC2 A2 SCLK1 TB0OUT $\overline{\text{CTS1}}$	I/O O I/O O I	I/O port Address bus Serial channel clock pin 16-bit timer / event counter output Handshake input pin
PORT C	Function	H6	PC3 A3 INT2 TB1OUT	I/O O I O	I/O port Address bus External interrupt pin 16-bit timer / event counter output
PORT C	Function	K7	PC4 A4 TXD2 TB3IN0	I/O O O I	I/O port Address bus Serial channel sending serial data Inputting 16-bit timer / event counter capture trigger
PORT C	Function	J7	PC5 A5 RXD2 TB3IN1	I/O O I I	I/O port Address bus Serial channel receiving serial data Inputting 16-bit timer / event counter capture trigger
PORT C	Function	H7	PC6 A6 SCLK2 TB4IN0 $\overline{\text{CTS2}}$	I/O O I/O I I	I/O port Address bus Serial channel clock pin Inputting 16-bit timer / event counter capture trigger Handshake input pin
PORT C	Function	H8	PC7 A7 INT3 TB4IN1	I/O O I I	I/O port Address bus External interrupt pin Inputting 16-bit timer / event counter capture trigger
PORT D	Function	K11	PD0 A8 SO1/SDA1 TB7OUT	I/O O I/O O	I/O port Address bus Data in SIO mode / Data in I2C mode 16-bit timer / event counter output
PORT D	Function	K10	PD1 A9 SCL1/SI1 TB8OUT	I/O O I/O O	I/O port Address bus Clock in I2C mode / Data in SIO mode 16-bit timer / event counter output

Table 1-2 Pin Names and Functions Sorted by Port (3/8)

PORT	Type	Pin No.	Pin Name	Input/ Out- put	Function
PORT D	Function	J11	PD2 A10 SCK1 TB9OUT	I/O O I/O O	I/O port Address bus Clock in SIO mode 16-bit timer / event counter output
PORT D	Function	J10	PD3 A11 INT4 $\overline{\text{ADTRG}}$	I/O O I I	I/O port Address bus External interrupt pin ADC trigger input
PORT D	Function	H11	PD4 A12 SPDO	I/O O O	I/O port Address bus SSP DO output
PORT D	Function	H10	PD5 A13 SPDI	I/O O I	I/O port Address bus SSP DI input
PORT D	Function	G11	PD6 A14 SPCLK	I/O O I/O	I/O port Address bus SSP clock input/ output
PORT D	Function	G10	PD7 A15 SPFSS SCOUT	I/O O I/O O	I/O port Address bus SSP FSS input/ output System clock output
PORT E	Function	J8	PE0 TXD0 A16	I/O O O	I/O port Serial channel sending serial data Address bus
PORT E	Function	K8	PE1 RXD0 A17	I/O I O	I/O port Serial channel receiving serial data Address bus
PORT E	Function	J9	PE2 SCLK0 A18 TB2OUT $\overline{\text{CTS0}}$	I/O I/O O O I	I/O port Serial channel clock pin Address bus 16-bit timer / event counter output Handshake input pin
PORT E	Function	K9	PE3 INT5 A19 TB3OUT	I/O I O O	I/O port External interrupt pin Address bus 16-bit timer / event counter output
PORT E	Function	L7	PE4 A20 TD0OUT0	I/O O O	I/O port Address bus Timer D output
PORT E	Function	L8	PE5 A21 TD0OUT1	I/O O O	I/O port Address bus Timer D output

Table 1-2 Pin Names and Functions Sorted by Port (4/8)

PORT	Type	Pin No.	Pin Name	Input/ Out- put	Function
PORT E	Function	L9	PE6 A22 TD1OUT0	I/O O O	I/O port Address bus Timer D output
PORT E	Function	L10	PE7 A23 TD1OUT1	I/O O O	I/O port Address bus Timer D output
PORT F	Function/ Control	F2	PF0 $\overline{\text{BOOT}}$ TB6OUT	O I O	I/O port Setting a single boot mode: TMPM341FDXBG/FYXBG goes into single boot mode by sampling "Low" at the rise of a $\overline{\text{RESET}}$ pin. 16-bit timer / event counter output
PORT F	Function	F3	PF1 $\overline{\text{RD}}$	I/O O	I/O port Read strobe signal
PORT F	Function	G4	PF2 $\overline{\text{WR}}$	I/O O	I/O port Write strobe signal
PORT F	Function	G3	PF3 $\overline{\text{BELL}}$	I/O O	I/O port Byte enable signal as an external 8-bit memory access
PORT F	Function	H3	PF4 $\overline{\text{BELH}}$ INT6 TB5IN0	I/O O I I	I/O port Byte enable signal as an external 16-bit memory access External interrupt pin Inputting 16-bit timer / event counter capture trigger
PORT F	Function	H4	PF5 $\overline{\text{CS1}}$ INT7 TB5IN1	I/O O I I	I/O port Chip select output External interrupt pin Inputting 16-bit timer / event counter capture trigger
PORT F	Function	J3	PF6 $\overline{\text{CS0}}$	I/O O	I/O port Chip select output
PORT F	Function	J4	PF7 ALE	I/O O	I/O port Address latch enable (output disable for noise-reduction can be selectable)
PORT G	Function	H9	PG0 SDA0/SO0 TB7IN0	I/O I/O I	I/O port with 5V tolerant input (note) Data in I2C mode / Data in SIO mode Inputting 16-bit timer / event counter capture trigger
PORT G	Function	G9	PG1 SCL0/SI0 TB7IN1	I/O I/O I	I/O port with 5V tolerant input (note) Clock in I2C mode / Data in SIO mode Inputting 16-bit timer / event counter capture trigger
PORT G	Function	G8	PG2 SCK0 INT8	I/O I/O I	I/O port with 5V tolerant input (note) Clock in SIO mode External interrupt pin
PORT G	Function	F9	PG3 INT0	I/O I	I/O port with 5V tolerant input (note) External interrupt pin
PORT G	Function	F8	PG4 TXD3 TB8IN0	I/O O I	I/O port with 5V tolerant input (note) Serial channel sending serial data Inputting 16-bit timer / event counter capture trigger

Table 1-2 Pin Names and Functions Sorted by Port (5/8)

PORT	Type	Pin No.	Pin Name	Input/ Out- put	Function
PORT G	Function	E9	PG5 RXD3 TB8IN1	I/O I I	I/O port with 5V tolerant input (note) Serial channel receiving serial data Inputting 16-bit timer / event counter capture trigger
PORT G	Function	D7	PG6 SCLK3 TB9IN0 $\overline{\text{CTS3}}$	I/O I/O I I	I/O port with 5V tolerant input (note) Serial channel clock pin Inputting 16-bit timer / event counter capture trigger Handshake input pin
PORT G	Function	D6	PG7 INT1 TB9IN1	I/O I I	I/O port with 5V tolerant input (note) External interrupt pin Inputting 16-bit timer / event counter capture trigger
PORT H	Function	C8	PH0 TXD4	I/O O	I/O port with 5V tolerant input (note) Serial channel sending serial data
PORT H	Function	D8	PH1 RXD4	I/O I	I/O port with 5V tolerant input (note) Serial channel receiving serial data
PORT H	Function	E8	PH2 SCLK4 $\overline{\text{CTS4}}$	I/O I/O I	I/O port with 5V tolerant input (note) Serial channel clock pin Handshake input pin
PORT H	Function	B7	PH3 PHC3IN0 TB4OUT	I/O I O	I/O port Inputting the capture trigger of two-phase pulse counter 16-bit timer / event counter output
PORT H	Function	C7	PH4 PHC3IN1 TB5OUT	I/O I O	I/O port Inputting the capture trigger of two-phase pulse counter 16-bit timer / event counter output
PORT H	Function/ Debug	A11	PH5 TRACEDATA3	I/O O	I/O port Debug pin
PORT H	Function/ Debug	B11	PH6 TRACEDATA2	I/O O	I/O port Debug pin
PORT I	Function/ Debug	F10	PI0 TRACEDATA1	I/O O	I/O port Debug pin
PORT I	Function/ Debug	E10	PI1 TRACEDATA0	I/O O	I/O port Debug pin
PORT I	Function/ Debug	D9	PI2 TRACECLK	I/O O	I/O port Debug pin
PORT I	Function/ Debug	D10	PI3 $\overline{\text{TRST}}$	I/O I	I/O port Debug pin
PORT I	Function/ Debug	B10	PI4 TDI	I/O I	I/O port Debug pin
PORT I	Function/ Debug	A10	PI5 TCK/SWCLK	I/O I	I/O port Debug pin
PORT I	Function/ Debug	C10	PI6 TMS/SWDIO	I/O I/O	I/O port Debug pin
PORT I	Function/ Debug	C9	PI7 TDO/SWV	I/O O	I/O port Debug pin

Table 1-2 Pin Names and Functions Sorted by Port (6/8)

PORT	Type	Pin No.	Pin Name	Input/ Out-put	Function
PORT J	Function	B5	PJ0 AIN00 PHC0IN0	I/O I I	I/O port Analog input Inputting the capture trigger of two-phase pulse counter
PORT J	Function	C5	PJ1 AIN01 PHC0IN1	I/O I I	I/O port Analog input Inputting the capture trigger of two-phase pulse counter
PORT J	Function	D5	PJ2 AIN02 PHC1IN0	I/O I I	I/O port Analog input Inputting the capture trigger of two-phase pulse counter
PORT J	Function	A4	PJ3 AIN03 PHC1IN1	I/O I I	I/O port Analog input Inputting the capture trigger of two-phase pulse counter
PORT J	Function	B4	PJ4 AIN04 PHC2IN0	I/O I I	I/O port Analog input Inputting the capture trigger of two-phase pulse counter
PORT J	Function	C4	PJ5 AIN05 PHC2IN1	I/O I I	I/O port Analog input Inputting the capture trigger of two-phase pulse counter
PORT J	Function	D4	PJ6 AIN06 TB0IN0	I/O I I	I/O port Analog input Inputting 16-bit timer / event counter capture trigger
PORT J	Function	A3	PJ7 AIN07 INT9 TB0IN1	I/O I I I	I/O port Analog input External interrupt pin Inputting 16-bit timer / event counter capture trigger
PORT K	Function	B3	PK0 AIN08 TB1IN0	I/O I I	I/O port Analog input Inputting 16-bit timer / event counter capture trigger
PORT K	Function	C3	PK1 AIN09 INTA TB1IN1	I/O I I I	I/O port Analog input External interrupt pin Inputting 16-bit timer / event counter capture trigger
PORT K	Function	D3	PK2 AIN10 TB6IN0	I/O I I	I/O port Analog input Inputting 16-bit timer / event counter capture trigger
PORT K	Function	E3	PK3 AIN11 INTB TB6IN1	I/O I I I	I/O port Analog input External interrupt pin Inputting 16-bit timer / event counter capture trigger
PORT K	Function	A2	PK4 AIN12	I/O I	I/O port Analog input
PORT K	Function	B2	PK5 AIN13	I/O I	I/O port Analog input

Table 1-2 Pin Names and Functions Sorted by Port (7/8)

PORT	Type	Pin No.	Pin Name	Input/ Out- put	Function
PORT K	Function	B1	PK6 AIN14	I/O I	I/O port Analog input
-	Control	A1	ENDIAN	I	Little Endian operation: Fixed DVSSA Big Endian operation: Fixed DVDD3A
-	Function	E4	DA0	O	Analog output
-	Function	E5	DA1	O	Analog output
-	Function	B6	$\overline{\text{RESET}}$	I	Reset input pin (note) With a pull-up and a noise filter (about 30ns (typical value))
-	Function	A6	$\overline{\text{NMI}}$	I	Non-maskable interrupt (note) With a noise filter (about 30ns (typical value))
-	Control	C6	MODE	I	Mode pin (note) MODE pin must be connected to GND.
-	Control	L11	FTEST3	−	TEST pin (note) TEST pin must be left OPEN.
-	Control	F4	INTLV	I	Interleave control pin (Refer to Clock/Mode control chapter) If fc > 40MHz, connect with pull-up. If fc ≤ 40MHz, connect with pull-down.
-	Control	L1	BSC	I	JTAG Boundary scan control pin (note) BSC must be connected to GND even if boundary scan is not used.
-	Clock	A7	X1/EHCLKIN	I	Connected to a high-speed oscillator/ External clock input pin
-	Clock	A9	X2	O	Connected to a high-speed oscillator.
-	PS	A5	DVDD3A	−	Power supply pin
-	PS	C11	DVDD3A	−	Power supply pin
-	PS	E2	DVDD3B	−	Power supply pin
-	PS	F11	DVDD3B	−	Power supply pin
-	PS	K6	DVDD3B	−	Power supply pin
-	PS	D11	DVSSA	−	GND pin
-	PS	E1	DVSSB	−	GND pin
-	PS	E11	DVSSB	−	GND pin
-	PS	L6	DVSSB	−	GND pin
-	PS	B9	RVDD3	−	Power supply pin for internal regulator
-	PS	B8	RVSS	−	GND pin for internal regulator
-	PS	A8	DVSSC	−	GND pin

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

PORT	Type	Pin No.	Pin Name	Input/ Out- put	Function
-	PS	C2	AVREFH	I	Power supply pin for AD and DA converters. (note) AVREFH must be connected to power supply even if A/D and D/A converters are not used.
-	PS	D2	AVREFL	I	Power supply pin for AD and DA converters. (note) AVREFL must be connected to GND even if A/D and D/A converters are not used.
-	PS	C1	AVDD3	I	Power supply pin for AD and DA converters. (note) AVDD3 must be connected to power supply even if A/D and D/A converters are not used.
-	PS	D1	AVSS	I	GND pin for AD and DA converters. (note) AVSS must be connected to GND even if A/D and D/A converters are not used.

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

1.5 Pin Numbers and Power Supply Pins

Table 1-3 Pin Numbers and Power Supplies

Power supply	Voltage range	Pin No.	Pin name
DVDD3B	1.65 to 3.6V	K6, E2, F11	PA, PB, PC, PD, PE, PF, BSC
DVDD3A	2.7 to 3.6V	A5, C11	PG, PH, PI, X1, X2, FTEST3, $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, MODE, INTLV
AVDD3		C1	PJ, PK, DA0, DA1, ENDIAN
RVDD3		B9	–

2. Processor Core

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

2.1 Information on the processor core

The following table shows the revision of the processor core in the TMPM341FDXBG/FYXBG.

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
TMPM341FDXBG/ FYXBG	r2p0

2.2 Configurable Options

The Cortex-M3 core has optional blocks. The optional blocks of the revision r2p0 are ETM™, MPU and WIC. The following table shows the configurable options in the TMPM341FDXBG/FYXBG.

Configurable Options	Implementation
FPB	Two literal comparators Six instruction comparators
DWT	Four comparators
ITM	Present
MPU	Absent
ETM	Present
AHB-AP	Present
AHB Trace Macrocell Interface	Absent
TPIU	Present
WIC	Absent
Debug Port	JTAG / Serial wire

2.3 Exceptions/ Interruptions

Exceptions and interruptions are described in the following section.

2.3.1 Number of Interrupt Inputs

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

TMPM341FDXBG/FYXBG has 84 interrupt inputs. The number of interrupt inputs is reflected in <INTLINESNUM[4:0]> bit of NVIC register. In this product, if read <INTLINESNUM[4:0]> bit, 0x00 is read out.

2.3.2 Number of Priority Level Interrupt Bits

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

TMPM341FDXBG/FYXBG has 3 priority level interrupt bits. The number of priority level interrupt bits is used for assigning a priority level in the interrupt priority registers and system handler priority registers.

2.3.3 SysTick

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

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

2.3.4 SYSRESETREQ

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

TMPM341FDXBG/FYXBG provides the same operation when SYSRESETREQ signal are output.

2.3.5 LOCKUP

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

TMPM341FDXBG/FYXBG does not use this signal. To return from LOCKUP status, it is necessary to use non-maskable interruput (NMI) or reset.

2.3.6 Auxiliary Fault Status register

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

However, TMPM341FDXBG/FYXBG is not defined this function. If auxiliary fault status register is read, always "0x0000_0000" is read out.

2.4 Events

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

TMPM341FDXBG/FYXBG does not use event output signals and event input signals. Please do not use SEV instruction and WFE instruction.

2.5 Power Management

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

These signals are output in the following circumstances:

-Wait-For-Interrupt (WFI) instruction execution

-Wait-For-Event (WFE) instruction execution

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

TMPM341FDXBG/FYXBG does not use SLEEPDEEP signal so that <SLEEPDEEP> bit must not be set. And also event signal is not used so that please do not use WFE instruction.

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

2.6 Exclusive access

In Cortex-M3 core, the DCode bus system supports exclusive access. However TMPM341FDXBG/FYXBG does not use this function.

3. Endian

3.1 Endianess of Cortex-M3 core

The Cortex-M3 core has a bi-endian CPU core allowing to support both little-endian and big-endian. The features are as follows:

1. Employed ARM endianess for Big-endian (BE8)

   Big-endian (BE8) specified by ARM is partly different in the format and operation from big-endian specified by MIPS which is used in Toshiba TX19 series.

   The difference between two is referred to Section"3.5.1 Differences in the big-endian format".

2. Space aligned in little-endian always exists

   Even when big-endian is used, CPU core operates and accesses the following area using little-endian.

   - Instruction fetch
   - Internal peripheral bus (0xE0000000 to 0xE003FFFF)
   - External peripheral bus (0xE0040000 to 0xE00FFFFF)

For details, see "Cortex-M series processors" of ARM in below URL.

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

3.2 Endianess of TMPM341FDXBG/FYXBG

This product has a bi-endian CPU core (Cortex-M3) and supports bi-endian operation.

If big-endian is used, users needs some consideration in operation modes, endian-types (BE8 format or MIPS format), operations and access areas (instructions or operands).

The following sections describes specifications and operations in each operation mode.

3.2.1 Single Mode

The Cortex-M3 instructions always use little-endian format, so that a result of compiling using big-endian (BE8) is mixed with little-endian. However a memory, where compiler data is assigned, must use big-endian of Cortex-M3. The internal memory and the memory, which is connected to the external memory, are applied.

To enable data transfer between external devices and the MCU, the MCU allows to use MIPS format data using big-endian in the external bus area. Cortex-M3 format and MIPS format are selectable in each CS space.

Control registers of built-in peripheral function unit uses MIPS format.

Transfer data of DMAC uses MIPS format as well.

3.2.2 Single Boot Mode

Boot programs in the built-in BOOT ROM use little-endian. When single boot mode is used, little-endian operation is used regardless of ENDIAN pin setting. Note that programs and data running on RAM with RAM transfer command must be little-endian format.

3.2.3 Miscellaneous

3.2.3.1 Endian in the DMAC

The DMAC can independently choose endianess with the endian selection bit (DMACxConfiguration<M>). However a user always must set the same endianess as product’s endianess.

3.2.3.2 Endian in the Debug Tools

When debug tools are connected, data must use big-endian of Cortex-M3 since data is observed via internal bus matrix in the CPU. Thus, instructions is observed in little-endian and operands observed in big-endian (BE8).

3.3 How to Set Endianess and Endian type

3.3.1 Operation Setting of TMPM341FDXBG/FYXBG

Endianess of this product is set with operation modes (determined by $\overline{\text{BOOT}}$ pin) and the ENDIAN pin. Additionally, endianess in the external bus area is set with EXBCSx<ENDTYPE> in EBIF. Endianess of DMAC is set with DMACxConfiguration<M>.

Table 3-1 and Table 3-2 show a list of bus masters and endian types in each setting and access area. This product has two bus masters (CPU and DMAC). Access by DMAC is the same as the access of CPU operand.

Table 3-1 Endian types (access by CPU)

		Endianess of product					
		Single mode $\overline{\text{BOOT}}$="High"				Single boot mode $\overline{\text{BOOT}}$="Low"	
		Little-endian ENDIAN="Low"		Big-endian ENDIAN="High"		Little-endian ENDIAN=don’t care	
Access area		Instruction	Operand	Instruction	Operand	Instruction	Operand
Internal FLASH		LE		LE	BE8	- (Disabled)	LE
Internal RAM		LE		LE	BE8	LE	
Internal BOOT ROM		-		-		LE	
Control register of peripheral function unit		LE		-	MIPS	LE	
External memory	EXBCSx<ENDTYPE>="0"	LE		LE	BE8	LE	
	EXBCSx<ENDTYPE>="1"	- (Disabled)	MIPS	- (Disabled)	MIPS	- (Disabled)	MIPS

LE:	Little-endian
BE8:	Big-endian of BE8
MIPS:	Big-endianof MIPS

Table 3-2 Endian types (access by DMAC)

		Endianess of product/DMAC		
		Single mode $\overline{\text{BOOT}}$="High"		Single boot mode $\overline{\text{BOOT}}$="Low"
Access area		Little-endian ENDIAN="Low" DMACxConfiguration <M>="0"	Big-endian ENDIAN="High" DMACxConfiguration <M>="1"	Little-endian ENDIAN=don’t care DMACxConfiguration <M>="0"
Internal FLASH		LE	BE8	LE
Internal RAM		LE	BE8	LE
Internal BOOT ROM		-	-	LE
Control register of peripheral function unit		LE	MIPS	LE
External memory	EXBCSx<ENDTYPE>="0"	LE	BE8	LE
	EXBCSx<ENDTYPE>="1"	MIPS	MIPS	MIPS

LE:	Little-endian
BE8:	Big-endian of BE8
MIPS:	Big-endian of MIPS

3.3.2 Operation Mode

Operation mode is determined by $\overline{\text{BOOT}}$ pin. This pin is booted in the single boot mode when the external reset pin is booted while AVSS is "Low" level. When AVDD3 is "High" level, the pin is booted in the single mode.

In the single boot mode, endianess of the product is little-endian regardless of ENDIAN pin setting.

3.3.3 ENDIAN Pin

This product has an ENDIAN pin to determine endianess. This ENDIAN pin is input-only. When this pin is set to "Low" (AVSS level), if the external reset pin is booted, the pin is booted as little-endian. When the pin is set to "High" (AVDD3 level), the pin is booted as big-endian. As mentioned before, in the single boot mode, little-endian is used regardless of ENDIAN pin setting.

3.3.4 Endian Selection of External Bus Area

Endianess is set with EXBCSx<ENDTYPE> of the external bus interface (EBIF) in each CS space. Endianess of external bus area setting with ENDIAN pin and <ENDTYPE> are as follows:

Endianess	EXBCSx<ENDTYPE>	
	"0" (same as CPU)	"1" (different from CPU)
Little-endian	Little-endian	MIPS type
Big-endian	BE8 type	MIPS type

3.3.5 Endian Selection of DMAC

Endianess is set with DMACxConfiguration<M>. Endianess of DMAC must be the same as those of ENDIAN pin. When <M>="0" is set, little-endian is used. When <M>="1" is set, big-endian is used.

3.4 Structure

This section shows a structure of TMPM341FDXBG/FYXBG. Data conversion circuit exists between AHB bus and bus-bridge, and AHB bus and DMAC respectively to deal with the difference of endianess.

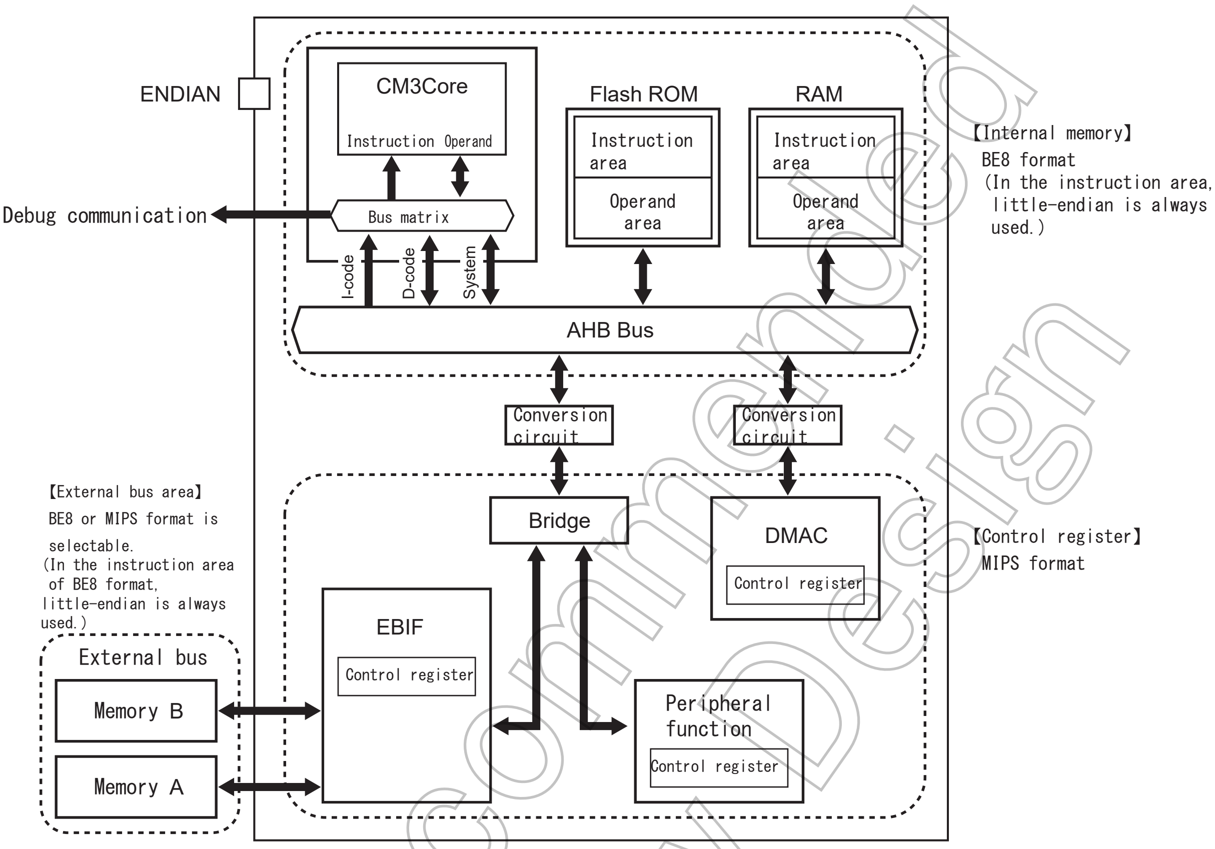


Figure 3-1 TMPM341FDXBG/FYXBG Block diagram

Next chapter describes the operation of little-endian and big-endian in detail.

3.4.1 Block Structure of Little-endian

When little-endian is used, CPU, internal memory and control registers of built-in peripheral function unit use little-endian. Only in the external bus area, either little-endian or big-endian of MIPS can be chosen in each memory (in each chip select signal).

DMAC uses little-endian.

The conversion circuit operates when external bus accesses using MIPS memory access.

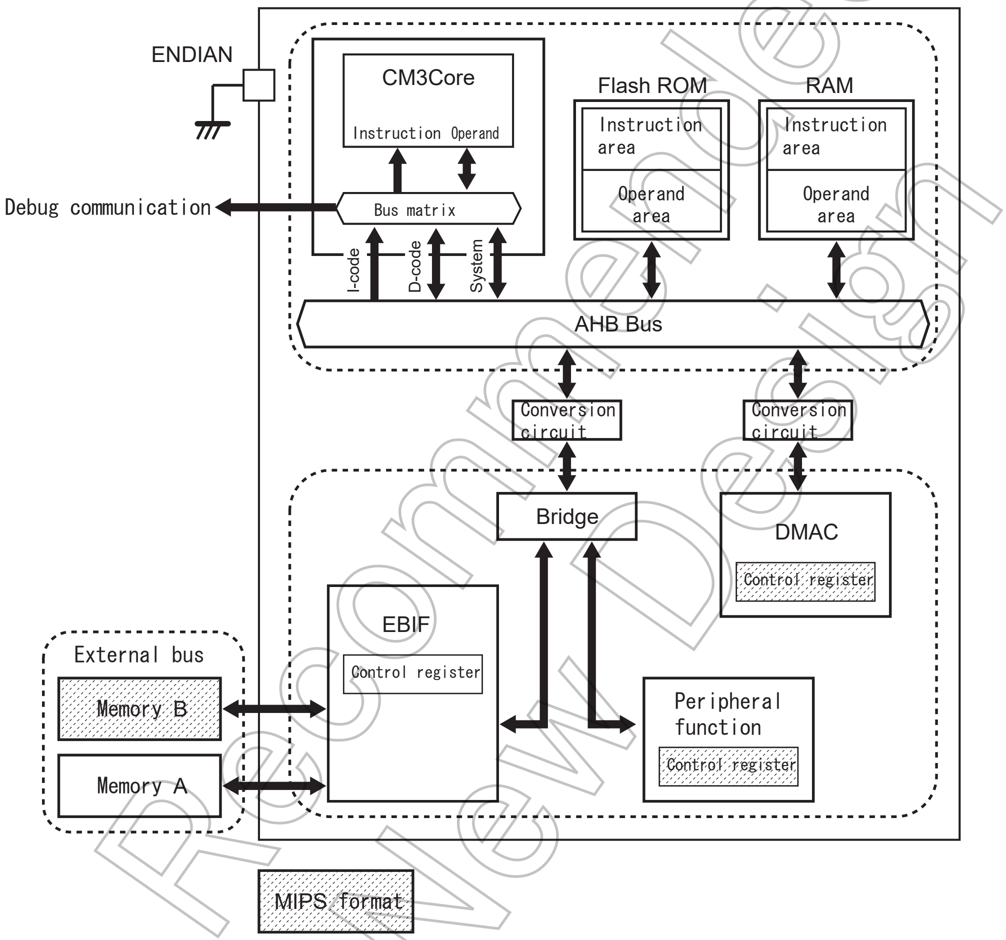


Figure 3-2 Little-endian Block diagram

3.4.2 Block Structure of Big-endian

When big-endian is used, instruction fetches uses little-endian and operand access uses BE8 among accesses from CPU to the internal memory. These operations are basics of Cortex-M3, so that the conversion is not required.

The memory connecting to the external bus can select endianess either BE8 or big-endian of MIPS.

The access to the control register of peripheral function unit is big-endian of MIPS.

DMAC uses big-endian. Transfer data is big-endian of MIPS.

The conversion circuit operates at the following conditions; when big-endian MIPS format data on the external bus accesses memory; when control registers of peripheral function unit access memory; when DMAC performs data transfer.

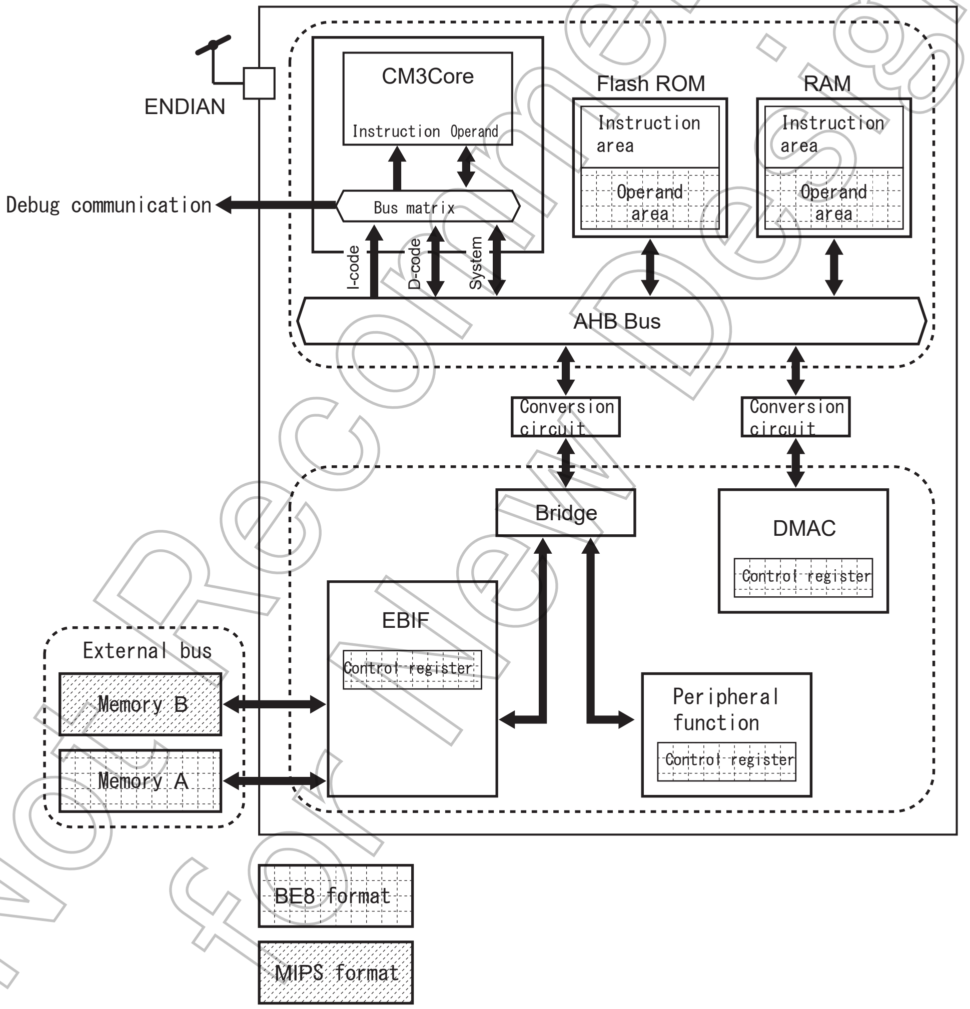


Figure 3-3 Big-endian Block diagram

3.5 Operational Description

3.5.1 Differences in the big-endian format

Data alignment is different between BE8 format and MIPS format.

Data in the general-purpose register of CPU is aligned on the bus connecting to CPU according to data size and address as follows:

< 4-byte data >

	31			0
General-purpose register	DD	CC	BB	AA

	<Little Endian>				<BE8>				<MIPS>			
	31			0	31			0	31			0
Data on the bus	DD	CC	BB	AA	AA	BB	CC	DD	DD	CC	BB	AA

< 2-byte data >

	31			0
General-purpose register			BB	AA

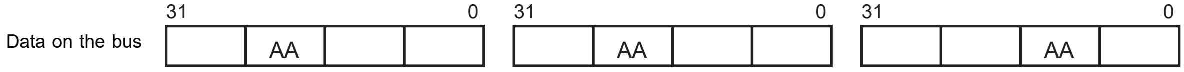
	<Little Endian>				<BE8>				<MIPS>			
[Address: 0x00]												
	31			0	31			0	31			0
Data on the bus			BB	AA			AA	BB	BB	AA		
[Address: 0x02]												
	31			0	31			0	31			0
Data on the bus	BB	AA			AA	BB					BB	AA

< 1-byte data >

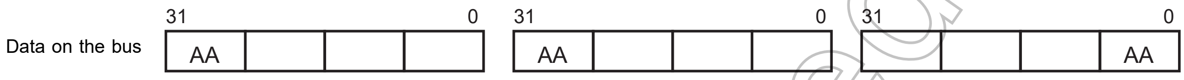
	31			0
General-purpose register				AA

	<Little Endian>				<BE8>				<MIPS>			
[Address: 0x00]												
	31			0	31			0	31			0
Data on the bus				AA				AA	AA			
[Address: 0x01]												
	31			0	31			0	31			0
Data on the bus			AA				AA			AA		

[Address: 0x02]



[Address: 0x03]

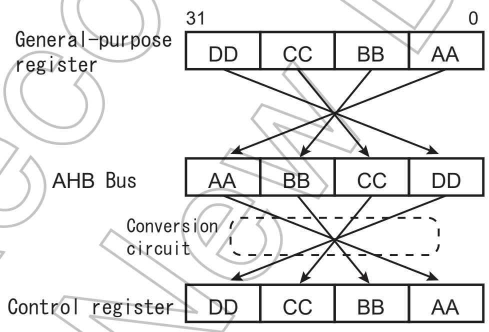


3.5.2 Access of Control Register

When the control registers of peripheral function unit is accessed using big-endian, i.e. operand access (BE8) by CPU, data alignment output to the bus is set as described in Section"3.5.1 Differences in the big-endian format". The data is converted in the conversion circuit which is located between AHB bus and the bus connecting to peripheral function unit. Data alignment is MIPS type when data is written to the control register.

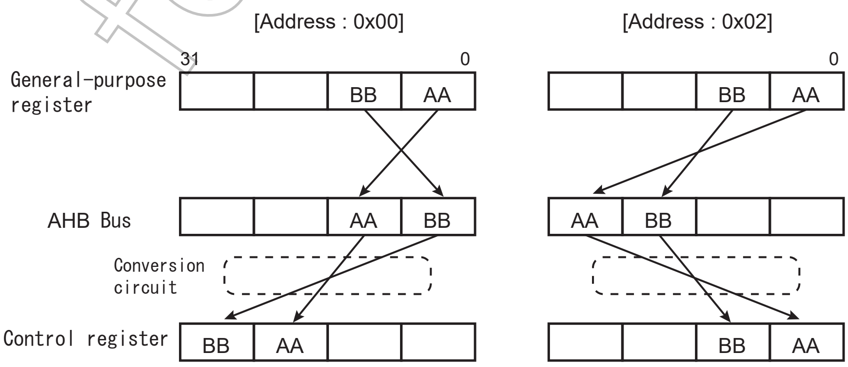
Usually, when big-endian is used in the Cortex-M3, write data to the control register is prepared on having some considerations to data alignment for BE8. In the MIPS format, data alignment is the same as general-purpose registers, so that write data can be used as it is in this product.

< 4-byte data >

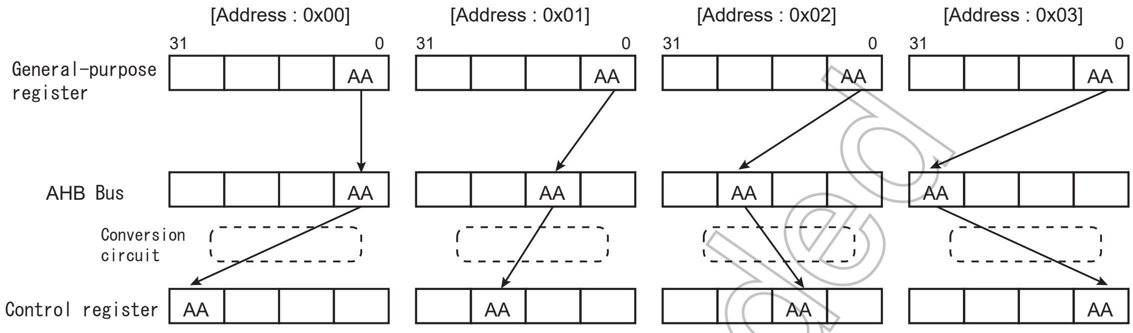


Data output position when the 2-byte data access and the 1-byte data access using big-endian is as follows. Note that little-endian address and big-endian address seem to be different.

< 2-byte data >



< 1-byte data >



The following table shows address differences depending on endianess taking example of EXBCS0 register in the external bus interface (EBIF).

Address	Bit	31	30	29	28	27	26	25	24
Little : 0x4005_C003	Symbol	CSR		WRR			RDR		
Big : 0x4005_C000	Initial value	0	1	0	0	1	0	0	1
	Bit	23	22	21	20	19	18	17	16
Little : 0x4005_C002	Symbol	-	-	ALEW		WRS		RDS	
Big : 0x4005_C001	Initial value	0	0	0	1	0	1	0	1
	Bit	15	14	13	12	11	10	9	8
Little : 0x4005_C001	Symbol	-	-	-	CSIW				
Big : 0x4005_C002	Initial value	0	0	0	0	0	0	1	0
	Bit	7	6	5	4	3	2	1	0
Little : 0x4005_C000	Symbol	-	-	-	-	-	CSW		CSW0
Big : 0x4005_C003	Initial value	0	0	0	0	0	0	1	0

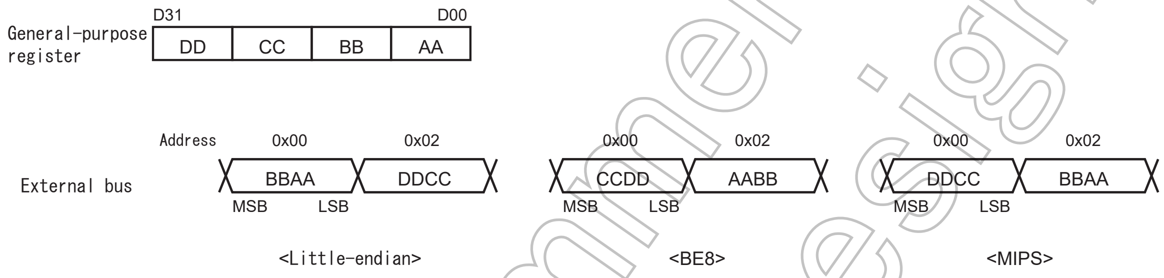
3.5.3 External Bus Operation

Data format of external bus access is determined by ENDIAN pin and EXBCSx<ENDTYPE> setting. (See Section"3.3.4 Endian Selection of External Bus Area")

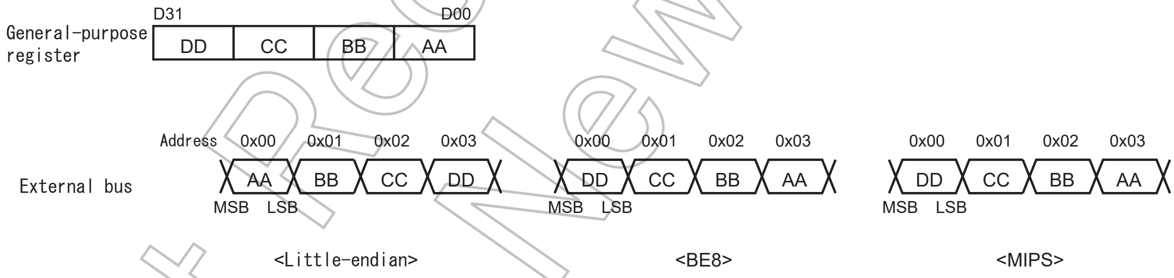
Data formats of little-endian, BE8 and MIPS on the external bus are shown in below.

3.5.3.1 Data Size: 32-bit

(1) External bus width: 16-bit

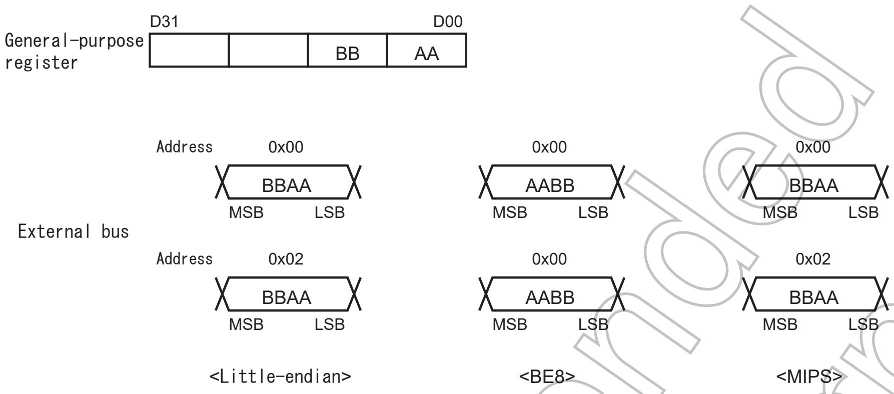


(2) External bus width: 8-bit

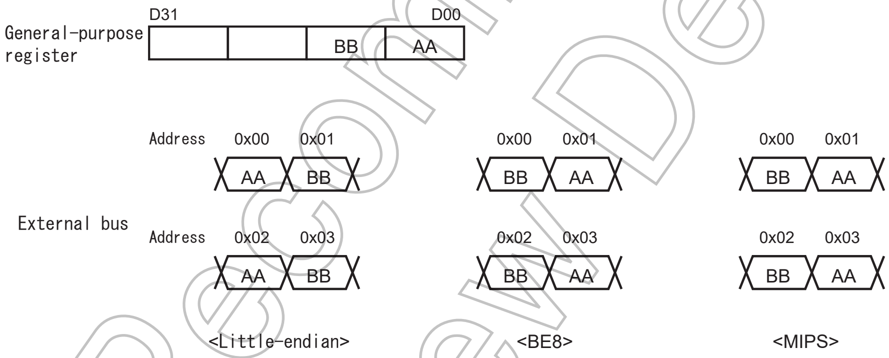


3.5.3.2 Data Size (16-bit)

(1) External bus width (16-bit)

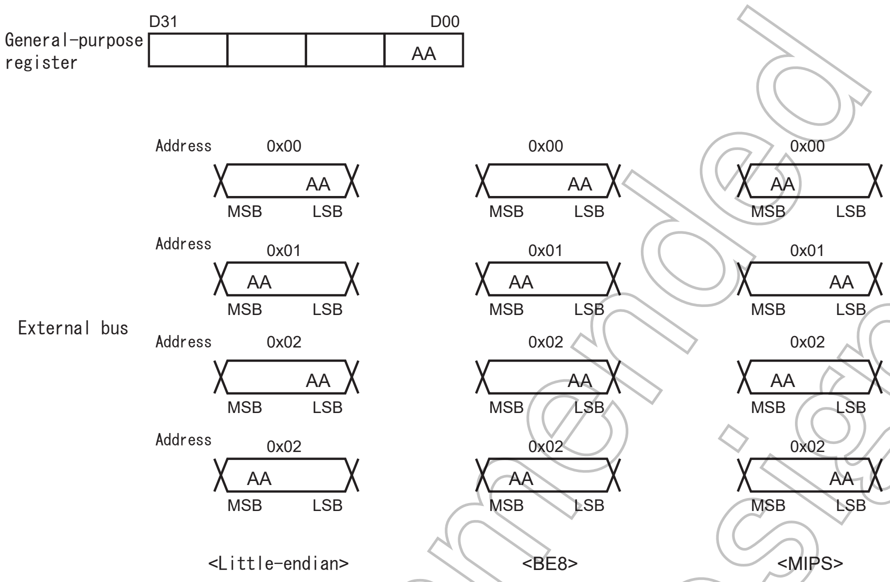


(2) External bus width (8-bit)

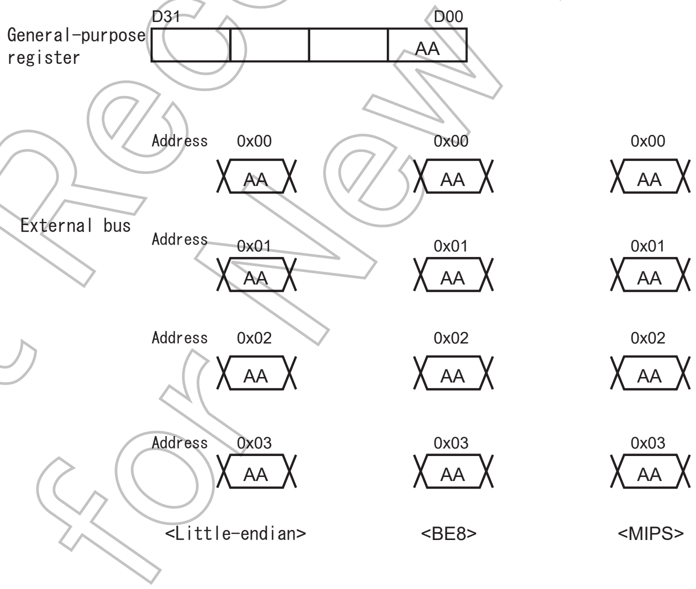


3.5.3.3 Data Size (8-bit)

(1) External bus width (16-bit)



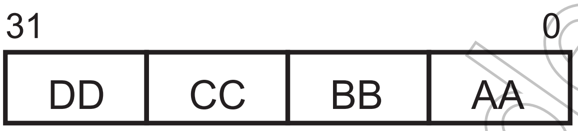
(2) External bus width (8-bit)



3.5.4 Image of Compile Result

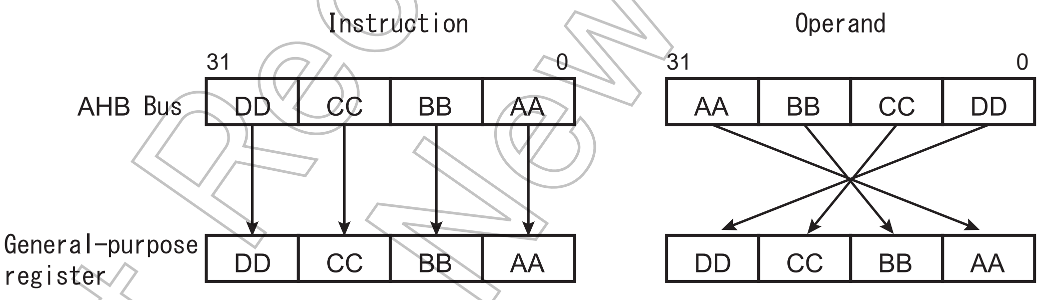
This section describes the data alignment in the memory, e.g. Motorola S record or Intel HEX.

If the following memory image of 32-bit data is assumed, the data alignment will be shown in the table below by setting endianess when compiling.



Endian		Address			
		0x00	0x01	0x02	0x03
Little-endian		0xAA	0xBB	0xCC	0xDD
Big-endian (Cortex-M3)	Instruction (Little-endian)	0xAA	0xBB	0xCC	0xDD
	Operand (BE8)	0xDD	0xCC	0xBB	0xAA
Big-endian (MIPS)		0xDD	0xCC	0xBB	0xAA

When using big-endian in the Cortex-M3, instructions use little-endian and operands use BE8 type. When CPU accesses operands using big-endian, CPU switches data alignment when read as shown below. This causes that an addressing of instructions (little-endian) and operands (BE8) seems to be same. Memory image is generated based on the above consideration, so that all data addressing can be treated as little-endian.



3.5.5 Operation in the Single Boot Mode

An operations in the single boot mode is a little-endian operation regardless of ENDIAN pin setting, since boot programs in the internal BOOT ROM are little-endian. In the boot program, a RAM transfer command is prepared. The data transfer program is developed on the RAM using this command and the data is transferred to the internal memory or external memory.

Note: Programs or data developed on the RAM using RAM transfer command must be prepared with little-endian format since CPU uses little-endian.

This section describes specific operation that a compile result (memory image), which is caused by different data format from little-endian, is transferred to the internal memory or external memory.

A memory image is the same as the example in Section"3.5.4 Image of Compile Result". The following example is Motorola S format.

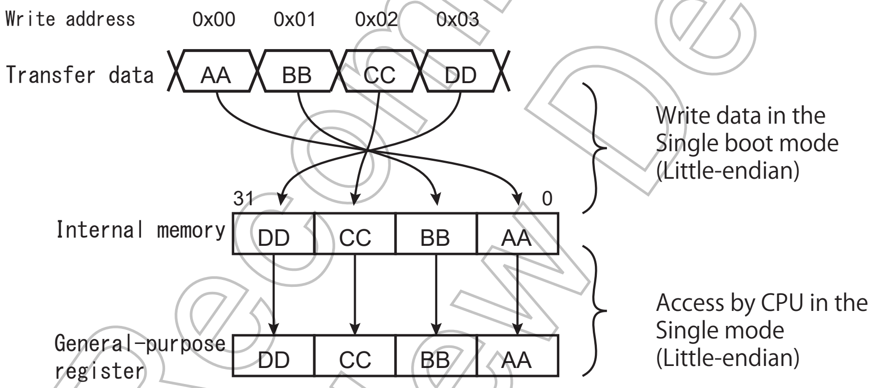
Little-endian		S1xx0000	AA	BB	CC	DD	:
Big-endian (Cortex-M3)	Instruction (Little-endian)	S1xx0000	AA	BB	CC	DD	:
	Operand (BE8)	S1xx0000	DD	CC	BB	AA	:
Big-endian (MIPS)		S1xx0000	DD	CC	BB	AA	:

3.5.5.1 Data Transfer to the Internal Memory

This section describes an example that 32-bit data is transferred with 1 byte at a time from the external device to the internal memory.

(1) Data Transfer using Little-endian Format

When little-endian is used, desired data is prepared by writing an address to the memory incrementing from 0x00.

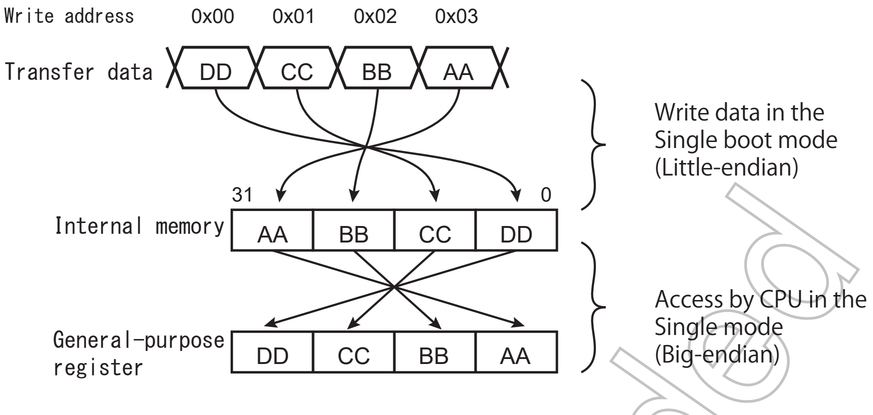


(2) Cortex-M3 Big-endian Transfer

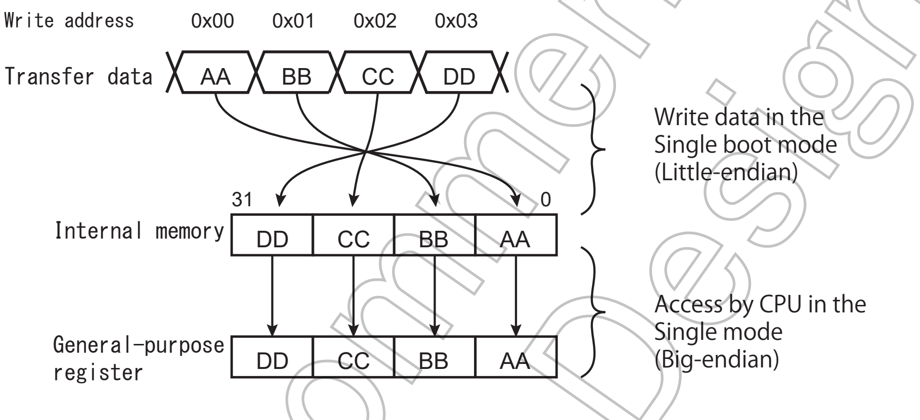
When data, which is mixed instructions (little-endian) with operands (BE8), is written to memory using little-endian, addressing is considered on becoming the same operation as little-endian operation as shown in Section"3.5.4 Image of Compile Result".

When CPU reads data, instructions are taken as it is; however operands is switched bytes around.

- Operand



- Instructions



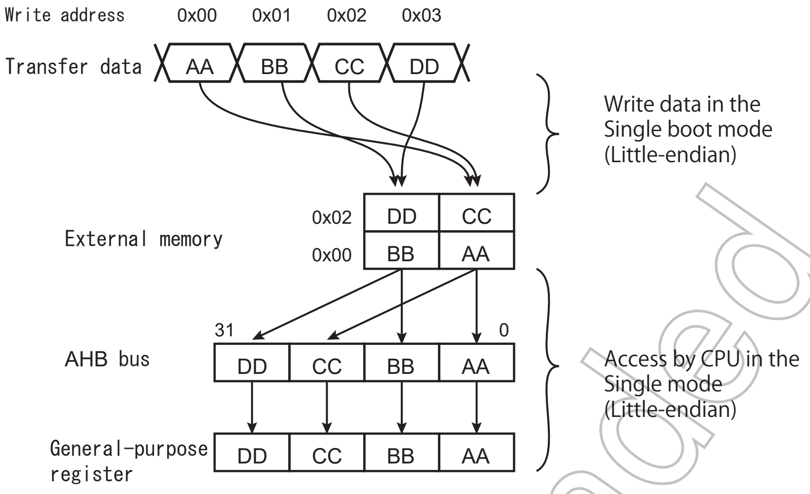
3.5.5.2 Data Transfer to External Memory

This section describes an example that 32-bit data is transferred to the 16-bit width external memory with 1 byte at a time.

(1) Data Transfer using Little-endian Format

When little-endian is used, a desired data is prepared by writing an address to the memory incrementing from 0x00.

When the CPU reads 32-bit data, each 16-bit read data is assigned from lower bit in the external bus interface to generate 32-bit data and outputs the data to the bus.

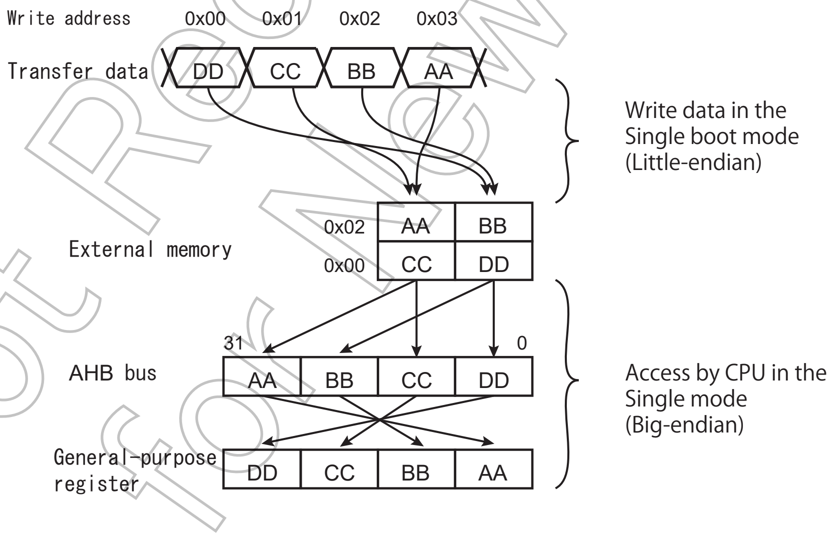


(2) Data Transfer using BE8 Format

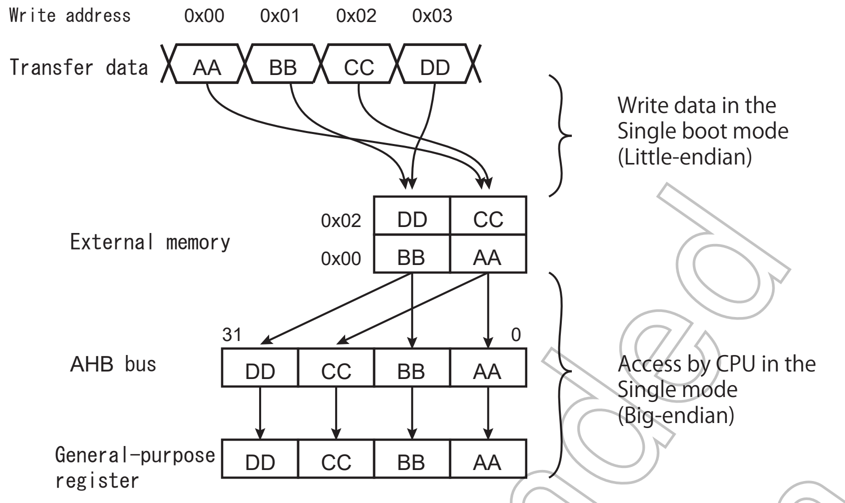
When data, which is mixed instructions (little-endian) with operands (BE8), is written to memory using little-endian, addressing is considered on becoming the same operation as little-endian operation as shown in Section"3.5.4 Image of Compile Result".

When CPU reads 32-bit data, each 16-bit read data is assigned from upper bit in the external bus interface to generate 32-bit data and outputs the data to the bus. When CPU reads data, instructions are taken as it is; however operands is switched bytes around.

- Operand



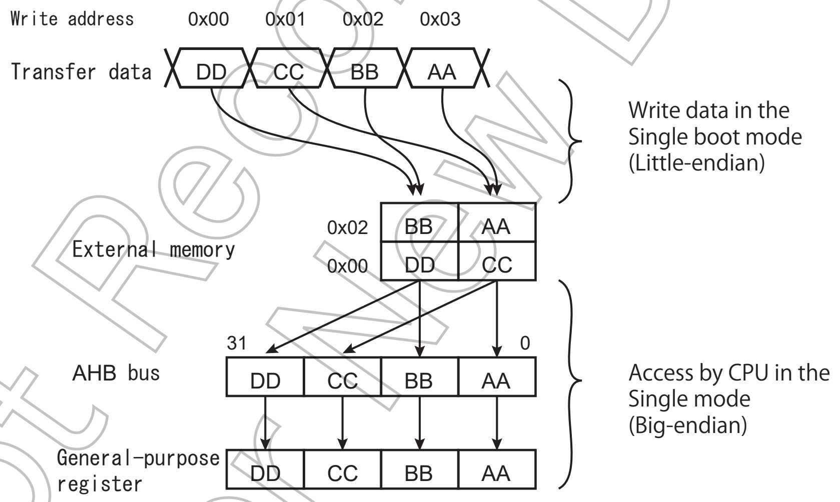
- Instruction



(3) Data Transfer using MIPS Format

To write MIPS format data to memory using little-endian, use EXBCSx<ENDTYPE> function of the external bus interface (EBIF). If <ENDTYPE> is set to "1", a corresponding CS area becomes MIPS format and EBIF automatically controls memory access.

When CPU reads 32-bit data, each 16-bit read data is assigned from upper bit in the external bus interface to generate 32-bit data and outputs the data to the bus.



4. Memory Map

4.1 Memory map

The memory maps for theTMPM341FDXBG/FYXBG are based on the ARM Cortex-M3 processor core memory map.

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

The special function register (SFR) indicates I/O ports and control registers for the peripheral function. The SRAM and SFR regions are all included in the bit-band region.

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

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

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

4.1.1 Memory map of the TMPM341FDXBG/FYXBG

Figure 4-1 and Figure 4-2 show the memory map of the TMPM341FDXBG/FYXBG.

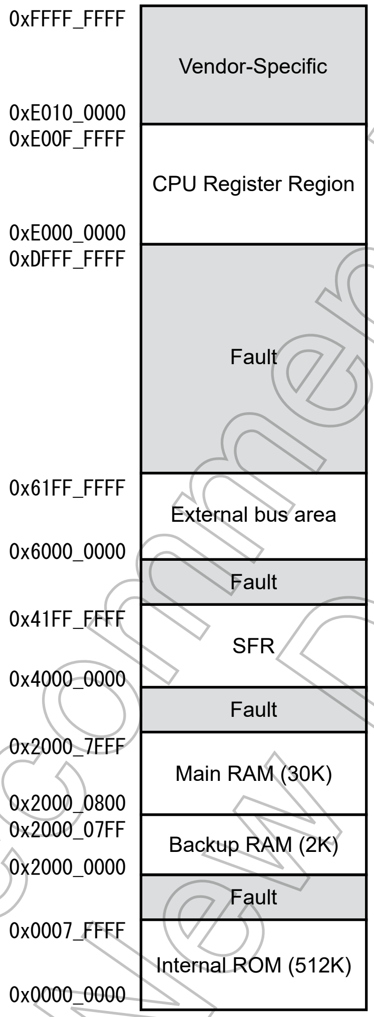


Figure 4-1 Memory Map (TMPM341FDXBG)

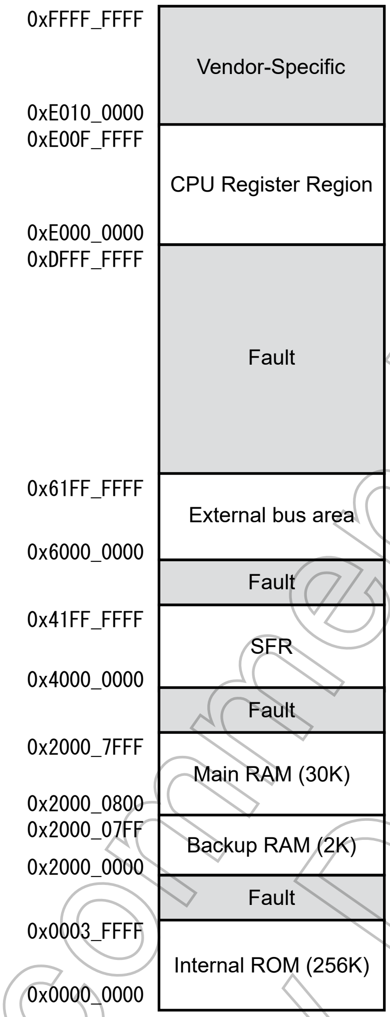


Figure 4-2 Memory Map (TMPM341FYXBG)

4.2 SFR area detail

This section contains the list of addresses in the SFR area (0x4000_0000 through 0x41FF_FFFF) assigned to peripheral function.

Access to the Reserved areas in the Table 4-1, the address not specified and the Reserved area in each chapter are prohibited. As for the SFR area, the areas not specified in each chapter is read an undefined value. Writing this area is ignored.

Table 4-1 SFR area detail

Start Address	End Address	Peripheral
0x4000_0000	0x4000_3FFF	DMAC (4ch)
0x4000_4000	0x4003_FFFF	Reserved
0x4004_0000	0x4004_7FFF	SSP (1ch)
0x4004_8000	0x4004_FFFF	Reserved
0x4005_0000	0x4005_3FFF	ADC (15ch)
0x4005_4000	0x4005_7FFF	DAC (2ch)
0x4005_8000	0x4005_BFFF	TMRD (2ch)
0x4005_C000	0x4005_CFFF	EBIF
0x4006_D000	0x400B_FFFF	Reserved
0x400C_0000	0x400C_3FFF	PORT (A to K)
0x400C_4000	0x400C_6FFF	TMRB (10ch)
0x400C_7000	0x400C_9FFF	Reserved
0x400C_A000	0x400C_BFFF	PHCNT (4ch)
0x400C_C000	0x400D_FFFF	Reserved
0x400E_0000	0x400E_0FFF	I2C/SIO (2ch)
0x400E_1000	0x400E_5FFF	SIO/UART (5ch)
0x400E_6000	0x400F_0FFF	Reserved
0x400F_1000	0x400F_1FFF	OFD
0x400F_2000	0x400F_2FFF	WDT
0x400F_3000	0x400F_3FFF	CG
0x400F_4000	0x41FF_FFFF	Reserved

5. External bus interface (EBIF)

5.1 Overview

The TMPM341FDXBG/FYXBG has a built-in external bus interface function to connect to external memory, I/Os, etc. This interface consists of an external bus interface circuit (EBIF), a chip selector (CS) and a wait controller.

The chip selector and wait controller designate mapping addresses in a 2-block address space and also control wait states and data bus widths (8- or 16-bit) in these and other external address spaces.

The external bus interface circuit (EBIF) controls the timing of external buses based on the chip selector and wait controller settings.

This product is a bi-endian capable product that supports both little and big endian byte ordering used for the CPU core Cortex-M3. This chapter mainly describes the CS/Wait controller. For information about bi-endian, see the chapter Endian. For detailed information, refer to the manual "Cortex-M series processors" at the below URL.

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

Table 5-1 Features of External bus interface

features	
Memory supports	Asynchronous memory (NOR Flash memory, SRAM, Peripheral I/O and e.t.c.) Selectable separate bus mode or multiplex bus mode.
Data bus width	Either an 8- or 16-bit width can be set for each channel.
Chip select	2 channels ($\overline{\text{CS0}}$, $\overline{\text{CS1}}$)
Endian	Bi-Endian is supported.
Address access spaces	Supports up to 16MB memory spaces CS0: 0x6000_0000 to 0x61FF_FFFF (Max. 16MB space) CS1: 0x6000_0000 to 0x61FF_FFFF (Max. 16MB space)
Internal wait function	This function can be enabled for each channel. A wait of up to 15 cycles can be automatically inserted.
ALE wait function	This function can be enabled for each channel. An ALE high pulse of up to 4 cycles can be automatically inserted.
Setup cycle insertion function	This function can be enabled for each channel. A $\overline{\text{RD}}$ or $\overline{\text{WR}}$ setup cycle can be automatically inserted. (tAC cycle expanded)
Recovery (Hold) cycle insertion function	When an external bus is selected, a dummy cycle of up to 8 clocks can be inserted and this dummy cycle can be specified for each channel. (tCAR, tRAE cycles expanded)
Bus expansion function	The internal wait, the ALE wait, the Setup wait and the Recovery cycle can be expanded double or quadruple.
Control pins	Separate bus mode: D[15:0], A[23:0], $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BELL}}$, $\overline{\text{BELH}}$, $\overline{\text{CS0}}$, $\overline{\text{CS1}}$, ENDIAN
	Multiplex bus mode: AD[15:0], A[23:16], $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BELL}}$, $\overline{\text{BELH}}$, $\overline{\text{CS0}}$, $\overline{\text{CS1}}$, ALE, ENDIAN

5.2 Address and Data Pins

5.2.1 Address and Data pin setting

The TMPM341FDXBG/FYXBG can be set to either separate bus or multiplexed bus mode. Setting the bit <EXBSEL> of EXBMOD register to "1" as the separate bus mode, and setting to "0" as the multiplexed bus mode. Port pins A to E which are to be connected to external devices (memory), are used as address buses, data buses and address/data buses. Table 5-2 shows these.

Table 5-2 Bus Mode, Address and Data Pins

PORT	Separate EXBMOD<EXBSEL> = "1"	Multiplex EXBMOD<EXBSEL> = "0"
Port A (PA0 to PA7)	D0 to D7	AD0 to AD7
Port B (PB0 to PB7)	D8 to D15 A0 to A7	AD8 to AD15
Port C (PC0 to PC7)	A0 to A7	-
Port D (PD0 to PD7)	A8 to A15	-
Port E (PE0 to PE7)	A16 to A23	A16 to A23

Each port is put into input mode after a reset. To access an external device, set the address and data bus functions by using the port control register (PxCR) and the port function register (PxFCm), and set the input enable register (PxIE).

When the access changing from the external area to internal area, the address buses are kept the previously external area address output and the data buses will be high impedance.

5.3 Registers

5.3.1 Registers List

Address and names of EBIF control registers are shown below.

Base Address = 0x4005_C000

Register name		Address (Base+)
External Bus Mode Control Register	EXBMOD	0x0000
Reserved	-	0x0004 to 0x000C
External Bus Area and Start Address Configuration Register 0	EXBAS0	0x0010
External Bus Area and Start Address Configuration Register 1	EXBAS1	0x0014
Reserved	-	0x0018 to 0x003C
External Bus Chip Select Control Register 0	EXBCS0	0x0040
External Bus Chip Select Control Register 1	EXBCS1	0x0044
Reserved	-	0x0048 to 0x0FFC

Note 1: Access the registers by using word reads and word writes.

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

5.3.2 EXBMOD (External Bus Mode Control Register)

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

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as "0".
2-1	EXBWAIT[1:0]	R/W	Bus cycle extension 00: None 01: Double 10: Quadruple 11: Prohibited These bits are used to set the setup, wait and recovery of the bus cycle to be double or quadruple. For example, if a Read setup cycle is set as two cycles by setting <EXBWAIT>="00" (no extension), the two cycles can be quadruplicated by changing the bit setting to <EXBWAIT>="01" (double). It also can be octuplicated by setting the bits to <EXBWAIT>="10" (quadruple). The extended cycle is configured by setting Read/Write setup, chip select/Read/Write recovery, ALE/internal wait cycle and <EXBWAIT> (double or quadruple).
0	EXBSEL	R/W	Select external bus mode (Note) 0: Multiplex bus 1: Separate bus

Note: Do not change the setting of external bus mode in operating the external bus access.

5.3.3 EXBASx (External Bus Area and Start Address 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	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	EXAR							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-24	–	R/W	Always write to "0110_0000"
23-16	SA23-SA16	R/W	Chip select Start address (Note) The A[23:16] is specified as start address.
15-8	–	R	Read as "0".
7-0	EXAR[7:0]	R/W	Chip select Address space size The size of address space can be specified nine kind of setting from 64Kbyte up to 16Mbyte. "0000_0000": 16 Mbyte, "0000_0011": 2 Mbyte, "0000_0110": 256 Kbyte, "0000_0001": 8 Mbyte, "0000_0100": 1 Mbyte, "0000_0111": 128 Kbyte, "0000_0010": 4 Mbyte, "0000_0101": 512 Kbyte, "0000_1000": 64 Kbyte, Others: Prohibited

Note 1: If same address space is specified between CS0 and CS1, the chip selector will be given priority to CS0.

Note 2: If the access address is exceeded space in 0x6000_0000 to 0x61FF_FFFF, a hard fault error will be generated.

5.3.4 EXBCSx (External Bus Chip Select Control Register)

	31	30	29	28	27	26	25	24
bit symbol	CSR		WRR			RDR		
After reset	0	1	0	0	1	0	0	1
	23	22	21	20	19	18	17	16
bit symbol	-	-	ALEW		WRS		RDS	
After reset	0	0	0	1	0	1	0	1
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	CSIW				
After reset	0	0	0	0	0	0	1	0
	7	6	5	4	3	2	1	0
bit symbol	ENDTYPE	-	-	-	-	CSW		CSW0
After reset	0	0	0	0	0	0	1	0

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-30</td><td>CSR[1:0]</td><td>R/W</td><td>Chip select ($\overline{\text{CSx}}$) Recovery cycle
"00": None, "01": 1 cycle, "10": 2 cycles, "11": 4 cycles</td></tr>
<tr><td>29-27</td><td>WRR[2:0]</td><td>R/W</td><td>Write ($\overline{\text{WR}}$) Recovery cycle
"000": None, "001": 1 cycle, "010": 2 cycles, "011": 3 cycles,
"100": 4 cycles, "101": 5 cycles, "110": 6 cycles, "111": 8 cycles</td></tr>
<tr><td>26-24</td><td>RDR[2:0]</td><td>R/W</td><td>Read ($\overline{\text{RD}}$) Recovery cycle
"000": None, "001": 1 cycle, "010": 2 cycles, "011": 3 cycles,
"100": 4 cycles, "101": 5 cycles, "110": 6 cycles, "111": 8 cycles</td></tr>
<tr><td>23-22</td><td>−</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>21-20</td><td>ALEW[1:0]</td><td>R/W</td><td>ALE wait cycle for multiplex bus
"000": None, "001": 1 cycle, "010": 2 cycles, "011": 4 cycles</td></tr>
<tr><td>19-18</td><td>WRS[1:0]</td><td>R/W</td><td>Write ($\overline{\text{WR}}$) Setup cycle
"000": None, "001": 1 cycle, "010": 2 cycles, "011": 4 cycles</td></tr>
<tr><td>17-16</td><td>RDS[1:0]</td><td>R/W</td><td>Read ($\overline{\text{RD}}$) Setup cycle
"000": None, "001": 1 cycle, "010": 2 cycles, "011": 4 cycles</td></tr>
<tr><td>15-13</td><td>−</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>12-8</td><td>CSIW[4:0]</td><td>R/W</td><td>Internal Wait (Automatically insertion)
0_0000: 0 wait, 0_0001: 1 waits, 0_0010: 2 waits, 0_0011: 3 waits,
0_0100: 4 waits, 0_0101: 5 waits, 0_0110: 6 waits, 0_0111: 7 waits,
0_1000: 8 waits, 0_1001: 9 waits, 0_1010: 10 waits, 0_1011: 11 waits,
0_1100: 12 waits, 0_1101: 13 waits, 0_1110: 14 waits, 0_1111: 15 waits,</td></tr>
<tr><td>7</td><td>ENDTYPE</td><td>W</td><td>Classifies an endian type for external memory or peripheral I/O (ASIC, etc.)
<table>
<tr><td rowspan="2">Endian setting</td><td colspan="2">&lt;ENDTYPE&gt;</td></tr>
<tr><td>"0"
(Same endian as CPU)</td><td>"1"
(Not same endian as CPU)</td></tr>
<tr><td>Little Endian</td><td>Little Endian</td><td>MIPS type</td></tr>
<tr><td>Big Endian</td><td>BE8</td><td>MIPS type</td></tr>
</table></td></tr>
<tr><td>6-4</td><td>−</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>3</td><td>−</td><td>R/W</td><td>Always write to "0"</td></tr>
<tr><td>2-1</td><td>CSW[2:1]</td><td>R/W</td><td>Data bus width
"00": 8-bit, "01": 16-bit, Others: Prohibited</td></tr>
<tr><td>0</td><td>CSW0</td><td>R/W</td><td>CS Enable
"0": Disable, "1": Enable</td></tr>
</table>

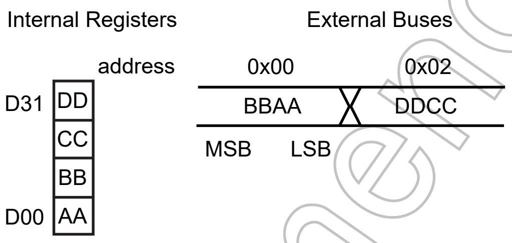
5.4 Data Format

Internal registers and external bus interfaces of the TMPM341FDXBG/FYXBG are configured as described below.

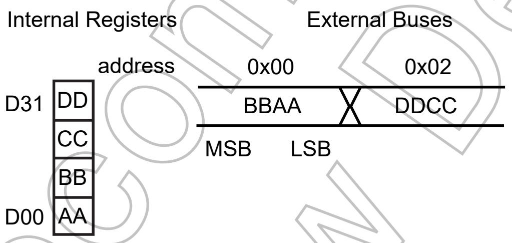
5.4.1 Little-endian mode

5.4.1.1 Word access

- 16-bit bus width

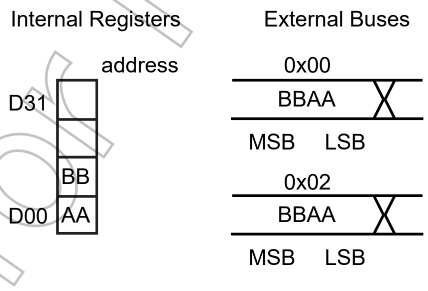


- 8-bit bus width

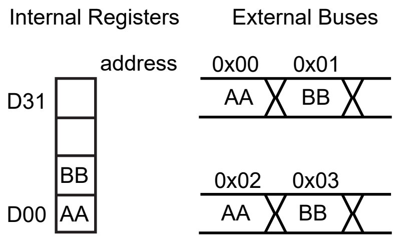


5.4.1.2 Half word access

- 16-bit bus width

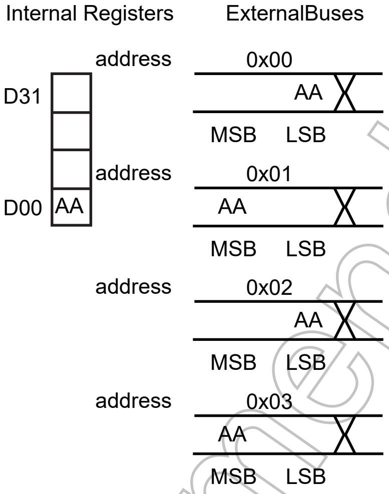


- 8-bit bus width

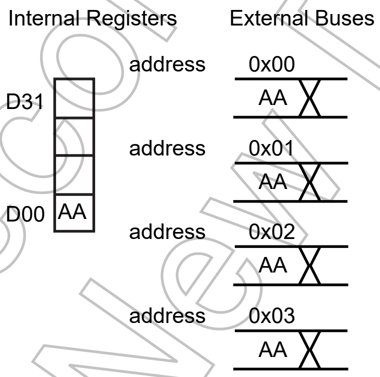

Internal Registers
External Buses
address
D31
D00
BB
AA
0x00
0x01
AA
BB
0x02
0x03
AA
BB

5.4.1.3 Byte access

- 16-bit bus width



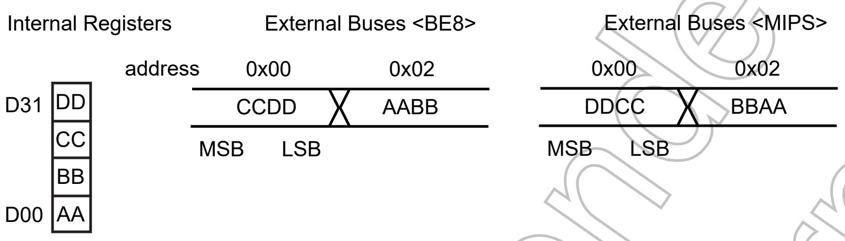
- 8-bit bus width



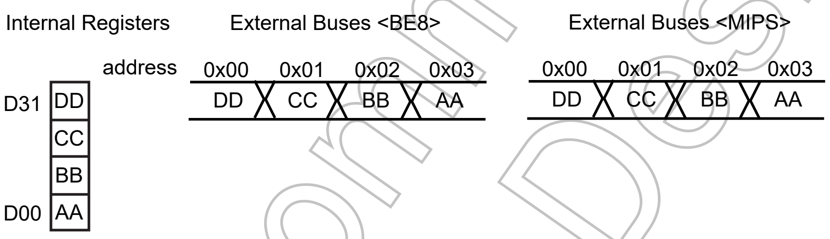
5.4.2 Big-endian mode

5.4.2.1 Word access

- 16-bit bus width

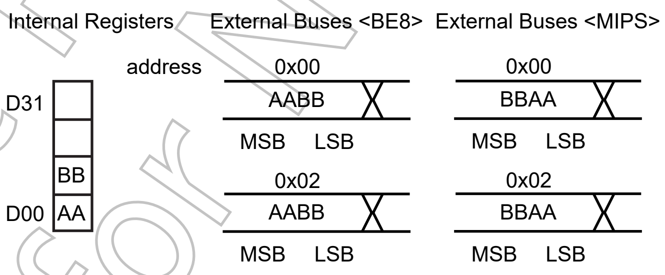


- 8-bit bus width

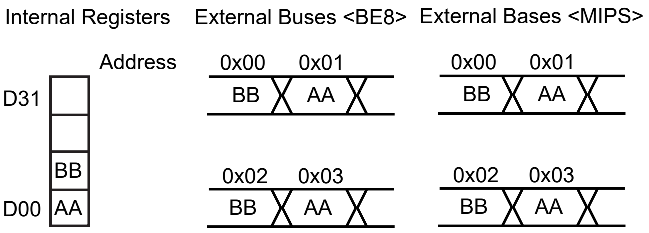


5.4.2.2 Half word access

- 16-bit bus width

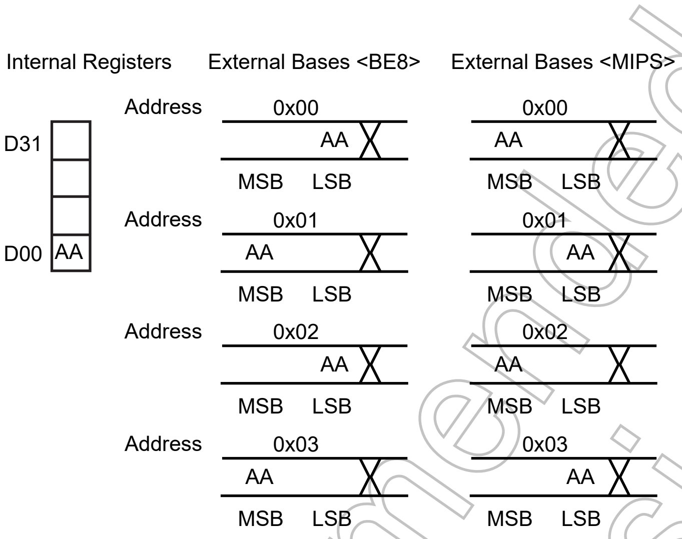


- 8-bit bus width

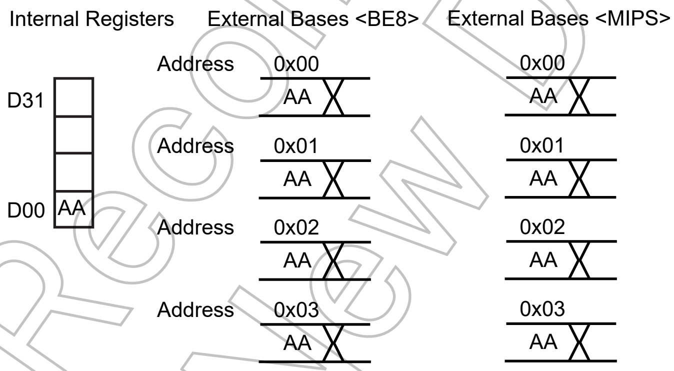


5.4.2.3 Byte access

- 16-bit bus width



- 8-bit bus width



5.5 External Bus Operations (Separete Bus Mode)

This section describes various bus timing values. The timing diagram shown below assumes that the address buses are A23 through A0 and that the data buses are D15 through D0.

5.5.1 Basic bus operation

The external bus cycle of the TMPM341FDXBG/FYXBG basically consists of three clock pulses. The basic clock of an external bus cycle is the same as the internal system clock.Figure 5-1 shows read bus timing and Figure 5-2 shows write bus timing. If internal areas are accessed, address buses remain unchanged as shown in these figures. Additionally, data buses are in a state of high impedance and control signals such as $\overline{RD}$ and $\overline{WR}$ do not become active.

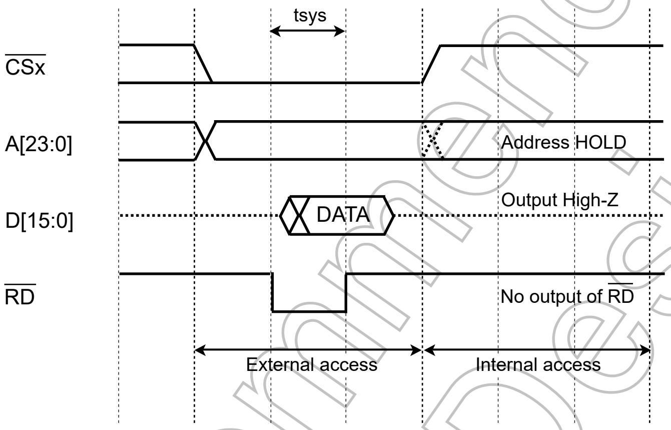


Figure 5-1 Read Operation Timing

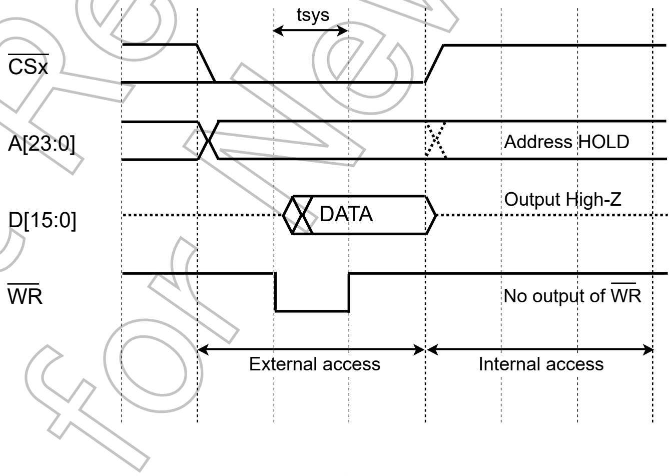


Figure 5-2 Write Operation Timing

5.5.2 Wait timing

A wait cycle can be inserted for each channel by using the chip selector (CS) and wait controller.

The following wait can be inserted.

- A wait of up to 15 clocks can be automatically inserted.

The setting of the number of waits to be automatically inserted and the setting can be made using the chip select control registers, EXBCSx<CSIW[4:0]>.

Figure 5-3 through Figure 5-4 show the timing diagrams in which waits have been inserted.

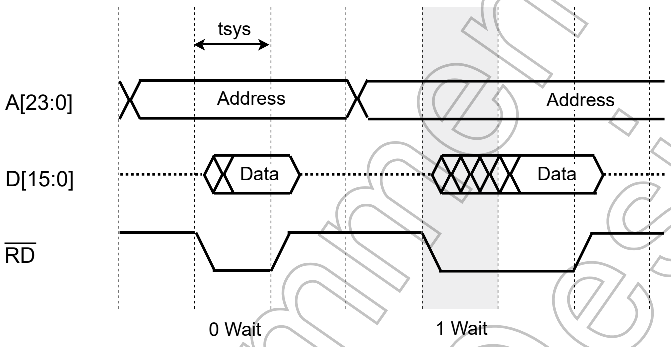


Figure 5-3 Read Operation Timing (0 Wait and 1 Wait Automatically Inserted)

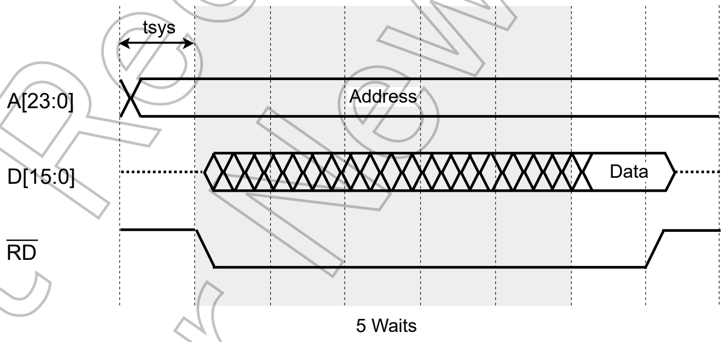


Figure 5-4 Read Operation Timing (5 Waits Automatically Inserted)

Figure 5-5 through Figure 5-6 shows the read and write operation timing when 0 wait and 2 waits automatically inserted in the separate bus mode.

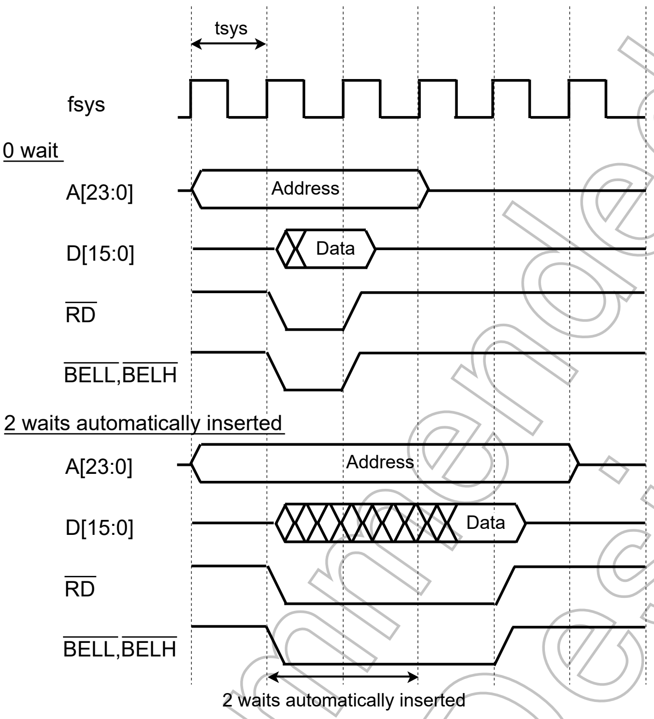


Figure 5-5 Read Operation Timing

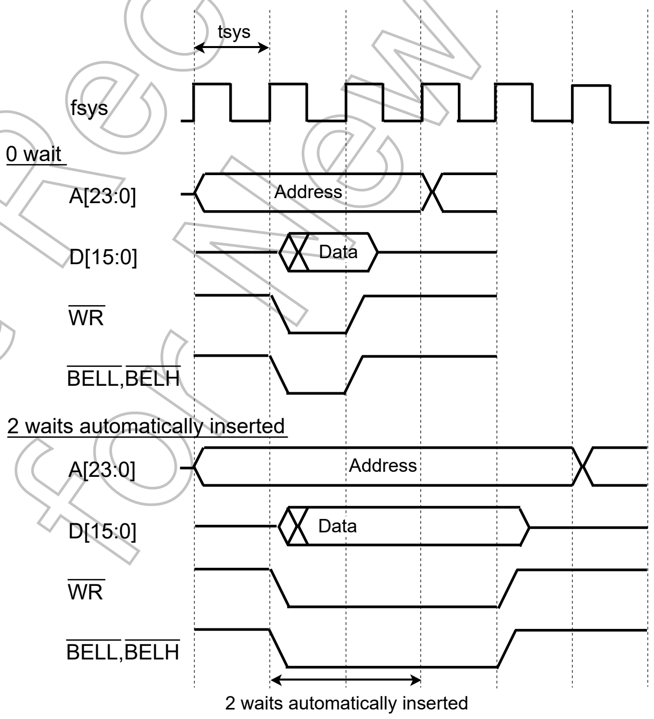


Figure 5-6 Write Operation Timing

5.5.3 Read and Write Recovery time

If access to external areas occurs consecutively, a dummy cycle can be inserted for recovery time.

A dummy cycle can be inserted in both a read and a write cycle. The dummy cycle insertion setting can be made in the chip select control registers, EXBCSx<WRR[2:0]> (write recovery cycle) and <RDR[2:0]> (read recovery cycle). As for dummy cycle, none, one to six or eight system clocks (internal) can be specified for each channel. Figure 5-7 shows the timing of recovery time insertion.

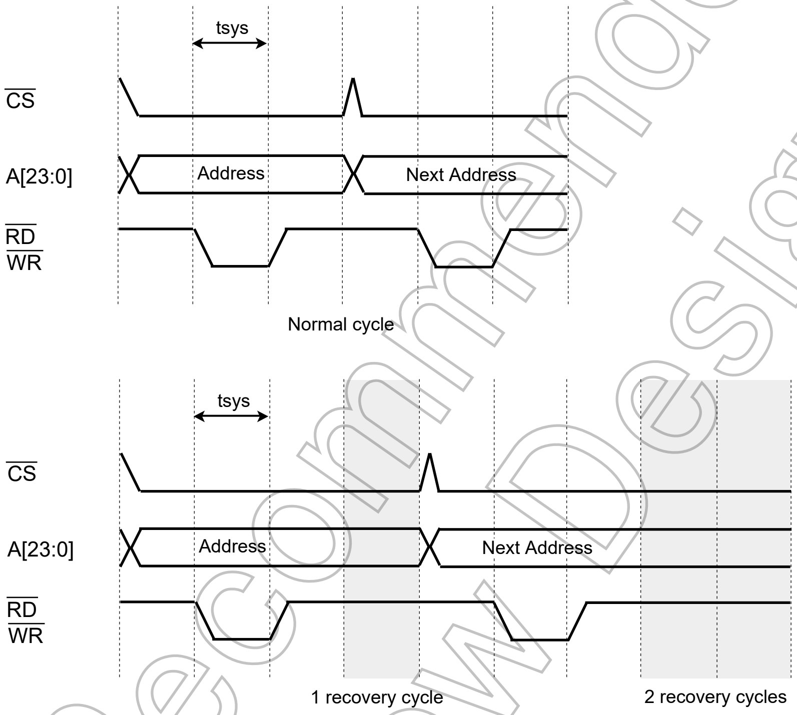


Figure 5-7 Timing of Recovery Time insertion

5.5.4 Chip select recovery time

If access to external areas occurs consecutively, a dummy cycle can be inserted for recovery time.

The dummy cycle insertion setting can be made in the chip select control registers, EXBCSx<CSR[1:0]>. As for the number of dummy cycles, none, one, two and four system clocks (internal) can be specified for each channel. Figure 5-8 shows the timing of recovery time insertion.

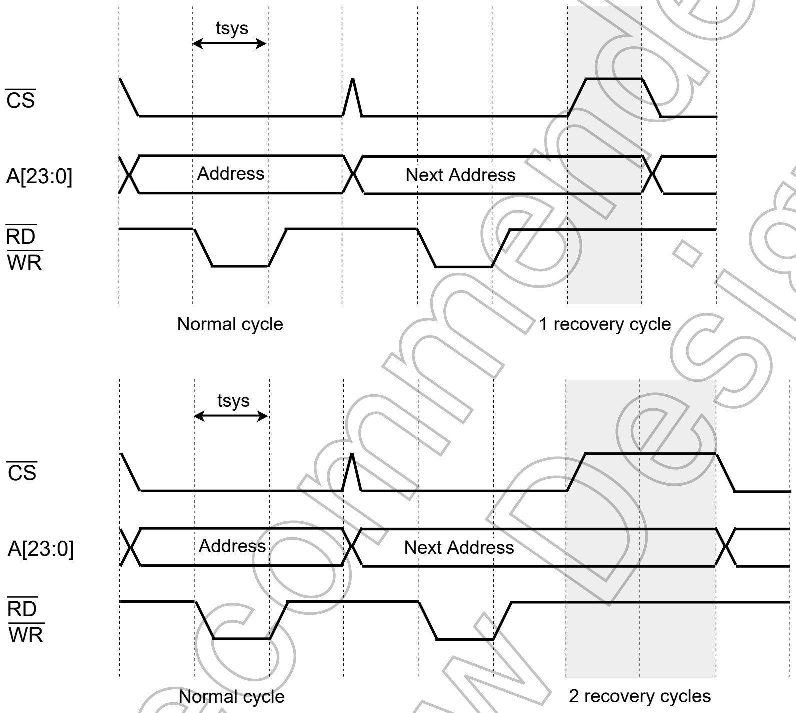


Figure 5-8 Timing of Chip Select Recovery Time Insertion

5.5.5 Read and Write setup cycle

A read and a write setup cycle can be inserted for each channel by using the chip selector (CS) and wait controller.

The following can be inserted.

- A read and a write setup cycle of up to 4 clocks can be automatically inserted.

The setting of the number of setup cycles to be automatically inserted and the setting can be made using the chip select control registers, EXBCSx<WRS[1:0]> and <RDS[1:0]>.

Figure 5-9 show the timing diagrams in which setup cycle have been inserted.

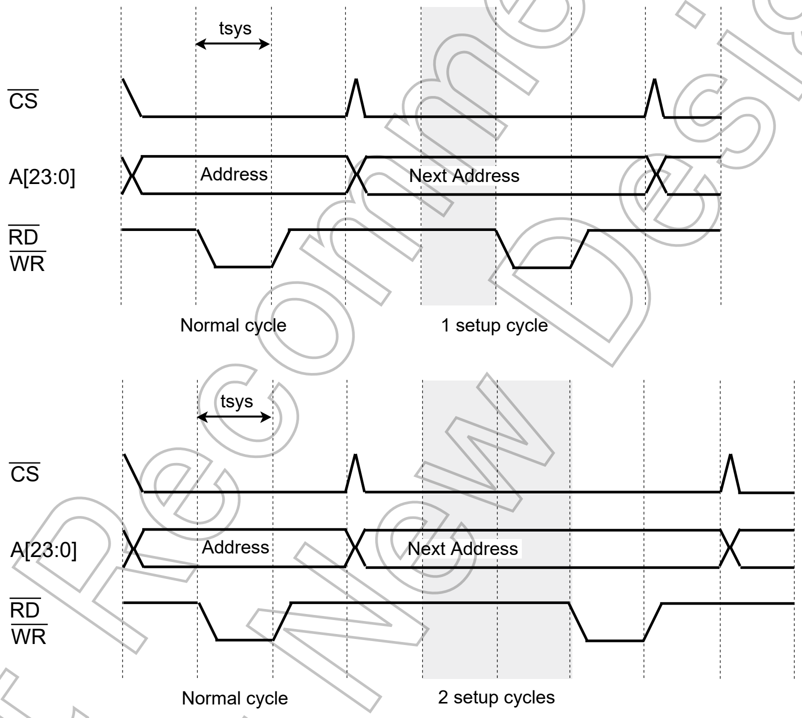


Figure 5-9 Timing of Read and Write Setup Time Insertion

5.6 External Bus Operations (Multiplexed Bus Mode)

This section describes various bus timing values. The timing diagram shown below assumes that the address buses are A23 through A16 and that the data buses are AD15 through AD0.

5.6.1 Basic bus operation

The external bus cycle of the TMPM341FDXBG/FYXBG basically consists of four clock pulses. The basic clock of an external bus cycle is the same as the internal system clock.Figure 5-10 shows read bus timing and Figure 5-11 shows write bus timing. If internal areas are accessed, address buses remain unchanged and the ALE does not output latch pulse as shown in these figures.

Additionally, address/data buses are in a state of high impedance and control signals such as $\overline{\text{RD}}$ and $\overline{\text{WR}}$ do not become active.

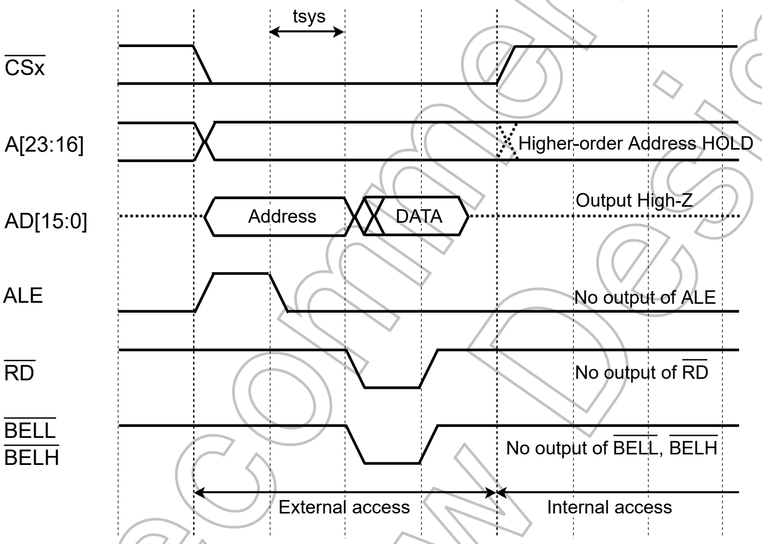


Figure 5-10 Read Operation Timing

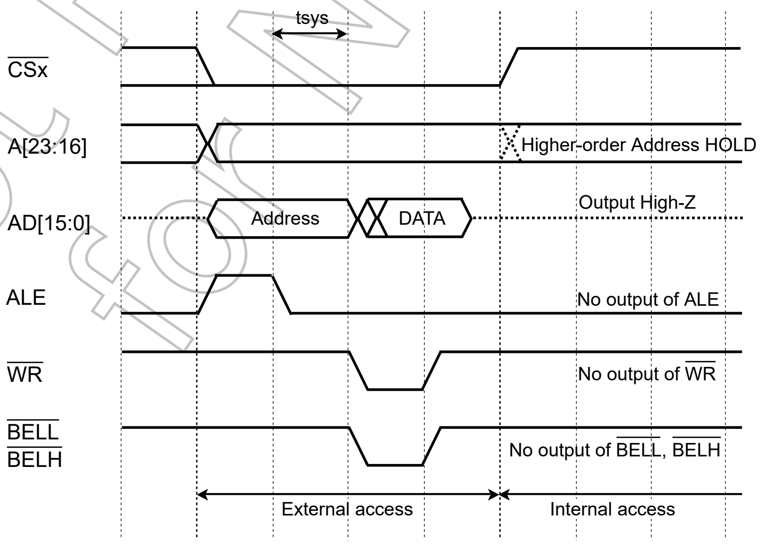


Figure 5-11 Write Operation Timing

5.6.2 Wait timing

A wait cycle can be inserted for each channel by using the chip selector (CS) and wait controller.

The following wait can be inserted.

- A wait of up to 15 clocks can be automatically inserted.

The setting of the number of waits to be automatically inserted and the setting can be made using the chip select control registers, EXBCSx<CSIW[4:0]>.

Figure 5-12 through Figure 5-13 show the timing diagrams in which waits have been inserted.

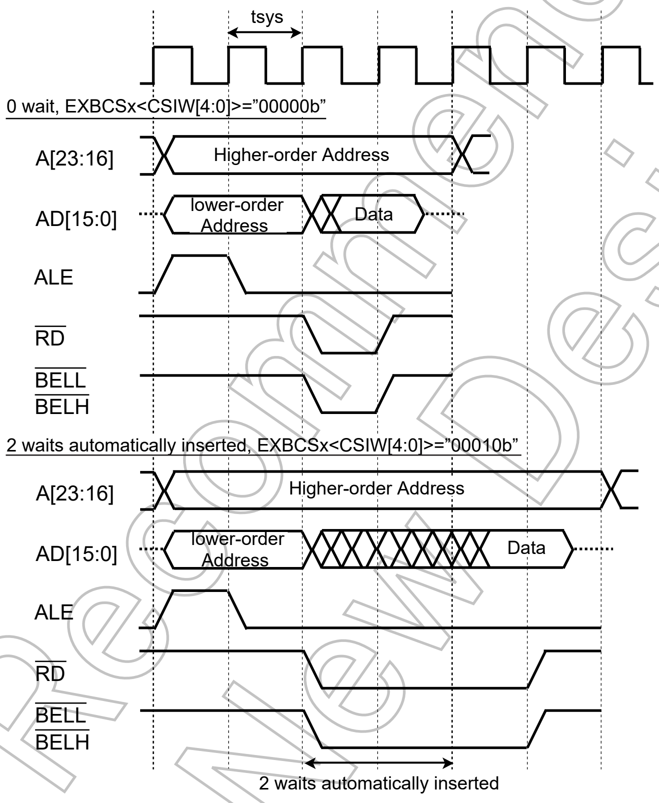


Figure 5-12 Read Operation Timing

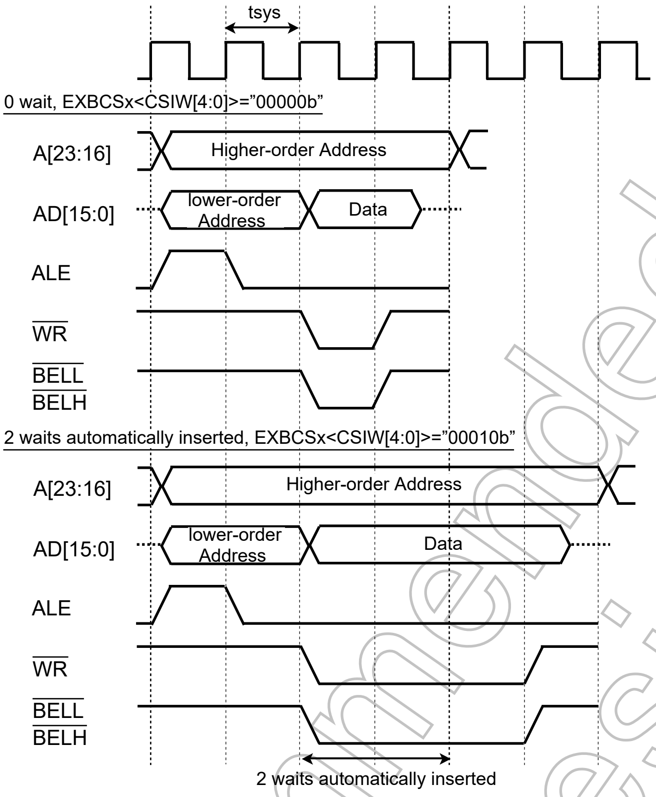


Figure 5-13 Write Operation Timing

5.6.3 Time that it takes before ALE is asserted

One of system clocks of 1, 2 or 4 can be selected as the time that it takes before ALE is asserted. The setting can be made using the chip select control registers, EXBCSx<ALEW[1:0]>. The default is asserted the $\overline{\text{RD}}$ or $\overline{\text{WR}}$ signal from the address is generated after 2 clocks.

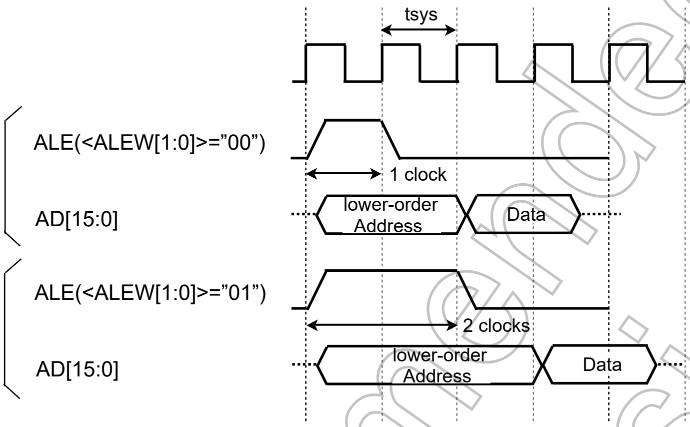


Figure 5-14 Time that it takes before ALE is asserted

Figure 5-15 shows the timing when the ALE is 1 clock or 2 clocks.

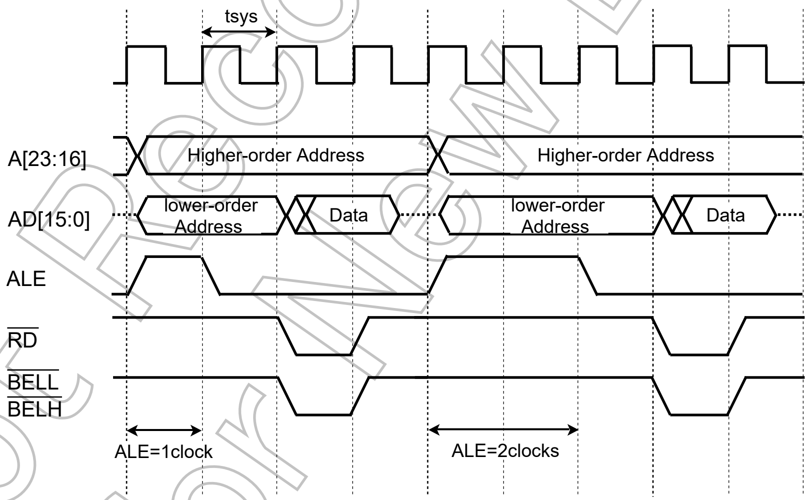


Figure 5-15 Read Operation Timing (When the ALE is 1 Clock or 2 Clocks)

5.6.4 Read and Write Recovery Time

If access to external areas occurs consecutively, a dummy cycle can be inserted for recovery time.

A dummy cycle can be inserted in both a read and a write cycle. The dummy cycle insertion setting can be made in the chip select control registers, EXBCSx<WRR[2:0]> (write recovery cycle) and <RDR[2:0]> (read recovery cycle). As for the number of dummy cycles, none, one to six or eight system clocks (internal) can be specified for each channel. Figure 5-16 shows the timing of recovery time insertion.

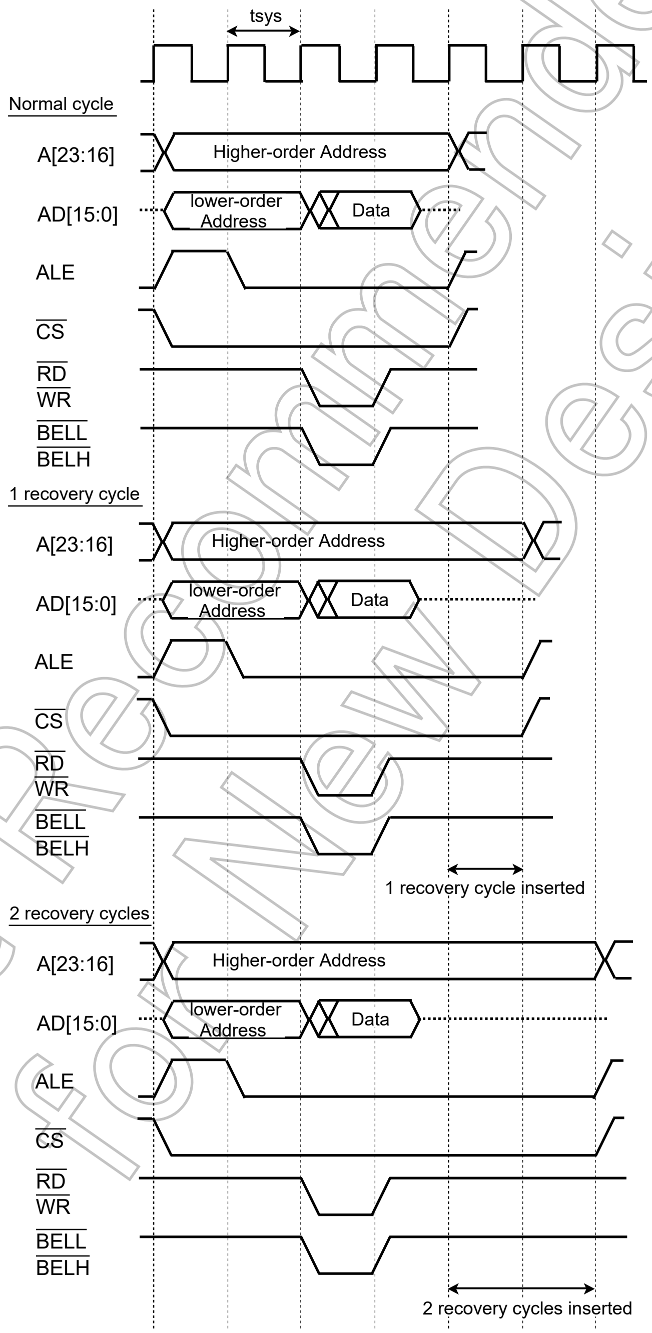


Figure 5-16 Timing of Recovery Time Insertion

5.6.5 Chip select recovery time

If access to external areas occurs consecutively, a dummy cycle can be inserted for recovery time.

The dummy cycle insertion setting can be made in the chip select control registers, EXBCSx<CSR[1:0]>. As for the number of dummy cycles, none, one, two and four system clocks (internal) can be specified for each channel. Figure 5-17 shows the timing of recovery time insertion.

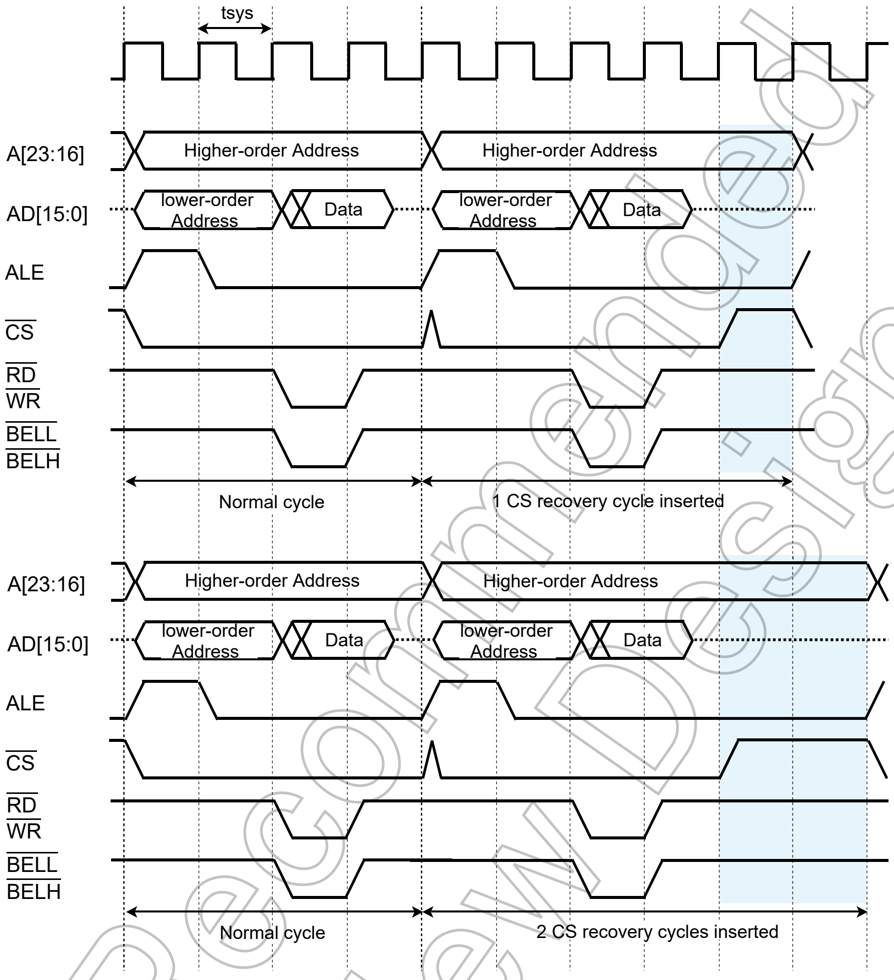


Figure 5-17 Timing of CS Recovery Time Insertion (ALE width: 1 clock)

5.6.6 Read and Write setup cycle

A read and a write setup cycle can be inserted for each channel by using the chip selector (CS) and wait controller.

The following can be inserted.

- A read and a write setup cycle of up to 4 clocks can be automatically inserted.

The setting of the number of setup cycles to be automatically inserted and the setting can be made using the chip select control registers, EXBCSx<WRS[1:0]> and <RDS[1:0]>.

Figure 5-18 shows the timing diagrams in which the read or write setup cycle have been inserted.

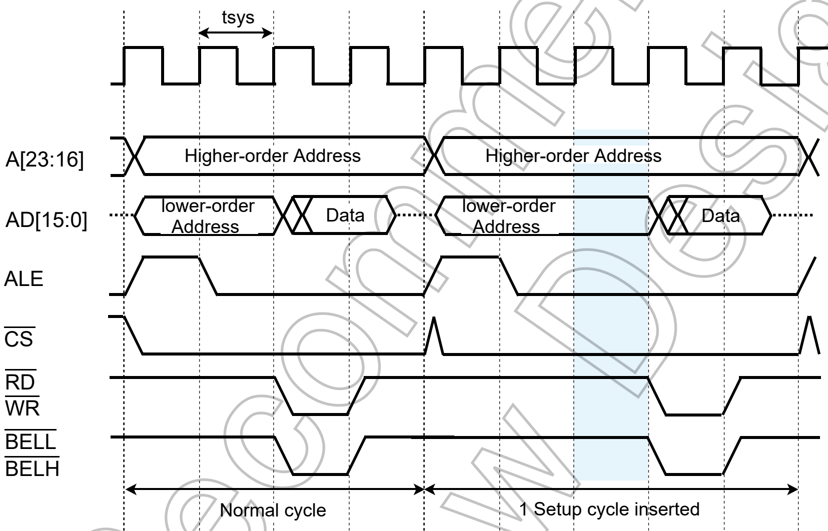


Figure 5-18 Timing of Read or Write Setup Time Insertion

5.7 Connection example for external memory

5.7.1 Connection Example for external 16-bit SRAM and NOR-Flash (Separate bus)

Below Figure 5-19 shows Separate bus connection example for external 16-bit NORF and 16-bit SRAM.

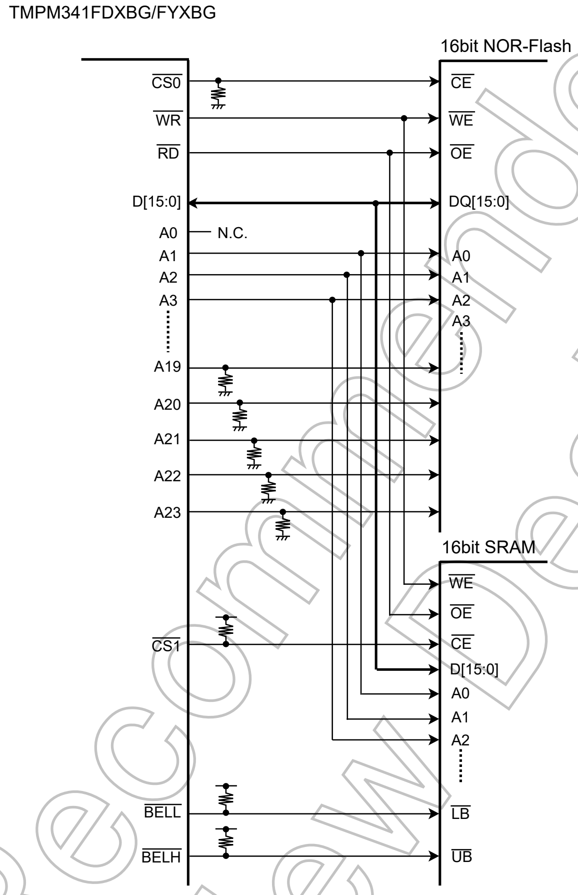


Figure 5-19 Connection Example for external 16-bit SRAM and NOR-Flash (Separate bus)

5.7.2 Connection Example for external 16-bit SRAM and NOR-Flash (Multiplex bus)

Below Figure 5-20 shows Multiplex bus connection example for external 16-bit NORF and 16-bit SRAM.

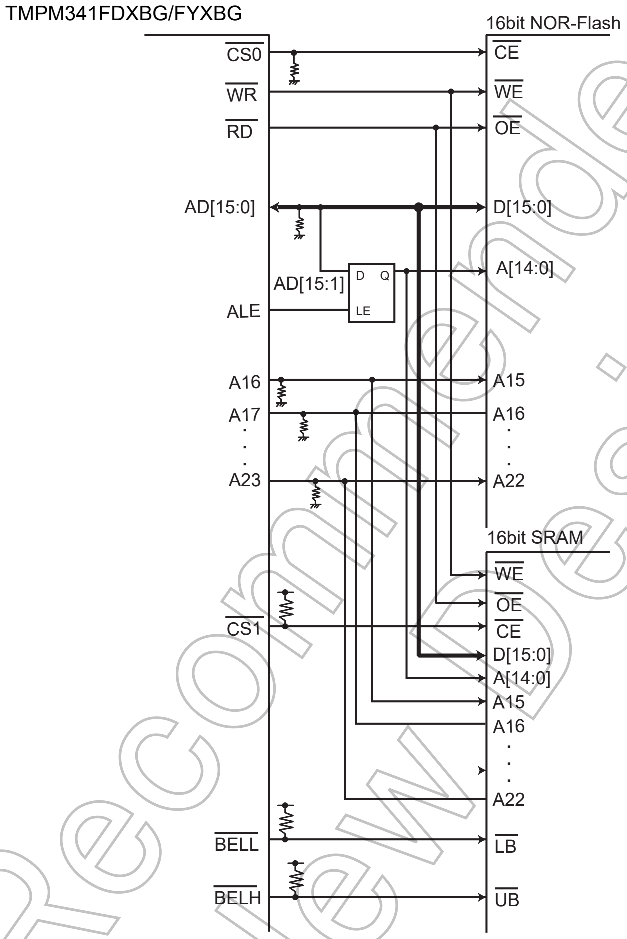


Figure 5-20 Connection Example for external 16-bit SRAM and NOR-Flash (Multiplex bus)

6. DMA Controller(DMAC)

6.1 Function Overview

The table below lists its major functions.

Table 6-1 DMA controller functions

Item	Function		Overview
Number of channels	4ch(2 unit)		DMACA × 2ch , DMACB × 2ch
Start trigger	Hardware start		Supports DMA requests for peripheral IPs.
	Software start		Started with a write to the DMACxSoftBReq register.
Bus master	32bit × 1 (AHB)		
Priority	(High) DMACA ch0 > DMACA ch1 > DMACB ch0 > DMACB ch1(low)		Fixed by hardware
FIFO	DMACA:4word × 2ch DMACB:4word × 2ch		
Bus width	8/16/32bit		Settable individually for transfer source and destination.
Burst size	1/4/8/16/32/64/128/256		
Number of transfers	up to 4095		
Address	Transfer source address	incr / no-incr	It is possible to specify whether Source and Destination addresses should increment or should not increment (should be fixed). (Address wrapping is not supported.)
	Transfer destination address	incr / no-incr	
Endian	Little endian / Big endian		
Transfer type	Peripheral circuit (register) → memory Memory → peripheral circuit (register) Memory → memory Peripheral circuit (register) → Peripheral circuit (register)		When "memory → memory" is selected, hardware start for DMA startup is not supported. See the DMACCxConfiguration register for more information.
Interrupt function	Transfer end interrupt Error interrupt		
Special Function	Scatter/gather function		

Note 1: 1 word = 32bit

6.2 DMA transfer type

Table 6-2 DMA transfer type

<table>
<tr><th></th><th>DMA direction</th><th>DMA request circuit</th><th>Support DMA request (Note3)</th><th>Other condition</th></tr>
<tr><td>1</td><td>Memory → peripheral circuit</td><td>peripheral circuit</td><td>Burst request</td><td>1) Uses all burst requests
2) In case of 1word transmission, set to the "1" for burst size of DMA controller.</td></tr>
<tr><td>2</td><td>Peripheral circuit → memory</td><td>peripheral circuit</td><td>Burst request / single request (Note1)</td><td>If the amount of data transfer is not an integral multiple of the burst size, both burst and single transfers are used.
・ Amount of remaining transfer data≥Burst size: Uses burst transfer.
・ Amount of remaining transfer data<Burst size: Uses single transfer.</td></tr>
<tr><td>3</td><td>Memory → memory</td><td>DMAC</td><td>None</td><td>Start condition : Enabling the DMAC starts data transfer with no DMAC request required.
Stop condition :
・ Transfer of all transfer data is complete.
・ The DMAC channel is disabled. (Note2)</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">Peripheral circuit → peripheral circuit</td><td>Source peripheral circuit</td><td>Burst request / single request (Note1)</td><td rowspan="2"><table><tr><th>Transfer size</th><th>Source</th><th>Destination</th></tr><tr><td>(1)an integral multiple of the burst size</td><td>Burst request</td><td rowspan="3">Burst request</td></tr><tr><td>(2)single transfer</td><td>Single request</td></tr><tr><td>(3)Not an integral multiple of the burst size</td><td>Burst request / single request</td></tr></table></td></tr>
<tr><td>Destination peripheral circuit</td><td>Burst request</td></tr>
</table>

Note 1: Single request circuit in the peripheral circuit: SSP

Note 2: Recommended: When transfers (big amounts of) data using memory → memory, Another AHB master can get bus rights even while transferring by using low priority channel (DMAC1). When using channel except DMAC1, wait until transfer is finished.

Note 3: Available DMA request: Refer to Table 6-3.

The DMA direction of "Peripheral circuit → Peripheral circuit" can be used as following conditions.

Source Addr.		Destination Addr.	Source Addr.		Destination Addr.	
IO	to	IO				
IO	to	SCxBUF	SCxBUF	to	IO	
IO	to	ADREGx	ADREGx	to	IO	(x=00 to 15)
IO	to	ADREGSP	ADREGSP	to	IO	
IO	to	ADCMP0/1	ADCMP0/1	to	IO	
IO	to	TBxREG0/1	TBxREG0/1	to	IO	(x=0 to 9)
IO	to	TBxCP0/1	TBxCP0/1	to	IO	(x=0 to 9)
IO	to	TDxRG0/1/2/3/4	TDxRG0/1/2/3/4	to	IO	(x=0,1)
IO	to	TD0RG5	TD0RG5	to	IO	

6.3 Block diagram

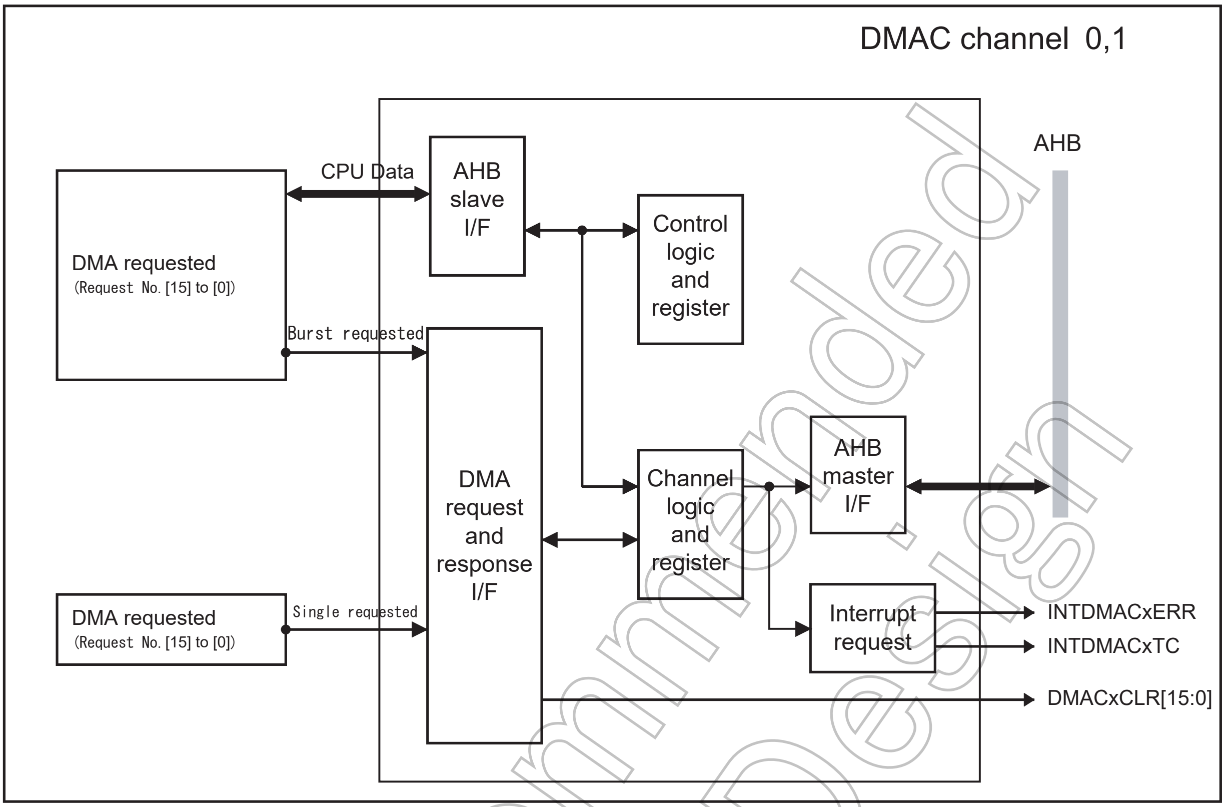


Figure 6-1 DMAC Block diagram

Table 6-3 DMA request number chart

DMA request No.	Corresponding peripheral			
	UNIT A (DMACA ch0,ch1)		UNIT B (DMACB ch0,ch1)	
	Burst	Single	Burst	Single
0	SIO0/UART0 Reception	−	TMRD00 compare match	−
1	SIO0/UART0 Transmission	−	TMRD10 compare match	−
2	SIO2/UART2 Reception	−	Two-phase pulse counter0 every count	−
3	SIO2/UART2 Transmission	−	Two-phase pulse counter1 every count	−
4	SIO4/UART4 Reception	−	TMRB6 compare match	−
5	SIO4/UART4 Transmission	−	TMRB7 compare match	−
6	Two-phase pulse counter2 every count	−	TMRB0 input capture 1	−
7	Two-phase pulse counter3 every count	−	TMRB2 input capture 0	−
8	TMRB8 compare match	−	TMRB2 input capture 1	−
9	TMRB9 compare match	−	TMRB3 input capture 0	−
10	TMRB0 input capture 0	−	TMRB3 input capture 1	−
11	TMRB4 input capture 0	−	TMRB6 input capture 0	−
12	TMRB4 input capture 1	−	TMRB6 input capture 1	−
13	TMRB5 input capture 0	−	Normal AD Conversion End	−
14	TMRB5 input capture 1	−	SSP Transmission	SSP Transmission
15	Highest priority AD Conversion End	−	SSP Reception	SSP Reception

6.4 Description of Registers

6.4.1 DMAC register list

The following lists the each unit and address:

Unit x	Base Address
UnitA	0x4000_0000
UnitB	0x4000_1000

Register Name		Address (Base+)
DMAC Interrupt Status Register	DMACxIntStaus	0x0000
DMAC Interrupt Terminal Count Status Register	DMACxIntTCStatus	0x0004
DMAC Interrupt Terminal Count Clear Register	DMACxIntTCClear	0x0008
DMAC Interrupt Error Status Register	DMACxIntErrorStatus	0x000C
DMAC Interrupt Error Clear Register	DMACxIntErrClr	0x0010
DMAC Raw Interrupt Terminal Count Status Register	DMACxRawIntTCStatus	0x0014
DMAC Raw Error Interrupt Status Register	DMACxRawIntErrorStatus	0x0018
DMAC Enabled Channel Register	DMACxEnbldChns	0x001C
DMAC Software Burst Request Register	DMACxSoftBReq	0x0020
DMAC Software Single Request Register	DMACxSoftSReq	0x0024
Reserved	-	0x0028
Reserved	-	0x002C
DMAC Configuration Register	DMACxConfiguration	0x0030
Reserved	-	0x0034
DMAC Channel0 Source Address Register	DMACxC0SrcAddr	0x0100
DMAC Channel0 Destination Address Register	DMACxC0DestAddr	0x0104
DMAC Channel0 Linked List Item Register	DMACxC0LLI	0x0108
DMAC Channel0 Control Register	DMACxC0Control	0x010C
DMAC Channel0 Configuration Register	DMACxC0Configuration	0x0110
DMAC Channel1 Source Address Register	DMACxC1SrcAddr	0x0120
DMAC Channel1 Destination Address Register	DMACxC1DestAddr	0x0124
DMAC Channel1 Linked List Item Register	DMACxC1LLI	0x0128
DMAC Channel1 Control Register	DMACxC1Control	0x012C
DMAC Channel 1 Configuration Register	DMACxC1Configuration	0x0130

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

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

Note 3: For the registers prepared for every channel, if the channel structure is the same, unit and channel number are expressed as "x" and "n" in detail description of registers.

Note 4: When the register which is not assigned with an each channel is read after the register which is assigned with an each channel is written, one machine cycle is inserted between the instructions or read the register which is not assigned with an each channel twice.

6.4.2 DMACxIntStatus (DMAC Interrupt Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	IntStatus1	IntStatus0
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Description
31-2	-	W	Write as zero.
1	IntStatus1	R	Status of DMAC channel 1 transfer end interrupt. 0 : Interrupt not requested 1 : Interrupt requested Status of the DMAC interrupt generation after passing through the transfer end interrupt enable register and error interrupt enable register. An interrupt is requested when there is a transfer error or when the counter completes counting.
0	IntStatus0	R	Status of DMAC channel 0 interrupt generation. 0 : Interrupt not requested 1 : Interrupt requested Status of the DMAC interrupt generation after passing through the transfer end interrupt enable register and error interrupt enable register. An interrupt is requested when there is a transfer error or when the counter completes counting.

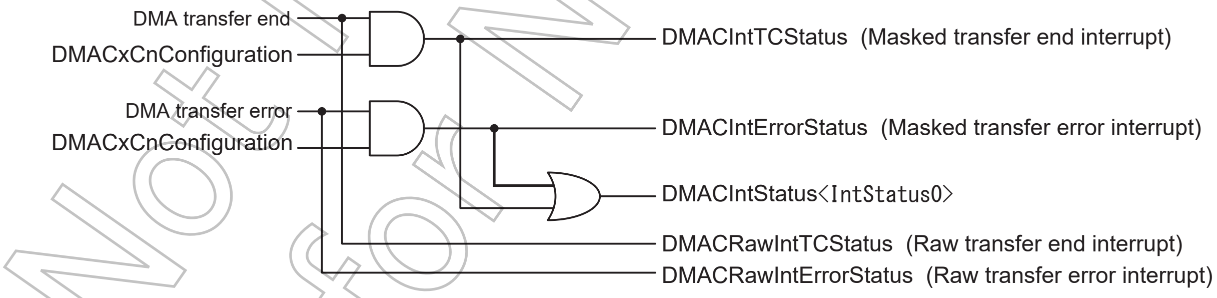

DMA transfer end
DMACxCnConfiguration
DMA transfer error
DMACxCnConfiguration
DMACIntTCStatus (Masked transfer end interrupt)
DMACIntErrorStatus (Masked transfer error interrupt)
DMACIntStatus<IntStatus0>
DMACRawIntTCStatus (Raw transfer end interrupt)
DMACRawIntErrorStatus (Raw transfer error interrupt)

Figure 6-2 Interrupt-related block diagram

6.4.3 DMACxIntTCStatus (DMAC Interrupt Terminal Count Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	IntTCStatus1	IntTCStatus0
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Description
31-2	-	W	Write as zero.
1	IntTCStatus1	R	Status of DMAC channel 1 transfer end interrupt. 0 : Interrupt not requested 1 : Interrupt requested The status of post-enable transfer end interrupt generation.
0	IntTCStatus0	R	Status of DMAC channel 0 transfer end interrupt. 0 : Interrupt not requested 1 : Interrupt requested The status of post-enable transfer end interrupt generation.

6.4.4 DMACxIntTCClear (DMAC Interrupt Terminal Count Clear Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	IntTCClear1	IntTCClear0
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Description
31-2	-	W	Write as zero.
1	IntTCClear1	W	Clear DMAC channel 1 transfer end interrupt. 0 : Do nothing 1 : Clear The DMACxIntTCStatus<IntTCStatus1> will be cleared when "1" is written.
0	IntTCClear0	W	Clear DMAC channel 0 transfer end interrupt. 0 : Do nothing 1 : Clear The DMACxIntTCStatus<IntTCStatus0> will be cleared when "1" is written.

6.4.5 DMACxIntErrorStatus (DMAC Interrupt Error Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	IntErrStatus1	IIntErrStatus0
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Description
31-2	-	W	Write as zero.
1	IntErrStatus1	R	Status of DMAC channel 1 error interrupt generation. 0 : Interrupt not requested 1 : Interrupt requested Shows error interrupt status after enabled.
0	IntErrStatus0	R	Status of DMAC channel 0 error interrupt generation. 0 : Interrupt not requested 1 : Interrupt requested Shows error interrupt status after enabled.

6.4.6 DMACxIntErrClr (DMAC Interrupt Error Clear Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	IntErrClr1	IntErrClr0
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Description
31-2	-	W	Write as zero.
1	IntErrClr1	W	Clear DMAC channel 1 transfer end interrupt. 0 : Do nothing 1 : Clear The DMACxIntErrorStatus<IntErrStatus1> will be cleared when "1" is written.
0	IntErrClr0	W	Clear DMAC channel 0 transfer end interrupt. 0 : Do nothing 1 : Clear The DMACxIntErrorStatus<IntErrStatus0> will be cleared when "1" is written.

6.4.7 DMACxRawIntTCStatus (DMAC Raw Interrupt Terminal Count Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	RawIntTCS1	RawIntTCS0
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Description
31-2	-	W	Read undefined. Write as zero.
1	RawIntTCS1	R	Status of DMAC channel 1 pre-enable transfer end interrupt generation 0 : Interrupt not requested 1 : Interrupt requested
0	RawIntTCS0	R	Status of DMAC channel 0 pre-enable transfer end interrupt generation 0 : Interrupt not requested 1 : Interrupt requested

6.4.8 DMACxRawIntErrorStatus (DMAC Raw Error Interrupt Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	RawIntErrS1	RawIntErrS0
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Description
31-2	-	W	Read undefined. Write as zero.
1	RawIntErrS1	R	Status of DMAC channel 1 pre-enable error interrupt. 0 : Interrupt not requested 1 : Interrupt requested
0	RawIntErrS0	R	Status of DMAC channel 0 pre-enable error interrupt. 0 : Interrupt not requested 1 : Interrupt requested

6.4.9 DMACxEnbldChns (DMAC Enabled Channel 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	-	-	-	-	-	-	EnabledCH1	EnabledCH0
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Description
31-2	-	W	Read undefined. Write as zero.
1	EnabledCH1	R	DMA channel 1 enable status. 0 : Disable 1 : Enable
0	EnabledCH0	R	DMA channel 0 enable status. 0 : Disable 1 : Enable

6.4.10 DMACxSoftBReq (DMAC Software Burst Request 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	SoftBReq15	SoftBReq14	SoftBReq13	SoftBReq12	SoftBReq11	SoftBReq10	SoftBReq9	SoftBReq8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SoftBReq7	SoftBReq6	SoftBReq5	SoftBReq4	SoftBReq3	SoftBReq2	SoftBReq1	SoftBReq0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-16	-	W	Write as zero.
15	SoftBReq15	R/W	DMA burst request by software (Request No. [15]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
14	SoftBReq14	R/W	DMA burst request by software (Request No. [14]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
13	SoftBReq13	R/W	DMA burst request by software (Request No. [13]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
12	SoftBReq12	R/W	DMA burst request by software (Request No. [12]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
11	SoftBReq11	R/W	DMA burst request by software (Request No. [11]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
10	SoftBReq10	R/W	DMA burst request by software (Request No. [10]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
9	SoftBReq9	R/W	DMA burst request by software (Request No. [9]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
8	SoftBReq8	R/W	DMA burst request by software (Request No. [8])

Bit	Bit Symbol	Type	Description
			Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
7	SoftBReq7	R/W	DMA burst request by software (Request No. [7]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
6	SoftBReq6	R/W	DMA burst request by software (Request No. [6]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
5	SoftBReq5	R/W	DMA burst request by software (Request No. [5]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
4	SoftBReq4	R/W	DMA burst request by software (Request No. [4]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
3	SoftBReq3	R/W	DMA burst request by software (Request No. [3]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
2	SoftBReq2	R/W	DMA burst request by software (Request No. [2]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
1	SoftBReq1	R/W	DMA burst request by software (Request No. [1]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request
0	SoftBReq0	R/W	DMA burst request by software (Request No. [0]) Read: 0 : DMA Burst under suspension 1 : DMA Burst running Write: 0 : Invalid 1 : DMA Burst request

Sets a DMA burst transfer request by software. When the DMA burst transfer by software is complete, the appropriate bits in SoftBReq are cleared.

Note 1: Do not execute DMA requests by software and hardware peripheral at the same time.

Note 2: Refer to "Table 6-3 DMA request number chart" for DMA request number.

6.4.11 DMACxSoftSReq (DMAC Software Single Request Register)

Case of x = A

	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	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Description
31-0	-	W	Write as zero.

Case of x = B

	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	SoftSReq13	SoftSReq13	-	-	-	-	-	-
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Description
31-16	-	W	Write as zero.
15	SoftSReq15	R/W	DMA single request by software (Request No. [15]). Read: 0 : DMA Single under suspension 1 : DMA Single running Write: 0 : Invalid 1 : DMA Single request
14	SoftSReq14	R/W	DMA single request by software (Request No. [14]). Read: 0 : DMA Single under suspension 1 : DMA Single running Write: 0 : Invalid 1 : DMA Single request
13-0	-	W	Write as zero.

Sets a DMA single transfer request by software. When the DMA single transfer by software is complete, the appropriate bits in SoftSReq are cleared.

Note 1: Do not execute a DMA request by software when a DMA request by hardware peripheral is generated.

Note 2: Refer to "Table 6-3 DMA request number chart" for DMA request number.

6.4.12 DMACxConfiguration (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	-	-	-	-	-	-	M	E
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Description
31-2	-	-	Read undefined. Write as zero.
1	M	R/W	DMA endian configuration: 0 : Little endian 1 : Big endian
0	E	R/W	DMA circuit control: 0 : Stop 1 : Operate When circuit stops, the registers for the DMA circuit cannot be written or read. When operating the DMA, always set <E>="1".

6.4.13 DMACxCnSrcAddr (DMAC Channeln Source Address Register)

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Description</th></tr>
<tr><td>31-0</td><td>SrcAddr[31:0]</td><td>R/W</td><td>Sets a DMA transfer source address.
Make sure to confirm the source address and the bit width before setting.
The below are the restrictions in setting of source address bit width.
<table>
<tr><th>Source address bit width
DMACxCnControl<Swidth[2:0]&gt;</th><th>Setting of least significant address</th></tr>
<tr><td>000 : Byte (8 bits)</td><td>no restriction</td></tr>
<tr><td>001 : Half word (16 bits)</td><td>Setting as multiples of 2, (0x0,0x02,0x4,0x06,0x8,0xA,0xC…)</td></tr>
<tr><td>010 : Word (32 bits)</td><td>Setting as multiples of 4, (0x0,0x4,0x8,0xC…)</td></tr>
</table>
</td></tr>
</table>

Because enabling channel"n" (DMACxCnConfiguration<E>="1") updates the data written in the registers, set DMACxCnSrcAddr before enabling the channels.

When the DMA is operating, the value in the DMACxCnSrcAddr register sequentially changes, so the read values are not fixed.

And do not update DMACxCnSrcAddr during transfer. To change DMACxCnSrcAddr, be sure to disable the channel "n" (DMACxCnConfiguration<E>="0") before change.

6.4.14 DMACxCnDestAddr (DMAC Channeln Destination Address Register)

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Description</th></tr>
<tr><td>31-0</td><td>DestAddr[31:0]</td><td>R/W</td><td>Sets a DMA transfer destination address.
Make sure to confirm the destination address and the bit width before setting.
The below are the restrictions in setting of destination address bit width.
<table>
<tr><th>Destination address bit width
DMACxCControl<Dwidth[2:0]&gt;</th><th>Setting of least significant address</th></tr>
<tr><td>000 : Byte (8 bits)</td><td>no restriction</td></tr>
<tr><td>001 : Half word (16 bits)</td><td>Setting as multiples of 2, (0x0,0x02,0x4,0x06,0x8,0xA,0xC…)</td></tr>
<tr><td>010 : Word (32 bits)</td><td>Setting as multiples of 4, (0x0,0x4,0x8,0xC…)</td></tr>
</table>
</td></tr>
</table>

Do not update DMACxCnDestAddr during transfer. To change DMACxCnDestAddr, be sure to disable the channel "n" (DMACxCnConfiguration<E>="0") before change.

6.4.15 DMACxCnLLI (DMAC Channelx Linked List Item Register)

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

Bit	Bit Symbol	Type	Description
31-2	LLI[29:0]	R/W	Sets the first address of the next transfer information. Set a value smaller than 0xFFFF_FFF0. When <LLI> = 0, LLI is the last chain. After DMA transfer finishes, the DMA channel is disabled.
1-0	-	R/W	Read undefined. Write as zero.

For <LLI> detailed operation, see "6.5 Special Functions".

6.4.16 DMACxCnControl (DMAC Channeln Control Register)

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

Bit	Bit Symbol	Type	Description
31	I	R/W	Register for enabling a transfer interrupt. 0 : Disable 1 : Enable The transfer end interrupt is generated by setting <I>="1" and DMACxCnConfiguration<ITC>="1". When the scatter/gather function is used in the last transfer DMAC setting flow and by setting this bit to enable, to generate the transfer end interrupt is enable only at the last transfer. To generate interrupt during normal transfer, set this bit to "1" and change to enable mode.
30-28	-	W	Write as zero.
27	DI	R/W	Increment the transfer destination address 0 : Do not increment 1 : Increment
26	SI	R/W	Increment the transfer source address 0 : Do not increment 1 : Increment
25-24	-	W	Write as zero.
23-21	Dwidth[2:0]	R/W	Transfer destination bit width. 000 : Byte (8 bits) 001 : Half-word (16 bits) 010 : Word (32 bits) other: Reserved
20-18	Swidth[2:0]	R/W	Transfer source bit width 000 : Byte (8 bits) 001 : Half-word (16 bits) 010 : Word (32 bits) other: Reserved Set a transfer destination bit width so that the transfer size becomes an integral multiple of the transfer destination bit width.
17-15	DBSize[2:0]	R/W	Transfer destination burst size: (Note 1) 000: 1 beat 100: 32 beats 001: 4 beats 101: 64 beats 010: 8 beats 110: 128 beats 011: 16 beats 111: 256 beats
14-12	SBSize[2:0]	R/W	Transfer source burst size: (Note 1) 000: 1 beat 100: 32 beats 001: 4 beats 101: 64 beats 010: 8 beats 110: 128 beats 011: 16 beats 111: 256 beats

Bit	Bit Symbol	Type	Description
11-0	TransferSize [11:0]	R/W	Set the total number of transfers. Set the total number of transfers. This value decrements to "0" as DMAC transfer is executed. The read operation reads the number of transfers that have been executed or transfer size is read as "0" when transfer have not been executed. The total number of transfers is used as the unit for the transfer source bit width. For examples: When <Swidth>="000" (8bit), the number of transfers is expressed in the units of byte. When <Swidth>="001" (16bit), the number of transfers is expressed in the units of half word. When <Swidth>="010" (32bit), the number of transfers is expressed in the units of word.

<Dwidth[2:0]> / <Swidth[2:0]>	Set the number so that the following expression is satisfied: Transfer source bit width × Total number of transfers = Transfer destination bit width × N (N : Integer number) (ex.1) Bit width of transfer source:8 bit, bit width of transfer destination:32 bit, total number of transfers:25 times 8 bit × 25 times = 200 bit (25 byte) N = 200 ÷ 32 = 6.25 word Since 6.25 is not an integer number, the above setting is invalid. If the transfer source bit width is smaller than the transfer destination bit width, care must be taken when setting the total number of transfers. (ex.2) Bit width of transfer source :32 bit, bit width of transfer destination:16 bit, total number of transfers: 13 times 32 bit × 13 times = 416 bit (13 word) N = 416 ÷ 16 = 26 half_word Since 26 is an integer number, the above setting is valid.
<DBSize[2:0]> / <SBSize[2:0]>	When peripheral to memory transfer or memory to peripheral transfer is performed, peripheral circuits generates DMA request signal to indicate the preparation is ready. This signal triggers to execute data transfers. (In the case of memory to memory transfers, only software start is used.) Set the burst size to define the amount of data transferred from peripherals per DMA request signal. This register is used with FIFO buffer that can be contained multiple data.

Note 1: The burst size to be set with DBsize and SBsize has nothing to do with the HBURST for the AHB bus.

6.4.17 DMACxCnConfiguration (DMAC Channelx 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	-	-	-	-	-	Halt	Active	Lock
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ITC	IE	FlowCntrl			-	DestPeripheral	
After reset	0	0	0	0	0	Undefined	0	0
	7	6	5	4	3	2	1	0
bit symbol	DestPeripheral		-	SrcPeripheral				E
After reset	0	0	Undefined	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-19	-	W	Write as zero.
18	Halt	R/W	Controls accepting a DMA request 0 : Accept a DMA request 1 : Ignore a DMA request
17	Active	R	Indicates whether data is present in the channel FIFO. 0 : No data exists in the FIFO 1 : Data exists in the FIFO
16	Lock	R/W	Sets a locked transfer (Non-divided transfer). 0 : Disable locked transfer 1: Enable locked transfer When locked transfer is enabled, as many burst transfers as specified are consecutively executed without releasing the bus. For detailed operation, see "6.5 Special Functions".
15	ITC	R/W	Transfer end interrupt enable register. 0 : Disable interrupt 1 : Enable interrupt
14	IE	R/W	Error interrupt enable register 0 :Disable interrupt 1: Enable interrupt
13-11	FlowCntrl[2:0]	R/W	Sets transfer method (Note 1) <FlowCntrl[2:0]> setting value — Transfer method 000: — Memory to Memory 001: — Memory to Peripheral 010: — Peripheral to Memory 011: — Peripheral to Peripheral 100~111: Reserved
10	-	W	Write as zero.
9-6	DestPeripheral [3:0]	R/W	Sets transfer destination peripheral (Note 2) 000 to 111 The DMA request peripheral number is expressed by binary. When a memory is the transfer destination, this setting is ignored.
5	-	W	Write as zero.
4-1	SrcPeripheral [3:0]	R/W	Sets transfer source peripheral (Note 2) 000 to 111 The DMA request peripheral number is expressed by binary. When a memory is the transfer source, this setting is ignored.

Bit	Bit Symbol	Type	Description
0	E	R/W	Channel enable 0 : Disable 1 : Enable This bit can be used to enable/disable the channels. Disabling channels during transfer loses the data in the FIFO. Initialize all the channels before restart. To pause the transfer, stop the DMA request by using the <HALT> bit, and poll the data until the <Active> bit becomes "0" and then disable the channel with the <E> bit.

Note 1: When "memory to memory" is selected, hardware start for DMA startup is not supported. Writing to <E>= 1 starts transfer.

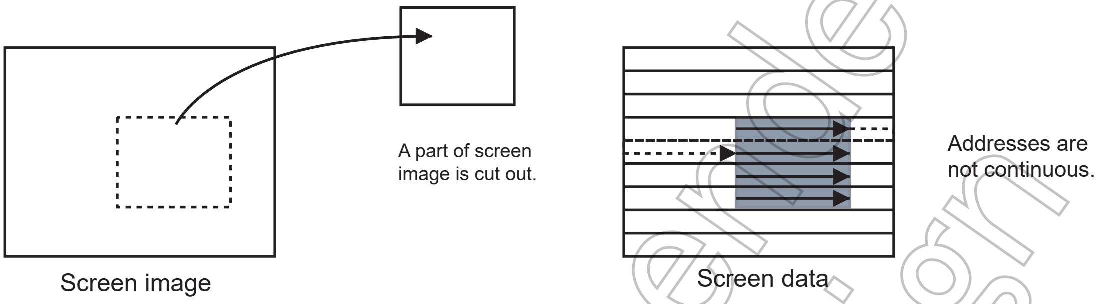
Note 2: Refer to Table 6-3 for DMA request peripheral number.

Note 3: When DMACxENableChns<EnabledCHx> is enabled and the corresponding DMACxCnConfiguration<Halt> is set to "1", write them after channel enable bit (E:bit0) is clear to "0". Without this, in the case of the slave error is occurred when writing them, the error is recovered by reset. Regarding slave error, when the width and address of transfer have mismatch, this error is occurred.

6.5 Special Functions

6.5.1 Scatter/gather function

When removing a part of image data and transferring it, image data cannot be handled as consecutive data, and the address changes dramatically depending on the special rule. Since DMA can transfer data only by using consecutive addresses, it is necessary to make required settings at locations where addresses changes.


A part of screen image is cut out.
Addresses are not continuous.
Screen image
Screen data

The scatter/gather function can consecutively operate DMA settings (transfer source address, destination address, number of transfers, and transfer bus width) by re-loading them each time a specified number of DMA executions have completed via a pre-set "Linked List" where the CPU does not need to control the operation.

Setting "1" in the DMACxCnLLI register enables/disables the operation.

The items that can be set with Linked List are configured with the following 4 words:

1. DMACxCnSrcAddr

2. DMACxCnDestAddr

3. DMACxCnLLI

4. DMACxCnControl

They can be used with the interrupt operation.

An interrupt depends on the count end interrupt enable bit of the DMACxCnControl register, and can be generated at the end of each LLI. When this bit is used, a condition can be added even during transfer using LLI to perform branch operation, etc. To clear the interrupt, control the appropriate bit of the DMACxIntTCClear register.

6.5.2 Linked list operation

To operate the scatter/gather function, a transfer source and source data areas need to be defined by creating a set of Linked Lists first.

Each setting is called LLI (LinkedList).

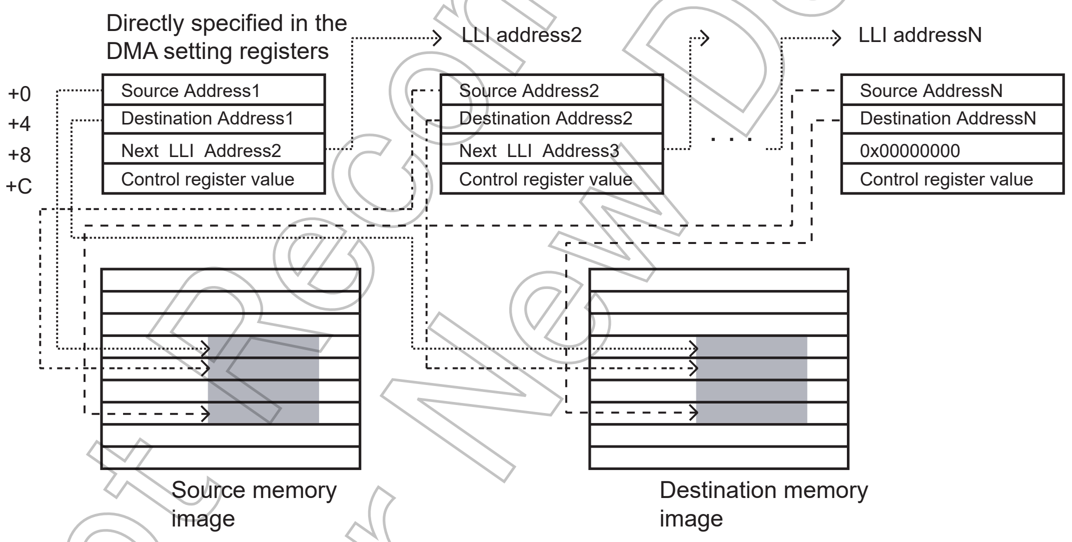
Each LLI controls the transfer of one block of data. Each LLI indicates normal DMA setting and controls transfer of successive data. Each time each DMA transfer is complete, the next LLI setting will be loaded to continue the DMA operation (Daisy Chain).

An example of the setting is shown below.

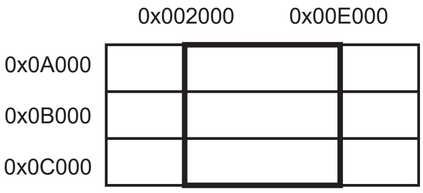
1. The first DMA transfer setting should be made directly in the DMA register.

2. The second and subsequent DMA transfer settings should be written in the addresses of the memory set in "next LLI AddressX."

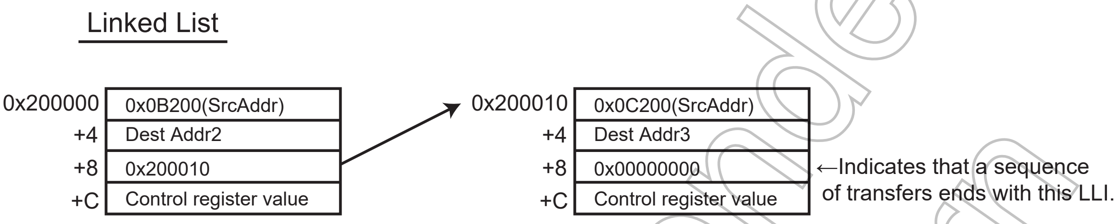
3. To stop up to N'th DMA transfer, set "next LLI AddressX" to 0x00000000.



When transferring data in the area enclosed by the square



	Setting register	Setting parameter
+0	DMACxCnSrcAddr	:0x0A200
+4	DMACxCnDestAddr	:Destination address 1
+8	DMACxCnLL	:0x200000
+C	DMACxCnControl	:Set the number of burst transfers and the number of transfers, etc.



Linked List

	0x200000
	0x0B200(SrcAddr)
+4	Dest Addr2
+8	0x200010
+C	Control register value

	0x200010
	0x0C200(SrcAddr)
+4	Dest Addr3
+8	0x00000000
+C	Control register value

←Indicates that a sequence of transfers ends with this LLI.

7. Reset

The TMPM341FDXBG/FYXBG has four reset sources: an external reset pin ($\overline{\text{RESET}}$), a watchdog timer (WDT), an oscillation frequency detection (OFD) and the setting <SYSRESETREQ> in the Application Interrupt and Reset Control Register.

For reset from the WDT, refer to the chapter on the WDT.

For reset from the OFD, refer to the chapter on the OFD.

For reset from <SYSRESETREQ>, refer to "Cortex-M3 Technical Reference Manual".

7.1 Initial state

7.1.1 State before input reset

The internal circuits, register settings and pin status of the TMPM341FDXBG/FYXBG are undefined right after the power-on. The state continues until the $\overline{\text{RESET}}$ pin receives low level input after all the power supply voltage (DVDD3A, RVDD3, AVDD3 and DVDD3B) is applied.

7.2 Cold reset

The power-on sequence must include the time for the internal regulator to be stable and the reset time. In the TMPM341FDXBG/FYXBG, the internal regulator requires at least 1ms to be stable.

At cold reset, the external reset pin must be kept "Low" for a duration of time sufficiently long enough for the internal regulator to be stable. If you use one of four power sources (DVDD3A, DVDD3B, RVDD3, AVDD3) with a different voltage, or you use only DVDD3B with a different voltage (1.65V), set the power supply to satisfy the following condition: DVDD3A=RVDD3=AVDD3 rising time (time to guaranteed operating voltage) ≤ DVDD3B rising time (time to guaranteed operating voltage).

After the external reset ($\overline{\text{RESET}}$) signal is released, the internal reset signal remains asserted for a further 400 µs.

Figure 7-1 shows the power-on sequence.

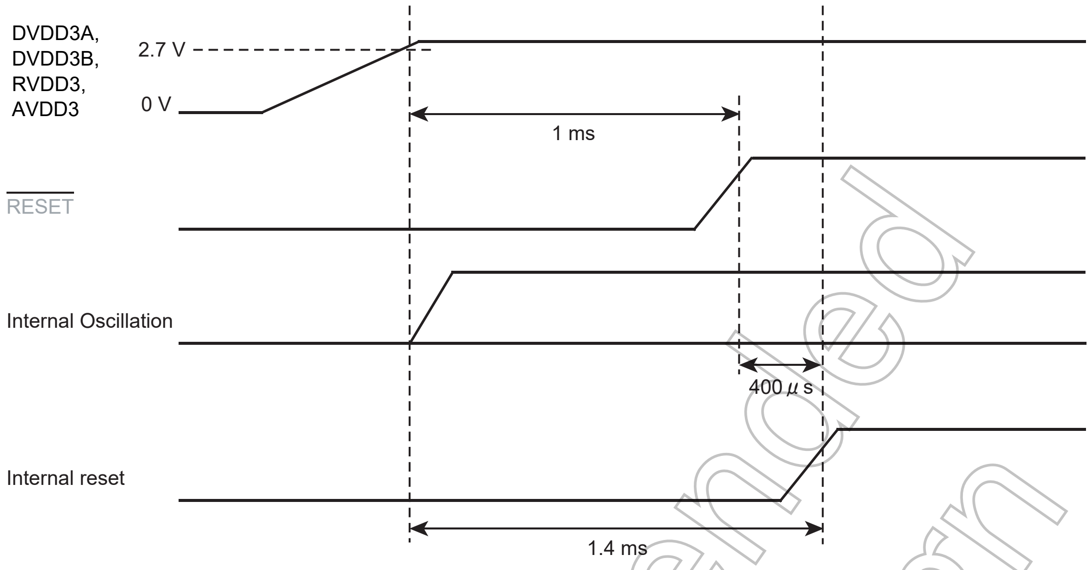


Figure 7-1 Cold Reset Sequence

Note 1: Turn on the power while the $\overline{\text{RESET}}$ pin is fixed to "Low". Release the $\overline{\text{RESET}}$ pin while all the power supplies are stabilized within operating voltage (DVDD3A/DVDD3B/RVDD3/AVDD3) and after an elapse of 1ms or more from while all the power supplies are stabilized within operating voltage.

Note 2: The above sequence is applied as well when restoring power.

7.3 Warm reset

7.3.1 Reset period

As a precondition, ensure that the power supply voltage is within the operating range and the internal high-frequency oscillator is providing stable oscillation.

To reset the TMPM341FDXBG/FYXBG, assert the $\overline{\text{RESET}}$ signal (active low) for a minimum duration of 12 system clocks. And When the $\overline{\text{RESET}}$ signal input at "Low" level while in STOP2 mode, the $\overline{\text{RESET}}$ signal is fixed to "Low" more than 500μs as internal regulator stability time.

After the external reset ($\overline{\text{RESET}}$) signal is released, the internal reset signal remains asserted for a further 400 μs.

7.4 After reset

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

The processor core's system debug components (FPB, DWT, ITM) register, the clock generator's CGRSTFLG register and the FCSECBIT register are initialized by following factors. And FCSECBIT register is initialized by STOP2 mode released.

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

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

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

The factor of register initialization

Register	Factors	
CGRSTFLG	Cold reset	External Reset
FCSECBIT	Cold reset	STOP2 mode released
FPB, DWT, ITM	Cold reset	STOP2 mode released (Note

Note: When a debug tool is connecting, these registers are not initialized.

8. Oscillation Frequency Detector (OFD)

The oscillation frequency detector circuit (OFD) detects abnormal clock frequency. To use the OFD, abnormal states of clock such as a harmonic, a subharmonic or stopped state can be detected.

The OFD monitors the target clock frequency using reference frequency and generates a reset signal if abnormal state is detected. TMPM341FDXBG/FYXBG uses internal high-speed oscillator clock as a reference and the target clock is an external high-speed oscillator clock.

Note: It is not guaranteed that OFD can detect all defects at any time, and it is not a circuit to measure error frequency.

8.1 Block diagram

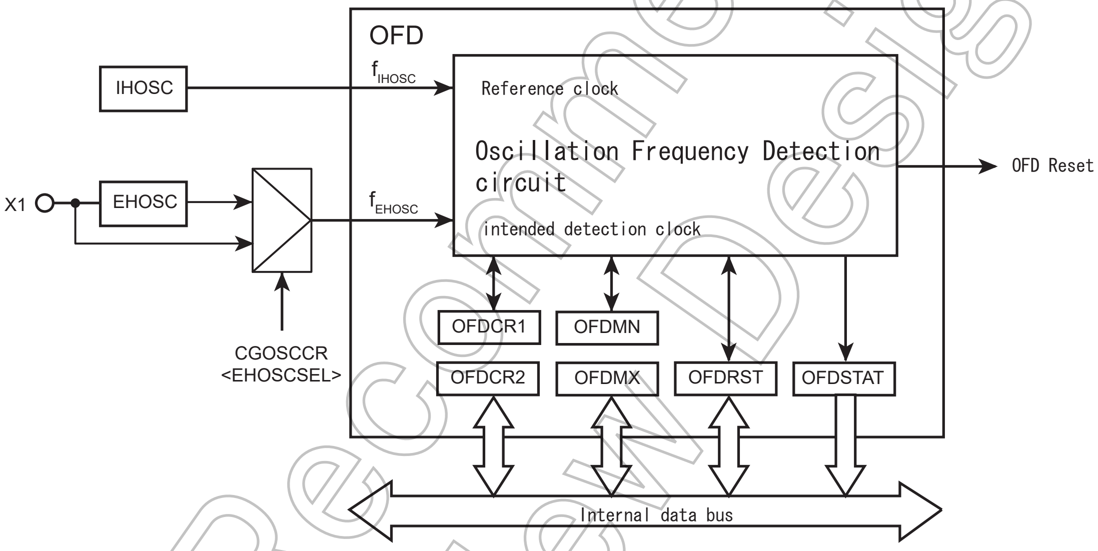


Figure 8-1 Oscillation Frequency Detector Block diagram

8.2 Registers

8.2.1 Register List

Base Address = 0x400F_1000

Register name		Address (Base+)
Control register 1	OFDCR1	0x0000
Control register 2	OFDCR2	0x0004
Lower detection frequency setting register	OFDMN	0x0008
Reserved	-	0x000C
Higher detection frequency setting register	OFDMX	0x0010
Reserved	-	0x0014
Reset control register	OFDRST	0x0018
Status register	OFDSTAT	0x001C

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

8.2.1.1 OFDCR1 (Control register 1)

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

Bit	Bit Symbol	Type	Description
31-8	−	R	Read as 0.
7-0	OFDWEN[7:0]	R/W	Controls register write 0x06: Disable 0xF9: Enable Setting 0xF9 enables to write registers except OFDCR1. When writing a value except 0x06 or 0xF9, 0x06 is written. If writing register is disabled, reading from each register is enabled.

8.2.1.2 OFDCR2 (Control register 2)

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

Bit	Bit Symbol	Type	Description
31-8	−	R	Read as 0.
7-0	OFDEN[7:0]	R/W	Controls frequency detecting. 0x00: Disable 0xE4: Enable Writing a value except 0x00 or 0xE4 is invalid and a value will not be changed.

8.2.1.3 OFDMN (Lower detection frequency setting register)

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

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMN[8:0]	R/W	Sets lower detection frequency.

Note: Writing to the register of OFDMN is protected while OFD circuit is operating.

8.2.1.4 OFDMX (Higher detection frequency setting register)

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

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMX[8:0]	R/W	Sets higher detection frequency.

Note: Writing to the register of OFDMX is protected while OFD circuit is operating.

8.2.1.5 OFDRST (Reset control register)

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

Bit	Bit Symbol	Type	Description
31-1	−	R	Read as 0.
0	OFDRSTEN	R/W	Controls generating a reset. 0: Disable 1: Enable

Note: Writing to the register of OFDRST is protected while OFD circuit is operating.

8.2.1.6 OFDSTAT (Status register)

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

Bit	Bit Symbol	Type	Description
31-2	−	R	Read as 0.
1	OFDBUSY	R	OFD operation 0: Run 1: Stop
0	FRQERR	R	Error detecting flag 0: No Error 1: Error

8.3 Operational Description

8.3.1 Setting

All register except OFDCR1 can not be written by reset. They are can be written by writing "0xF9" to OFDCR1.

The range of detection frequency is setting by OFDMX and OFDMN for each clock. The reset generation is set for enabling or disabling by OFDRST, and writing "0xE4" to OFDCR2 with OFDCR1="0xF9" enables the oscillation frequency detection.

To protect the mistaken writing, shout be written "0x06" to OFDCR1. And the register should be modified when OFD is stopped.

8.3.2 Operation

From the operation start-up to detection start-up, time length as two cycle of detecting clock is needed.

OFDSTAT<OFDBSY> can confirm whether it is operating. Detecting cycle is (reference clock frequency) / 2^8MHz.

When generating of reset is enabled, reset is generated on condition of the following.

- When the target clock frequency exceeds the frequency limit set by OFDMN and OFDMX.
- When a reference clock stopped.

OFD reset is the same operation as warm reset and the frequency sensing circuit itself is reset.

When generating reset is disabled, OFDSTAT<FRQERR> can be confirmed the condition.

Note: There are several factors of reset. Clock generator register CGRSTFLG can confirm the factors. For details of CGRSTFLG see chapter "exception."

8.3.3 Detection Frequency

The detection frequency have a detection frequency range and an undetectable frequency range because of oscillation accuracy. Therefore, it is undefined whether to be detected between detection frequency range and undetectable it.

Figure 8-2 shows the detection or undetectable frequency range when the target clock error is ±10% and the reference clock error is ± 5%.

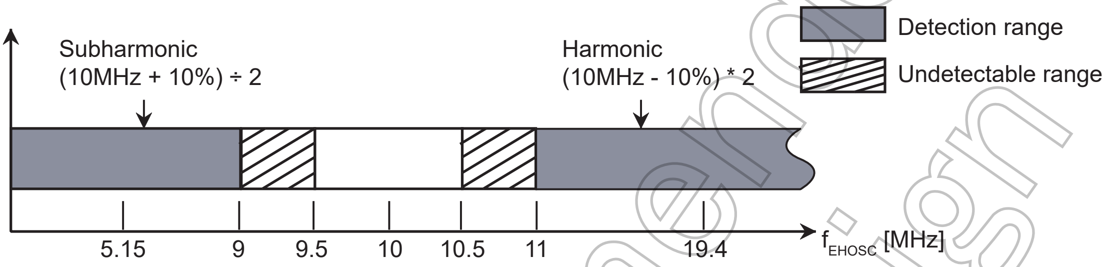


Figure 8-2 Example of detection frequency range (in case of 10MHz)

The higher and lower limit of the detection frequency is calculated from the maximum error of the target clock and the reference.

How to calculate the setup value of OFDMN/OFDMX is shown below when the target clock error is ±10% and the reference clock error is ± 5%.

target clock (f_{EHOSC})	10MHz ± 10%	Max.11MHz	--------	a
		Min. 9MHz	--------	b
reference clock (f_{IHOSC})	10MHz ± 5%	Max.10.5MHz	--------	c
		Min. 9.5MHz	--------	d

higher limit of the detection frequency $= 1 \div \{(d \div 2^8) \div (a \div 4)\}$

lower limit of the detection frequency $= 1 \div \{(c \div 28) \div (b \div 4)\}$

higher limit of the detection frequency $= 1 \div \{(9.5 \times 10^6 \div 2^8) \div (11 \times 10^6 \div 4)\}$ $= 74.10 = 74$ (truncate after the decimal places)

lower limit of the detection frequency $= 1 \div \{(10.5 \times 10^6 \div 2^8) \div (9 \times 10^6 \div 4)\}$ $= 54.85 = 55$ (round up after the decimal places)

Setting "0x74" to the register OFDMX and "0x55" to the register OFDMN, when the external oscillation of higher than 11MHz or lower than 9MHz is detected, the oscillation frequency detector outputs a reset signal.

8.3.4 Available Operation Mode

The oscillation frequency detection is available only in NORMAL and IDLE mode. Before shifting to another mode, disable the oscillation frequency detection.

8.3.5 Example of Operational Procedure

The example of operational procedure is shown below.

After reset, confirms various reset factor by CGRSTFLG. If the reset factor is not by the oscillation frequency detect, enable external oscillation, set register to use OFD and enable operation. Reset output must be disabled at this time.

After waiting the OFD operation is started, confirms abnormal status flag, and if there is not abnormal status, change to external oscillation clock.

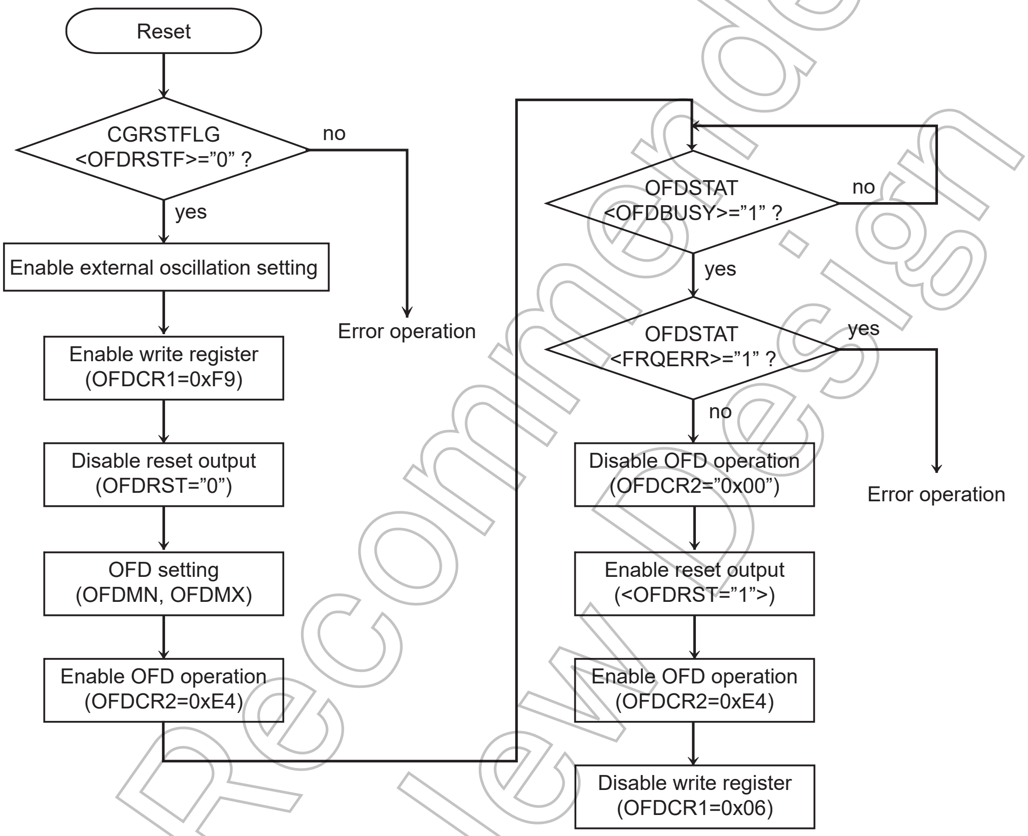

Reset
CGRSTFLG <OFDRSTF>="0" ?
no
yes
Error operation
Enable external oscillation setting
Enable write register (OFDCR1=0xF9)
Disable reset output (OFDRST="0")
OFD setting (OFDMN, OFDMX)
Enable OFD operation (OFDCR2=0xE4)
OFDSTAT <OFDBUSY>="1" ?
no
yes
OFDSTAT <FRQERR>="1" ?
yes
no
Error operation
Disable OFD operation (OFDCR2="0x00")
Enable reset output (<OFDRST="1">)
Enable OFD operation (OFDCR2=0xE4)
Disable write register (OFDCR1=0x06)

Figure 8-3 Example of operational procedure

9. Watchdog Timer(WDT)

The watchdog timer (WDT) is for detecting malfunctions (runaways) of the CPU caused by noises or other disturbances and remedying them to return the CPU to normal operation.

If the watchdog timer detects a runaway, it generates a INTWDT interrupt or reset.

Note: INTWDT interrupt is a factor of the non-maskable interrupts (NMI).

Also, the watchdog timer notifies of the detecting malfunction to the external peripheral devices from the watchdog timer pin ($\overline{\text{WDTOUT}}$) by outputting "Low".

Note: This product does not have the watchdog timer out pin ($\overline{\text{WDTOUT}}$).

9.1 Configuration

Figure 9-1shows the block diagram of the watchdog timer.

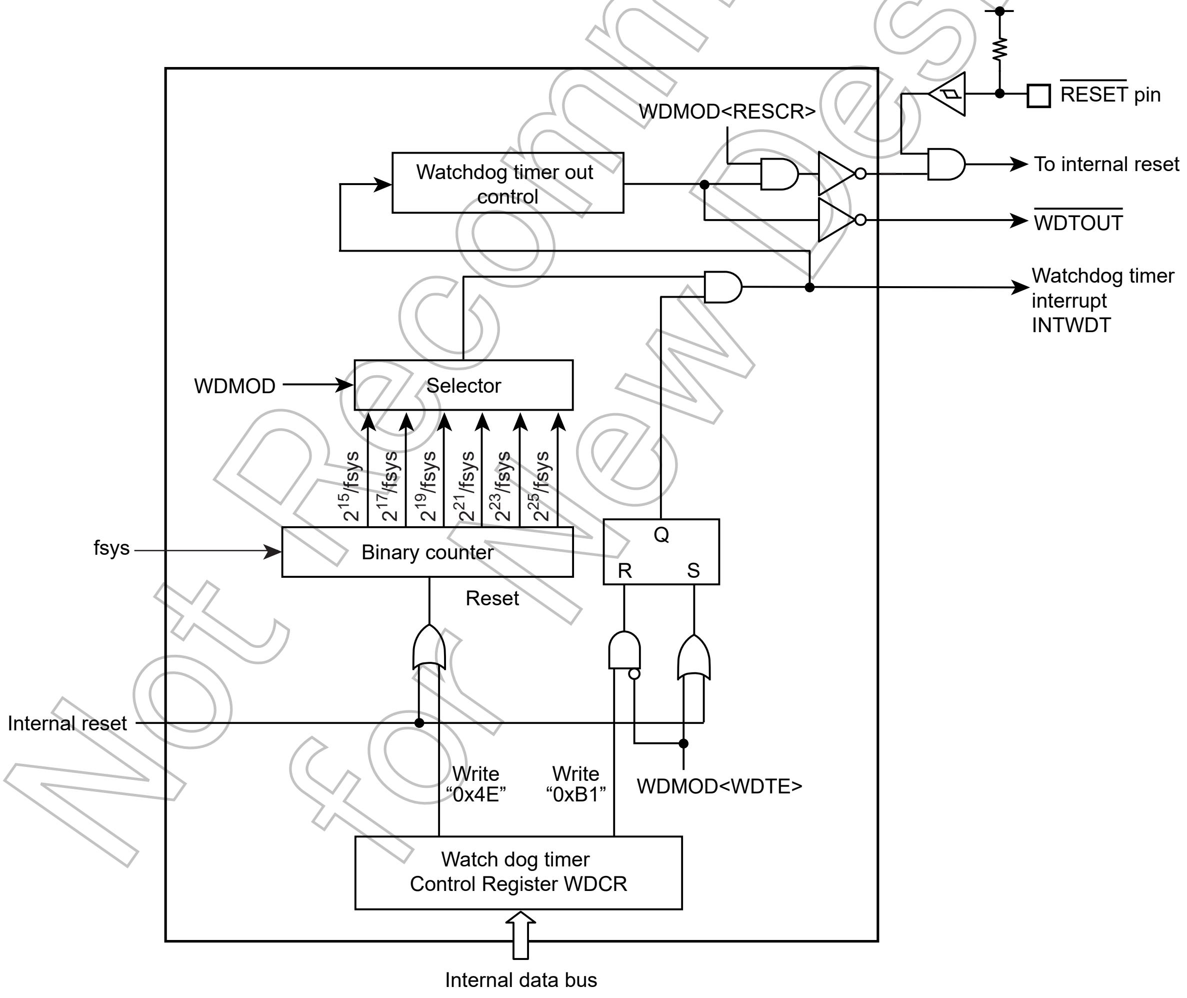


Figure 9-1 Block Diagram of the Watchdog Timer

9.2 Register

The followings are the watchdog timer control registers and addresses.

Base Address = 0x400F_2000

Register name		Address(Base+)
Watchdog Timer Mode Register	WDMOD	0x0000
Watchdog Timer Control Register	WDCR	0x0004

9.2.1 WDMOD(Watchdog Timer Mode Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	WDTE	WDTP			-	I2WDT	RESCR	-
After reset	1	0	0	0	0	0	1	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	WDTE	R/W	Enable/Disable control 0:Disable 1:Enable
6-4	WDTP[2:0]	R/W	Selects WDT detection time(Refer toTable 9-1) 000: 2^{15}/fsys 100: 2^{23}/fsys 001: 2^{17}/fsys 101: 2^{25}/fsys 010: 2^{19}/fsys 110:Setting prohibited. 011: 2^{21}/fsys 111:Setting prohibited.
3	−	R	Read as 0.
2	I2WDT	R/W	Operation when IDLE mode 0: Stop 1:In operation
1	RESCR	R/W	Operation after detecting malfunction 0: INTWDT interrupt request generates. (Note) 1: Reset
0	−	R/W	Write 0.

Note: INTWDT interrupt is a factor of the non-maskable interrupts (NMI).

Table 9-1 Detection time of watchdog timer (fc = 40 MHz)

Clock gear value CGSYSCR<GEAR[2:0]>	WDMOD<WDTP[2:0]>					
	000	001	010	011	100	101
000 (fc)	0.82 ms	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms
100 (fc/2)	1.63 ms	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s
101 (fc/4)	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s
110 (fc/8)	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s	6.71 s
111 (fc/16)	13.12 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s	13.42 s

9.2.2 WDCR (Watchdog Timer Control Register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	WDCR	W	Disable/Clear code 0xB1:Disable code 0x4E: Clear code Others:Reserved

9.3 Operations

9.3.1 Basic Operation

The Watchdog timer is consists of the binary counters that work using the system clock (fsys) as an input. Detecting time can be selected between 2^{15}, 2^{17}, 2^{19} , 2^{21}, 2^{23} and 2^{25} by the WDMOD<WDTP[2:0]>. The detecting time as specified is elapsed, the watchdog timer interrupt (INTWDT) generates, and the watchdog timer out pin ($\overline{\text{WDTOUT}}$) output "Low".

To detect malfunctions (runaways) of the CPU caused by noise or other disturbances, the binary counter of the watchdog timer should be cleared by software instruction before INTWDT interrupt generates. If the binary counter is not cleared, the non-maskable interrupt generates by INTWDT. Thus CPU detects malfunction (runway), malfunction countermeasure program is performed to return to the normal operation.

Additionally, it is possible to resolve the problem of a malfunction (runaway) of the CPU by connecting the watchdog timer out pin to reset pins of peripheral devices.

Note: This product does not include a watchdog timer out pin ($\overline{\text{WDTOUT}}$).

9.3.2 Operation Mode and Status

The watchdog timer begins operation immediately after a reset is cleared.

If not using the watchdog timer, it should be disabled.

The watchdog timer cannot be used as the high-speed frequency clock is stopped. Before transition to below modes, the watchdog timer should be disabled.

- STOP1 mode
- STOP2 mode

In IDLE mode, its operation depends on the WDMOD <I2WDT> setting.

Also, the binary counter is automatically stopped during debug mode.

9.4 Operation when malfunction (runaway) is detected

9.4.1 INTWDT interrupt generation

In the Figure 9-2 shows the case that INTWDT interrupt generates (WDMOD<RESCR>="0").

When an overflow of the binary counter occurs, INTWDT interrupt generates. It is a factor of non-maskable interrupt (NMI). Thus CPU detects non-maskable interrupt and performs the countermeasure program.

The factor of non-maskable interrupt is the plural. CGNMIFLG identifies the factor of non-maskable interrupts. In the case of INTWDT interrupt, CGNMIFLG<NMIFLG0> is set.

When INTWDT interrupt generates, simultaneously the watchdog timer out ($\overline{\text{WDTOUT}}$) output "Low". $\overline{\text{WDTOUT}}$ becomes "High" by the watchdog timer clearing that is writing clear code 0x4E to the WDCR register.

Note: This product does not have the watchdog timer output pin($\overline{\text{WDTOUT}}$).

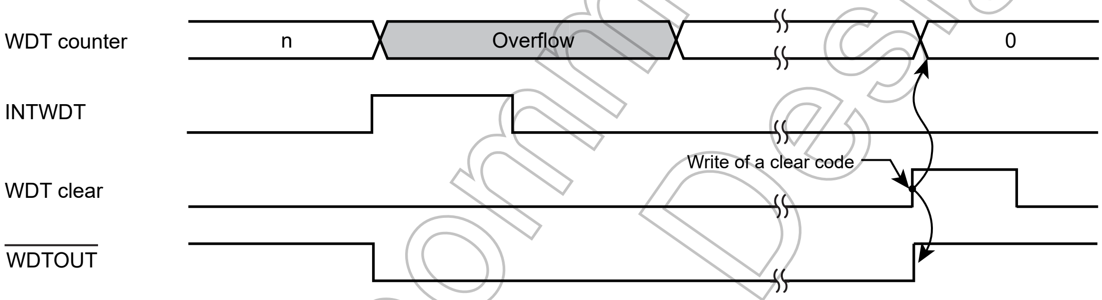


Figure 9-2 INTWDT interrupt generation

9.4.2 Internal reset generation

Figure 9-3 shows the internal reset generation (WDMOD<RESCR>="1").

MCU is reset by the overflow of the binary counter. In this case, reset status continues for 32 states. A clock is initialized so that input clock (fsys) is the same as a high-speed frequency clock (fosc). This means fsys = fosc.

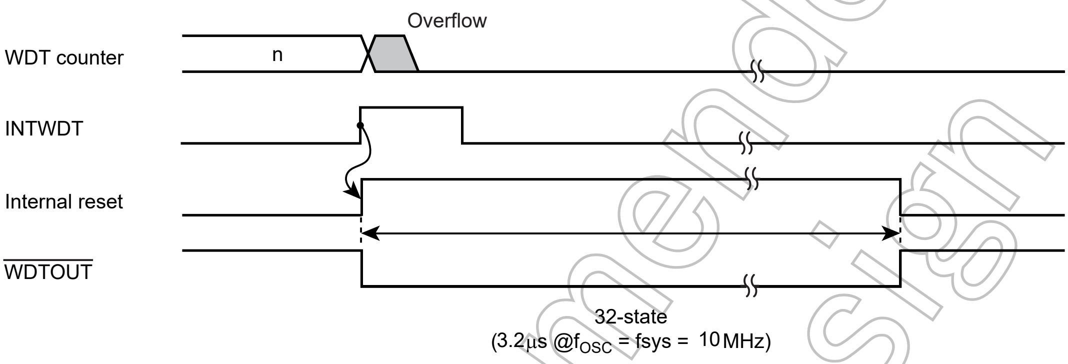


Figure 9-3 Internal reset generation

9.5 Control register

The watchdog timer (WDT) is controlled by two control registers WDMOD and WDCR.

9.5.1 Watchdog Timer Mode Register (WDMOD)

1. Specifying the detection time of the watchdog timer <WDTP[2:0]>.

   Set the watchdog timer detecting time to WDMOD<WDTP[2:0]>. After reset, it is initialized to WDMOD<WDTP[2:0]> = "000".

2. Enabling/disabling the watchdog timer <WDTE>.

   When resetting, WDMOD <WDTE> is initialized to "1" and the watchdog timer is enabled.

   To disable the watchdog timer to protect from the error writing by the malfunction, first <WDTE> bit is set to "0", and then the disable code (0xB1) must be written to WDCR register.

   To change the status of the watchdog timer from "disable" to "enable," set the <WDTE> bit to "1".

3. Watchdog timer out reset connection <RESCR>

   This register specifies whether WDTOUT is used for internal reset or interrupt. After reset, WDMOD<RESCR> is initialized to "1", the internal reset is generated by the overflow of binary counter.

9.5.2 Watchdog Timer Control Register(WDCR)

This is a register for disabling the watchdog timer function and controlling the clearing function of the binary counter.

9.5.3 Setting example

9.5.3.1 Disabling control

By writing the disable code (0xB1) to this WDCR register after setting WDMOD <WDTE> to "0," the watchdog timer can be disabled and the binary counter can be cleared.

		7	6	5	4	3	2	1	0	
WDMOD	←	0	−	−	−	−	−	−	−	Set <WDTE> to "0".
WDCR	←	1	0	1	1	0	0	0	1	Writes the disable code (0xB1).

9.5.3.2 Enabling control

Set WDMOD <WDTE> to "1".

		7	6	5	4	3	2	1	0	
WDMOD	←	1	−	−	−	−	−	−	−	Set <WDTE> to "1".

9.5.3.3 Watchdog timer clearing control

Writing the clear code (0x4E) to the WDCR register clears the binary counter and it restarts counting.

		7	6	5	4	3	2	1	0	
WDCR	←	0	1	0	0	1	1	1	0	Writes the clear code (0x4E).

9.5.3.4 Detection time of watchdog timer

In the case that 2^{21}/fsys is used, set "011" to WDMOD<WDTP[2:0]>.

		7	6	5	4	3	2	1	0
WDMOD	←	1	0	1	1	−	−	−	−

10. Clock/Mode control

10.1 Features

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

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

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

The clock/mode control block has the following functions:

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

In addition to NORMAL mode, the TMPM341FDXBG/FYXBG can operate low power mode to reduce power consumption according to its usage conditions.

10.2 Registers

10.2.1 Register List

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

Base Address = 0x400F_3000

Register name		Address (Base+)
System control register	CGSYSCR	0x0000
Oscillation control register	CGOSCCR	0x0004
Standby control register	CGSTBYCR	0x0008
PLL selection register	CGPLLSEL	0x000C
System clock selection register	CGCKSEL	0x0010
Reserved	-	0x0014
Reserved	-	0x0038
Protect register	CGPROTECT	0x003C

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

10.2.2 CGSYSCR (System control register)

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

Bit	Bit Symbol	Type	Function
31-21	−	R	Read as "0".
20	FCSTOP	R/W	ADC clock 0: Enable 1: Stop Enables to stop providing ADC clock. ADC clock is provided after reset. Confirming that ADC is stopped or finished in advance is required when setting "1"(stop).
19-18	−	R	Read as "0".
17-16	SCOSEL[1:0]	R/W	SCOUT out 00: Reserved 01: fsys/2 10: fsys 11: φT0 Enables to output the specified clock from SCOUT pin.
15-14	−	R	Read as "0".
13	−	R/W	Write "0".
12	FPSEL	R/W	Selects fperiph source clock 0: fgear 1: fc Specifies the source clock to fperiph. Selecting fc fixes fperiph regardless of the clock gear.
11	−	R	Read as "0".
10-8	PRCK[2:0]	R/W	Prescaler clock 000: fperiph 100: fperiph/16 001: fperiph/2 101: fperiph/32 010: fperiph/4 110: Reserved 011: fperiph/8 111: Reserved Specifies the prescaler clock to peripheral I/O.
7-3	−	R	Read as "0".
2-0	GEAR[2:0]	R/W	High-speed gear clock (fgear) 000: fc 100: fc/2 001: Reserved 101: fc/4 010: Reserved 110: fc/8 011: Reserved 111: fc/16

Note: Do not set the reserved value.

10.2.3 CGOSCCR (Oscillation control register)

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

Bit	Bit Symbol	Type	Function
31-20	WUODR[11:0]	R/W	Warm-up counter setup value. Setup the 16-bit timer for warm-up timer of upper 12-bits counter value.
19	HWUPSEL	R/W	High-speed warm-up clock. 0: internal (IHOSC) 1: external (feosc) Selects warm-up counter by high-speed oscillator.
18	EHOSCSEL	R/W	External oscillator. 0: input external clock (EHCLKIN) 1: external crystal oscillator (EHOSC)
17	OSCSEL	R/W	High-speed oscillator 0: internal high-speed oscillator (IHOSC) 1: external high-speed oscillator (EHOSC) The source clock of fosc is selected.
16	XEN2	R/W	Internal high-speed oscillator operation 0: Stop 1: Oscillation
15-12	−	R/W	Write "0".
11-10	−	R	Read as "0".
9	−	R/W	Write "0".
8	XEN1	R/W	External high-speed oscillator operation 0: Stop 1: Oscillation
7-3	−	R/W	Write "0_0110"
2	PLLON	R/W	PLL (multiplying circuit) operation 0: Stop 1: Oscillation
1	WUEF	R	Operation status of warm-up timer 0: warm-up completed 1: warm-up operation Can be confirmed the status of the warm-up timer.
0	WUEON	W	Operation of warm-up timer 0: don’t care 1: starting warm-up Enables to start the warm-up timer. Read as "0".

Note 1: Refer to Section "10.3.4 Warm-up function" for setting the time of warm-up timer.

Note 2: When selecting external oscillator, select <OSCSEL> after setting <EHOSCSEL>.

(Do not change <EHOSCSEL> and <OSCSEL> simultaneously)

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

Note 4: When using internal high-speed oscillator (IHOSC) as system clock, do not use PLL multiplying.

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

10.2.4 CGSTBYCR (Standby control register)

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

Bit	Bit Symbol	Type	Function
31-20	−	R	Read as "0".
19-18	−	R/W	Write "0".
17	PTKEEP	R/W	I/O port control in STOP2 mode. 0: Output the contents of output latch 1: Hold the port state when CGSTBYCR <PTKEEP> is changed from "0" to "1". (It is necessary to set <PTKEEP> before entering STOP2 mode)
16	DRVE	R/W	Pin status in STOP1 mode. 0: Inactive in STOP1 mode 1: Active in STOP1 mode
15-3	−	R	Read as "0".
2-0	STBY[2:0]	R/W	Selects low power consumption mode 000: Reserved 001: STOP1 010: Reserved 011: IDLE 100: Reserved 101: STOP2 110: Reserved 111: Reserved

Note:Do not set the reserved value.

10.2.5 CGPLLSEL (PLL Selection Register)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-1	PLLSET[14:0]	R/W	PLL multiplying value (Do not use except below) 0x303D: 16 times 0x281D: 8 times
0	PLLSEL	R/W	Use of PLL 0: fosc 1: f_{PLL} / 4 Specifies use or disuse of the clock multiplied by the PLL. fosc is automatically set after reset. Setting is required when using the PLL.

Note 1: Select PLL multiplying value which is shown in Table 10-2 and Table 10-3.

Note 2: Select PLL multiplying value when CGOSCCR<PLLON>=0 (PLL stop).

Note 3: Returning from the STOP1/STOP2 mode, related bits <PLLSEL>, CGOSCCR<HWUPSEL>, <OSCSEL>, <XEN2>, <XEN1>, and <PLLON> are initialized in order to start internal high-speed oscillator.

Note 4: When using internal high-speed oscillator (IHOSC) as system clock, do not use PLL multiplying.

10.2.6 CGPWMGEAR (Timer clock selection register)

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

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as "0".
5-4	PWMGEAR [1:0]	R/W	Source clock 00 : f_{PLL} 01 : f_{PLL} / 2 10 : f_{PLL} / 4 11 : Reserved Selects source clock to TMRD.
3-1	−	R	Read as "0".
0	TMRDCLKEN	R/W	Provides TMRDCLK to TMRD 0 : Disable 1 : Enable

Note 1: Do not set the reserved value.

Note 2: When changing source clock by <PWMGEAR>, stop TMRD (<TMRDCLKEN>="0").

10.2.7 CGPROTECT (Protect register)

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

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

10.3 Clock control

10.3.1 Clock Type

Each clock is defined as follows:

fosc : Clock generated by internal oscillator. Clock input from the X1 and X2 pins.

f_{PLL} : Clock multiplied by 8 or 16 by PLL.

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

fgear : Clock specified by CGSYSCR<GEAR[2:0]> (gear clock)

fsys : Clock specified by same as fgear clock (system clock)

fperiph : Clock specified by CGSYSCR<FPSEL>

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

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

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

10.3.2 Initial Values after Reset

Reset operation initializes the clock configuration as follows.

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

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

fc = fosc
fsys = fosc
φT0 = fosc

10.3.3 Clock system Diagram

Figure 10-1 shows the clock system diagram.

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

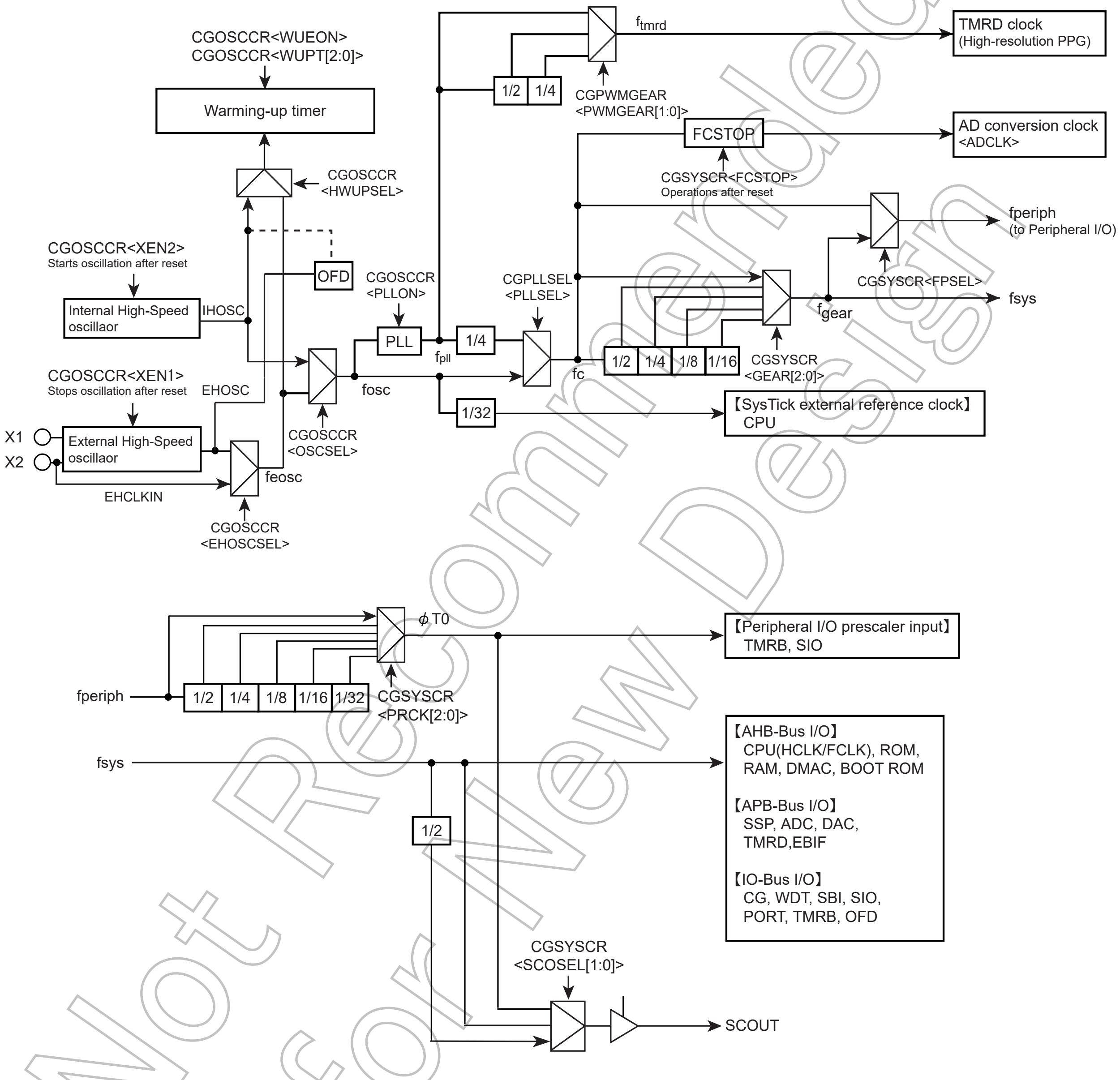


Figure 10-1 Clock Block Diagram

10.3.4 Warm-up function

The warm-up function secures the stability time for the oscillator and the PLL with the warm-up timer. When using external clock inputting, Warm-up function is not necessary when using stable external clock.

Refer to "10.6.7 Warm-up" for a detail.

How to configure the warm-up function is described.

1. Specify the count up clock

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

2. Specify the warm-up counter value

   The warm-up time can be calculated by following formula with round lower 4 bit off, set to the bit of <WUODR[11:0]>.

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

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

$$\frac{\text{warm-up time to set}}{\text{input frequency cycle (s)}} = \frac{5\text{ms}}{1/8\text{MHz}} = 4000 \text{ cycle} = 0\text{x}9\text{C}40$$

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

3. Start warm-up function and confirm the completion of warm-up

   When the warm-up is started by software, by setting CGOSCCR<WUEON> to "1", the warm-up start a count up. The CGOSCCR <WUEF> is used to confirm the start and completion of warm-up. <WUEF> = "1" shows under warm-up and <WUEF> = "0" shows completion of warm-up.

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

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

The example of warm-up function setup is shown below.

Table 10-1 The example of warm-up function setup (When Internal high-speed oscillator is used)

CGOSCCR<WUODR[11:0]> = "0x9C4"	: Specify the warm-up time
CGOSCCR<WUODR[11:0]> read	: Confirm warm-up time reflecting Repaet until the read data is "0x9C4".
CGOSCCR<XEN2> = "1"	: Internal high-speed oscillator (fosc) enable
CGOSCCR<WUEON> = "1"	: Start the warm-up timer (WUP)
CGOSCCR<WUEF> read	: Wait until the state becomes "0" (warm-up is finished)

Note 1: It is not required the warm-up time in using the external clock to be stabled.

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

Note 3: After setting warm-up count value to OSCCR<WUDOR>, wait until confirming of the value to be reflected, then change to the standby mode by WFI instruction.

Note 4: When returning from STOP1/STOP2 mode, related bits CGPLLSEL<PLLSEL>, CGOSCCR<HWUPSEL>, <OSCSEL>, <XEN2>, <XEN1> and <PLLON> are initialized in order to start internal high-speed oscillator and CGOSCCR<WUDOR[11:0]> is not initialized.

10.3.5 Clock Multiplication Circuit (PLL)

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

10.3.5.1 How to configure the PLL function

The PLL is disabled after reset.

To enable the PLL, set CGPLLSEL<PLLSET> to multiplying value. And set <PLLON> to "1" after 100μs for initialize time of PLL. After 100μs for lock-up time elapses, set CGPLLSEL<PLLSEL> to "1", f_{PLL} which is multiplied by 8 or 16 from fosc is used.

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

Note: When using internal high-speed oscillator (IHOSC) as system clock, do not use PLL multiplying.

As for the 8 or 16 multiplying value, only the following setting are permitted.

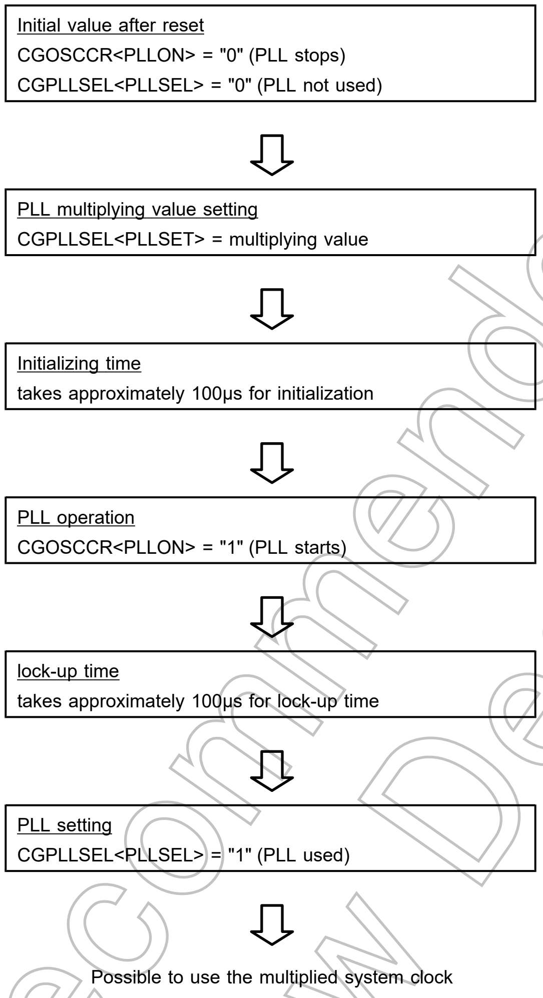
Multiplying	<PLLSEL>
8	0x281D
16	0x303D

10.3.5.2 Change PLL multiplying

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

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

(1) PLL setup sequence



Initial value after reset
CGOSCCR<PLLON> = "0" (PLL stops)
CGPLLSEL<PLLSEL> = "0" (PLL not used)

PLL multiplying value setting
CGPLLSEL<PLLSET> = multiplying value

Initializing time
takes approximately 100μs for initialization

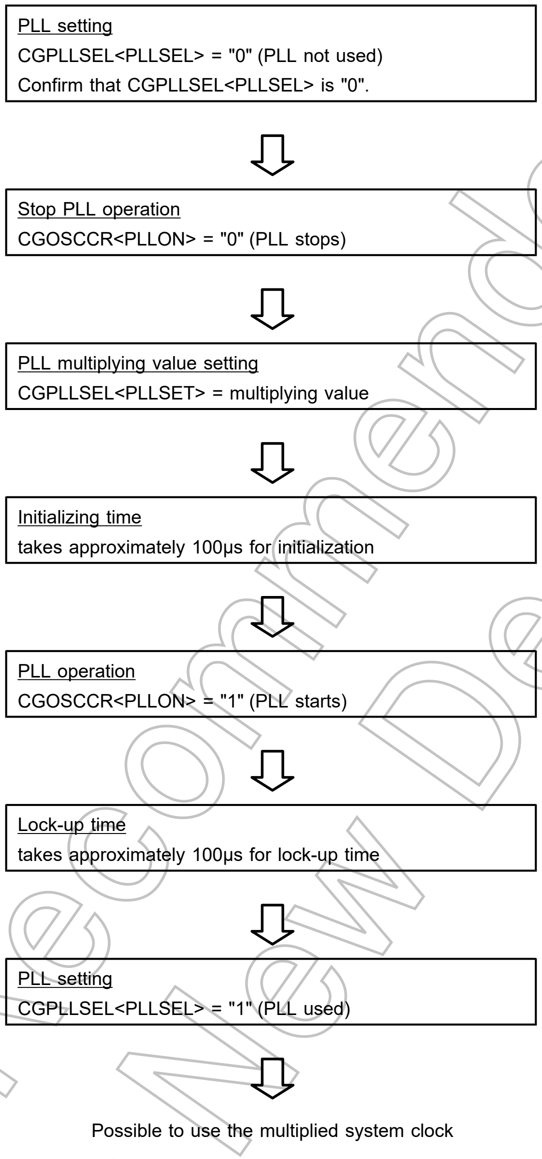
PLL operation
CGOSCCR<PLLON> = "1" (PLL starts)

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

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

Possible to use the multiplied system clock

(2) Change PLL multiplying sequence



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

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

PLL multiplying value setting
CGPLLSEL<PLLSET> = multiplying value

Initializing time
takes approximately 100μs for initialization

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

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

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

Possible to use the multiplied system clock

10.3.6 System clock

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

When using internal high-speed oscillation clock, do not use it as system clock which high accuracy assurance is required.

When using external high-speed oscillation clock, the PLL function can be used by multiplying.

Source clock		Frequency	Using PLL
Internal high-speed oscillation (IHOSC)		10MHz	can not use
External high-speed oscillation	Oscillator (EHOSC)	8 to 16MHz	Not use, 8 or 16 multiplying
	Input clock (EHCLKIN)	8 to 27MHz	

Note: About multiplying of PLL and the external high-speed oscillation, refer to Table 10-2.

The clock generated with PLL by multiplying is used by high resolution 16-bit timer/PPG output function. Moreover, the clock four dividing can be used as a system clock and an ADC clock. The frequency that can be used respectively is as follows.

	System clock	ADC clock	High-resolution 16-bit timer/PPG output clock
Operation frequency (MHz)	1 to 54	40 (Max.)	160 (Max.)

The system clock can be divided by CGSYSCR<GEAR[2:0]>. Although the setting can be changed while operating, the actual switching takes place after a slight delay.

Table 10-2 shows the example of the operation frequency by the setting of PLL and the clock gear.

Table 10-2 System clock frequency when PLL is 8 times or 16 times

External oscillator (MHz)	External clock input (MHz)	PLL multiplying	Max. operation frequency (fc) (MHz)	Clock gear (CG) PLL = ON					Clock gear (CG) PLL = OFF				
				1/1	1/2	1/4	1/8	1/16	1/1	1/2	1/4	1/8	1/16
8	8	16	32	32	16	8	4	2	8	4	2	1	-
10	10		40	40	20	10	5	2.5	10	5	2.5	1.25	-
13.5	13.5		54	54	27	13.5	6.75	3.38	13.5	6.75	3.38	1.69	-
16	16	8	32	32	16	8	4	2	16	8	4	2	1
-	27		54	54	27	13.5	6.75	3.38	27	13.5	6.75	3.38	1.69

↑ Initial value after reset

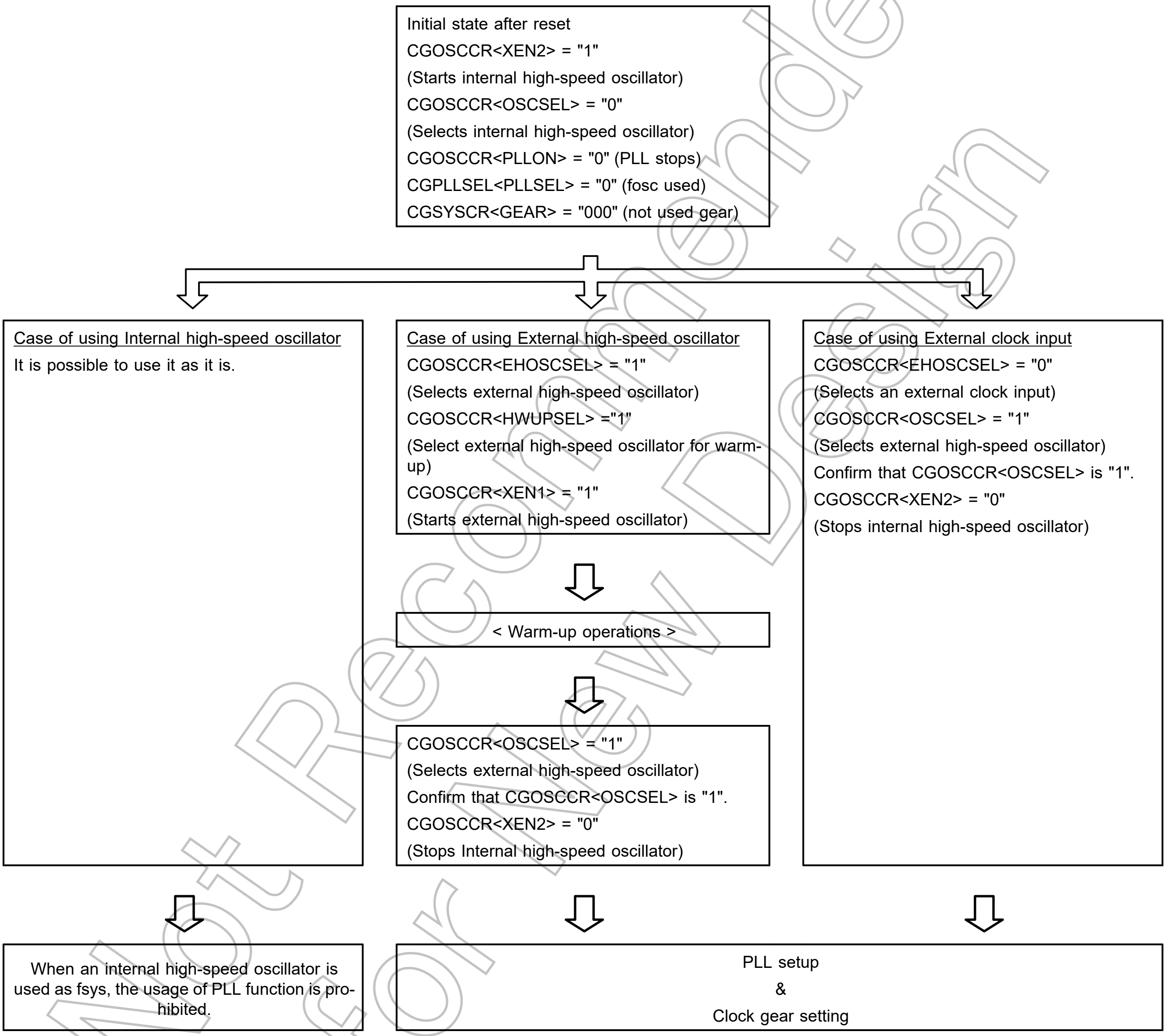
Note: Do not use 1/16 when SysTick is used.

10.3.6.1 Clock setting

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

The clock setup sequence is shown as follow.

Clock setup sequence



Initial state after reset
CGOSCCR<XEN2> = "1"
(Starts internal high-speed oscillator)
CGOSCCR<OSCSEL> = "0"
(Selects internal high-speed oscillator)
CGOSCCR<PLLON> = "0" (PLL stops)
CGPLLSEL<PLLSEL> = "0" (fosc used)
CGSYSCR<GEAR> = "000" (not used gear)

Case of using Internal high-speed oscillator
It is possible to use it as it is.

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

Case of using External high-speed oscillator
CGOSCCR<EHOSCSEL> = "1"
(Selects external high-speed oscillator)
CGOSCCR<HWUPSEL> ="1"
(Select external high-speed oscillator for warm-up)
CGOSCCR<XEN1> = "1"
(Starts external high-speed oscillator)

< Warm-up operations >

CGOSCCR<OSCSEL> = "1"
(Selects external high-speed oscillator)
Confirm that CGOSCCR<OSCSEL> is "1".
CGOSCCR<XEN2> = "0"
(Stops Internal high-speed oscillator)

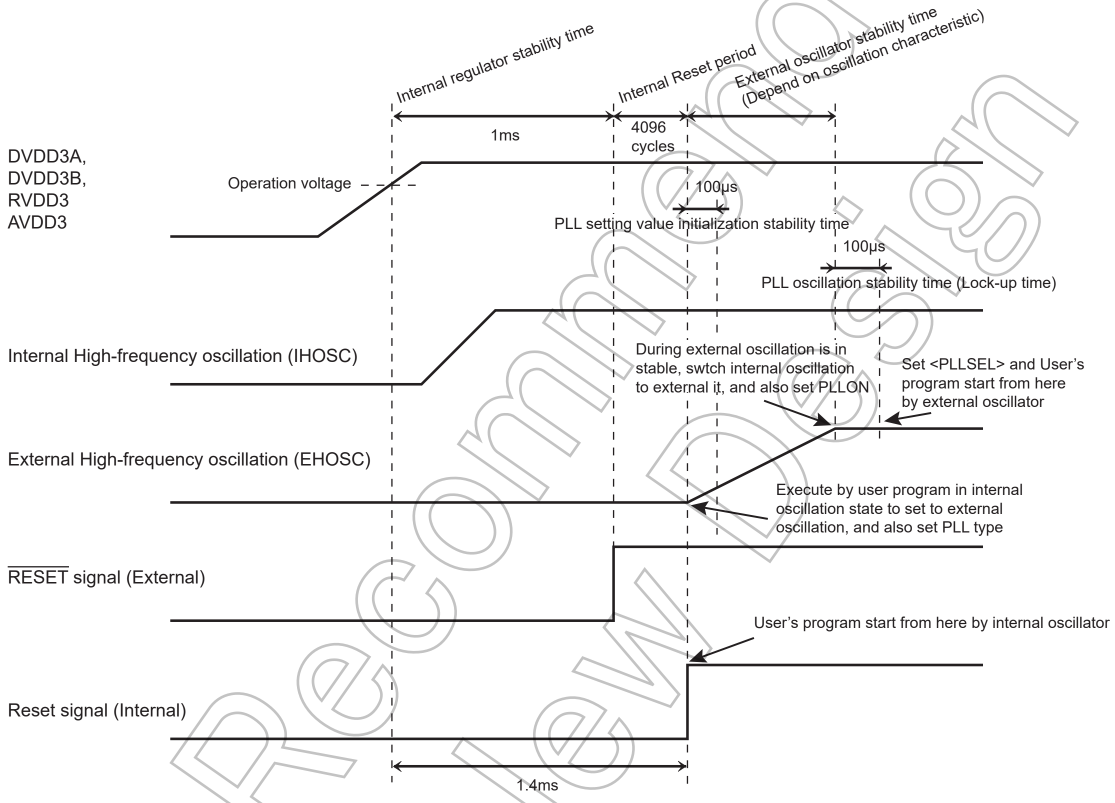
Case of using External clock input
CGOSCCR<EHOSCSEL> = "0"
(Selects an external clock input)
CGOSCCR<OSCSEL> = "1"
(Selects external high-speed oscillator)
Confirm that CGOSCCR<OSCSEL> is "1".
CGOSCCR<XEN2> = "0"
(Stops internal high-speed oscillator)

PLL setup
&
Clock gear setting

10.3.6.2 When using external oscillator

This product activates from an internal high-speed oscillator after releasing reset. When an external high-speed oscillator and Clock multiplication circuit (PLL) are used, it sets it according to the procedure of "10.3.5 Clock Multiplication Circuit (PLL)" and "10.3.6.1 Clock setting".

The following figures show the transition when external high-speed oscillator and clock multiplication circuit are used.



Internal regulator stability time
Internal Reset period
External oscillator stability time
(Depend on oscillation characteristic)
1ms
4096
cycles
DVDD3A,
DVDD3B,
RVDD3
AVDD3
Operation voltage
100μs
PLL setting value initialization stability time
100μs
PLL oscillation stability time (Lock-up time)
Internal High-frequency oscillation (IHOSC)
During external oscillation is in stable, swtch internal oscillation to external it, and also set PLLON
Set <PLLSEL> and User's program start from here by external oscillator
External High-frequency oscillation (EHOSC)
Execute by user program in internal oscillation state to set to external oscillation, and also set PLL type
$\overline{\text{RESET}}$ signal (External)
User's program start from here by internal oscillator
Reset signal (Internal)
1.4ms

Figure 10-2 Transition when high-speed oscillator is set with PLL

10.3.7 Clock for Peripherals

Table 10-3 shows when the high resolution 16-bit timer/PPG output function (TMRD), the AD converter function or the flash memory is used, the frequency that can be used respectively.

Table 10-3 Peripheral system clock setting example when PLL is 8 times or 16 times

External oscillator (MHz)	External clock input (MHz)	PLL multiplying	Max. operation frequency (fc) (MHz)	TMRD Max. operation frequency (MHz)	ADC Max. operation frequency (MHz)	INTLV pin setting (Note3)
8	8	16	32	128	32	Pull-down
10	10		40	160	40	Pull-down
13.5	13.5		54	108 (Note1)	27 (Note2)	Pull-up
16	16	8	32	128	32	Pull-down
-	27		54	108 (Note1)	27 (Note2)	Pull-up

Note 1: Maximum operating frequency of TMRD is 160MHz, which is 2 dividing f_{PLL}/2 frequency specified by CGPWMGEAR<PWMGEAR[1:0]>.

Note 2: Maximum operating frequency of AD convertor is 40MHz, which is 2 dividing fc/2 frequency specified by ADCLK<ADCLK>.

Note 3: The INTLV pin setting for the flash interleave access is defined by the maximum operation frequency (fc) that the INTLV pin must be set to pull-down setting when the value of fc is equal or less than 40MHz, meanwhile, the INTLV pin must be set to pull-up setting when the value of fc is over 40MHz.

10.3.8 Prescaler Clock Control

Peripheral function has a prescaler for dividing a clock. As the clock φT0 to be input to each prescaler, the "fperiph" clock specified in the CGSYSCR<FPSEL> can be divided according to the setting in the CGSYSCR<PRCK[2:0]>.

After reset, fperiph/1 is selected as φT0.

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

10.3.9 System Clock Pin Output Function

The TX03 enables to output the system clock from a pin. The SCOUT pin can output the system clock fsys and fsys/2, and the prescaler input clock for peripheral function φT0.

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

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

The setting to use port as SCOUT pin, refer to "Input/Output port". The output clock is selected by setting the CGSYSCR<SCOSEL[1:0]>.

Table 10-4 shows the pin status in each mode when the SCOUT pin is set to the SCOUT output.

Table 10-4 SCOUT Output Status in Each Mode

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

10.4 Modes and Mode Transitions

10.4.1 Mode Transitions

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

And TMPM341FDXBG/FYXBG has STOP2 mode that enables to reduce power consumption significantly by halting main voltage supply, retaining some functional operations.

Figure 10-3 shows a mode transition diagram.

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

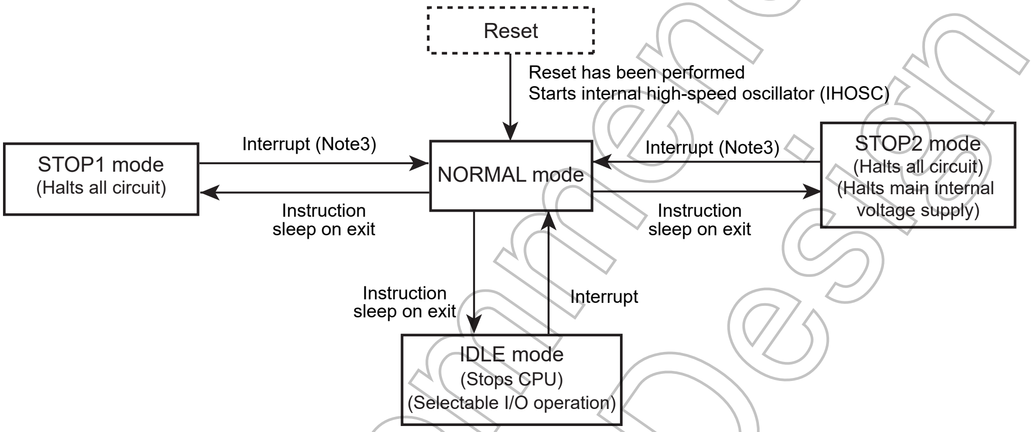


Figure 10-3 Mode Transition Diagram

Note 1: Returning from the STOP1/STOP2 mode, related bits <HWUPSEL>, <OSCSEL>, <XEN2>, <XEN1>, <PLLON> of CGOSCCR and CGPLLSEL<PLLSEL> are initialized in order to start internal high-speed oscillator.

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

Note 3: The warm-up is needed. The warm-up time must be set in NORMAL mode before entering to STOP mode. Regarding warm-up time, refer to "10.6.7 Warm-up".

10.5 Operation mode

10.5.1 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.

10.6 Low Power Consumption Modes

The TMPM341FDXBG/FYXBG has low power consumption modes: IDLE, STOP1 and STOP2. To shift to the low power consumption mode, specify the mode in the system control register CGSTBYCR<STBY[2:0]> and execute the WFI (Wait For Interrupt) instruction. In this case, execute reset or generate the interrupt to release the mode. Releasing by the interrupt requires settings in advance. See the chapter "Exceptions" for details.

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

Note 2: The TX03 does not support the low power consumption mode configured with the SLEEPDEEP bit in the Cortex-M3 core. Setting the <SLEEPDEEP> bit of the system control register is prohibited.

The features of IDLE, STOP1, STOP2 mode are described as follows.

10.6.1 IDLE mode

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

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

- 16-bit timer/event counter (TMRB)
- Two-phase pulse counter (PHCNT)
- High resolution 16-bit timer/PPG output (TMRD)
- Serial channel (SIO/UART)
- Serial bus interface (I2C/SIO)
- Analog Digital converter (ADC)
- Watch dog timer (WDT)

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

10.6.2 STOP1 mode

All the internal circuits including the internal oscillator are brought to a stop in STOP1 mode. And the internal oscillator activates the clock after releasing the STOP1 mode then transit to the Normal mode.

The STOP1 mode enables to select the pin status by setting the CGSTBYCR<DRVE>. Table 10-5 shows the pin status in the STOP1 mode.

10.6.3 STOP2 mode

This mode halts main voltage supply, retaining some function operation. This enables to reduce power consumption significantly compares to STOP1 mode. After releasing the STOP2 mode, voltage is supplied to the halted blocks, an internal high-speed oscillator starts, branches to ISR of reset and then returns to NORMAL mode. Before entering STOP2 mode, set CGSTBYCR<PTKEEP>="0"→"1" and keeps each port conditions. If internal voltage is halted, it can be held interface to the external IC, and STOP2 release source interrupt is available.

The state of pins are shown in Table 10-5.

Note 1: Returning from the STOP1/STOP2 mode, it is required to set the warm-up time in NORMAL mode before entering STOP1/STOP2 mode. Regarding the warm-up time, refer to 10.6.8.1 and 10.6.8.2.

Note 2: Returning from the STOP1/STOP2 mode, related bits <HWUPSEL>, <OSCSEL>, <XEN2>, <XEN1>, <PLLON> of CGOSCCR and CGPLLSEL<PLLSEL> are initialized in order to start internal high-speed oscillator. CGOSCCR<WUODR[11:0]> is not initialized.

Note 3: When releasing the STOP1/STOP2 mode by reset, the power-on counter activates an usual reset operation instead of the warm-up counter.

Note 4: Because STOP2 mode halts internal voltage supply, more than 45μs time is required from transition to release. If the STOP2 mode is released during this time, internal voltage cannot operate normally.

Table 10-5 Pin States in the STOP1/STOP2 mode

Function	Pin Name	I/O	STOP1		STOP2	
			<DRVE> = 0	<DRVE> = 1	<PTKEEP> = 0	<PTKEEP> = 1
Control Pin	$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, MODE, INTLV, BSC, ENDIAN	Input	o	o	o	o
Oscillator	X1/EHCLKIN	Input	×	×	×	×
	X2	Output	"High" level output.		×	×
Special pin	DA0, DA1	Output	×	×	×	×
Port	PJ0 to PJ5 (PHC0IN0, PHC0IN1, PHC1IN0, PHC1IN1, PHC2IN0, PHC2IN1) (Functional setting, case of PxFRn<PxmFn>="1")	Input	o	o	o	o
	PI3 to PI5 ($\overline{\text{TRST}}$, TDI, SWCLK/TCK) (Debug I/F setting, case of PxFRn<PxmFn>="1")	Input	Depends on PxIE[m]		o	Depends on PxIE[m]
	PI6 (SWDIO/TMS) (Debug I/F setting, case of PxFRn<PxmFn>="1")	Input	Depends on PxIE[m]		o	Depends on PxIE[m]
		Output	Enabled when data is valid. Disabled when data is invalid.		o	Depends on PxCR[m]
	PI7, PI2, PI1, PI0, PH6, PH5 (TDO/SWV, TRACECLK, TRACEDATA0 to 3) (Debug I/F setting, case of PxFRn<PxmFn>="1")	Output	Depends on PxCR[m]		o	Depends on PxCR[m]
	PG3, PG7, PC3, PC7, PD3, PE3, PF4, PF5, PG2, PJ7, PK1, PK3 (INT0 to B) (Interrupt setting, case of PxFRn<PxmFn>="1" and PxIE<PxmIE>="1")	Input	o	o	o	o
	If using other than listed above	Input	×	Depends on PxIE[m]	×	Depends on PxIE[m]
		Output	×	Depends on PxCR[m]	×	Depends on PxCR[m]

o : Valid input or output.

× : Invalid input or output.

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

10.6.4 Low power Consumption Mode Setting

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

Table 10-6 shows the mode setting in the <STBY[2:0]>.

Table 10-6 Low power consumption mode setting

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

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

10.6.5 Operational Status in Each Mode

Table 10-7 show the operational status in each mode.

Table 10-7 Operational Status in Each Mode

Block	NORMAL Internal high-speed oscillator use (IHOSC)	NORMAL External high-speed oscillator use (EHOSC)	IDLE Internal high-speed oscillator use (IHOSC)	IDLE External high-speed oscillator use (EHOSC)	STOP1 (Note 1)	STOP2 (Note 1)
Processor core	o	o	−	−	−	×
DMAC	o	o	o	o	−	×
INTC	o	o	o	o	o	×
EBIF	o	o	o	o	−	×
I/O port	o	o	o	o	o(Note 2)	Δ(Note 3)
SIO/UART	o	o	Δ	Δ	−	×
I2C/SIO	o	o	Δ	Δ	−	×
TMRB	o	o	Δ	Δ	−	×
PHCNT	o	o	Δ	Δ	o	×
TMRD	o	o	Δ	Δ	−	×
WDT	o	o	Δ(Note 6)	Δ(Note 6)	−	×
SSP	o	o	o	o	−	×
12-bit ADC	o	o	Δ	Δ	−	×
10-bit DAC	o	o	o	o	−	−
CG	o	o	o	o	o	o
PLL	o	o	Δ	Δ	−	×
External high-speed oscillator (EHOSC)	Δ	o	Δ	o	−	×
OFD	Δ(Note 4)	o	Δ(Note 4)	o	−	×
Internal high-speed oscillator (IHOSC)	o	o(Note 5)	o	o(Note 5)	−	×
Main RAM	o	o	o	o	o	×
Backup RAM	o	o	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.

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

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

Note 1: Before transit to STOP1/STOP2 mode, stop peripheral functions of "−" and "×". It is available to reduce leakage current by stopping reference voltage for AD converter or DA converter.

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

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

Note 4: When using internal high-speed oscillator, OFD can not be used.

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

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

Note 7: Regarding to the way of stopping ADC when the CPU enters in the low power consumption mode. Refer to the way of stopping ADC in the low power consumption mode.

10.6.6 Releasing the Low Power Consumption Mode

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

Details are shown in Table 10-8.

Table 10-8 Release Source in Each Mode

Low power consumption mode			IDLE	STOP1	STOP2
Release source	Interrupt	INT0 to B (Note2)	o	o	・ (Note3)
		INTTB0 to 9	o	×	×
		INTTD0CMP0 to 4, INTTD1CMP0 to 4	o	×	×
		INTPHT00 to 31, INTPHEVRY0 to 3	o	o	×
		INTCAP00 to 91	o	×	×
		INTRX0 to 4, INTTX0 to 4	o	×	×
		INTSBI0 to 1	o	×	×
		INTAD/INTADHP/INTADM0 to 1	o	×	×
		INTDMAC0TC, INTDMAC1TC, INTDMAC0ERR, INTDMAC1ERR	o	×	×
		INTSSP	o	×	×
	SysTick interrupt		o	×	×
	Non-Maskable Interrupt (INTWDT)		o	×	×
	Non-Maskable Interrupt ($\overline{\text{NMI}}$ pin)		o	o	・
	RESET ($\overline{\text{RESET}}$ pin)		o	o	o

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

Note 1: After STOP2 mode is released, reset operation initializes the internal supply voltage cut off block. But back-up module is not initialized.

Note 2: For shifting to the low power consumption mode, set the CPU to prohibit all the interrupts other than the release source. If not, releasing may be executed by an unspecified interrupt.

Note 3: When STOP2 mode is released, set the active request of corresponding interrupt control register CGIMCGA,B,F to "rising edge". When "High" pulse width over than 500µs is detected, STOP2 mode is released by the timing at falling edge of the corresponding interrupt. And releasing by $\overline{\text{NMI}}$ pin, input pulse which "Low" width is over than 500µs.

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

- Release by interrupt request

  To release the low power consumption mode by an interrupt, the CPU must be set in advance to detect the interrupt. In addition to the setting in the CPU, the clock generator must be set to detect the interrupt to be used to release the STOP1 and STOP2 modes.

- Release by Non-Maskable Interrupt (NMI)

  INTWDT can only be used in the IDLE mode.

  The NMI pin can be used to release all the lower power consumption modes.

- Release by reset

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

  Note that returning to the STOP mode by reset does not induce the automatic warm-up. The reset signal as same as cold reset should be inputted.

- Release by SysTick interrupt

  SysTick interrupt can only be used in the IDLE mode.

Refer to "Interrupts" for details.

10.6.7 Warm-up

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

In the mode transition from STOP1/STOP2 to the NORMAL, 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 CGOSCCR<WUODR[11:0]> before executing the instruction to enter the STOP1/STOP2 mode.

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

Table 10-9 shows whether the warm-up setting of each mode transition is required or not.

Table 10-9 Warm-up setting in mode transition

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

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

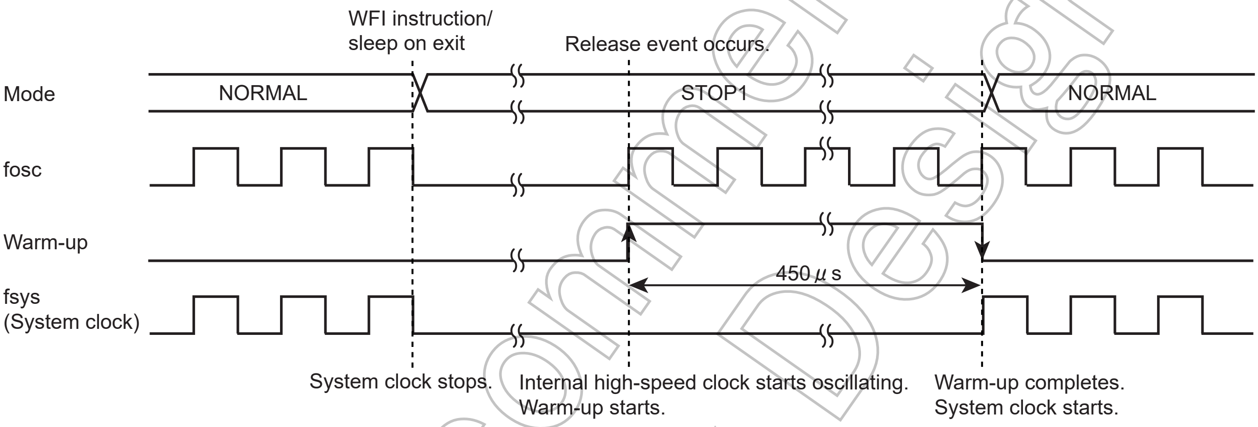
10.6.8 Clock Operations in Mode Transition

The clock operations in mode transition are described as follows.

10.6.8.1 Transition of operation modes: NORMAL → STOP1 → NORMAL

When returning to the NORMAL mode from the STOP1 mode, the warm-up is activated automatically. It is necessary for the warm-up to be set CGOSCCR<WUODR[11:0]>=0x119 in this case as a stability time (450μs) of on chip Flash ROM before entering the STOP1 mode.

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



10.6.8.2 Transition of operation modes: NORMAL → STOP2 → NORMAL

When returning to the NORMAL mode from the STOP2 mode, the warm-up is activated automatically. It is necessary for the warm-up to be set CGOSCCR<WUODR[11:0]>=0x1D4 in this case as a stability time (750μs) of on chip Flash ROM before entering the STOP2 mode.

When releasing STOP2 mode by external interrupt input pin, the active request of CGIMGGA, B, F are set to rising edge. The rising edge of the corresponding external interrupt input pin releases STOP2 mode. At the falling edge of the corresponding external interrupt input pin after keeping "High" width equal or more than 500μs, an internal oscillator starts. After the warm-up of a stability time (750μs) for on chip Flash ROM, entering to NORMAL mode.

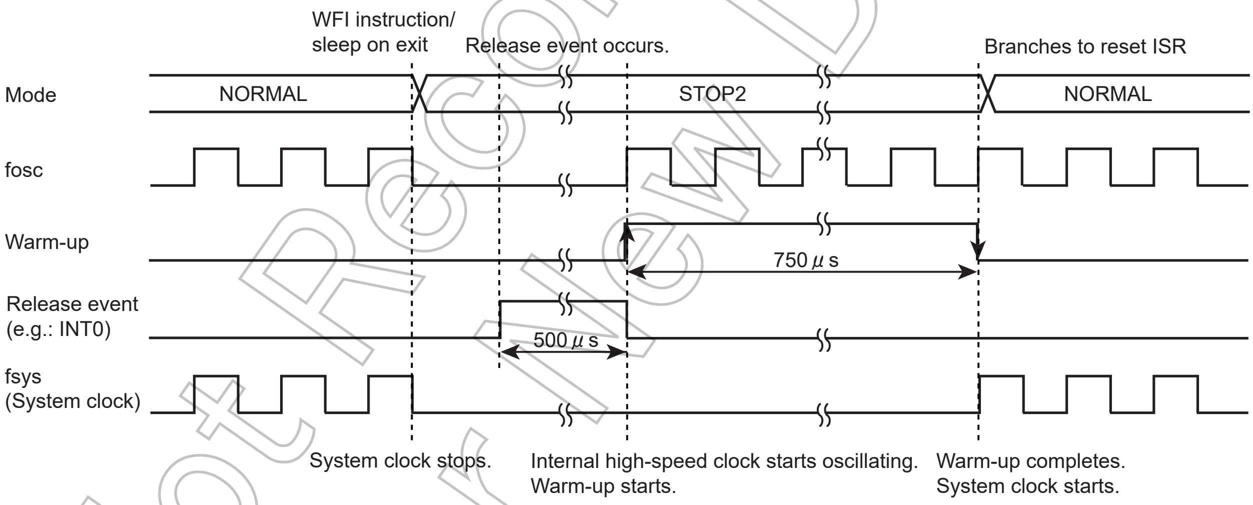
When releasing STOP2 mode by NMI, The falling edge of $\overline{\text{NMI}}$ pin releases STOP2 mode. At the rising edge of the $\overline{\text{NMI}}$ pin after keeping "Low" width equal or more than 500μs, an internal oscillator starts. After the warm-up of a stability time (750μs) for on chip Flash ROM, entering to NORMAL mode.

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

Even returning to the NORMAL mode by without reset, it would be branched to interrupt service routine of reset. After STOP2 mode is released, reset operation initializes the internal supply voltage cut off block. But back-up module is not initialized.

Note 1: When entering STOP2 mode, the level of corresponding interrupt must be "Low". And the level of $\overline{\text{NMI}}$ pin must be "High".

Note 2: When releasing STOP2 by external interrupt pin, set <PTKEEP> to "1" before entering STOP2 mode.



11. Exceptions

This chapter describes features, types and handling of exceptions.

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

11.1 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.

11.1.1 Exception Types

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

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

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

11.1.2 Handling Flowchart

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

Each step is described later in this chapter.

Processing	Description	See
Detection by CG/CPU	The CG/CPU detects the exception request.	Section 11.1.2.1
⇩		
Handling by CPU	The CPU handles the exception request.	Section 11.1.2.2
⇩		
Branch to ISR	The CPU branches to the corresponding interrupt service routine (ISR).	
⇩		
Execution of ISR	Necessary processing is executed.	Section 11.1.2.3
⇩		
Return from exception	The CPU branches to another ISR or returns to the previous program.	Section 11.1.2.4

11.1.2.1 Exception Request and Detection

(1) Exception occurrence

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

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

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

An interrupt request is generated from an external interrupt pin or peripheral function.For interrupts that are used for releasing a standby mode, relevant settings must be made in the clock generator.For details, refer to "11.5 Interrupts".

(2) Exception detection

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

Table 11-1 shows the priority of exceptions. "Configurable" means that you can assign a priority level to that exception. Memory Management, Bus Fault and Usage Fault exceptions can be enabled or disabled. If a disabled exception occurs, it is handled as Hard Fault.

Table 11-1 Exception Types and Priority

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

Note 1: **This product does not contain the MPU.**

Note 2: **External interrupts have different sources and numbers in each product. For details, see "11.5.1.5 List of Interrupt Sources".**

(3) Priority setting

- Priority levels

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

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

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

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

- Priority grouping

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

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

The Table 11-2 shows the priority group setting. The pre-emption priority and the sub priority in the table are the number in the case that <PRI_n> is defined as an 8-bit configuration.

Table 11-2 Priority grouping setting

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

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

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

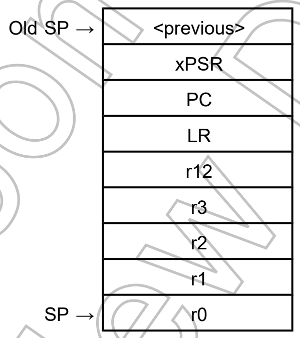
When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called "pre-emption".

(1) Stacking

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

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

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



(2) Fetching an ISR

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

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

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

(3) Late-arriving

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

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

(4) Vector table

The vector table is configured as shown below.

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

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

11.1.2.3 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 "11.5 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.

11.1.2.4 Exception exit

(1) Execution after returning from an ISR

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

- Tail-chaining

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

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

- Returning to the last stacked ISR

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

- Returning to the previous program

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

(2) Exception exit sequence

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

- Pop eight registers

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

- Load current active interrupt number

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

- Select SP

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

11.2 Reset Exceptions

Reset exceptions are generated from the following three sources.

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

- External reset pin

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

- Reset exception by WDT

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

- Reset exception by SYSRESETREQ

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

- Reset exception by OFD

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

11.3 Non-Maskable Interrupts (NMI)

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

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

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

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

- Non-maskable interrupt by WDT

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

11.4 SysTick

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

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

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

Note: In this product, fosc which is selected by CGOSCCR <OSCSEL> <EHOSCSEL> by 32 is used as external reference clock.

11.5 Interrupts

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

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

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

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

11.5.1 Interrupt Sources

11.5.1.1 Interrupt Route

Figure 11-1 shows an interrupt request route.

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

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

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

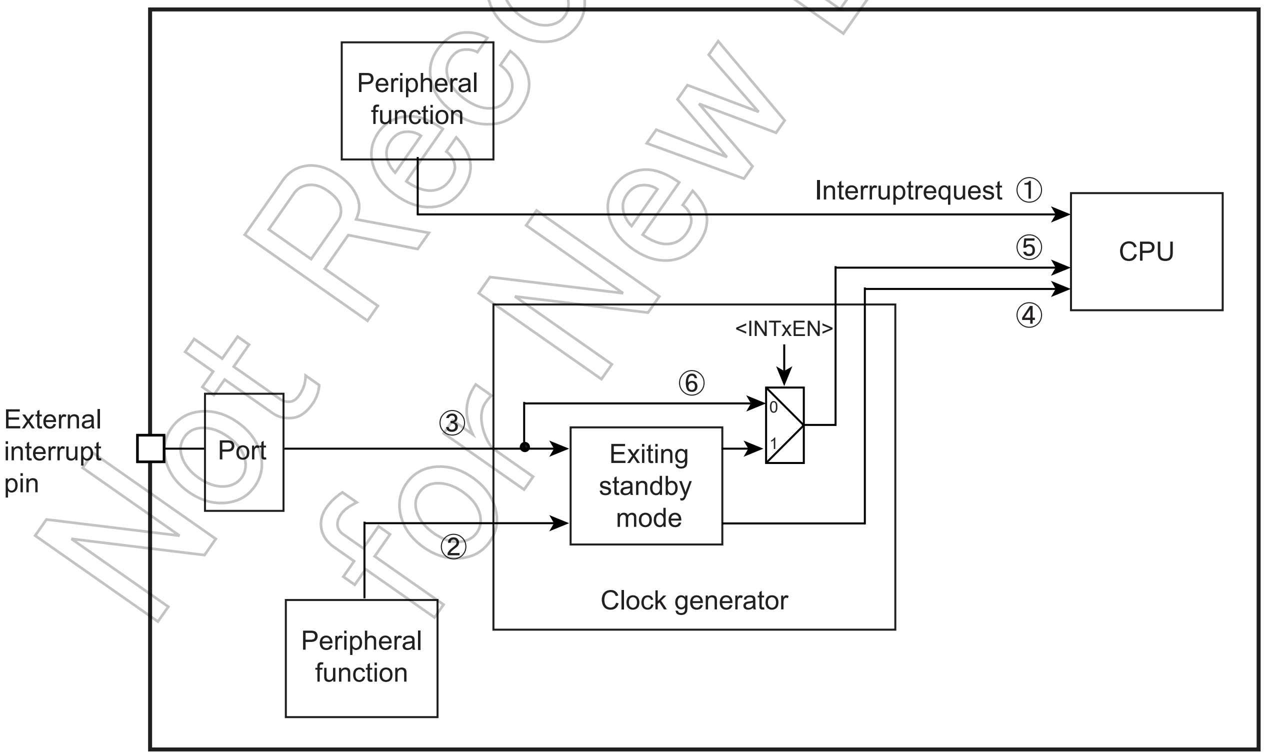


Figure 11-1 Interrupt Route

11.5.1.2 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.

11.5.1.3 Transmission

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

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

11.5.1.4 Precautions when using external interrupt pins

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

If input disabled (PxIE<PxmIE>="0"), inputs from external interrupt pins are "High". Also, if external interrupts are not used as a trigger to release standby (route 6 of "Figure 11-1 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.

11.5.1.5 List of Interrupt Sources

Table 11-3 shows the list of interrupt sources.

Table 11-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
0	INT0	Interrupt pin 0	Selectable	CGIMCGA
1	INT1	Interrupt pin 1		
2	INT2	Interrupt pin 2		
3	INT3	Interrupt pin 3		
4	INT4	Interrupt pin 4		CGIMCGB
5	INT5	Interrupt pin 5		
6	INT6	Interrupt pin 6		
7	INT7	Interrupt pin 7		
8	INTRX0	Serial channel 0 reception interrupt		
9	INTTX0	Serial channel 0 transmission interrupt		
10	INTRX1	Serial channel 1 reception interrupt		
11	INTTX1	Serial channel 1 transmission interrupt		
12	INTRX2	Serial channel 2 reception interrupt		
13	INTTX2	Serial channel 2 transmission interrupt		
14	INTSBI0	Serial bus interface 0 interrupt		
15	INTSBI1	Serial bus interface 1 interrupt		
16	INTADHP	Highest priority AD conversion complete interrupt		
17	INTAD	AD conversion completion interrupt		
18	INTADM0	AD conversion monitoring function 0 interrupt		
19	INTADM1	AD conversion monitoring function 1 interrupt		
20	INTTB0	16-bit timer / event counter 0 match detection interrupt		
21	INTTB1	16-bit timer / event counter 1 match detection interrupt		
22	INTTB2	16-bit timer / event counter 2 match detection interrupt		
23	INTTB3	16-bit timer / event counter 3 match detection interrupt		
24	INTTB4	16-bit timer / event counter 4 match detection interrupt		
25	INTTB5	16-bit timer / event counter 5 match detection interrupt		
26	INTTB6	16-bit timer / event counter 6 match detection interrupt		
27	INTTB7	16-bit timer / event counter 7 match detection interrupt		
28	INTTB8	16-bit timer / event counter 8 match detection interrupt		
29	INTTB9	16-bit timer / event counter 9 match detection interrupt		
30	INTTD0CMP0	High resolution 16-bit timer / PPG output 0 compare interrupt 0		
31	INTTD0CMP1	High resolution 16-bit timer / PPG output 0 compare interrupt 1		
32	INTTD0CMP2	High resolution 16-bit timer / PPG output 0 compare interrupt 2		
33	INTTD0CMP3	High resolution 16-bit timer / PPG output 0 compare interrupt 3		
34	INTTD0CMP4	High resolution 16-bit timer / PPG output 0 compare interrupt 4		
35	INTTD1CMP0	High resolution 16-bit timer / PPG output 1 compare interrupt 0		
36	INTTD1CMP1	High resolution 16-bit timer / PPG output 1 compare interrupt 1		
37	INTTD1CMP2	High resolution 16-bit timer / PPG output 1 compare interrupt 2		
38	INTTD1CMP3	High resolution 16-bit timer / PPG output 1 compare interrupt 3		
39	INTTD1CMP4	High resolution 16-bit timer / PPG output 1 compare interrupt 4		

Table 11-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
40	INTPHT00	2-phase pulse input counter compare interrupt 00	Rising edge	CGIMCGC
41	INTPHT01	2-phase pulse input counter compare interrupt 01		
42	INTPHT10	2-phase pulse input counter compare interrupt 10		
43	INTPHT11	2-phase pulse input counter compare interrupt 11		
44	INTPHT20	2-phase pulse input counter compare interrupt 20		CGIMCGD
45	INTPHT21	2-phase pulse input counter compare interrupt 21		
46	INTPHT30	2-phase pulse input counter compare interrupt 30		
47	INTPHT31	2-phase pulse input counter compare interrupt 31		
48	INTPHEVRY0	2-phase pulse input counter every count interrupt 0		CGIMCGE
49	INTPHEVRY1	2-phase pulse input counter every count interrupt 1		
50	INTPHEVRY2	2-phase pulse input counter every count interrupt 2		
51	INTPHEVRY3	2-phase pulse input counter every count interrupt 3		
52	INTRX3	Serial channel 3 reception interrupt		
53	INTTX3	Serial channel 3 transmission interrupt		
54	INTRX4	Serial channel 4 reception interrupt		
55	INTTX4	Serial channel 4 transmission interrupt		
56	INTCAP00	16-bit timer / event counter 0 input capture interrupt 0		
57	INTCAP01	16-bit timer / event counter 0 input capture interrupt 1		
58	INTCAP10	16-bit timer / event counter 1 input capture interrupt 0		
59	INTCAP11	16-bit timer / event counter 1 input capture interrupt 1		
60	INTCAP20	16-bit timer / event counter 2 input capture interrupt 0		
61	INTCAP21	16-bit timer / event counter 2 input capture interrupt 1		
62	INTCAP30	16-bit timer / event counter 3 input capture interrupt 0		
63	INTCAP31	16-bit timer / event counter 3 input capture interrupt 1		
64	INTCAP40	16-bit timer / event counter 4 input capture interrupt 0		
65	INTCAP41	16-bit timer / event counter 4 input capture interrupt 1		
66	INTCAP50	16-bit timer / event counter 5 input capture interrupt 0		
67	INTCAP51	16-bit timer / event counter 5 input capture interrupt 1		
68	INTCAP60	16-bit timer / event counter 6 input capture 0		
69	INTCAP61	16-bit timer / event counter 6 input capture interrupt 1		
70	INTCAP70	16-bit timer / event counter 7 input capture interrupt 0		
71	INTCAP71	16-bit timer / event counter 7 input capture interrupt 1		
72	INTCAP80	16-bit timer / event counter 8 input capture interrupt 0		
73	INTCAP81	16-bit timer / event counter 8 input capture interrupt 1		
74	INTCAP90	16-bit timer / event counter 9 input capture interrupt 0		
75	INTCAP91	16-bit timer / event counter 9 input capture interrupt 1		
76	INT8	Interrupt pin 8	Selectable	CGIMCGF
77	INT9	Interrupt pin 9		
78	INTA	Interrupt pin A		
79	INTB	Interrupt pin B		
80	INTDMACATC	DMAC unit A terminal count status interrupt		
81	INTDMACBTC	DMAC unit B terminal count status interrupt		
82	INTDMACAERR	DMAC unit A error status interrupt		
83	INTDMACBERR	DMAC unit B error status interrupt		
84	INTSSP	SSP interrupt		

11.5.1.6 Active level

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

Active level is set to the clock generator for interrupts which can be a trigger to release standby. Interrupt requests from peripheral functions are set as rising-edge or falling-edge triggered. Interrupt requests from interrupt pins can be set as level-sensitive ("High" or "Low") or edge-triggered (rising or falling).

If an interrupt source is used for clearing a standby mode, setting the relevant clock generator register is also required. Enable the CGIMCGx<INTxEN> bit and specify the active level in the CGIMCGx<EMCGx> bits. You must set the active level for interrupt requests from each peripheral function as shown in Table 11-3.

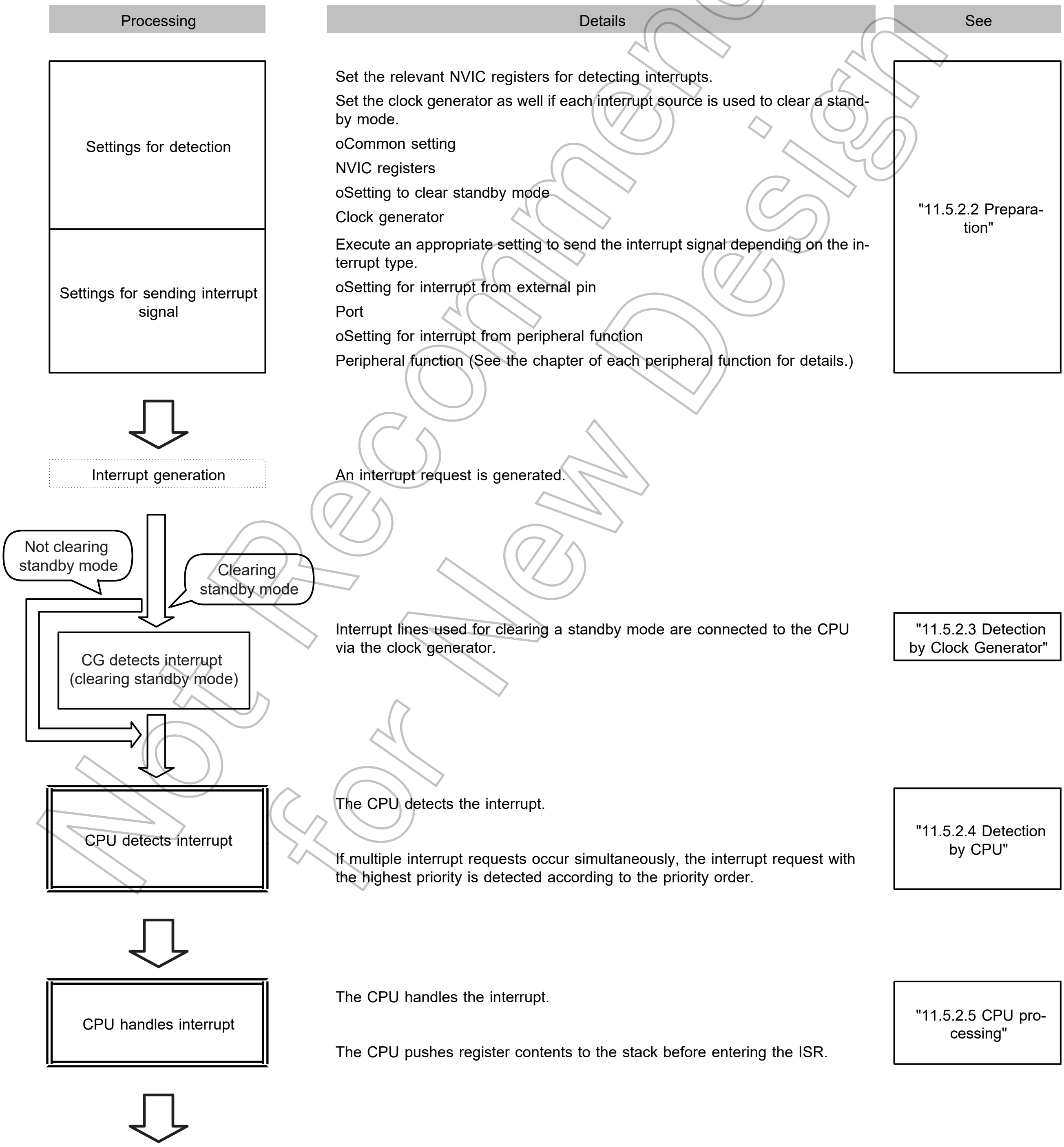
An interrupt request detected by the clock generator is notified to the CPU with a signal in "High" level.

11.5.2 Interrupt Handling

11.5.2.1 Flowchart

The following shows how an interrupt is handled.

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



Processing	Details	See
Settings for detection	Set the relevant NVIC registers for detecting interrupts. Set the clock generator as well if each interrupt source is used to clear a standby mode. oCommon setting NVIC registers oSetting to clear standby mode Clock generator	"11.5.2.2 Preparation"
Settings for sending interrupt signal	Execute an appropriate setting to send the interrupt signal depending on the interrupt type. oSetting for interrupt from external pin Port oSetting for interrupt from peripheral function Peripheral function (See the chapter of each peripheral function for details.)	
Interrupt generation	An interrupt request is generated.	
Not clearing standby mode / Clearing standby mode CG detects interrupt (clearing standby mode)	Interrupt lines used for clearing a standby mode are connected to the CPU via the clock generator.	"11.5.2.3 Detection by Clock Generator"
CPU detects interrupt	The CPU detects the interrupt. If multiple interrupt requests occur simultaneously, the interrupt request with the highest priority is detected according to the priority order.	"11.5.2.4 Detection by CPU"
CPU handles interrupt	The CPU handles the interrupt. The CPU pushes register contents to the stack before entering the ISR.	"11.5.2.5 CPU processing"

Processing	Details	See
ISR execution	Program for the ISR. Clear the interrupt source if needed.	"11.5.2.6 Interrupt Service Routine (ISR)"
⇩		
Return to preceding program	Configure to return to the preceding program of the ISR.	

11.5.2.2 Preparation

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

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

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

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

1. Disabling interrupt by CPU
2. CPU registers setting
3. Preconfiguration (1) (Interrupt from external pin)
4. Preconfiguration (2) (Interrupt from peripheral function)
5. Preconfiguration (3) (Interrupt Set-Pending Register)
6. Configuring the clock generator
7. Enabling interrupt by CPU

(1) Disabling interrupt by CPU

To make the CPU for not accepting any interrupt, write "1" to the corresponding bit of the PRIMASK Register. All interrupts and exceptions other than non-maskable interrupts and hard faults can be masked.

Use "MSR" instruction to set this register.

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

Note 1: PRIMASK register cannot be modified by the user access level.

Note 2: **If a fault causes when "1" is set to the PRIMASK register, it is treated as a hard fault.**

(2) CPU registers setting

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

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

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

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

Note: **"n" indicates the corresponding exceptions/interrupts.**
This product uses three bits for assigning a priority level.

(3) Preconfiguration (1) (Interrupt from external pin)

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

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

Note: **x: port number / m: corresponding bit / n: function register number**
In modes other than STOP mode, setting PxIE to enable input enables the corresponding interrupt input regardless of the PxFR setting. Be careful not to enable interrupts that are not used. Also, be aware of the description of "11.5.1.4 Precautions when using external interrupt pins".

(4) Preconfiguration (2) (Interrupt from peripheral function)

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

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

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

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

Note: **m: corresponding bit**

(6) Configuring the clock generator

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

Before enabling an interrupt, clear the corresponding interrupt request already held. This can avoid unexpected interrupt.To clear corresponding interrupt request, write a value corresponding to the interrupt to be used to the CGICRCG register.See "11.6.3.7 CGICRCG(CG Interrupt Request Clear Register)" for each value.

Interrupt requests from external pins can be used without setting the clock generator if they are not used for exiting a standby mode. However, an "High" pulse or "High"-level signal must be input so that the CPU can detect it as an interrupt request. Also, be aware of the description of"11.5.1.4 Precautions when using external interrupt pins".

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

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

(7) Enabling interrupt by CPU

Enable the interrupt by the CPU as shown below.

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

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

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

At the end, PRIMASK register is zero cleared.

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

Note 1: **m : corresponding bit**

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

11.5.2.3 Detection by Clock Generator

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

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

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

11.5.2.4 Detection by CPU

The CPU detects an interrupt request with the highest priority.

11.5.2.5 CPU processing

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

11.5.2.6 Interrupt Service Routine (ISR)

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

(1) Pushing during ISR

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

Push the contents of other registers if needed.

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

(2) Clearing an interrupt source

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

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

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

11.6 Exception/Interrupt-Related Registers

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

11.6.1 Register List

NVIC registers Base Address = 0xE000_E000

Register name	Address
SysTick Control and Status Register	0x0010
SysTick Reload Value Register	0x0014
SysTick Current Value Register	0x0018
SysTick Calibration Value Register	0x001C
Interrupt Set-Enable Register 1	0x0100
Interrupt Set-Enable Register 2	0x0104
Interrupt Set-Enable Register 3	0x0108
Interrupt Clear-Enable Register 1	0x0180
Interrupt Clear-Enable Register 2	0x0184
Interrupt Clear-Enable Register 3	0x0188
Interrupt Set-Pending Register 1	0x0200
Interrupt Set-Pending Register 2	0x0204
Interrupt Set-Pending Register 3	0x0208
Interrupt Clear-Pending Register 1	0x0280
Interrupt Clear-Pending Register 2	0x0284
Interrupt Clear-Pending Register 3	0x0288
Interrupt Priority Register	0x0400 to 0x0460
Vector Table Offset Register	0x0D08
Application Interrupt and Reset Control Register	0x0D0C
System Handler Priority Register	0x0D18, 0x0D1C, 0x0D20
System Handler Control and State Register	0x0D24

Clock generator registers Base Address = 0x400F_3000

Register name		Address
CG Interrupt Mode Control Register A	CGIMCGA	0x0040
CG Interrupt Mode Control Register B	CGIMCGB	0x0044
CG Interrupt Mode Control Register C	CGIMCGC	0x0048
CG Interrupt Mode Control Register D	CGIMCGD	0x004C
CG Interrupt Mode Control Register E	CGIMCGE	0x0050
CG Interrupt Mode Control Register F	CGIMCGF	0x0054
CG Interrupt Request Clear Register	CGICRCG	0x0060
Reset Flag Register	CGRSTFLG	0x0064
NMI Flag Register	CGNMIFLG	0x0068

11.6.2 NVIC Registers

11.6.2.1 SysTick Control and Status Register

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

Bit	Bit Symbol	Type	Function
31-17	−	R	Read as 0.
16	COUNTFLAG	R/W	0: Timer not counted to 0 1: Timer counted to 0 Returns "1" if timer counted to "0" since last time this was read. Clears on read of any part of the SysTick Control and Status Register.
15-3	−	R	Read as 0.
2	CLKSOURCE	R/W	0: External reference clock (fosc/32) (Note) 1: CPU clock (fsys)
1	TICKINT	R/W	0: Do not pend SysTick 1: Pend SysTick
0	ENABLE	R/W	0: Disable 1: Enable If "1" is set, it reloads with the value of the Reload Value Register and starts operation.

Note: In this product, fosc which is selected by CGOSCCR <OSCSEL> <EHOSCSEL> by 32 is used as external reference clock.

11.6.2.2 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".

11.6.2.3 SysTick Current Value Register

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

Bit	Bit Symbol	Type	Function
31-24	−	R	Read as 0.
23-0	CURRENT	R/W	[Read] Current SysTick timer value [Write] Clear Writing to this register with any value clears it to 0. Clearing this register also clears the <COUNTFLAG> bit of the SysTick Control and Status Register.

11.6.2.4 SysTick Calibration Value Register

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

Bit	Bit Symbol	Type	Function
31	NOREF	R	0: Reference clock provided 1: No reference clock
30	SKEW	R	0: Calibration value is 10 ms. 1: Calibration value is not 10 ms.
29-24	−	R	Read as 0.
23-0	TENMS	R	Calibration value Reload value to use for 10 ms timing (0xC35) by external reference clock (Note).

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

11.6.2.5 Interrupt Set-Enable Register 1

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

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

Note: **For descriptions of interrupts and interrupt numbers, see Section "11.5.1.5 List of Interrupt Sources".**

11.6.2.6 Interrupt Set-Enable Register 2

	31	30	29	28	27	26	25	24
bit symbol	SETENA (Interrupt 63)	SETENA (Interrupt 62)	SETENA (Interrupt 61)	SETENA (Interrupt 60)	SETENA (Interrupt 59)	SETENA (Interrupt 58)	SETENA (Interrupt 57)	SETENA (Interrupt 56)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	SETENA (Interrupt 55)	SETENA (Interrupt 54)	SETENA (Interrupt 53)	SETENA (Interrupt 52)	SETENA (Interrupt 51)	SETENA (Interrupt 50)	SETENA (Interrupt 49)	SETENA (Interrupt 48)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SETENA (Interrupt 47)	SETENA (Interrupt 46)	SETENA (Interrupt 45)	SETENA (Interrupt 44)	SETENA (Interrupt 43)	SETENA (Interrupt 42)	SETENA (Interrupt 41)	SETENA (Interrupt 40)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SETENA (Interrupt 39)	SETENA (Interrupt 38)	SETENA (Interrupt 37)	SETENA (Interrupt 36)	SETENA (Interrupt 35)	SETENA (Interrupt 34)	SETENA (Interrupt 33)	SETENA (Interrupt 32)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	SETENA	R/W	Interrupt number [63:32] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

Note: **For descriptions of interrupts and interrupt numbers, see Section "11.5.1.5 List of Interrupt Sources".**

11.6.2.7 Interrupt Set-Enable Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	SETENA (Interrupt 84)	SETENA (Interrupt 83)	SETENA (Interrupt 82)	SETENA (Interrupt 81)	SETENA (Interrupt 80)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SETENA (Interrupt 79)	SETENA (Interrupt 78)	SETENA (Interrupt 77)	SETENA (Interrupt 76)	SETENA (Interrupt 75)	SETENA (Interrupt 74)	SETENA (Interrupt 73)	SETENA (Interrupt 72)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SETENA (Interrupt 71)	SETENA (Interrupt 70)	SETENA (Interrupt 69)	SETENA (Interrupt 68)	SETENA (Interrupt 67)	SETENA (Interrupt 66)	SETENA (Interrupt 65)	SETENA (Interrupt 64)
After reset	0	0	0	0	0	0	0	0

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

Note: **For descriptions of interrupts and interrupt numbers, see Section "11.5.1.5 List of Interrupt Sources".**

11.6.2.8 Interrupt Clear-Enable Register 1

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

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

Note: **For descriptions of interrupts and interrupt numbers, see Section "11.5.1.5 List of Interrupt Sources".**

11.6.2.9 Interrupt Clear-Enable Register 2

	31	30	29	28	27	26	25	24
bit symbol	CLRENA (Interrupt 63)	CLRENA (Interrupt 62)	CLRENA (Interrupt 61)	CLRENA (Interrupt 60)	CLRENA (Interrupt 59)	CLRENA (Interrupt 58)	CLRENA (Interrupt 57)	CLRENA (Interrupt 56)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	CLRENA (Interrupt 55)	CLRENA (Interrupt 54)	CLRENA (Interrupt 53)	CLRENA (Interrupt 52)	CLRENA (Interrupt 51)	CLRENA (Interrupt 50)	CLRENA (Interrupt 49)	CLRENA (Interrupt 48)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CLRENA (Interrupt 47)	CLRENA (Interrupt 46)	CLRENA (Interrupt 45)	CLRENA (Interrupt 44)	CLRENA (Interrupt 43)	CLRENA (Interrupt 42)	CLRENA (Interrupt 41)	CLRENA (Interrupt 40)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CLRENA (Interrupt 39)	CLRENA (Interrupt 38)	CLRENA (Interrupt 37)	CLRENA (Interrupt 36)	CLRENA (Interrupt 35)	CLRENA (Interrupt 34)	CLRENA (Interrupt 33)	CLRENA (Interrupt 32)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	CLRENA	R/W	Interrupt number [63:32] [Write] 1: Disabled [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. It can be performed to enable interrupts and to check if interrupts are disabled. Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

Note: **For descriptions of interrupts and interrupt numbers, see Section "11.5.1.5 List of Interrupt Sources".**

11.6.2.10 Interrupt Clear-Enable Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	CLRENA (Interrupt 84)	CLRENA (Interrupt 83)	CLRENA (Interrupt 82)	CLRENA (Interrupt 81)	CLRENA (Interrupt 80)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CLRENA (Interrupt 79)	CLRENA (Interrupt 78)	CLRENA (Interrupt 77)	CLRENA (Interrupt 76)	CLRENA (Interrupt 75)	CLRENA (Interrupt 74)	CLRENA (Interrupt 73)	CLRENA (Interrupt 72)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CLRENA (Interrupt 71)	CLRENA (Interrupt 70)	CLRENA (Interrupt 69)	CLRENA (Interrupt 68)	CLRENA (Interrupt 67)	CLRENA (Interrupt 66)	CLRENA (Interrupt 65)	CLRENA (Interrupt 64)
After reset	0	0	0	0	0	0	0	0

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

Note: **For descriptions of interrupts and interrupt numbers, see Section "11.5.1.5 List of Interrupt Sources".**

11.6.2.11 Interrupt Set-Pending Register 1

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

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

Note: **For descriptions of interrupts and interrupt numbers, see Section "11.5.1.5 List of Interrupt Sources".**

11.6.2.12 Interrupt Set-Pending Register 2

	31	30	29	28	27	26	25	24
bit symbol	SETPEND (Interrupt 63)	SETPEND (Interrupt 62)	SETPEND (Interrupt 61)	SETPEND (Interrupt 60)	SETPEND (Interrupt 59)	SETPEND (Interrupt 58)	SETPEND (Interrupt 57)	SETPEND (Interrupt 56)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	SETPEND (Interrupt 55)	SETPEND (Interrupt 54)	SETPEND (Interrupt 53)	SETPEND (Interrupt 52)	SETPEND (Interrupt 51)	SETPEND (Interrupt 50)	SETPEND (Interrupt 49)	SETPEND (Interrupt 48)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	SETPEND (Interrupt 47)	SETPEND (Interrupt 46)	SETPEND (Interrupt 45)	SETPEND (Interrupt 44)	SETPEND (Interrupt 43)	SETPEND (Interrupt 42)	SETPEND (Interrupt 41)	SETPEND (Interrupt 40)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	SETPEND (Interrupt 39)	SETPEND (Interrupt 38)	SETPEND (Interrupt 37)	SETPEND (Interrupt 36)	SETPEND (Interrupt 35)	SETPEND (Interrupt 34)	SETPEND (Interrupt 33)	SETPEND (Interrupt 32)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-0	SETPEND	R/W	Interrupt number [63:32] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts. Clear and Interrupt Set-Pending Register bit by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.

Note: **For descriptions of interrupts and interrupt numbers, see Section "11.5.1.5 List of Interrupt Sources".**

11.6.2.13 Interrupt Set-Pending 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	-	-	-	SETPEND (Interrupt 84)	SETPEND (Interrupt 83)	SETPEND (Interrupt 82)	SETPEND (Interrupt 81)	SETPEND (Interrupt 80)
After reset	0	0	0	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	SETPEND (Interrupt 79)	SETPEND (Interrupt 78)	SETPEND (Interrupt 77)	SETPEND (Interrupt 76)	SETPEND (Interrupt 75)	SETPEND (Interrupt 74)	SETPEND (Interrupt 73)	SETPEND (Interrupt 72)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	SETPEND (Interrupt 71)	SETPEND (Interrupt 70)	SETPEND (Interrupt 69)	SETPEND (Interrupt 68)	SETPEND (Interrupt 67)	SETPEND (Interrupt 66)	SETPEND (Interrupt 65)	SETPEND (Interrupt 64)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

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

Note: **For descriptions of interrupts and interrupt numbers, see Section "11.5.1.5 List of Interrupt Sources".**

11.6.2.14 Interrupt Clear-Pending Register 1

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

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

Note: **For descriptions of interrupts and interrupt numbers, see Section "11.5.1.5 List of Interrupt Sources".**

11.6.2.15 Interrupt Clear-Pending Register 2

	31	30	29	28	27	26	25	24
bit symbol	CLRPEND (Interrupt 63)	CLRPEND (Interrupt 62)	CLRPEND (Interrupt 61)	CLRPEND (Interrupt 60)	CLRPEND (Interrupt 59)	CLRPEND (Interrupt 58)	CLRPEND (Interrupt 57)	CLRPEND (Interrupt 56)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	CLRPEND (Interrupt 55)	CLRPEND (Interrupt 54)	CLRPEND (Interrupt 53)	CLRPEND (Interrupt 52)	CLRPEND (Interrupt 51)	CLRPEND (Interrupt 50)	CLRPEND (Interrupt 49)	CLRPEND (Interrupt 48)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	CLRPEND (Interrupt 47)	CLRPEND (Interrupt 46)	CLRPEND (Interrupt 45)	CLRPEND (Interrupt 44)	CLRPEND (Interrupt 43)	CLRPEND (Interrupt 42)	CLRPEND (Interrupt 41)	CLRPEND (Interrupt 40)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	CLRPEND (Interrupt 39)	CLRPEND (Interrupt 38)	CLRPEND (Interrupt 37)	CLRPEND (Interrupt 36)	CLRPEND (Interrupt 35)	CLRPEND (Interrupt 34)	CLRPEND (Interrupt 33)	CLRPEND (Interrupt 32)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31-0	CLRPEND	R/W	Interrupt number [63:32] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts.

Note: **For descriptions of interrupts and interrupt numbers, see Section "11.5.1.5 List of Interrupt Sources".**

11.6.2.16 Interrupt Clear-Pending 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	-	-	-	CLRPEND (Interrupt 84)	CLRPEND (Interrupt 83)	CLRPEND (Interrupt 82)	CLRPEND (Interrupt 81)	CLRPEND (Interrupt 80)
After reset	0	0	0	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	CLRPEND (Interrupt 79)	CLRPEND (Interrupt 78)	CLRPEND (Interrupt 77)	CLRPEND (Interrupt 76)	CLRPEND (Interrupt 75)	CLRPEND (Interrupt 74)	CLRPEND (Interrupt 73)	CLRPEND (Interrupt 72)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	CLRPEND (Interrupt 71)	CLRPEND (Interrupt 70)	CLRPEND (Interrupt 69)	CLRPEND (Interrupt 68)	CLRPEND (Interrupt 67)	CLRPEND (Interrupt 66)	CLRPEND (Interrupt 65)	CLRPEND (Interrupt 64)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

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

Note: **For descriptions of interrupts and interrupt numbers, see Section "11.5.1.5 List of Interrupt Sources".**

11.6.2.17 Interrupt Priority Register

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

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

	31 24	23 16	15 8	7 0
0xE000_E400	PRI_3	PRI_2	PRI_1	PRI_0
0xE000_E404	PRI_7	PRI_6	PRI_5	PRI_4
0xE000_E408	PRI_11	PRI_10	PRI_9	PRI_8
0xE000_E40C	PRI_15	PRI_14	PRI_13	PRI_12
0xE000_E410	PRI_19	PRI_18	PRI_17	PRI_16
0xE000_E414	PRI_23	PRI_22	PRI_21	PRI_20
0xE000_E418	PRI_27	PRI_26	PRI_25	PRI_24
0xE000_E41C	PRI_31	PRI_30	PRI_29	PRI_28
0xE000_E420	PRI_35	PRI_34	PRI_33	PRI_32
0xE000_E424	PRI_39	PRI_38	PRI_37	PRI_36
0xE000_E428	PRI_43	PRI_42	PRI_41	PRI_40
0xE000_E42C	PRI_47	PRI_46	PRI_45	PRI_44
0xE000_E430	PRI_51	PRI_50	PRI_49	PRI_48
0xE000_E434	PRI_55	PRI_54	PRI_53	PRI_52
0xE000_E438	PRI_59	PRI_58	PRI_57	PRI_56
0xE000_E43C	PRI_63	PRI_62	PRI_61	PRI_60
0xE000_E440	PRI_67	PRI_66	PRI_65	PRI_64
0xE000_E444	PRI_71	PRI_70	PRI_69	PRI_68
0xE000_E448	PRI_75	PRI_74	PRI_73	PRI_72
0xE000_E44C	PRI_79	PRI_78	PRI_77	PRI_76
0xE000_E450	PRI_83	PRI_82	PRI_81	PRI_80
0xE000_E454	Reserved	Reserved	Reserved	PRI_84

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

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

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

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

11.6.2.18 Vector Table Offset Register

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

Bit	Bit Symbol	Type	Function
31-30	−	R	Read as 0.
29	TBLBASE	R/W	Table base The vector table is in: 0: Code space 1: SRAM space
28-7	TBLOFF	R/W	Offset value Set the offset value from the top of the space specified in TBLBASE. The offset must be aligned based on the number of exceptions in the table.This means that the minimum alignment is 32 words that you can use for up to 16 interrupts.For more interrupts, you must adjust the alignment by rounding up to the next power of two.
6-0	−	R	Read as 0.

11.6.2.19 Application Interrupt and Reset Control Register

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

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

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

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

11.6.2.20 System Handler Priority Register

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

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

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

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

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

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

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

11.6.2.21 System Handler Control and State Register

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

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

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

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

11.6.3 Clock generator registers

11.6.3.1 CGIMCGA(CG Interrupt Mode Control Register A)

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

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

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

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

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

11.6.3.2 CGIMCGB(CG Interrupt Mode Control Register B)

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

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

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

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

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

11.6.3.3 CGIMCGC(CG Interrupt Mode Control Register C)

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

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-28	EMCGB[2:0]	R/W	active level setting of INTPHT11 standby clear request. **Set it as shown below.** 011: Rising edge
27-26	EMSTB[1:0]	R	Reads as undefined.
25	−	R	Reads as undefined.
24	INTBEN	R/W	INTPHT11 clear input 0:Disable 1: Enable
23	−	R	Read as 0.
22-20	EMCGA[2:0]	R/W	active level setting of INTPHT10 standby clear request. **Set it as shown below.** 011: Rising edge
19-18	EMSTA[1:0]	R	Reads as undefined.
17	−	R	Reads as undefined.
16	INTAEN	R/W	INTPHT10 clear input 0:Disable 1: Enable
15	−	R	Read as 0.
14-12	EMCG9[2:0]	R/W	active level setting of INTPHT01 standby clear request. **Set it as shown below.** 011: Rising edge
11-10	EMST9[1:0]	R	Reads as undefined.
9	−	R	Reads as undefined.
8	INT9EN	R/W	INTPHT01 clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCG8[2:0]	R/W	active level setting of INTPHT00 standby clear request. **Set it as shown below.** 011: Rising edge
3-2	EMST8[1:0]	R	Reads as undefined.
1	−	R	Reads as undefined.
0	INT8EN	R/W	INTPHT00 clear input 0:Disable 1: Enable

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

11.6.3.4 CGIMCGD(CG Interrupt Mode Control Register D)

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

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-28	EMCGF[2:0]	R/W	active level setting of INTPHT31 standby clear request. **Set it as shown below.** 011: Rising edge
27-26	EMSTF[1:0]	R	Reads as undefined.
25	−	R	Reads as undefined.
24	INTFEN	R/W	INTPHT31 clear input 0:Disable 1: Enable
23	−	R	Read as 0.
22-20	EMCGE[2:0]	R/W	active level setting of INTPHT30 standby clear request. **Set it as shown below.** 011: Rising edge
19-18	EMSTE[1:0]	R	Reads as undefined.
17	−	R	Reads as undefined.
16	INTEEN	R/W	INTPHT30 clear input 0:Disable 1: Enable
15	−	R	Read as 0.
14-12	EMCGD[2:0]	R/W	active level setting of INTPHT21 standby clear request. **Set it as shown below.** 011: Rising edge
11-10	EMSTD[1:0]	R	Reads as undefined.
9	−	R	Reads as undefined.
8	INTDEN	R/W	INTPHT21 clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCGC[2:0]	R/W	active level setting of INTPHT20 standby clear request. **Set it as shown below.** 011: Rising edge
3-2	EMSTC[1:0]	R	Reads as undefined.
1	−	R	Reads as undefined.

Bit	Bit Symbol	Type	Function
0	INTCEN	R/W	INTPHT20 clear input 0:Disable 1: Enable

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

11.6.3.5 CGIMCGE(CG Interrupt Mode Control Register E)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCG13			EMST13		-	INT13EN
After reset	0	0	1	0	0	0	Undefined	0
	23	22	21	20	19	18	17	16
bit symbol	-	EMCG12			EMST12		-	INT12EN
After reset	0	0	1	0	0	0	Undefined	0
	15	14	13	12	11	10	9	8
bit symbol	-	EMCG11			EMST11		-	INT11EN
After reset	0	0	1	0	0	0	Undefined	0
	7	6	5	4	3	2	1	0
bit symbol	-	EMCG10			EMST10		-	INT10EN
After reset	0	0	1	0	0	0	Undefined	0

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-28	EMCG13[2:0]	R/W	active level setting of INTPHEVRY3 standby clear request. **Set it as shown below.** 011: Rising edge
27-26	EMST13[1:0]	R	Reads as undefined.
25	−	R	Reads as undefined.
24	INT13EN	R/W	INTPHEVRY3 clear input 0:Disable 1: Enable
23	−	R	Read as 0.
22-20	EMCG12[2:0]	R/W	active level setting of INTPHEVRY2 standby clear request. **Set it as shown below.** 011: Rising edge
19-18	EMST12[1:0]	R	Reads as undefined.
17	−	R	Reads as undefined.
16	INT12EN	R/W	INTPHEVRY2 clear input 0:Disable 1: Enable
15	−	R	Read as 0.
14-12	EMCG11[2:0]	R/W	active level setting of INTPHEVRY1 standby clear request. **Set it as shown below.** 011: Rising edge
11-10	EMST11[1:0]	R	Reads as undefined.
9	−	R	Reads as undefined.
8	INT11EN	R/W	INTPHEVRY1 clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCG10[2:0]	R/W	active level setting of INTPHEVRY0 standby clear request. **Set it as shown below.** 011: Rising edge
3-2	EMST10[1:0]	R	Reads as undefined.
1	−	R	Reads as undefined.

Bit	Bit Symbol	Type	Function
0	INT10EN	R/W	INTPHEVRY0 clear input 0:Disable 1: Enable

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

11.6.3.6 CGIMCGF(CG Interrupt Mode Control Register F)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCG17			EMST17		-	INT17EN
After reset	0	0	1	0	0	0	Undefined	0
	23	22	21	20	19	18	17	16
bit symbol	-	EMCG16			EMST16		-	INT16EN
After reset	0	0	1	0	0	0	Undefined	0
	15	14	13	12	11	10	9	8
bit symbol	-	EMCG15			EMST15		-	INT15EN
After reset	0	0	1	0	0	0	Undefined	0
	7	6	5	4	3	2	1	0
bit symbol	-	EMCG14			EMST14		-	INT14EN
After reset	0	0	1	0	0	0	Undefined	0

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-28	EMCG17[2:0]	R/W	active level setting of INTB standby clear request. (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
27-26	EMST17[1:0]	R	active level of INTB standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
25	−	R	Reads as undefined.
24	INT17EN	R/W	INTB clear input 0:Disable 1: Enable
23	−	R	Read as 0.
22-20	EMCG16[2:0]	R/W	active level setting of INTA standby clear request. (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
19-18	EMST16[1:0]	R	active level of INTA standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
17	−	R	Reads as undefined.
16	INT16EN	R/W	INTA clear input 0:Disable 1: Enable
15	−	R	Read as 0.
14-12	EMCG15[2:0]	R/W	active level setting of INT9 standby clear request. (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges

Bit	Bit Symbol	Type	Function
11-10	EMST15[1:0]	R	active level of INT9 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
9	−	R	Reads as undefined.
8	INT15EN	R/W	INT9 clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCG14[2:0]	R/W	active level setting of INT8 standby clear request. (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
3-2	EMST14[1:0]	R	active level of INT8 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
1	−	R	Reads as undefined.
0	INT14EN	R/W	INT8 clear input 0:Disable 1: Enable

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

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

Note: Be sure to set active state of the clear request if interrupt is enabled for clearing the STOP1/ STOP2/ IDLE standby modes.

1. When using interrupt, be sure to follow the sequence of actions shown below:
   - If shared with other general ports, enable the port to receive the interrupt.
   - Setup conditions such as active state when initializing.
   - Clear interrupt requests.
   - Enable interrupts.

2. Each setup must be done while interrupt is in a disable condition.

3. As an interrupt of STOP1 release, it is available to set 24 factors of INT0 to INT7, INT8 to INTB, INTPHT00 to INTPHT31 and INTPHEVRY0 to INTPHEVRY3 interrupts. It is set by CG whether to be used for STOP1 release interrupt, and the edge/level when in active condition.

4. As an interrupt of STOP2 release, it is available to set 12 factors of IINT0 to INT7 and INT8 to INTB interrupts. Set "rising edge" CG whether to be used for STOP2 release interrupt, and the setup active condition.

5. In the assigned factors to STOP1/STOP2/IDLE standby modes release request interrupt, INT0 to INT7 and INT8 to INTB are available to be used as normal interrupt without setting CG.

11.6.3.7 CGICRCG(CG Interrupt Request Clear Register)

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

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4-0	ICRCG[4:0]	W	Clear interrupt requests. 0_0000: INT0 0_1000: INTPHT00 1_0000: INTPHEVRY0 1_1000 to 1_1111: setting prohibited. 0_0001: INT1 0_1001: INTPHT01 1_0001: INTPHEVRY1 0_0010: INT2 0_1010: INTPHT10 1_0010: INTPHEVRY2 0_0011: INT3 0_1011: INTPHT11 1_0011: INTPHEVRY3 0_0100: INT4 0_1100: INTPHT20 1_0100: INT8 0_0101: INT5 0_1101: INTPHT21 1_0101: INT9 0_0110: INT6 0_1110: INTPHT30 1_0110: INTA 0_0111: INT7 0_1111: INTPHT31 1_0111: INTB Read as 0.

11.6.3.8 CGNMIFLG(NMI Flag Register)

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

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

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

11.6.3.9 CGRSTFLG (Reset Flag Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After pin reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After pin reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After pin reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	OFDRSTF	DBGRSTF	STOP2RSTF	WDTRSTF	-	PINRSTF
After pin reset	0	0	0	0	0	0	Undefined	1

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as 0.
5	OFDRSTF	R/W	OFD reset flag 0: "0" is written 1: Reset from OFD
4	DBGRSTF	R/W	Debug reset flag (Note1) 0: "0" is written 1: Reset from SYSRESETREQ
3	STOP2RSTF	R/W	STOP2 reset flag 0: "0" is written 1: Reset flag by STOP2 mode release
2	WDTRSTF	R/W	WDT reset flag 0: "0" is written 1: Reset from WDT
1	−	R/W	Read as undefined. Write as 0.
0	PINRSTF	R/W	$\overline{\text{RESET}}$ pin flag 0: "0" is written 1: Reset from $\overline{\text{RESET}}$ pin.

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

Note:**This product is initialized by the external reset.**

12. Input/Output Ports

12.1 Port Functions

12.1.1 Function Lists

TMPM341FDXBG/FYXBG has 86 ports. Besides the ports function, these ports can be used as I/O pins for peripheral functions.

Table 12-1, Table 12-2 and Table 12-3 show the port function table.

Table 12-1 Port Function List (Port A to Port C)

Port	Pin	Input/ Output	Programmable Pull-up Pull-down	Schmitt input	Noise Filter	Program- mable Open-drain	Function pin
Port A	PA0	I/O	Pull-up	−	−	o	D0/AD0
	PA1	I/O	Pull-up	−	−	o	D1/AD1
	PA2	I/O	Pull-up	−	−	o	D2/AD2
	PA3	I/O	Pull-up	−	−	o	D3/AD3
	PA4	I/O	Pull-up	−	−	o	D4/AD4
	PA5	I/O	Pull-up	−	−	o	D5/AD5
	PA6	I/O	Pull-up	−	−	o	D6/AD6
	PA7	I/O	Pull-up	−	−	o	D7/AD7
Port B	PB0	I/O	Pull-up	−	−	o	D8/AD8/A0
	PB1	I/O	Pull-up	−	−	o	D9/AD9/A1
	PB2	I/O	Pull-up	−	−	o	D10/AD10/A2
	PB3	I/O	Pull-up	−	−	o	D11/AD11/A3
	PB4	I/O	Pull-up	−	−	o	D12/AD12/A4
	PB5	I/O	Pull-up	−	−	o	D13/AD13/A5
	PB6	I/O	Pull-up	−	−	o	D14/AD14/A6
	PB7	I/O	Pull-up	−	−	o	D15/AD15/A7
Port C	PC0	I/O	Pull-up	o	−	o	A0/TXD1/TB2IN0
	PC1	I/O	Pull-up	o	−	o	A1/RXD1/TB2IN1
	PC2	I/O	Pull-up	o	−	o	A2/SCLK1/TB0OUT/$\overline{\text{CTS1}}$
	PC3	I/O	Pull-up	o	o(INT2 only)	o	A3/INT2/TB1OUT
	PC4	I/O	Pull-up	o	−	o	A4/TXD2/TB3IN0
	PC5	I/O	Pull-up	o	−	o	A5/RXD2/TB3IN1
	PC6	I/O	Pull-up	o	−	o	A6/SCLK2/TB4IN0/$\overline{\text{CTS2}}$
	PC7	I/O	Pull-up	o	o(INT3 only)	o	A7/INT3/TB4IN1

o : Exist

- : Not exist

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

Table 12-2 Port Function List (Port D to Port G)

Port	Pin	Input/Out-put	Programmable Pull-up Pull-down	Schmitt Input	Noise Filter	Program-mable Open-drain	Function pin
Port D	PD0	I/O	Pull-up	o	−	o	A8/SO1/SDA1/TB7OUT
	PD1	I/O	Pull-up	o	−	o	A9/SI1/SCL1/TB8OUT
	PD2	I/O	Pull-up	o	−	o	A10/SCK1/TB9OUT
	PD3	I/O	Pull-up	o	o(INT4 only)	o	A11/INT4/$\overline{\text{ADTRG}}$
	PD4	I/O	Pull-up	o	−	o	A12/SPDO
	PD5	I/O	Pull-up	o	−	o	A13/SPDI
	PD6	I/O	Pull-up	o	−	o	A14/SPCLK
	PD7	I/O	Pull-up	o	−	o	A15/SPFSS/SCOUT
Port E	PE0	I/O	Pull-up	o	−	o	TXD0/A16
	PE1	I/O	Pull-up	o	−	o	RXD0/A17
	PE2	I/O	Pull-up	o	−	o	SCLK0/A18/TB2OUT/ $\overline{\text{CTS0}}$
	PE3	I/O	Pull-up	o	o(INT5 only)	o	INT5/A19/TB3OUT
	PE4	I/O	Pull-up	o	−	o	A20/TD0OUT0
	PE5	I/O	Pull-up	o	−	o	A21/TD0OUT1
	PE6	I/O	Pull-up	o	−	o	A22/TD1OUT0
	PE7	I/O	Pull-up	o	−	o	A23/TD1OUT1
Port F	PF0	Output	After Reset, Pull-up	o	−	o	$\overline{\text{BOOT}}$/TB6OUT
	PF1	I/O	Pull-up	o	−	o	$\overline{\text{RD}}$
	PF2	I/O	Pull-up	o	−	o	$\overline{\text{WR}}$
	PF3	I/O	Pull-up	o	−	o	$\overline{\text{BELL}}$
	PF4	I/O	Pull-up	o	o(INT6 only)	o	$\overline{\text{BELH}}$/INT6/TB5IN0
	PF5	I/O	Pull-up	o	o(INT7 only)	o	$\overline{\text{CS1}}$/INT7/TB5IN1
	PF6	I/O	Pull-up	o	−	o	$\overline{\text{CS0}}$
	PF7	I/O	Pull-up	o	−	o	ALE
Port G	PG0	I/O	Pull-up	o	−	o	SO0/SDA0/TB7IN0
	PG1	I/O	Pull-up	o	−	o	SI0/SCL0/TB7IN1
	PG2	I/O	Pull-up	o	o(INT8 only)	o	SCK0/INT8
	PG3	I/O	Pull-up	o	o	o	INT0
	PG4	I/O	Pull-up	o	−	o	TXD3/TB8IN0
	PG5	I/O	Pull-up	o	−	o	RXD3/TB8IN1
	PG6	I/O	Pull-up	o	−	o	SCLK3/TB9IN0/$\overline{\text{CTS3}}$
	PG7	I/O	Pull-up	o	o(INT1 only)	o	INT1/TB9IN1

o : Exist
- : Not exist

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

Table 12-3 Port Function List (Port H to Port K)

Port	Pin	Input/Output	Programmable Pull-up Pull-down	Schmitt input	Noise Filter	Programmable Open-drain	Function pin
Port H	PH0	I/O	Pull-up	o	−	o	TXD4
	PH1	I/O	Pull-up	o	−	o	RXD4
	PH2	I/O	Pull-up	o	−	o	SCLK4/$\overline{\text{CTS4}}$
	PH3	I/O	Pull-up	o	−	o	PHC3IN0/TB4OUT
	PH4	I/O	Pull-up	o	−	o	PHC3IN1/TB5OUT
	PH5	I/O	Pull-up	o	−	o	TRACEDATA3
	PH6	I/O	Pull-up	o	−	o	TRACEDATA2
Port I	PI0	I/O	Pull-up	o	−	o	TRACEDATA1
	PI1	I/O	Pull-up	o	−	o	TRACEDATA0
	PI2	I/O	Pull-up	o	−	o	TRACECLK
	PI3	I/O	After Reset, Pull-up	o	o	−	$\overline{\text{TRST}}$
	PI4	I/O	After Reset, Pull-up	o	−	−	TDI
	PI5	I/O	After Reset, Pull-down	o	−	−	TCK/SWCLK
	PI6	I/O	After Reset, Pull-up	o	−	−	TMS/SWDIO
	PI7	I/O	Pull-up	o	−	−	TDO/SWV
Port J	PJ0	I/O	Pull-up	o	−	−	AIN00/PHC0IN0
	PJ1	I/O	Pull-up	o	−	−	AIN01/PHC0IN1
	PJ2	I/O	Pull-up	o	−	−	AIN02/PHC1IN0
	PJ3	I/O	Pull-up	o	−	−	AIN03/PHC1IN1
	PJ4	I/O	Pull-up	o	−	−	AIN04/PHC2IN0
	PJ5	I/O	Pull-up	o	−	−	AIN05/PHC2IN1
	PJ6	I/O	Pull-up	o	−	−	AIN06/TB0IN0
	PJ7	I/O	Pull-up	o	o(INT9 only)	−	AIN07/INT9/TB0IN1
Port K	PK0	I/O	Pull-up	o	−	−	AIN08/TB1IN0
	PK1	I/O	Pull-up	o	o(INTA only)	−	AIN09/INTA/TB1IN1
	PK2	I/O	Pull-up	o	−	−	AIN10/TB6IN0
	PK3	I/O	Pull-up	o	o(INTB only)	−	AIN11/INTB/TB6IN1
	PK4	I/O	Pull-up	o	−	−	AIN12
	PK5	I/O	Pull-up	o	−	−	AIN13
	PK6	I/O	Pull-up	o	−	−	AIN14

o : Exist

- : Not exist

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

12.1.2 Port Registers Outline

The following registers need to be configured to use ports.

- PxDATA: Port x data register

  To read/ write port data.

- PxCR: Port x output control register

  To control output.

  PxIE needs to be configured to control input.

- PxFRn: Port x function register n

  To set functions.

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

- PxOD: Port x open drain control register

  To control the programmable open drain.

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

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

- PxPUP: Port x pull-up control register

  To control program pull ups.

- PxPDN: Port x pull-down control register

  To control programmable pull downs.

- PxIE: Port x input control register

  To control inputs.

  For avoided through current, default setting prohibits inputs.

12.1.3 Port States in STOP Mode

On the one hand, input and output in STOP1 mode are enabled/disabled by the CGSTBYCR<DRVE> bit; on the other hand input and output in STOP2 mode are enabled/disabled by the CGSTBYCR<PTKEEP> bit.

If PxIE or PxCR is enabled with <DRVE>=1 or <PTKEEP>="0" → "1", input or output is enabled respectively in STOP1/STOP2 mode. If <DRVE>=0, both input and output are disabled in STOP1 mode except for some ports even if PxIE or PxCR are enabled. However, to transmit from Normal mode to STOP2 mode, the <PTKEEP> bit need to be set "0" to "1".

Table 12-4 shows the pin conditions in STOP mode.

Table 12-4 Port conditions in STOP mode

Fanction	Pin Name	I/O	STOP1		STOP2	
			<DRVE> = 0	<DRVE> = 1	<PTKEEP> = 0	<PTKEEP> = 1
Control Ports	$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, MODE, INTLV, BSC, ENDIAN	Input	o	o	o	o
Oscillation Circuit	X1/EHCLKIN	Input	×	×	×	×
	X2	Output	"High" Level Output.		×	×
Special Ports	DA0, DA1	Output	×	×	×	×
Port	PJ0 to PJ5 (PHC0IN0, PHC0IN1, PHC1IN0, PHC1IN1, PHC2IN0, PHC2IN1) (Functional pin configured: PxFRn<PxmFn>="1")	Input	o	o	o	o
	PI3 to PI5 ($\overline{\text{TRST}}$, TDI, SWCLK/TCK) (Debug I/F configured: PxFRn<PxmFn>="1")	Input	Depends on PxIE[m].		o	Input mode is sustained by PxIE[m].
	PI6 (SWDIO/TMS) (Debug I/F configured: PxFRn<PxmFn>="1")	Input	Depends on PxIE[m].		o	Input mode is sustained by PxIE[m].
		Output	Enabled when data is valid. Disabled when data is invalid.		o	Output mode is sustained by PxCR[m].
	PI7, PI2, PI1, PI0, PH6, PH5 (TDO/SWV, TRACECLK, TRACEDATA0 to 3) (Debug I/F configured: PxFRn<PxmFn>="1")	Output	Depends on PxCR[m].		Depends on PxCR[m].	Output mode is sustained by PxCR[m].
	PG3, PG7, PC3, PC7, PD3, PE3, PF4, PF5, PG2, PJ7, PK1, PK3 (INT0 to B) (Interruption function configured: PxFRn<PxmFn>="1" and PxIE<PxmIE>="1")	Input	o	o	o	o
	Other ports	Input	×	Depends on PxIE[m].	×	Input mode is sustained by PxIE[m].
		Output	×	Depends on PxIE[m].	×	Output mode is sustained by PxCR[m].

o :Input or output enabled

× :Input or output disabled

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

12.2 Port functions

This chapter describes the port registers detail.

This chapter describes only "circuit type" reading circuit configuration.For detailed circuit diagram, refer to "12.3 Block Diagrams of Ports".

12.2.1 Port A (PA0 to PA7)

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

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

The port A has one types of function register. If you use the port A as a general-purpose port, set "0" to the corresponding bit of the two registers. If you use the port A as other than a general-purpose port, set "1" to the corresponding bit of the function register.

12.2.1.1 Port A register

Base Address = 0x400C_0000

Register name		Address (Base+)
Port A data register	PADATA	0x0000
Port A output control register	PACR	0x0004
Port A function register 1	PAFR1	0x0008
Reserved	-	0x000C
Reserved	-	0x0010
Reserved	-	0x0014
Reserved	-	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port A open drain control register	PAOD	0x0028
Port A pull-up control register	PAPUP	0x002C
Reserved	-	0x0030
Reserved	-	0x0034
Port A input control register	PAIE	0x0038

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

12.2.1.2 PADATA (Port A data register)

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

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

12.2.1.3 PACR (Port A output control register)

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

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

12.2.1.4 PAFR1 (Port A function register 1)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PA7F1	R/W	0: PORT 1: D7/AD7
6	PA6F1	R/W	0: PORT 1: D6/AD6
5	PA5F1	R/W	0: PORT 1: D5/AD5
4	PA4F1	R/W	0: PORT 1: D4/AD4
3	PA3F1	R/W	0: PORT 1: D3/AD3
2	PA2F1	R/W	0: PORT 1: D2/AD2
1	PA1F1	R/W	0: PORT 1: D1/AD1
0	PA0F1	R/W	0: PORT 1: D0/AD0

12.2.1.5 PAOD (Port A open drain control register)

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

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

12.2.1.6 PAPUP (Port A pull-up control register)

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

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

12.2.1.7 PAIE (Port A input control register)

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

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

12.2.2 Port B (PB0 to PB7)

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

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

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

12.2.2.1 Port B Register

Base Address = 0x400C_0100

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

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

12.2.2.2 PBDATA (Port B data register)

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

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

12.2.2.3 PBCR (Port B output control register)

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

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

12.2.2.4 PBFR1 (Port B function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PB7F1	PB6F1	PB5F1	PB4F1	PB3F1	PB2F1	PB1F1	PB0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PB7F1	R/W	0: PORT 1: D15/AD15
6	PB6F1	R/W	0: PORT 1: D14/AD14
5	PB5F1	R/W	0: PORT 1: D13/AD13
4	PB4F1	R/W	0: PORT 1: D12/AD12
3	PB3F1	R/W	0: PORT 1: D11/AD11
2	PB2F1	R/W	0: PORT 1: D10/AD10
1	PB1F1	R/W	0: PORT 1: D9/AD9
0	PB0F1	R/W	0: PORT 1: D8/AD8

12.2.2.5 PBFR2 (Port B function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PB7F2	PB6F2	PB5F2	PB4F2	PB3F2	PB2F2	PB1F2	PB0F2
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PB7F2	R/W	0: PORT 1: A7
6	PB6F2	R/W	0: PORT 1: A6
5	PB5F2	R/W	0: PORT 1: A5
4	PB4F2	R/W	0: PORT 1: A4
3	PB3F2	R/W	0: PORT 1: A3
2	PB2F2	R/W	0: PORT 1: A2
1	PB1F2	R/W	0: PORT 1: A1
0	PB0F2	R/W	0: PORT 1: A0

12.2.2.6 PBOD (Port B open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PB7OD	PB6OD	PB5OD	PB4OD	PB3OD	PB2OD	PB1OD	PB0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PB7OD-PB0OD	R/W	0: Push-pull output 1: Open drain

12.2.2.7 PBPUP (Port B pull-up control register)

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

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

12.2.2.8 PBIE (Port B input control register)

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

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

12.2.3 Port C (PC0 to PC7)

The port C is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port C performs the external bus interface, the serial channel, the 16-bit timer /event counter and the external interrupt input.

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

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

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

12.2.3.1 Port C Register

Base Address = 0x400C_0200

Register name		Address (Base+)
Port C data register	PCDATA	0x0000
Port C output control register	PCCR	0x0004
Port C function register 1	PCFR1	0x0008
Port C function register 2	PCFR2	0x000C
Port C function register 3	PCFR3	0x0010
Port C function register 4	PCFR4	0x0014
Reserved	-	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port C open drain control register	PCOD	0x0028
Port C pull-up control register	PCPUP	0x002C
Reserved	-	0x0030
Reserved	-	0x0034
Port C input control register	PCIE	0x0038

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

12.2.3.2 PCDATA (Port C data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PC7-PC0	R/W	Port C data register.

12.2.3.3 PCCR (Port C output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7C	PC6C	PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PC7C-PC0C	R/W	Output 0: Disable 1: Enable

12.2.3.4 PCFR1 (Port C function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7F1	PC6F1	PC5F1	PC4F1	PC3F1	PC2F1	PC1F1	PC0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PC7F1	R/W	0: PORT 1: A7
6	PC6F1	R/W	0: PORT 1: A6
5	PC5F1	R/W	0: PORT 1: A5
4	PC4F1	R/W	0: PORT 1: A4
3	PC3F1	R/W	0: PORT 1: A3
2	PC2F1	R/W	0: PORT 1: A2
1	PC1F1	R/W	0: PORT 1: A1
0	PC0F1	R/W	0: PORT 1: A0

12.2.3.5 PCFR2 (Port C function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7F2	PC6F2	PC5F2	PC4F2	PC3F2	PC2F2	PC1F2	PC0F2
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PC7F2	R/W	0: PORT 1: INT3
6	PC6F2	R/W	0: PORT 1: SCLK2
5	PC5F2	R/W	0: PORT 1: RXD2
4	PC4F2	R/W	0: PORT 1: TXD2
3	PC3F2	R/W	0: PORT 1: INT2
2	PC2F2	R/W	0: PORT 1: SCLK1
1	PC1F2	R/W	0: PORT 1: RXD1
0	PC0F2	R/W	0: PORT 1: TXD1

12.2.3.6 PCFR3 (Port C function register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7F3	PC6F3	PC5F3	PC4F3	PC3F3	PC2F3	PC1F3	PC0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PC7F3	R/W	0: PORT 1: TB4IN1
6	PC6F3	R/W	0: PORT 1: TB4IN0
5	PC5F3	R/W	0: PORT 1: TB3IN1
4	PC4F3	R/W	0: PORT 1: TB3IN0
3	PC3F3	R/W	0: PORT 1: TB1OUT
2	PC2F3	R/W	0: PORT 1: TB0OUT
1	PC1F3	R/W	0: PORT 1: TB2IN1
0	PC0F3	R/W	0: PORT 1: TB2IN0

12.2.3.7 PCFR4 (Port C function register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PC6F4	-	-	-	PC2F4	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6	PC6F4	R/W	0: PORT 1: $\overline{\text{CTS2}}$
5-3	−	R	Read as "0".
2	PC2F4	R/W	0: PORT 1: $\overline{\text{CTS1}}$
1-0	−	R	Read as "0".

12.2.3.8 PCOD (Port C open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7OD	PC6OD	PC5OD	PC4OD	PC3OD	PC2OD	PC1OD	PC0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PC7OD- PC0OD	R/W	0: Push-pull output 1: Open drain

12.2.3.9 PCPUP (Port C pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7UP	PC6UP	PC5UP	PC4UP	PC3UP	PC2UP	PC1UP	PC0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PC7UP-PC0UP	R/W	Pull-up 0: Disable 1: Enable

12.2.3.10 PCIE (Port C input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7IE	PC6IE	PC5IE	PC4IE	PC3IE	PC2IE	PC1IE	PC0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PC7IE-PC0IE	R/W	input 0: Disable 1: Enable

12.2.4 Port D (PD0 to PD7)

The port D is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port D performs the external bus interface, the serial bus interface, the external interrupt input, the SSP, the 16-bit timer /event counter, the clock output and the ADC.

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

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

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

12.2.4.1 Port D Register

Base Address = 0x400C_0300

Register name		Address (Base+)
Port D data register	PDDATA	0x0000
Port D output control register	PDCR	0x0004
Port D function register 1	PDFR1	0x0008
Port D function register 2	PDFR2	0x000C
Port D function register 3	PDFR3	0x0010
Reserved	-	0x0014
Reserved	-	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port D open drain control register	PDOD	0x0028
Port D pull-up control register	PDPUP	0x002C
Reserved	-	0x0030
Reserved	-	0x0034
Port D input control register	PDIE	0x0038

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

12.2.4.2 PDDATA (Port D data register)

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

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

12.2.4.3 PDCR (Port D output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PD7C	PD6C	PD5C	PD4C	PD3C	PD2C	PD1C	PD0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PD7C-PD0C	R/W	Output 0: Disable 1: Enable

12.2.4.4 PDFR1 (Port D function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PD7F1	PD6F1	PD5F1	PD4F1	PD3F1	PD2F1	PD1F1	PD0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PD7F1	R/W	0: PORT 1: A15
6	PD6F1	R/W	0: PORT 1: A14
5	PD5F1	R/W	0: PORT 1: A13
4	PD4F1	R/W	0: PORT 1: A12
3	PD3F1	R/W	0: PORT 1: A11
2	PD2F1	R/W	0: PORT 1: A10
1	PD1F1	R/W	0: PORT 1: A9
0	PD0F1	R/W	0: PORT 1: A8

12.2.4.5 PDFR2 (Port D function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PD7F2	PD6F2	PD5F2	PD4F2	PD3F2	PD2F2	PD1F2	PD0F2
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PD7F2	R/W	0: PORT 1: SPSS
6	PD6F2	R/W	0: PORT 1: SPCK
5	PD5F2	R/W	0: PORT 1: SPDI
4	PD4F2	R/W	0: PORT 1: SPDO
3	PD3F2	R/W	0: PORT 1: INT4
2	PD2F2	R/W	0: PORT 1: SCK1
1	PD1F2	R/W	0: PORT 1: SI1/SCL1
0	PD0F2	R/W	0: PORT 1: SO1/SDA1

12.2.4.6 PDFR3 (Port D function register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PD7F3	-	-	-	PD3F3	PD2F3	PD1F3	PD0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PD7F3	R/W	0: PORT 1: SCOUT
6-4	−	R	Read as "0".
3	PD3F3	R/W	0: PORT 1: $\overline{\text{ADTRG}}$
2	PD2F3	R/W	0: PORT 1: TB9OUT
1	PD1F3	R/W	0: PORT 1: TB8OUT
0	PD0F3	R/W	0: PORT 1: TB7OUT

12.2.4.7 PDOD (Port D open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PD7OD	PD6OD	PD5OD	PD4OD	PD3OD	PD2OD	PD1OD	PD0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	PD7OD-PD0OD	R/W	0: Push-pull output 1: Open drain

12.2.4.8 PDPUP (Port D pull-up control register)

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

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

12.2.4.9 PDIE (Port D input control register)

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

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

12.2.5 Port E (PE0 to PE7)

The port E is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port E performs the external bus interface, the serial channel, the external interrupt input, the 16-bits timer (TMRB) and the high resolution PPG output (TMRD).

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

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

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

12.2.5.1 Port E Register

Base Address = 0x400C_0400

Register name		Address (Base+)
Port E data register	PEDATA	0x0000
Port E output control register	PECR	0x0004
Port E function register 1	PEFR1	0x0008
Port E function register 2	PEFR2	0x000C
Port E function register 3	PEFR3	0x0010
Port E function register 4	PEFR4	0x0014
Reserved	-	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port E open drain control register	PEOD	0x0028
Port E pull-up control register	PEPUP	0x002C
Reserved	-	0x0030
Reserved	-	0x0034
Port E input control register	PEIE	0x0038

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

12.2.5.2 PEDATA (Port E data register)

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

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

12.2.5.3 PECR (Port E output control register)

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

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

12.2.5.4 PEFR1 (Port E function register 1)

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

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3	PE3F1	R/W	0: PORT 1: INT5
2	PE2F1	R/W	0: PORT 1: SCLK0
1	PE1F1	R/W	0: PORT 1: RXD0
0	PE0F1	R/W	0: PORT 1: TXD0

12.2.5.5 PEFR2 (Port E function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PE7F2	PE6F2	PE5F2	PE4F2	PE3F2	PE2F2	PE1F2	PE0F2
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PE7F2	R/W	0: PORT 1: A23
6	PE6F2	R/W	0: PORT 1: A22
5	PE5F2	R/W	0: PORT 1: A21
4	PE4F2	R/W	0: PORT 1: A20
3	PE3F2	R/W	0: PORT 1: A19
2	PE2F2	R/W	0: PORT 1: A18
1	PE1F2	R/W	0: PORT 1: A17
0	PE0F2	R/W	0: PORT 1: A16

12.2.5.6 PEFR3 (Port E function register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PE7F3	PE6F3	PE5F3	PE4F3	PE3F3	PE2F3	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PE7F3	R/W	0: PORT 1: TD1OUT1
6	PE6F3	R/W	0: PORT 1: TD1OUT0
5	PE5F3	R/W	0: PORT 1: TD0OUT1
4	PE4F3	R/W	0: PORT 1: TD0OUT0
3	PE3F3	R/W	0: PORT 1: TB3OUT
2	PE2F3	R/W	0: PORT 1: TB2OUT
1-0	−	R	Read as "0".

12.2.5.7 PEFR4 (Port E function register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PE2F4	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as "0".
2	PE2F4	R/W	0: PORT 1: $\overline{\text{CTS0}}$
1-0	−	R	Read as "0".

12.2.5.8 PEOD (Port E open drain control register)

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

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

12.2.5.9 PEPUP (Port E pull-up control register)

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

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

12.2.5.10 PEIE (Port E input control register)

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

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

12.2.6 Port F (PF0 to PF7)

The port F is a general-purpose, 1-bit output port and 7-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose port function, the port F performs the external bus interface, the external interrupt input, 16-bit timer / event counter and the mode setting function.

Reset initializes from bit7 to bit1 of the port F as general-purpose ports with input, output and pull-up disabled. Reset initializes bit0 of PF as output port with output disabled and pull-up enabled.

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

While $\overline{\text{RESET}}$ pin is "Low", input and pull-up of PF0($\overline{\text{BOOT}}$) are enabled. At the rising edge of $\overline{\text{RESET}}$ pin, if PF0($\overline{\text{BOOT}}$) is "High", TMPM341FDXBG/FYXBG enters the single chip mode and boots from the on chip Flash ROM. If PF0($\overline{\text{BOOT}}$) is "Low", TMPM341FDXBG/FYXBG enters the single boot mode and boots from the on chip BOOTROM. For details of the single boot mode, refer to "Flash memory operation".

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

12.2.6.1 Port F Register

Base Address = 0x400C_0500

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

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

12.2.6.2 PFDATA (Port F data register)

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

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

12.2.6.3 PFCR (Port F output control register)

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

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

12.2.6.4 PFFR1 (Port F function register 1)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PF7F1	R/W	0: PORT 1: ALE
6	PF6F1	R/W	0: PORT 1: $\overline{\text{CS0}}$
5	PF5F1	R/W	0: PORT 1: $\overline{\text{CS1}}$
4	PF4F1	R/W	0: PORT 1: $\overline{\text{BELH}}$
3	PF3F1	R/W	0: PORT 1: $\overline{\text{BELL}}$
2	PF2F1	R/W	0: PORT 1: $\overline{\text{WR}}$
1	PF1F1	R/W	0: PORT 1: $\overline{\text{RD}}$
0	PF0F1	R/W	0: PORT 1: -

12.2.6.5 PFFR2 (Port F function register 2)

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

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as "0".
5	PF5F2	R/W	0: PORT 1: INT7
4	PF4F2	R/W	0: PORT 1: INT6
3-0	−	R	Read as "0".

12.2.6.6 PFFR3 (Port F function register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	PF5F3	PF4F3	-	-	-	PF0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as "0".
5	PF5F3	R/W	0: PORT 1: TB5IN1
4	PF4F3	R/W	0: PORT 1: TB5IN0
3-1	-	R	Read as "0".
0	PF0F3	R/W	0: PORT 1: TB6OUT

12.2.6.7 PFOD (Port F open drain control register)

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

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

12.2.6.8 PFPUP (Port F pull-up control register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-1	PF7UP-PF1UP	R/W	Pull-up 0: Disable 1: Enable
0	PF0UP	R/W	Pull-up 0: - 1: Always set to "1"

12.2.6.9 PFIE (Port F input control register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-1	PF7IE-PF1IE	R/W	Input 0: Disable 1: Enable
0	-	R	Read as "0".

12.2.7 Port G (PG0 to PG7)

The port G is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port G performs the serial bus interface, the serial channel, the external interrupt input and the 16-bit timer / event counter.

Only when input is enabled, PG tolerates 5V input. Note that this pin cannot be pulled up over the power supply voltage when using as open-drain output.

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

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

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

12.2.7.1 Port G Register

Base Address = 0x400C_0600

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

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

12.2.7.2 PGDATA (Port G data register)

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

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

12.2.7.3 PGCR (Port G output control register)

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

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

12.2.7.4 PGFR2 (Port G function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7F2	PG6F2	PG5F2	PG4F2	PG3F2	PG2F2	PG1F2	PG0F2
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PG7F2	R/W	0: PORT 1: INT1
6	PG6F2	R/W	0: PORT 1: SCLK3
5	PG5F2	R/W	0: PORT 1: RXD3
4	PG4F2	R/W	0: PORT 1: TXD3
3	PG3F2	R/W	0: PORT 1: INT0
2	PG2F2	R/W	0: PORT 1: SCK0
1	PG1F2	R/W	0: PORT 1: SI0/SCL0
0	PG0F2	R/W	0: PORT 1: SO0/SDA0

12.2.7.5 PGFR3 (Port G function register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7F3	PG6F3	PG5F3	PG4F3	-	PG2F3	PG1F3	PG0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PG7F3	R/W	0: PORT 1: TB9IN1
6	PG6F3	R/W	0: PORT 1: TB9IN0
5	PG5F3	R/W	0: PORT 1: TB8IN1
4	PG4F3	R/W	0: PORT 1: TB8IN0
3	−	R	Read as "0".
2	PG2F3	R/W	0: PORT 1: INT8
1	PG1F3	R/W	0: PORT 1: TB7IN1
0	PG0F3	R/W	0: PORT 1: TB7IN0

12.2.7.6 PGFR4 (Port G function register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PG6F4	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6	PG6F4	R/W	0: PORT 1: $\overline{\text{CTS3}}$
5-0	-	R	Read as "0".

12.2.7.7 PGOD (Port G open drain control register)

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

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

Note: Only when input is enabled, PG tolerates 5V input. Note that this pin cannot be pulled up over the power supply voltage when using as open-drain output.

12.2.7.8 PGPUP (Port G pull-up control register)

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

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

Note: PG5 is 5V tolerant input. When PG5 is pulled-up to 5V, PG5 should be pulled-up by external register. Do not use programable pull-up register of PG5. And PG5 should be connected to the external pull-up after all power supply pins are supplied voltage.

12.2.7.9 PGIE (Port G input control register)

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

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

12.2.8 Port H (PH0 to PH6)

The port H is a general-purpose, 7-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port H performs the debug interface, the serial channel and the two-phase pulse input counter.

Only when input is enabled, PH0 to PH2 tolerates 5V input. Note that this pin cannot be pulled up over the power supply voltage when using as open-drain output.

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

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

12.2.8.1 Port H Register

Base Address = 0x400C_0700

Register name		Address (Base+)
Port H data register	PHDATA	0x0000
Port H output control register	PHCR	0x0004
Port H function register 1	PHFR1	0x0008
Port H function register 2	PHFR2	0x000C
Port H function register 3	PHFR3	0x0010
Port H function register 4	PHFR4	0x0014
Reserved	-	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port H open drain control register	PHOD	0x0028
Port H pull-up control register	PHPUP	0x002C
Reserved	-	0x0030
Reserved	-	0x0034
Port H input control register	PHIE	0x0038

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

12.2.8.2 PHDATA (Port H data register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-0	PH6-PH0	R/W	Port H data register.

12.2.8.3 PHCR (Port H output control register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-0	PH6C-PH0C	R/W	Output 0: Disable 1: Enable

12.2.8.4 PHFR1 (Port H function register 1)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6	PH6F1	R/W	0: PORT 1: TRACEDATA2
5	PH5F1	R/W	0: PORT 1: TRACEDATA3
4-0	-	R	Read as "0".

12.2.8.5 PHFR2 (Port H function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PH4F2	PH3F2	PH2F2	PH1F2	PH0F2
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4	PH4F2	R/W	0: PORT 1: PHC3IN1
3	PH3F2	R/W	0: PORT 1: PHC3IN0
2	PH2F2	R/W	0: PORT 1: SCLK4
1	PH1F2	R/W	0: PORT 1: RXD4
0	PH0F2	R/W	0: PORT 1: TXD4

12.2.8.6 PHFR3 (Port H function register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PH4F3	PH3F3	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4	PH4F3	R/W	0: PORT 1: TB5OUT
3	PH3F3	R/W	0: PORT 1: TB4OUT
2-0	PH0F2	R	Read as "0".

12.2.8.7 PHFR4 (Port H function register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PH2F4	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as "0".
2	PH2F4	R/W	0: PORT 1: $\overline{\text{CTS4}}$
1-0	−	R	Read as "0".

12.2.8.8 PHOD (Port H open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PH6OD	PH5OD	PH4OD	PH3OD	PH2OD	PH1OD	PH0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-0	PH6OD-PH0OD	R/W	0: Push-pull output 1: Open-drain output

Note: Only when input is enabled, PH0 to PH2 tolerates 5V input. Note that this pin cannot be pulled up over the power supply voltage when using as open-drain output.

12.2.8.9 PHPUP (Port H pull-up control register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-0	PH6UP-PH0UP	R/W	Pull-up 0: Disable 1: Enable

12.2.8.10 PHIE (Port H input control register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-0	PH6IE-PH0IE	R/W	Input 0: Disable 1: Enable

12.2.9 Port I (PI0 to PI7)

The port I is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port I performs the debug interface.

Reset initializes PI3, PI4, PI5, PI6 and PI7 as debug interface. PI3 is initialized as the TRST function with input and pull-up enabled. PI4 is initialized as the TDI function with input and pull-up enabled. PI5 initialized as the TCK / SWCLK function with input and pull-down enabled. PI6 is initialized as the TMS / SWDIO function with input, output and pull-up enabled. PI7 is initialized as the TDO / SWV function with output enabled.

Reset initialize the other bits as general-purpose ports with input, output and pull-up disabled.

The port I has one types of function register. If you use the port I as a general-purpose port, set "0" to the corresponding bit of the four registers. If you use the port I as other than a general-purpose port, set "1" to the corresponding bit of the function register.

Note 1: If PI6 or PI7 is configured as the TMS/SWDIO or TDO/SWV pin, output is enabled even in STOP1/STOP2 mode regardless of the CGSTBYCR<DRVE>/<PTKEEP> bit setting

Note 2: If PI5 is configured as the TCK/SWCLK pin, it prevents the low power consumption mode from being fully effective. Configure PI5 to function as a general-purpose port if the TCK/SWCLK is not used.

12.2.9.1 Port I Register

Base Address = 0x400C_0800

Register name		Address (Base+)
Port I data register	PIDATA	0x0000
Port I output control register	PICR	0x0004
Port I function register 1	PIFR1	0x0008
Reserved	-	0x000C
Reserved	-	0x0010
Reserved	-	0x0014
Reserved	-	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Port I open drain control register	PIOD	0x0028
Port I pull-up control register	PIPUP	0x002C
Port I pull-down control register	PIPDN	0x0030
Reserved	-	0x0034
Port I input control register	PIIE	0x0038

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

12.2.9.2 PIDATA (Port I data register)

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

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

12.2.9.3 PICR (Port I output control register)

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

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

12.2.9.4 PIFR1(Port I function register 1)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PI7F1	R/W	0: PORT 1: TDO/SWV
6	PI6F1	R/W	0: PORT 1: TMS/SWDIO
5	PI5F1	R/W	0: PORT 1: TCK/SWCLK
4	PI4F1	R/W	0: PORT 1: TDI
3	PI3F1	R/W	0: PORT 1: $\overline{\text{TRST}}$
2	PI2F1	R/W	0: PORT 1: TRACECLK
1	PI1F1	R/W	0: PORT 1: TRACEDATA0
0	PI0F1	R/W	0: PORT 1: TRACEDATA1

12.2.9.5 PIOD (Port I open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PI2OD	PI1OD	PI0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as "0".
2-0	PI2OD-PI0OD	R/W	0: Push-pull output 1: Open-drain output

12.2.9.6 PIPUP (Port I pull-up control register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PI7UP	R/W	Pull-up 0: Disable 1: Enable
6	PI6UP	R/W	Pull-up 0: Disable 1: Enable, Always set to "1" as the debug interface.
5	-	R	Read as "0".
4-3	PI4UP-PI3UP	R/W	Pull-up 0: Disable 1: Enable, Always set to "1" as the debug interface.
2-0	PI2UP-PI0UP	R/W	Pull-up 0: Disable 1: Enable

12.2.9.7 PIPDN (Port I pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	PI5DN	-	-	-	-	-
After reset	0	0	1	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as "0".
5	PI5DN	R/W	Pull-up 0: Disable 1: Enable, Always set to "1" as the debug interface.
4-0	-	R	Read as "0".

12.2.9.8 PIIE (Port I input control register)

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

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

12.2.10 Port J (PJ0 to PJ7)

The port J is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port J performs the ADC, the external interrupt input, the 16-bit timer /event counter and the two-phase pulse input counter.

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

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

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

Note 1: Unless you use all the bits of port J and port K as analog input pins, conversion accuracy may be reduced. Be sure to verify that this causes no problem on your system.

Note 2: In modes other than STOP mode, interrupt input is enabled regardless of the PxFR register setting if input is enabled in PxIE. Make sure to disable unused interrupts when programming the device.

12.2.10.1 Port J Register

Base Address = 0x400C_0900

Register name		Address (Base+)
Port J data register	PJDATA	0x0000
Port J output control register	PJCR	0x0004
Reserved	-	0x0008
Port J function register 2	PJFR2	0x000C
Port J function register 3	PJFR3	0x0010
Reserved	-	0x0014
Reserved	-	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Reserved	-	0x0028
Port J pull-up control register	PJPUP	0x002C
Reserved	-	0x0030
Reserved	-	0x0034
Port J input control register	PJIE	0x0038

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

12.2.10.2 PJDATA (Port J data register)

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

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

12.2.10.3 PJCR (Port J output control register)

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

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

12.2.10.4 PJFR2 (Port J function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7F2	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PJ7F2	R/W	0: PORT 1: INT9
6-0	-	R	Read as "0".

12.2.10.5 PJFR3 (Port J function register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7F3	PJ6F3	PJ5F3	PJ4F3	PJ3F3	PJ2F3	PJ1F3	PJ0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	PJ7F3	R/W	0: PORT 1: TB0IN1
6	PJ6F3	R/W	0: PORT 1: TB0IN0
5	PJ5F3	R/W	0: PORT 1: PHC2IN1
4	PJ4F3	R/W	0: PORT 1: PHC2IN0
3	PJ3F3	R/W	0: PORT 1: PHC1IN1
2	PJ2F3	R/W	0: PORT 1: PHC1IN0
1	PJ1F3	R/W	0: PORT 1: PHC0IN1
0	PJ0F3	R/W	0: PORT 1: PHC0IN0

12.2.10.6 PJPUP (Port J pull-up control register)

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

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

12.2.10.7 PJIE (Port J input control register)

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

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

12.2.11 Port K (PK0 to PK6)

The port K is a general-purpose, 7-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port K performs the ADC, the external interrupt input and the 16-bit timer / event counter.

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

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

To use the Port K as an analog input of the AD converter, disable input on PKIE and disable pull-up on PJPUP.

Note 1: Unless you use all the bits of port J and port K as analog input pins, conversion accuracy may be reduced. Be sure to verify that this causes no problem on your system.

Note 2: In modes other than STOP mode, interrupt input is enabled regardless of the PxFR register setting if input is enabled in PxIE. Make sure to disable unused interrupts when programming the device.

12.2.11.1 Port K Register

Base Address = 0x400C_0A00

Register name		Address (Base+)
Port K data register	PKDATA	0x0000
Port K output control register	PKCR	0x0004
Reserved	-	0x0008
Port K function register 2	PKFR2	0x000C
Port K function register 3	PKFR3	0x0010
Reserved	-	0x0014
Reserved	-	0x0018
Reserved	-	0x001C
Reserved	-	0x0020
Reserved	-	0x0024
Reserved	-	0x0028
Port K pull-up control register	PKPUP	0x002C
Reserved	-	0x0030
Reserved	-	0x0034
Port K input control register	PKIE	0x0038

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

12.2.11.2 PKDATA (Port K data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PK6	PK5	PK4	PK3	PK2	PK1	PK0
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
6-0	PK6-PK0	R/W	Port K data register.

12.2.11.3 PKCR (Port K output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PK6C	PK5C	PK4C	PK3C	PK2C	PK1C	PK0C
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
6-0	PK6C-PK0C	R/W	Output 0: Disable 1: Enable

12.2.11.4 PKFR2 (Port K function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PK3F2	-	PK1F2	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3	PK3F2	R/W	0: PORT 1: INTB
2	-	R	Read as "0".
1	PK1F2	R/W	0: PORT 1: INTA
0	-	R	Read as "0".

12.2.11.5 PKFR3 (Port K function register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PK3F3	PK2F3	PK1F3	PK0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3	PK3F3	R/W	0: PORT 1: TB6IN1
2	PK2F3	R/W	0: PORT 1: TB6IN0
1	PK1F3	R/W	0: PORT 1: TB1IN1
0	PK0F3	R/W	0: PORT 1: TB1IN0

12.2.11.6 PKPUP (Port K pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PK6UP	PK5UP	PK4UP	PK3UP	PK2UP	PK1UP	PK0UP
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
6-0	PK6UP-PK0UP	R/W	Pull-up 0: Disable 1: Enable

12.2.11.7 PKIE (Port K input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PK6IE	PK5IE	PK4IE	PK3IE	PK2IE	PK1IE	PK0IE
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
6-0	PK7IE-PK0IE	R/W	Input 0: Disable 1: Enable

12.3 Block Diagrams of Ports

12.3.1 Port Types

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

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

Table 12-5 Function Lists

Type	GP Port	Function	Analog	Pull-up	Pull-down	Programmable open-drain	Note
FT1	I/O	I/O	−	R	−	o	
FT2	I/O	I/O	−	EnR	EnR	o	Function output triggered by enable signal
FT3	I/O	I/O	−	R	−	o	Function output triggered by enable signal
FT4	I/O	Input (int)	−	R	−	o	with Noise filter
FT5	I/O	Input	o	R	-	-	ADC pin
FT6	Output	Output	−	EnR	−	o	$\overline{\text{BOOT}}$ input enabled during reset
FT7	I/O	I/O	−	R	−	−	Function output triggered by enable signal
FT8	I/O	Input	−	R	−	o	
FT9	I/O	I/O	−	R	R	o	
FT10	I/O	I/O	−	R	R	o	Function output triggered by enable signal

int: Interrupt input
−: Not exist
o: Exist
R: Forced disable during reset.
EnR: Force enable during reset.

12.3.2 Type FT1

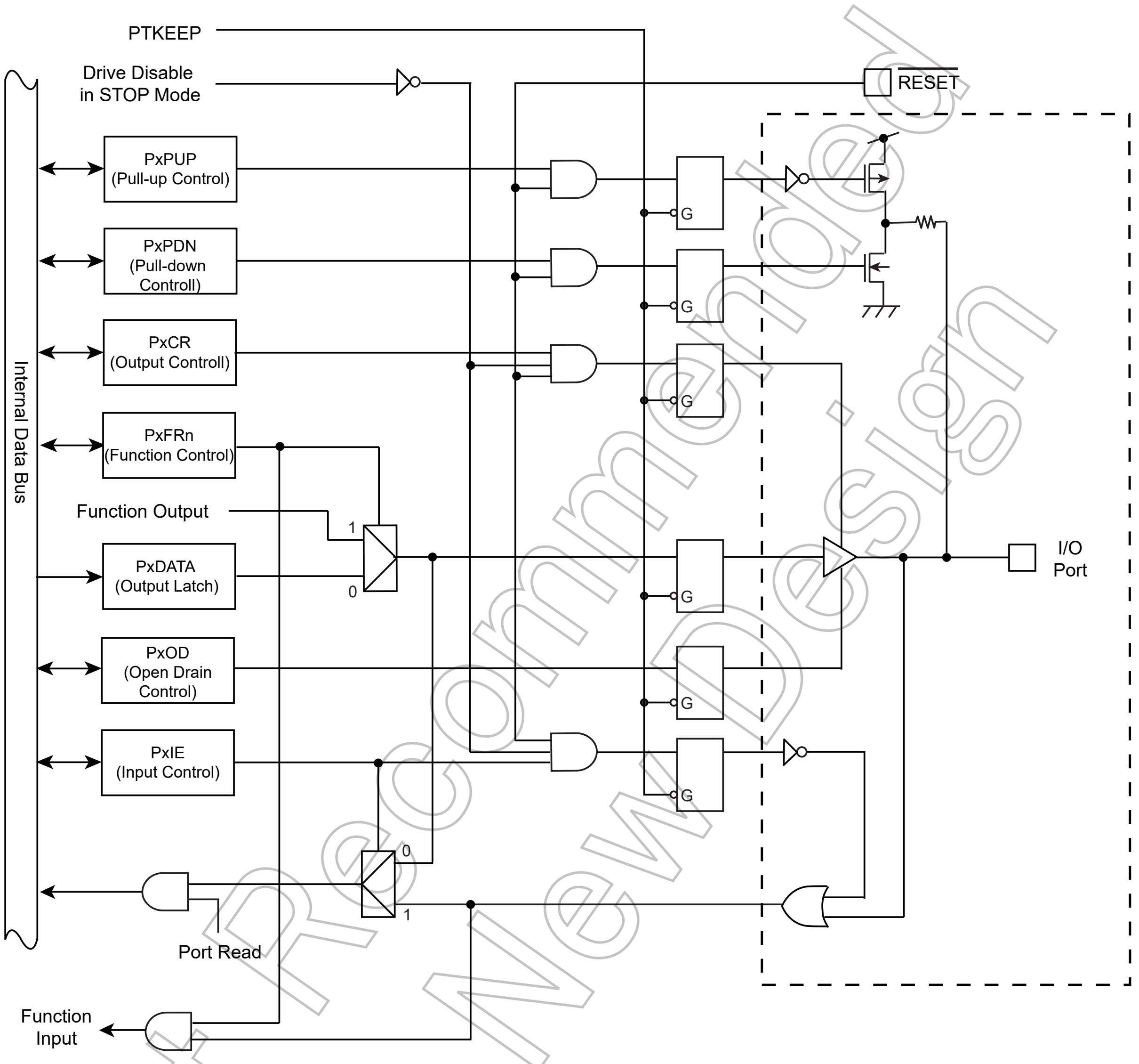


Figure 12-1 Port Type FT1

12.3.3 Type FT2

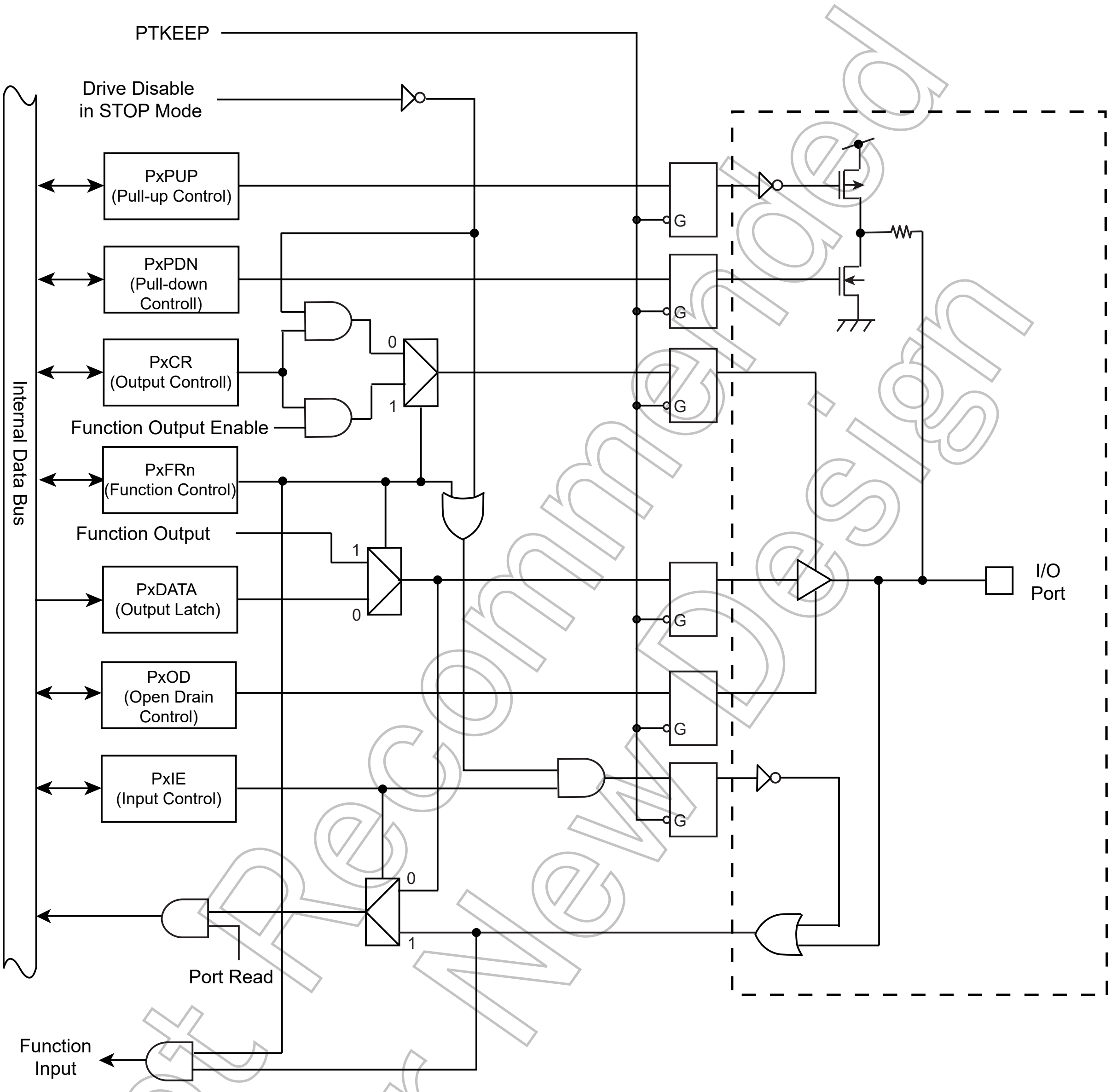


Figure 12-2 Port type FT2

Note: $\overline{\text{TRST}}$ has noise filter(30ns Typ.).

12.3.4 Type FT3

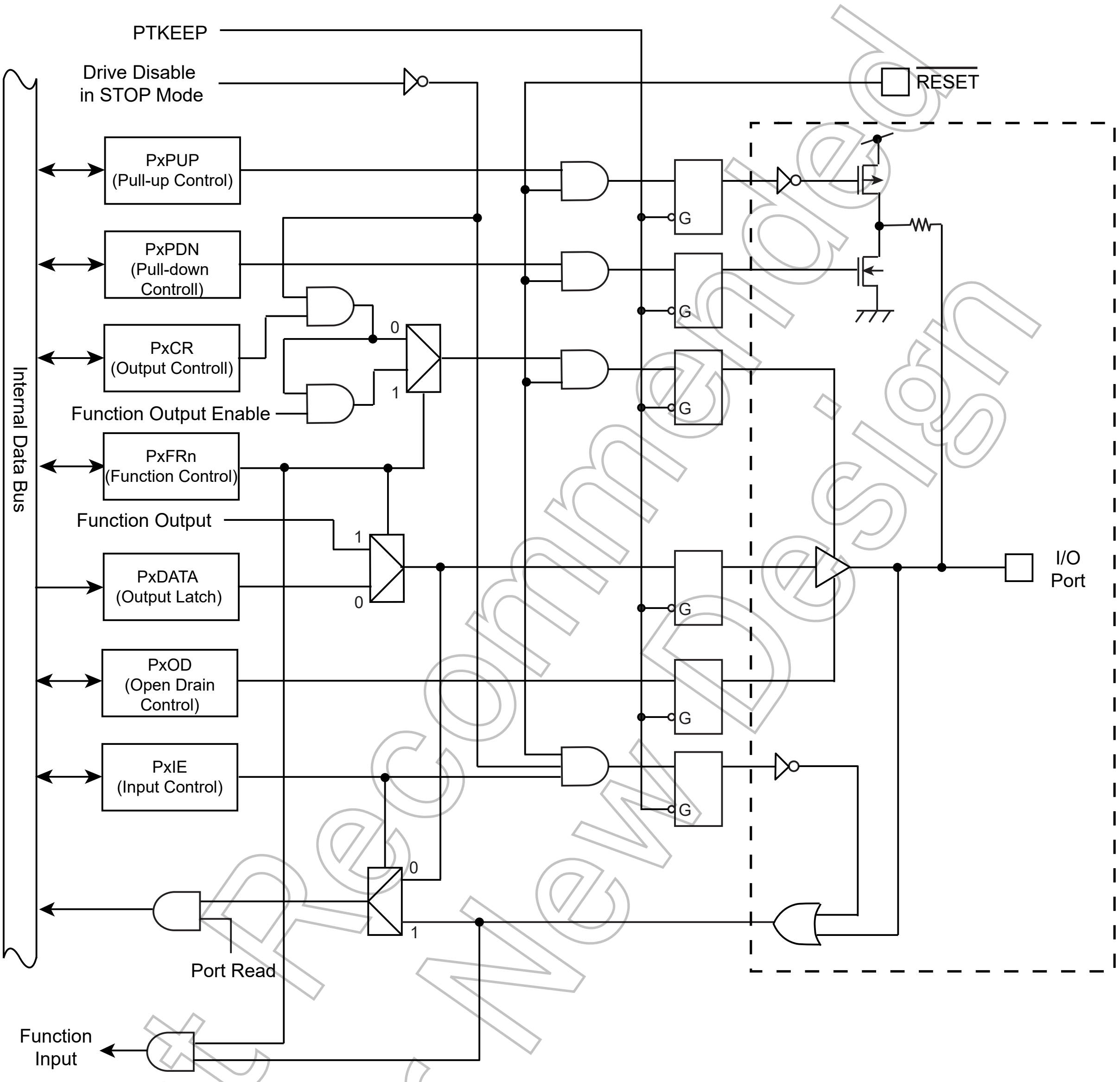


Figure 12-3 Port Type FT3

12.3.5 Type FT4

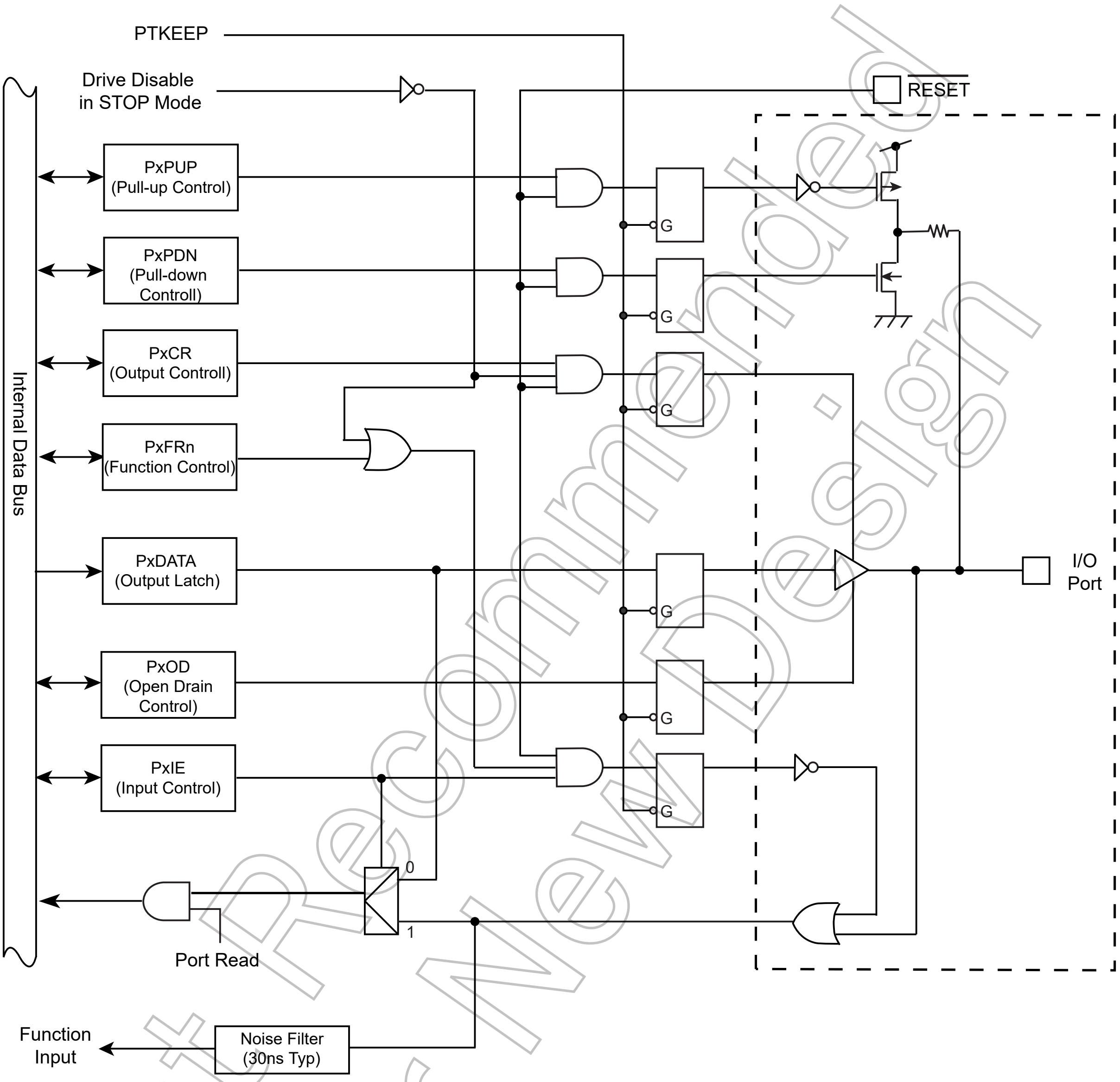


Figure 12-4 Port Type FT4

12.3.6 Type FT5

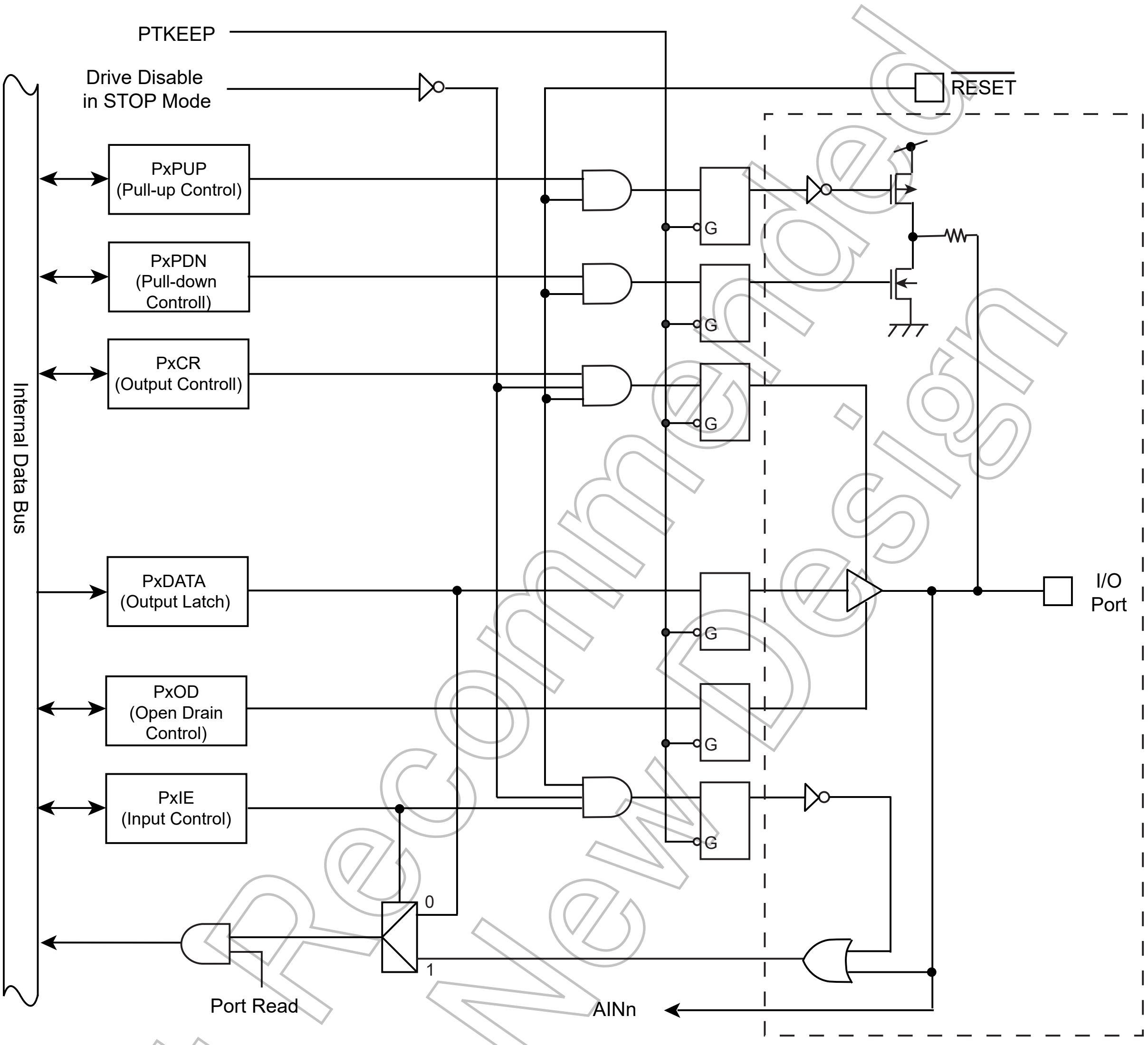


Figure 12-5 Port Type FT5

12.3.7 Type FT6

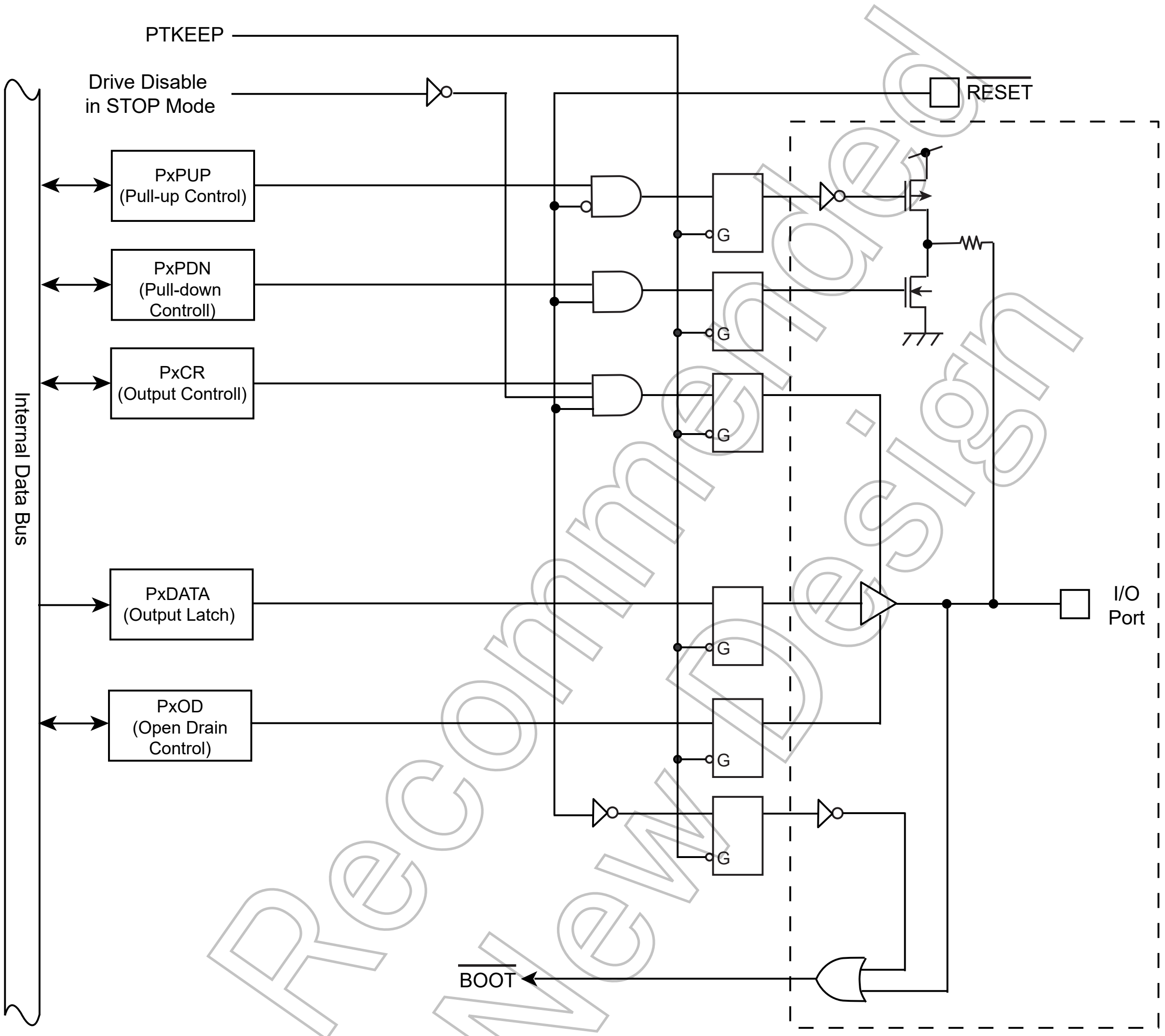


Figure 12-6 Port Type FT6

12.3.8 Type FT7

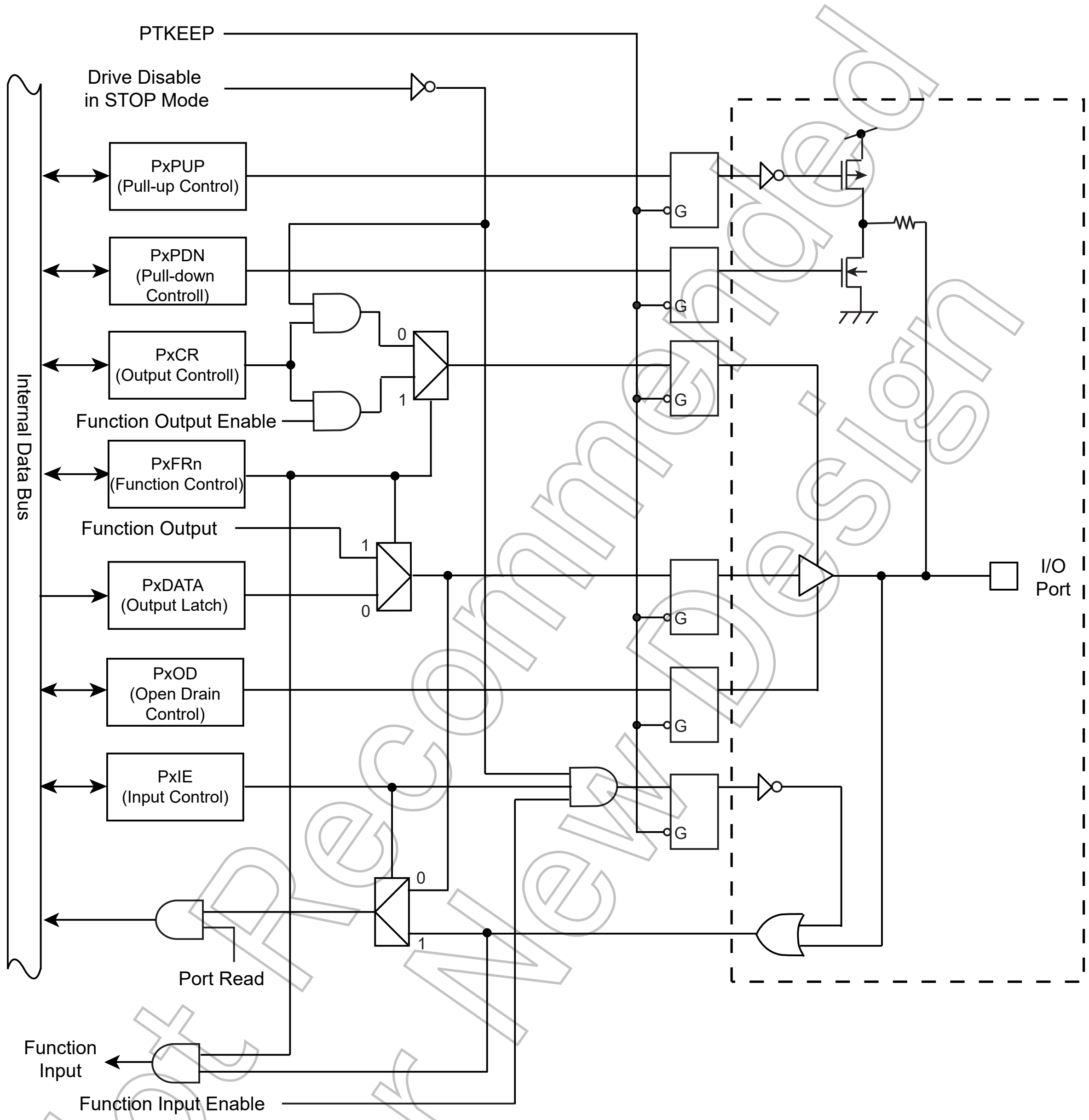


Figure 12-7 Port Type FT7

12.3.9 Type FT8

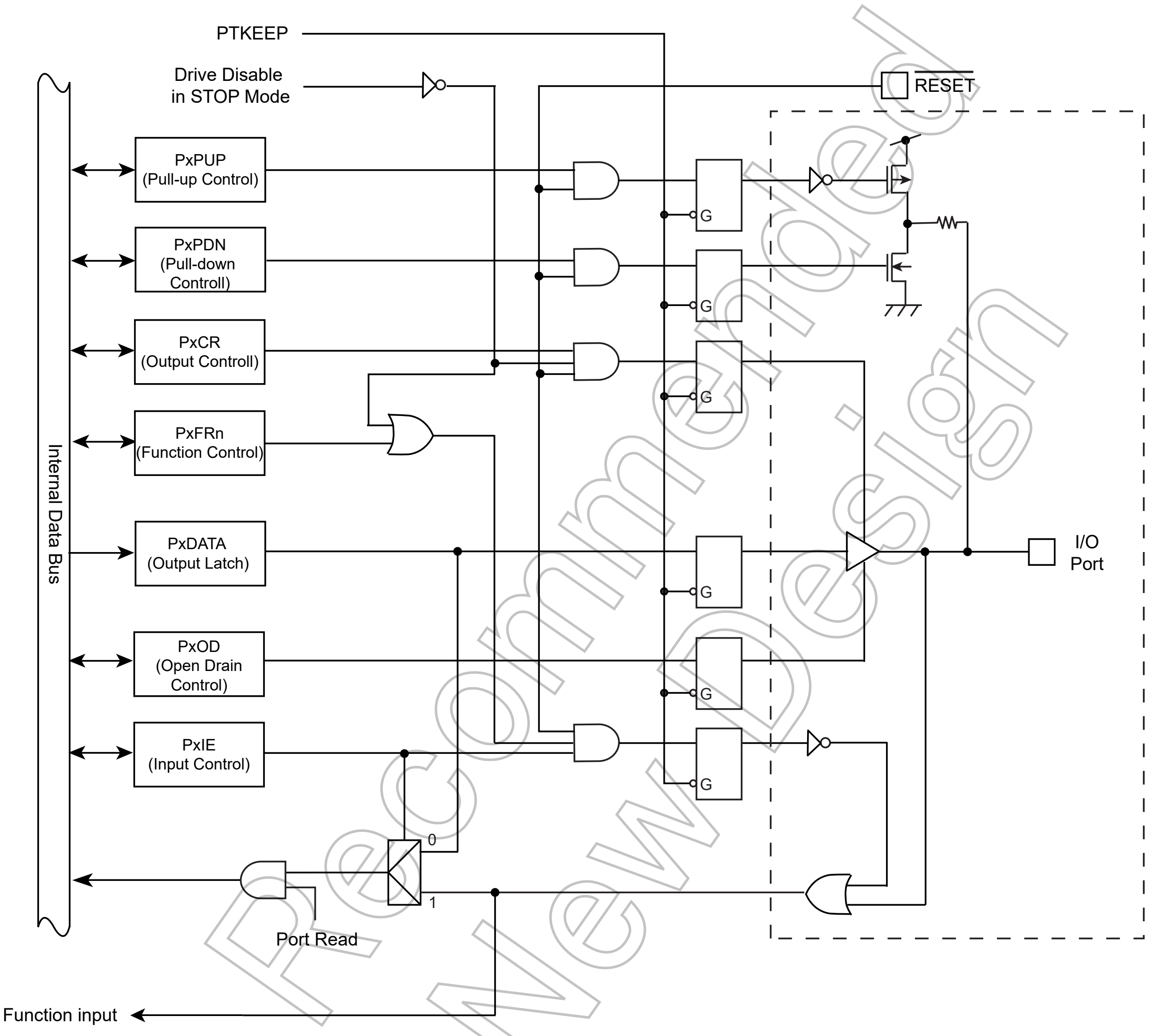


Figure 12-8 Port Type FT8

12.3.10 Type FT9

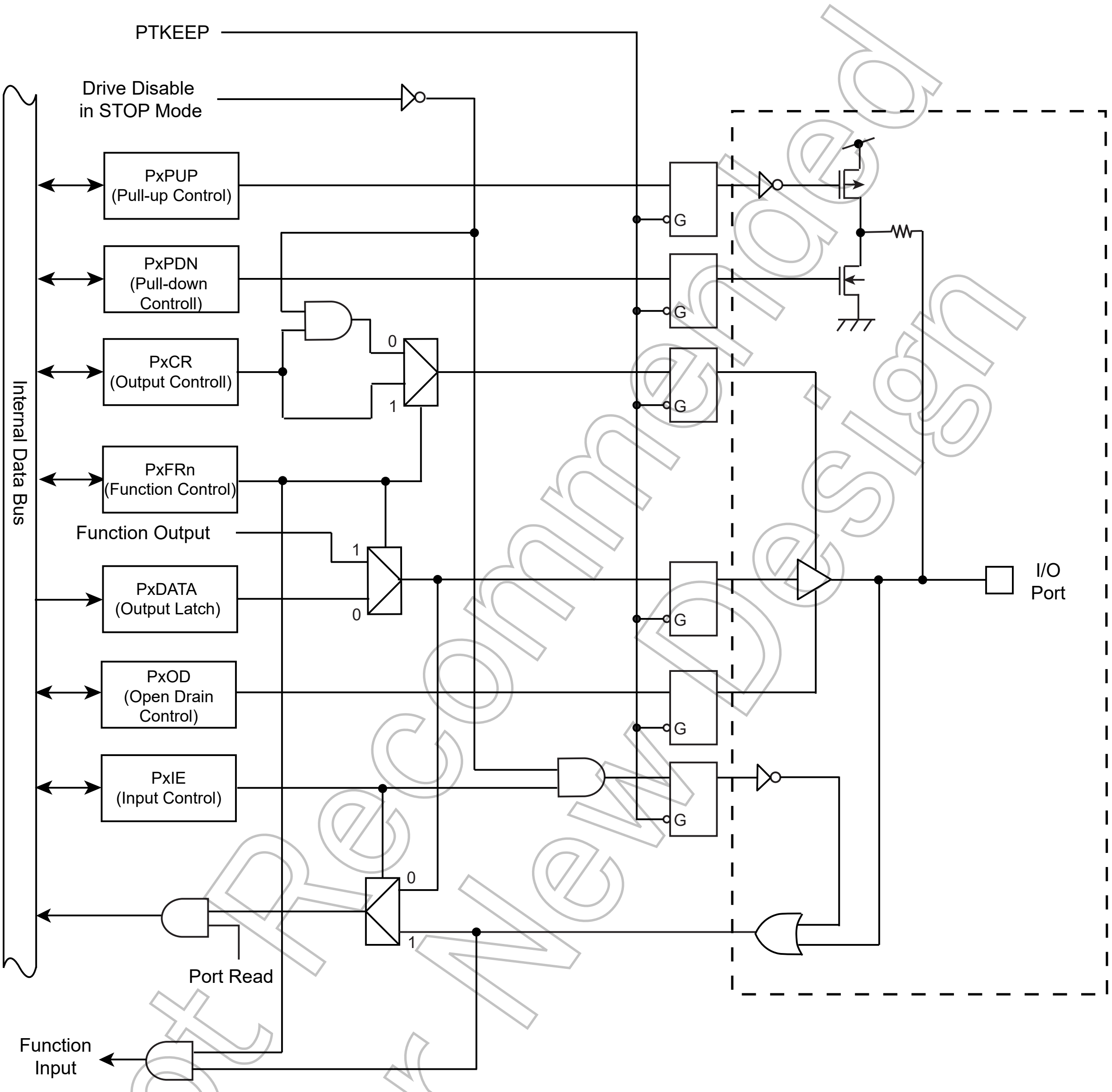


Figure 12-9 Port Type FT9

12.3.11 Type FT10

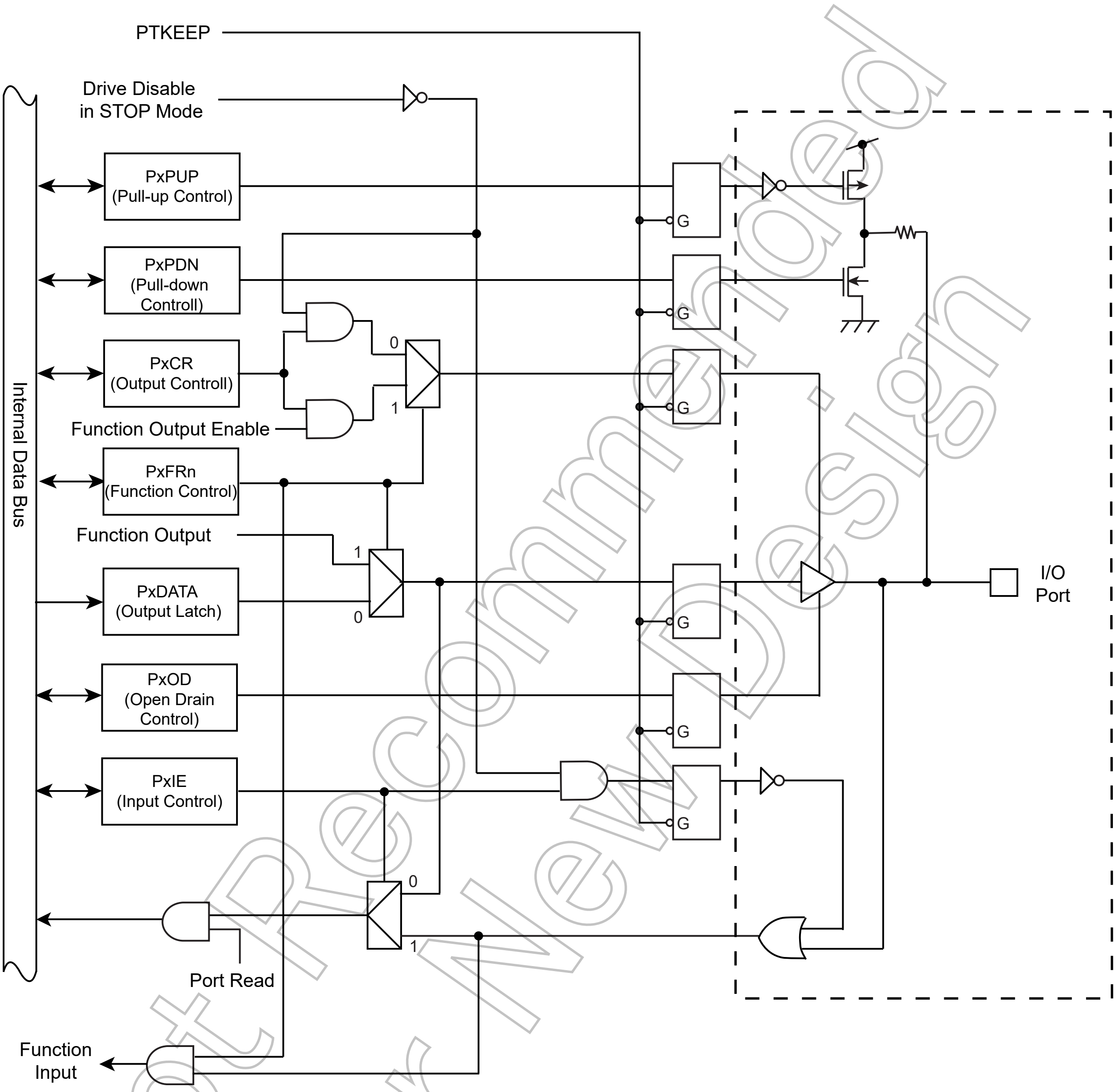


Figure 12-10 Port Type FT10

12.4 Appendix (Port setting List)

The following table shows the register setting for each function.

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

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

12.4.1 Port A Setting

Table 12-6 Port Setting List (Port A)

Pin	Port Type	Function	After reset	PACR	PAFR1	PAOD	PAPUP	PAIE
PA0	FT1	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
	FT7	D0/AD0 (Input/Output)		1	1	x	x	1
PA1	FT1	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
	FT7	D1/AD1 (Input/Output)		1	1	x	x	1
PA2	FT1	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
	FT7	D2/AD2 (Input/Output)		1	1	x	x	1
PA3	FT1	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
	FT7	D3/AD3 (Input/Output)		1	1	x	x	1
PA4	FT1	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
	FT7	D4/AD4 (Input/Output)		1	1	x	x	1
PA5	FT1	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
	FT7	D5/AD5 (Input/Output)		1	1	x	x	1
PA6	FT1	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
	FT7	D6/AD6 (Input/Output)		1	1	x	x	1
PA7	FT1	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
	FT7	D7/AD7 (Input/Output)		1	1	x	x	1

12.4.2 Port B Setting

Table 12-7 Port Setting List (Port B)

Pin	Port Type	Function	After reset	PBCR	PBFR1	FBFR2	PBOD	PBPUP	PBIE
PB0	FT1	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
	FT7	D8/AD8 (Input/Output)		1	1	0	x	x	1
	FT9	A0 (Output)		1	0	1	x	x	0
PB1	FT1	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
	FT7	D9/AD9 (Input/Output)		1	1	0	x	x	1
	FT9	A1 (Output)		1	0	1	x	x	0
PB2	FT1	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
	FT7	D10/AD10 (Input/Output)		1	1	0	x	x	1
	FT9	A2 (Output)		1	0	1	x	x	0
PB3	FT1	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
	FT7	D11/AD11 (Input/Output)		1	1	0	x	x	1
	FT9	A3 (Output)		1	0	1	x	x	0
PB4	FT1	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
	FT7	D12/AD12 (Input/Output)		1	1	0	x	x	1
	FT9	A4 (Output)		1	0	1	x	x	0
PB5	FT1	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
	FT7	D13/AD13 (Input/Output)		1	1	0	x	x	1
	FT9	A5 (Output)		1	0	1	x	x	0
PB6	FT1	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
	FT7	D14/AD14 (Input/Output)		1	1	0	x	x	1
	FT9	A6 (Output)		1	0	1	x	x	0
PB7	FT1	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
	FT7	D15/AD15 (Input/Output)		1	1	0	x	x	1
	FT9	A7 (Output)		1	0	1	x	x	0

12.4.3 Port C Setting

Table 12-8 Port Setting List (Port C)

Pin	Port Type	Function	After reset	PCCR	PCFR1	PCFR2	PCFR3	PCFR4	PCOD	PCPUP	PCIE
PC0	FT1	Input Port	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
		Output Port									
	FT9	A0 (Output)	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9
	FT1	TXD1 (Output)	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
		TB2IN0 (Input)									
PC1	FT1	Input Port	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
		Output Port									
	FT9	A1 (Output)	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9
	FT1	RXD1 (Input)	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
		TB2IN1 (Input)									
PC2	FT1	Input Port	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
		Output Port									
	FT9	A2 (Output)	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9
	FT1	SCLK1 (Input)	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
		SCLK1 (Output)									
		TB0OUT (Output)									
		$\overline{\text{CTS1}}$ (Input)									
PC3	FT1	Input Port	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
		Output Port									
	FT9	A3 (Output)	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9
	FT4	INT2 (Input)	FT4	FT4	FT4	FT4	FT4	FT4	FT4	FT4	FT4
	FT1	TB1OUT (Output)	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
PC4	FT1	Input Port	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
		Output Port									
	FT9	A4 (Output)	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9
	FT1	TXD2 (Output)	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
		TB3IN0 (Input)									
PC5	FT1	Input Port	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
		Output Port									
	FT9	A5 (Output)	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9
	FT1	RXD2 (Input)	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
		TB3IN1 (Input)									
PC6	FT1	Input Port	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
		Output Port									
	FT9	A6 (Output)	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9
	FT1	SCLK2 (Input)	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
		SCLK2 (Output)									
		TB4IN0 (Input)									
		$\overline{\text{CTS2}}$ (Input)									

Table 12-8 Port Setting List (Port C)

Pin	Port Type	Function	After reset	PCCR	PCFR1	PCFR2	PCFR3	PCFR4	PCOD	PCPUP	PCIE
PC7	FT1	Input Port	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1
		Output Port									
	FT9	A7 (Output)	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9	FT9
	FT4	INT3 (Input)	FT4	FT4	FT4	FT4	FT4	FT4	FT4	FT4	FT4
	FT1	TB4IN1 (Input)	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1	FT1

12.4.4 Port D Setting

Table 12-9 Port Setting List (Port D)

Pin	Port Type	Function	After re-set	PDCR	PDFR1	PDFR2	PDFR3	PDOD	PDPUP	PDIE
PD0	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	A8 (Output)		1	1	0	0	x	x	0
	FT1	SO1 (Output)		1	0	1	0	x	x	0
		SDA1 (Input/Output)		1	0	1	0	1	x	1
		TB7OUT (Output)		1	0	0	1	x	x	0
PD1	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	A9 (Output)		1	1	0	0	x	x	0
	FT1	SI1 (Input)		0	0	1	0	x	x	1
		SCL1 (Input/Output)		1	0	1	0	1	x	1
		TB8OUT (Output)		1	0	0	1	x	x	0
PD2	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	A10 (Output)		1	1	0	0	x	x	0
	FT1	SCK1 (Input)		0	0	1	0	x	x	1
		SCK1 (Output)		1	0	1	0	x	x	0
		TB9OUT (Output)		1	0	0	1	x	x	0
PD3	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	A11 (Output)		1	1	0	0	x	x	0
	FT4	INT4 (Input)		0	0	1	0	x	x	1
	FT1	ADTRG (Input)		0	0	0	1	x	x	1
PD4	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	A12 (Output)		1	1	0	0	x	x	0
	FT3	SPDO (Output)		1	0	1	0	x	x	0
PD5	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	A13 (Output)		1	1	0	0	x	x	0
	FT3	SPDI (Input)		0	0	1	0	x	x	1
PD6	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	A14 (Output)		1	1	0	0	x	x	0
	FT3	SPCLK (Input)		0	0	1	0	x	x	1
		SPCLK (Output)		1	0	1	0	x	x	0
PD7	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	A15 (Output)		1	1	0	0	x	x	0
	FT3	SPSS (Input)		0	0	1	0	x	x	1
		SPSS (Output)		1	0	1	0	x	x	0
	FT1	SCOUT (Output)		1	0	0	1	x	x	0

12.4.5 Port E Setting

Table 12-10 Port Setting List (Port E)

Pin	Port Type	Function	After reset	PECR	PEFR1	PEFR2	PEFR3	PEFR4	PEOD	PEPUP	PEIE
PE0	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT1	TXD0 (Output)		1	1	0	0	0	x	x	0
	FT9	A16 (Output)		1	0	1	0	0	x	x	0
PE1	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT1	RXD0 (Input)		0	1	0	0	0	x	x	1
	FT9	A17 (Output)		1	0	1	0	0	x	x	0
PE2	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT1	SCLK0 (Input)		0	1	0	0	0	x	x	1
		SCLK0 (Output)		1	1	0	0	0	x	x	0
	FT9	A18 (Output)		1	0	1	0	0	x	x	0
	FT1	TB2OUT (Output)		1	0	0	1	0	x	x	0
		$\overline{\text{CTS0}}$ (Input)		0	0	0	0	1	x	x	1
PE3	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT4	INT5 (Input)		0	1	0	0	0	x	x	1
	FT9	A19 (Output)		1	0	1	0	0	x	x	0
	FT1	TB3OUT (Output)		1	0	0	1	0	x	x	0
PE4	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT9	A20 (Output)		1	0	1	0	0	x	x	0
	FT1	TD0OUT0 (Output)		1	0	0	1	0	x	x	0
PE5	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT9	A21 (Output)		1	0	1	0	0	x	x	0
	FT1	TD0OUT1 (Output)		1	0	0	1	0	x	x	0
PE6	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT9	A22 (Output)		1	0	1	0	0	x	x	0
	FT1	TD1OUT0 (Output)		1	0	0	1	0	x	x	0
PE7	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT9	A23 (Output)		1	0	1	0	0	x	x	0
	FT1	TD1OUT1 (Output)		1	0	0	1	0	x	x	0

12.4.6 Port F Setting

Table 12-11 Port Setting List (Port F)

Pin	Port Type	Function	After re-set	PFCR	PFFR1	PFFR2	PFFR3	PFOD	PFPUP	PFIE
PF0	FT6	Output Port		1	0	0	0	x	1	0
	FT1	TB6OUT (output)		1	0	0	1	x	1	0
PF1	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	$\overline{\text{RD}}$ (output)		1	1	0	0	0	x	0
PF2	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	$\overline{\text{WR}}$ (output)		1	1	0	0	0	x	0
PF3	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	$\overline{\text{BELL}}$ (output)		1	1	0	0	x	x	0
PF4	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	$\overline{\text{BELH}}$ (output)		1	1	0	0	0	x	0
	FT4	INT6 (Input)		0	0	1	0	0	x	1
	FT1	TB5IN0 (Input)		0	0	0	1	0	x	1
PF5	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	$\overline{\text{CS1}}$ (output)		1	1	0	0	0	x	0
	FT4	INT7 (Input)		0	0	1	0	0	x	1
	FT1	TB5IN1 (Input)		0	0	0	1	0	x	1
PF6	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	$\overline{\text{CS0}}$ (output)		1	1	0	0	0	x	0
PF7	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT9	ALE (output)		1	1	0	0	0	x	0

Note:The PF0 input and pull-up are enabled and act as $\overline{\text{BOOT}}$ input pin while a $\overline{\text{RESET}}$ pin is "Low".

12.4.7 Port G Setting

Table 12-12 Port Setting List (Port G)

Pin	Port Type	Function	After reset	PGCR	PGFR2	PGFR3	PGFR4	PGOD	PGPUP	PGIE
PG0	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT1	SO0 (Output)		1	1	0	0	x	x	0
		SDA0 (Input/Output)		1	1	0	0	1	x	1
		TB7IN0 (Input)		0	0	1	0	0	x	1
PG1	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT1	SI0 (Input)		0	1	0	0	0	x	1
		SCL0 (Input/Output)		1	1	0	0	1	x	1
		TB7IN1 (Input)		0	0	1	0	0	x	1
PG2	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT1	SCK0 (Input)		0	1	0	0	0	x	1
		SCK0 (Output)		1	1	0	0	x	x	0
	FT4	INT8 (Input)		0	0	1	0	0	x	1
PG3	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT4	INT0 (Input)		0	1	0	0	0	x	1
PG4	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT1	TXD3 (Output)		1	1	0	0	x	x	0
		TB8IN0 (Input)		0	0	1	0	0	x	1
PG5	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT1	RXD3 (Input)		0	1	0	0	0	x	1
		TB8IN0 (Input)		0	0	1	0	0	x	1
PG6	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT1	SCLK3 (Input)		0	1	0	0	0	x	1
		SCLK3 (Output)		1	1	0	0	x	x	0
		TB9IN0 (Input)		0	0	1	0	0	x	1
		$\overline{\text{CTS3}}$ (Input)		0	0	0	1	x	x	1
PG7	FT1	Input Port		0	0	0	0	x	x	1
		Output Port		1	0	0	0	x	x	0
	FT4	INT1 (Input)		0	1	0	0	0	x	1
	FT1	TB9IN1 (Input)		0	0	1	0	0	x	1

12.4.8 Port H Setting

Table 12-13 Port Setting List (Port H)

Pin	Port Type	Function	After reset	PHCR	PHFR1	PHFR2	PHFR3	PHFR4	PHOD	PHPUP	PHIE
PH0	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT1	TXD4 (Output)		1	0	1	0	0	x	x	0
PH1	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT1	RXD4 (Input)		0	0	1	0	0	x	x	1
PH2	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT1	SCLK4 (Input)		0	0	1	0	0	x	x	1
		SCLK4 (Output)		1	0	1	0	0	x	x	0
		$\overline{\text{CTS4}}$ (Input)		0	0	0	0	1	x	x	1
PH3	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT8	PHC3IN0 (Input)		0	0	1	0	0	x	x	1
	FT1	TB4OUT (Output)		1	0	0	1	0	x	x	0
PH4	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT8	PHC3IN1 (Input)		0	0	1	0	0	x	x	1
	FT1	TB5OUT (Output)		1	0	0	1	0	x	x	0
PH5	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT9	TRACEDATA3 (Output)		1	1	0	0	0	x	0	0
PH6	FT1	Input Port		0	0	0	0	0	x	x	1
		Output Port		1	0	0	0	0	x	x	0
	FT9	TRACEDATA2 (Output)		1	1	0	0	0	x	0	0

12.4.9 Port I Setting

Table 12-14 Port Setting List (Port I)

Pin	Port Type	Function	After re-set	PICR	PIFR1	PIOD	PIPUP	PIPDN	PIIE
PI0	FT1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
	FT9	TRACEDATA1 (Output)		1	1	0	0	0	0
PI1	FT1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
	FT9	TRACEDATA0 (Output)		1	1	0	0	0	0
PI2	FT1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
	FT9	TRACECLK (Output)		1	1	0	0	0	0
PI3	FT1	Input Port		0	0	x	1	x	1
		Output Port		1	0	x	1	x	0
	FT2	$\overline{\text{TRST}}$ (Input)	o	0	1	0	1	0	1
PI4	FT1	Input Port		0	0	x	1	x	1
		Output Port		1	0	x	1	x	0
	FT2	TDI (Input)	o	0	1	0	1	0	1
PI5	FT1	Input Port		0	0	x	x	1	1
		Output Port		1	0	x	x	1	0
	FT2	TCK (Input)/ SWCLK (Input)	o	0	1	0	0	1	1
PI6	FT1	Input Port		0	0	x	1	x	1
		Output Port		1	0	x	1	x	0
	FT2	TMS (Input)/ SWDIO (Input/Output)	o	1	1	0	1	0	1
PI7	FT1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
	FT2	TDO (Output)/ SWV (Output)	o	1	1	0	0	0	0

12.4.10 Port J Setting

Table 12-15 Port Setting List (Port J)

pin	Port Type	Function	After reset	PJCR	PJFR2	PJFR3	PJPUP	PJIE
PJ0	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT8	PHC0IN0 (Input)		0	0	1	x	1
	FT5	AIN00 (Input)		0	0	0	0	0
PJ1	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT8	PHC0IN1 (Input)		0	0	1	x	1
	FT5	AIN01 (Input)		0	0	0	0	0
PJ2	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT8	PHC1IN0 (Input)		0	0	1	x	1
	FT5	AIN02 (Input)		0	0	0	0	0
PJ3	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT8	PHC1IN1 (Input)		0	0	1	x	1
	FT5	AIN03 (Input)		0	0	0	0	0
PJ4	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT8	PHC2IN0 (Input)		0	0	1	x	1
	FT5	AIN04 (Input)		0	0	0	0	0
PJ5	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT8	PHC2IN1 (Input)		0	0	1	x	1
	FT5	AIN05 (Input)		0	0	0	0	0
PJ6	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT1	TB0IN0 (Input)		0	0	1	x	1
	FT5	AIN06 (Input)		0	0	0	0	0
PJ7	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT4	INT9 (Input)		0	1	0	x	1
	FT1	TB0IN1 (Input)		0	0	1	x	1
	FT5	AIN07 (Input)		0	0	0	0	0

12.4.11 Port K Setting

Table 12-16 Port Setting List (Port K)

Pin	Port Type	Function	After re-set	PKCR	PKFR2	PKFR3	PKPUP	PKIE
PK0	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT1	TB1IN0 (Input)		0	0	1	x	1
	FT5	AIN08 (Input)		0	0	0	0	0
PK1	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT4	INTA (Input)		0	1	0	x	1
	FT1	TB1IN1 (Input)		0	0	1	x	1
	FT5	AIN09 (Input)		0	0	0	0	0
PK2	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT1	TB6IN0 (Input)		0	0	1	x	1
	FT5	AIN10 (Input)		0	0	0	0	0
PK3	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT4	INTB (Input)		0	1	0	x	1
	FT1	TB6IN1 (Input)		0	0	1	x	1
	FT5	AIN11 (Input)		0	0	0	0	0
PK4	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT5	AIN12 (Input)		0	0	0	0	0
PK5	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT5	AIN13 (Input)		0	0	0	0	0
PK6	FT1	Input Port		0	0	0	x	1
		Output Port		1	0	0	x	0
	FT5	AIN14 (Input)		0	0	0	0	0

13. 16-bit Timer/Event Counters(TMRB)

13.1 Outline

TMRB operate in the following four operation modes:

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

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

- Frequency measurement
- Pulse width measurement
- Time difference measurement

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

13.2 Differences in the Specifications

TMPM341FDXBG/FYXBG contains 10-channel of TMRB.

Each channel functions independently and the channels operate in the same way except for the differences in their specification as shown in Table 13-1.

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

1. The flip-flop output of TMRB 7 through TMRB 9 can be used as the capture trigger of other channels.
   - TB7OUT → available for TMRB0 through TMRB1
   - TB8OUT → available for TMRB2 through TMRB3
   - TB9OUT → available for TMRB4 through TMRB6

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

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

Table 13-1 Differences in the Specifications of TMRB Modules

Specification	External pins		Trigger function between timers		Interrupt		Internal connection			
Channel	Timer flip-flop output pin	External clock/capture trigger input pins	Capture trigger	Synchronous start trigger	Capture interrupt	TMRB interrupt	A/DC highest priority conversion start	ADC normal conversion start	Timer flip-flop output TBxOUT to SIO/UART (TXTRG :transfer clock)	Two phase pulse input to capture trigger
TMRB0	TB0OUT	TB0IN0 TB0IN1	TB7OUT	–	INTCAP00 INTCAP01	INTTB0	–	–	–	–
TMRB1	TB1OUT	TB1IN0 TB1IN1	TB7OUT	TB0PRUN,TB0RUN	INTCAP10 INTCAP11	INTTB1	–	–	–	–
TMRB2	TB2OUT	TB2IN0 TB2IN1	TB8OUT	TB0PRUN,TB0RUN	INTCAP20 INTCAP21	INTTB2	–	–	–	PHC0IN0 PHC0IN1
TMRB3	TB3OUT	TB3IN0 TB3IN1	TB8OUT	TB0PRUN,TB0RUN	INTCAP30 INTCAP31	INTTB3	–	–	–	PHC0IN1
TMRB4	TB4OUT	TB4IN0 TB4IN1	TB9OUT	–	INTCAP40 INTCAP41	INTTB4	TB4RG0 Match	–	–	PHC1IN0 PHC1IN1
TMRB5	TB5OUT	TB5IN0 TB5IN1	TB9OUT	TB4PRUN,TB4RUN	INTCAP50 INTCAP51	INTTB5	–	TB5RG0 Match	–	PHC1IN1
TMRB6	TB6OUT	TB6IN0 TB6IN1	TB9OUT	TB4PRUN,TB4RUN	INTCAP60 INTCAP61	INTTB6	–	–	–	PHC2IN0 PHC2IN1
TMRB7	TB7OUT	TB7IN0 TB7IN1	–	TB4PRUN,TB4RUN	INTCAP70 INTCAP71	INTTB7	–	–	–	PHC2IN1
TMRB8	TB8OUT	TB8IN0 TB8IN1	–	–	INTCAP80 INTCAP81	INTTB8	–	–	SIO0, SIO1, SIO2	PHC3IN0 PHC3IN1
TMRB9	TB9OUT	TB9IN0 TB9IN1	–	–	INTCAP90 INTCAP91	INTTB9	–	–	SIO3, SIO4	PHC3IN1

13.3 Configuration

Each channel consists of a 16-bit up-counter, two 16-bit timer registers (double-buffered), two 16-bit capture registers, two comparators, a capture input control, a timer flip-flop and its associated control circuit. Timer operation modes and the timer flip-flop are controlled by a register.

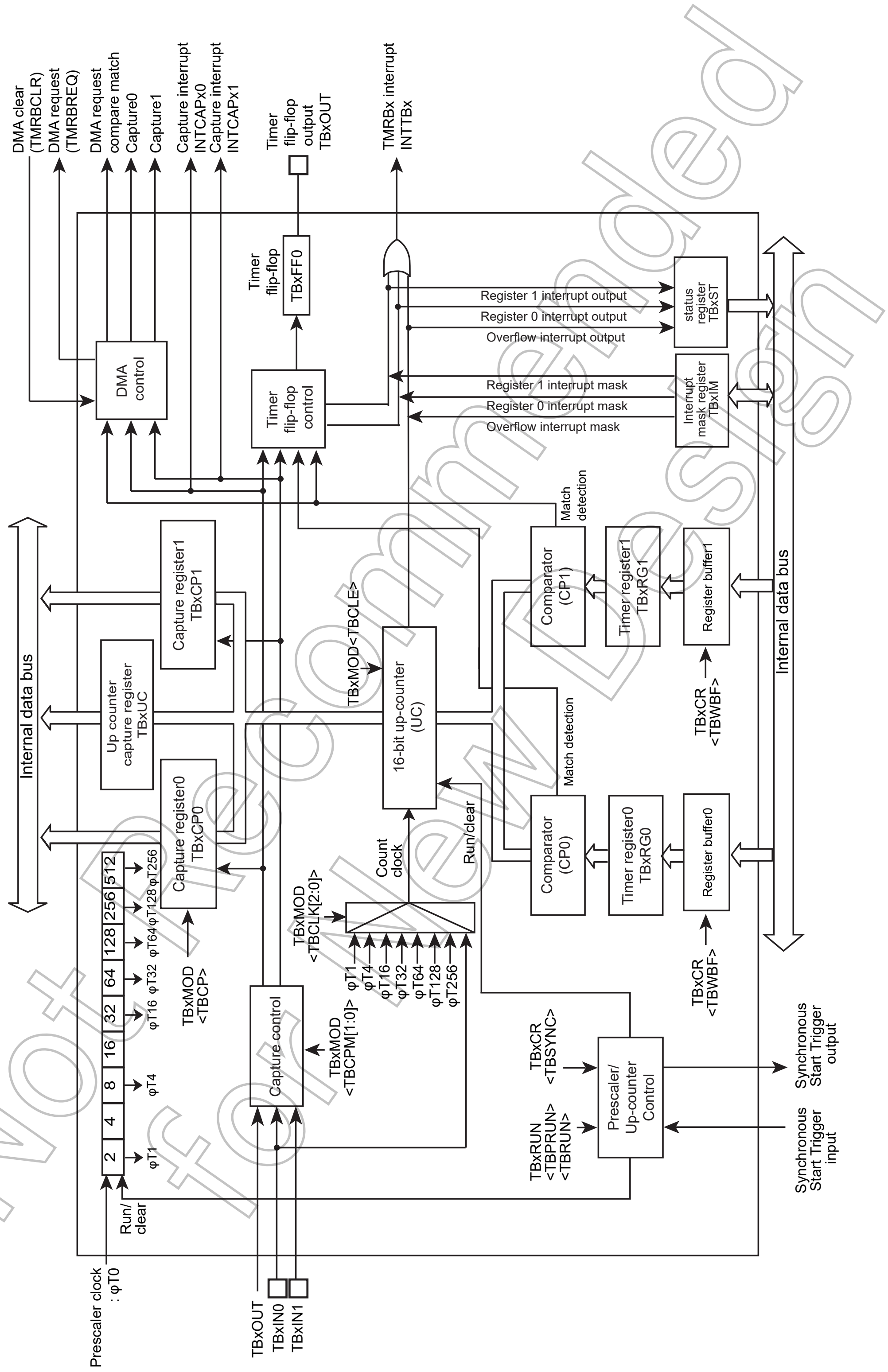


Figure 13-1 TMRBx Block Diagram(x= 0 to 9)

13.4 Registers

13.4.1 Register list according to channel

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

Channel x	Base Address
Channel0	0x400C_4000
Channel1	0x400C_4100
Channel2	0x400C_4200
Channel3	0x400C_4300
Channel4	0x400C_4400
Channel5	0x400C_4500
Channel6	0x400C_4600
Channel7	0x400C_4700
Channel8	0x400C_4800
Channel9	0x400C_4900

Register name(x=0~9)		Address (Base+)
Enable register	TBxEN	0x0000
RUN register	TBxRUN	0x0004
Control register	TBxCR	0x0008
Mode register	TBxMOD	0x000C
Flip-flop control register	TBxFFCR	0x0010
Status register	TBxST	0x0014
Interrupt mask register	TBxIM	0x0018
Up counter capture register	TBxUC	0x001C
Timer register 0	TBxRG0	0x0020
Timer register 1	TBxRG1	0x0024
Capture register 0	TBxCP0	0x0028
Capture register 1	TBxCP1	0x002C
DMA request enable register	TBxDMA	0x0030

Note: Timer control register, timer mode register and timer flip-flop control register can not be changed during timer operation. Make any changes of the above registers after timer stopped.

13.4.2 TBxEN (Enable register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	TBEN	R/W	TMRBx operation 0: Disabled 1: Enabled Specifies the TMRB operation. When the operation is disabled, no clock is supplied to the other registers in the TMRB module. This can reduce power consumption. (This disables reading from and writing to the other registers except TBxEN register.) To use the TMRB, enable the TMRB operation (set to "1") before programming each register in the TMRB module. If the TMRB operation is executed and then disabled, the settings will be maintained in each register.
6	TBHALT	R/W	Clock operation during debug HALT 0: run 1: stop Specifies the TMRB clock setting to run or stop when the debug tool transits to HALT mode while in use.
5-0	−	R	Read as 0.

13.4.3 TBxRUN(RUN register)

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

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

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

13.4.4 TBxCR (Control register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	TBWBF	R/W	Double Buffer 0: Disabled 1: Enabled
6	−	R/W	Write 0.
5	TBSYNC	R/W	Synchronous mode switching 0: individual (unit of channel) 1: synchronous
4	−	R	Read as 0.
3	I2TB	R/W	Operation at IDLE mode 0: Stop 1:Operation
2	TBINSEL	R/W	Selects the external inputs. 0: TBxIN0/1 1: PHCxIN0/1 Controls the external inputs selection between TBxIN0/1 or PHCxIN0/1.
1	TRGSEL	R/W	Selects the external triggers. 0: rising 1: falling Controls the edge selection (of signal to TBxIN0 pin) when the external triggers is selected.
0	CSSEL	R/W	Selects the count start 0: starts by software 1: starts by external trigger

13.4.5 TBxMOD (Mode register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	−	R/W	Write 0.
6	TBCP	W	Capture control by software 0: Capture by software 1: Don’t care When "0" is written, the capture register 0 (TBxCP0) takes count value. Read as 1.
5-4	TBCPM[1:0]	R/W	Capture timing (Refer to Table 13-2) 00: Disable Capture timing 01: TBxIN0↑, TBxIN1↑ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN0 pin input. Takes count values into capture register 1 (TBxCP1) upon rising of TBxIN1 pin input. 10: TBxIN0↑, TBxIN0↓ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN0 pin input. Takes count values into capture register 1 (TBxCP1) upon falling of TBxIN0 pin input. 11: TBxOUT↑, TBxOUT↓ Takes count values into capture register 0 (TBnCP0) upon rising of 16-bit timer match output (TBxOUT) and into capture register 1 (TBnCP1) upon falling of TBxOUT. (x = 7, n = 0,1), (x = 8, n = 2,3), (x = 9, n = 4,5,6), (TMRB0 to 1: TB7OUT, TMRB2 to 3: TB8OUT, TMRB4 to 6: TB9OUTÅj
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 there is a match with Timer Regsiter1 (TBxRG1).
2-0	TBCLK[2:0]	R/W	Selects the TMRBx source clock. 000: TBxIN0 pin input 001: φT1 010: φT4 011: φT16 100: φT32 101: φT64 110: φT128 111: φT256

Note: When the channel of TBxMOD register is 7, 8 or 9, the setting of <TBCPM[1:0]>="11" is prohibited.

Table 13-2 Relationship between <TBCPM[1:0]>, capture register and two phase plus counter input

TMRB Channel	TBCPM1	TBCPM0	Capture register	Capture timing	Two pulse counter channel
TMRB2	0	1	TB2CP0, TB2CP1	PHC0IN0↑, PHC0IN1↑	PHCNT0
	1	0	TB2CP0, TB2CP1	PHC0IN0↑, PHC0IN0↓	
TMRB3			TB3CP0, TB3CP1	PHC0IN1↑, PHC0IN1↓	
TMRB4	0	1	TB4CP0, TB4CP1	PHC1IN0↑, PHC1IN1↑	PHCNT1
	1	0	TB4CP0, TB4CP1	PHC1IN0↑, PHC1IN0↓	
TMRB5			TB5CP0, TB5CP1	PHC1IN1↑, PHC1IN1↓	
TMRB6	0	1	TB6CP0, TB6CP1	PHC2IN0↑, PHC2IN1↑	PHCNT2
	1	0	TB6CP0, TB6CP1	PHC2IN0↑, PHC2IN0↓	
TMRB7			TB7CP0, TB7CP1	PHC2IN1↑, PHC2IN1↓	
TMRB8	0	1	TB8CP0, TB8CP1	PHC3IN0↑, PHC3IN1↑	PHCNT3
	1	0	TB8CP0, TB8CP1	PHC3IN0↑, PHC3IN0↓	
TMRB9			TB9CP0, TB9CP1	PHC3IN1↑, PHC3IN1↓	

Note: TMRB0 and TMRB1 do not have capture function of two pulse counter input.

Note: Do not make any changes of TBxMOD register while corresponding TMRBx or PHCNTx is running.

13.4.6 TBxFFCR (Flip-flop control register)

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

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

13.4.7 TBxST (Status register)

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

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

Note 1: When the interrupt mask configuration is disabled by the corresponding bit of TBxIM register, the interrupt is issued to the CPU.

Note 2: To clear the flag, TBxST register should be read.

Note 3: Even if mask configuration by TBxIM register is valid, the status is set to TBxST register.

13.4.8 TBxIM (Interrupt mask register)

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

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

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

13.4.9 TBxUC (Up counter capture register)

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

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

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

13.4.10 TBxRG0 (Timer register 0)

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

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

13.4.11 TBxRG1 (Timer register 1)

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

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

13.4.12 TBxCP0 (Capture register 0)

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

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

13.4.13 TBxCP1 (Capture register 1)

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

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

13.4.14 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. 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: When mask configuration by TBxIM register is valid, DMA request does not issue even if it is enabled.

Note: The assignment of DMA request factor differs by chennel, TMRB0 to 9. Refer to Chapter "DMAC " for details.

13.5 Description of Operations for Each Circuit

The channels operate in the same way, except for the differences in their specifications as shown in Table 13-3.

13.5.1 Prescaler

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

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

The operation or the stoppage of a prescaler is set with TBxRUN<TBPRUN> where writing "1" starts counting and writing "0" clears and stops counting. Table 13-3 and Table 13-4 show prescaler output clock resolutions.

Table 13-3 Prescaler Output Clock Resolutions(fc = 54MHz)

Select peripheral clock CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
0 (fgear)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.0370 μs)	fc/2^{3} (0.148 μs)	fc/2^{5} (0.593 μs)
		001 (fperiph/2)	fc/2^{2} (0.0741 μs)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)
		010 (fperiph/4)	fc/2^{3} (0.148 μs)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)
		011 (fperiph/8)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)
		100 (fperiph/16)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)
		101 (fperiph/32)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^{2} (0.0741 μs)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)
		001 (fperiph/2)	fc/2^{3} (0.148 μs)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)
		010 (fperiph/4)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)
		011 (fperiph/8)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)
		100 (fperiph/16)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)
		101 (fperiph/32)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^{3} (0.148 μs)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)
		001 (fperiph/2)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)
		010 (fperiph/4)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)
		011 (fperiph/8)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)
		100 (fperiph/16)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)
		101 (fperiph/32)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)	fc/2^{12} (75.9μs)
	110 (fc/8)	000 (fperiph/1)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)
		001 (fperiph/2)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)
		010 (fperiph/4)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)
		011 (fperiph/8)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)
		100 (fperiph/16)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)	fc/2^{12} (75.9μs)
		101 (fperiph/32)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)	fc/2^{13} (152 μs)
	111 (fc/16)	000 (fperiph/1)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)
		001 (fperiph/2)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)
		010 (fperiph/4)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)
		011 (fperiph/8)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)	fc/2^{12} (75.9μs)
		100 (fperiph/16)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)	fc/2^{13} (152 μs)
		101 (fperiph/32)	fc/2^{10} (19.0 μs)	fc/2^{12} (75.9 μs)	fc/2^{14} (303 μs)

Table 13-3 Prescaler Output Clock Resolutions(fc = 54MHz)

Select peripheral clock CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
1 (fc)	000 (fc)	000 (fperiph/1)	fc/2^1 (0.0370 µs)	fc/2^3 (0.148 µs)	fc/2^5 (0.593 µs)
		001 (fperiph/2)	fc/2^2 (0.0741 µs)	fc/2^4 (0.296 µs)	fc/2^6 (1.19 µs)
		010 (fperiph/4)	fc/2^3 (0.148 µs)	fc/2^5 (0.593 µs)	fc/2^7 (2.37 µs)
		011 (fperiph/8)	fc/2^4 (0.296 µs)	fc/2^6 (1.19 µs)	fc/2^8 (4.74 µs)
		100 (fperiph/16)	fc/2^5 (0.593 µs)	fc/2^7 (2.37 µs)	fc/2^9 (9.48 µs)
		101 (fperiph/32)	fc/2^6 (1.19 µs)	fc/2^8 (4.74 µs)	fc/2^{10} (19.0 µs)
	100 (fc/2)	000 (fperiph/1)	–	fc/2^3 (0.148 µs)	fc/2^5 (0.593 µs)
		001 (fperiph/2)	fc/2^2 (0.0741 µs)	fc/2^4 (0.296 µs)	fc/2^6 (1.19 µs)
		010 (fperiph/4)	fc/2^3 (0.148 µs)	fc/2^5 (0.593 µs)	fc/2^7 (2.37 µs)
		011 (fperiph/8)	fc/2^4 (0.296 µs)	fc/2^6 (1.19 µs)	fc/2^8 (4.74 µs)
		100 (fperiph/16)	fc/2^5 (0.593 µs)	fc/2^7 (2.37 µs)	fc/2^9 (9.48 µs)
		101 (fperiph/32)	fc/2^6 (1.33 µs)	fc/2^8 (5.33 µs)	fc/2^{10} (21.3 µs)
	101 (fc/4)	000 (fperiph/1)	–	fc/2^3 (0.148 µs)	fc/2^5 (0.593 µs)
		001 (fperiph/2)	–	fc/2^4 (0.296 µs)	fc/2^6 (1.19 µs)
		010 (fperiph/4)	fc/2^3 (0.148 µs)	fc/2^5 (0.593 µs)	fc/2^7 (2.37 µs)
		011 (fperiph/8)	fc/2^4 (0.296 µs)	fc/2^6 (1.19 µs)	fc/2^8 (4.74 µs)
		100 (fperiph/16)	fc/2^5 (0.593 µs)	fc/2^7 (2.37 µs)	fc/2^9 (9.48 µs)
		101 (fperiph/32)	fc/2^6 (1.33 µs)	fc/2^8 (5.33 µs)	fc/2^{10} (21.3 µs)
	110 (fc/8)	000 (fperiph/1)	–	–	fc/2^5 (0.593 µs)
		001 (fperiph/2)	–	fc/2^4 (0.296 µs)	fc/2^6 (1.19 µs)
		010 (fperiph/4)	–	fc/2^5 (0.593 µs)	fc/2^7 (2.37 µs)
		011 (fperiph/8)	fc/2^4 (0.296 µs)	fc/2^6 (1.19 µs)	fc/2^8 (4.74 µs)
		100 (fperiph/16)	fc/2^5 (0.593 µs)	fc/2^7 (2.37 µs)	fc/2^9 (9.48 µs)
		101 (fperiph/32)	fc/2^6 (1.33 µs)	fc/2^8 (5.33 µs)	fc/2^{10} (21.3 µs)
	111 (fc/16)	000 (fperiph/1)	–	–	fc/2^5 (0.593 µs)
		001 (fperiph/2)	–	–	fc/2^6 (1.19 µs)
		010 (fperiph/4)	–	fc/2^5 (0.593 µs)	fc/2^7 (2.37 µs)
		011 (fperiph/8)	–	fc/2^6 (1.19 µs)	fc/2^8 (4.74 µs)
		100 (fperiph/16)	fc/2^5 (0.593 µs)	fc/2^7 (2.37 µs)	fc/2^9 (9.48 µs)
		101 (fperiph/32)	fc/2^6 (1.33 µs)	fc/2^8 (5.33 µs)	fc/2^{10} (21.3 µs)

Note 1: The prescaler output clock φTn must be selected so that φTn < fsys is satisfied (so that φTn is slower than fsys).
Note 2: Do not change the clock gear while the timer is operating.
Note 3: "−" denotes a setting prohibited.

Table 13-4 Prescaler Output Clock Resolutions(fc = 40MHz)

Select peripheral clock CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function			
			φT32	φT64	φT128	φT256
0 (fgear)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.0370 μs)	fc/2^{3} (0.148 μs)	fc/2^{5} (0.593 μs)	0 (fgear)
		001 (fperiph/2)	fc/2^{2} (0.0741 μs)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	
		010 (fperiph/4)	fc/2^{3} (0.148 μs)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	
		011 (fperiph/8)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	
		100 (fperiph/16)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	
		101 (fperiph/32)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)	
	100 (fc/2)	000 (fperiph/1)	fc/2^{2} (0.0741 μs)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	
		001 (fperiph/2)	fc/2^{3} (0.148 μs)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	
		010 (fperiph/4)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	
		011 (fperiph/8)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	
		100 (fperiph/16)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)	
		101 (fperiph/32)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)	
	101 (fc/4)	000 (fperiph/1)	fc/2^{3} (0.148 μs)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	
		001 (fperiph/2)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	
		010 (fperiph/4)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	
		011 (fperiph/8)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)	
		100 (fperiph/16)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)	
		101 (fperiph/32)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)	fc/2^{12} (75.9μs)	
	110 (fc/8)	000 (fperiph/1)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	
		001 (fperiph/2)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	
		010 (fperiph/4)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)	
		011 (fperiph/8)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)	
		100 (fperiph/16)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)	fc/2^{12} (75.9μs)	
		101 (fperiph/32)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)	fc/2^{13} (152 μs)	
	111 (fc/16)	000 (fperiph/1)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	
		001 (fperiph/2)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)	
		010 (fperiph/4)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)	
		011 (fperiph/8)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)	fc/2^{12} (75.9μs)	
		100 (fperiph/16)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)	fc/2^{13} (152 μs)	
		101 (fperiph/32)	fc/2^{10} (19.0 μs)	fc/2^{12} (75.9 μs)	fc/2^{14} (303 μs)	

Table 13-4 Prescaler Output Clock Resolutions(fc = 40MHz)

Select peripheral clock CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function			
			φT32	φT64	φT128	φT256
1 (fc)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.0370 µs)	fc/2^{3} (0.148 µs)	fc/2^{5} (0.593 µs)	1 (fc)
		001 (fperiph/2)	fc/2^{2} (0.0741 µs)	fc/2^{4} (0.296 µs)	fc/2^{6} (1.19 µs)	
		010 (fperiph/4)	fc/2^{3} (0.148 µs)	fc/2^{5} (0.593 µs)	fc/2^{7} (2.37 µs)	
		011 (fperiph/8)	fc/2^{4} (0.296 µs)	fc/2^{6} (1.19 µs)	fc/2^{8} (4.74 µs)	
		100 (fperiph/16)	fc/2^{5} (0.593 µs)	fc/2^{7} (2.37 µs)	fc/2^{9} (9.48 µs)	
		101 (fperiph/32)	fc/2^{6} (1.19 µs)	fc/2^{8} (4.74 µs)	fc/2^{10} (19.0 µs)	
	100 (fc/2)	000 (fperiph/1)	–	fc/2^{3} (0.148 µs)	fc/2^{5} (0.593 µs)	
		001 (fperiph/2)	fc/2^{2} (0.0741 µs)	fc/2^{4} (0.296 µs)	fc/2^{6} (1.19 µs)	
		010 (fperiph/4)	fc/2^{3} (0.148 µs)	fc/2^{5} (0.593 µs)	fc/2^{7} (2.37 µs)	
		011 (fperiph/8)	fc/2^{4} (0.296 µs)	fc/2^{6} (1.19 µs)	fc/2^{8} (4.74 µs)	
		100 (fperiph/16)	fc/2^{5} (0.593 µs)	fc/2^{7} (2.37 µs)	fc/2^{9} (9.48 µs)	
		101 (fperiph/32)	fc/2^{6} (1.33 µs)	fc/2^{8} (5.33 µs)	fc/2^{10} (21.3 µs)	
	101 (fc/4)	000 (fperiph/1)	–	fc/2^{3} (0.148 µs)	fc/2^{5} (0.593 µs)	
		001 (fperiph/2)	–	fc/2^{4} (0.296 µs)	fc/2^{6} (1.19 µs)	
		010 (fperiph/4)	fc/2^{3} (0.148 µs)	fc/2^{5} (0.593 µs)	fc/2^{7} (2.37 µs)	
		011 (fperiph/8)	fc/2^{4} (0.296 µs)	fc/2^{6} (1.19 µs)	fc/2^{8} (4.74 µs)	
		100 (fperiph/16)	fc/2^{5} (0.593 µs)	fc/2^{7} (2.37 µs)	fc/2^{9} (9.48 µs)	
		101 (fperiph/32)	fc/2^{6} (1.33 µs)	fc/2^{8} (5.33 µs)	fc/2^{10} (21.3 µs)	
	110 (fc/8)	000 (fperiph/1)	–	–	fc/2^{5} (0.593 µs)	
		001 (fperiph/2)	–	fc/2^{4} (0.296 µs)	fc/2^{6} (1.19 µs)	
		010 (fperiph/4)	–	fc/2^{5} (0.593 µs)	fc/2^{7} (2.37 µs)	
		011 (fperiph/8)	fc/2^{4} (0.296 µs)	fc/2^{6} (1.19 µs)	fc/2^{8} (4.74 µs)	
		100 (fperiph/16)	fc/2^{5} (0.593 µs)	fc/2^{7} (2.37 µs)	fc/2^{9} (9.48 µs)	
		101 (fperiph/32)	fc/2^{6} (1.33 µs)	fc/2^{8} (5.33 µs)	fc/2^{10} (21.3 µs)	
	111 (fc/16)	000 (fperiph/1)	–	–	fc/2^{5} (0.593 µs)	
		001 (fperiph/2)	–	–	fc/2^{6} (1.19 µs)	
		010 (fperiph/4)	–	fc/2^{5} (0.593 µs)	fc/2^{7} (2.37 µs)	
		011 (fperiph/8)	–	fc/2^{6} (1.19 µs)	fc/2^{8} (4.74 µs)	
		100 (fperiph/16)	fc/2^{5} (0.593 µs)	fc/2^{7} (2.37 µs)	fc/2^{9} (9.48 µs)	
		101 (fperiph/32)	fc/2^{6} (1.33 µs)	fc/2^{8} (5.33 µs)	fc/2^{10} (21.3 µs)	

Note 1: The prescaler output clock φTn must be selected so that φTn < fsys is satisfied (so that φTn is slower than fsys).
Note 2: Do not change the clock gear while the timer is operating.
Note 3: "−" denotes a setting prohibited.

13.5.2 Up-counter (UC)

UC is a 16-bit binary counter.

- Source clock

  UC source clock, specified by TBxMOD<TBCLK[2:0]>, can be selected from either seven types - φT1, φT4, φT16, φT32, φT64, φT128 and φT256 - of prescaler output clock or the external clock of the TBxIN0 pin.

- Count start/ stop

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

- Timing to clear UC

  1. When a match is detected

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

  2. When UC stops

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

- UC overflow

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

13.5.3 Timer registers (TBxRG0, TBxRG1)

TBxRG0 and TBxRG1 are registers for setting values to compare with up-counter values and two registers are built into each channel. If the comparator detects a match between a value set in this timer register and that in a UC up-counter, it outputs the match detection signal.

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

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

13.5.4 Capture

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

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

13.5.5 Capture registers (TBxCP0, TBxCP1)

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

13.5.6 Up-counter capture register (TBxUC)

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

13.5.7 Comparators (CP0, CP1)

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

13.5.8 Timer Flip-flop (TBxFF0)

The timer flip-flop (TBxFF0) is reversed by a match signal from the comparator and a latch signal to the capture registers. It can be enabled or disabled to reverse by setting the TBxFFCR<TBC1T1, TBC0T1, TBE1T1, TBE0T1>.

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

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

13.5.9 Capture interrupt (INTCAPx0, INTCAPx1)

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

13.6 Description of Operations for Each Mode

13.6.1 16-bit Interval Timer Mode

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

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits INTTBx interrupt by setting corresponding bit to "1".
TBxFFCR	←	X	X	0	0	0	0	1	1	Disable to TBxFF0 reverse trigger
TBxMOD	←	0	1	0	0	1	*	*	*	Changes to prescaler output clock as input clock. Specifies Capture function to disable.
						(*** = 001 to 111)				
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a time interval. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note: X; Don’t care −; No change

13.6.2 16-bit Event Counter Mode

It is possible to make it the event counter by using an input clock as an external clock (TBxIN0 pin input).

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

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

Note 1: m: corresponding bit of port

Note 2: X; Don’t care

−; No change

13.6.3 16-bit PPG (Programmable Pulse Generation) Output Mode

Square waves with any frequency and any duty (programmable square waves) can be output. The output pulse can be either low-active or high-active.

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

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

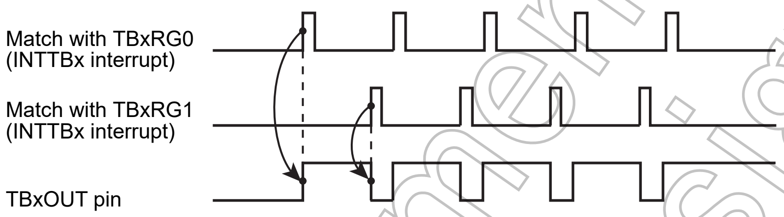


Figure 13-2 Example of Output of Programmable Pulse Generation (PPG)

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

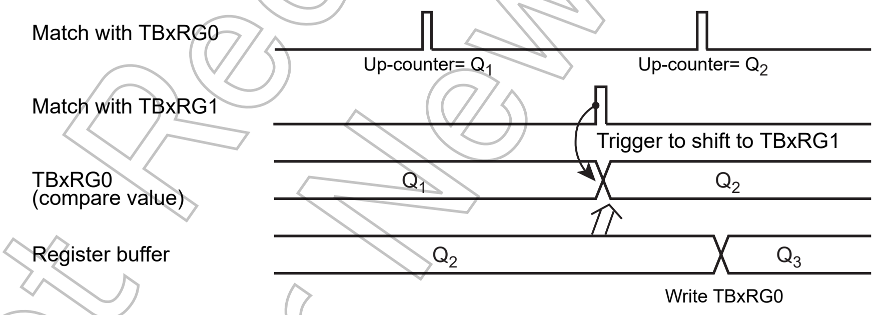


Figure 13-3 Register Buffer Operation

The block diagram of this mode is shown below.

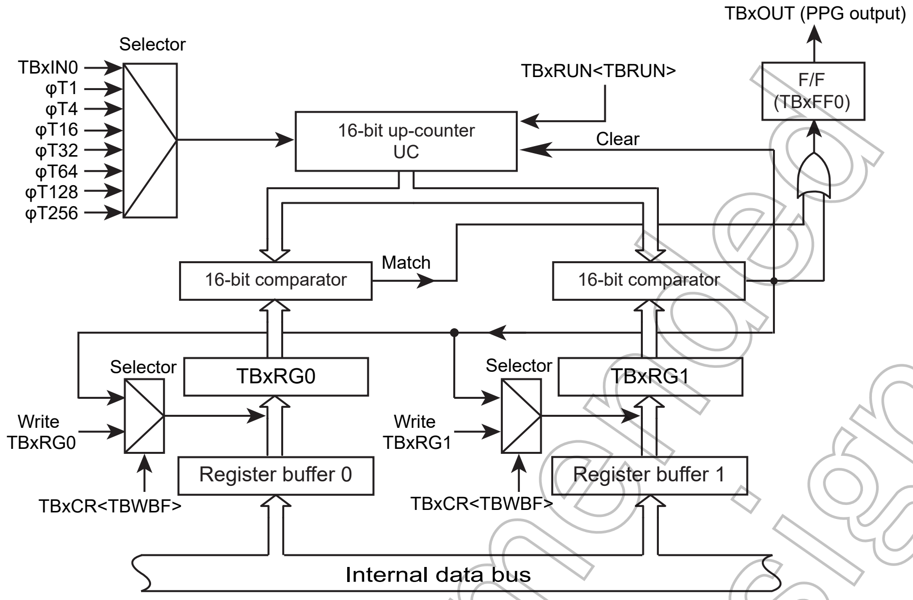


Figure 13-4 Block Diagram of 16-bit PPG Mode

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

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
TBxCR	←	0	0	−	X	−	X	X	X	Disables double buffering.
TBxRG0	←	*	*	*	*	*	*	*	*	Specifies a duty. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a cycle. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxCR	←	1	0	X	0	0	0	0	0	Enables the TBxRG0 double buffering. (Changes the duty/cycle when the INTTBx interrupt is generated)
TBxFFCR	←	X	X	0	0	1	1	1	0	Specifies to trigger TBxFF0 to reverse when a match with TBxRG0 or TBxRG1 is detected, and sets the initial value of TBxFF0 to "0".
TBxMOD	←	0	1	0	0	1	*	*	*	Designates the prescaler output clock as the input clock, and disables the capture function.
							(** = 001 to 111)			
PxCR[m]	←						1			Allocates corresponding port to TBxOUT.
PxFR1[m]	←						1			
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note 1: m: corresponding bit of port

Note 2: X; Don’t care

−; No change

13.6.4 Timer synchronous mode

This mode enables the timers to start synchronously.

If the mode is used with PPG output, the output can be applied to drive a motor.

TMRB is consisted of two pairs of 4-channel TMRB. If one channel starts, remaining 3 channels can be start synchronously. In the TMPM341FDXBG/FYXBG, the following combinations allow to use.

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

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

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

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

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

Note 1: Set TBxCR<TBSYNC> to "0" unless the timer synchronous mode is used. The timer synchronous mode keeps the other channels operation waiting until TMRB0 and TMRB4 start operation.

Note 2: TMRB0 and TMRB4 are the master clocks of the timer synchronous mode. Therefore, their <TBSYNC> bit must be set to "0".

Note 3: This mode cannot be applied to TMRB8 and TMRB9.

13.6.5 External trigger count start mode

The external trigger count start mode can be set to start counting by an external signal.

Set TBxCR<CSSEL> to select the count start mode.

- <CSSEL> = "0" : Start counting according to the timing of each timer channel.
- <CSSEL> = "1" : Start counting by an external signal.

Set TBxCR<TRGSEL> to select the active edge of the external trigger signal.

- <TRGSEL> = "0" : The rising edge of TBxIN0 is selected.
- <TRGSEL> = "1" : The falling edge of TBxIN0 is selected.

Timer synchronous mode has a priority if its mode is specified.

13.7 Applications using the Capture Function

The capture function can be used to develop many applications, including those described below:

1. One-shot pulse output triggered by an external pulse
2. Frequency measurement
3. Pulse width measurement
4. Time difference measurement

13.7.1 One-shot pulse output triggered by an external pulse

One-shot pulse output triggered by an external pulse is carried out as follows:

The 16-bit up-counter is made to count up by putting it in a free-running state using the prescaler output clock. An external pulse is input through the TBxIN0 pin. A trigger is generated at the rising of the external pulse by using the capture function and the value of the up-counter is taken into the capture registers (TBxCP0).

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

In addition, the timer flip-flop control registers(TBxFFCR<TBE1T1, TBE0T1>) must be set to "11". This enables triggering the timer flip-flop (TBxFF0) to reverse when TBxUC matches TBxRG0 and TBxRG1. This trigger is disabled by the INTTBx interrupt after a one-shot pulse is output.

Symbols (c), (d) and (p) used in the text correspond to symbols c, d and p in "Figure 13-5 One-shot Pulse Output (With Delay)".

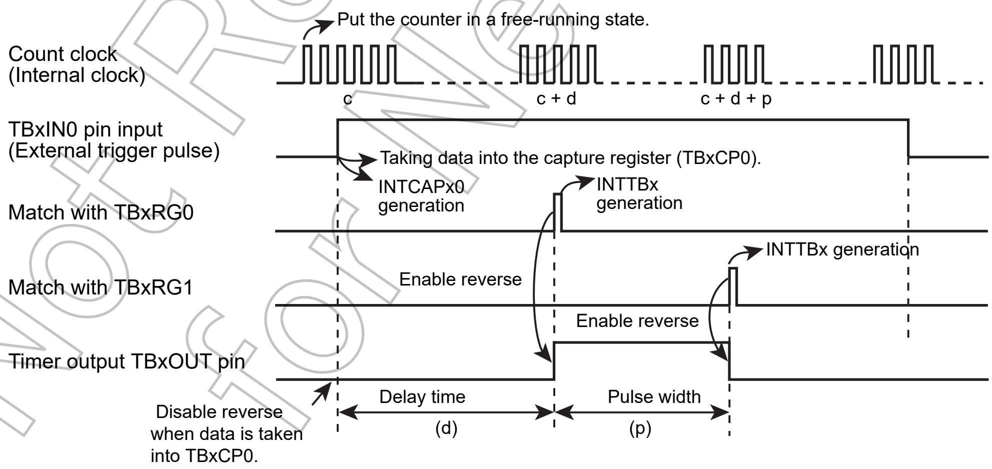


Figure 13-5 One-shot Pulse Output (With Delay)

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

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

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

Note 1: m: corresponding bit of port

Note 2: X; Don’t care
−; No change

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

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

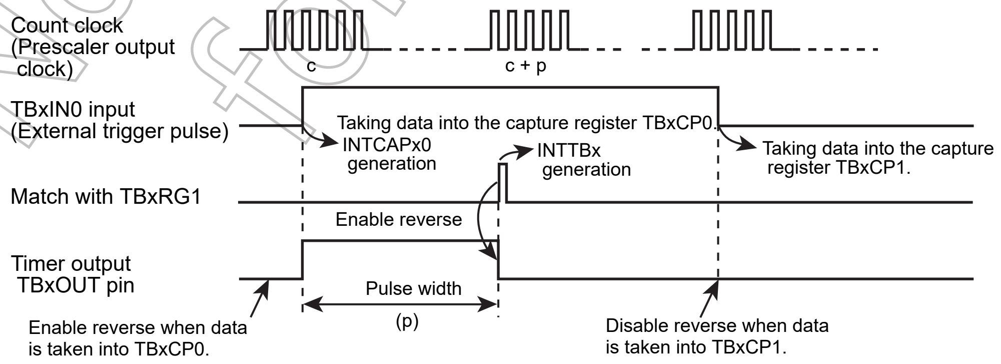


Figure 13-6 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

13.7.2 Frequency measurement

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

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

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

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

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

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

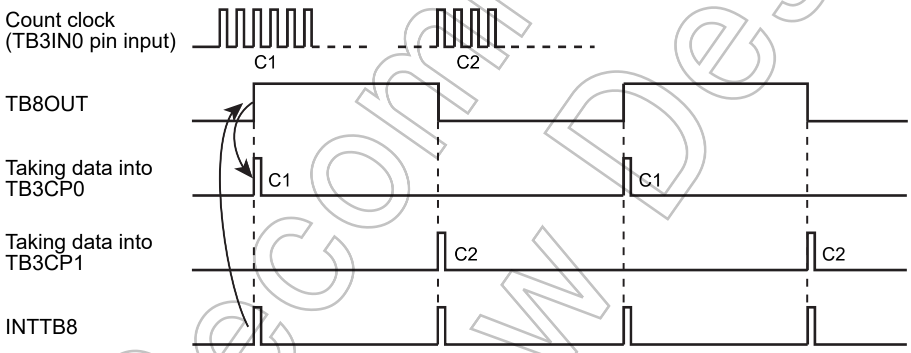


Figure 13-7 Frequency Measurement

13.7.3 Pulse width measurement

By using the capture function, the "High" level width of an external pulse can be measured. Specifically, by putting it in a free-running state using the prescaler output clock, an external pulse is input through the TBxIN0 pin and the up-counter (UC) is made to count up. A trigger is generated at each rising and falling edge of the external pulse by using the capture function and the value of the up-counter is taken into the capture registers (TBxCP0, TBxCP1). The CPU must be programmed so that INTCAPx1 is generated at the falling edge of an external pulse input through the TBxIN0 pin.

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

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

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

The "Low" level width of an external pulse can also be measured. In such cases, the difference between C2 generated the first time and C1 generated the second time is initially obtained by performing the second stage of INTCAPx0 interrupt processing as shown in "Figure 13-8 Pulse Width Measurement" and this difference is multiplied by the cycle of the prescaler output clock to obtain the "Low" level width.

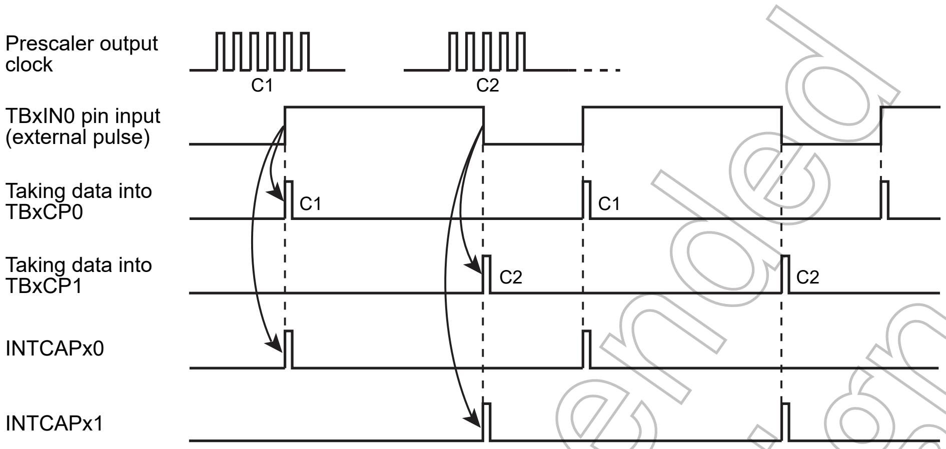


Figure 13-8 Pulse Width Measurement

13.7.4 Time Difference Measurement

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

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

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

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

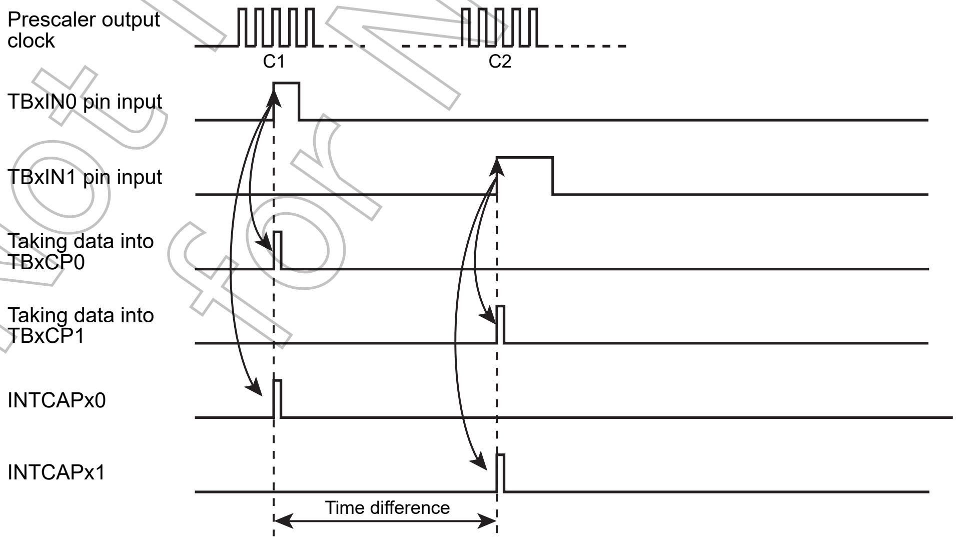


Figure 13-9 Time Difference Measurement

14. Two-phase pulse input counter (PHCNT)

14.1 Outline

The counter is increment or decremented by one depending on the state transition of the two phase pulse that is input through PHCxIN0 and PHCxIN1 and has phase difference. PHCxIN0 and PHCxIN1 have a noise canceller which is selected either enabled and disabled.

Two-phase pulse input counter has two compare register. An interrupt is occurred when the up-and- down counter matches them. And an interrupt is also occurred by increment or decrement the up-and-down counter.

Two-phase pulse input counter has three operation mode, they are controlled by register.

1. Normal operation mode (up/down at the fourth count)
2. Quadruple mode (up/down at each count)
3. Multiplied by two mode
   - PHCxIN0 input
   - PHCxIN1 input

It is available to decide overflow, underflow and compare interrupt by the status register and to make two phase pulse as a trigger signal of the capture in 16-bit timer.

14.2 Differences in the Specifications

TMPM341FDXBG/FYXBG has four channel two-phase pulse input counters.

Each channel (PHCNT0 to PHCNT3) operates independently and the channel in the same way except for the differences in their specification as shown in Table 14-1.

Table 14-1 Differences in the Specifications of PHCNT Modules

Specification channel	External signals	TMRB capture register		
		PHCxIN0↑ PHCxIN1↑	PHCxIN0↑ PHCxIN0↓	PHCxIN1↑ PHCxIN1↓
PHCNT0	PHC0IN0 PHC0IN1	TB2CP0 TB2CP1	TB2CP0 TB2CP1	TB3CP0 TB3CP1
PHCNT1	PHC1IN0 PHC1IN1	TB4CP0 TB4CP1	TB4CP0 TB4CP1	TB5CP0 TB5CP1
PHCNT2	PHC2IN0 PHC2IN1	TB6CP0 TB6CP1	TB6CP0 TB6CP1	TB7CP0 TB7CP1
PHCNT3	PHC3IN0 PHC3IN1	TB8CP0 TB8CP1	TB8CP0 TB8CP1	TB9CP0 TB9CP1

14.3 Configuration

Two-phase pulse input counter is made from up-and-down counter PHCxCNT, compare circuit and counter control circuit. The counting operation is controlled by the registers.

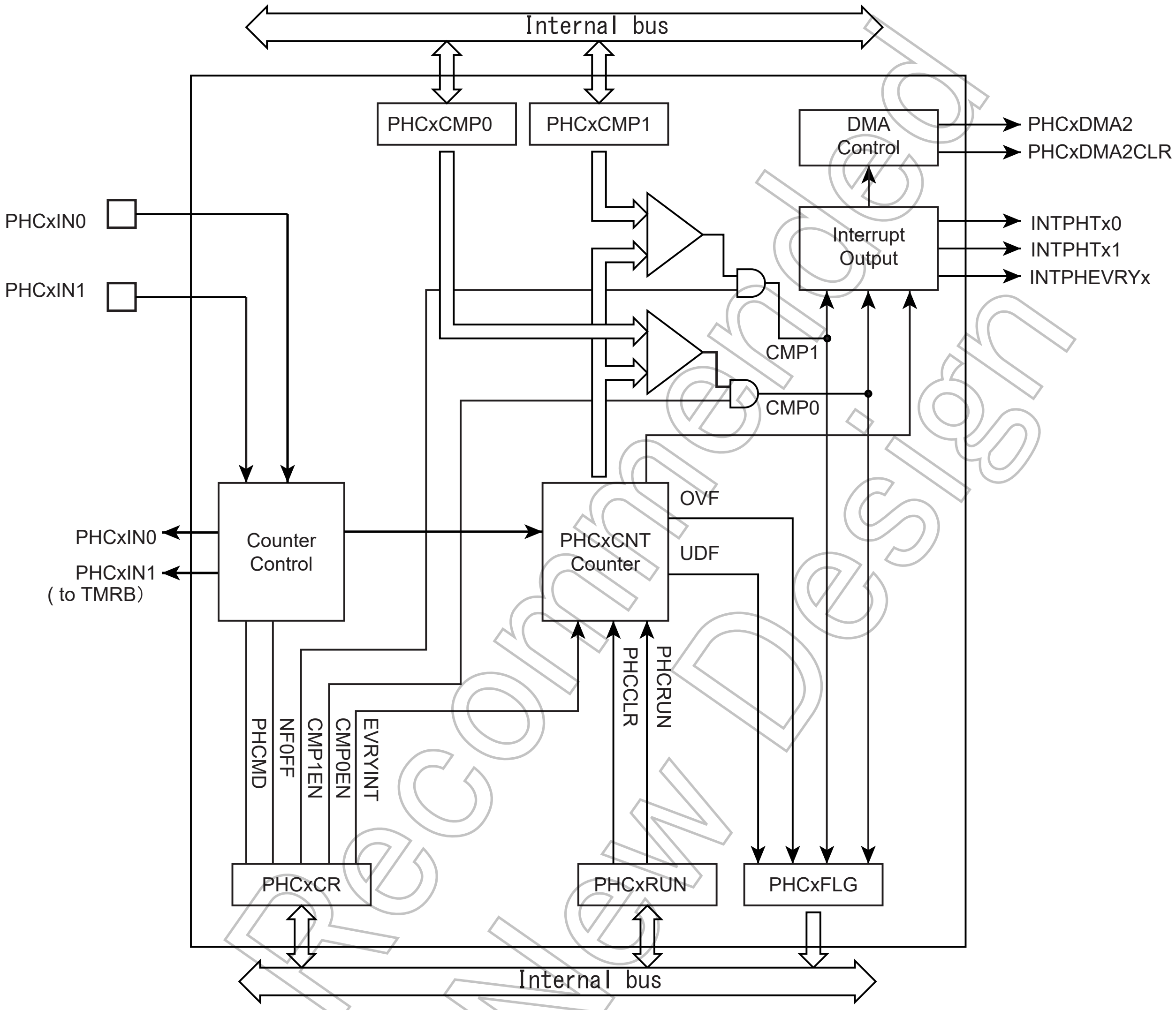


Figure 14-1 Two-phase Pulse Input Counter (PHCNTx) Block Diagram

14.4 Registers

14.4.1 Register List according to channel

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

Channel x	Base Address
Channel0	0x400C_A000
Channel1	0x400C_A100
Channel2	0x400C_A200
Channel3	0x400C_A300

Register name (x=0~1)		Address(Base+)
Timer RUN Register	PHCxRUN	0x0000
Timer Control Register	PHCxCR	0x0004
Timer Enable Register	PHCxEN	0x0008
Timer Status Register	PHCxFLG	0x000C
Timer Compare Register 0	PHCxCMP0	0x0010
Timer Compare Register 1	PHCxCMP1	0x0014
Counter Read Register	PHCxCNT	0x0018
DMA request permitted register	PHCxDMA	0x001C

Note: Do not modify timer control register or timer enable register when timer is operating. Modify them after two-phase pulse counter timer stops.

14.4.2 PHCxRUN (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	-	-	-	-	-	-	PHCCLR	PHCRUN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-2	−	R	Read as 0.
1	PHCCLR	R/W	Clears up-and-down counter value of PHCxCNT register to 0x7FFF 0: Don’t care 1: Clear
0	PHCRUN	R/W	Controls up-and-down counter PHCxCNT counting operation 0: Stop 1: Active

14.4.3 PHCxCR (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	-	-	EVRYINT	CMP1EN	CMP0EN	NFOFF	PHCMD	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-6	−	R	Read as 0.
5	EVRYINT	R/W	Interrupt at every count 0 : Disable 1 : Enable Enables to prohibit interrupt at every counting operation when using compare interrupt.
4	CMP1EN	R/W	Compare interrupt 1 0 : Disable 1 : Enable Interrupt is generated when up-and-down counter PHCxCNT matches compare register 1.
3	CMP0EN	R/W	Compare interrupt 0 0 : Disable 1 : Enable Interrupt is generated when up-and-down counter PHCxCNT matches compare register 0.
2	NFOFF	R/W	Noise filter 0 : ON 1 : OFF Controls noise cancellation. Noise filter value is 20ns (typ.).
1-0	PHCMD[1:0]	R/W	Sets operation mode 00: Normal mode 01: Quadruple mode 10: Multiplied by two mode (PHCxIN0 input) 11: Multiplied by two mode (PHCxIN1 input)

Note: To change mode when two-phase input counter operates is prohibited.

14.4.4 PHCxEN (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	-	-	-	-	-	-	-	PHCEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-1	−	R	Read as 0.
0	PHCEN	R/W	PHCNT operation 0: Disable 1: Enable Specified the PHCNT operation. To use the PHCNT operation, set <PHCEN> = "1". When the operation is disabled, no clock is supplied to the other registers in the PHCNT module. This can reduce the power consumption. If the PHCNT operation is executed and then disabled, the setting will be maintained in each register.

14.4.5 PHCxFLG (Timer Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	UDF	OVF	CMP1	CMP0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-4	−	R	Read as 0.
3	UDF	R/W	underflow interrupt 0: not occurs 1: occurs Underflow interrupt of up-and-down counter
2	OVF	R/W	overflow interrupt 0: not occurs 1: occurs Overflow interrupt of up-and-down counter
1	CMP1	R/W	Compare 1 match interrupt 0: not occurs 1: occurs Match interrupt with compare register 1 (PHCxCMP1)
0	CMP0	R/W	Compare 0 match interrupt 0: not occurs 1: occurs Match interrupt with compare register (PHCxCMP0)

Note:The flag is cleared by writing "1" to each bit. Because each flag is not cleared automatically, initialize before using them.

14.4.6 PHCxCMP0 (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	PHCCMP0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PHCCMP0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-16	−	R	Read as 0.
15-0	PHCCMP0 [15:0]	R/W	Sets compare value 0x0000 to 0xFFFF

14.4.7 PHCxCMP1(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	PHCCMP1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PHCCMP1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-16	−	R	Read as 0.
15-0	PHCCMP1 [15:0]	R/W	Sets compare value 0x0000 to 0xFFFF

14.4.8 PHCxCNT (Timer Read Register)

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

Bit	Bit Symbol	Type	Description
31-16	−	R	Read as 0.
15-0	PHCCNT[15:0]	R	Data read from the up-and-down counter

Note: Pulse is counted not synchronized with MCU operation clock. Because there is a possibility that data is read while rewriting, depending on the timing, reading out twice is recommended. In this case if the data is different read out data again.

Note: When PHCxRUN register is "0" it is initialized.

14.4.9 PHCxDMA (DMA request permitted register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PHCDMA2	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-3	−	R	Read as 0.
2	PHCDMA2	R/W	Controls DMA request permitted 0: Disable 1: Enable Controls DMA request permitted for every interrupt of up-and-down counting.
1-0	−	R/W	Write "0".

Note: When interrupt request is disabled, DMA request is not occurred even if DMA request is enabled.

14.5 Operation in Each Mode

The channel in the same way except for the differences in their specification as shown in Table 14-1.

14.5.1 Count operation mode

The count operation mode is selected normal mode, quadruple mode or multiplied by two mode by the PHCxCR<PHCMD[1:0]>.

The up-and-down counter is increment or decremented by one depending on the state transition of the two- phase pulse that is input through PHCxIN0 and PHCxIN1 and has phase difference.

PHCNT has two compare registers, an interrupt is occurred when the up-and-down counter matches a compare register. And an interrupt is occurred at every counting.

The state transition of the two-phase pulse is shown as below.

Table 14-2 The state of the two-phase input in normal mode and quadruple mode

PHCxIN1	PHCxIN0	State
0	0	0
0	1	1
1	0	2
1	1	3

The count operation of each mode is described below.

1. Normal mode

   In normal mode, the up-and-down counter is increment or decrement when the state transition of the two-phase pulse changes at fourth count.

   To avoid the miss counting, up-and-down counter is not increment or decrement until the set state is input after the clear state is input.

- Count up

When "2" is input at the previous clock and the current state is "3", the up-and-down counter is increment.

When "3" is input at the previous clock and the current state is "2", it is the clear state.

When "3" is input at the previous clock and the current state is "1", it is the set state.

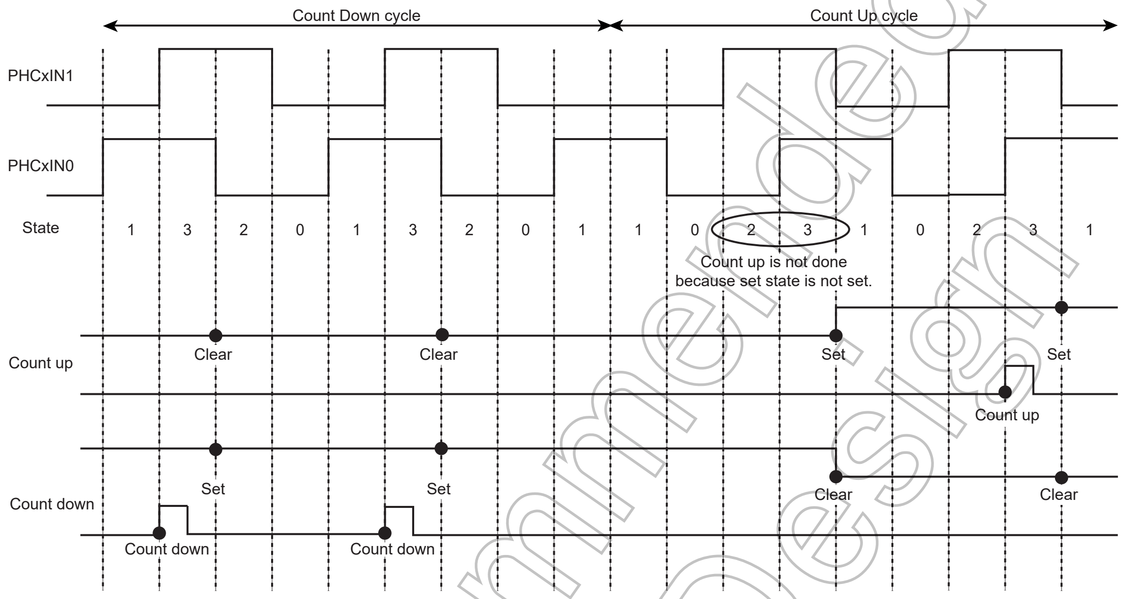


Figure 14-2 Count up in normal mode

- Count down

When "1" is input at the previous clock and the current state is "3", the up-and-down counter is decrement.

When "3" is input at the previous clock and the current state is "1", it is the clear state.

When "3" is input at the previous clock and the current state is "2", it is the set state.

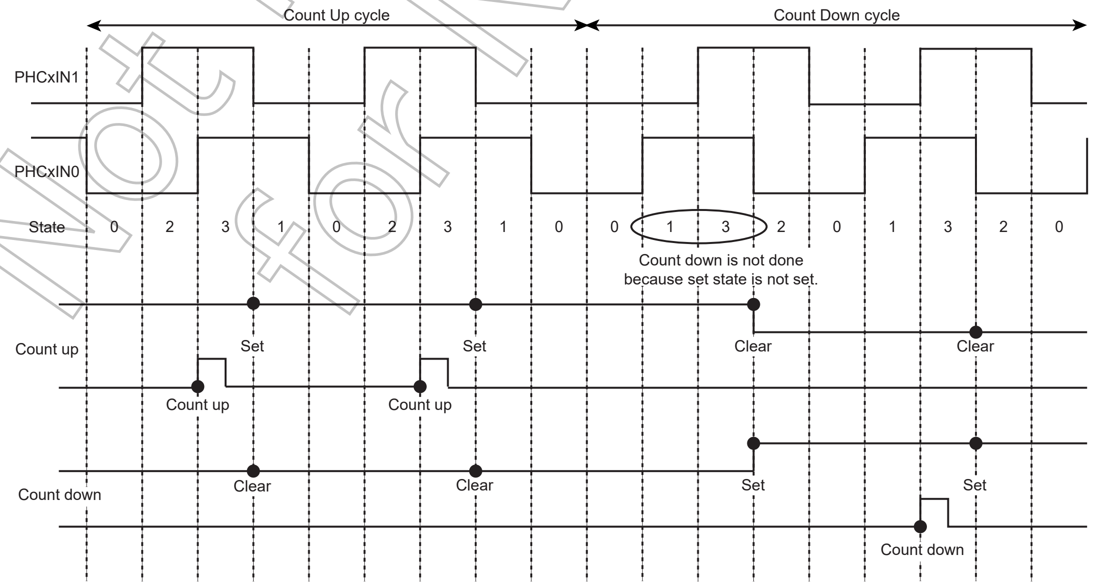


Figure 14-3 Count down in normal mode

2. Quadruple mode

In quadruple mode, the up-and-down counter is increment or decrement when the state transition of the two-phase pulse changes once.

- Count up

When "3" is input at the previous clock and the current state is "1", the up-and-down counter is increment.

When "1" is input at the previous clock and the current state is "0", the up-and-down counter is increment.

When "0" is input at the previous clock and the current state is "2", the up-and-down counter is increment.

When "2" is input at the previous clock and the current state is "3", the up-and-down counter is increment.

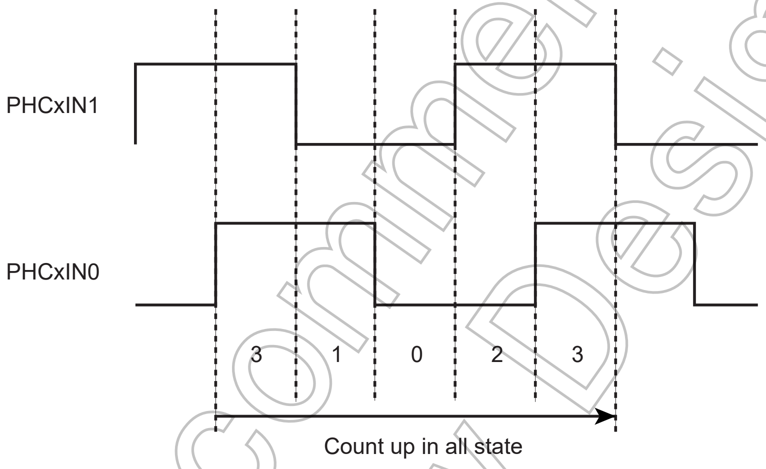


Figure 14-4 Count up in quadruple mode

- Count down

When "3" is input at the previous clock and the current state is "2", the up-and-down counter is decrement.

When "2" is input at the previous clock and the current state is "0", the up-and-down counter is decrement.

When "0" is input at the previous clock and the current state is "1", the up-and-down counter is decrement.

When "1" is input at the previous clock and the current state is "3", the up-and-down counter is decrement.

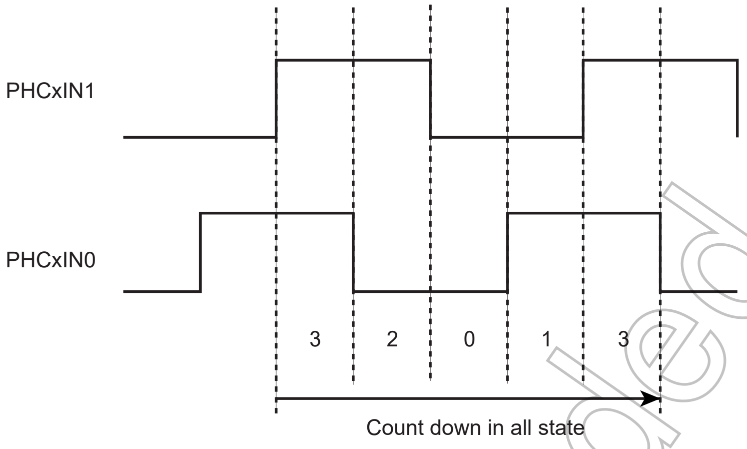


Figure 14-5 Count down in quadruple mode

3. Multiplied by two mode

   In multiplied by two mode, the up-and-down counter is increment when the state transition of either PHCxIN0 or PHCxIN1 changes once.

   - Count up

     When "0" is input at the previous clock and the current state is "1", the up-and-down counter is increment.

     When "1" is input at the previous clock and the current state is "0", the up-and-down counter is increment.

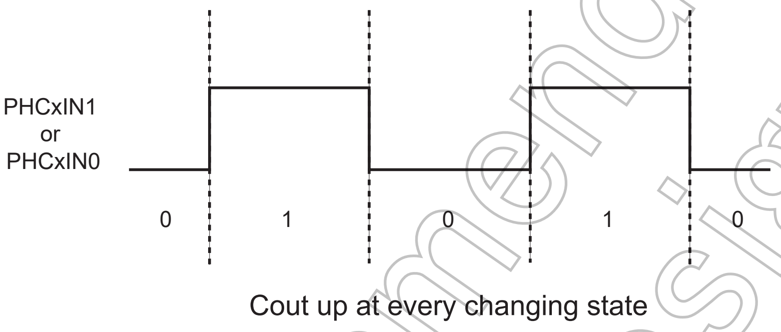


Figure 14-6 Count up in multiplied by two mode

   - Count down

     In this mode, there is not count down state.

14.5.2 Interrupt

The PHCNTx interrupt is enabled using the interrupt controller (NVIC).

1. Interrupt every count

   INTPHEVRYx is occurred by increment or decrement.

   In the interrupt service routine, it is possible to check for occurrences of an overflow and an underflow. If PHCxFLG<OVF> is "1", an overflow is occurred. If <UDF> is "1", an underflow is occurred. This register is cleared after it is written to "1".

2. Compare interrupt

   INTPHTx0/1 is occurred by match up-and-down counter and compare register 0/1.

   In the interrupt service routine, it is possible to check the register which is matched. If PHCxFLG<CMP0> is "1", the up-and-down counter matches a compare register 0. If <CMP1> is "1", the up-and-down counter matches a compare register 1. This register is cleared after it is written to "1".

14.5.3 Up-and-down counter

When the two-phase input count is started (PHCxRUN<PHCRUN> = "1"), the up-and-down counter value is counted from 0x7FFF which is an initial value. And the up-and-down counter value is initialized to 0x7FFF, when the PHCxRUN<PHCCLR> is set to "1".

If a counter overflow occurs, the counter returns to 0x0000. If a counter underflow occurs, the counter returns to 0xFFFF. After that, the counter continues the counting operation.

Therefore, overflow and underflow can be checked by reading the counter value and the status flag PHCxFLG after an interrupt is occurred.

Note: The reading the up-and-down counter should be read in a interrupt service routine of INTPHEVRYx.

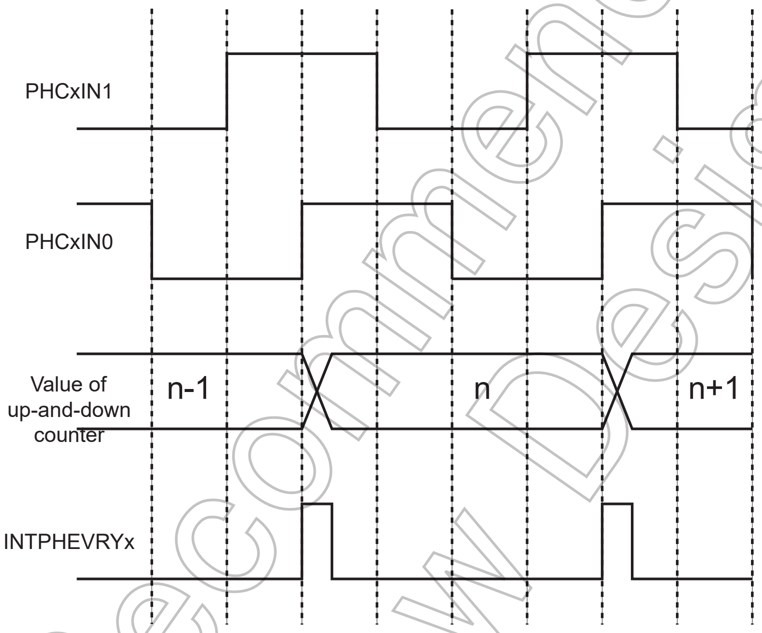


Figure 14-7 Up count interrupt timing in Normal mode

15. High Resolution 16-bit Timer/PPG Outputs (TMRD)

15.1 Outline

TMRD consists of two timer units (TMRD0 and TMRD1) and two clock setting circuits (prescalers) for supplying clocks to these timer units. The functions are as follows.

- 16-bit interval timer
- 16-bit programmable pulse generation (PPG)

16-bit interval timer has the following two modes.

- Timer mode that TMRD0 and TMRD1 operate independently.
- Interlock timer mode that can start TMRD0 and TMRD1 at the same time.

16-bit programmable pulse generation has the following two modes.

- PPG mode that TMRD0 and TMRD1 independently output preprogrammed pulses.
- PPG mode that can change the phase releations between the pilse output by TMRD0 and that output by TMRD1 in the range from -180 degree to +180 degree.

TMRD consists of the clock setting circuit and two unit timers.

Figure 15-1 shows the clock setting and Figure 15-2 shows the block diagram of the timer unit.

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

15.2 Block Diagram

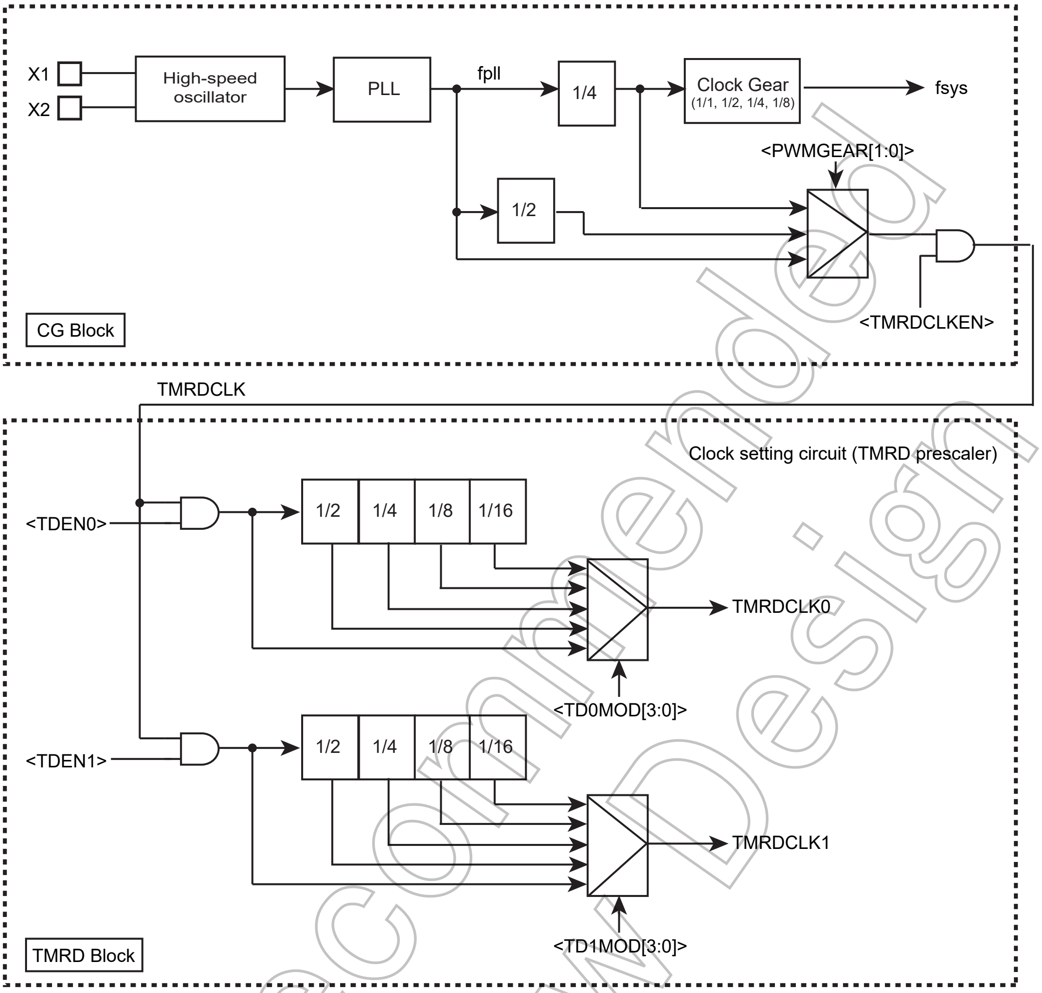


Figure 15-1 TMRD Clock Diagram

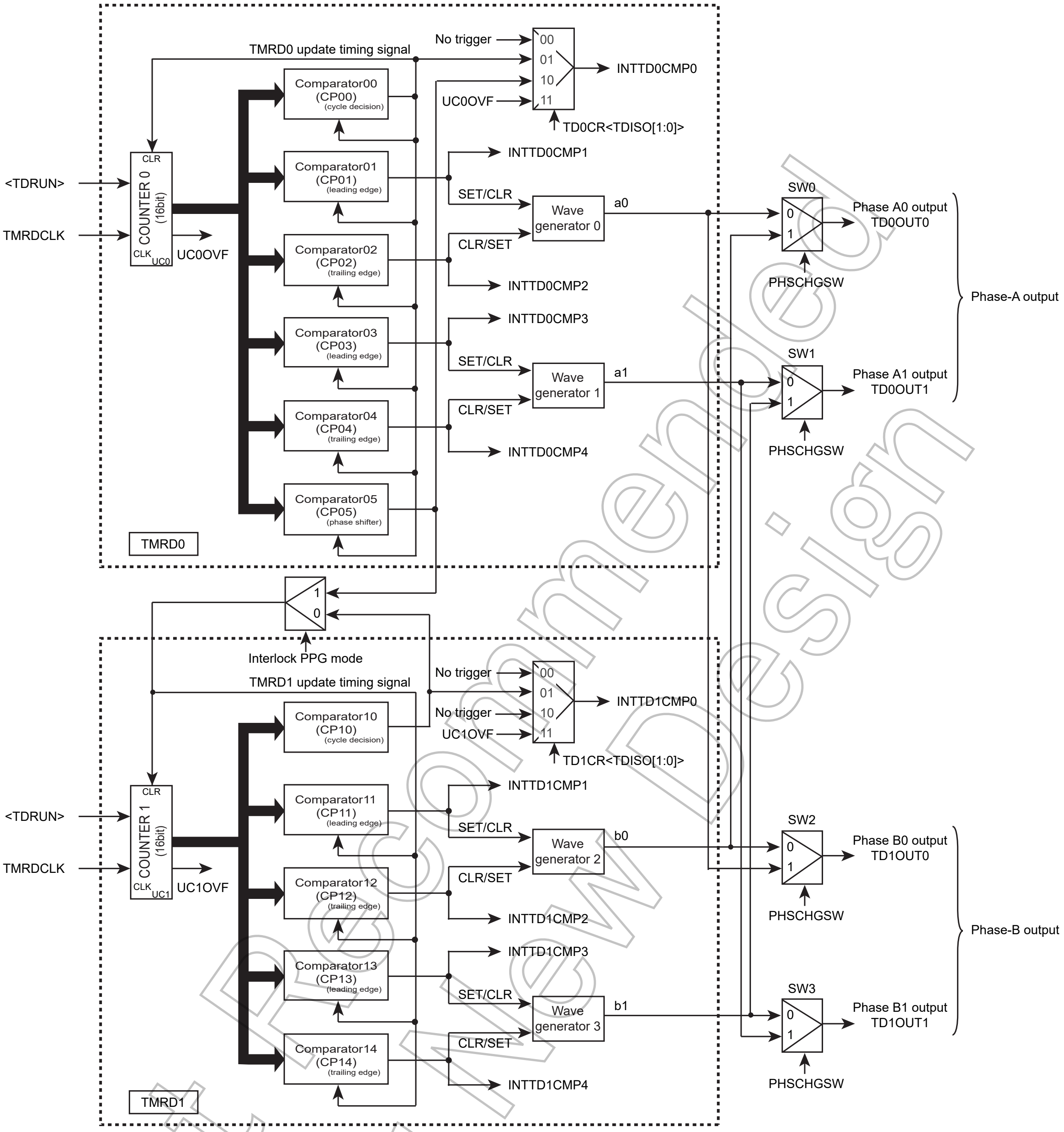


Figure 15-2 TMRD Unit Block Diagram

15.2.1 Prescaler clock

TMRDCLK[1:0] is a prescaler clock to get the TMRD to work. After setting the input clock for the PLL circuit to the register CGPWMGEAR[5:4], the prescaler clock TMRDCLK[1:0] selects the source clock. The prescaler clock TMRD0 is set into the clock setting register TD0MOD[3:0] in the TMRD block, TMRD1 is set into the clock setting register TD1MOD[3:0].

Note: The maximum operating frequency is able to be provided to the TMRD is 160MHz. The configuration must not exceed this value.

15.2.2 Timer unit (TMRD0, TMRD1)

As shown in the Figure 15-2, the timer unit (TMRD0) consists of the counter (UC0), six comparators (CP0m, m =0 to 5) and two waveform generating circuits (0 and 1). The other timer unit (TMRD1) consists of the counter (UC1), five comparators (CP1m, m=0 to 4) and two waveform generating circuits (2 and 3).

1. Counters (UC0/UC1)

   UC0 and UC1 are the 16-bit binary counters which are counted up by the clocks (TMRDCLK0/TMRDCLK1) output from the clock setting unit. These binary counters operate as free-run counters according to the values set in the registers TD0MOD<TDCLE>/TD1MOD<TDCLE>, or operate as the counters which are zero cleared if they match the value of the counter CP00 and the counter CP10. However, in interlock PPG mode, the counter UC1 operate as a counter that is zero cleared when the value of the UC1 matches that of the comparator CP05, not the value of the comparator CP10.

UC0 :	TD0MOD<TDCLE> = 0	Free-run counter
	TD0MOD<TDCLE> = 1	A counter that is zero cleared when the values of the binary counter UC0 and the counter CP00 match.
UC1 :	TD1MOD<TDCLE> = 0	Free-run counter
	TD1MOD<TDCLE> = 1	A counter that is zero cleared when the value of the binary counter UC1 matches that of CP10 or CP05.

   The binary counter UC0 and UC1 control the start and stop of the counters using the control signal TD0RUN and TD1RUN.

UC0 :	TD0RUN<TDRUN>= 0 :	Counter Stop. Zero cleared.
	TD0RUN<TDRUN> = 1 :	Operation of the counter starts.
UC1 :	TD1RUN<TDRUN> = 0 :	Counter Stop. Zero cleared.
	TD1RUN<TDRUN> = 1 :	Operation of the counter starts.

2. Comparators (CPxm : x=0,1, x=0 : m=0 to 5, x=1 : m=0 to 4)

   As shown in the Figure 15-3, comparators (CPxm) consist of a 16-bit width timer register (TDxRGm), 16-bit width compae register (TDxCPm) and a circuit to detect a match between the output value of the counter UCx and the values set in the TDxCPm (CPRGm[15:0]).

These comparators have a double-buffer scheme, which consists of TDxRGm and TDxCPm, and a path to write data to TDxCPm is selected according to the value of the register TDxCR<TDRDE>.

TDxCR<TDRDE> = 0 : A value written to the TDxRGm is written to the TDxCPm at the same time.

However, since the register TDxCPm is a read-only register, direct write to this address with specified address is not possible. (The initial value can be set into the TDxCPm at random timing.)

TDxCR<TDRDE> = 1 : If the condition is that the update enable flag TDBCRx<TDSFT> = 1, a value of the TDxRGm is written into the TDxCPm at the timing shown below.

Update timing TMRD0 : At the timing of a detection signal output when the value matches the CP00.

TMRD1 : At the timing of a detection signal output when the value matches the CP10.

However, in the interlock PPG mode, at the timing of a detection signal output when the value matches the CP05.

Figure 15-4 shows the waveform of the timing for writing a value of the TDxRGm to the TDxCPm when TDxCR<TDRDE> = 1.

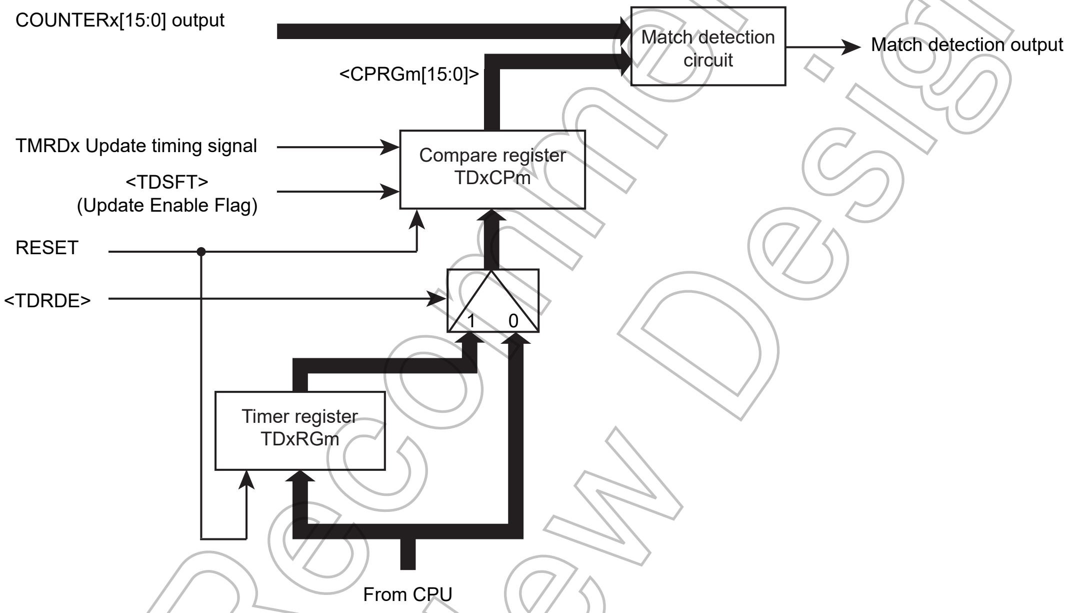


Figure 15-3 Comparator (CPxm) Configuration

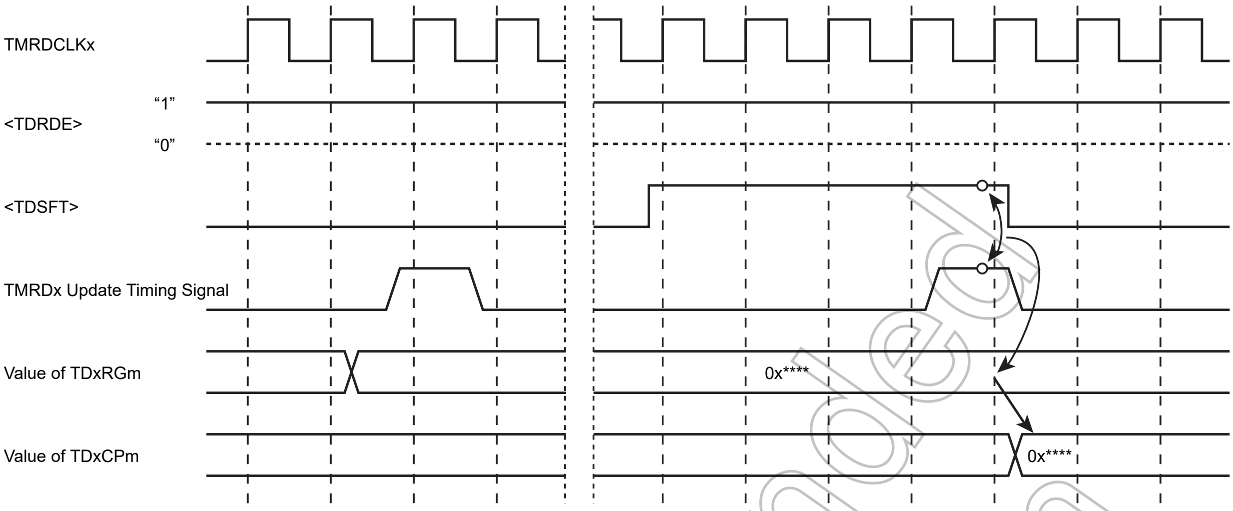


Figure 15-4 Timing for writing a value of TDxRGm to TDxCPm (<TDRDE>=1)

3. Waveform generating circuit

The waveform generating circuits are pulse output circuits. They generate a leading edge signal at the timing a match is respectively detected for CP01/CP03/CP11/CP13, and a trailing edge signal at the timing a match is respectively detected for CP02/CP04/CP12/CP14. The polarity (rising/falling) of leading edge and trailing edge can be set into the register TD0MOD<TDIV[1:0]> and the register TD1MOD<TDIV[1:0]>.

15.3 Registers

15.3.1 Register list according to channel

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

Channel x	Base Address
Channel0	0x4005_8000
Channel1	0x4005_8100

Register name(x=0 to 1)		Address(Base+)
Timer RUN register	TDxRUN	0x0000
Timer Control register	TDxCR	0x0004
Timer Enable register	TD0EN (Note1)	0x0050
Timer Configuration register	TD0CONF (Note1)	0x0054
Timer Mode register	TDxMOD	0x0008
Update flag set register	TDxBCR	0x000C
DMA request enable register	TDxDMA	0x0010
Timer register 0	TDxRG0	0x0014
Timer register 1	TDxRG1	0x0018
Timer register 2	TDxRG2	0x001C
Timer register 3	TDxRG3	0x0020
Timer register 4	TDxRG4	0x0024
Timer register 5	TD0RG5 (Note1)	0x0028
Compare register 0	TDxCP0	0x002C
Compare register 1	TDxCP1	0x0030
Compare register 2	TDxCP2	0x0034
Compare register 3	TDxCP3	0x0038
Compare register 4	TDxCP4	0x003C
Compare register 5	TD0CP5 (Note1)	0x0040
Reserved	-	0x0060

Base Address = 0x400F_3000

Register name		Address(Base+)
Timer clock setup register	CGPWMGEAR	0x0014

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

Note: Access the registers by using word reads and word writes.

Note 1: These registers do not have channel 1.

15.3.2 CGPWMGEAR (Timer clock setup register)

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

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as 0.
5-4	PWMGEAR [1:0]	R/W	Select a source clock to input in the TMRD block. 00 : fpll 01 : fpll / 2 10 : fpll / 4 11 : Reserved
3-1	−	R	Read as 0.
0	TMRDCLKEN	R/W	Disable or enable TMRDCLK to be provided to the TMRD. 0 : Stop (OFF) 1 : Operation (ON)

Note: Clocks cannot be selected and provided at the same time. To use TMRD, select a source clock at <PWMGEAR[1:0]> in the condition that a clock is disabled<TMRDCLKEN>="0". Then, Set "1" to <TMRDCLKEN> to start providing a clock.

Note: If you try to change the setting of <PWMGEAR1:0>, set the TMRD first (<TMRDCLKEN>="0">.

15.3.3 TD0EN (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 input to the TMRD1 is configured (ON/OFF). 0: Stop (OFF) 1: Operation (ON)
6	TDEN0	R/W	Clock input to the TMRD0 is configured (ON/OFF). 0: Stop (OFF) 1: Operation (ON)
5	TDHALT	R/W	Settings for the operation if a HALT instruction is executed during debugging. 0: Stop (Only an up counter stops) 1: Operation (An up counter does not stop)
4-0	−	R	Read as 0.

15.3.4 TD0CONF (Timer 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	TDI2TD1	TDI2TD0	-	-	-	TMRDMOD		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	TDI2TD1	R/W	TMRD1 operation settings (ON/OFF) during the IDLE mode. 0: Stop (OFF) 1: Operation (ON)
6	TDI2TD0	R/W	TMRD0 operation settings (ON/OFF) during the IDLE mode. 0: Stop (OFF) 1: Operation (ON)
5-3	−	R	Read as 0.
2-0	TMRDMOD [2:0]	R/W	Operation mode settings for TMRD1 and TMRD0. 000 : TMRD0: Timer mode, TMRD1: timer mode 001 : TMRD0: Timer mode, TMRD1: PPG mode 010 : TMRD0: PPG mode, TMRD1: Timer mode 011 : TMRD0: PPG mode, TMRD1: PPG mode 100 : Timer mode that allows TMRD0 and TMRD1 to start simultaneously. 101 : Setting is prohibited. 110 : Setting is prohibited. 111 : Interlock PPG mode that allows TMRD0 and TMRD1 to operate in tandem. (Phase relationships between pulse generated by TMRD1 and TMRD0 can be changed.)

Note: In the condition of <TMRDMOD[2:0]>="111", TMRDCLK0 and TMRDCLK1 cannot be configured respectively; TMRDCLK1 and TMRDCLK0 have the same frequencies.

15.3.5 TD0MOD (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	Initial settings of leading edge/trailing edge of a signal a1 0: Rises when a value set in the CP03 matches a value in the counter, and falls when a value in the CP04 matches a value in the counter. 1: Falls when a value set in the CP03 matches a value in the counter, and rises when a value in the CP04 matches a value in the counter.
6	TDIV0	R/W	Initial settings of leading edge/trailing edge of a signal a0 0: Rises when a value set in the CP01 matches a value in the counter, and falls when a value in the CP02 matches a value in the counter. 1: Falls when a value set in the CP01 matches a value in the counter, and rises when a value in the CP02 matches a value in the counter.
5	−	R	Read as 0.
4	TDCLE	R/W	Selects the COUNTER0 (UC0) operation when the match signal of CP00 is generated. 0: Operate as a free-run counter even if a match is detected. 1: Zero cleared if a match is detected.
3-0	TDCLK[3:0]	R/W	Selects a prescaler of TMRD0 (a frequency of TMRDCLK0). 0000: ftmrd 1000: ftmrd/2 1001: ftmrd/4 1010: ftmrd/8 1011: ftmrd/16 Settings other than those shown above are prohibited.

Note:In the PPG mode, the setting of <TDCLE>="0" becomes invalid. (Not operate as a free-run counter.)

15.3.6 TD1MOD (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	Initial settings of leading edge/trailing edge of a signal b1 0: Rises when a value set in the CP13 matches a value in the counter, and falls when a value in the CP14 matches a value in the counter. 1: Falls when a value set in the CP13 matches a value in the counter, and rises when a value in the CP14 matches a value in the counter.
6	TDIV0	R/W	Initial settings of leading edge/trailing edge of a signal b0 0: Rises when a value set in the CP11 matches a value in the counter, and falls when a value in the CP12 matches a value in the counter. 1: Falls when a value set in the CP11 matches a value in the counter, and rises when a value in the CP12 matches a value in the counter.
5	−	R	Read as 0.
4	TDCLE	R/W	Selects the COUNTER1 (UC1) operation when the match signal of CP10 is generated. 0: Operate as a free-run counter even if a match is detected. 1: Zero cleared if a match is detected.
3-0	TDCLK[3:0]	R/W	Selects a prescaler of TMRD1 (a frequency of TMRDCLK1). 0000: ftmrd 1000: ftmrd/2 1001: ftmrd/4 1010: ftmrd/8 1011: ftmrd/16 Settings other than those shown above are prohibited.

Note: In the PPG mode, the setting of <TDCLE>="0" becomes invalid. (Not operate as a free-run counter.) Also in the interlock PPG mode, the settings of <TDCLE> bit become invalid.

Note: In the interlock PPG mode, the values set in the TD0MOD are selected as the values for <TDCLK[3:0] >.

15.3.7 TD0CR (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	-	-	-	-	-	TDRDE	TDISO	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	TDRDE	R/W	Sets a write path to the compare register (TD0CPm) of the TMRD0. (m = 0, 1, 2, 3, 4, 5) 0: Directly written to the compare register by CPU instructions. At the time when data is written into the timer register (TD0RGm), the same value is written to the corresponding compare register (TD0CPm) simultaneously. In this case, it is not neccesary for an enable flag to update of TD0BCR<TDSFT> to set to "1". 1: Write via the timer register (TD0RGm) of the TMRD0. TD0BCR<TDFST>="1" is required. In the timer mode: TD0MOD<TDCLE>="0": A value in the compare register (TD0CPm) is updated to the value of the timer register (TD0RGm) when the COUNTER0 (UC0) overflows. TD0MOD<TDCLE>="1": A value in the compare register (TD0CPm) is updated to the value of the timer register (TD0RGm) when the value set in the comparator 00 (CP00) matches the value in the counter. In the PPG mode or the interlock PPG mode: When the value set in the comparator 00 (CP00) matches the value in the counter, the value in the compare register (TD0CPm) is updated to the value in the timer register (TD0RGm).
1-0	TDISO[1:0]	R/W	Selects the interrupt factor of INTTD0CMP0. 00: No interrupt factor. 01: A match signal from CP00 10: A match signal from CP05. 11: Overflow of COUNTER0 (UC0). (In the PPG mode, this setting is invalid as an interrupt factor.)

15.3.8 TD1CR (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	-	-	-	-	-	TDRDE	TDISO	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	TDRDE	R/W	Sets a write path to the compare register (TD1CPm) of the TMRD1. (m = 0, 1, 2, 3, 4) 0: Directly written to the compare register by CPU instructions. At the time when data is written into the timer register (TD1RGm), the same value is written to the corresponding compare register (TD1CPm) simultaneously. In this case, it is not neccesary for an enable flag to update of TD1BCR<TDSFT> to set to "1". 1: Write via the timer register (TD1RGm) of the TMRD1. TD1BCR<TDFST>="1" is required. In the timer mode: TD1MOD<TDCLE>="0": A value in the compare register (TD1CPm) is updated to the value of the timer register (TD1RGm) when the COUNTER1 (UC1) overflows. TD0MOD<TDCLE>="1": A value in the compare register (TD1CPm) is updated to the value of the timer register (TD1RGm) when the value set in the comparator 10 (CP10) matches the value in the counter. In the PPG mode or the interlock PPG mode: In the PPG mode, when the value set in the comparator 10 (CP10) matches the value in the counter, the value in the compare register (TD1CPm) is updated to the value of the timer register (TD1RGm). In the interlock mode, the value set in the comparator 05 (CP05) matches the value in the compare register, the value of the compare register (TD1CPm) is updated to the value of the timer register (TD1RGm).
0	TDISO[1:0]	R/W	Selects the interrupt factor of INTTD1CMP0. 00: No interrupt factor. 01: CP10 match signal (Invalid as an interrupt factor in the interlock PPG mode) 10: No interrupt factor. 11: Overflow of COUNTER1 (UC1). (In the PPG mode, this setting is invalid as an interrupt factor.)

Note: In the interlock PPG mode, the values set in the TD0CR are selected as the values for <TDRDE>.

15.3.9 TDxRUN (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	Controls the count operation of TMRDx. 0: Stops the count operation of TMRDx. (Stops the operation of COUNTERx (UCx) and zero clears it.) 1: Starts the count operation of TMRDx. (Starts the operation of COUNTERx (UCx) (count up).)

Note: As for TD1RUN<TDRUN>, the settings become invalid in both the interlock timer mode and interlock PPG mode, and starts operation in tandem with COUNTER0 (UC0).

15.3.10 TD0BCR (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	TDSFT
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	PHSCHG	R/W	Sets the phase relation (delay/fast) of the phase B to the phase A. 0: Delay or same phase 1: Fast or same phase
0	TDSFT	W	An enable flag to update the compare register with the value of the timer register TMRD0. When the data in the compare register is updated, an update enable flag bit <TDSFT> is cleared. 0: Invalid 1: Update enabled

Note: For the update timing of <TDSFT>, please refer to the section writing about TD0CR<TDRDE>.

Note: <PHSCHG> is valid only in the interlock PPG mode. (The output of the phase A and the phase B cannot be switched in the timer mode, the interlock timer mode and the PPG mode.)

15.3.11 TD1BCR (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	-	-	-	-	-	-	-	TDSFT
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	−	R/W	Always write 0.
0	TDSFT	W	An enable flag to update the compare register with the value of the timer register TMRD1. When the data in the compare register is updated, an update enable flag bit <TDSFT> is cleared. 0: Invalid 1: Update enabled

Note: For the update timing of <TDSFT>, please refer to the section writing about TD1CR<TDRDE>.

Note: In the interlock PPG mode, when TD0BCR<TDSFT> is written to "1", TD1BCR<TDSFT> is set to "1" at the same time. Do not write to this register. And this bit is zero cleared when a match of compare register 05 (CP05) is detected.

15.3.12 TDxDMA (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	Sets the DMA request enable/disable. (INTTDxCMP0 is a factor to generate a DMArequest.) 1: Enable 0: Disable

15.3.13 TDxRG0 (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	The timer register TDxRG0 is double-buffered with the compare register TDxCP0. The setting of the compare register is performed on the timer register.

15.3.14 TDxCP0 (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	Frequency cycle setting register: The minimum setting value is 0x0001 and the maximum setting value is 0xFFFF. In the 16-bit interval timer: Sets the match timing of CPx0 or sets the timer frequency. In the 16-bit PPG: Sets the frequency of the pulse output. However, in the interlock PPG mode, TD1CP0 is not used as a frequency cycle setting register.

15.3.15 TDxRG1 (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	The timer register TDxRG1 is double-buffered with the compare register TDxCP1. The setting of the compare register is performed on the timer register.

15.3.16 TDxCP1 (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 register: In the 16-bit interval timer: Sets a timing to generate an interrupt signal INTTDxCMP1. In the 16-bit PPG: Sets a timing of a leading edge of a signal a0/b0.

15.3.17 TDxRG2 (Timer register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TDRG2							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TDRG2							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-0	TDRG2[15:0]	R/W	The timer register TDxRG2 is double-buffered with the compare register TDxCP2. The setting of the compare register is performed on the timer register.

15.3.18 TDxCP2 (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	-	-	-	-	-	-	-	-
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							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-0	CPRG2[15:0]	R	Timing setting register: In the 16-bit interval timer: Sets a timing to generate an interrupt signal INTTDxCMP2. In the 16-bit PPG: Sets a timing of a trailing edge of a signal a0/b0.

15.3.19 TDxRG3 (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	The timer register TDxRG3 is double-buffered with the compare register TDxCP3. The setting of the compare register is performed on the timer register.

15.3.20 TDxCP3 (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
31-16	−	R	Read as 0.
15-0	CPRG3[15:0]	R	Timing setting register: In the 16-bit interval timer: Sets a timing to generate an interrupt signal INTTDxCMP3. In the 16-bit PPG: Sets a timing of a leading edge of a signal a1/b1.

15.3.21 TDxRG4 (Timer register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TDRG4							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TDRG4							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-0	TDRG4[15:0]	R/W	The timer register TDxRG4 is double-buffered with the compare register TDxCP4. The setting of the compare register is performed on the timer register.

15.3.22 TDxCP4 (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	-	-	-	-	-	-	-	-
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							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-0	CPRG4[15:0]	R	Timing setting register: In the 16-bit interval timer: Sets a timing to generate an interrupt signal INTTDxCMP4. In the 16-bit PPG: Sets a timing of a trailing edge of a signal a1/b1.

15.3.23 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	-	-	-	-	-	-	-	-
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
31-16	−	R	Read as 0.
15-0	TDRG5[15:0]	R/W	The timer register TD0RG5 is double-buffered with the compare register TD0CP5. The setting of the compare register is performed on the timer register.

15.3.24 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	-	-	-	-	-	-	-	-
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
31-16	−	R	Read as 0.
15-0	CPRG5[15:0]	R	Register to set timing and width of phase shift: In the 16-bit interval timer: Sets a match timing of the CP05. In the 16-bit PPG: Sets the width of phase shift in the interlock PPG mode. (Invalid in the PPG mode)

15.4 Operation mode

Modes of the TMRD can be set into the register TD0CONF<TMRDMOD[2:0]>. The correlation between the setting value of the register and operation modes of the timer units will be found in the chapter of the register. The operations of the modes are described below.

Hereinafter, numbers in the names of circuit elements in timer units and registers are collectively described as "x" and "m". e.g. CPxm, UCx, TDxRGm and TDxCPm.

In this document, unless otherwise noted, x = 0,1. When x = 0, m = 0 to 5, when x =1, m = 0 to 4.

15.4.1 16-bit interval timer

15.4.1.1 Timer mode

In timer mode, the timer TMRD0 and the timer TMRD1 operate independently. Since the operation of the timer TMRD0 is the same as the timer TMRD1, this section describes the TMRD0 only.

If the value of the register TD0MOD<TDCLE> is "0", the counter UC0 operates as a free-run counter, which starts counting up from 0x0000, and returns to 0x0000 when it counts up to 0xFFFF. It generates an overflow interrupt (UC0OVF) if the count value becomes 0xFFFF.

If the value of the register TD0MOD<TDCLE> is "1", the counter UC0 starts counting up from 0x0000, and returns to 0x0000 if the count value matches the value set in the compare register (TD0CP0) of the comparator. The timer cycle is configured by the comparator CP00.

In either case, each comparator (CPxm) generates a timer interrupt if the count value of the UC0 matches the value set in the each compare register (TD0CPm).

15.4.1.2 Interlock timer (synchronization start) mode

This mode is a timer mode that can start the timer TMRD0 and TMRD1 at the same time. The two counters UC0 and UC1 are started simultaneously by setting the register TD0RUN<TDRUN> to "1". In this mode, the values set in the register TD1RUN<TDRUN> are ignored for starting and stopping the timers. However, the timer cycle of the timer TMRD0 and TMRD1 can be respectively set.

15.4.1.3 Timer interrupt

Table 15-1 shows the relation of timer interrupt signals and factors.

Interrupt factors of INTTD0CMP0 is configured by a value of the register TD0CR<TDISO[1:0]>, and INTTD1CMP0 is configured by a value of the register TD1CR<TDISO[1:0]>.

Table 15-1 Timer mode interrupts and factors.

Interrupts		Factors
INTTD0CMP0	TD0CR<TDISO[1:0]> = "00"	No factors
	TD0CR<TDISO[1:0]> = "01"	A match detected by the comparator 00 (CP00)
	TD0CR<TDISO[1:0]> = "10"	A match detected by the comparator 05(CP05)
	TD0CR<TDISO[1:0]> = "11"	Overflow of the counter 0(UC0).
INTTD0CMP1		A match detected by the comparator 01(CP01)
INTTD0CMP2		A match detected by the comparator 02(CP02)
INTTD0CMP3		A match detected by the comparator 03(CP03)
INTTD0CMP4		A match detected by the comparator 04(CP04)
INTTD1CMP0	TD1CR<TDISO[1:0]> = "00"	No factors
	TD1CR<TDISO[1:0]> = "01"	A match detected by the comparator 10(CP10)
	TD1CR<TDISO[1:0]> = "10"	No factors
	TD1CR<TDISO[1:0]> = "11"	Overflow of the counter 1 (UC1)
INTTD1CMP1		A match detected by the comparator 11(CP11)
INTTD1CMP2		A match detected by the comparator 12(CP12)
INTTD1CMP3		A match detected by the comparator 13(CP13)
INTTD1CMP4		A match detected by the comparator 14(CP14)

Note: In 16-bit programmable pulse generation output mode, interrupt factors do not occur in the below conditions.

- Overflow interrupt factors in the counter UC0 and UC1.
- The factor that the detection of a match between the counter and the comparator CP10 in interlock PPG mode.

15.4.1.4 Registger configuration process in timer mode

(1) Process to start up

(1-1) Timer mode

Follow the following steps to configure the register settings before starting the timer mode.

a. Select a clock to be provided

1. Select a clock to be provided to the TMRD and set it to the register CGPWMGEAR<PWMGEAR[1:0]>.
2. Set the register CGPWMGEAR<TMRDCLKEN> to "1" to enable providing a clock to the TMRD. Be sure to set the <TMRDCLKEN> after setting the value of <PWMGEAR[1:0]>.

b. Select operation modes of timer units

1. Set the timer unit to be used in the timer mode to the register TD0CONF<TMRDMOD[2:0]>. Simultaneuosly, set the operation status of the timer unit used in the IDLE mode to <TDI2TD1> and <TDI2TD0> of the same register TD0CONF.

c. Select clocks to be used in the timer units (TMRD0/TMRD1).

1. Set "1" to the bit <TDEN1> and <TDEN0> of the register TD0EN which disables and enables a clock signal to be provided to the timer unists. (It’s not necessary to configure timer units which don’t need to be enabled.)
2. Select a clock to be used in timer units and set it to the register TDxMOD<TDCLK[3:0]>.

   Simultaneously, set the operation status of the counters used in in the timer mode to the register TDxMOD<TDCLE>.

d. Initial settings of timer registers and compare rgisters

1. Set "0" to <TDRDE> of the register TDxCR to write to the compare register in the direct write mode. (It’s a mode that the same data is written to the corresponding compare register when data is written to the timer register.) Set an interrupt factor of the INTTDxCMPm into the register TDxCR<TDISO[1:0]>.
2. Set a required value to timer register (TDxRGm).

   In the mode which condition is TDxCR<TDRDE> = "0", an operation to set "1" to the register TDxBCR<TDSFT> and an update timing signal are not required.

e. Start the timer mode

1. Set "1" to <TDRDE> of the register TDnCR to enable writing data through the timer register to the compare register. (The value of the timer register is written into the corresponding timer register at the predefined update time in this mode.)
2. Start the timer mode by setting "1" to <TDRUN> of the register TDxRUN.

(1-2) Interlock timer mode

Follow the following steps to configure the register settings before starting the interlock timer mode.

a. Select a clock to be provided

1. Select a clock to be provided to the TMRD and set it to the register CGPWMGEAR<PWMGEAR[1:0]>.
2. Set the register CGPWMGEAR<TMRDCLKEN> to "1" to enable providing a clock to the TMRD. Be sure to set the <TMRDCLKEN> after setting the value of <PWMGEAR[1:0]>.

b. Select operation modes of timer units

1. Set the timer unit to be used in the timer mode to the register TD0CONF<TMRDMOD[2:0]>. Simultaneuosly, set the operation status of the timer unit used in the IDLE mode to <TDI2TD1> and <TDI2TD0> of the same register TD0CONF.

c. Select clocks to be used in the timer units (TMRD0/TMRD1).

1. Set "1" to both bits of <TDEN1> and <TDEN0> in the register TD0EN. The register TD0EN enables and disables providing clocks to the two timer units. (Be sure to set both <TDEN1> and <TDEN0>.)
2. Select a clock to be used in timer units and set it to the register TDxMOD<TDCLK[3:0]>.

   Simultaneously, set the operation status of the counters used in in the timer mode to the register TDxMOD<TDCLE>.

d. Initial settings of timer registers and compare rgisters

1. Set "0" to <TDRDE> of the register TDxCR to write to the compare register in the direct write mode. (It’s a mode that the same data is written to the corresponding compare register when data is written to the timer register.) Set an interrupt factor of the INTTDxCMPm into the register TDxCR<TDISO[1:0]>.
2. Set a required value to timer register (TDxRGm).

   In the mode which condition is TDxCR<TDRDE> = 0, an operation to set "1" to the register TDxBCR<TDSFT> and an update timing signal are not required.

e. Start the timer mode

1. Set "1" to <TDRDE> of the register TDnCR to enable writing data through the timer register to the compare register. (The value of the timer register is written into the corresponding timer register at the predefined update time in this mode.)
2. Set "1" to <TDRUN> of the register TD0RUN, start two timer units at the same time. Since the mode is a interlock timer mode, settings to the register TD1RUN<TDRUN> is not required.

(2) Configuration after starting the timer modes

(2-1) Timer mode and interlock timer mode

To set the register after starting the timer mode, make settings in the following procedures.

a. Update values of timer registers and compare registers

1. Set an arbitrary value to timer register (TDxRGm) corresponding to the compare register to be updated.
2. After the above setting is done, set "1" to register TDxBCR<TDSFT> which is corresponding to the timer unit to be updated.

With the above settings, a value in a timer register is set in the corresponding compare register at the predefined update timing. (See Figure 15-4)

Note: Registers which can be configured when the timer is running are the following three registers: (TDxRGm), (TDxBCR) and (TDxRUN). Other registers must be configured when the timer is being stopped.

15.4.1.5 Setting range of compare registers

The setting range of the compare registers in 16-bit interval timer mode is shown in the Table 15-2.

Table 15-2 Setting range of the compare registers in 16-bit interval timer mode

Timer Unit	Compare register	16-bit interval timer	
		<TDCLE> = "0"	<TDCLE> = "1"
TMRD0	TD0CP0	0x0000 ≤ CPRG0[15:0] ≤ 0xFFFF	0x0001 ≤ CPRG0[15:0] ≤ 0xFFFF
	TD0CP1	0x0000 ≤ CPRG1[15:0] ≤ 0xFFFF	0x0000 ≤ CPRG1[15:0] ≤ CPRG0[15:0]
	TD0CP2	0x0000 ≤ CPRG2[15:0] ≤ 0xFFFF	0x0000 ≤ CPRG2[15:0] ≤ CPRG0[15:0]
	TD0CP3	0x0000 ≤ CPRG3[15:0] ≤ 0xFFFF	0x0000 ≤ CPRG3[15:0] ≤ CPRG0[15:0]
	TD0CP4	0x0000 ≤ CPRG4[15:0] ≤ 0xFFFF	0x0000 ≤ CPRG4[15:0] ≤ CPRG0[15:0]
	TD0CP5	0x0000 ≤ CPRG5[15:0] ≤ 0xFFFF	0x0000 ≤ CPRG5[15:0] ≤ CPRG0[15:0]
TMRD1	TD1CP0	0x0000 ≤ CPRG0[15:0] ≤ 0xFFFF	0x0001 ≤ CPRG0[15:0] ≤ 0xFFFF
	TD1CP1	0x0000 ≤ CPRG1[15:0] ≤ 0xFFFF	0x0000 ≤ CPRG1[15:0] ≤ CPRG0[15:0]
	TD1CP2	0x0000 ≤ CPRG2[15:0] ≤ 0xFFFF	0x0000 ≤ CPRG2[15:0] ≤ CPRG0[15:0]
	TD1CP3	0x0000 ≤ CPRG3[15:0] ≤ 0xFFFF	0x0000 ≤ CPRG3[15:0] ≤ CPRG0[15:0]
	TD1CP4	0x0000 ≤ CPRG4[15:0] ≤ 0xFFFF	0x0000 ≤ CPRG4[15:0] ≤ CPRG0[15:0]

If the frequency of TMRDCLK is f_{CLK}, a timer cycle Tx can be obtained using the following equation :

1. If <TDCLE> = 0, $Tx = (1/f_{CLK}) \times 2^{16}$
2. If <TDCLE> = 1, $Tx = (1/f_{CLK}) \times (TDxCP0<CPRG0[15:0]> + 1)$

15.4.2 16-bit programmable pulse generation output

15.4.2.1 PPG mode

In PPG (Programmable Pulse Generator) mode, TMRD0 and TMRD1 operate independently, and these timers can output a pulse with a programmable frequency and a duty cycle.

Pulse output of TMRD0: a0, a1, Pulse output of TMRD1: b0, b1

(1) Cycle of a pulse generation (PPG) output

The cycle of pulse ouput (a0/a1) of TMRD0 is configured with the value of the compare register TD0CP0<CPRG0[15:0]> of CP00. As shown in the Figure 15-5, UC0 operates, like a timer mode, as a counter which is zero cleared when a match signal is output by CP00.

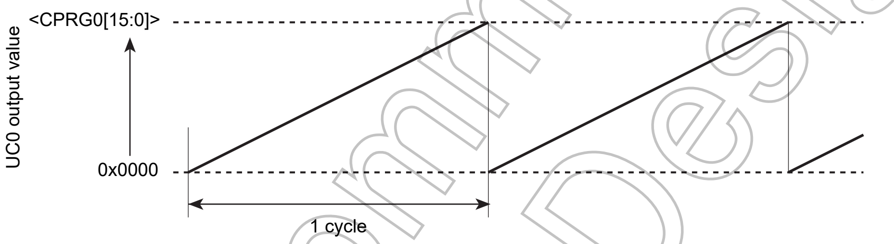


Figure 15-5 The cycle of a square-wave output in PPG mode

The cycle of a pulse output (b0/b1) of TMRD1 is also configured by the value of the compare register (TD1CP0<CPRG0[15:0]>) of CP10.

(2) Duty cycle of a pulse generation (PPG) output

As for TMRD0, a leading edge of the pulse output a0 is generated when a match signal of CP01 is detected and a trailing edge is generated when a match signal of CP02 is detected, as shown in the Figure 15-6. These leading edge and trailing edge are generated the pulse outputs.

Equally, a leading edge of the pulse output a1 is generated when a match signal of CP03 is detected and a trailing edge is generated when a match signal of CP04 is detected. These leading edge and trailing edge are generated the pulse outputs.

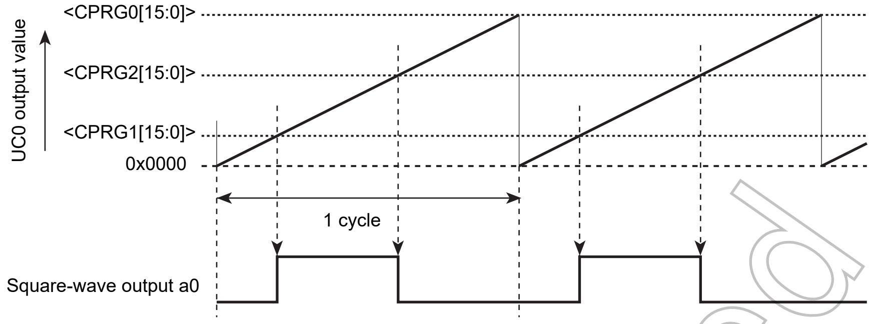


Figure 15-6 Square-wave output in PPG mode (when square-wave output is a a0)

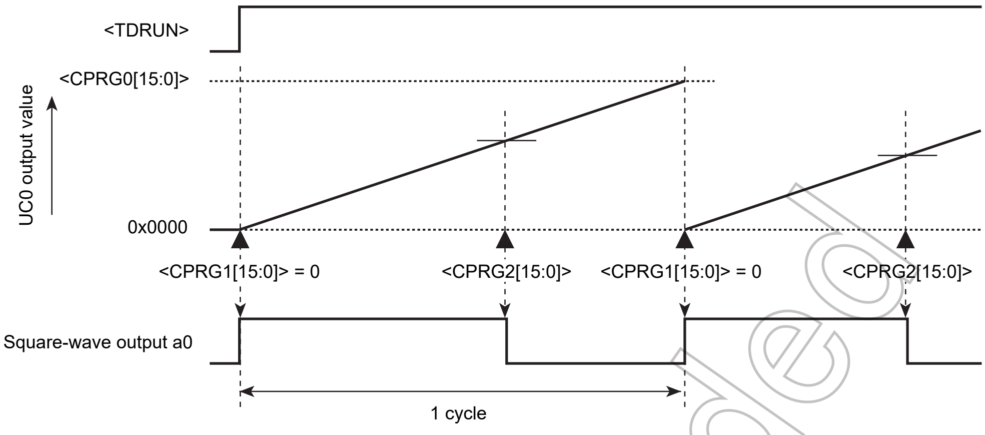
Figure 15-6 shows that a leading edge is a rising waveform and a trailing edge is a falling waveform. Switching the waveforms is possible by setting the register TD0MOD<TDIV[1:0]>; a leading edge is to be a falling waveform and a trailing edge is to be a rising waveform.

As for TMRD1, a leading edge of a pulse output b0 is generated by detecting a match signal of CP11, and a trailing edge is gnenerated by detecting a match signal of CP12. A leading edge of a pulse output b1 is generated by detecting a match signal of CP13, and a trailing edge is generated by detecting a match signal of CP14.

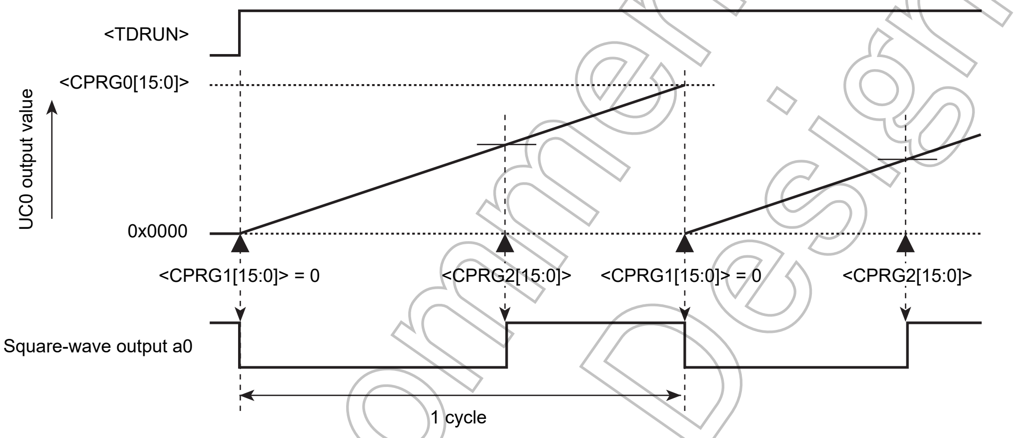
Also in PPG mode, the control signal PHSCHGSW of SW0/SW1/SW2/SW3 is fixed to "0". Signals from TMRD0 are output only through SW0 and SW1. Signals from TMRD1 are output only through SW2 and SW3. The description is shown in "Figure 15-2 TMRD Unit Block Diagram".

(3) Start timing of pulse generation (PPG) output

Pulse generation (PPG) output starts by writing "1" to the register TDxRUN<TDRUN>. Figure 15-7 shows that the arbitrary pulse can be generated from the first cycle even a leading edge is set to rise/fall as the operation starts. Although the figure below shows the case of a0, a1 and b0/b1 are the same. The figure below shows the timing in the condition <CPRG1[15:0]> = "0".



(1) If a leading edge is specified as a rising waveform



(2) If a leading edge is specified as a falling waveform

Figure 15-7 Timing Chart at the start of operation (when square-wave output is a0)

(4) Configuration of the register in PPG mode

(4-1) Process to start up PPG mode

Register settings to start up PPG mode are made in the following steps.

a. Select a clock to be provided

1. Select a clock to be provided to the TMRD and set it to the register CGPWMGEAR<PWMGEAR[1:0]>.
2. Set the register CGPWMGEAR<TMRDCLKEN> to "1" to enable providing a clock to the TMRD. Be sure to set the <TMRDCLKEN> after setting the value of <PWMGEAR[1:0]>.

b. Select operation modes of timer units

1. Set the timer unit to be used in the PPG mode to the register TD0CONF<TMRDMOD[2:0]>. Simultaneuosly, set the operation status of the timer unit used in the IDLE mode to <TDI2TD1> and <TDI2TD0> of the same register TD0CONF.

c. Select clocks to be used in the timer units (TMRD0/TMRD1).

1. Set "1" to the bit <TDEN1> and <TDEN0> of the register TD0EN which disables and enables a clock signal to be provided to the timer unists. (It’s not necessary to configure timer units which don’t need to be enabled.)
2. Select a clock to be used in timer units and set it to the register TDxMOD<TDCLK[3:0]>.

   At the same time, configure <TDIV[1:0]> in the same register which specify the leading edge and trailing edge of the pulse output.

d. Initial settings of timer registers and compare rgisters

1. Set "0" to <TDRDE> of the register TDxCR to write to the compare register in the direct write mode. (It’s a mode that the same data is written to the corresponding compare register when data is written to the timer register.)
2. Set a required value to timer register (TDxRGm).

   In the mode that <TDRDE> of the register TDxCR is set to "0", neither the operation to set "1" to <TDSFT> of the register TDxBCR nor an update timing signal.

e. Start the PPG mode

1. Set "1" to <TDRDE> of the register TDnCR to enable writing data through the timer register to the compare register. (The value of the timer register is written into the corresponding timer register at the predefined update time in this mode.)
2. Start the PPG mode by setting "1" to <TDRUN> of the register TDxRUN.

(4-2) Process after starting the PPG mode

To set the register after starting the PPG mode, make settings in the following procedures.

a. Update values of timer registers and compare registers

1. Set an arbitrary value to timer register (TDxRGm) corresponding to the compare register to be updated.
2. After the above setting is done, set "1" to register TDxBCR<TDSFT> which is corresponding to the timer unit to be updated.

With the above settings, a value in a timer register is set in the corresponding compare register at the predefined update timing. (See Figure 15-4)

b. Stops the pulse generation output.

1. To stop a pulse output from timer units, set "0" to <TDRUN> of the register TDxRUN.

Note: Registers which can be configured when the PPG is running are the following three registers: (TDxRGm), (TDxBCR) and (TDxRUN). Other registers must be configured when the timer is being stopped.

15.4.2.2 Interlock PPG mode

In this mode, the phase relation between a pulse (a0/a1) output by TMRD0 and a pulse (b0/b1) output by TMRD1 can be dynamically specified in the following range: -180 degree < θ < +180 degree.

However, the phase relation between a0 and b0 and that between a1 and b1 are the same.

(1) Operation Outline

In this mode, the phase relation between the phase-A output which is output by SW0/SW1 and the phase-B output which is output by SW2/SW3 can be dynamically specified in the following range: -180 degree < θ < +180 degree. However, the phase relation between the phase-A0 output and the phase-B0 output and that between the phase-A1 output and the phase-B1 output are the same values.

A pair of the phase-A0 and the phase-B0 and the other pair of the phase-A1 and the phase-B1 output operate the same unless otherwise stated. The following explanation is presented taking the phase-A0 and the phase-B0 as the operation outline of the phase relation. Refer to the Figure 15-8.

Figure 15-8 is a simplified block diagram of the timer unit in Figure 15-2.

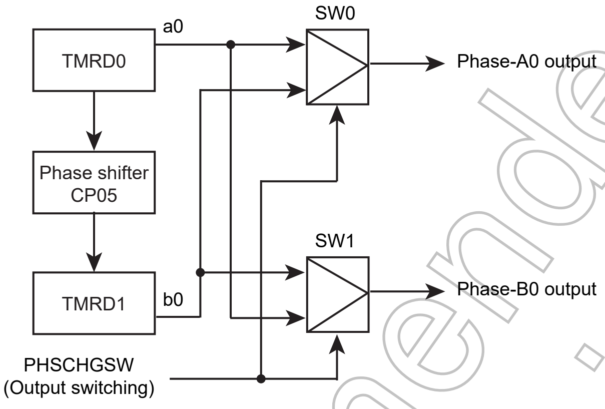


Figure 15-8 Interlock PPG Configuration diagram (In case of phase A0 and phase B0)

In the Figure 15-8, CP05 operates as a phase shifter (delay generator) which can delay the pulse output b0 in the following range (relative to the pulse output a0):

Phase 0 degree ≤ θ < 180 degree

To delay the phase of the phase-B0 output to the phase-A0 output, switch SW0/SW2 using an output change signal (PHSCHGSW) to satisfy the following condition:

Phase-A0 output = a0, phase-B0 output = b0

To set the phase-B0 output as the same phase or faster than the phase-A0 output, switch SW0/SW2 using an output change signal (PHSCHGSW) to satisfy the following condition:

Phase-A0 output = b0, phase-B0 output = a0

Unlike the PPG mode, a phase shifter can be achieved by setting the counter UC1 to "0" when a match signal of CP05 is detected, not that of CP10.

(2) Operation Details

The mode is changed to PPG mode by setting "111" to TD0CONF<TMRDMOD[2:0]>, and the counter UC1 operates as a counter which is zero cleared when a match of CP05 of TMRD0 is detected, not a match of CP10. After this, TMRDCLK0 and TMRDCLK1 cannot be configured respectively, the same frequencies are set to TMRDCLK1 and TMRDCLK0. Therefore, the pulse b0 is output later than the pulse a0 for the corresponding period to the setting value of TD0CP5<CPRG5[15:0]> in the compare register CP05. The phase relation is shown in the Figure 15-9. As seen in the Figure 15-9, the phase of the pulse a0 is the same phase or faster than the phase of the pulse b0.

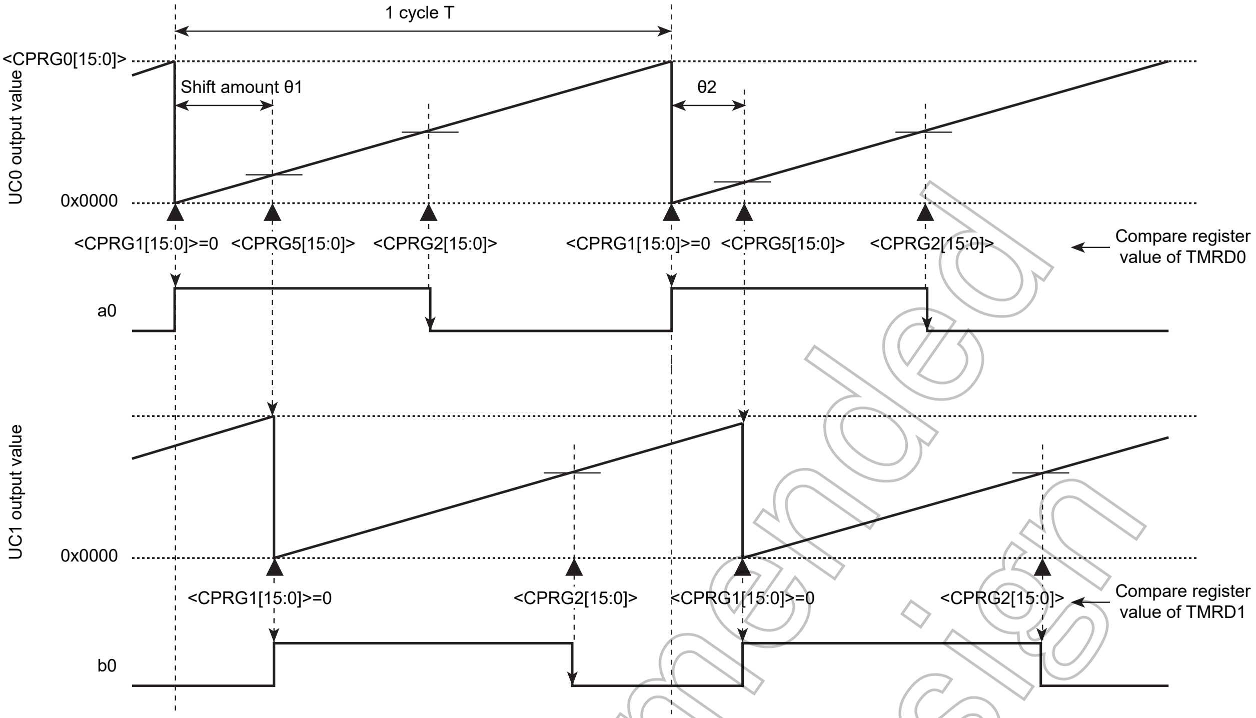


Figure 15-9 Phase relation between square-wave a0 and b0

The cycles of the pulse a0 and b0 are depending on the setting value of the compare register TD0CP0<CPRG0[15:0]> of CP00. Thus, the phase shift (delay) (θ) can be calculated by using the formula:

θ = 360 degree × (<CPRG5[15:0]> ÷ (<CPRG0[15:0]> + 1))

Also the condition of this operation mode is set to 0 degree ≤ θ < 180, the setting range of <CPRG5 [15:0]> is 0x0000 ≤ CPRG5[15:0] < (CPRG0[15:0] ÷ 2).

As with PPG mode, duties of the pulse a0 and b0 are specified by the comparator CP01, CP02 and CP11, CP12.

(2-1) Setting the phase of the phase-A0 faster or the same as the phase of the phase B0

To set the phase of the phase-A0 faster or the same as the phase of the phase B0, PHSCHGSW of SW0/SW2 needs to set in the following condition:

Phase-A0 output = pulse a0, phase-B0 output = pulse b0

Settings of PHSCHGSW will hereinafter be described.

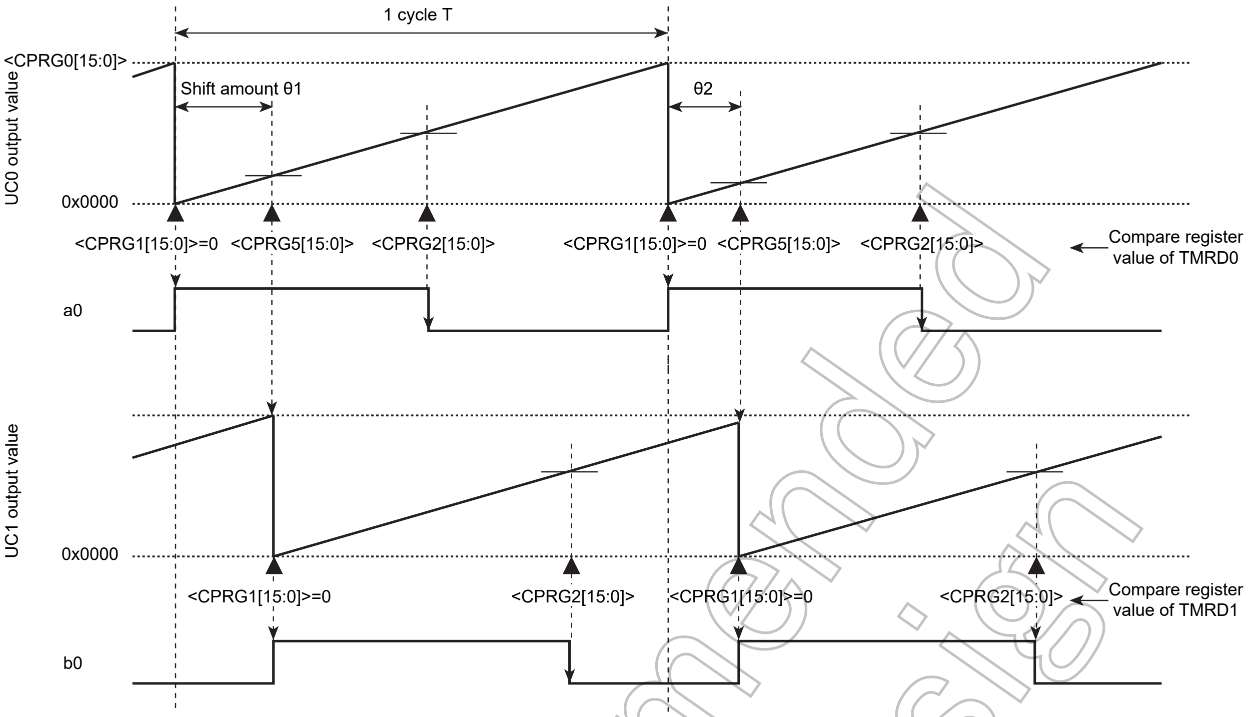


Figure 15-10 In case that the phase of phase-A0 is set faster than the phase-B0 (phase-A0 output = square-wave a0, phase-B0 output = square-wave b0)

(2-2) Delaying the phase of the phase-A0 to that of the phase-B0

To delay the phase of the phase-A0 output to that of the phase-B0 output, or to set he phase of the phase-A0 output the same as that of the phase-B0 output, PHSCHGSW of SW0/SW2 needs to set in the following condition:

Phase-A0 output = pulse b0, phase-B0 output = pulse a0

Figure 15-11 shows the waveform chart.

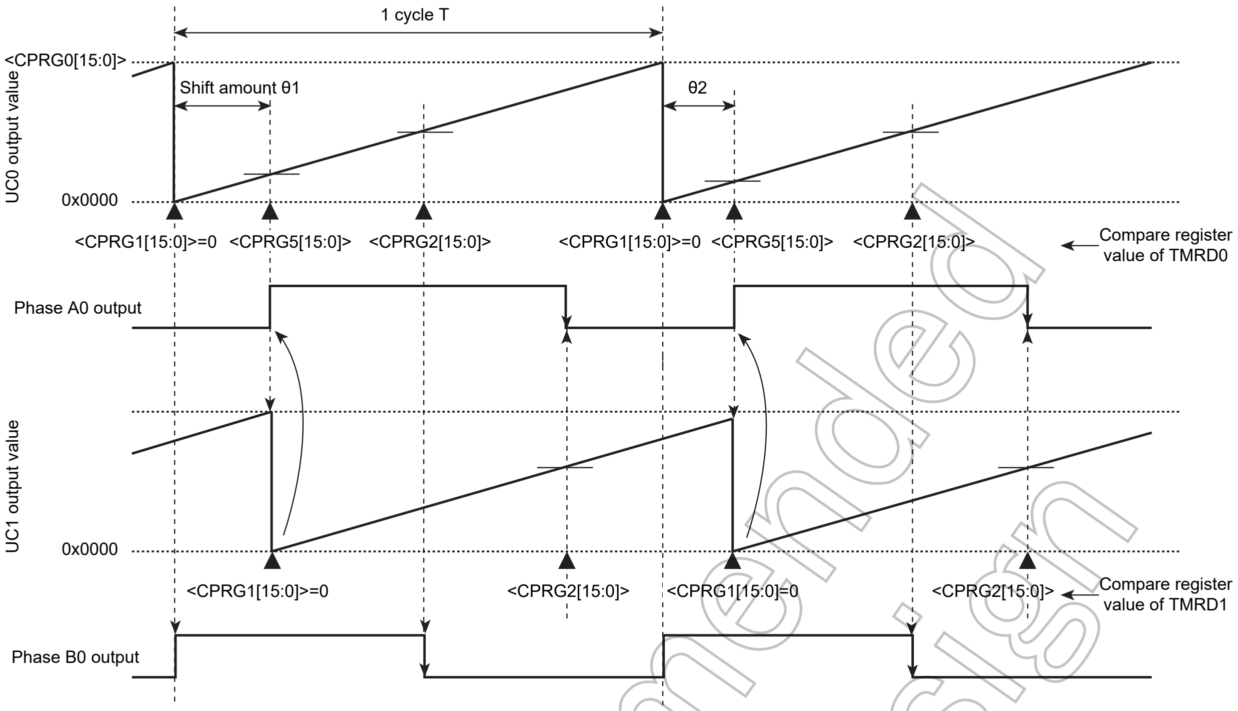


Figure 15-11 In case that the phase of phase-A0 delay to the phase-B0 (phase-A0 output = square-wave b0, phase-B0 output = square-wave a0)

(2-3) Switching output SW (SW0/SW1/SW2/SW3)

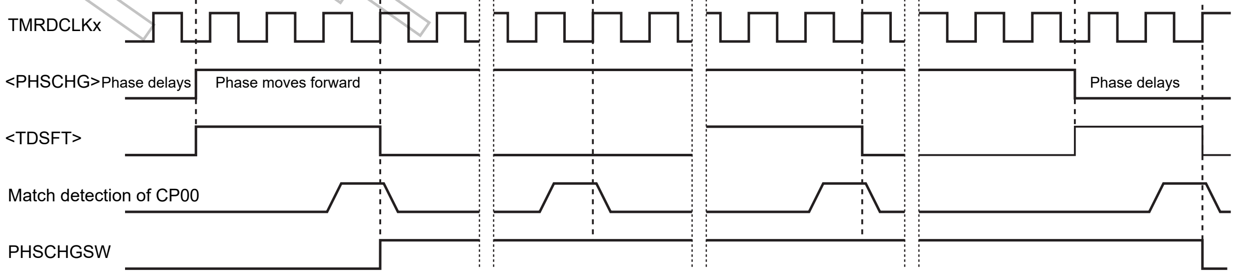
The control of SW0/SW1/SW2/SW3 is done by the settings of the register TD0BCR<PHSCHG>. However, SW0/SW1/SW2/SW3 can not be respectively controlled.

The phase relation of the phase-B to the phase-A can be set by the register TD0BCR<PHSCHG>.

TD0BCR<PHSCHG> = "0" : Delay or same phase
(Phase-A output = pulse a0/a1, phase-B output = pulse b0/b1)
TD0BCR<PHSCHG> = "1" : Fast or same phase
(Phase-A output = pulse b0/b1, phase-B output = pulse a0/a1)

However, the bit operation of these registers is valid only in interlock PPG mode. The bit operation is ignored in PPG mode, and the condition in PPG mode is always as follows:

Phase-A output = pulse a0/a1, phase-B output = pulse b0/b1.



* Forward/delay are the phase-B's to phase-A

Figure 15-12 Switching time of output SW (SW0/SW1/SW2/SW3)

Figure 15-12 shows the switching time waveform of SW(SW0 through SW3). The signal PHSCHGSW for switching SW changes depending on the value of the register TD0BCR<PHSCHG> when a match signal of CP00 is detected or TD0RUN<TDRUN>="1" in the condition the update enable flag TDBCRx<TDSFT> is "1". The signal PHSCHGSW switches SW to fulfill the following conditions:

PHSCHGSW = "0" : Phase-A output = pulse a0/a1, phase-B output = pulse b0/b1

PHSCHGSW = "1" : Phase-A output = pulse b0/b1, phase-B = pulse a0/a1

Figure 15-13 shows an example of waveform when PHSCHGSW changes from "0" to "1".

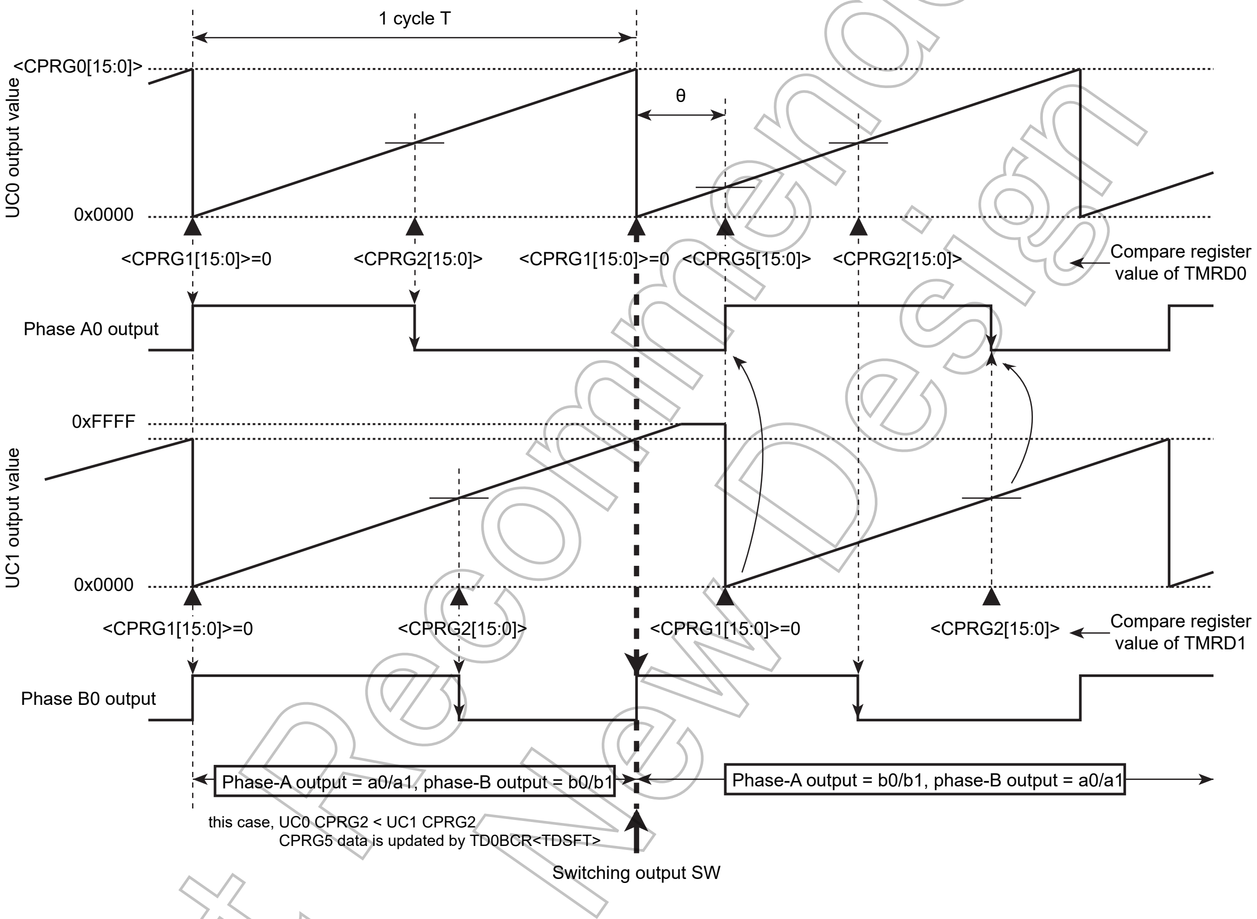


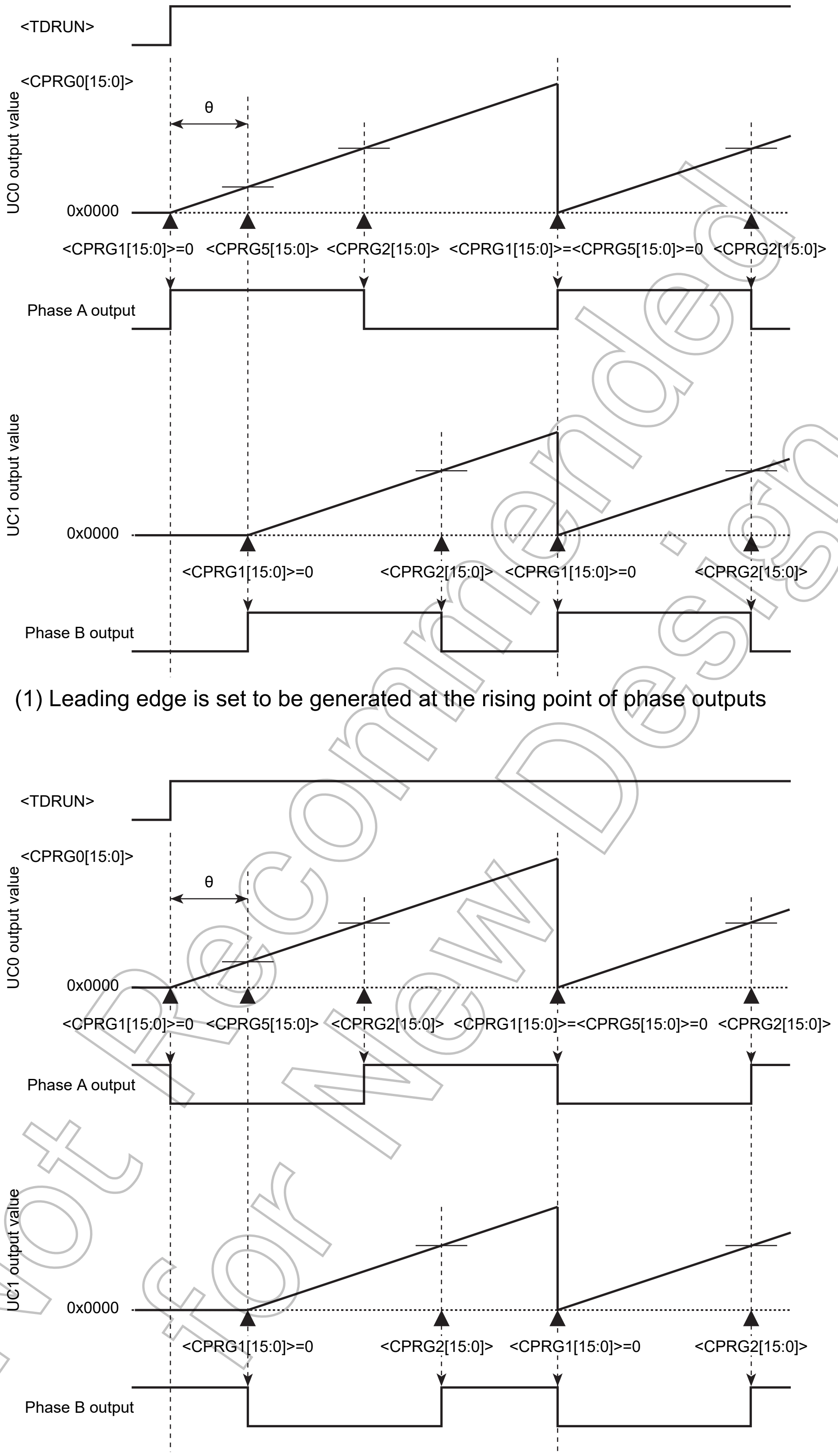
Figure 15-13 Waveform in case that PHSCHGSW changes from 0 to 1

(2-4) Counter 1 overflow processing

Since the range of phase shift is 0 degree ≤ θ < 180, the range from CP05 match detection to the next match detection could be 0.5T ≤ Tθ < 1.5T, if the cycle of the phase A and the phase B output is T. Therefore, depending to the value of the cycle T, the counter 1 (UC1) may overflow. In this mode, if the value of the UC1 counter exceeds 0xFFFF, overflow processing stops the counting up operation and maintains the value of 0xFFFF until the next CP05 match is detected.

(2-5) Start timing of interlock PPG mode

The output of the interlock PPG starts by writing "1" to <TDRUN>of the register TD0RUN. As shown in the Figure 15-14, arbitrary waveform output can be generated from the first cycle even if the leading edge is set to rises/falls concomitantly with the start. The figure shows the case in the following condition: TDxCP1<CPRG1[15:0]> = "0".



(1) Leading edge is set to be generated at the rising point of phase outputs

(2) Leading edge is set to be generated at the falling point of phase outputs

Figure 15-14 Timing chart at start up

(3) Configuration of the register in interlock PPG mode

(3-1) Register setting priority in interlock PPG mode

Since TMRD0 operates as a maser and TMRD1 operates a slave in this mode, the setting of the TMRD0 has priority over the setting of the TMRD1 for some bits. Table 15-3 shows those bits.

Table 15-3 Bits that TMRD0 setting has priority

TMRD0 register	TMRD1 register
TD0MOD<TDCLK[3:0]>	TD1MOD<TDCLK[3:0]>
TD0CR<TDRDE>	TD1CR<TDRDE>
TD0RUN<TDRUN>	TD1RUN<TDRUN>

Therefore, TMRD1 registers of the above registers do not need to be configured in this mode. If these registers are configured, the values are ignored but please note that these values remain in the registers. For example, <TDRUN> of the register TD1RUN is set to "1", TMRD1 restarts after this mode stops. Set "0" to <TDRUN> of the register TD1RUN.

(3-2) Process to start up the interlock PPG mode

Register settings to start up the interlock PPG mode are made in the following steps.

a. Select a clock to be provided

1. Select a clock to be provided to the TMRD and set it to the register CGPWMGEAR<PWMGEAR[1:0]>.
2. Set the register CGPWMGEAR<TMRDCLKEN> to "1" to enable providing a clock to the TMRD. Be sure to set the <TMRDCLKEN> after setting the value of <PWMGEAR[1:0]>.

b. Select operation modes of timer units

1. Since timer units operate in the interlock PPG mode, set "111" to <TMRDMOD[2:0]> of the register TD0CONF. Simultaneuosly, set the operation status of the timer unit used in the IDLE mode to <TDI2TD1> and <TDI2TD0> of the same register TD0CONF.

c. Select clocks to be used in the timer units (TMRD0/TMRD1).

1. Set "1" to the bit <TDEN1> and <TDEN0> of the register TD0EN which disables and enables a clock signal to be provided to the timer unists. (Both <TDEN1> and <TDEN0> must be set to "1".)
2. Select a clock to be used in timer units and set it to the register TD0MOD<TDCLK[3:0]>.

   At the same time, configure <TDIV[1:0]> in the same register which specify the polar of the leading edge and trailing edge of a0 and a1.
3. Configure TD1MOD<TDIV[1:0]> which specify the polar of the leading edge and trailing edge of b0 and b1.

d. Initial settings of timer registers and compare rgisters

1. Set "0" to <TDRDE> of the register TDxCR to write to the compare register in the direct write mode. (It's a mode that the same data is written to the corresponding compare register when data is written to the timer register.)
2. Set a required value to timer register (TDxRGm).

   In the mode which condition is TDxCR<TDRDE> = 0, an operation to set "1" to the register TDxBCR<TDSFT> and an update timing signal are not required.

e. Initialization of phase relation (delay/fast) between the phase A and the phase B

1. Sets the phase relation between the phase A and the phase B, delay or fast, to the register TD0BCR<PHSCHG>. It's not necessary to set "1" to TD0BCR<TDSFT>.

f. Start the interlock PPG mode

1. Set "1" to <TDRDE> of the register TDnCR to enable writing data through the timer register to the compare register. (The value of the timer register is written into the corresponding timer register at the predefined update time in this mode.)
2. Start the PPG mode by setting "1" to <TDRUN> of the register TD0RUN.

(3-3) Process after starting the interlock PPG mode

To set the register after starting the interlock PPG mode, make settings in the following procedures.

a. Update values of timer registers and compare registers

1. Set an arbitrary value to timer register (TDxRGm) corresponding to the compare register to be updated. However, the value in TD1RG0 does not affect this mode.
2. If you change the phase relation between the phase A and the phase B (delay/fast) after above setting, change the setting value of TD0BCR<PHSCHG> and set "1" to TD0BCR<TDSFT> simultaneuosly. Do not set "1" to TD1BCR<TDSFT>.

With the above settings, a value in a timer register is set in the corresponding compare register at the predefined update timing. (See Figure 15-4)

b. Stops the pulse generation output.

1. To stop a pulse output from timer units, set "0" to <TDRUN> of the register TDxRUN.

Note: Registers which can be configured when the PPG is running are the following three registers: (TDxRGm), (TDxBCR) and (TDxRUN). Other registers must be configured when the timer is being stopped.

(3-4) Notes on switching output SW in the interlock PPG mode

In the interlock PPGmode, a trailing edge of pulse b0/b1 can be configured in a moment exceeding the cycle of COUNTER0(UC0). If you change output SW, the phase from delay to fast or fast to delay, special attention is required.

Hereinafter, it’s described that waveform of the phase A and B when the output SW is switched in both cases; trailing edge of a pulse b0/b1 is in the UC0 cycle and out of the UC0 cycle.

The term "UC0 cycle" means the period that the counter value changes from "0" to "0".

(3-4.1) When the trailing edge of the pulse b0/b1 is in the UC0 cycle

As shown in Figure 15-15, please note the following if you try to change the status of the phase A from fast state to delay state when the phase relation between A0 and B0 is in the condition that the trailing edge is in the UC0 cycle. As shown in Figure 15-16, switch the SW at predefined match detection timing of CP00, and change the conditions that the phase A0 output = pulse a0 and the phase B0 output = pulse b0 to the condition that the phase A0 output = pulse b0 and the phase B0 output = pulse a0. The output SW is switched without generating outlier waveform since the signal level of the pulse a0 and b0 are the same when the output SW is switched.

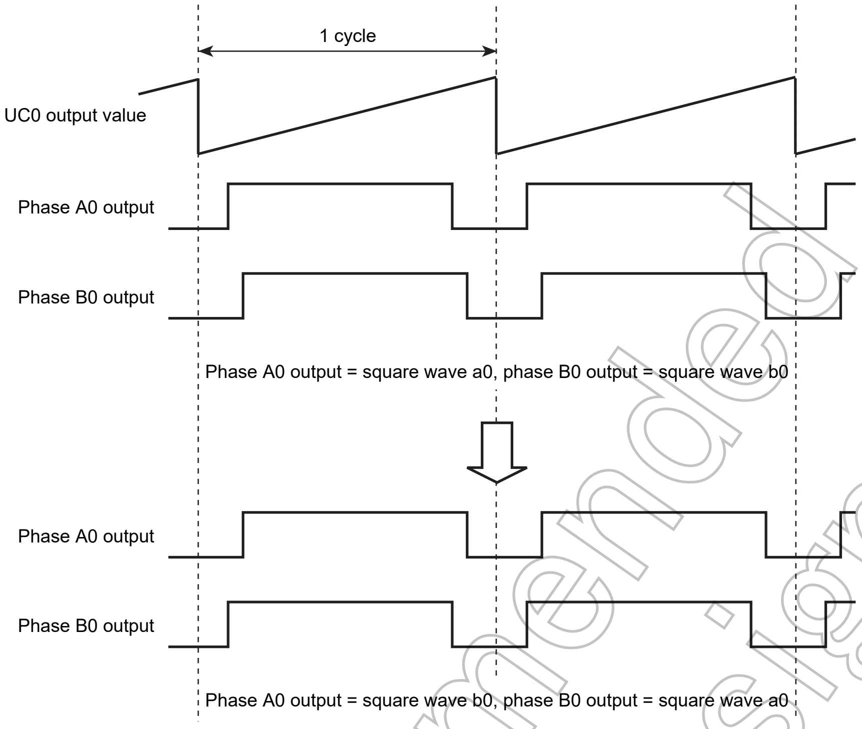

1 cycle
UC0 output value
Phase A0 output
Phase B0 output
Phase A0 output = square wave a0, phase B0 output = square wave b0
Phase A0 output
Phase B0 output
Phase A0 output = square wave b0, phase B0 output = square wave a0

Figure 15-15 Waveform before and after switching the phase status (output SW) between delay and fast (Trailing edge of the squarewave b0 is in the range of the UC0 cycle)

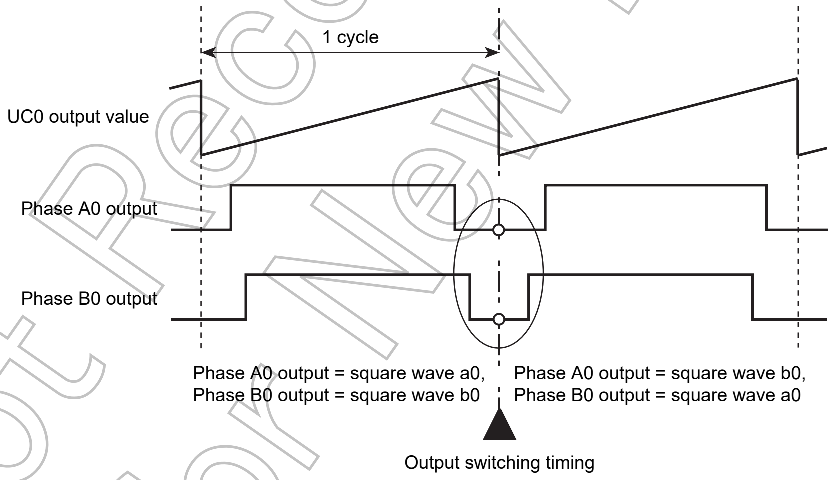

1 cycle
UC0 output value
Phase A0 output
Phase B0 output
Phase A0 output = square wave a0,
Phase B0 output = square wave b0
Phase A0 output = square wave b0,
Phase B0 output = square wave a0
Output switching timing

Figure 15-16 Waveform before and after switching the output SW0 (Trailing edge of the squarewave b0 is in the range of the UC0 cycle)

Switching the phase A0 status from faster than the phase B0 output to delay is previously described. The vice versa is also true; outlier waveform is not generated when switching the phase A0 status from delay to faster than the phase B0. The case of the phase A1 and the phase B1 is the same.

(3-4.2) When the trailing edge of the pulse b0/b1 exceeds the UC0 cycle

As shown in the Figure 15-17, please note the following if you try to change the status of the phase A from fast state to delay state when the phase relation between A0 and B0 is in the condition that the trailing edge exceeds the UC0 cycle. As shown in the Figure 15-18, switch the SW0 at predefined match detection timing of CP00, and change the conditions that the phase A0 output = pulse a0 and the phase B0 output = pulse b0 to the condition that the phase A0 output = pulse b0 and the phase B0 output = pulse a0. As shown in the Figure 15-18, a pulse-type waveform is generated at this time.

And also a pulse-type waveform is generated when switching the phase A0 status from delay to faster than the phase B0, and in the case of the phase A1 and B1 as well.

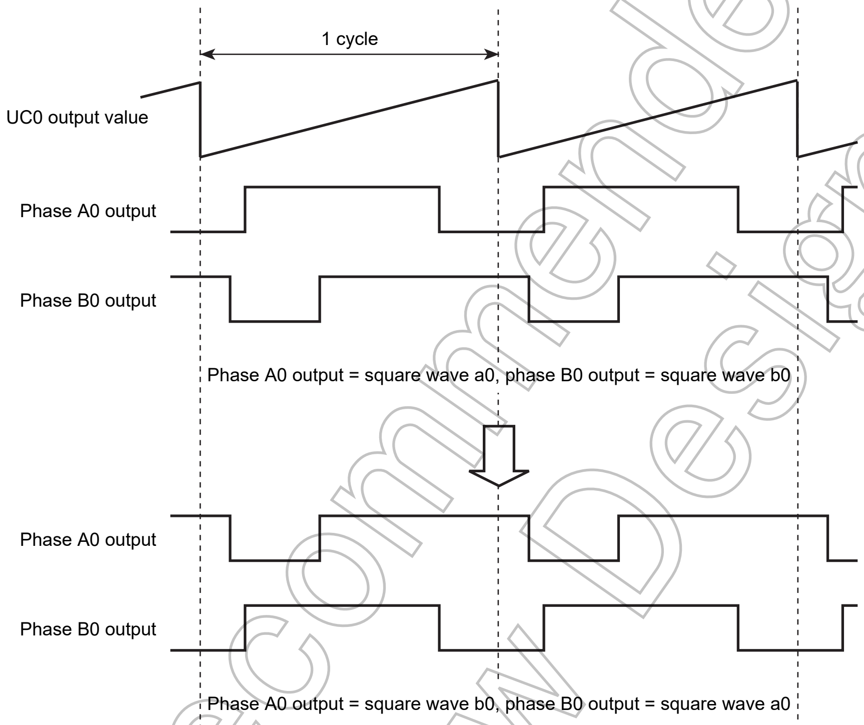

1 cycle
UC0 output value
Phase A0 output
Phase B0 output
Phase A0 output = square wave a0, phase B0 output = square wave b0
Phase A0 output
Phase B0 output
Phase A0 output = square wave b0, phase B0 output = square wave a0

Figure 15-17 Waveform before and after switching the phase delay (output SW) (Trailing edge of the squarewave b0 exceeds the UC0 cycle)

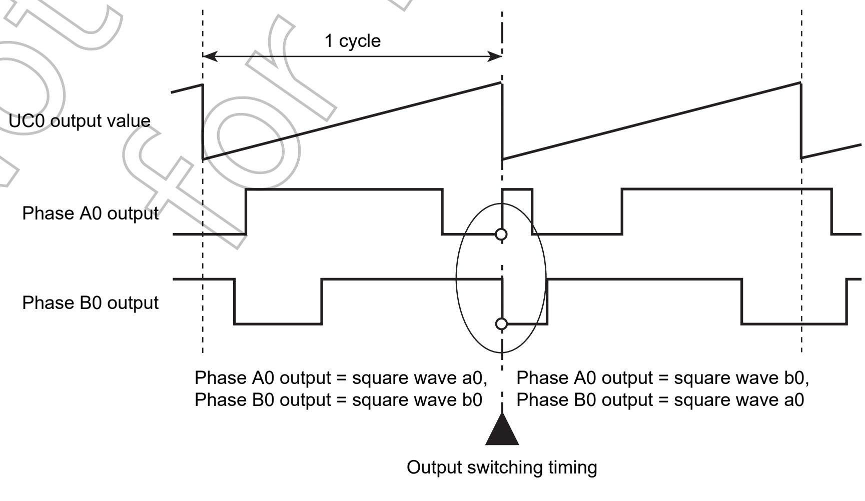

1 cycle
UC0 output value
Phase A0 output
Phase B0 output
Phase A0 output = square wave a0,
Phase B0 output = square wave b0
Phase A0 output = square wave b0,
Phase B0 output = square wave a0
Output switching timing

Figure 15-18 Waveform before and after switching the phase delay (output SW0) (Trailing edge of the squarewave b0 exceeds the UC0 cycle)

Switching the phase A0 status from faster than the phase B0 output to delay is previously described. The vice versa is also true; a pulse-type wavefoem is generated as shown in the Figure 15-18 when switching the phase A0 status from delay to faster than the phase B0. The case of the phase A1 and the phase B1 is the same.

If the pulse-type waveform generated when switching the phase between the delay state and the fast dtate affects the system, it’s preventable through the coordination phase before switching the phases.

15.4.2.3 The setting range of compare register

The setting range of the compare register when 16-bit programmable pulse output is shown in the Table 15-4.

Table 15-4 Setting range of the compare register when 16-bit programmable pulse is output

Timer Unit	Compare register	16-bit programmable pulse generation	
		PPG	Interlock PPG
TMRD0	TD0CP0	0x0001 ≤ CPRG0[15:0] ≤ 0xFFFF	0x0001 ≤ CPRG0[15:0] ≤ 0xFFFF
	TD0CP1	0x0000 ≤ CPRG1[15:0] < CPRG2[15:0]	0x0000 ≤ CPRG1[15:0] < CPRG2[15:0]
	TD0CP2	CPRG1[15:0] < CPRG2[15:0] ≤ CPRG0[15:0]	CPRG1[15:0] < CPRG2[15:0] ≤ CPRG0[15:0]
	TD0CP3	0x0000 ≤ CPRG3[15:0] < CPRG4[15:0]	0x0000 ≤ CPRG3[15:0] < CPRG4[15:0]
	TD0CP4	CPRG3[15:0] < CPRG4[15:0] ≤ CPRG0[15:0]	CPRG3[15:0] < CPRG4[15:0] ≤ CPRG0[15:0]
	TD0CP5	don’t care	0x0000 ≤ CPRG5[15:0] < (CPRG0[15:0] ÷ 2)
TMRD1	TD1CP0	0x0001 ≤ CPRG0[15:0] ≤ 0xFFFF	don’t care
	TD1CP1	0x0000 ≤ CPRG1[15:0] < CPRG2[15:0]	0x0000 ≤ CPRG1[15:0] < CPRG2[15:0]
	TD1CP2	CPRG1[15:0] < CPRG2[15:0] ≤ CPRG0[15:0]	CPRG1[15:0] < CPRG2[15:0] ≤ TD0CP0<CPRG0[15:0]>
	TD1CP3	0x0000 ≤ CPRG3[15:0] < CPRG4[15:0]	0x0000 ≤ CPRG3[15:0] < CPRG4[15:0]
	TD1CP4	CPRG3[15:0] < CPRG4[15:0] ≤ CPRG0[15:0]	CPRG3[15:0] < CPRG4[15:0] ≤ TD0CP0<CPRG0[15:0]>

The cycle Tx of pulse output, when the frequency of TMRDCLK is f_{CLK}x, can be obtained by the following formulas.

(1) In the case of PPG mode, x = 0,1

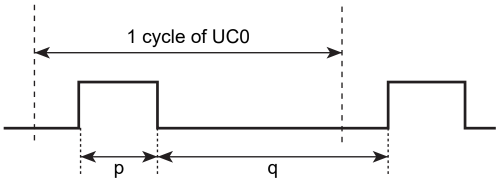
Tx = (1/f_{CLK}x) × (TDxCP0<CPRG0[15:0]> + 1)

(2) In the case of Interlock PPG mode

T = (1/f_{CLK}0) × (TD0CP0<CPRG0[15:0]> + 1)

The duty of the pulse output when the frequency of TMRDCLK0 is f_{CLK}0 can be obtained by the following formula. (in the case of a0)

p : q = {CPRG2[15:0] - CPRG1[15:0]} : {(CPRG0[15:0]+1) - CPRG2[15:0] + CPRG1[15:0]}



16. Serial Channel (SIO/UART)

16.1 Overview

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

Their features are given in the following.

- Transfer Clock
  - Dividing by the prescaler, from the peripheral clock (φT0) frequency into 1/2, 1/8, 1/32, 1/128.
  - Make it possible to divide from the prescaler output clock frequency into 1-16.
  - Make it possible to divide from the prescaler output clock frequency into 1, N+m/16 (N=2-15, m=1-15), 16. (only UART mode)
  - The usable system clock (only UART mode).
- Double Buffer /FIFO

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

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

16.2 Difference in the Specifications of SIO Modules

TMPM341FDXBG/FYXBG has five SIO channels.

Each channel functions independently. The used pins, interrupt, DMA request and UART source clock in each channel are collected in the following.

Table 16-1 Difference in the Specifications of SIO Modules

	Pin name			Interrupt		DMA request	UART source clock
	TXD	RXD	$\overline{\text{CTS}}$x/ SCLKx	Receive Interrupt	Transmit Interrupt		
Channel 0	PE0	PE1	PE2	INTRX0	INTTX0	Support	TB8OUT
Channel 1	PC0	PC1	PC2	INTRX1	INTTX1	-	TB8OUT
Channel 2	PC4	PC5	PC6	INTRX2	INTTX2	Support	TB8OUT
Channel 3	PG4	PG5	PG6	INTRX3	INTTX3	-	TB9OUT
Channel 4	PH0	PH1	PH2	INTRX4	INTTX4	Support	TB9OUT

16.3 Configuration

Figure 16-1 shows SIO block diagram.

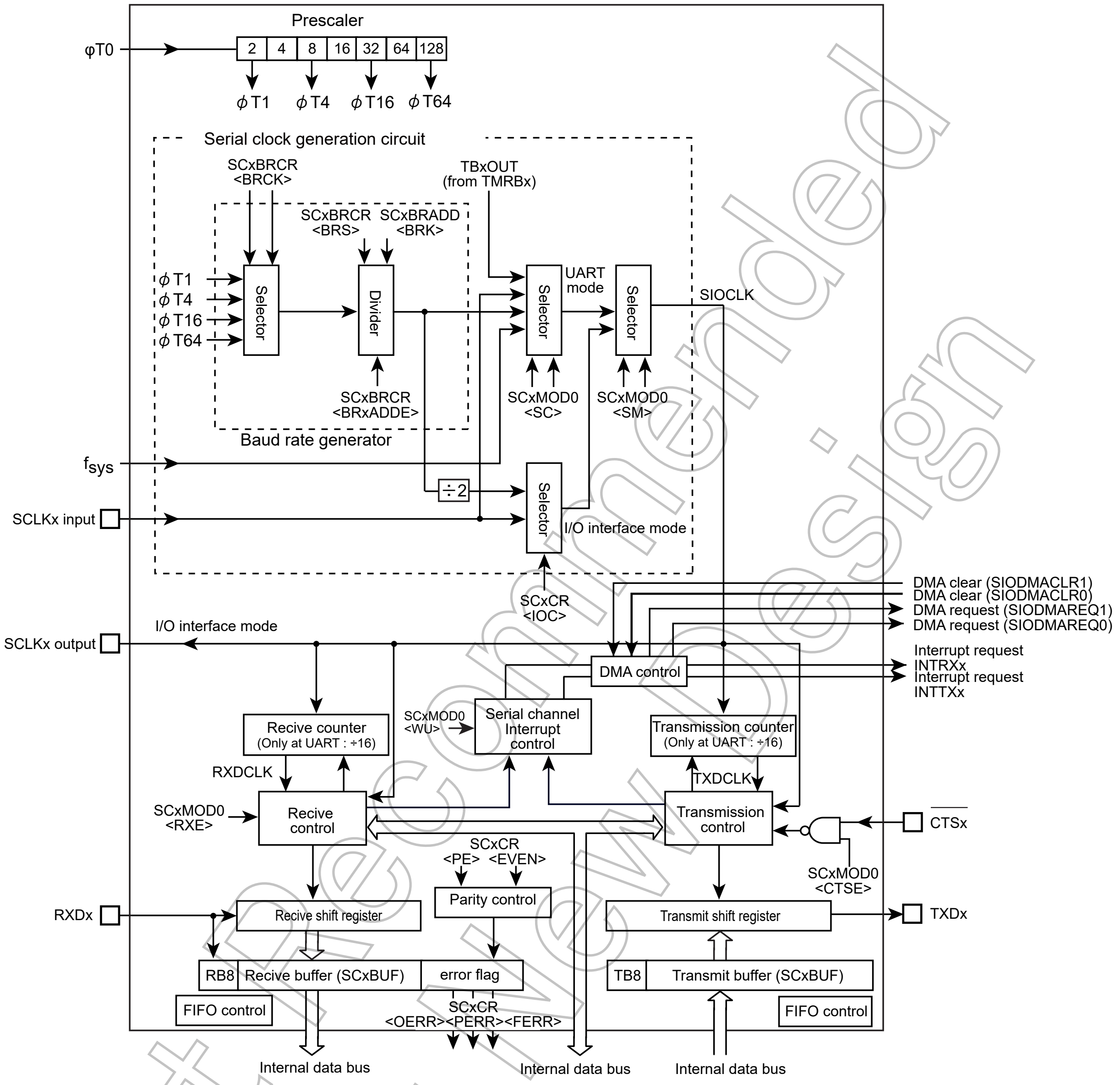


Figure 16-1 SIO Block Diagram

16.4 Registers Description

16.4.1 Registers List in Each Channel

The each channel registers and addresses are shown below.

Channel x	Base Address
Channel0	0x400E_1000
Channel1	0x400E_1100
Channel2	0x400E_1200
Channel3	0x400E_1300
Channel4	0x400E_1400

Register name (x=0 to 4)		Address (Base+)
Enable register	SCxEN	0x0000
Buffer register	SCxBUF	0x0004
Control register	SCxCR	0x0008
Mode control register 0	SCxMOD0	0x000C
Baud rate generator control register	SCxBRCR	0x0010
Baud rate generator control register 2	SCxBRADD	0x0014
Mode control register 1	SCxMOD1	0x0018
Mode control register 2	SCxMOD2	0x001C
RX FIFO configuration register	SCxRFC	0x0020
TX FIFO configuration register	SCxTFC	0x0024
RX FIFO status register	SCxRST	0x0028
TX FIFO status register	SCxTST	0x002C
FIFO configuration register	SCxFCNF	0x0030
DMA request enable register (note2)	SCxDMA	0x0034

Note 1: Do not modify any control register when data is being transmitted or received.
Channel 1 and channel 3 do not have this register.

16.4.2 SCxEN (Enable Register)

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

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

Note: **In case that SCxEN<SIOE>="0" (Stop SIO operation) or the operation mode is changed to IDLE mode with SCxMOD1<I2SC>="0" (Stop SIO operation in IDLE mode), SCxTFC is initiailzed again.**

16.4.3 SCxBUF (Buffer Register)

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

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

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

16.4.4 SCxCR (Control Register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	RB8	R	Receive data bit 8 (For UART) 9th bit of the received data in the 9 bits UART mode.
6	EVEN	R/W	Parity (For UART) 0: Odd 1: Even Selects even or odd parity. "0" : odd parity, "1" : even parity. The parity bit may be used only in the 7- or 8-bit UART mode.
5	PE	R/W	Add parity (For UART) 0: Disabled 1: Enabled Controls enabling/ disabling parity. The parity bit may be used only in the 7- or 8-bit UART mode.
4	OERR	R	Overrun error flag (Note) 0: Normal operation 1: Error
3	PERR	R	Parity / Under-run error flag (Note) 0: Normal operation 1: Error
2	FERR	R	Framing error flag (Note) 0: Normal operation 1: Error
1	SCLKS	R/W	Selecting input clock edge (For I/O Interface) Set to "0" in the clock output mode. 0: Data in the transmit buffer is sent to TXDx pin one bit at a time on the falling edge of SCLKx. Data from RXDx pin isrecieved in the recieve buffer one bit at a time on the rising edge of SCLKx. In this case, the SCLKx starts from high level. 1: Data in the transmit buffer is sent to TXDx pin one bit at a time on the rising edge of SCLKx. Data from RXDx pin isrecieved in the recieve buffer one bit at a time on the falling edge of SCLKx. In this case, the SCLKx starts from low level.
0	IOC	R/W	Selecting clock (For I/O Interface) 0: Baud rate generator 1: SCLK pin input

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

16.4.5 SCxMOD0 (Mode Control Register 0)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	TB8	R/W	Transmit data bit 8 (For UART) Writes the 9th bit of transmit data in the 9 bits UART mode.
6	CTSE	R/W	Handshake function control (For UART) 0: CTS disabled 1: CTS enabled Controls handshake function. Setting "1" enables handshake function using $\overline{\text{CTS}}$ pin.
5	RXE	R/W	Receive control (Note) 0: Disabled 1: Enabled
4	WU	R/W	Wake-up function (For UART) 0: Disabled 1: Enabled This function is available only at 9-bit UART mode. In other mode, this function has no meaning. In it is Enabled, Interrupt only when RB9 = "1" at 9-bit UART mode.
3-2	SM[1:0]	R/W	Specifies transfer mode. 00: I/O interface mode 01: 7-bit length UART mode 10: 8-bit length UART mode 11: 9-bit length UART mode
1-0	SC[1:0]	R/W	Serial transfer clock (For UART) 00: Timer TBxOUT (Note1) 01: Baud rate generator 10: Internal clock fsys 11: External clock (SCLK input) (As for the I/O interface mode, the serial transfer clock can be set in the control register (SCxCR).

Note 1: When setting to the <SC[1:0]>="00", the clock source is different according to the channel. (SIO channel0, channel1 and channel2: TB8OUT, SIO channel3 and channel4: TB9OUT)

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

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

16.4.6 SCxMOD1 (Mode Control Register 1)

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

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

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

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

Note 3: **When SIO is transferred in the full duplex mode, transfer can not be used by SIO transmit or receive.**

Note 4: **In case that SCxEN<SIOE>="0" (Stop SIO operation) or the operation mode is changed to IDLE mode with SCxMOD1<I2SC>="0" (Stop SIO operation in IDLE mode), SCxTFC is initiailzed again.**

16.4.7 SCxMOD2 (Mode Control Register 2)

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-8</td><td>−</td><td>R</td><td>Read as 0.</td></tr>
<tr><td>7</td><td>TBEMP</td><td>R</td><td>Transmit buffer empty flag.
0: Full
1: Empty
If double buffering is disabled, this flag is insignificant.
This flag shows that the transmit double buffers are empty. When data in the transmit double buffers is moved to the transmit shift register and the double buffers are empty, this bit is set to "1".
Writing data again to the double buffers sets this bit to "0".</td></tr>
<tr><td>6</td><td>RBFLL</td><td>R</td><td>Receive buffer full flag.
0: Empty
1: Full
This is a flag to show that the receive double buffers are full.
When a receive operation is completed and received data is moved from the receive shift register to the receive double buffers, this bit changes to "1" while reading this bit changes it to "0".
If double buffering is disabled, this flag is insignificant.</td></tr>
<tr><td>5</td><td>TXRUN</td><td>R</td><td>In transmission flag
0: Stop
1: Operate
This is a status flag to show that data transmission is in progress.
<TXRUN> and <TBEMP> bits indicate the following status.
<table>
<tr><th>&lt;TXRUN&gt;</th><th>&lt;TBEMP&gt;</th><th>Status</th></tr>
<tr><td>1</td><td>−</td><td>Transmission in progress</td></tr>
<tr><td rowspan="2">0</td><td>1</td><td>Transmission completed</td></tr>
<tr><td>0</td><td>Wait state with data in Transmit buffer</td></tr>
</table></td></tr>
<tr><td>4</td><td>SBLEN</td><td>R/W</td><td>STOP bit (for UART)
0 : 1-bit
1 : 2-bit
This specifies the length of transmission stop bit in the UART mode.
On the receive side, the decision is made using only a single bit regardless of the <SBLEN> setting.</td></tr>
<tr><td>3</td><td>DRCHG</td><td>R/W</td><td>Setting transfer direction
0: LSB first
1: MSB first
Specifies the direction of data transfer in the I/O interface mode.
In the UART mode, set this bit to LSB first.</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>2</td><td>WBUF</td><td>R/W</td><td>Double-buffer
0: Disabled
1 : Enabled
This parameter enables or disables the transmit/receive double buffers to transmit (in both SCLK output/input modes) and receive (in SCLK output mode) data in the I/O interface mode and to transmit data in the UART mode.
When receiving data in the I/O interface mode (SCLK input) and UART mode, double buffering is enabled in both cases that 0 or 1 is set to <WBUF&gt; bit.</td></tr>
<tr><td>1-0</td><td>SWRST[1:0]</td><td>R/W</td><td>Software reset
Overwriting "01" in place of "10" generates a software reset. When this software reset is executed, the following bits are initialized and the transmit/receive circuit, the transmit circuit and the FIFO become initial state (see Note1 and Note2).
<table>
<tr><th>Register</th><th>Bit</th></tr>
<tr><td>SCxMOD0</td><td>&lt;RXE&gt;</td></tr>
<tr><td>SCxMOD1</td><td>&lt;TXE&gt;</td></tr>
<tr><td>SCxMOD2</td><td>&lt;TBEMP>, <RBFLL>, <TXRUN&gt;</td></tr>
<tr><td>SCxCR</td><td>&lt;OERR&gt;, <PERR>, <FERR&gt;</td></tr>
<tr><td>SCxDMA
(note2)</td><td>&lt;DMAEN1>, <DMAEN0&gt;</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.

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

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

SCxBRCR

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

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

SCxBRADD

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

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

Table 16-2 lists the settings of baud rate generator division ratio.

Table 16-2 Setting division ratio

	<BRADDE> = "0"	<BRADDE> = "1" (Note1) (Only UART mode)
<BRS>	Specify "N" (Note2) (Note3)	
<BRK>	No setting required	Specify "K" (Note4)
Division ratio	Divide by N	$N + \frac{(16 - K)}{16}$ division.

Note 1: **To use the "N + (16 - K)/16" division function, be sure to set <BRADDE> to "1" after setting the K value to <BRK>. The "N + (16 - K)/16" division function can only be used in the UART mode.**

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

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

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

16.4.9 SCxFCNF (FIFO Configuration Register)

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

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

Note 1: **Regarding TX FIFO, the maximum number of bytes being configured is always available. The available number of bytes is the bytes already written to the TX FIFO.**

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

16.4.10 SCxRFC (RX FIFO Configuration Register)

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-8</td><td>−</td><td>R</td><td>Read as 0.</td></tr>
<tr><td>7</td><td>RFCS</td><td>W</td><td>RX FIFO clear (Note)
1: Clear
When SCxRFC<RFCS> is set to "1", the receive FIFO is cleared and SCxRST<RLVL> is "000". And also the read pointer is initialized.</td></tr>
<tr><td>6</td><td>RFIS</td><td>R/W</td><td>Select interrupt generation condition
0: when the data reaches to the specified fill level.
1: when the data reaches to the specified fill level or the data exceeds the specified fill level at the time data is read.</td></tr>
<tr><td>5-2</td><td>−</td><td>R</td><td>Read as 0.</td></tr>
<tr><td>1-0</td><td>RIL[1:0]</td><td>R/W</td><td>FIFO fill level to generate RX interrupts
<table>
<tr><th></th><th>Half duplex</th><th>Full duplex</th></tr>
<tr><td>00</td><td>4byte</td><td>2byte</td></tr>
<tr><td>01</td><td>1byte</td><td>1byte</td></tr>
<tr><td>10</td><td>2byte</td><td>2byte</td></tr>
<tr><td>11</td><td>3byte</td><td>1byte</td></tr>
</table></td></tr>
</table>

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

16.4.11 SCxTFC (TX FIFO Configuration Register) (Note2)

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-8</td><td>−</td><td>R</td><td>Read as 0.</td></tr>
<tr><td>7</td><td>TFCS</td><td>W</td><td>TX FIFO clear (Note 1)
1: Clears TX FIFO.
When SCxTST<TFCS> is set to "1", the transmit FIFO is cleared and SCxTST<TLVL> is "000". And also the write pointer is initialized.</td></tr>
<tr><td>6</td><td>TFIS</td><td>R/W</td><td>Selects interrupt generation condition.
0: An interrupt is generated when the data reaches to the specified fill level.
1: An interrupt is generated when the data reaches to the specified fill level or the data can not reach the specified fill level at the time new data is read.</td></tr>
<tr><td>5-2</td><td>−</td><td>R</td><td>Read as 0.</td></tr>
<tr><td>1-0</td><td>TIL[1:0]</td><td>R/W</td><td>Selects FIFO fill level.
<table>
<tr><th></th><th>Other than full duplex</th><th>Full duplex</th></tr>
<tr><td>00</td><td>Empty</td><td>Empty</td></tr>
<tr><td>01</td><td>1 byte</td><td>1 byte</td></tr>
<tr><td>10</td><td>2 byte</td><td>Empty</td></tr>
<tr><td>11</td><td>3 byte</td><td>1 byte</td></tr>
</table></td></tr>
</table>

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

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

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

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

16.4.12 SCxRST (RX FIFO Status Register)

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

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

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

16.4.13 SCxTST (TX FIFO Status Register)

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

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

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

16.4.14 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 receive interrupt INTTX. 0: disable 1: enable

Note 1: **When DMA request is generated during DMA transfer is being, it is not kept and nesting.**

Note 2: **Channel 1 and channel 3 do not support DMA request.**

16.5 Operation in Each Mode

Table 16-3 shows the modes and data formats.

Table 16-3 Mode and Data format

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

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

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

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

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

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

16.6 Data Format

16.6.1 Data Format List

Figure 16-2 shows data format.

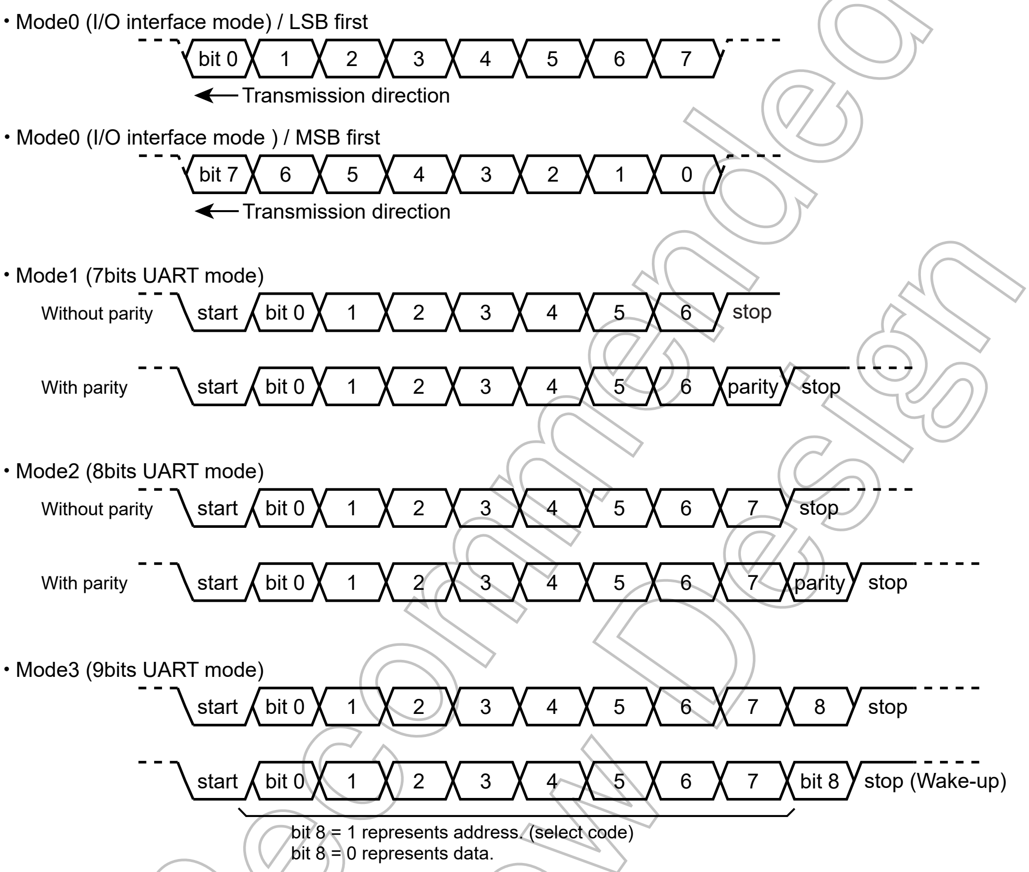

• Mode0 (I/O interface mode) / LSB first
bit 0 1 2 3 4 5 6 7
Transmission direction
• Mode0 (I/O interface mode) / MSB first
bit 7 6 5 4 3 2 1 0
Transmission direction
• Mode1 (7bits UART mode)
Without parity
start bit 0 1 2 3 4 5 6 stop
With parity
start bit 0 1 2 3 4 5 6 parity stop
• Mode2 (8bits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

Figure 16-2 Data Format

16.6.2 Parity Control

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

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

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

16.6.2.1 Transmission

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

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

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

16.6.2.2 Receiving Data

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

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

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

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

16.6.3 STOP Bit Length

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

16.7 Clock Control

16.7.1 Prescaler

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

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

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

The below tables show the resolution of the input clock to the baud rate generator.

Table 16-4 Clock Resolution to the Baud Rate Generator fc = 40 MHz

peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
0 (fgear)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.05 μs)	fc/2^{3} (0.2 μs)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)
		001 (fperiph/2)	fc/2^{2} (0.1 μs)	fc/2^{4} (0.4 μs)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)
		010 (fperiph/4)	fc/2^{3} (0.2 μs)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)
		011 (fperiph/8)	fc/2^{4} (0.4 μs)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)
		100 (fperiph/16)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)
		101 (fperiph/32)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^{2} (0.1 μs)	fc/2^{4} (0.4 μs)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)
		001 (fperiph/2)	fc/2^{3} (0.2 μs)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)
		010 (fperiph/4)	fc/2^{4} (0.4 μs)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)
		011 (fperiph/8)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)
		100 (fperiph/16)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
		101 (fperiph/32)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)	fc/2^{13} (204.8 μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^{3} (0.2 μs)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)
		001 (fperiph/2)	fc/2^{4} (0.4 μs)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)
		010 (fperiph/4)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)
		011 (fperiph/8)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
		100 (fperiph/16)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)	fc/2^{13} (204.8 μs)
		101 (fperiph/32)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)	fc/2^{14} (409.6 μs)
	110 (fc/8)	000 (fperiph/1)	fc/2^{4} (0.4 μs)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)
		001 (fperiph/2)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)
		010 (fperiph/4)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
		011 (fperiph/8)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)	fc/2^{13} (204.8 μs)
		100 (fperiph/16)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)	fc/2^{14} (409.6 μs)
		101 (fperiph/32)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)	fc/2^{13} (204.8 μs)	fc/2^{15} (819.2 μs)
	111 (fc/16)	000 (fperiph/1)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)
		001 (fperiph/2)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
		010 (fperiph/4)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)	fc/2^{13} (204.8 μs)
		011 (fperiph/8)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)	fc/2^{14} (409.6 μs)
		100 (fperiph/16)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)	fc/2^{13} (204.8 μs)	fc/2^{15} (819.2 μs)
		101 (fperiph/32)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)	fc/2^{14} (409.6 μs)	fc/2^{16} (1638 μs)

Table 16-4 Clock Resolution to the Baud Rate Generator fc = 40 MHz

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

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

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

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

Table 16-5 Clock Resolution to the Baud Rate Generator fc = 48 MHz

peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
0 (fgear)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.0417 μs)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)
		001 (fperiph/2)	fc/2^{2} (0.0833 μs)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)
		010 (fperiph/4)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		011 (fperiph/8)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		100 (fperiph/16)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^{2} (0.0833 μs)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)
		001 (fperiph/2)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		010 (fperiph/4)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		011 (fperiph/8)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		100 (fperiph/16)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
		101 (fperiph/32)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)	fc/2^{13} (171 μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		001 (fperiph/2)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		010 (fperiph/4)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		011 (fperiph/8)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
		100 (fperiph/16)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)	fc/2^{13} (171 μs)
		101 (fperiph/32)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3μs)	fc/2^{14} (341 μs)
	110 (fc/8)	000 (fperiph/1)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		001 (fperiph/2)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		010 (fperiph/4)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
		011 (fperiph/8)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)	fc/2^{13} (171 μs)
		100 (fperiph/16)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3μs)	fc/2^{14} (341 μs)
		101 (fperiph/32)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)	fc/2^{13} (171 μs)	fc/2^{15} (683 μs)
	111 (fc/16)	000 (fperiph/1)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		001 (fperiph/2)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
		010 (fperiph/4)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)	fc/2^{13} (171 μs)
		011 (fperiph/8)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3μs)	fc/2^{14} (341 μs)
		100 (fperiph/16)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)	fc/2^{13} (171 μs)	fc/2^{15} (683 μs)
		101 (fperiph/32)	fc/2^{10} (21.4 μs)	fc/2^{12} (85.4 μs)	fc/2^{14} (342 μs)	fc/2^{16} (1366 μs)

Table 16-5 Clock Resolution to the Baud Rate Generator fc = 48 MHz

peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
1 (fc)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.0417 μs)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)
		001 (fperiph/2)	fc/2^{2} (0.0833 μs)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)
		010 (fperiph/4)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		011 (fperiph/8)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		100 (fperiph/16)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
	100 (fc/2)	000 (fperiph/1)	–	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)
		001 (fperiph/2)	fc/2^{2} (0.0833 μs)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)
		010 (fperiph/4)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		011 (fperiph/8)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		100 (fperiph/16)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
	101 (fc/4)	000 (fperiph/1)	–	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)
		001 (fperiph/2)	–	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)
		010 (fperiph/4)	fc/2^{3} (0.167 μs)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		011 (fperiph/8)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		100 (fperiph/16)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
	110 (fc/8)	000 (fperiph/1)	–	–	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)
		001 (fperiph/2)	–	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)
		010 (fperiph/4)	–	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		011 (fperiph/8)	fc/2^{4} (0.333 μs)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		100 (fperiph/16)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
	111 (fc/16)	000 (fperiph/1)	–	–	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)
		001 (fperiph/2)	–	–	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)
		010 (fperiph/4)	–	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)
		011 (fperiph/8)	–	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)
		100 (fperiph/16)	fc/2^{5} (0.667 μs)	fc/2^{7} (2.67 μs)	fc/2^{9} (10.7 μs)	fc/2^{11} (42.7 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)

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

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

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

Table 16-6 Clock Resolution to the Baud Rate Generator fc = 54 MHz

peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
0 (fgear)	000 (fc)	000 (fperiph/1)	$fc/2^{1}$ (0.0370 μs)	$fc/2^{3}$ (0.148 μs)	$fc/2^{5}$ (0.593 μs)	$fc/2^{7}$ (2.37 μs)
		001 (fperiph/2)	$fc/2^{2}$ (0.0741 μs)	$fc/2^{4}$ (0.296 μs)	$fc/2^{6}$ (1.19 μs)	$fc/2^{8}$ (4.74 μs)
		010 (fperiph/4)	$fc/2^{3}$ (0.148 μs)	$fc/2^{5}$ (0.593 μs)	$fc/2^{7}$ (2.37 μs)	$fc/2^{9}$ (9.48 μs)
		011 (fperiph/8)	$fc/2^{4}$ (0.296 μs)	$fc/2^{6}$ (1.19 μs)	$fc/2^{8}$ (4.74 μs)	$fc/2^{10}$ (19.0 μs)
		100 (fperiph/16)	$fc/2^{5}$ (0.593 μs)	$fc/2^{7}$ (2.37 μs)	$fc/2^{9}$ (9.48 μs)	$fc/2^{11}$ (37.9 μs)
		101 (fperiph/32)	$fc/2^{6}$ (1.19 μs)	$fc/2^{8}$ (4.74 μs)	$fc/2^{10}$ (19.0 μs)	$fc/2^{12}$ (75.9 μs)
	100 (fc/2)	000 (fperiph/1)	$fc/2^{2}$ (0.0741 μs)	$fc/2^{4}$ (0.296 μs)	$fc/2^{6}$ (1.19 μs)	$fc/2^{8}$ (4.74 μs)
		001 (fperiph/2)	$fc/2^{3}$ (0.148 μs)	$fc/2^{5}$ (0.593 μs)	$fc/2^{7}$ (2.37 μs)	$fc/2^{9}$ (9.48 μs)
		010 (fperiph/4)	$fc/2^{4}$ (0.296 μs)	$fc/2^{6}$ (1.19 μs)	$fc/2^{8}$ (4.74 μs)	$fc/2^{10}$ (19.0 μs)
		011 (fperiph/8)	$fc/2^{5}$ (0.593 μs)	$fc/2^{7}$ (2.37 μs)	$fc/2^{9}$ (9.48 μs)	$fc/2^{11}$ (37.9 μs)
		100 (fperiph/16)	$fc/2^{6}$ (1.19 μs)	$fc/2^{8}$ (4.74 μs)	$fc/2^{10}$ (19.0 μs)	$fc/2^{12}$ (75.9 μs)
		101 (fperiph/32)	$fc/2^{7}$ (2.37 μs)	$fc/2^{9}$ (9.48 μs)	$fc/2^{11}$ (37.9 μs)	$fc/2^{13}$ (152 μs)
	101 (fc/4)	000 (fperiph/1)	$fc/2^{3}$ (0.148 μs)	$fc/2^{5}$ (0.593 μs)	$fc/2^{7}$ (2.37 μs)	$fc/2^{9}$ (9.48 μs)
		001 (fperiph/2)	$fc/2^{4}$ (0.296 μs)	$fc/2^{6}$ (1.19 μs)	$fc/2^{8}$ (4.74 μs)	$fc/2^{10}$ (19.0 μs)
		010 (fperiph/4)	$fc/2^{5}$ (0.593 μs)	$fc/2^{7}$ (2.37 μs)	$fc/2^{9}$ (9.48 μs)	$fc/2^{11}$ (37.9 μs)
		011 (fperiph/8)	$fc/2^{6}$ (1.19 μs)	$fc/2^{8}$ (4.74 μs)	$fc/2^{10}$ (19.0 μs)	$fc/2^{12}$ (75.9 μs)
		100 (fperiph/16)	$fc/2^{7}$ (2.37 μs)	$fc/2^{9}$ (9.48 μs)	$fc/2^{11}$ (37.9 μs)	$fc/2^{13}$ (152 μs)
		101 (fperiph/32)	$fc/2^{8}$ (4.74 μs)	$fc/2^{10}$ (19.0 μs)	$fc/2^{12}$ (75.9μs)	$fc/2^{14}$ (303 μs)
	110 (fc/8)	000 (fperiph/1)	$fc/2^{4}$ (0.296 μs)	$fc/2^{6}$ (1.19 μs)	$fc/2^{8}$ (4.74 μs)	$fc/2^{10}$ (19.0 μs)
		001 (fperiph/2)	$fc/2^{5}$ (0.593 μs)	$fc/2^{7}$ (2.37 μs)	$fc/2^{9}$ (9.48 μs)	$fc/2^{11}$ (37.9 μs)
		010 (fperiph/4)	$fc/2^{6}$ (1.19 μs)	$fc/2^{8}$ (4.74 μs)	$fc/2^{10}$ (19.0 μs)	$fc/2^{12}$ (75.9 μs)
		011 (fperiph/8)	$fc/2^{7}$ (2.37 μs)	$fc/2^{9}$ (9.48 μs)	$fc/2^{11}$ (37.9 μs)	$fc/2^{13}$ (152 μs)
		100 (fperiph/16)	$fc/2^{8}$ (4.74 μs)	$fc/2^{10}$ (19.0 μs)	$fc/2^{12}$ (75.9μs)	$fc/2^{14}$ (303 μs)
		101 (fperiph/32)	$fc/2^{9}$ (9.48 μs)	$fc/2^{11}$ (37.9 μs)	$fc/2^{13}$ (152 μs)	$fc/2^{15}$ (607 μs)
	111 (fc/16)	000 (fperiph/1)	$fc/2^{5}$ (0.593 μs)	$fc/2^{7}$ (2.37 μs)	$fc/2^{9}$ (9.48 μs)	$fc/2^{11}$ (37.9 μs)
		001 (fperiph/2)	$fc/2^{6}$ (1.19 μs)	$fc/2^{8}$ (4.74 μs)	$fc/2^{10}$ (19.0 μs)	$fc/2^{12}$ (75.9 μs)
		010 (fperiph/4)	$fc/2^{7}$ (2.37 μs)	$fc/2^{9}$ (9.48 μs)	$fc/2^{11}$ (37.9 μs)	$fc/2^{13}$ (152 μs)
		011 (fperiph/8)	$fc/2^{8}$ (4.74 μs)	$fc/2^{10}$ (19.0 μs)	$fc/2^{12}$ (75.9μs)	$fc/2^{14}$ (303 μs)
		100 (fperiph/16)	$fc/2^{9}$ (9.48 μs)	$fc/2^{11}$ (37.9 μs)	$fc/2^{13}$ (152 μs)	$fc/2^{15}$ (607 μs)
		101 (fperiph/32)	$fc/2^{10}$ (19.0 μs)	$fc/2^{12}$ (75.9 μs)	$fc/2^{14}$ (303 μs)	$fc/2^{16}$ (1210 μs)

Table 16-6 Clock Resolution to the Baud Rate Generator fc = 54 MHz

peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
1 (fc)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.0370 μs)	fc/2^{3} (0.148 μs)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)
		001 (fperiph/2)	fc/2^{2} (0.0741 μs)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)
		010 (fperiph/4)	fc/2^{3} (0.148 μs)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)
		011 (fperiph/8)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)
		100 (fperiph/16)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)
		101 (fperiph/32)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)	fc/2^{12} (75.9 μs)
	100 (fc/2)	000 (fperiph/1)	−	fc/2^{3} (0.148 μs)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)
		001 (fperiph/2)	fc/2^{2} (0.0741 μs)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)
		010 (fperiph/4)	fc/2^{3} (0.148 μs)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)
		011 (fperiph/8)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)
		100 (fperiph/16)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
	101 (fc/4)	000 (fperiph/1)	−	fc/2^{3} (0.148 μs)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)
		001 (fperiph/2)	−	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)
		010 (fperiph/4)	fc/2^{3} (0.148 μs)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)
		011 (fperiph/8)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)
		100 (fperiph/16)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
	110 (fc/8)	000 (fperiph/1)	−	−	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)
		001 (fperiph/2)	−	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)
		010 (fperiph/4)	−	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)
		011 (fperiph/8)	fc/2^{4} (0.296 μs)	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)
		100 (fperiph/16)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)
	111 (fc/16)	000 (fperiph/1)	−	−	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)
		001 (fperiph/2)	−	−	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)
		010 (fperiph/4)	−	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)
		011 (fperiph/8)	−	fc/2^{6} (1.19 μs)	fc/2^{8} (4.74 μs)	fc/2^{10} (19.0 μs)
		100 (fperiph/16)	fc/2^{5} (0.593 μs)	fc/2^{7} (2.37 μs)	fc/2^{9} (9.48 μs)	fc/2^{11} (37.9 μs)
		101 (fperiph/32)	fc/2^{6} (1.33 μs)	fc/2^{8} (5.33 μs)	fc/2^{10} (21.3 μs)	fc/2^{12} (85.3 μs)

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

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

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

16.7.2 Serial Clock Generation Circuit

The serial clock circuit is a block to generate transmit and receive clocks (SIOCLK) and consists of the circuits in which clocks can be selected by the settings of the baud rates generator and modes.

16.7.2.1 Baud Rate Generator

The baud rate generator generates transmit and receive clocks to determine the serial channel transfer rate.

(1) Baud Rate Generator input clock

The input clock of the baud rate generator is selected from the prescaler outputs divided by 2, 8, 32 and 128.

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

(2) Baud Rate Generator output clock

The frequency division ratio of the output clock in the baud rate generator is set by SCxBRCR and SCxBRADD.

The following frequency divide ratios can be used; 1/N frequency division in the I/O interface mode ,either 1/N or N + (16-K)/16 in the UART mode.

The table below shows the frequency division ratio which can be selected.

Mode	Divide Function Setting SCxBRCR<BRADDE>	Divide by N SCxBRCR<BRS>	Divide by K SCxBRADD<BRK>
I/O interface	Divide by N	1 to 16 (Note)	-
UART	Divide by N	1 to 16	-
	N + (16-K)/16 division	2 to 15	1 to 15

Note: 1/N (N=1)frequency division ratio can be used only when a double buffer is enabled.

16.7.2.2 Clock Selection Circuit

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

Modes can be specified by setting the SCxMOD0<SM>.

The input clock in I/O interface mode is selected by setting SCxCR. The clock in UART mode is selected by setting SCxMOD0<SC>.

(1) Transfer Clock in I/O interface mode

Table 16-7 shows clock selection in I/O interface mode.

Table 16-7 Clock Selection in I/O Interface Mode

Mode SCxMOD0<SM>	Input/Output selection SCxCR<IOC>	Clock edge selection SCxCR<SCLKS>	Clock of use
I/O interface mode	SCLK output	Set to "0". (Fixed to the rising edge)	Divided by 2 of the baud rate generator output.
	SCLK input	Rising edge	SCLK input rising edge
		Falling edge	SCLK input falling edge

To get the highest baud rate, the baud rate generator must be set as below.

Note: **When deciding clock settings, make sure that AC electrical character is satisfied.**

- Clock/mode control block settings
  - fc = 40MHz
  - fgear = 40MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
  - φT0 = 40MHz (CGSYSCR<PRCK[2:0]> = "000" : 1 division ratio)
- SIO settings (if double buffer is used)
  - Clock (SCxBRCR<BRCK[1:0]> = "00" : φT1 selected) = 20MHz
  - Divided clock frequency (SCxBRCR<BRS[3:0]> = "0001" : 1 division ratio) = 20MHz

  1 division ratio can be selected if double buffer is used. In this case, baud rate is 10Mbps because 20MHz is divided by 2.
- SIO settings (if double buffer is not used)
  - Clock (SCxBRCR<BRCK[1:0]> = "00" : φT1 selected) = 20MHz
  - Divided clock frequency (SCxBRCR<BRS[3:0]> = "0010" : 2 division ratio) = 10MHz

  2 division ratio is the highest if double buffer is not used. In this case, baud rate is 5Mbps because 10MHz is divided by 2.

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

- If double buffer is used

  - SCLK cycle > 6/fsys

  The highest baud rate is less than 54 ÷ 6 = 9 Mbps.
- If double buffer is not used

  - SCLK cycle > 8/fsys

The highest baud rate is less than 54 ÷ 8 = 6.75 Mbps.

(2) Transfer clock in the UART mode

Table 16-8 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 16-8 Clock Selection in UART Mode

Mode SCxMOD0<SM>	Clock selection SCxMOD0<SC>
UART Mode	Timer output
	Baud rate generator
	fsys
	SCLK input

The examples of baud rate in each clock settings.

- If the baud rate generator is used
  - fc = 40MHz
  - fgear = 40MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
  - φT0 = 40MHz (CGSYSCR<PRCK[2:0]> = "000" : 1 division ratio)
  - Clock = ΦT1 = 20MHz (SCxBRCR<BRCK[1:0]> = "00" : φT1 selected)

  The highest baud rate is 1.25Mbps because 20MHz is divided by 16.

  Table 16-9 shows examples of baud rate when the baud rate generator is used with the following clock settings.

  - fc = 9.8304MHz
  - fgear = 9.8304MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
  - φT0 = 4.9152MHz (CGSYSCR<PRCK[2:0]> = "001" : 2 division ratio)

Table 16-9 Example of UART Mode Baud Rate (Using the Baud Rate Generator)

fc [MHz]	Division ratio N (SCxBRCR<BRS>)	φT1 (fc/4)	φT4 (fc/16)	φT16 (fc/64)	φT64 (fc/256)
9.830400	2	76.800	19.200	4.800	1.200
	4	38.400	9.600	2.400	0.600
	8	19.200	4.800	1.200	0.300
	16	9.600	2.400	0.600	0.150

Unit : kbps

- If the SCLK input is used

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

  - SCLK cycle > 2/fsys

  The highest baud rate must be less than 54 ÷ 2 ÷ 16 = 1.68 Mbps.

- If fsys is used

  Since the highest value of fsys is 54MHz, the highest baud rate is 54 ÷ 16 = 3.37Mbps.

- If timer output is used

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

Baud rates can be obtained by using the following formula.

Baud rate calculation

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by CGSYSCR<PRCK[1:0]>}}{\underline{(\text{TBxRG1} \times 2)} \times \underline{2} \times 16}$$

In the case the timer prescaler clock ΦT1 (2divition ratio) is selected.

One clock cycle is a period that the timer flip-flop is inverted twice.

Table 16-10 shows the examples of baud rates when the timer output is used with the following clock settings.

- fc = 32MHz / 9.8304MHz / 8MHz
- fgear = 32MHz / 9.8304MHz / 8MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
- φT0 = 16MHz / 4.9152MHz / 4MHz (CGSYSCR<PRCK[2:0]> = "001" : 2 division ratio)
- Timer count clock = 4MHz / 1.2287MHz / 1MHz (TBxMOD<TBCLK[1:0]> = "01" : φT1 selected)

Table 16-10 Example of UART Mode Baud Rate (Using the Timer Output)

TBxRG0 setting	fc		
	32MHz	9.8304MHz	8MHz
0x0001	250	76.8	62.5
0x0002	125	38.4	31.25
0x0003	-	25.6	-
0x0004	62.5	19.2	15.625
0x0005	50	15.36	12.5
0x0006	-	12.8	-
0x0008	31.25	9.6	-
0x000A	25	7.68	6.25
0x0010	15.625	4.8	-
0x0014	12.5	3.84	3.125

Unit : kbps

16.8 Transmit/Receive Buffer and FIFO

16.8.1 Configuration

Figure 16-3 shows the configuration of transmit buffer, receive buffer and FIFO.

Appropriate settings are required for using buffer and FIFO. The configuration may be predefined depending on the mode.

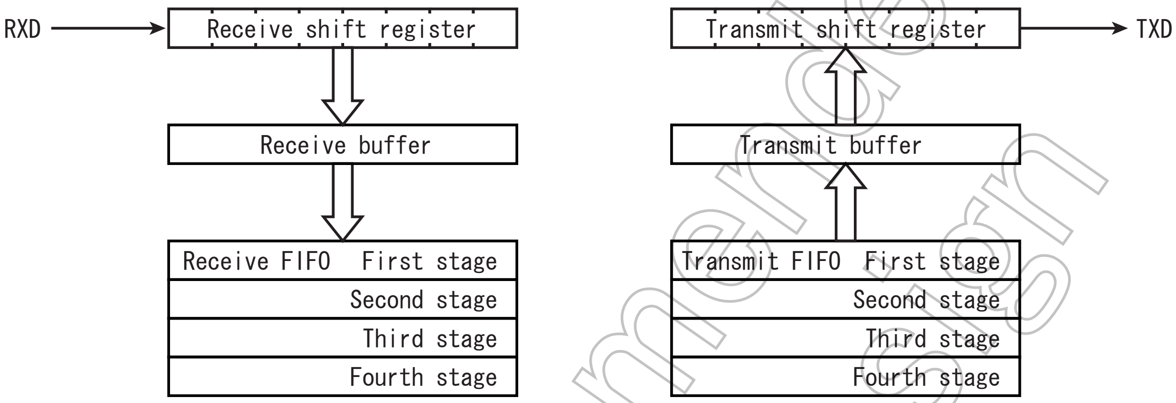


Figure 16-3 The Configuration of Buffer and FIFO

16.8.2 Transmit/Receive Buffer

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

In the case of using a receive buffer, if SCLK input is set to generate clock output in the I/O interface mode or the UART mode is selected, it’s double buffered despite the <WBUF> settings. In other modes, it’s according to the <WBUF> settings.

Table 16-11 shows correlation between modes and buffers.

Table 16-11 Mode and buffer Composition

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

16.8.3 FIFO

In addition to the double buffer function above described, 4-byte FIFO can be used.

To enable FIFO, enable the double buffer by setting SCxMOD2<WBUF> to "1" and SCxFCNF<CNFG> to "1". The FIFO buffer configuration is specified by SCxMOD1<FDPX[1:0]>.

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

Table 16-12 shows correlation between modes and FIFO.

Table 16-12 Mode and FIFO Composition

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

16.9 Status Flag

The SCxMOD2 register has two types of flag. This bit is significant only when the double buffer is enabled.

<RBFLL> is a flag to show that the receive buffer is full. When one frame of data is received and the data is moved from the receive shift register to the receive buffers, this bit changes to "1" while reading this bit changes it to "0".

<TBEMP> shows that the transmit buffers are empty. When data in the transmit buffers is moved to the transmit shift register, this bit is set to "1" When data is set to the transmit buffers, the bit is cleared to "0".

16.10 Error Flag

Three error flags are provided in the SCxCR register. The meaning of the flags is changed depending on the modes. The table below shows the meanings in each mode.

These flags are cleared to "0" after reading the SCxCR register.

Mode	Flag		
	<OERR>	<PERR>	<FERR>
UART	Overrun error	Parity error	Framing error
I/O Interface (SCLK input)	Overrun error	Underrun error (When using double buffer or FIFO)	Fixed to 0
		Fixed to 0 (When a double buffer and FIFO unused)	
I/O Interface (SCLK output)	Undefined	Undefined	Fixed to 0

16.10.1 OERR Flag

In both UART and I/O interface modes, this bit is set to "1" when an error is generated by completing the reception of the next frame of receive data before the receive buffer has been read. If the receive FIFO is enabled, the received data is automatically moved to the receive FIFO and no overrun error will be generated until the receive FIFO is full (or until the usable bytes are fully occupied).

In the I/O interface with SCLK output mode, the SCLK output stops upon setting the flag.

Note: To switch the I/O interface SCLK output mode to other modes, read the SCxCR register and clear the overrun flag.

16.10.2 PERR Flag

This flag indicates a parity error in the UART mode and an under-run error in the I/O interface mode.

In the UART mode, <PERR> is set to "1" when the parity generated from the received data is different from the parity received.

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

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

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

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

16.10.3 FERR Flag

A framing error is generated if the corresponding stop bit is determined to be "0" by sampling the bit at around the center. Regardless of the stop bit length settings in the SCxMOD2<SBLEN>register, the stop bit status is determined by only 1.

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

16.11 Receive

16.11.1 Receive Counter

The receive counter is a 4-bit binary counter and is up-counted by SIOCLK. In the UART mode, sixteen SIOCLK clock pulses are used in receiving a single data bit and the data symbol is sampled at the seventh, eighth, and ninth pulses. From these three samples, majority logic is applied to decide the received data.

16.11.2 Receive Control Unit

16.11.2.1 I/O interface mode

In the SCLK output mode with SCxCR <IOC> set to "0", the RXD pin is sampled on the rising edge of the shift clock outputted to the SCLK pin.

In the SCLK input mode with SCxCR <IOC> set to "1", the serial receive data RXD pin is sampled on the rising or falling edge of SCLK input signal depending on the SCxCR <SCLKS> setting.

16.11.2.2 UART Mode

The receive control unit has a start bit detection circuit, which is used to initiate receive operation when a normal start bit is detected.

16.11.3 Receive Operation

16.11.3.1 Receive Buffer

The received data is stored by 1 bit in the receive shift register. When a complete set of bits has been stored, the interrupt INTRXx is generated.

When the double buffer is enabled, the data is moved to the receive buffer (SCxBUF) and the receive buffer full flag (SCxMOD2<RBFLL>) is set to "1". The receive buffer full flag is "0" cleared by reading the receive buffer.

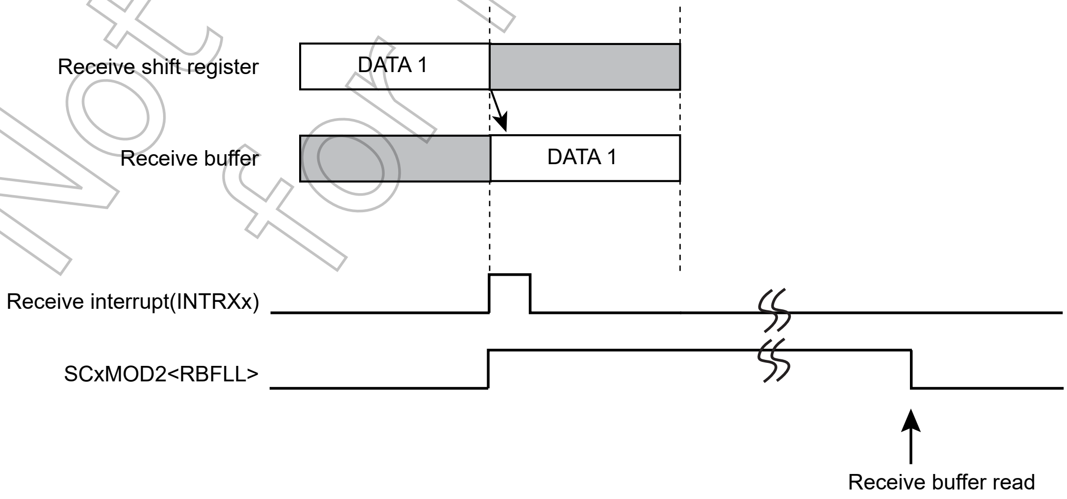


Figure 16-4 Receive Buffer Operation

16.11.3.2 Receive FIFO Operation

When FIFO is enabled, the received data is moved from receive buffer to receive FIFO and the receive buffer full flag is cleared immediately. An interrupt will be generated according to the SCxRFC<RIL> setting.

Note: When the data with parity bit are received in UART mode by using the FIFO, the parity error flag is shown the occurring the parity error in the received data.

The following describes configurations and operations in the half duplex RX mode.

SCxMOD1[6:5] =01	: Transfer mode is set to half duplex mode
SCxFCNF[4:0] = 10111	: Automatically inhibits continuous reception after reaching the fill level. : The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.
SCxRFC[1:0] = 00	: The fill level of FIFO in which generated receive interrupt is set to 4-byte.
SCxRFC[7:6] = 11	: Clears receive FIFO and sets the condition of interrupt generation.

After setting of the above FIFO configuration, the data reception is started by writing "1" to the SCxMOD0 <RXE>. When the data is stored all in the receive shift register, receive buffer and receive FIFO, SCxMOD0<RXE> is automatically cleared and the receive operation is finished.

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

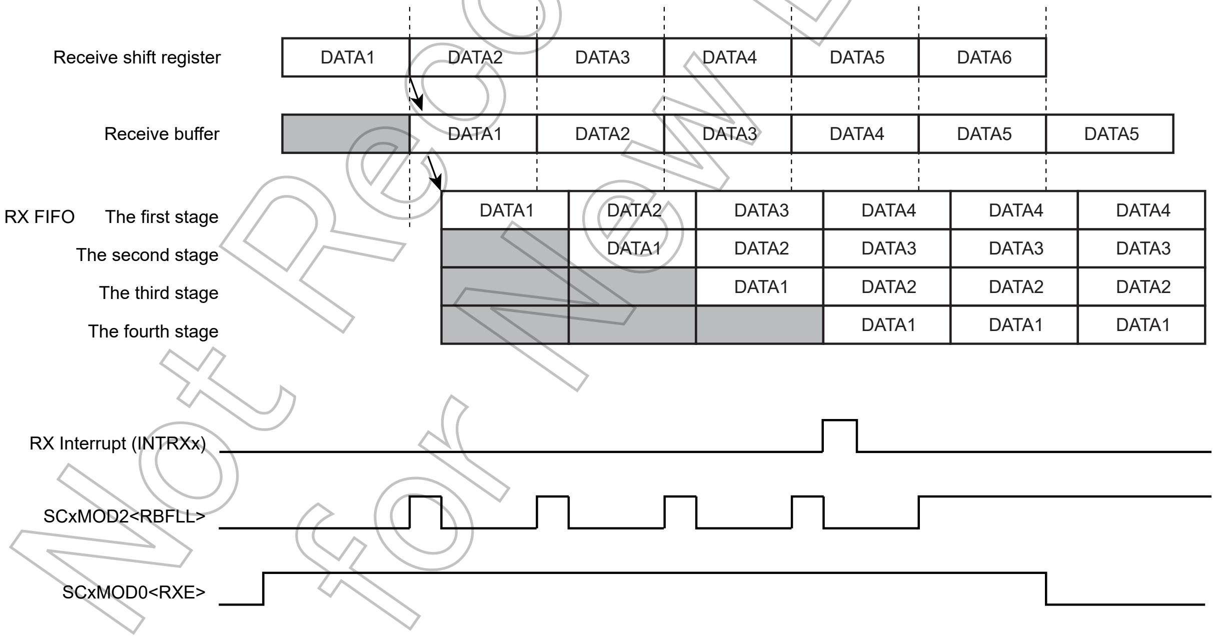


Figure 16-5 Receive FIFO Operation

16.11.3.3 I/O interface mode with SCLK output

In the I/O interface mode and SCLK output setting, SCLK output stops when all received data is stored in the receive buffer and FIFO. So, in this mode, the overrun error flag has no meaning.

The timing of SCLK output stop and re-output depends on receive buffer and FIFO.

(1) Case of single buffer

Stop SCLK output after receiving a data. In this mode, I/O interface can transfer each data with the transfer device by hand-shake.

When the data in a buffer is read, SCLK output is restarted.

(2) Case of double buffer

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

When the data is read, SCLK output is restarted.

(3) Case of FIFO

Stop SCLK output after receiving the data into a shift register, received buffer and FIFO.

When one byte data is read, the data in the received buffer is transferred into FIFO and the data in the receive shift register is transferred into received buffer and SCLK output is restarted.

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

16.11.3.4 Read Received Data

In spite of enabling or disabling FIFO, read the received data from the receive buffer (SCxBUF).

When receive FIFO is disabled, the buffer full flag SCxMOD2<RBFLL> is cleared to "0" by this reading. In the case of the next data can be received in the receive shift register before reading a data from the receive buffer. The parity bit to be added in the 8-bit UART mode as well as the most significant bit in the 9-bit UART mode will be stored in SCxCR<RB8>.

When the receive FIFO is available, the 9-bit UART mode is prohibited because up to 8-bit data can be stored in FIFO. In the 8-bit UART mode, the parity bit is lost but parity error is determined and the result is stored in SCxCR<PERR>.

16.11.3.5 Wake-up Function

In the 9-bit UART mode, the slave controller can be operated in the wake-up mode by setting the wake-up function SCxMOD0 <WU> to "1." In this case, the interrupt INTRXx will be generated only when SCxCR <RB8> is set to "1."

16.11.3.6 Overrun Error

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

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

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

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

16.12 Transmission

16.12.1 Transmission Counter

The transmit counter is a 4-bit binary counter and is counted by SIOCLK as in the case of the receive counter. In UART mode, it generates a transmit clock (TXDCLK) on every 16th clock pulse.

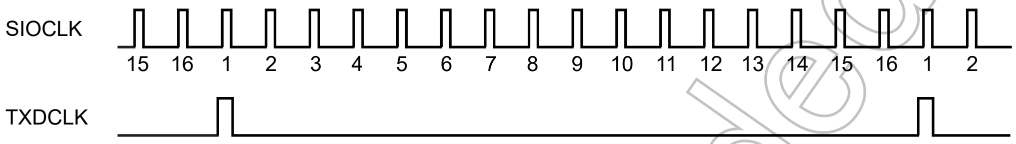


Figure 16-6 Generation of Transmission Clock

16.12.2 Transmission Control

16.12.2.1 I/O Interface Mode

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

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

16.12.2.2 UART Mode

When the transmit data is written in the transmit buffer, data transmission is initiated on the rising edge of the next TXDCLK and the transmit shift clock signal is also generated.

16.12.3 Transmit Operation

16.12.3.1 Operation of Transmission Buffer

If double buffering is disabled, the CPU writes data only to Transmit shift Buffer and the transmit interrupt INTTXx is generated upon completion of data transmission.

If double buffering is enabled (including the case the transmit FIFO is enabled), data written to the transmit buffer is moved to the transmit shift register. The INTTXx interrupt is generated at the same time and the transmit buffer empty flag (SCxMOD2<TBEMP>) is set to "1". This flag indicates that the next transmit data can be written. When the next data is written to the transmit buffer, the <TBEMP> flag is cleared to "0".

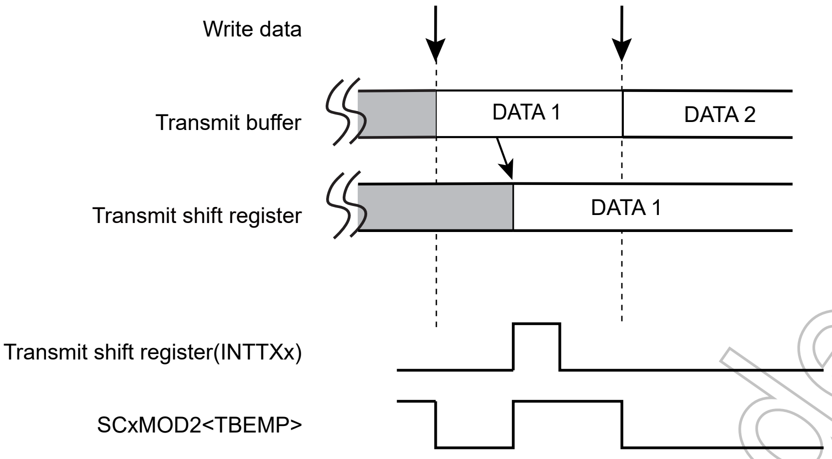


Figure 16-7 Operation of Transmission Buffer (Double-buffer is enabled)

16.12.3.2 Transmit FIFO Operation

When FIFO is enabled, the maximum 5-byte data can be stored using the transmit buffer and FIFO. Once transmission is enabled, data is transferred to the transmit shift register from the transmit buffer and start transmission. If data exists in the FIFO, the data is moved to the transmit buffer immediately, and the <TBEMP> flag is cleared to "0".

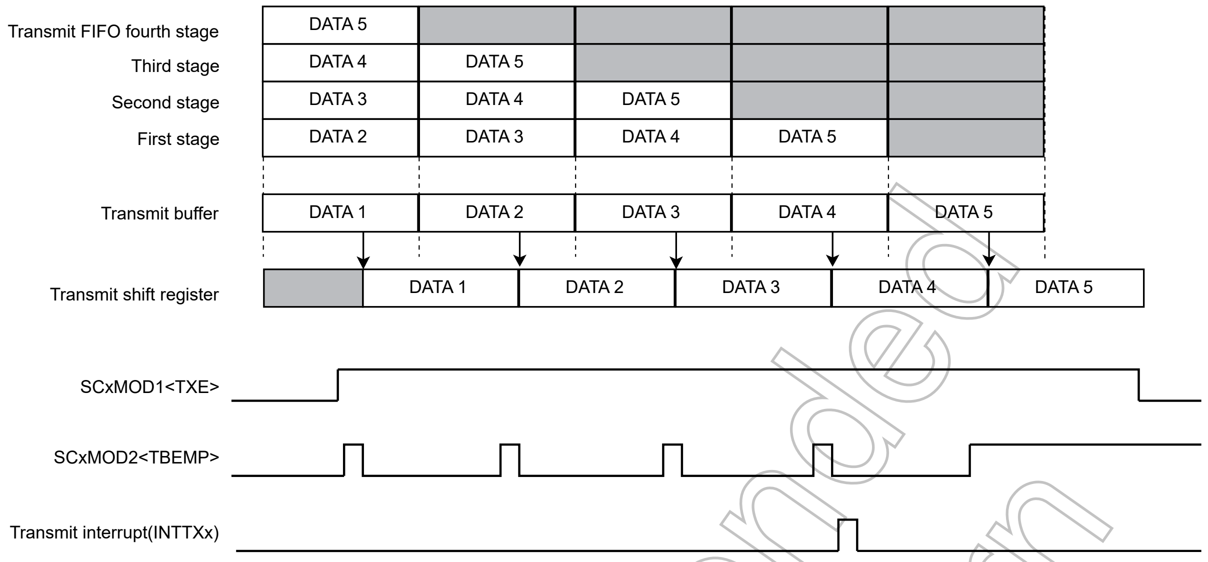
Note: **To use TX FIFO buffer, TX FIFO must be cleared after setting the SIO transfer mode (half duplex/full duplex) and enabling FIFO (SCxFCNF<CNFG> = "1").**

Settings and operations to transmit 4-byte data stream by setting the transfer mode to half duplex are shown as below.

SCxMOD1[6:5] =10	: Transfer mode is set to half duplex.
SCxFCNF[4:0] = 11011	: Transmission is automatically disabled if FIFO becomes empty. The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.
SCxTFC[1:0] = 00	: Sets the interrupt generation fill level to "0".
SCxTFC[7:6] = 11	: Clears receive FIFO and sets the condition of interrupt generation.

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

Once above settings are configured, if the transmission is not set as auto disabled, the transmission should lasts by writing transmit data.



16.12.3.3 I/O interface Mode/Transmission by SCLK Output

If SCLK is set to generate clock the I/O interface mode, the SCLK output automatically stops when all data transmission is completed and underrun error will not occur.

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

(1) Single Buffer

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

(2) Double Buffer

The SCLK output stops upon completion of data transmission of the transmit shift register and the transmit buffer. The SCLK output resumes when the next data is written in the buffer.

(3) FIFO

The transmission of all data stored in the transmit shift register, transmit buffer and FIFO is completed, the SCLK output stops. The next data is written, SCLK output resumes.

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

16.12.3.4 Under-run error

If the transmit FIFO is disabled in the I/O interface SCLK input mode and if no data is set in transmit buffer before the next frame clock input, which occurs upon completion of data transmission from transmit shift register, an under-run error occurs and SCxCR<PERR> is set to "1".

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

Note: **Before switching the I/O interface SCLK output mode to other modes, read the SCxCR register and clear the underrun flag.**

16.13 Handshake function

The function of the handshake is to enable frame-by-frame data transmission by using the CTS (Clear to send) pin and to prevent overrun errors. This function can be enabled or disabled by SCxMOD0<CTSE>.

When the $\overline{\text{CTS}}$ pin is set to "High" level, the current data transmission can be completed but the next data transmission is suspended until the $\overline{\text{CTS}}$ pin returns to the "Low" level. However in this case, the INTTXx interrupt is generated in the normal timing, the next transmit data is written in the transmit buffer, and it waits until it is ready to transmit data.

Note: **(1) If the $\overline{\text{CTS}}$ signal is set to "H" during transmission, the next data transmission is suspended after the current transmission is completed.**

(2) Data transmission starts on the first falling edge of the TXDCLK clock after $\overline{\text{CTS}}$ is set to "L".

Although no $\overline{\text{RTS}}$ pin is provided, a handshake control function can easily implemented by assigning one bit of the port for the $\overline{\text{RTS}}$ function. By setting the port to "High" level upon completion of data reception (in the receive interrupt routine), the transmit side can be requested to suspend data transmission.

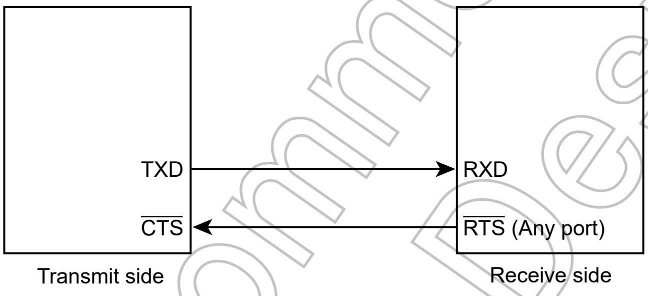


Figure 16-8 Handshake Function

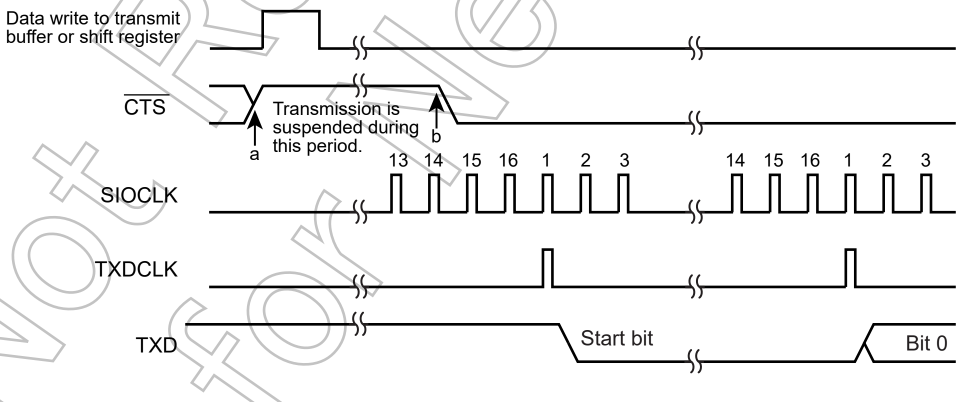
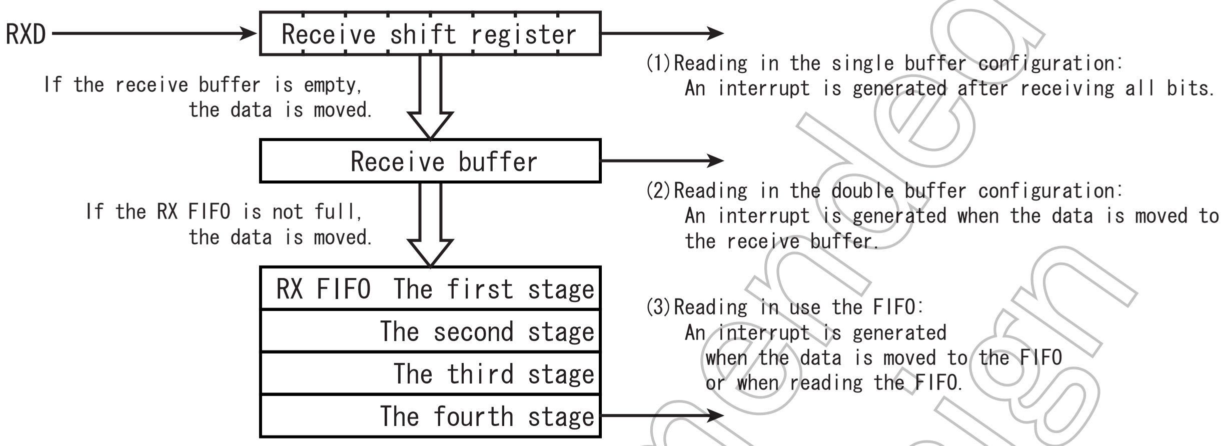

Data write to transmit buffer or shift register
CTS
Transmission is suspended during this period.
a
b
13 14 15 16 1 2 3
14 15 16 1 2 3
SIOCLK
TXDCLK
TXD
Start bit
Bit 0

Figure 16-9 $\overline{\text{CTS}}$ Signal timing

16.14 Interrupt/Error Generation Timing

16.14.1 RX Interrupts

Figure 16-10 shows the data flow of receive operation and the route of read.



(1)Reading in the single buffer configuration:
An interrupt is generated after receiving all bits.

(2)Reading in the double buffer configuration:
An interrupt is generated when the data is moved to the receive buffer.

(3)Reading in use the FIFO:
An interrupt is generated when the data is moved to the FIFO or when reading the FIFO.

Figure 16-10 Receive Buffer/FIFO Configuration Diagram

16.14.1.1 Single Buffer / Double Buffer

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

Buffer Configurations	UART modes	IO interface modes
Single Buffer	–	・Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	・Around the center of the first stop bit	・Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.) ・On data transfer from the shift register to the buffer by reading buffer.

Note: Interrupts are not generated when an overrun error is occurred.

16.14.1.2 FIFO

When the FIFO is used, a receive interrupt occurs depending on the timing described in Table 16-13 and the condition specified with SCxRFC<RFIS>.

Table 16-13 Receive Interrupt conditions in use of FIFO

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

16.14.2 TX interrupts

Figure 16-11 shows the data flow of transmit operation and the route of read.

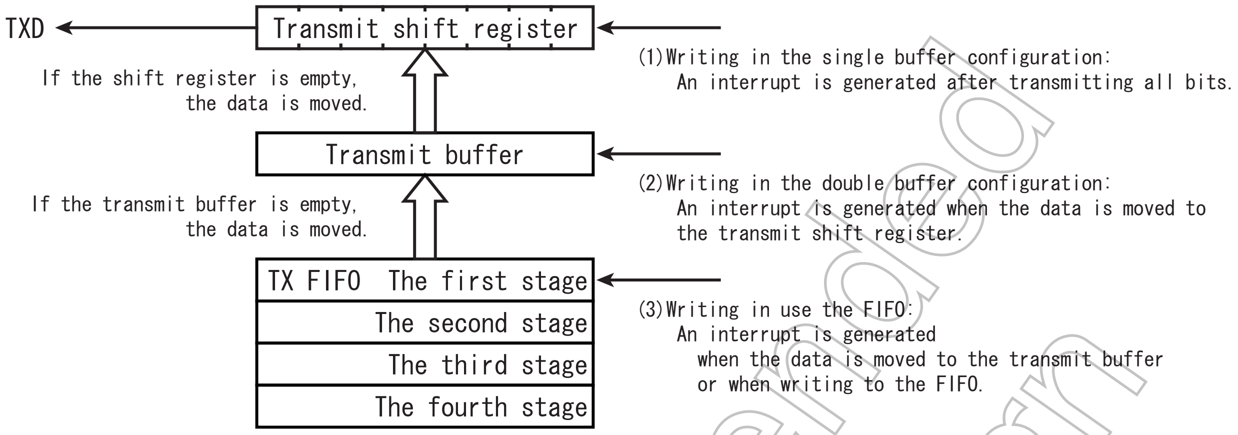

TXD
Transmit shift register
If the shift register is empty, the data is moved.
(1)Writing in the single buffer configuration:
An interrupt is generated after transmitting all bits.
Transmit buffer
If the transmit buffer is empty, the data is moved.
(2)Writing in the double buffer configuration:
An interrupt is generated when the data is moved to the transmit shift register.
TX FIFO The first stage
The second stage
The third stage
The fourth stage
(3)Writing in use the FIFO:
An interrupt is generated when the data is moved to the transmit buffer or when writing to the FIFO.

Figure 16-11 Transmit Buffer/FIFO Configuration Diagram

16.14.2.1 Single Buffer / Double Buffer

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

Buffer Configurations	UART modes	IO interface modes
Single Buffer	Just before the stop bit is sent	Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	When a data is moved from the transmit buffet to the transmit shift register.	

Note: If double buffer is enabled, a interrupt is also generated when the data is moved from the buffer to the shift register by writing to the buffer.

16.14.2.2 FIFO

When the FIFO is used, a transmit interrupt occurs depending on the timing described in Table 16-14 and the condition specified with SCxTFC<TFIS>.

Table 16-14 Transmit Interrupt conditions in use of FIFO

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

16.14.3 Error Generation

16.14.3.1 UART Mode

modes	9 bits	7 bits 8 bits 7 bits+ Parity 8 bits + Parity
Framing Error Overrun Error	Around the center of stop bit	
Parity Error	−	Around the center of parity bit

16.14.3.2 IO Interface Mode

Overrun Error	Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Underrun Error	Immediately after the rising or falling edge of the next SCLK. (Rising or falling is determined according to SCxCR<SCLKS> setting.)

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

16.15 Software Reset

Software reset is generated by writing SCxMOD2<SWRST[1:0]> as "10" followed by "01".

As a result, SCxMOD0<RXE>, SCxMOD1<TXE>, SCxMOD2<TBEMP><RBFLL><TXRUN>, SCxCR

<OERR><PERR><FERR> are initialized. And the receive circuit, the transmit circuit and the FIFO become initial state. Other states are maintained.

16.16 DMA request

DMA request to DMAC is generated at the timing of UART/SIO interrupt request (INTRX0, INTTX0, INTRX2, INTTX2, INTRX4, INTTX4). When DMA transfer is used, set SCxDMA (x=0, 2, 4).

16.17 Operation in Each Mode

16.17.1 Mode 0 (I/O interface mode)

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

The following operational descriptions are for the case use of FIFO is disabled. For details of FIFO operation, refer to the previous sections describing receive/transmit FIFO functions.

16.17.1.1 Transmitting Data

(1) SCLK Output Mode

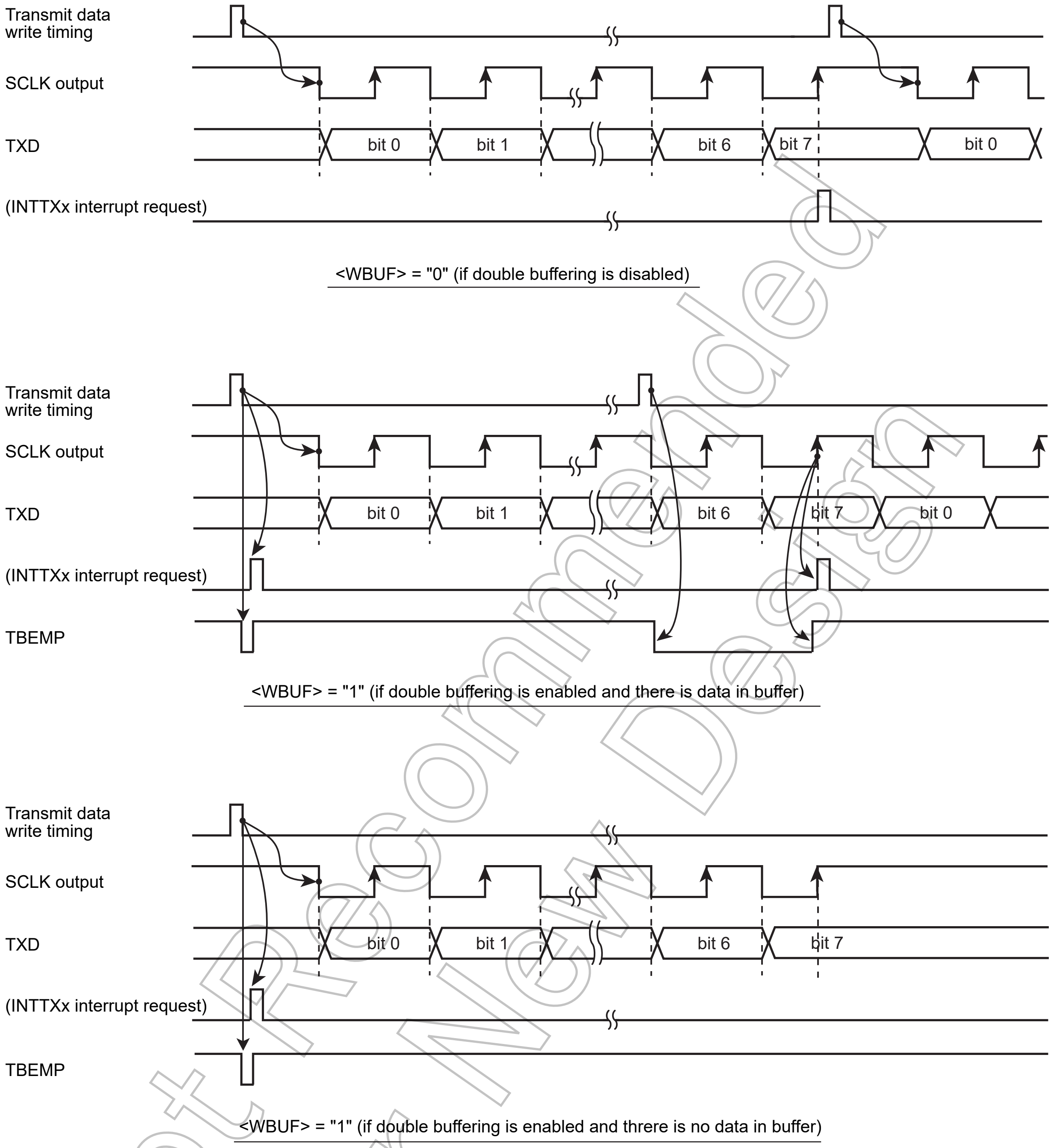
- If the transmit double buffer is disabled (SCxMOD2<WBUF> = "0")

  Data is output from the TXD pin and the clock is output from the SCLK pin each time the CPU writes data to the transmit buffer. When all data is output, an interrupt (INTTXx) is generated.

- If the transmit double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data is moved from the transmit buffer to the transmit shift register when the CPU writes data to the transmit buffer while data transmission is halted or when data transmission from the transmit buffer (shift register) is completed. Simultaneously, the transmit buffer empty flag SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

  When data is moved from the transmit buffer to the transmit shift register, if the transmit buffer has no data to be moved to the transmit shift register, INTTXx interrupt is not generated and the SCLK output stops.


Transmit data write timing
SCLK output
TXD
bit 0
bit 1
bit 6
bit 7
bit 0
(INTTXx interrupt request)
Transmit data write timing
SCLK output
TXD
bit 0
bit 1
bit 6
bit 7
bit 0
(INTTXx interrupt request)
TBEMP
Transmit data write timing
SCLK output
TXD
bit 0
bit 1
bit 6
bit 7
(INTTXx interrupt request)
TBEMP

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

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

<WBUF> = "1" (if double buffering is enabled and threre is no data in buffer)

Figure 16-12 Transmit Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

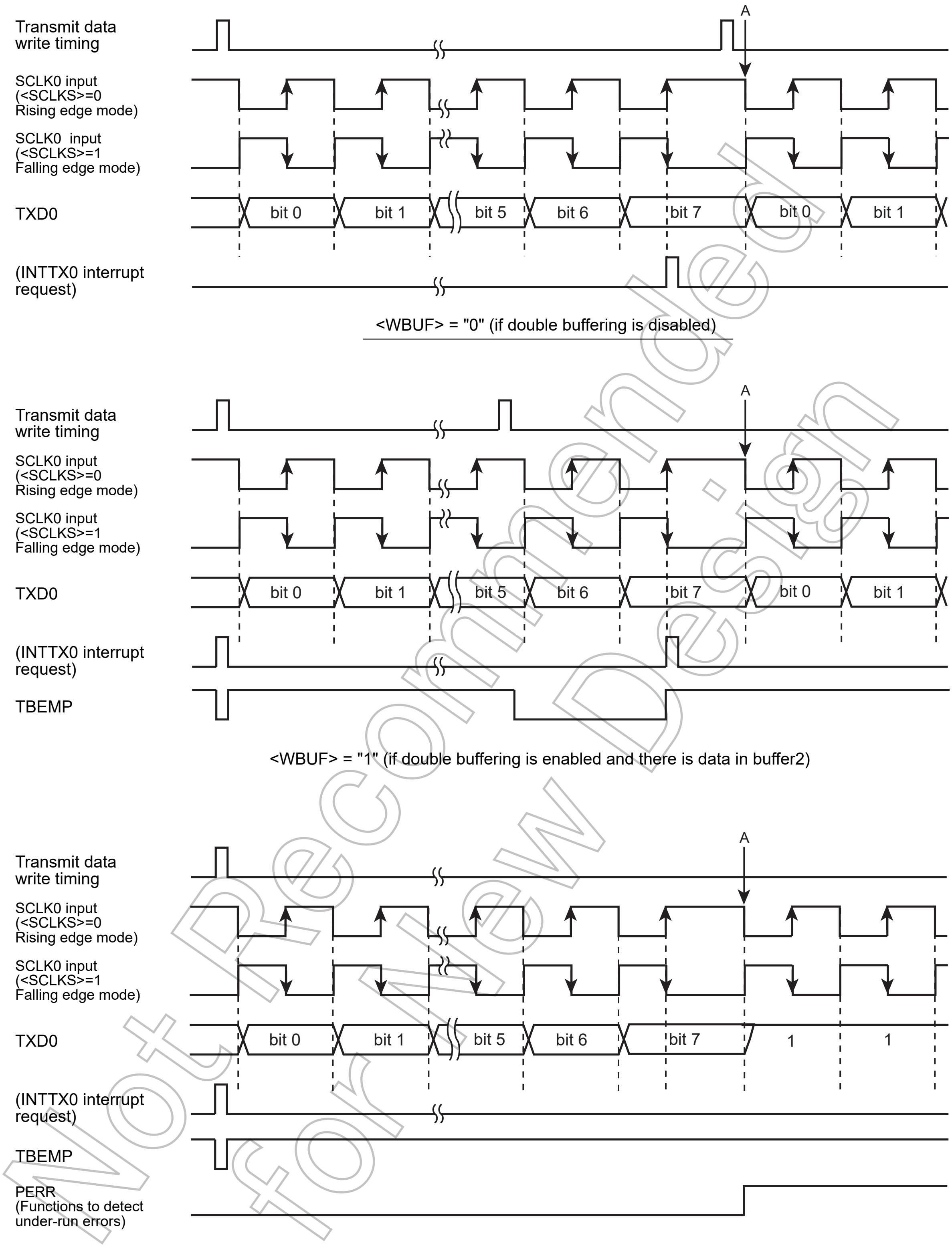
- If double buffering is disabled (SCxMOD2<WBUF> = "0")

  If the SCLK is input in the condition where data is written in the transmit buffer, 8-bit data is outputted from the TXD pin. When all data is output, an interrupt INTTXx is generated. The next transmit data must be written before the timing point "A" as shown in Figure 16-13.

- If double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data is moved from the transmit buffer to the transmit shift register when the CPU writes data to the transmit buffer before the SCLK input becomes active or when data transmission from the transmit shift register is completed. Simultaneously, the transmit buffer empty flag SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

  If the SCLK input becomes active while no data is in the transmit buffer, although the internal bit counter is started, an under-run error occurs and 8-bit dummy data (0xFF) is sent.


A
Transmit data write timing
SCLK0 input (<SCLKS>=0 Rising edge mode)
SCLK0 input (<SCLKS>=1 Falling edge mode)
TXD0
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
(INTTX0 interrupt request)
TBEMP
PERR (Functions to detect under-run errors)
1
1

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

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

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

Figure 16-13 Transmit Operation in the I/O Interface Mode (SCLK Input Mode)

16.17.1.2 Receive

(1) SCLK Output Mode

The SCLK output can be started by setting the receive enable bit SCxMOD0<RXE> to "1".

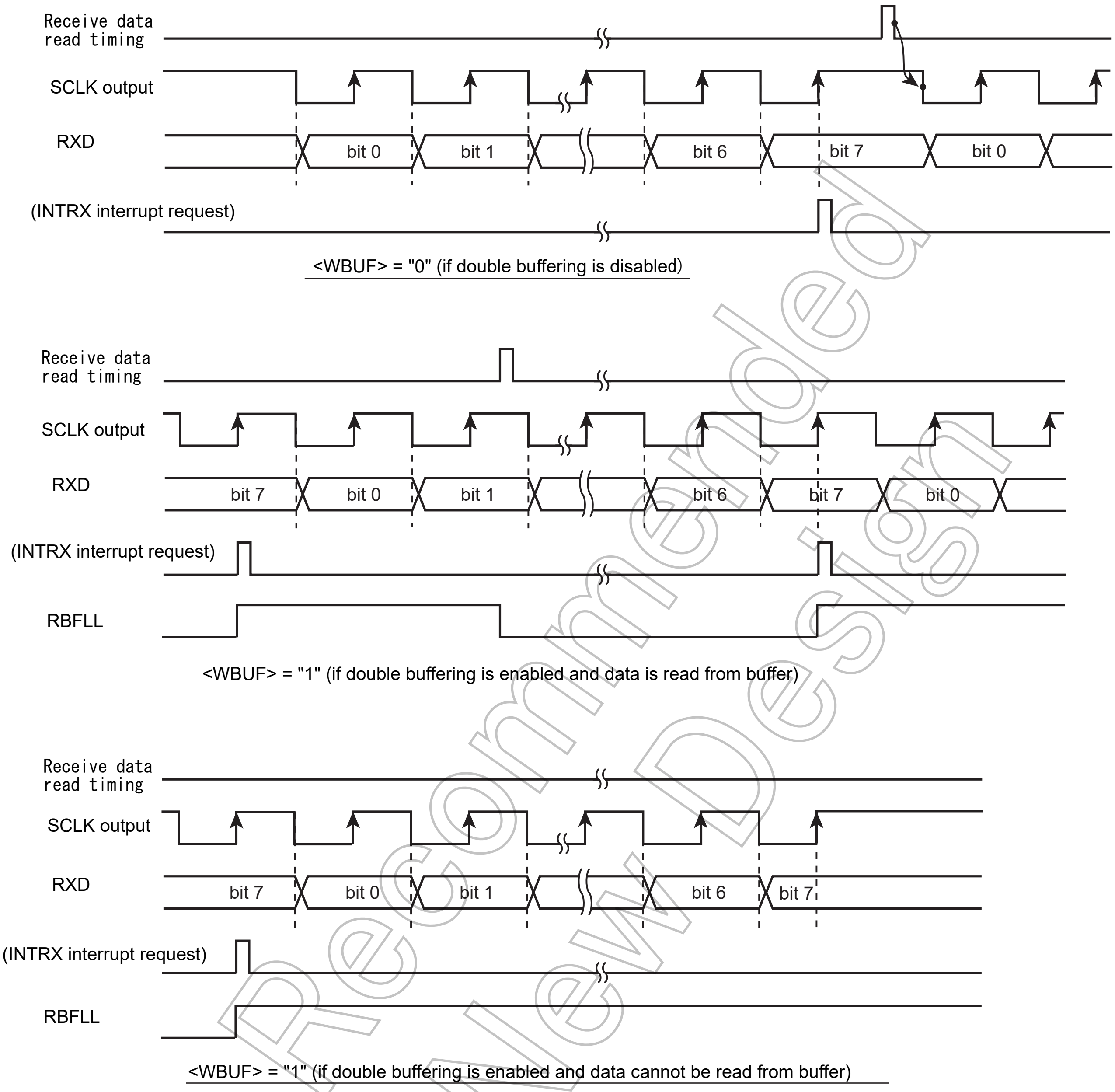
- If double buffer is disabled (SCxMOD2<WBUF> = "0")

  A clock pulse is outputted from the SCLK pin and the next data is stored into the shift register each time the CPU reads received data. When all the 8 bits are received, the INTRXx interrupt is generated.

- If double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data stored in the shift register is moved to the receive buffer and the receive buffer can receive the next frame. A data is moved from the shift register to the receive buffer, the receive buffer full flag SCxMOD2<RBFLL> is set to "1" and the INTRXx is generated.

  While data is in the receive buffer, if the data cannot be read from the receive buffer before completing reception of the next 8 bits, the INTRXx interrupt is not generated and the SCLK output stops. In this state, reading data from the receive buffer allows data in the shift register to move to the receive buffer and thus the INTRXx interrupt is generated and data reception resumes.



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

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

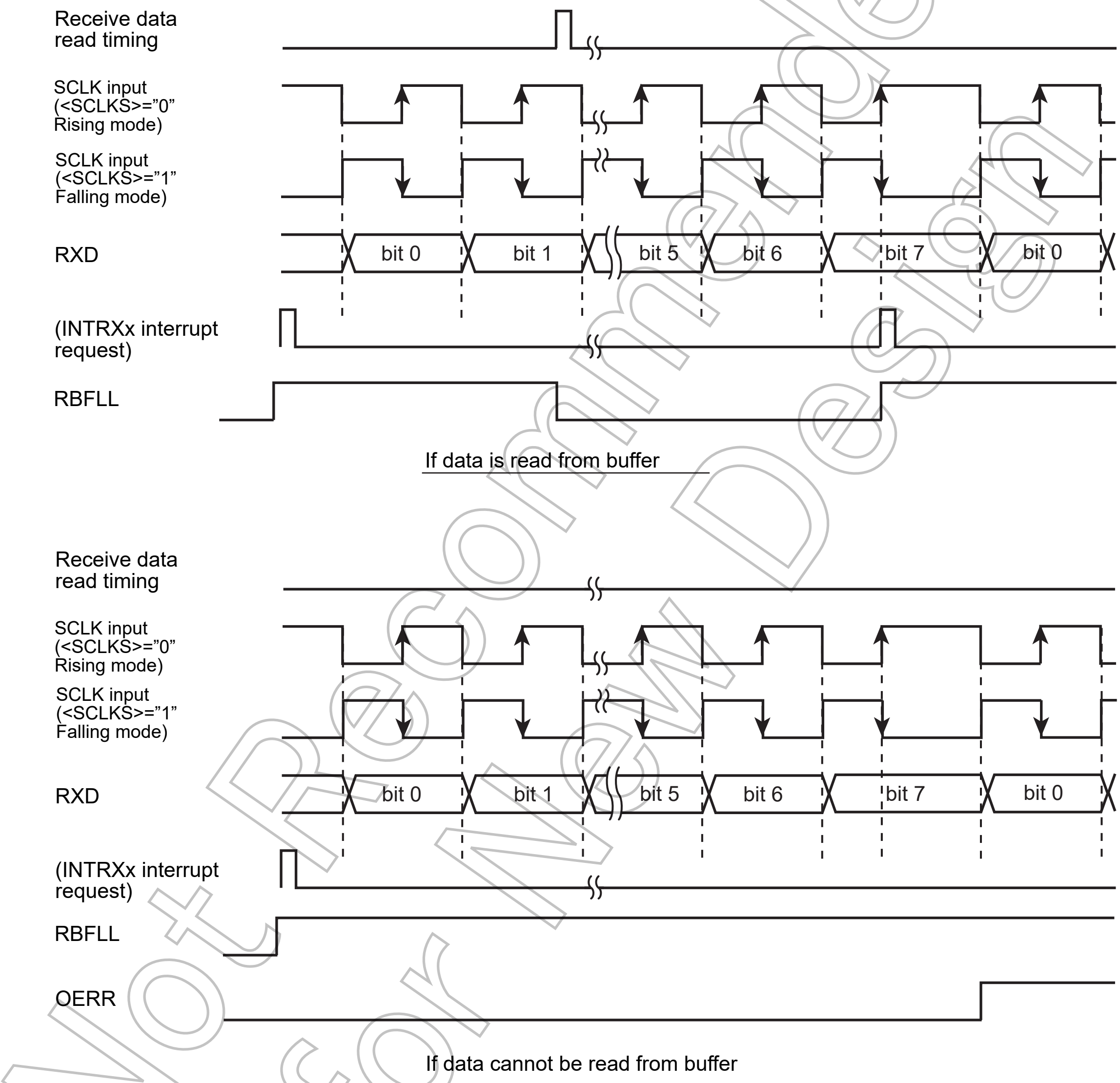
<WBUF> = "1" (if double buffering is enabled and data cannot be read from buffer)

Figure 16-14 Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

In the SCLK input mode, receiving double buffering is always enabled, the received frame can be moved to the receive buffer from the shift register, and the receive buffer can receive the next frame successively.

The INTRx receive interrupt is generated each time received data is moved to the receive buffer.


Receive data read timing
SCLK input (<SCLKS>="0" Rising mode)
SCLK input (<SCLKS>="1" Falling mode)
RXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
(INTRXx interrupt request)
RBFLL
OERR

If data is read from buffer

If data cannot be read from buffer

Figure 16-15 Receive Operation in the I/O Interface Mode (SCLK Input Mode)

16.17.1.3 Transmit and Receive (Full-duplex)

(1) SCLK Output Mode

- If SCxMOD2<WBUF> is set to "0" and the double buffers are disabled

  SCLK is outputted when the CPU writes data to the transmit buffer.

  Subsequently, 8 bits of data are shifted into receive buffer and the INTRXx receive interrupt is generated. Concurrently, 8 bits of data written to the transmit buffer are outputted from the TXD pin, the INTTXx transmit interrupt is generated when transmission of all data bits has been completed. Then, the SCLK output stops.

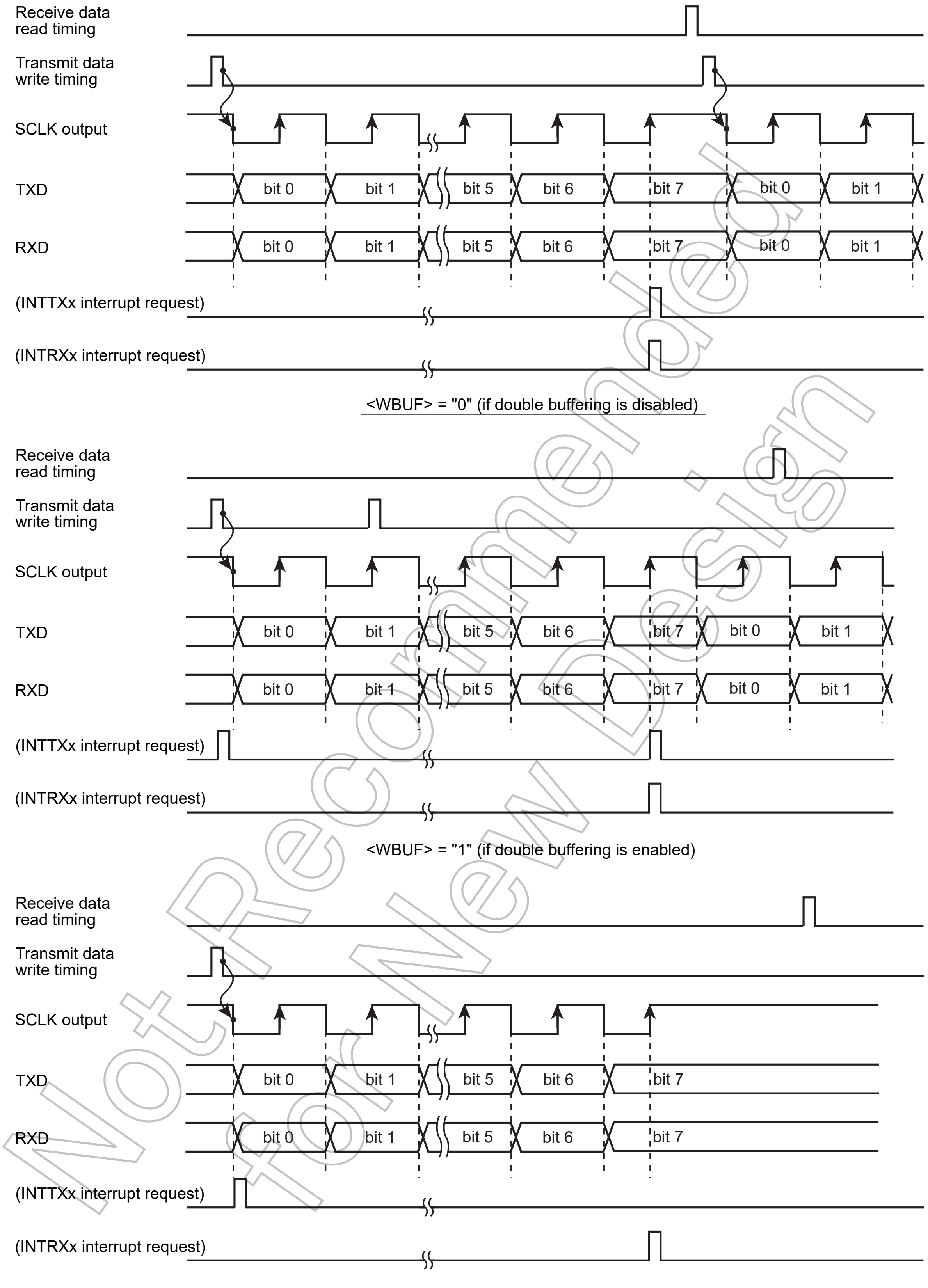
  The next round of data transmission and reception starts when the data is read from the receive buffer and the next transmit data is written to the transmit buffer by the CPU. The order of reading the receive buffer and writing to the transmit buffer can be freely determined. Data transmission is resumed only when both conditions are satisfied.

- If SCxMOD2<WBUF> is set to "1" and the double buffers are enabled

  SCLK is outputted when the CPU writes data to the transmit buffer.

  8 bits of data are shifted into the receive shift register, moved to the receive buffer, and the INTRXx interrupt is generated. While 8 bits of data is received, 8 bits of transmit data is outputted from the TXD pin. When all data bits are sent out, the INTTXx interrupt is generated and the next data is moved from the transmit buffer to the transmit shift register.

  If the transmit buffer has no data to be moved to the transmit buffer (SCxMOD2<TBEMP> = 1) or when the receive buffer is full (SCxMOD2<RBFULL> = 1), the SCLK output is stopped. When both conditions, receive data is read and transmit data is written, are satisfied, the SCLK output is resumed and the next round of data transmission and reception is started.


Receive data read timing
Transmit data write timing
SCLK output
TXD
RXD
(INTTXx interrupt request)
(INTRXx interrupt request)
bit 0
bit 1
bit 5
bit 6
bit 7

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

<WBUF> = "1" (if double buffering is enabled)

<WBUF> = "1" (if double buffering is enabled)

Figure 16-16 Transmit/Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

- If SCxMOD2<WBUF> is set to "0" and the transmit double buffer is disabled

  When receiving data, double buffer is always enabled regardless of the SCxMOD2 <WBUF> settings.

  8-bit data written in the transmit buffer is outputted from the TXD pin and 8 bit of data is shifted into the receive buffer when the SCLK input becomes active.The INTTXx interrupt is generated upon completion of data transmission. The INTTRXx interrupt is generated when the data is moved from shift register to receive buffer after completion of data reception.

  Note that transmit data must be written into the transmit buffer before the SCLK input for the next frame (data must be written before the point A in Figure 16-17). Data must be read before completing reception of the next frame data.

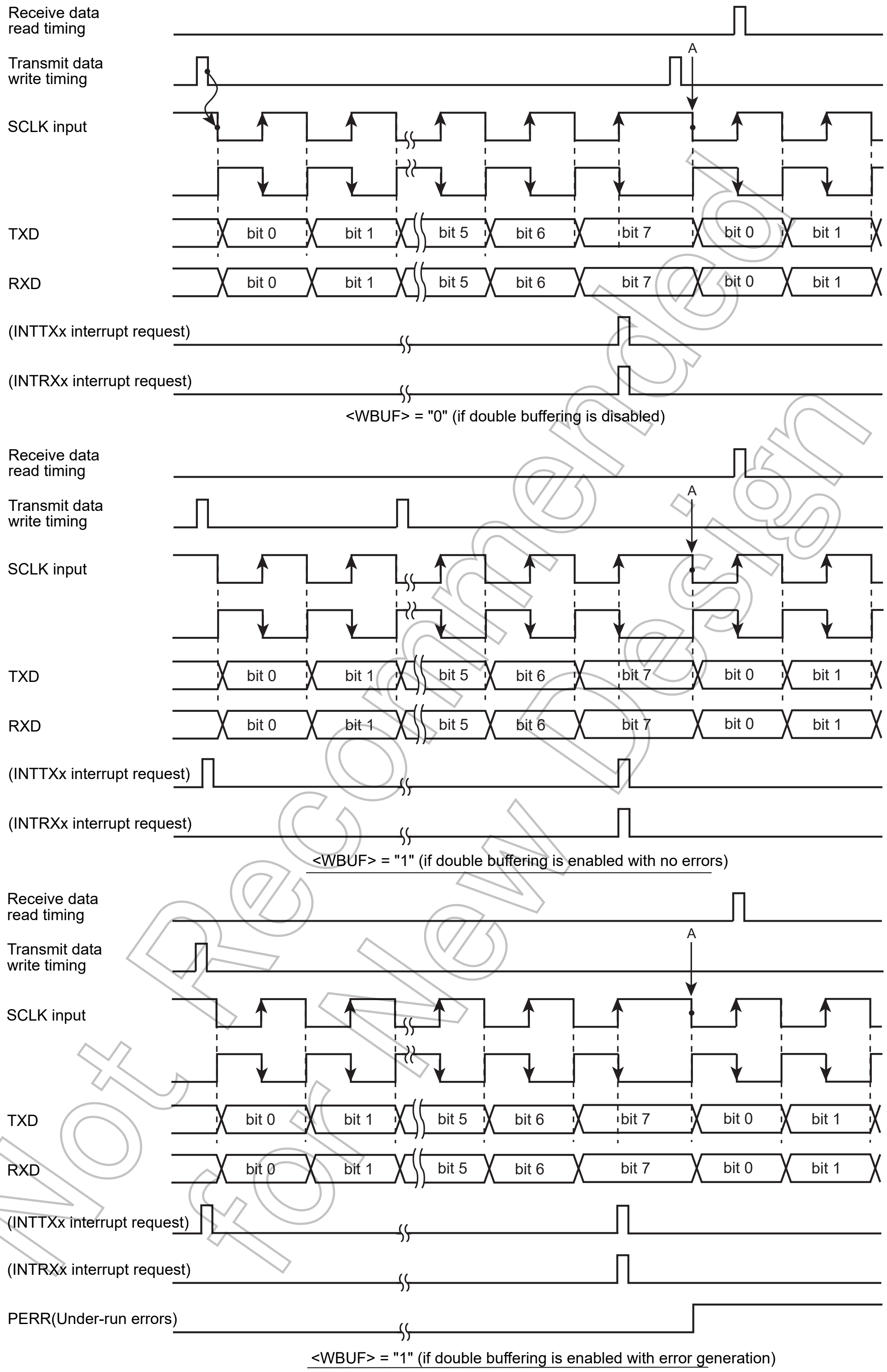
- If SCxMOD2<WBUF> is set to "1" and the double buffer is enabled.

  The interrupt INTRXx is generated at the timing the transmit buffer data is moved to the transmit shift register after completing data transmission from the transmit shift register. At the same time, data received is shifted to the shift register, it is moved to the receive buffer, and the INTRXx interrupt is generated.

  Note that transmit data must be written into the transmit buffer before the SCLK input for the next frame (data must be written before the point A in Figure 16-17). Data must be read before completing reception of the next frame data.

  Upon the SCLK input for the next frame, transmission from transmit shift register (in which data has been moved from transmit buffer) is started while receive data is shifted into receive shift register simultaneously.

  If data in receive buffer has not been read when the last bit of the frame is received, an overrun error occurs. Similarly, if there is no data written to transmit buffer when SCLK for the next frame is input, an under-run error occurs.



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

<WBUF> = "1" (if double buffering is enabled with no errors)

<WBUF> = "1" (if double buffering is enabled with error generation)

Figure 16-17 Transmit/Receive Operation in the I/O Interface Mode (SCLK Input Mode)

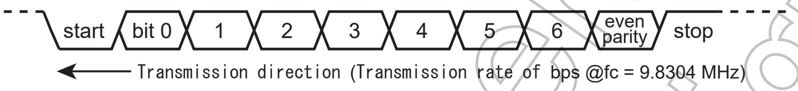
16.17.2 Mode 1 (7-bit UART mode)

The 7-bit UART mode can be selected by setting the serial mode control register (SCxMOD<SM[1:0]>) to "01".

In this mode, parity bits can be added to the transmit data stream; the serial mode control register (SCxCR<PE>) controls the parity enable/disable setting.

When <PE> is set to "1" (enable), either even or odd parity may be selected using the SCxCR<EVEN> bit. The length of the stop bit can be specified using SCxMOD2<SBLEN>.

The following table shows the control register settings for transmitting in the following data format.


start
bit 0
1
2
3
4
5
6
even parity
stop
Transmission direction (Transmission rate of bps @fc = 9.8304 MHz)

Clocking conditions	system clock :	High-speed (fc)
	High-speed clock gear:	X1 (fc)
	Prescaler clock:	fperiph/2 (fperiph = fsys)

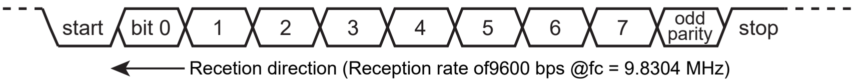
		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	-	0	0	1	0	1	Set 7-bit UART mode
SCxCR	←	x	1	1	x	x	x	0	0	Even parity enabled
SCxBRCR	←	0	0	1	0	0	1	0	0	Set 2400bps
SCxBUF	←	*	*	*	*	*	*	*	*	Set transmit data

x : don't care - : no change

16.17.3 Mode 2 (8-bit UART mode)

The 8-bit UART mode can be selected by setting SCxMOD0<SM[1:0]> to "10." In this mode, parity bits can be added and parity enable/disable is controlled using SCxCR<PE>. If <PE> = "1" (enabled), either even or odd parity can be selected using SCxCR<EVEN>.

The control register settings for receiving data in the following format are as follows:


start
bit 0
1
2
3
4
5
6
7
odd parity
stop
Recetion direction (Reception rate of9600 bps @fc = 9.8304 MHz)

Clocking conditions	System clock:	High-speed (fc)
	High-speed clock gear:	X1
	Prescaler clock:	fperiph/2 (fperiph = fsys)

		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	0	0	1	0	0	1	SEt 8-bit UART mode
SCxCR	←	x	0	1	x	x	x	0	0	Odd parity enabled
SCxBRCR	←	0	0	0	1	0	1	0	0	Set 9600bps
SCxMOD0	←	-	-	1	-	-	-	-	-	Reception enabled

x : don’t care - : no change

16.17.4 Mode 3 (9-bit UART mode)

The 9-bit UART mode can be selected by setting SCxMOD0<SM[1:0]> to "11." In this mode, parity bits must be disabled (SCxCR<PE> = "0").

The most significant bit (9th bit) is written to bit 7 <TB8> of the serial mode control register 0 (SCxMOD0) for transmitting data. The data is stored in bit 7 <RB8> of the serial control register SCxCR.

When writing or reading data to/from the buffers, the most significant bit must be written or read first before writing or reading to/from SCxBUF.

The stop bit length can be specified using SCxMOD2<SBLEN>.

16.17.4.1 Wake up function

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting the wake-up function control bit SCxMOD0<WU> to "1."

In this case, the interrupt INTRXx will be generated only when SCxCR<RB8> is set to "1".

Note: The TXD pin of the slave controller must be set to the open drain output mode using the ODE register.

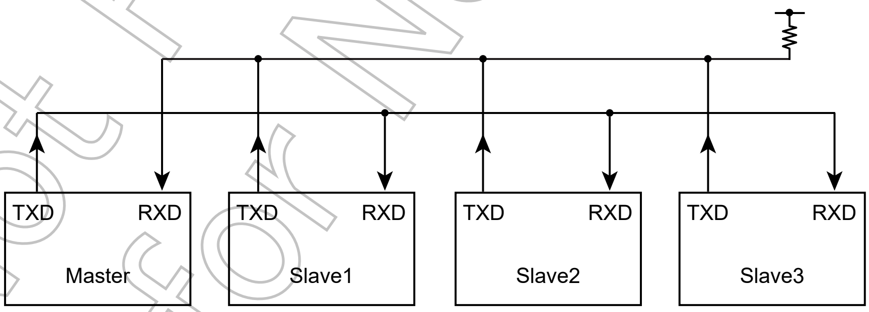


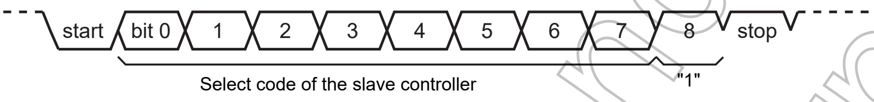
Figure 16-18 Serial Links to Use Wake-up Function

16.17.4.2 Protocol

1. Select the 9-bit UART mode for the master and slave controllers.

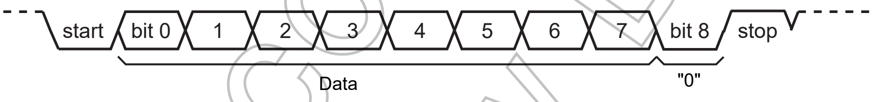
2. Set SCxMOD<WU> to "1" for the slave controllers to make them ready to receive data.

3. The master controller is to transmit a single frame of data that includes the slave controller select code (8 bits). In this, the most significant bit (bit 8) <TB8> must be set to "1".



4. Each slave controller receives the above data frame; if the code received matches with the controller's own select code, it clears the WU bit to "0".

5. The master controller transmits data to the designated slave controller (the controller of which SCxMOD<WU> bit is cleared to "0"). In this, the most significant bit (bit 8) <TB8> must be set to "0".



6. The slave controllers with the <WU> bit set to "1" ignore the receive data because the most significant bit (bit 8) <RB8> is set to "0" and thus no interrupt (INTRXx) is generated.Also, the slave controller with the <WU> bit set to "0" can transmit data to the master controller to inform that the data has been successfully received.

17. Synchronous Serial Port (SSP)

17.1 Overview

This LSI contains the SSP (Synchronous Serial Port) with 1 channel. This channel has the following features.

Communication protocol		Three types of synchronous serial ports including the SPI ・ Motorola SPI (SPI) frame format ・ TI synchronous (SSI) frame format ・ National Microwire (Microwire) frame format
Operation mode		Master/slave mode
Transmit FIFO		16bits wide / 8 tiers deep
Receive FIFO		16bits wide / 8 tiers deep
Transmitted/received data size		4 to 16 bits
Interrupt type		Transmit interrupt Receive interrupt Receive overrun interrupt Time-out interrupt
Communication speed	In master mode	fsys / 2 (max. 20Mbps)
	In slave mode (Note)	fsys / 12 (max. 4.5Mbps)
DMA		Supported
Internal test function		The internal loopback test mode is available.
Control pin		SPCLK,SPFSS,SPDO,SPDI

Note: Set a clock prescaler to SSPCR0<SCR[7:0]> = 0x00 , SSPCPSR<CPSDVSR[7:0]> = 0x02, when slave mode is selected.

17.2 Block Diagram

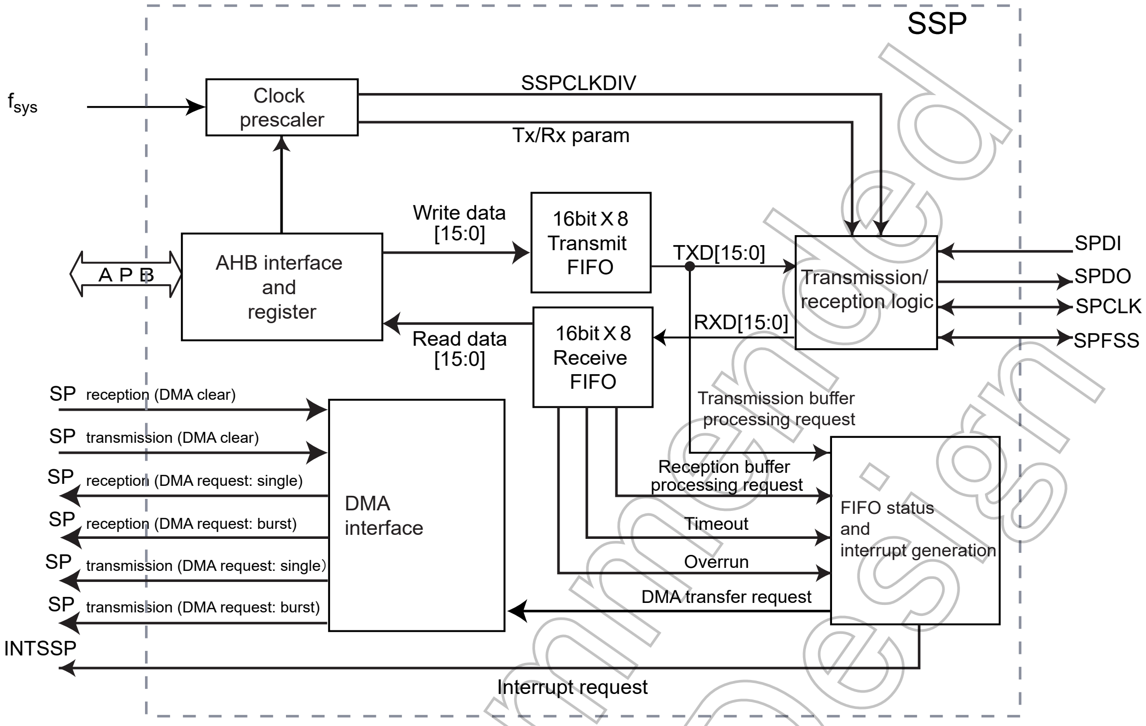

SSP
f_{sys}
Clock prescaler
SSPCLKDIV
Tx/Rx param
APB
AHB interface and register
Write data [15:0]
Read data [15:0]
16bit X 8 Transmit FIFO
16bit X 8 Receive FIFO
TXD[15:0]
RXD[15:0]
Transmission/ reception logic
SPDI
SPDO
SPCLK
SPFSS
Transmission buffer processing request
Reception buffer processing request
Timeout
Overrun
DMA transfer request
FIFO status and interrupt generation
DMA interface
SP reception (DMA clear)
SP transmission (DMA clear)
SP reception (DMA request: single)
SP reception (DMA request: burst)
SP transmission (DMA request: single)
SP transmission (DMA request: burst)
INTSSP
Interrupt request

Figure 17-1 SSP block diagram

17.3 Register

17.3.1 Register List

Base Address = 0x4004_0000

Register Name		Address(Base+)
Control register 0	SSPCR0	0x0000
Control register 1	SSPCR1	0x0004
Receive FIFO (read) and transmit FIFO (write) data register	SSPDR	0x0008
Status register	SSPSR	0x000C
Clock prescale register	SSPCPSR	0x0010
Interrupt enable/disable register	SSPIMSC	0x0014
Pre-enable interrupt status register	SSPRIS	0x0018
Post-enable interrupt status register	SSPMIS	0x001C
Interrupt clear register	SSPICR	0x0020
DMA control register	SSPDMACR	0x0024
Reserved	-	0x0028 to 0x0FFC

Note 1: These registers in the above table allows to access only word (32 bits) basis.

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

17.3.2 SSPCR0(Control register 0)

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Description</th></tr>
<tr><td>31-16</td><td>−</td><td>W</td><td>Write as "0".</td></tr>
<tr><td>15-8</td><td>SCR[7:0]</td><td>R/W</td><td>For serial clock rate setting.
Parameter : 0x00 to 0xFF.

Bits to generate the SSP transmit bit rate and receive bit rate.
This bit rate can be obtained by the following equation.
Bit rate = f_{sys} / (<CPSDVSR> × (1+ <SCR>))
<CPSDVSR> is an even number between 2 to 254, which is programmed by the SSPCPSR register, and <SCR> takes a value between 0 to 255.</td></tr>
<tr><td>7</td><td>SPH</td><td>R/W</td><td>SPCLK phase:
0 : Captures data at the 1st clock edge.
1 : Captures data at the 2nd clock edge.
This is applicable to Motorola SPI frame format only. Refer to Section "Motorola SPI frame format"</td></tr>
<tr><td>6</td><td>SPO</td><td>R/W</td><td>SPCLK polarity:
0:SPCLK is in Low state.
1:SPCLK is in High state.
This is applicable to Motorola SPI frame format only. Refer to Section "Motorola SPI frame format"</td></tr>
<tr><td>5-4</td><td>FRF[1:0]</td><td>R/W</td><td>Frame format:
00: SPI frame format
01: SSI serial frame format
10: Microwire frame format
11: Reserved, undefined operation</td></tr>
<tr><td>3-0</td><td>DSS[3:0]</td><td>R/W</td><td>Data size select:
<table>
<tr><td>0000:</td><td>Reserved, undefined operation</td><td>1000:</td><td>9 bits data</td></tr>
<tr><td>0001:</td><td>Reserved, undefined operation</td><td>1001:</td><td>10 bits data</td></tr>
<tr><td>0010:</td><td>Reserved, undefined operation</td><td>1010:</td><td>11 bits data</td></tr>
<tr><td>0011:</td><td>4 bits data</td><td>1011:</td><td>12 bits data</td></tr>
<tr><td>0100:</td><td>5 bits data</td><td>1100:</td><td>13 bits data</td></tr>
<tr><td>0101:</td><td>6 bits data</td><td>1101:</td><td>14 bits data</td></tr>
<tr><td>0110:</td><td>7 bits data</td><td>1110:</td><td>15 bits data</td></tr>
<tr><td>0111:</td><td>8 bits data</td><td>1111:</td><td>16 bits data</td></tr>
</table>
</td></tr>
</table>

17.3.3 SSPCR1(Control register1)

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

Bit	Bit Symbol	Type	Description
31-4	−	W	Write as "0".
3	SOD	R/W	Slave mode SPDO output control: 0: Enable 1: Disable Slave mode output disable. This bit is relevant only in the slave mode (<MS>="1").
2	MS	R/W	Master/slave mode select: (Note) 0: Device configured as a master. 1: Device configured as a slave.
1	SSE	R/W	SSP enable/disable 0: Disable 1: Enable
0	LBM	R/W	Loop back mode 0: Normal serial port operation enabled. 1: Output of transmit serial shifter is connected to input of receive serial shifter internally.

Note: This bit is for switching between master and slave. Be sure to configure in the following steps in slave mode and in transmission.

1)	Set to slave mode	:<MS>=1
2)	Set transmit data in FIFO	:<DATA>=0x****
3)	Set SSP to Enable.	:<SSE>=1

17.3.4 SSPDR(Data register)

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

Bit	Bit Symbol	Type	Description
31-16	−	W	Write as "0".
15-0	DATA[15:0]	R/W	Transmit/receive FIFO data: 0x0000 to 0xFFFF Read: Receive FIFO Write: Transmit FIFO If the data size used in the program is less than 16bits, write the data to fit LSB.The transmit control circuit ignores unused bits of MSB side. The receive control circuit receives the data to fit LSB automatically.

17.3.5 SSPSR(Status register)

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

Bit	Bit Symbol	Type	Description
31-5	−	W	Write as "0".
4	BSY	R	Busy flag: 0: Idle 1: Busy <BSY>="1" indicates that the SSP is currently transmitting and/or receiving a frame or the transmit FIFO is not empty.
3	RFF	R	Receive FIFO full flag: 0: Receive FIFO is not full. 1: Receive FIFO is full.
2	RNE	R	Receive FIFO empty flag: 0: Receive FIFO is empty. 1: Receive FIFO is not empty.
1	TNF	R	Transmit FIFO full flag: 0: Transmit FIFO is full. 1: Transmit FIFO is not full.
0	TFE	R	Transmit FIFO empty flag: 0: Transmit FIFO is not empty. 1: Transmit FIFO is empty.

17.3.6 SSPCPSR (Clock prescale register)

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

Bit	Bit Symbol	Type	Description
31-8	−	W	Write as "0".
7-0	CPSDVSR[7:0]	R/W	Clock prescale divisor: Set an even number from 2 to 254. Clock prescale divisor: Must be an even number from 2 to 254, depending on the frequency of fsys. The least significant bit always returns zero when read.

17.3.7 SSPIMSC (Interrupt enable/disable register)

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

Bit	Bit Symbol	Type	Description
31-4	−	W	Write as "0".
3	TXIM	R/W	Transmit FIFO interrupt enable: 0: Disable 1: Enable Enable or disable a conditional interrupt to occur if the transmit FIFO is half empty or less.
2	RXIM	R/W	Receive FIFO interrupt enable: 0: Disable 1: Enable Enable or disable a conditional interrupt to occur if the receive FIFO is half full or less.
1	RTIM	R/W	Receive time-out interrupt enable: 0: Disable 1: Enable Enable or disable a conditional interrupt to indicate that data exists in the receive FIFO to the time-out period and data is not read.
0	RORIM	R/W	Receive overrun interrupt enable: 0: Disable 1: Enable Enable or disable a conditioal interrupt to indicate that data was written when the receive FIFO was in the full condition.

17.3.8 SSPRIS (Pre-enable interrupt status register)

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

Bit	Bit Symbol	Type	Description
31-4	−	W	Write as "0".
3	TXRIS	R	Pre-enable transmit interrupt flag: 0: Interrupt not present 1: Interrupt present
2	RXRIS	R	Pre-enable receive interrupt flag: 0: Interrupt not present 1: Interrupt present
1	RTRIS	R	Pre-enable timeout interrupt flag: 0: Interrupt not present 1: Interrupt present
0	RORRIS	R	Pre-enable overrun interrupt flag: 0: Interrupt not present 1: Interrupt present

17.3.9 SSPMIS (Post-enable interrupt status register)

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

Bit	Bit Symbol	Type	Description
31-4	−	W	Write as "0".
3	TXMIS	R	Post-enable transmit interrupt flag: 0: Interrupt not present 1: Interrupt present
2	RXMIS	R	Post-enable receive interrupt flag: 0: Interrupt not present 1: Interrupt present
1	RTMIS	R	Post-enable time-out interrupt flag: 0: Interrupt not present 1: Interrupt present
0	RORMIS	R	Post-enable overrun interrupt flag: 0: Interrupt not present 1: Interrupt present

17.3.10 SSPICR (Interrupt clear register)

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

Bit	Bit Symbol	Type	Description
31-2	−	W	Write as "0".
1	RTIC	W	Clear the time-out interrupt flag: 0: Invalid 1: Clear
0	RORIC	W	Clear the overrun interrupt flag: 0: Invalid 1: Clear

17.3.11 SSPDMACR (DMA control register)

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

Bit	Bit Symbol	Type	Description
31-2	−	W	Write as "0".
1	TXDMAE	R/W	Transmit FIFO DMA control: 0:Disable 1:Enable
0	RXDMAE	R/W	Transmit FIFO DMA control: 0:Disable 1:Enable

17.4 Overview of SSP

This LSI contains the SSP with 1channels.

The SSP is an interface that enables serial communications with the peripheral devices with three types of synchronous serial interface functions.

The SSP performs serial-parallel conversion of the data received from a peripheral device.

The transmit buffers data in the independent 16-bit wide and 8-layered transmit FIFO in the transmit mode, and the receive buffers data in the 16-bit wide and 8-layered receive FIFO in receive mode. Serial data is transmitted via SPDO and received via SPDI.

The SSP contains a programmable prescaler to generate the serial output clock SPCLK from the input clock fsys . The operation mode, frame format, and data size of the SSP are programmed in the control registers SSPCR0 and SSPCR1.

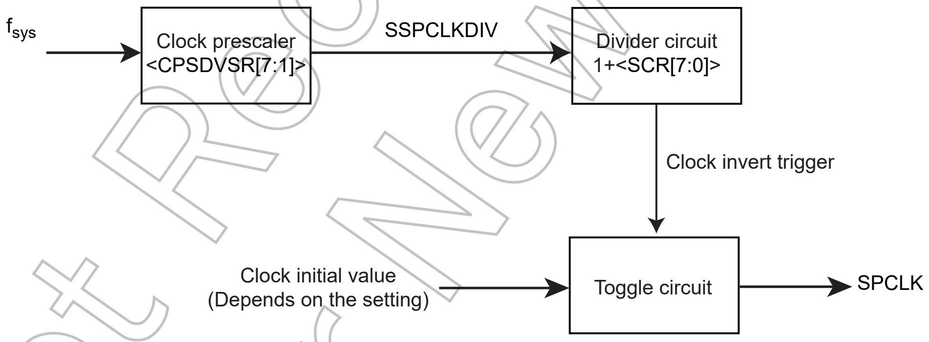
17.4.1 Clock prescaler

When configured as a master, a clock prescaler comprising two free-running serially linked counters is used to provide the serial output clock SPCLK.

You can program the clock prescaler through the SSPCPSR register, to divide fsys by a factor of 2 to 254 in steps of two. Because the least significant bit of the SSPCPSR register is not used, division by an odd number is not possible.

The output of the prescaler is further divided by a factor of 1 to 256, which is obtained by adding 1 to the value programmed in the SSPCR0 register, to give the master output clock SPCLK.

Bitrate = fsys / (<CPSDVSR>×(1+<SCR>))



17.4.2 Transmit FIFO

This is a 16-bit wide, 8-layered transmit FIFO buffer, which is shared in master and slave modes.

17.4.3 Receive FIFO

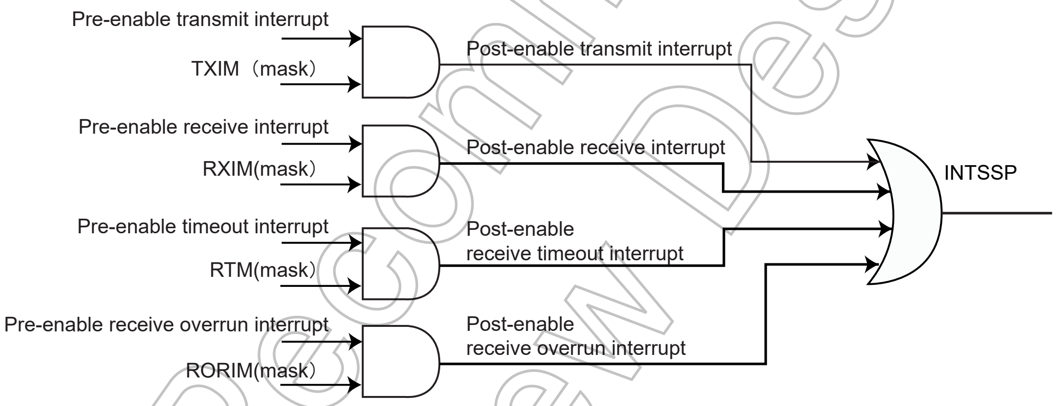
This is a 16-bit wide 8-layered receive FIFO buffer, which is shared in master and slave modes.

17.4.4 Interrupt generation logic

The High active interrupts, each of which can be masked separately, are generated.

Transmit interrupt	A conditional interrupt to occur when the transmit FIFO has free space more than (including half) of the entire capacity. (Number of valid data items in the transmit FIFO ≤ 4)
Receive interrupt	A conditional interrupt to occur when the receive FIFO has valid data more than half (including half) the entire capacity. (Number of valid data items in the receive FIFO ≥ 4)
Time-out interrupt	A conditional interrupt to indicate that the data exists in the receive FIFO to the time-out period.
Overrun interrupt	Conditional interrupts indicating that data is written to receive FIFO when it is full.

Also, The individual masked sources are combined into a single interrupt. When any of the above interrupts is asserted, the combined interrupt INTSSP is asserted.


Pre-enable transmit interrupt
TXIM（mask）
Post-enable transmit interrupt
Pre-enable receive interrupt
RXIM(mask）
Post-enable receive interrupt
Pre-enable timeout interrupt
RTM(mask）
Post-enable
receive timeout interrupt
Pre-enable receive overrun interrupt
RORIM(mask）
Post-enable
receive overrun interrupt
INTSSP

a. Transmit interrupt

The transmit interrupt is asserted when there are four or fewer valid entries in the transmit FIFO. The transmit interrupt is also generated when the SSP operation is disabled (SSPCR1 <SSE> = "0").

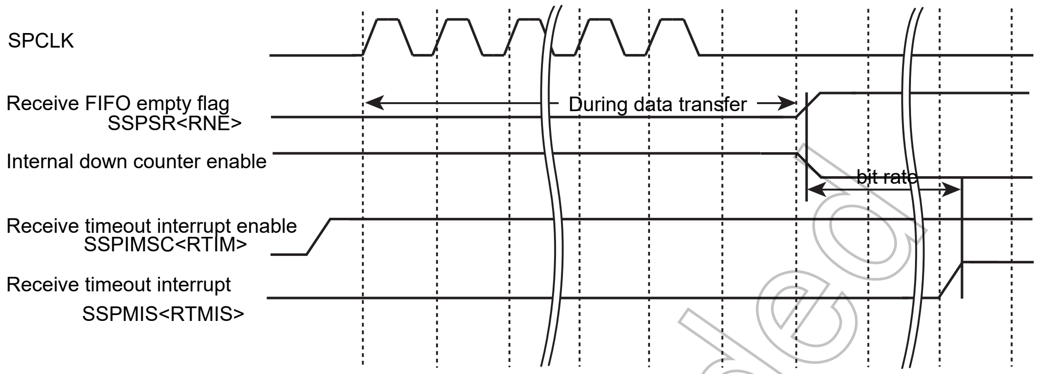
The first transmitted data can be written in the FIFO by using this interrupt.

b. Receive interrupt

The receive interrupt is asserted when there are four or more valid entries in the receive FIFO.

c. Time-out interrupt

The time-out interrupt is asserted when the receive FIFO is not empty and the SSP has remained idle for a fixed 32-bit period (bit rate). This mechanism ensures that the user is aware that data is still present in the receive FIFO and requires servicing. This operation occurs in both master and slave modes. When the time-out interrupt is generated, read all data from the receive FIFO. Even if all the data is not read, data can be transmitted / received if the receive FIFO has a free space and the number of data to be transmitted does not exceed the free space of the receive FIFO. When transfer starts, the timeout interrupt will be cleared. If data is transmitted / received when the receive FIFO has no free space, the time-out interrupt will not be cleared and an overrun interrupt will be generated.


SPCLK
Receive FIFO empty flag
SSPSR<RNE>
During data transfer
Internal down counter enable
bit rate
Receive timeout interrupt enable
SSPIMSC<RTIM>
Receive timeout interrupt
SSPMIS<RTMIS>

d. Overrun interrupt

When the next data (9th data item) is received when the receive FIFO is already full, an overrun interrupt is generated immediately after transfer. The data received after the overrun interrupt is generated (including the 9th data item) will become invalid and be discarded. However, if data is read from the receive FIFO while the 9th data item is being received (before the interrupt is generated), the 9th received data will be written in the receive FIFO as valid data. To perform transfer properly when the overrun interrupt has been generated, write "1" to SSPICR<RORIC> register, and then read all data from the receive FIFO. Even if all the data is not read, data can be transmitted / received if the receive FIFO has free space and the number of data to be transmitted does not exceed the free space of the receive FIFO. Note that if the receive FIFO is not read (provided that the receive FIFO is not empty) within a certain 32-bit period (bit rate) after the overrun interrupt is cleared, a time-out interrupt will be generated.

17.4.5 DMA interface

The DMA operation of the SSP is controlled through SSPxDMACR register.

When there are more data than the watermark level (half of the FIFO) in the receive FIFO, the receive DMA request is asserted.

When the amount of data left in the transmit FIFO is less than the watermark level (half of the FIFO), the transmit DMA request is asserted.

To clear the transmit/receive DMA request, an input pin for the transmit/receive DMA request clear signals, which are asserted by the DMA controller, is provided.

Set the DMA burst length to four words.

Note: For the remaining three words, the SSP does not assert the burst request.

Each request signal remains asserted until the relevant DMA clear signal is asserted. After the request clear signal is deasserted, a request signal can become active again, depending on the conditions described above. All request signals are deasserted if the SSP is disabled or the DMA enable signal is cleared.

The following table shows the trigger points for DMABREQ, for both the transmit and receive FIFOs.

	Burst length	
Watermark level	Transmit (number of empty locations)	Receive (number of filled locations)
1/2	4	4

17.5 SSP operation

17.5.1 Initial setting for SSP

Settings for the SSP communication protocol must be made with the SSP disabled.

Control registers SSPCR0 and SSPCR1 need to configure this SSP as a master or slave operating under one of the following protocols. In addition, make the settings related to the communication speed in the clock prescale registers SSPCPSR and SSPCR0 <SCR>.

This SSP supports the following protocols:

- SPI
- SSI
- Microwire

17.5.2 Enabling SSP

The transfer operation starts when the operation is enabled with the transmitted data written in the transmit FIFO, or when transmitted data is written in the transmit FIFO with the operation enabled.

However, if the transmit FIFO contains only four or fewer entries when the operation is enabled, a transmit interrupt will be generated. This interrupt can be used to write the initial data.

Note: When the SSP is in the SPI slave mode and the SPFSS pin is not used, be sure to transmit data of one byte or more in the FIFO before enabling the operation. If the operation is enabled with the transmit FIFO empty, the transfer data will not be output correctly.

17.5.3 Clock ratios

When setting a frequency for fsys , the following conditions must be met.

- In master mode

  f_{SPCLK} (maximum) → f_{sys} /2

  f_{SPCLK} (minimum) → f_{sys} /(254×256)

- In slave mode

  f_{SPCLK} (maximum) → f_{sys} /12

  f_{SPCLK} (minimum) → f_{sys} /(254×256)

Note: The maximum baud-rate in the master mode is equal or less than 20Mbps.

17.6 Frame Format

Each frame format is between 4 and 16 bits wide depending on the size of data programmed, and is transmitted starting from the MSB.

- Serial clock (SPCLK)

  Signals remain "Low" in the SSI and Microwire formats and as inactive in the SPI format while the SSP is in the idle state. In addition, data is output at the set bit rate only during data transmission.

- Serial frame (SPFSS)

  In the SPI and Microwire frame formats, signals are set to "Low" active and always asserted to "Low" during frame transmission.

  In the SSI frame format, signals are asserted only during 1 bit rate before each frame transmission. In this frame format, output data is transmitted at the rising edge of SPCLK and the input data is received at its falling edge.

Refer to Section "17.6.1" to "17.6.3" for details of each frame format.

17.6.1 SSI frame format

In this mode, the SSP is in idle state, SPCLK and SPFSS are forcedly set to "Low", and the transmit data line SPDO becomes Hi-Z. When data is written in the transmit FIFO, the master outputs "High" pulses of 1 SPCLK to the SPFSS line. The transmitted data will be transferred from the transmit FIFO to the transmit serial shift register. Data of 4 to 16 bits will be output from the SPDO pin at the next rising edge of SPCLK.

Likewise, the received data will be input starting from the MSB to the SPDI pin at the falling edge of SPCLK. The received data will be transferred from the serial shift register into the receive FIFO at the rising edge of SPCLK after its LSB data is latched.

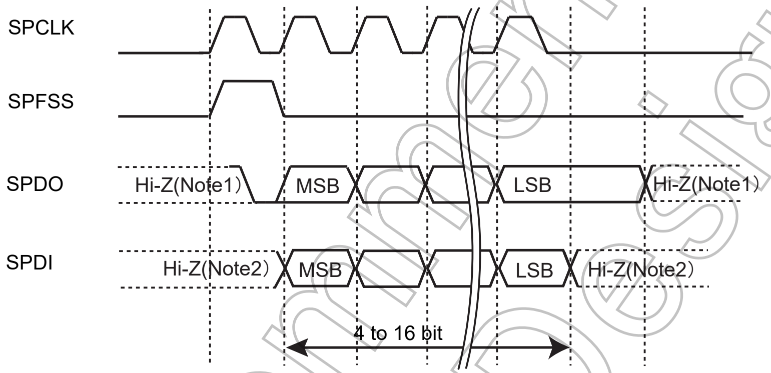


Figure 17-2 SSI frame format (transmission/reception during single transfer)

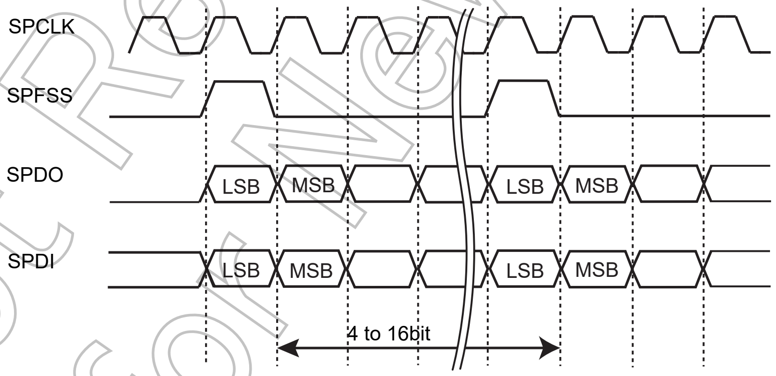


Figure 17-3 SSI frame format (transmission/reception during continuous transfer)

Note 1: When transmission is disable , SPDO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to valid the voltage level.

Note 2: SPDI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to valid the voltage level.

17.6.2 SPI frame format

The SPI interface has 4 lines. SPFSS is used for slave selection. One of the main features of the SPI format is that the <SPO> and <SPH> bits in the SSPCR0 register can be used to set the SPCLK operation timing.

SSPCR0 <SPO> is used to set the level at which SPCLK in idle state is held.

SSPCR0 <SPH> is used to select the clock edge at which data is latched.

	SSPCR0<SPO>	SSPCR0<SPH>
0	"Low" state	Capture data at the 1st clock edge.
1	"High" state	Capture data at the 2nd clock edge.

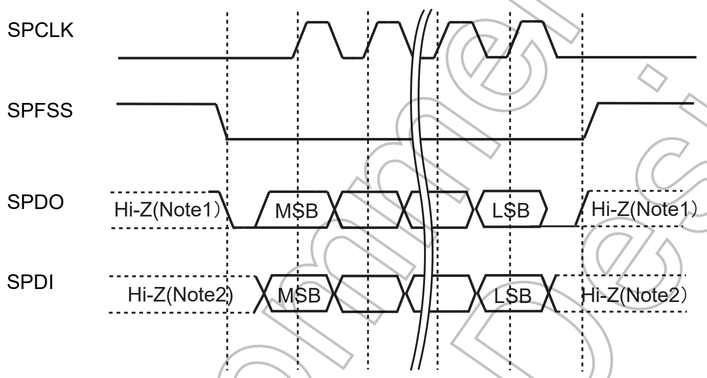


Figure 17-4 SPI frame format (single transfer, <SPO>="0" & <SPH>="0")

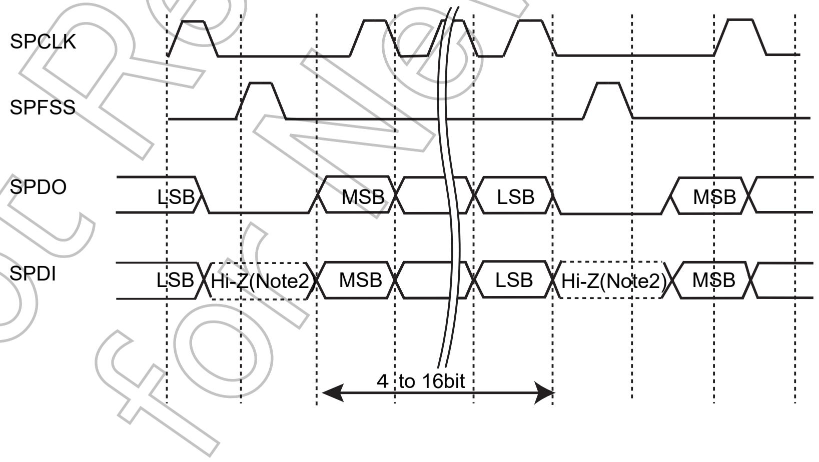


Figure 17-5 SPI frame format (continuous transfer,<SPO>="0" & <SPH>="0")

Note 1: When transmission is disable, SPDO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to valid the voltage level.

Note 2: SPDI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to valid the voltage level.

With this setting <SPO>="0", during the idle period:

- The SPCLK signal is set to "Low".
- SPFSS is set to "High".
- The transmit data line SPDO is set to "Low".

If the SSP is enabled and valid data exists in the transmit FIFO, the SPFSS master signal driven by "Low" notifies of the start of transmission. This enables the slave data in the SPDI input line of the master.

When a half of the SPCLK period has passed, valid master data is transferred to the SPDO pin. Both the master data and slave data are now set. When another half of SPCLK has passed, the SPCLK master clock pin becomes "High". After that, the data is captured at the rising edge of the SPCLK signal and transmitted at its falling edge.

In the single transfer, the SPFSS line will return to the idle "High" state when all the bits of that data word have been transferred, and then one cycle of SPCLK has passed after the last bit was captured.

However, for continuous transfer, the SPFSS signal must be pulsed at HIGH between individual data word transfers. This is because change is not enabled when the slave selection pin freezes data in its peripheral register and the <SPH> bit is logical 0.

Therefore, to enable writing of serial peripheral data, the master device must drive the SPFSS pin of the slave device between individual data transfers. When the continuous transfer is completed, the SPFSS pin will return to the idle state when one cycle of SPCLK has passed after the last bit is captured.

17.6.3 Microwire frame format

The Microwire format uses a special master/slave messaging method, which operates in half-duplex mode. In this mode, when a frame begins, an 8-bit control message is transmitted to the slave. During this transmission, no incoming data is received by the SSP. After the message has been transmitted, the slave decodes it, and after waiting one serial clock after the last bit of the 8-bit control message has been sent, it responds with the requested data. The returned data can be 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits.

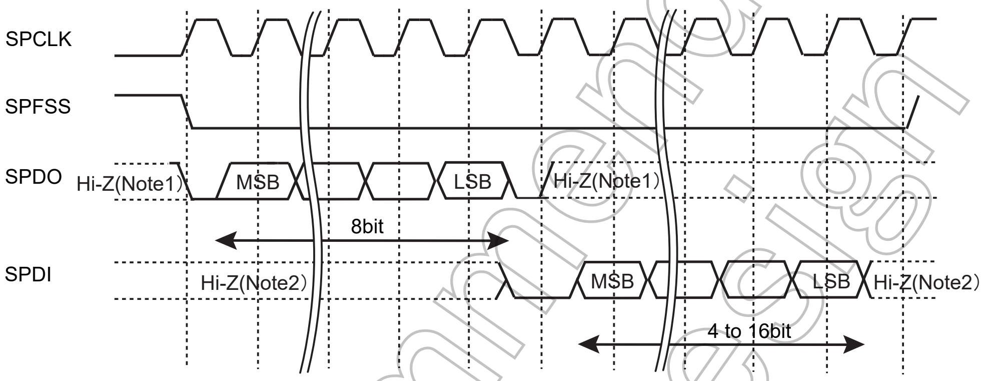


Figure 17-6 Microwire frame format (single transfer)

Note 1: When transmission is disabled, SPDO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to fix the voltage level.

Note 2: SPDI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to fix the voltage level.

Though the Microwire format is similar to the SPI format, it uses the master/slave message transmission method for half-duplex communications. Each serial transmission is started by an 8-bit control word, which is sent to the off-chip slave device. During this transmission, the SSP does not receive input data. After the message has been transmitted, the off-chip slave decodes it, and after waiting one serial clock after the last bit of the 8-bit control message has been sent, responds with the requested data. The returned data can be 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits. With this configuration, during the idle period:

- The SPCLK signal is set to "Low".
- SPFSS is set to "High".
- The transmit data line SPDO is set to "Low".

A transmission is triggered by writing a control byte to the transmit FIFO. The falling edge of SPFSS causes the value stored in the bottom entry of the transmit FIFO to be transferred to the serial shift register for the transmit logic, and the MSB of the 8-bit control frame to be shifted out onto the SPDO pin.

SPFSS remains "Low" and the SPDI pin remains tristated during this transmission. The off-chip serial slave device latches each control bit into its serial shifter on the rising edge of each SPCLK.

After the last bit is latched by the slave device, the control byte is decoded during a one clock wait-state, and the slave responds by transmitting data back to the SSP. Each bit is driven onto SPDI line on the falling edge of SPCLK.

The SSP in turn latches each bit on the rising edge of SPCLK. At the end of the frame, for single transfers, the SPFSS signal is pulled "High" one clock period after the last bit has been latched in the receive serial shifter, which causes the data to be transferred to the receive FIFO.

Note: The off-chip slave device can tristate the receive line either on the falling edge of SPCLK after the LSB has been latched by the receive shifter, or when the SPFSS pin goes "High".

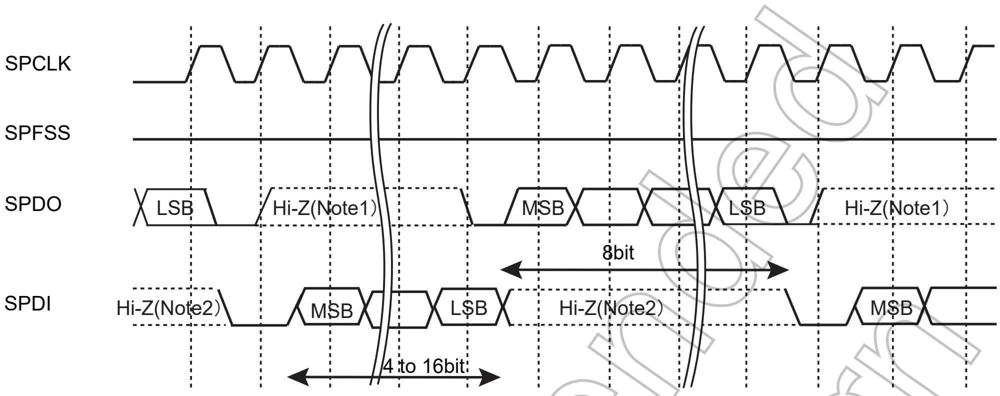


Figure 17-7 Microwire frame format (continuous transfer)

Note 1: When transmission is disabled, SPDO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to fix the voltage level.

Note 2: SPDI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to fix the voltage level.

For continuous transfers, data transmission begins and ends in the same manner as a single transfer. However, the SPFSS line is continuously asserted (held Low) and transmission of data occurs back to back.

The control byte of the next frame follows directly after the LSB of the received data from the current frame. Each of the received values is transferred from the receive shifter on the falling edge of SPCLK, after the LSB of the frame has been latched into the SSP.

Note: [Example of connection] The SSP does not support dynamic switching between the master and slave in the system. Each sample SSP is configured and connected as either a master or slave.

18. Serial Bus Interface (I2C/SIO)

The TMPM341FDXBG/FYXBG contains 2 Serial Bus Interface (I2C/SIO) channels, in which the following two operating modes are included:

- I2C bus mode (with multi-master capability)
- Clock-synchronous 8-bit SIO mode

In the I2C bus mode, the I2C/SIO is connected to external devices via SCL and SDA.

In the clock-synchronous 8-bit SIO mode, the I2C/SIO is connected to external devices via SCK, SI and SO.

The following table shows the programming required to put the I2C/SIO in each operating mode.

Table 18-1 Port settings for using serial bus interface

channel	Operating mode	pin	Port Function Register	Port Output Control Register	Port Input Control Register	Port Open Drain Output Control Register
SBI0	I2C bus mode	SCL0 :PG1 PL1PK3 SDA0 :PG0 PL0PK2	PGFR1[1:0] = 11 PLFR1[1:0] = 11 PKFR3[3:2] = 11	PGCR[1:0] = 11 PLCR[1:0] = 11 PKCR[3:2] = 11	PGIE[1:0] = 11 PLIE[1:0] = 11 PKIE[3:2] = 11	PGOD[1:0] = 11 PLOD[1:0] = 11 PKOD[3:2] = 11
	SIO mode	SCK0 :PG2 PL2PK4 SI0 :PG1PL 1PK3 SO0 :PG0P L0PK2	PGFR1[2:0] = 111 PLFR1[2:0] = 111 PKFR3[4:2] = 111	PGCR[2:0] = 101(SCK0 output) PGCR[2:0] = 001(SCK0 input) PLCR[2:0] = 101(SCK0 output) PLCR[2:0] = 001(SCK0 input) PKCR[4:2] = 101(SCK0 output) PKCR[4:2] = 001(SCK0 input)	PGIE[2:0] = 010(SCK0 output) PGIE[2:0] = 110(SCK0 input) PLIE[2:0] = 010(SCK0 output) PLIE[2:0] = 110(SCK0 input) PKIE[4:2] = 010(SCK0 output) PKIE[4:2] = 110(SCK0 input)	PGOD[2:0] = xxx PLOD[2:0] = xxx PKOD[4:2] = xxx
SBI1	I2C bus mode	SCL1 :PG1 PD1PF6 SDA1 :PG0 PD0PF7	PGFR1[1:0] = 11 PDFR1[1:0] = 11 PFFR4[6:5] = 11	PGCR[1:0] = 11 PDCR[1:0] = 11 PFCR[6:5] = 11	PGIE[1:0] = 11 PDIE[1:0] = 11 PFIE[6:5] = 11	PGOD[1:0] = 11 PDOD[1:0] = 11 PFOD[6:5] = 11
	SIO mode	SCK1 :PG2 PD2PF5 SI1 :PG1PD 1PF6 SO1 :PG0P D0PF7	PGFR1[2:0] = 111 PDFR1[2:0] = 111 PFFR4[7:5] = 111	PGCR[2:0] = 101(SCK1 output) PGCR[2:0] = 001(SCK1 input) PDCR[2:0] = 101(SCK1 output) PDCR[2:0] = 001(SCK1 input) PFCR[7:5] = 101(SCK0 output) PFCR[7:5] = 100(SCK0 input)	PGIE[2:0] = 010(SCK1 output) PGIE[2:0] = 110(SCK1 input) PDIE[2:0] = 010(SCK1 output) PDIE[2:0] = 110(SCK1 input) PFIE[7:5] = 010(SCK0 output) PFIE[7:5] = 011(SCK0 input)	PGOD[2:0] = xxx PDOD[2:0] = xxx PFOD[7:5] = xxx

Note:x: Don't care

18.1 Configuration

The configuration is shown in Figure 18-1.

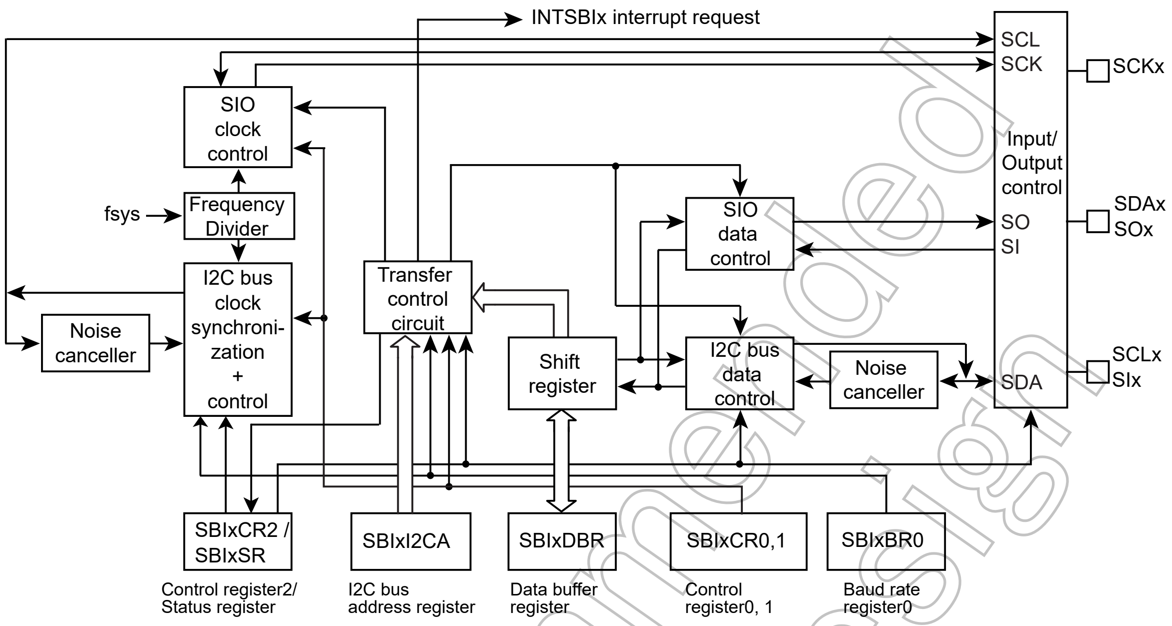


Figure 18-1 (I2C/SIO) Block Interface

18.2 Register

The following registers control the serial bus interface and provide its status information for monitoring.

The register below performs different functions depending on the mode. For details, refer to "18.4 Control Registers in the I2C Bus Mode" and "18.7 Control register of SIO mode".

18.2.1 Registers for each channel

The tables below show the registers and register addresses for each channel.

Channel x	Base Address
Channel0	0x400E_0000
Channel1	0x400E_0100

Register name(x=0,1,)		Address(Base+)
Control register 0	SBIxCR0	0x0000
Control register 1	SBIxCR1	0x0004
Data buffer register	SBIxDBR	0x0008
I2C bus address register	SBIxI2CAR	0x000C
Control register 2	SBIxCR2 (writing)	0x0010
Status register	SBIxSR (reading)	
Baud rate register 0	SBIxBR0	0x0014

18.3 I2C Bus Mode Data Format

Figure 18-2 shows the data formats used in the I2C bus mode.

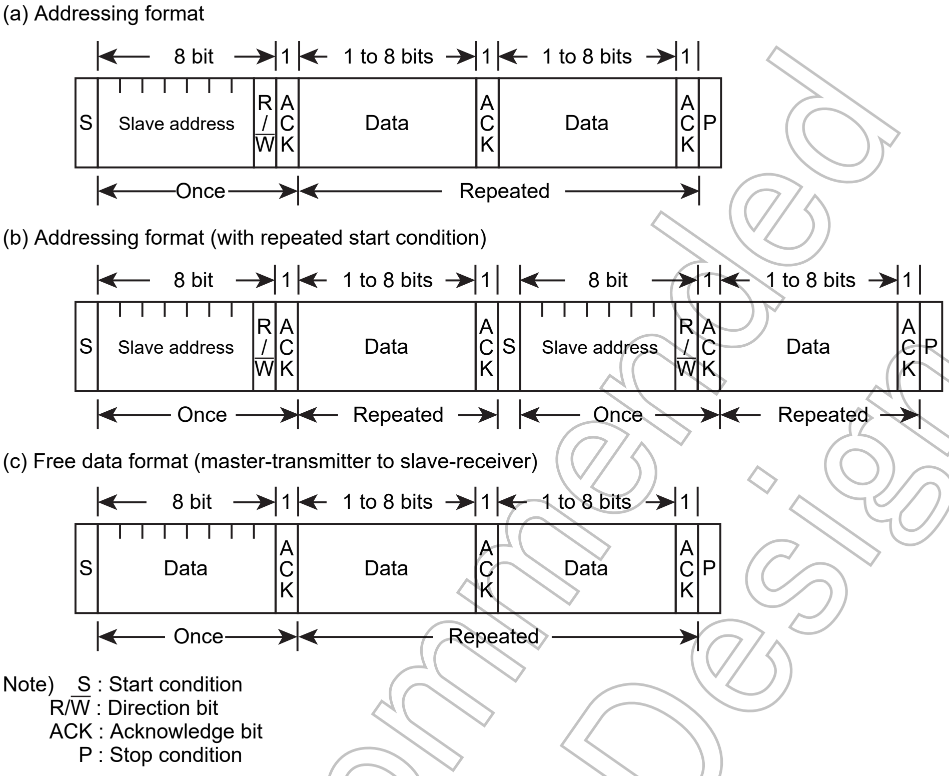


Figure 18-2 I2C Bus Mode Data Formats

18.4 Control Registers in the I2C Bus Mode

The following registers control the serial bus interface in the I2C bus mode and provide its status information for monitoring.

18.4.1 SBIxCR0(Control register 0)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SBIEN	R/W	Serial bus interface operation 0:Disable 1:Enable To use the serial bus interface, enable this bit first. For the first time in case of setting to enable, the relevant SBI registers can be read or written. Since all clocks except SBIxCR0 stop if this bit is disabled, power consumption can be reduced by disabling this bit. If this bit is disabled after it’s been enabled once, the settings of each register are retained.
6-0	-	R	Read as 0.

Note: To use the serial bus interface, enable this bit first.

18.4.2 SBIxCR1(Control register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	BC			ACK	-	SCK2	SCK1	SCK0 / SWRMON
After reset	0	0	0	0	1	0	0	1(Note3)

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-8</td><td>-</td><td>R</td><td>Read as 0.</td></tr>
<tr><td>7-5</td><td>BC[2:0]</td><td>R/W</td><td>Select the number of bits per transfer (Note 1)
<table>
<tr><th rowspan="2">&lt;BC&gt;</th><th colspan="2">When &lt;ACK&gt; = 0</th><th colspan="2">When &lt;ACK&gt; = 1</th></tr>
<tr><th>Number of clock cycles</th><th>Data length</th><th>Number of clock cycles</th><th>Data length</th></tr>
<tr><td>000</td><td>8</td><td>8</td><td>9</td><td>8</td></tr>
<tr><td>001</td><td>1</td><td>1</td><td>2</td><td>1</td></tr>
<tr><td>010</td><td>2</td><td>2</td><td>3</td><td>2</td></tr>
<tr><td>011</td><td>3</td><td>3</td><td>4</td><td>3</td></tr>
<tr><td>100</td><td>4</td><td>4</td><td>5</td><td>4</td></tr>
<tr><td>101</td><td>5</td><td>5</td><td>6</td><td>5</td></tr>
<tr><td>110</td><td>6</td><td>6</td><td>7</td><td>6</td></tr>
<tr><td>111</td><td>7</td><td>7</td><td>8</td><td>7</td></tr>
</table></td></tr>
<tr><td>4</td><td>ACK</td><td>R/W</td><td>Master mode
0: Acknowledgement clock pulse is not generated.
1: Acknowledgement clock pulse is generated.
Slave mode
0: Acknowledgement clock pulse is not counted.
1: Acknowledgement clock pulse is counted.</td></tr>
<tr><td>3</td><td>-</td><td>R</td><td>Read as 1.</td></tr>
<tr><td>2-1</td><td>SCK[2:1]</td><td>R/W</td><td>Select internal SCL output clock frequency (Note 2).</td></tr>
<tr><td rowspan="2">0</td><td>SCK[0]</td><td>W</td><td>
<table>
<tr><td>000</td><td>n = 5</td><td>385 kHz</td><td rowspan="8">System Clock: fsys (≒40MHz)
Clock gear : fc/1
Frequency = $\frac{fsys}{2^n + 72}$ [Hz]</td></tr>
<tr><td>001</td><td>n = 6</td><td>294 kHz</td></tr>
<tr><td>010</td><td>n = 7</td><td>200 kHz</td></tr>
<tr><td>011</td><td>n = 8</td><td>122 kHz</td></tr>
<tr><td>100</td><td>n = 9</td><td>68 kHz</td></tr>
<tr><td>101</td><td>n = 10</td><td>36 kHz</td></tr>
<tr><td>110</td><td>n = 11</td><td>19 kHz</td></tr>
<tr><td>111</td><td></td><td>reserved</td></tr>
</table></td></tr>
<tr><td>SWRMON</td><td>R</td><td>On reading <SWRMON>: Software reset status monitor
0:Software reset operation is in progress.
1:Software reset operation is not in progress.</td></tr>
</table>

Note 1: **Clear <BC[2:0]> to "000" before switching the operation mode to the SIO mode.**

Note 2: **For details on the SCL line clock frequency, refer to "18.5.1 Serial Clock".**

Note 3: **After a reset, the <SCK[0]/SWRMON> bit is read as "1". However, if the SIO mode is selected at the SBIxCR2 register, the initial value of the <SCK[0]> bit is "0".**

Note 4: **The initial value for selecting a frequency is <SCK[2:0]>=000 and is independent of the read initial value.**

Note 5: **When <BC[2:0]>="001" and <ACK>="0" in master mode, SCL line may be fixed to "L" by falling edge of SCL line after generation of STOP condition and the other master devices can not use the bus. In the case of bus which is connected with several master devices, the bumber of bits per transfer should be set equal or more than 2 before generation of STOP condition.**

18.4.3 SBIxCR2(Control register 2)

This register serves as SBIxSR register by reading it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	W	Select master/slave 0: Slave mode 1: Master mode
6	TRX	W	Select transmit/ receive 0: Receive 1: Transmit
5	BB	W	Start/stop condition generation 0: Stop condition generated 1: Start condition generated
4	PIN	W	Clear INTSBIx interrupt request 0: − 1: Clear interrupt request
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note) 00: Port mode (Disables a serial bus interface output) 01: SIO mode 10: I2C bus mode 11: Reserved
1-0	SWRST[1:0]	W	Software reset generation Write "10" followed by "01" to generate a reset. When writing, set <SBIM[1:0]> to "10"; I2Cbus mode.

Note: **Make sure that modes are not changed during a communication session.Ensure that the bus is free before switching the operating mode to the port mode. Ensure that the port is at the "High" level before switching the operating mode from the port mode to the I2C bus or clock-synchronous 8-bit SIO mode.**

18.4.4 SBIxSR (Status Register)

This register serves as SBIxCR2 by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	R	Master/slave selection monitor 0: Slave mode 1: Master mode
6	TRX	R	Transmit/receive selection monitor 0: Receive 1: Transmit
5	BB	R	I2C bus state monitor 0: Free 1: Busy
4	PIN	R	INTSBIx interrupt request monitor 0:Interrupt request generated 1: Interrupt request cleared
3	AL	R	Arbitration lost detection 0: − 1:Detected
2	AAS	R	Slave address match detection 0: − 1: Detected (This bit is set when the general call is detected as well.)
1	ADO	R	General call detection 0: − 1:Detected
0	LRB	R	Last received bit monitor 0:Last received bit "0" 1:Last received bit "1"

18.4.5 SBIxBR0(Serial bus interface baud rate register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
After reset	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	-	R	Read as 1.
6	I2SBI	R/W	Operation at the IDLE mode 0: Stop 1: Operate
5-1	-	R	Read as 1.
0	-	R/W	Be sure to write "0".

18.4.6 SBIxDBR (Serial bus interface data buffer register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DB[7:0]	R (Receive)/ W (Transmit)	Receive data / Transmit data

Note 1: **The transmission data must be written in to the register from the MSB (bit 7). The received data is stored in the LSB.**

Note 2: **Since SBIxBDR has independent buffers for writing and reading, a written data cannot be read. Thus, read-modify-write instructions, such as bit manipulation, cannot be used.**

18.4.7 SBIxI2CAR (I2Cbus address register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-1	SA[6:0]	R/W	Set the slave address when the SBI acts as a slave device.
0	ALS	R/W	Specify address recognition mode. 0: Recognize its slave address. 1: Do not recognize its slave address (free-data format).

Note 1: **Please set the bit 0 <ALS> of I2C bus address register SBIxI2CAR to "0", except when you use a free data format. It operates as a free data format when setting it to "1". Selecting the master fixes to transmission. Selecting the slave fixes to reception.**

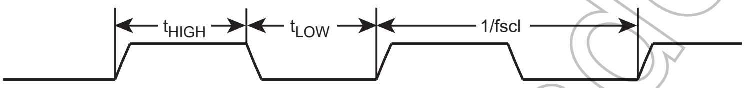
Note 2: **Do not set SBIxI2CAR to "0x00" in slave mode. (If SBIxI2CAR is set to "0x00", it’s recognized that the slave address matches the START byte ("0x01") of the I2C standard received in slave mode.)**

18.5 Control in the I2C Bus Mode

18.5.1 Serial Clock

18.5.1.1 Clock source

SBIxCR1<SCK[2:0]> specifies the maximum frequency of the serial clock to be output from the SCL pin in the master mode.



$$t_{LOW} = 2^{n-1}/fsys + 58/fsys$$
$$t_{HIGH} = 2^{n-1}/fsys + 14/fsys$$
$$fscl = 1/(t_{LOW} + t_{HIGH}) = \frac{fsys}{2^n + 72}$$

SBIxCR1<SCK[2:0]>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Figure 18-3 Clock source

Note: **The maximum speeds in the standard and high-speed modes are specified to 100kHz and 400kHz respectively following the communications standards. Notice that the internal SCL clock frequency is determined by the fsys used and the calculation formula shown above.**

18.5.1.2 Clock Synchronization

The I2C bus is driven by using the wired-AND connection due to its pin structure. The first master that pulls its clock line to the "Low" level overrides other masters producing the "High" level on their clock lines. This must be detected and responded by the masters producing the "High" level.

Clock synchronization assures correct data transfer on a bus that has two or more master.

For example, the clock synchronization procedure for a bus with two masters is shown below.

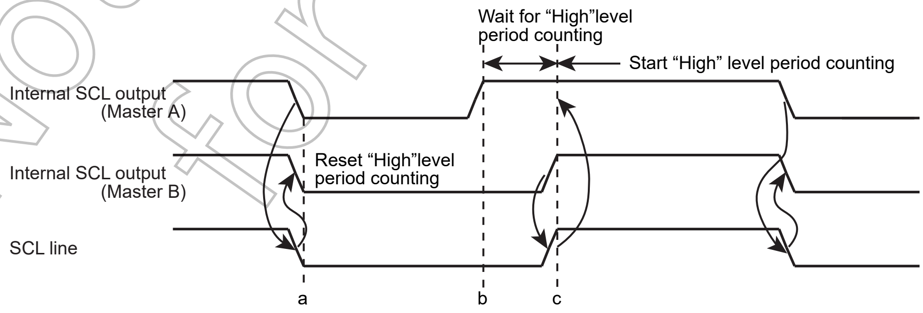

Wait for “High”level period counting
Start “High” level period counting
Internal SCL output (Master A)
Internal SCL output (Master B)
Reset “High”level period counting
SCL line
a
b
c

Figure 18-4 Example of Clock Synchronization

At the point a, Master A pulls its internal SCL output to the "Low" level, bringing the SCL bus line to the "Low" level. Master B detects this transition, resets its "High" level period counter, and pulls its internal SCL output level to the "Low" level.

Master A completes counting of its "Low" level period at the point b, and brings its internal SCL output to the "High" level. However, Master B still keeps the SCL bus line at the "Low" level, and Master A stops counting of its "High" level period counting.After Master A detects that Master B brings its internal SCL output to the "High" level and brings the SCL bus line to the "High" level at the point c, it starts counting of its "High" level period.

After that Master finishes counting the "High" level period, the Master pulls the SCL pin to "Low" and the SCL bus line becomes "Low".

This way, the clock on the bus is determined by the master with the shortest "High" level period and the master with the longest "Low" level period among those connected to the bus.

18.5.2 Setting the Acknowledgement Mode

Setting SBIxCR1<ACK> to "1" selects the acknowledge mode.When operating as a master, the SBI adds one clock for acknowledgment signal. In slave mode, the clock for acknowledgement signals is counted. In transmitter mode, the SBI releases the SDAx pin during clock cycle to receive acknowledgement signals from the receiver. In receiver mode, the SBI pulls the SDAx pin to the "Low" level during the clock cycle and generates acknowledgement signals. Also in slave mode, if a general-call address is received, the SBI pulls the SDAx pin to the "Low" level during the clock cycle and generates acknowledgement signals.

By setting <ACK> to "0", the non-acknowledgment mode is activated. When operating as a master, the SBI does not generate clock for acknowledgement signals. In slave mode, the clock for acknowledgement signals is counted.

18.5.3 Setting the Number of Bits per Transfer

SBIxCR1<BC[2:0]> specifies the number of bits of the next data to be transmitted or received.

Under the start condition, <BC[2:0]> is set to "000", causing a slave address and the direction bit to be transferred in a packet of eight bits. At other times, <BC[2:0]> keeps a previously programmed value.

18.5.4 Slave Addressing and Address Recognition Mode

Setting "0" to SBIxI2CAR<ALS> and a slave address in SBIxI2CAR<SA[6:0]> sets addressing format, and then the SBI recognizes a slave address transmitted by the master device and receives data in the addressing format.

If <ALS> is set to "1", the SBI does not recognize a slave address and receives data in the free data format. In the case of free data format, a slave address and a direction bit are not recognized; they are recognized as data immediately after generation of the start condition.

18.5.5 Operating mode

The setting of SBIxCR2<SBIM[1:0]> controls the operating mode. To operate in I2C mode, ensure that the serial bus interface pins are at "High" level before setting <SBIM[1:0]> to "10". Also, ensure that the bus is free before switching the operating mode to the port mode.

18.5.6 Configuring the SBI as a Transmitter or a Receiver

Setting SBIxCR2<TRX> to "1" configures the SBI as a transmitter. Setting <TRX> to "0" configures the SBI as a receiver.

At the slave mode:

- when data is transmitted in the addressing format.
- when the received slave address matches the value specified at SBIxI2CAR.
- when a general-call address is received; i.e., the eight bits following the start condition are all zeros.

If the value of the direction bit ($R/\overline{W}$) is "1", <TRX> is set to "1" by the hardware. If the bit is "0", <TRX> is set to "0".

As a master device, the SBI receives acknowledgement from a slave device. If the direction bit of "1" is transmitted, <TRX> is set to "0" by the hardware. If the direction bit is "0", <TRX> changes to "1". If the SBI does not receive acknowledgement, <TRX> retains the previous value.

<TRX> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

If SBI is used in free data format, <TRX> is not changed by the hardware.

18.5.7 Configuring the SBI as a Master or a Slave

Setting SBIxCR2<MST> to "1" configures the SBI to operate as a master device.

Setting <MST> to "0" configures the SBI as a slave device. <MST> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

18.5.8 Generating Start and Stop Conditions

When SBIxSR<BB> is "0", writing "1" to SBIxCR2<MST, TRX, BB, PIN> causes the SBI to start a sequence for generating the start condition and to output the slave address and the direction bit prospectively written in the data buffer register. <ACK> must be set to "1" in advance.

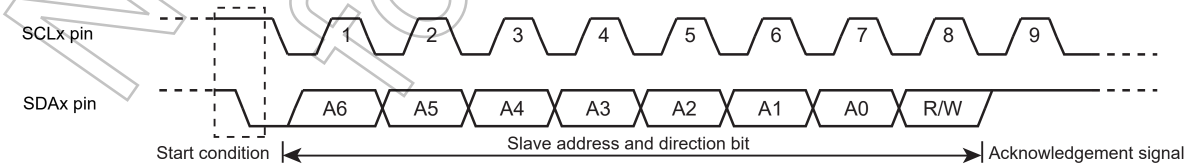


Figure 18-5 Generating the Start Condition and a Slave Address

When <BB> is "1", writing "1" to <MST, TRX, PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus. The contents of <MST, TRX, BB, PIN> should not be altered until the stop condition appears on the bus.

If SCL bus line is pulled "Low" by other devices when the stop condition is generated, the stop condition is generated after the SCL line is released.

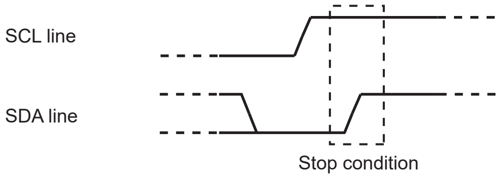


Figure 18-6 Generating the Stop Condition

SBIxSR<BB> can be read to check the bus state. <BB> is set to "1" when the start condition is detected on the bus (the bus is busy), and cleared to "0" when the stop condition is detected (the bus is free).

18.5.9 Interrupt Service Request and Release

In master mode, a serial bus interface request (INTSBIx) is generated when the transfer of the number of clock cycles set by <BC> and <ACK> is completed.

In slave mode, INTSBIx is generated under the following conditions.

- After output of the acknowledge signal which is generated when the received slave address matches the slave address set to SBIxI2CAR<SA[6:0]>.
- After the acknowledge signal is generated when a general-call address is received.
- When the slave address matches or a data transfer is completed after receiving a general-call address.

In the address recognition mode (<ALS> = "0"), INTSBIx is generated when the received slave address matches the values specified at SBIxI2CAR or when a general-call (eight bits data following the start condition is all "0") is received.

When an interrupt request (INTSBIx) is generated, SBIxCR2<PIN> is cleared to "0". While <PIN> is cleared to "0", the SBI pulls the SCL line to the "Low" level.

<PIN> is set to "1" when data is written to or read from SBIxDBR. It takes a period of t_{LOW} for the SCL line to be released after <PIN> is set to "1". When the program writes "1" to <PIN>, it is set to "1". However, writing "0" does not clear this bit to "0".

Note: When arbitration is lost in master mode, <PIN> is not cleared to "0" if the slave address does not match (INTSBIx is generated).

18.5.10 Arbitration Lost Detection Monitor

The I2C bus has the multi-master capability (there are two or more masters on a bus), and requires the bus arbitration procedure to ensure correct data transfer.

A master that attempts to generate the start condition while the bus is busy loses bus arbitration, with no start condition occurring on the SDA and SCL lines.The I2C-bus arbitration takes place on the SDA line.

The arbitration procedure for two masters on a bus is shown below.

Up until the point a, Master A and Master B output the same data. At the point a, Master A outputs the "Low" level and Master B outputs the "High" level.

Then Master A pulls the SDA bus line to the "Low" level because the line has the wired-AND connection. When the SCL line goes high at the point b, the slave device reads the SDA line data, i.e., data transmitted by Master A. At this time, data transmitted by Master B becomes invalid.

This condition of Master B is called "Arbitration Lost". Master B releases its SDA pin, so that it does not affect the data transfer initiated by another master. If two or more masters have transmitted exactly the same first data word, the arbitration procedure continues with the second data word.

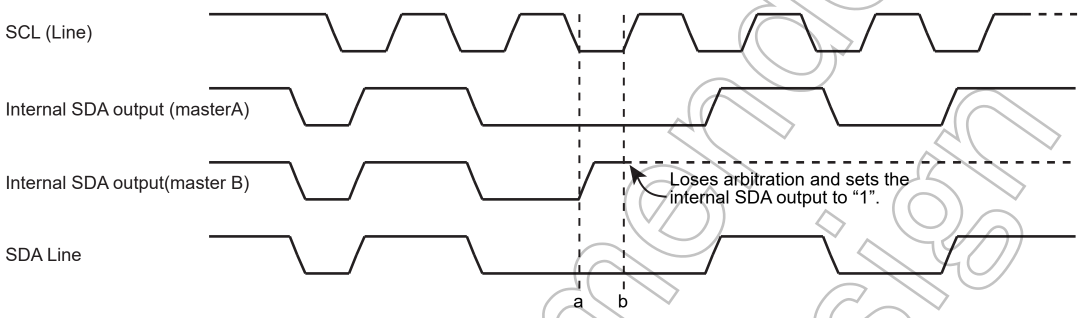

SCL (Line)
Internal SDA output (masterA)
Internal SDA output(master B)
SDA Line
Loses arbitration and sets the internal SDA output to “1”.
a
b

Figure 18-7 Lost Arbitration

A master compares the SDA bus line level and the internal SDA output level at the rising of the SCL line. If there is a difference between these two values, Arbitration Lost occurs and SBIxSR<AL> is set to "1".

When <AL> is set to "1", SBIxSR<MST, TRX> are cleared to "0", causing the SBI to operate as a slave receiver.Therefore, the serial bus interface circuit stops the clock output during data transfer after <AL> is set to "1".

<AL> is cleared to "0" when data is written to or read from SBIxDBR or data is written to SBIxCR2.

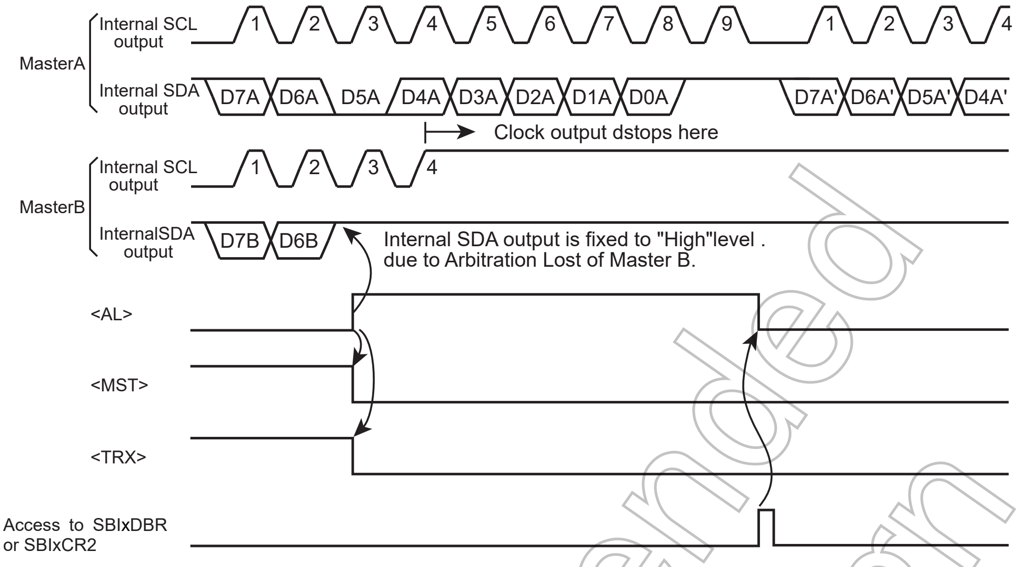


Figure 18-8 Example of Master B Lost Arbitration (D7A = D7B, D6A = D6B)

18.5.11 Slave Address Match Detection Monitor

When the SBI operates as a slave device in the address recognition mode (SBIxI2CAR<ALS>="0"), SBIxSR<AAS> is set to "1" on receiving the general-call address or the slave address that matches the value specified at SBIxI2CAR.

When <ALS> is "1", <AAS> is set to "1" when the first data word has been received. <AAS> is cleared to "0" when data is written to or read from SBIxDBR.

18.5.12 General-call Detection Monitor

When the SBI operates as a slave device, SBIxSR<ADO> is set to "1" when it receives the general-call address; i.e., the eight bits following the start condition are all zeros.

<ADO> is cleared to "0" when the start or stop condition is detected on the bus.

18.5.13 Last Received Bit Monitor

SBIxSR<LRB> is set to the SDA line value that was read at the rising of the SCL line.

In the acknowledgment mode, reading SBIxSR<LRB> immediately after generation of the INTSBIx interrupt request causes ACK signal to be read.

18.5.14 Data Buffer Register (SBIxDBR)

Reading or writing SBIxDBR initiates reading received data or writing transmitted data.

When the SBI is acting as a master, setting a slave address and a direction bit to this register generates the start condition.

18.5.15 Baud Rate Register (SBIxBR0)

The SBIxBR0<I2SBI> register determines if the SBI operates or not when it enters the IDLE mode.

This register must be programmed before executing an instruction to switch to the standby mode.

18.5.16 Software Reset

If the serial bus interface circuit locks up due to external noise, it can be initialized by using a software reset.

Writing "10" followed by "01" to SBIxCR2<SWRST[1:0]> generates a reset signal that initializes the serial bus interface circuit. When writing, set <SBIM[1:0]> to "10"; I2Cbus mode. After a reset, all control registers and status flags are initialized to their reset values. When the serial bus interface is initialized, <SWRST> is automatically cleared to "0".

Note: A software reset causes the SBI operating mode to switch from the I2C mode to the port mode.

18.6 Data Transfer Procedure in the I2C Bus ModeI2C

18.6.1 Device Initialization

First, program SBIxCR1<ACK, SCK[2:0]>. Writing "000" to SBIxCR1<BC[2:0]> at the time.

Next, program SBIxI2CAR by specifying a slave address at <SA[6:0]> and an address recognition mode at <ALS>. (<ALS> must be cleared to "0" when using the addressing format).

To configure the Serial Bus Interface as a slave receiver, ensure that the serial bus interface pin is at "High" first. Then write "0" to SBIxCR2<MST, TRX, BB>, "1" to <PIN>, "10" to <SBIM[1:0]> and "0" to the bit 1 and 0.

Note: Initialization of the serial bus interface circuit must be completed within a period that any device does not generate start condition after all devices connected to the bus were initialized. If this rule is not followed, data may not be received correctly because other devices may start transfer before the initialization of the serial bus interface circuit is completed.

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	0	0	X	0	X	X	X	Specifies ACK and SCL clock.
SBIxI2CAR	←	X	X	X	X	X	X	X	X	Specifies a slave address and an address recognition mode.
SBIxCR2	←	0	0	0	1	1	0	0	0	Configures the SBI as a slave receiver.

Note: X; Don't care

18.6.2 Generating the Start Condition and a Slave Address

18.6.2.1 Master mode

In the master mode, the following steps are required to generate the start condition and a slave address.

First, ensure that the bus is free (<BB> = "0"). Then, write "1" to SBIxCR1<ACK> to select the acknowledgment mode. Write to SBIxDBR a slave address and a direction bit to be transmitted.

When <BB> = "0", writing "1111" to SBIxCR2<MST, TRX, BB, PIN> generates the start condition on the bus. Following the start condition, the SBI generates nine clocks from the SCL pin. The SBI outputs the slave address and the direction bit specified at SBIxDBR with the first eight clocks, and releases the SDA line in the ninth clock to receive an acknowledgment signal from the slave device.

The INTSBIx interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0". In the master mode, the SBI holds the SCL line at the "Low" level while <PIN> is = "0".<TRX> changes its value according to the transmitted direction bit at generation of the INTSBIx interrupt request, provided that an acknowledgment signal has been returned from the slave device.

Note: To output salve address, check with software that the bus is free before writing to SBIxDBR. If this rule is not followed, data being output on the bus may get ruined.

Settings in main routine

```
                       7   6   5   4   3   2   1   0
Reg.            ←      SBIxSR
Reg.            ←      Reg. e 0x20
if Reg.         ≠      0x00                                 Ensures that the bus is free.

Then

SBIxCR1         ←      X   X   X   1   0   X   X   X        Selects the acknowledgement mode.
SBIxDBR         ←      X   X   X   X   X   X   X   X        Specifies the desired slave address and direction.
SBIxCR2         ←      1   1   1   1   1   0   0   0        Generates the start condition.
                Example of INTSBI0 interrupt routine
                       Clears the interrupt request.
                       Processing
                       End of interrupt
```

18.6.2.2 Slave mode

In the slave mode, the SBI receives the start condition and a slave address.

After receiving the start condition from the master device, the SBI receives a slave address and a direction bit from the master device during the first eight clocks on the SCL line.

If the received address matches its slave address specified at SBIxI2CAR or is equal to the general-call address, the SBI pulls the SDA line to the "Low" level during the ninth clock and outputs an acknowledgment signal.

The INTSBIx interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0". In the slave mode, the SBI holds the SCL line at the "Low" level while <PIN> is "0".

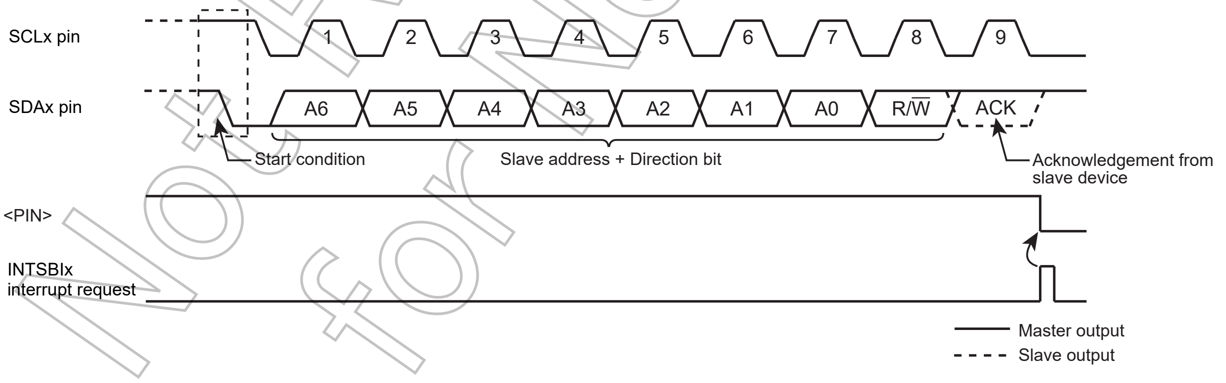


Figure 18-9 Generation of the Start Condition and a Slave Address

18.6.3 Transferring a Data Word

At the end of a data word transfer, the INTSBIx interrupt is generated to test <MST> to determine whether the SBI is in the master or slave mode.

18.6.3.1 Master mode (<MST> = "1")

Test <TRX> to determine whether the SBI is configured as a transmitter or a receiver.

(1) Transmitter mode (<TRX> = "1")

Test <LRB>. If <LRB> is "1", that means the receiver requires no further data.

The master then generates the stop condition as described later to stop transmission.

If <LRB> is "0", that means the receiver requires further data.If the next data to be transmitted has eight bits, the data is written into SBIxDBR. If the data has different length, <BC[2:0]> and <ACK> are programmed and the transmit data is written into SBIxDBR.Writing the data makes <PIN> to "1", causing the SCL pin to generate a serial clock for transferring a next data word, and the SDA pin to transfer the data word.

After the transfer is completed, the INTSBIx interrupt request is generated, <PIN> is cleared to "0", and the SCL pin is pulled to the "Low" level.

To transmit more data words, test <LRB> again and repeat the above procedure.

INTSBIx interrupt

```
if MST = 0
Then go to the slave-mode processing.
if TRX = 0
Then go to the receiver-mode processing.
if LRB = 0
Then go to processing for generating the stop condition.
SBIxCR1    ←  X  X  X  X  0  X  X  X    Specifies the number of bits to be transmitted and
                                        specify whether ACK is required.
SBIxDBR    ←  X  X  X  X  X  X  X  X    Writes the transmit data.
End of interrupt processing.
```

Note: **X; Don’t care**

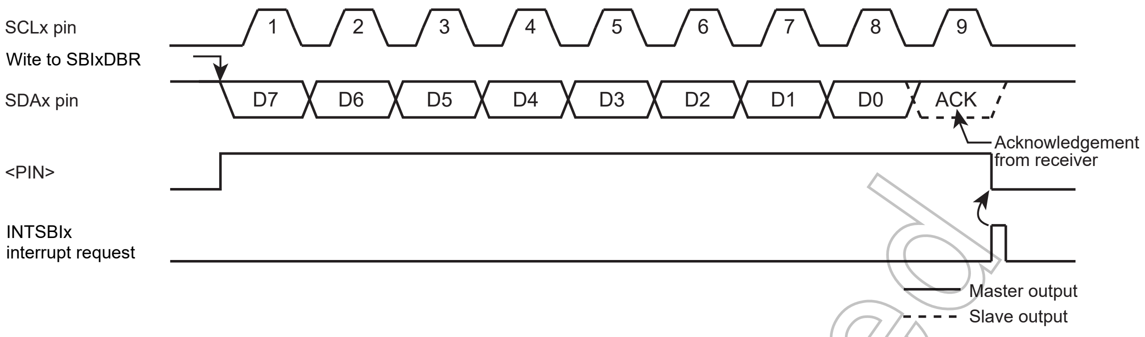


Figure 18-10 <BC[2:0]>= "000",<ACK>= "1" (Transmitter Mode)

(2) Receiver mode (<TRX> = "0")

If the next data to be transmitted has eight bits, the transmit data is written into SBIxDBR.

If the data has different length, <BC[2:0]> and <ACK> are programmed and the received data is read from SBIxDBR to release the SCL line. (The data read immediately after transmission of a slave address is undefined.)On reading the data, <PIN> is set to "1", and the serial clock is output to the SCL pin to transfer the next data word.In the last bit, when the acknowledgment signal becomes the "Low" level, "0" is output to the SDA pin.

After that, the INTSBIx interrupt request is generated, and <PIN> is cleared to "0", pulling the SCL pin to the "Low" level.Each time the received data is read from SBIxDBR, one-word transfer clock and an acknowledgement signal are output.

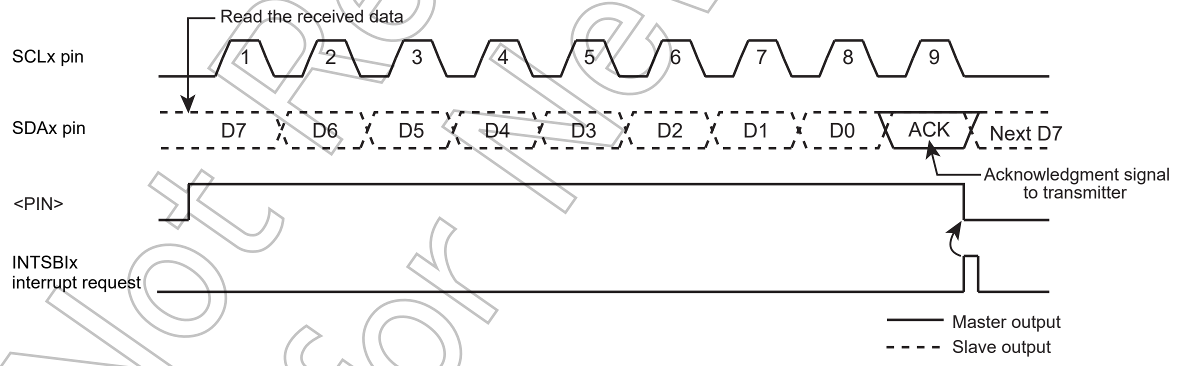


Figure 18-11 <BC[2:0]>= "000",<ACK>= "1" (Receiver Mode)

To terminate the data transmission from the transmitter, <ACK> must be cleared to "0" immediately before reading the data word second to last.

This disables generation of an acknowledgment clock for the last data word.

When the transfer is completed, an interrupt request is generated. After the interrupt processing, <BC[2:0]> must be set to "001" and the data must be read so that a clock is generated for 1-bit transfer.

At this time, the master receiver holds the SDA bus line at the "High" level, which signals the end of transfer to the transmitter as an acknowledgment signal.

In the interrupt processing for terminating the reception of 1-bit data, the stop condition is generated to terminate the data transfer.

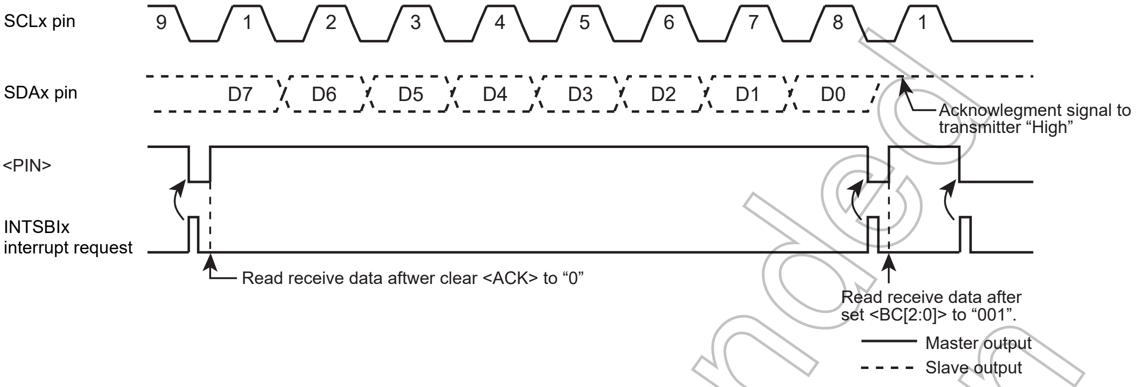

SCLx pin
9
1
2
3
4
5
6
7
8
1
SDAx pin
D7
D6
D5
D4
D3
D2
D1
D0
Acknowlegment signal to transmitter "High"
<PIN>
INTSBIx interrupt request
Read receive data aftwer clear <ACK> to "0"
Read receive data after set <BC[2:0]> to "001".
Master output
Slave output

Figure 18-12 Terminating Data Transmission in the Master Receiver Mode

Example: When receiving N data word

INTSBIx interrupt (after data transmission)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	X	X	X	X	0	X	X	X	Sets the number of bits of data to be received and specify whether ACK is required.
Reg.	←	SBIxDBR								Reads dummy data.

End of interrupt

INTSBIx interrupt (first to (N-2)th data reception)

		7	6	5	4	3	2	1	0	
Reg.	←	SBIxDBR								Reads the first to (N-2)th data words.

End of interrupt

INTSBIx interrupt ((N-1)th data reception)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	X	X	X	0	0	X	X	X	Disables generation of acknowledgement clock.
Reg.	←	SBIxDBR								Reads the (N-1)th data word.

End of interrupt

INTSBIx interrupt (Nth data reception)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	0	1	0	0	X	X	X	Disables generation of acknowledgement clock.
Reg.	←	SBIxDBR								Reads the Nth data word.

End of interrupt

INTSBIx interrupt (after completing data reception)

Processing to generate the stop condition.	Terminates the data transmission.

End of interrupt

Note: X; Don't care

18.6.3.2 Slave mode (<MST> = "0")

In the slave mode, the SBI generates the INTSBIx interrupt request on four occasions:

1) when the SBI has received any slave address from the master.

2) when the SBI has received a general-call address.

3) when the received slave address matches its address.

4) when a data transfer has been completed in response to a general-call.

Also, if the SBI detects Arbitration Lost in the master mode, it switches to the slave mode.

Upon the completion of data word transfer in which Arbitration Lost is detected, the INTSBIx interrupt request is generated, <PIN> is cleared to "0", and the SCL pin is pulled to the "Low" level.

When data is written to or read from SBIxDBR or when <PIN> is set to "1", the SCLx pin is released after a period of t_{LOW}.

In the slave mode, the normal slave mode processing or the processing as a result of Arbitration Lost is carried out.

SBIxSR<AL>, <TRX>, <AAS> and <ADO> are tested to determine the processing required.

"Table 18-2 Processing in Slave Mode"shows the slave mode states and required processing.

Example: When the received slave address matches the SBI's own address and the direction bit is "1" in the slave receiver mode.

INTSBIx interrupt

```
if TRX = 0
Then go to other processing.
if AL = 0
Then go to other processing.
if AAS = 0
Then go to other processing.
SBIxCR1     ←   X   X   X   1   0   X   X   X      Sets the number of bits to be transmitted.
SBIxDBR     ←   X   X   X   X   X   X   X   X      Sets the transmit data.
```

Note:X; Don’t care

Table 18-2 Processing in Slave Mode

<TRX>	<AL>	<AAS>	<ADO>	State	Processing
1	1	1	0	Arbitration Lost is detected while the slave address was being transmitted and the SBI received a slave address with the direction bit "1" transmitted by another master.	Set the number of bits in a data word to <BC[2:0]> and write the transmit data into SBIxDBR.
	0	1	0	In the slave receiver mode, the SBI received a slave address with the direction bit "1" transmitted by the master.	
		0	0	In the slave transmitter mode, the SBI has completed a transmission of one data word.	Test LRB. If it has been set to "1", that means the receiver does not require further data. Set <PIN> to 1 and reset <TRX> to 0 to release the bus. If <LRB> has been reset to "0", that means the receiver requires further data. Set the number of bits in the data word to <BC[2:0]> and write the transmit data to the SBIxDBR.
0	1	1	1/0	Arbitration Lost is detected while a slave address is being transmitted, and the SBI receives either a slave address with the direction bit "0" or a general-call address transmitted by another master.	Read the SBIxDBR (a dummy read) to set <PIN> to 1, or write "1" to <PIN>.
		0	0	Arbitration Lost is detected while a slave address or a data word is being transmitted, and the transfer is terminated.	
	0	1	1/0	In the slave receiver mode, the SBI received either a slave address with the direction bit "0" or a general-call address transmitted by the master.	
		0	1/0	In the slave receiver mode, the SBI has completed a reception of a data word.	Set the number of bits in the data word to <BC [2:0].> and read the received data from SBIxDBR.

18.6.4 Generating the Stop Condition

When SBIxSR<BB> is "1", writing "1" to SBIxCR2<MST, TRX, PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus.

Do not alter the contents of <MST, TRX, BB, PIN> until the stop condition appears on the bus.

If another device is holding down the SCL bus line, the SBI waits until the SCL line is released.

After that, the SDA pin goes "High", causing the stop condition to be generated.

		7	6	5	4	3	2	1	0	
SBIxCR2	←	1	1	0	1	1	0	0	0	Generates the stop condition.

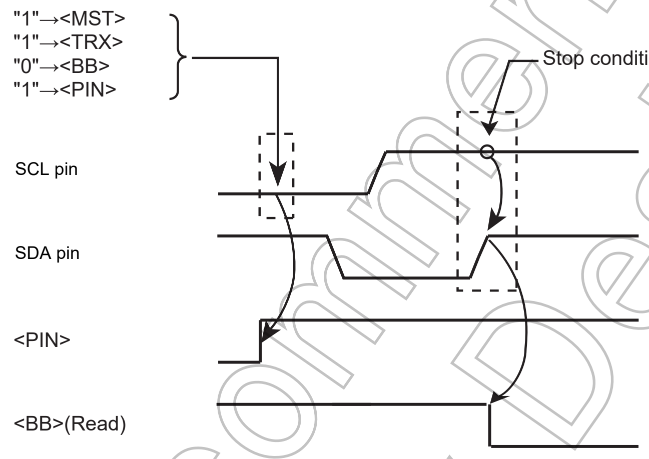


Figure 18-13 Generating the Stop Condition

18.6.5 Restart Procedure

Restart is used when a master device changes the data transfer direction without terminating the transfer to a slave device.The procedure of generating a restart in the master mode is described below.

First, write SBIxCR2<MST, TRX, BB> to "0" and write "1" to <PIN> to release the bus. At this time, the SDAx pin is held at the "High" level and the SCLx pin is released. Because no stop condition is generated on the bus, other devices recognize that the bus is busy.

Then, test SBIxSR<BB> and wait until it becomes "0" to ensure that the SCLx pin is released.

Next, test <LRB> and wait until it becomes "1" to ensure that no other device is pulling the SCLx bus line to the "Low" level.

Once the bus is determined to be free by following the above procedures, follow the procedures described in "18.6.2 Generating the Start Condition and a Slave Address"to generate the start condition.

To satisfy the setup time of restart, at least 4.7μs wait period (in the standard mode) must be created by the software after the bus is determined to be free.

Note 1: **Do not write <MST> to "0" when it is "0". (Restart cannot be initiated.)**

Note 2: **When the master device is acting as a receiver, data transmission from the slave device which serves as a transmitter must be completed before generating a restart. To complete data transfer, slave device must receive a "High" level acknowledge signal. For this reason, <LBR> before generating a restart becomes "1", the rising edge of the SCL line is not detected even <LBR>=**

"1" is confirmed by following the restart procedure. To check the status of the SCL line, read the port.

```
                            7   6   5   4   3   2   1   0
┌→
│   SBIxCR2        ←        0   0   0   1   1   0   0   0      Releases the bus.
└
    if SBIxSR<BB> ≠ 0                                         Checks that the SCL pin is released.
┌→
│   Then
└   if SBIxSR<LRB> ≠ 1                                        Checks that no other device is pulling the SCL pin to the "Low".
    Then
    4.7 μs Wait
    SBIxCR1        ←        X   X   X   1   0   X   X   X      Selects the acknowledgment mode.
    SBIxDBR        ←        X   X   X   X   X   X   X   X      Sets the desired slave address and direction.
    SBIxCR2        ←        1   1   1   1   1   0   0   0      Generates the start condition.
```

Note: X; Don’t care

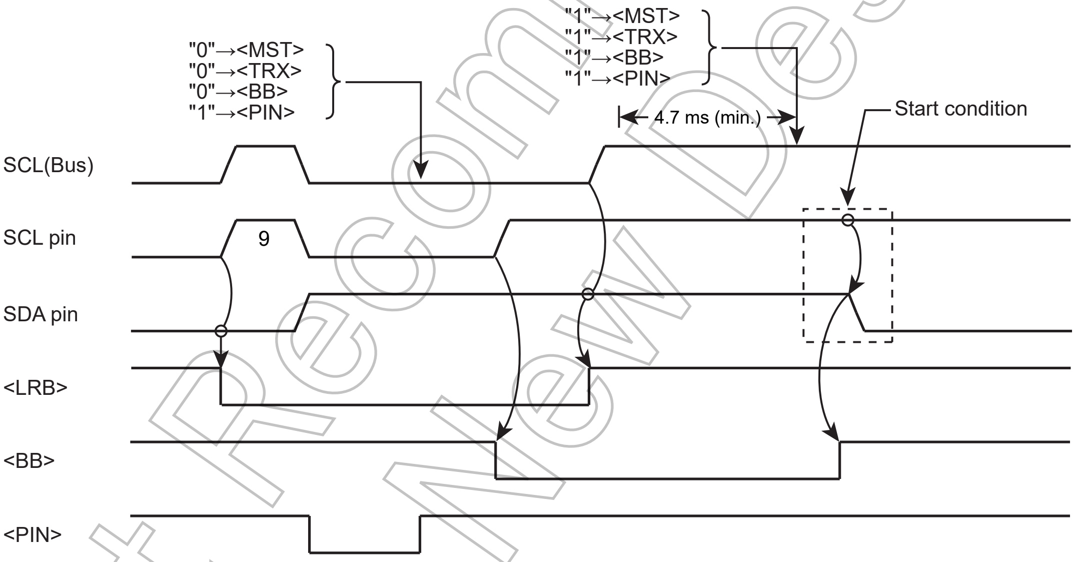


Figure 18-14 Timing Chart of Generating a Restart

18.7 Control register of SIO mode

The following registers control the serial bus interface in the clock-synchronous 8-bit SIO mode and provide its status information for monitoring.

18.7.1 SBIxCR0(control register 0)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SBIEN	R/W	Serial bus interface operation. 0:Disable 1: Enable Enable this bit before using the serial bus interface. If this bit is disabled, power consumption can be reduced because all clocks except SBIxCR0 stop. If the serial bus interface operation is enabled and then disabled, the settings will be maintained in each register.
6-0	-	R	Read as 0.

18.7.2 SBIxCR1(Control register 1)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SIOS	R/W	Transfer Start/Stop 0: Stop 1: Start
6	SIOINH	R/W	Transfer 0: Continue 1: Forced termination
5-4	SIOM[1:0]	R/W	Select transfer mode 00: Transmit mode 01: Reserved 10:Transmit/receive mode 11:Receive mode
3	-	R	Read as 1.
2-0	SCK[2:0]	R/W	On writing <SCK[2:0]>: Select serial clock frequency. (Note 1) 000 n = 3 2.5 MHz 001 n = 4 1.25 MHz 010 n = 5 625 kHz 011 n = 6 313 kHz 100 n = 7 156 kHz 101 n = 8 78 kHz 110 n = 9 39 kHz 111 − External clock System clock: fsys (=40MHz) Clock gear: fc/1 Frequency = $\frac{fsys/2}{2^n}$ [Hz]

Note 1: **After a reset, the <SCK[0]> bit is read as "1". However, if the SIO mode is selected at the SBIxCR2 register, the initial value is read as "0". In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state. The descriptions of the SBIxCR2 register and the SBIxSR register are the same.**

Note 2: **Set <SIOS> to "0" and <SIOINH> to "1" before programming the transfer mode and the serial clock.**

18.7.3 SBIxDBR (Data buffer register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DB[7:0]	R	Receive data
		W	Transmit data

Note 1: **The transmission data must be written in to the register from the MSB (bit 7). The received data is stored in the LSB.**

Note 2: **Since SBIxDBR has independent buffers for writing and reading, a written data cannot be read. Thus, read-modify-write instructions, such as bit manipulation, cannot be used.**

18.7.4 SBIxCR2(Control register 2)

This register serves as SBIxSR register by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-4	-	R	Read as 1. (Note 1)
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note 2) 00: Port mode 01: SIO mode 10: I2Cbus mode 11: Reserved
1-0	-	R	Read as 1. (Note 1)

Note 1: **In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state.**

Note 2: **Make sure that modes are not changed during a communication session.**

18.7.5 SBIxSR (Status Register)

This register serves as SBIxCR2 by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-4	-	R	Read as 1.(Note 1)
3	SIOF	R	Serial transfer status monitor. 0: Completed 1: In progress
2	SEF	R	Shift operation status monitor 0: Completed. 1: In progress
1-0	-	R	Read as 1. (Note 1)

Note: **In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state.**

18.7.6 SBIxBR0 (Baud rate register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
After reset	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	-	R	Read as 1.
6	I2SBI	R/W	Operation in IDLE mode. 0: Stop 1: Operate
5-1	-	R	Read as 1.
0	-	R/W	Make sure to write "0".

18.8 Control in SIO mode

18.8.1 Serial Clock

18.8.1.1 Clock source

Internal or external clocks can be selected by programming SBIxCR1<SCK[2:0]>.

(1) Internal clocks

In the internal clock mode, one of the seven frequencies can be selected as a serial clock, which is output to the outside through the SCKx pin.

At the beginning of a transfer, the SCKx pin output becomes the "High" level.

If the program cannot keep up with this serial clock rate in writing the transmit data or reading the received data, the SBI automatically enters a wait period. During this period, the serial clock is stopped automatically and the next shift operation is suspended until the processing is completed.

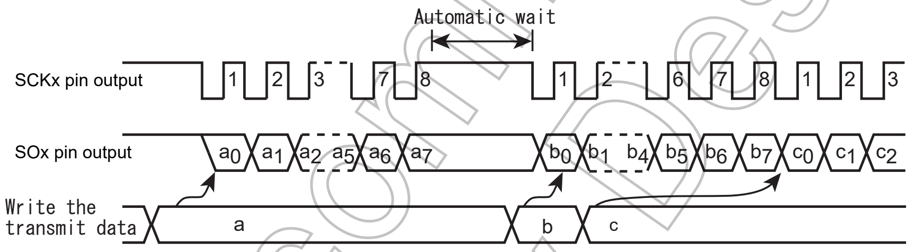


Figure 18-15 Automatic Wait

(2) External clock (<SCK[2:0]> = "111")

The SBI uses an external clock supplied from the outside to the SCKx pin as a serial clock.

For proper shift operations, the serial clock at the "High" and "Low" levels must have the pulse widths as shown below.

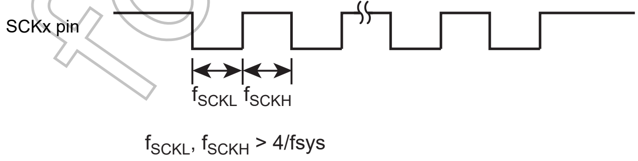


Figure 18-16 Maximum Transfer Frequency of External Clock Input

18.8.1.2 Shift Edge

Leading-edge shift is used in transmission. Trailing-edge shift is used in reception.

- Leading-edge shift

  Data is shifted at the leading edge of the serial clock (or the falling edge of the SCKx pin input/output).

- Trailing-edge shift

  Data is shifted at the trailing edge of the serial clock (or the rising edge of the SCKx pin input/output).

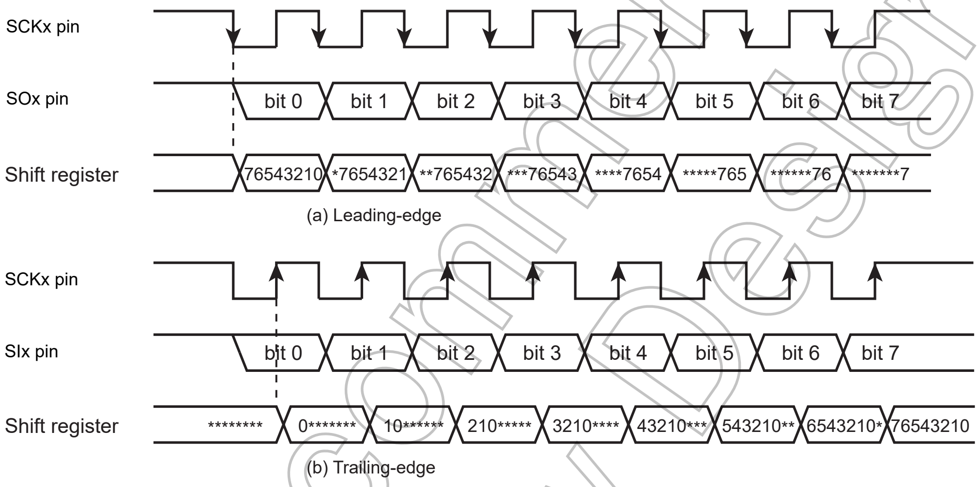


Figure 18-17 Shift Edge

18.8.2 Transfer Modes

The transmit mode, the receive mode or the transmit/receive mode can be selected by programming SBIxCR1<SIOM[1:0]>.

18.8.2.1 8-bit transmit mode

Set the control register to the transmit mode and write the transmit data to SBIxDBR.

After writing the transmit data, writing "1" to SBIxCR1<SIOS> starts the transmission. The transmit data is moved from SBIxDBR to a shift register and output to the SO pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the transmit data is transferred to the shift register, SBIxDBR becomes empty, and the INTSBIx (buffer-empty) interrupt is generated, requesting the next transmit data.

In the internal clock mode, the serial clock will be stopped and automatically enter the wait state, if next data is not loaded after the 8-bit data has been fully transmitted. The wait state will be cleared when SBIxDBR is loaded with the next transmit data.

In the external clock mode, SBIxDBR must be loaded with data before the next data shift operation is started. Therefore, the data transfer rate varies depending on the maximum latency between when the interrupt request is generated and when SBIxDBR is loaded with data in the interrupt service program.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting SBIxSR<SIOF> to "1" to the falling edge of SCK.

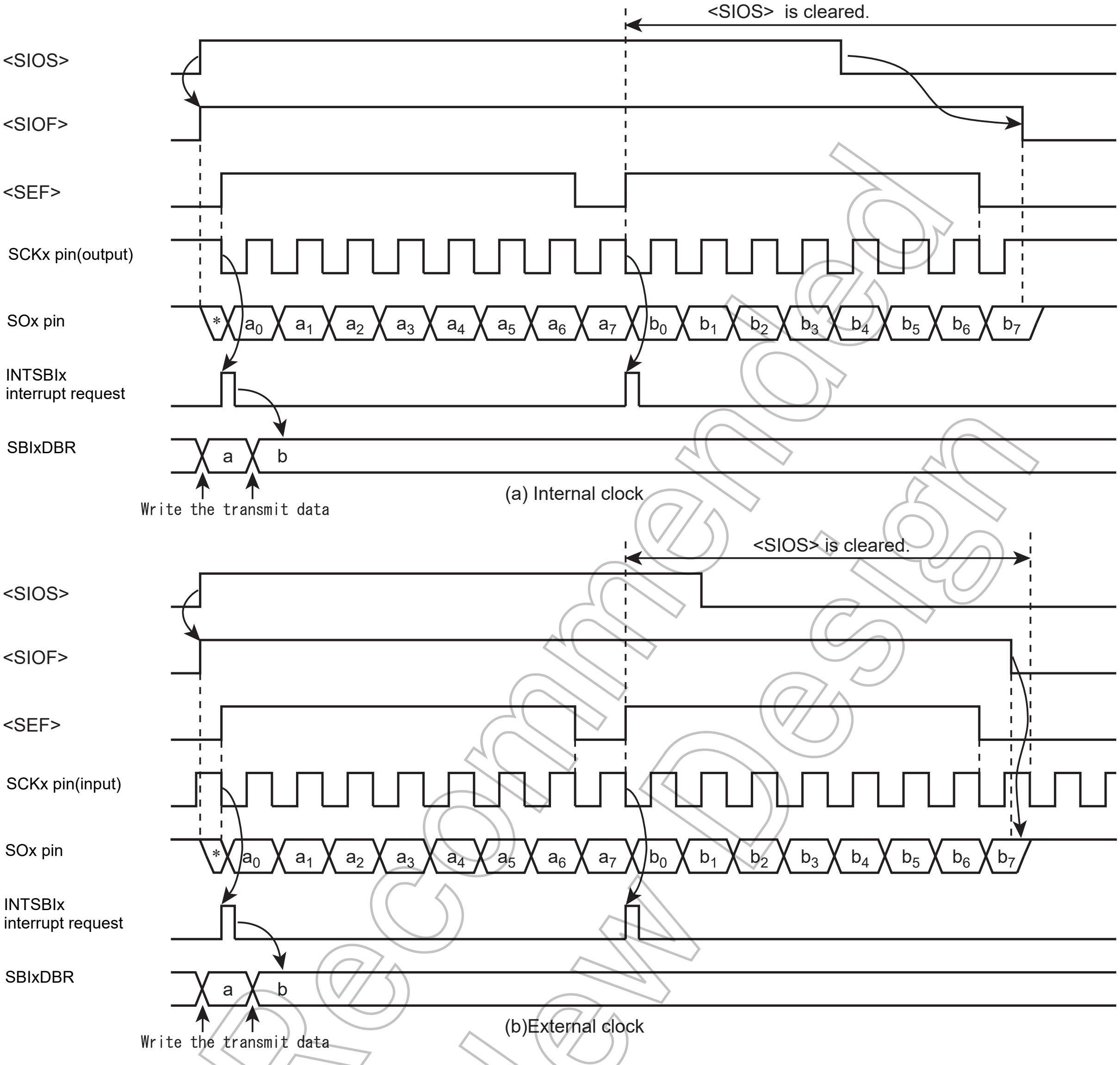
Transmission can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, remaining data is output before transmission ends. The program checks SBIxSR<SIOF> to determine whether transmission has come to an end. <SIOF> is cleared to "0" at the end of transmission. If <SIOINH> is set to "1", the transmission is aborted immediately and <SIOF> is cleared to "0".

When in the external clock mode, <SIOS> must be cleared to "0" before next data shifting. If <SIOS> does not be cleared to "0" before next data shifting, SBI output dummy data and stopped.

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	0	0	0	X	X	X	Selects the transmit mode.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBIxCR1	←	1	0	0	0	0	X	X	X	Starts transmission.

INTSBIx interrupt

SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.



(a) Internal clock

(b)External clock

Figure 18-18 Transmit Mode

Example: Example of programming (external clock) to terminate transmission by <SIO>

		7	6	5	4	3	2	1	0	
if SBIxSR<SIOF> ≠ 0										Recognizes the completion of the transmission.
Then										
if SCK ≠ 1										Recognizes "1" is set to the SCK pin by monitoring the port.
Then										
SBIxCR1	←	0	0	0	0	0	1	1	1	Completes the transmission by setting <SIOS> = 0.

18.8.2.2 8-bit receive mode

Set the control register to the receive mode. Then writing "1" to SBIxCR1<SIOS> enables reception.Data is taken into the shift register from the SI pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIxDBR and the INTSBIx (buffer-full) interrupt request is generated to request reading the received data. The interrupt service program then reads the received data from SBIxDBR.

In the internal clock mode, the serial clock will be stopped and automatically be in the wait state until the received data is read from SBIxDBR.

In the external clock mode, shift operations are executed in synchronization with the external clock. The maximum data transfer rate varies, depending on the maximum latency between generating the interrupt request and reading the received data

Reception can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, reception continues until all the bits of received data are written to SBIxDBR. The program checks SBIxSR<SIOF> to determine whether reception has come to an end.<SIOF> is cleared to "0" at the end of reception. After confirming the completion of the reception, last received data is read. If <SIOINH> is set to "1", the reception is aborted immediately and <SIOF> is cleared to "0". (The received data becomes invalid, and there is no need to read it out.)

Note: **The contents of SBIxDBR will not be retained after the transfer mode is changed. The ongoing reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.**

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	1	1	0	X	X	X	Selects the receive mode.
SBIxCR1	←	1	0	1	1	0	X	X	X	Starts reception.

INTSBIx interrupt

Reg.	←	SBIxDBR	Reads the received data.

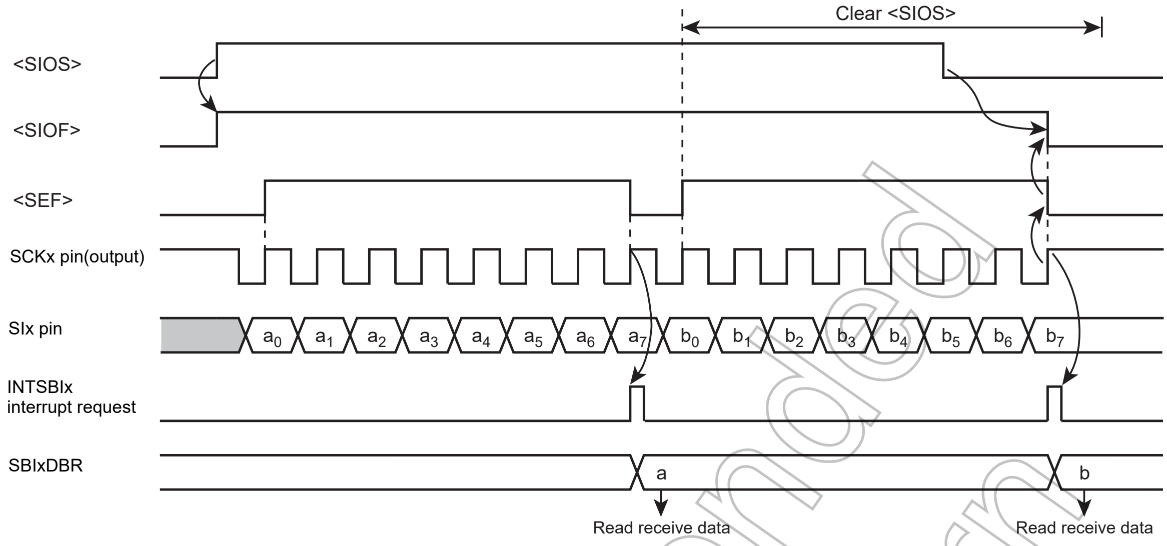

Clear <SIOS>
<SIOS>
<SIOF>
<SEF>
SCKx pin(output)
SIx pin
a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7
INTSBIx interrupt request
SBIxDBR
a
b
Read receive data
Read receive data

Figure 18-19 Receive Mode (Example: Internal Clock)

18.8.2.3 8-bit transmit/receive mode

Set the control register to the transfer/receive mode. Then writing the transmit data to SBIxDBR and setting SBIxCR1<SIOS> to "1" enables transmission and reception.The transmit data is output through the SOx pin at the falling of the serial clock, and the received data is taken in through the SI pin at the rising of the serial clock, with the least-significant bit (LSB) first. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIxDBR and the INTSBIx interrupt request is generated.The interrupt service program reads the received data from the data buffer register and writes the next transmit data. Because SBIxDBR is shared between transmit and receive operations, the received data must be read before the next transmit data is written.

In the internal clock operation, the serial clock will be automatically in the wait state until the received data is read and the next transmit data is written.

In the external clock mode, shift operations are executed in synchronization with the external serial clock. Therefore, the received data must be read and the next transmit data must be written before the next shift operation is started.The maximum data transfer rate for the external clock operation varies depending on the maximum latency between when the interrupt request is generated and when the transmit data is written.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting <SIOF> to "1" to the falling edge of SCK.

Transmission and reception can be terminated by clearing <SIOS> to "0" or setting SBIxCR1<SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, transmission and reception continue until the received data is fully transferred to SBIxDBR. The program checks SBIxSR<SIOF> to determine whether transmission and reception have come to an end. <SIOF> is cleared to "0" at the end of transmission and reception.If <SIOINH> is set to "1", the transmission and reception is aborted immediately and <SIOF> is cleared to "0".

Note: **The contents of SBIxDBR will not be retained after the transfer mode is changed. The ongoing transmission and reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.**

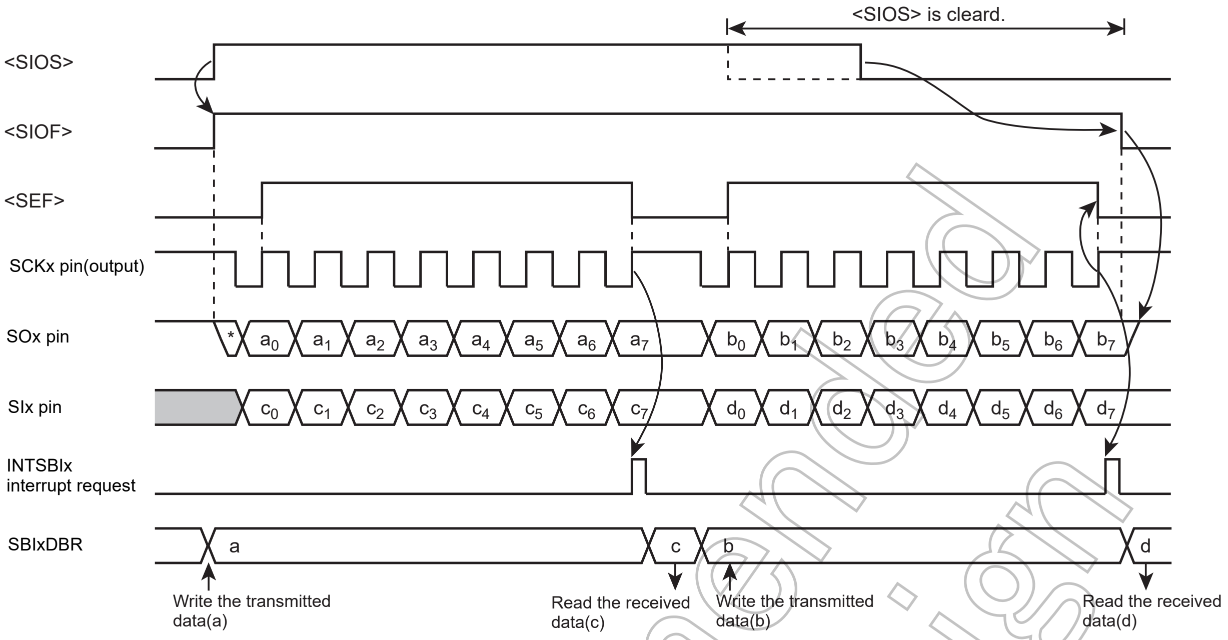


Figure 18-20 Transmit/Receive Mode (Example: Internal Clock)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	1	0	0	X	X	X	Selects the transmit mode.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBIxCR1	←	1	0	1	0	0	X	X	X	Starts reception/transmission.

INTSBIx interrupt

Reg.	←	SBIxDBR								Reads the received data.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

18.8.2.4 Data retention time of the last bit at the end of transmission

Under the condition SBIxCR1<SIOS>= "0", the last bit of the transmitted data retains the data of SCK rising edge as shown below. Transmit mode and transmit/receive mode are the same.

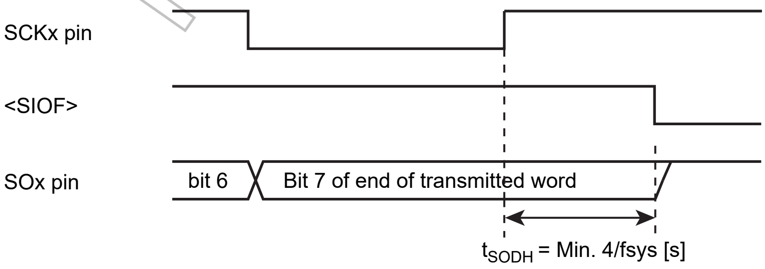

SCKx pin
<SIOF>
SOx pin
bit 6
Bit 7 of end of transmitted word
t_{SODH} = Min. 4/fsys [s]

Figure 18-21 Data retention time of the last bit at the end of transmission

19. Analog/Digital Converter (ADC)

19.1 Outline

TMPM341FDXBG/FYXBG contains a 12-bit, sequential-conversion analog/digital converter (ADC) with 15 analog input channels.

These 15 analog input channels (pins AIN00 through AIN14) are also used as input/output ports.

The 12-bit AD converter has the following features:

- Starting normal AD conversion and highest-priority AD conversion

  Software activation

  Activation with the 16-bit timer (TMRB)

  Hardware activation with an external trigger input ($\overline{\text{ADTRG}}$ pin)
- AD conversion

  Fixed-channel single conversion mode

  Channel scan single conversion mode

  Fixed-channel repeat conversion mode

  Channel scan repeat conversion mode
- Highest-priority AD conversion
- Normal AD conversion completion interrupt and highest-priority AD conversion completion interrupt
- Normal AD conversion and highest-priority AD conversion status flags.

  A flag indicating the AD conversion result data is valid, <ADRxRF>

  A flag indicating the AD conversion result data is overwritten, <OVRx>

  Normal AD conversion completion flag and highest-priority AD conversion completion flag

  Normal AD conversion busy flag and highest-priority AD conversion busy flag
- AD Monitor Function

  When the AD monitor function is enabled, an interrupt is generated if any comparison result is matched.
- AD conversion clock can be controlled from 1/fc to 1/16fc.
- When AD conversion is completed, two types of DMA requests are supported.
- Standby mode is supported.
- Output switching monitor function

  This is the function that monitors output switching operation of general input-output ports, which is also used as analog input channels (pins AIN00 through AIN14), during AD conversion. This monitor function is used to suggest the possibility that output switching operation during AD conversion affects conversion accuracy.

19.2 Configuration

Figure 19-1 shows the block diagram of the AD converter.

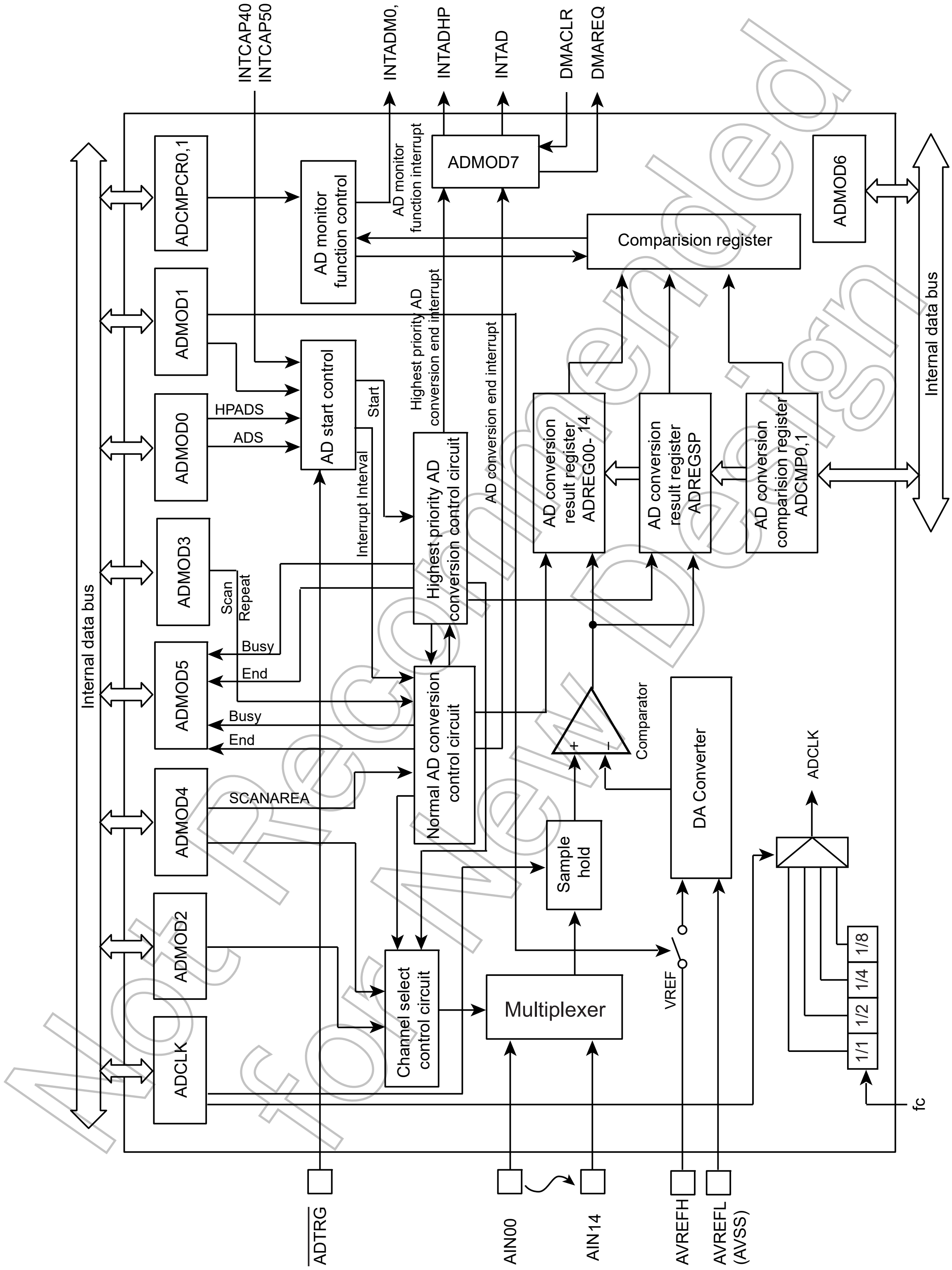

INTCAP40
INTCAP50
INTADM0, 1
INTADHP
INTAD
DMACLR
DMAREQ
ADCMPCR0,1
AD monitor function control
AD monitor function interrupt
ADMOD7
Comparision register
ADMOD6
ADMOD1
ADMOD0
HPADS
ADS
AD start control
Start
Highest priority AD conversion end interrupt
AD conversion end interrupt
Interrupt Interval
Highest priority AD conversion control circuit
AD conversion result register ADREG00- 14
AD conversion result register ADREGSP
AD conversion comparision register ADCMP0,1
Internal data bus
Internal data bus
ADMOD3
Scan
Repeat
Busy
End
ADMOD5
Busy
End
Normal AD conversion control circuit
Comparator
+
−
DA Converter
ADCLK
ADMOD4
SCANAREA
Sample hold
ADMOD2
Channel select control circuit
Multiplexer
VREF
1/8
1/4
1/2
1/1
ADCLK
fc
ADTRG
AIN00
AIN14
AVREFH
AVREFL (AVSS)

Figure 19-1 AD Converter Block Diagram

19.3 Registers

19.3.1 Register list

The control registers and addresses of the AD converter are as follows.

The AD converter is controlled by the AD mode control registers (ADMOD0 through ADMOD7). The result of AD conversion is stored in 15 AD conversion result registers, ADREG00 through ADREG14. The highest-priority conversion result is stored in the register ADREGSP.

Base Address = 0x4005_0000

Register name		Address (Base+)
Conversion Clock Setting Register	ADCLK	0x0000
Mode Control Register 0	ADMOD0	0x0004
Mode Control Register 1	ADMOD1	0x0008
Mode Control Register 2	ADMOD2	0x000C
Mode Control Register 3	ADMOD3	0x0010
Mode Control Register 4	ADMOD4	0x0014
Mode Control Register 5	ADMOD5	0x0018
Mode Control Register 6	ADMOD6	0x001C
Mode Control Register 7	ADMOD7	0x0020
Monitor Function Control Register 0	ADCMPCR0	0x0024
Monitor Function Control Register 1	ADCMPCR1	0x0028
Conversion Result Comparison Register 0	ADCMP0	0x002C
Conversion Result Comparison Register 1	ADCMP1	0x0030
Conversion Result Register 0	ADREG00	0x0034
Conversion Result Register 1	ADREG01	0x0038
Conversion Result Register 2	ADREG02	0x003C
Conversion Result Register 3	ADREG03	0x0040
Conversion Result Register 4	ADREG04	0x0044
Conversion Result Register 5	ADREG05	0x0048
Conversion Result Register 6	ADREG06	0x004C
Conversion Result Register 7	ADREG07	0x0050
Conversion Result Register 8	ADREG08	0x0054
Conversion Result Register 9	ADREG09	0x0058
Conversion Result Register 10	ADREG10	0x005C
Conversion Result Register 11	ADREG11	0x0060
Conversion Result Register 12	ADREG12	0x0064
Conversion Result Register 13	ADREG13	0x0068
Conversion Result Register 14	ADREG14	0x006C
Reserved	-	0x0070
Conversion Result Register SP	ADREGSP	0x0074
Reserved	-	0x0F00
Reserved	-	0x0F04
Reserved	-	0x0F08

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

19.3.2 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	ADSH				-	ADCLK		
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-4	ADSH[3:0]	R/W	Select the AD sample hold time. 0000: 10 × <ADCLK> 0001: 20 × <ADCLK> 0010: 30 × <ADCLK> 0011: 40 × <ADCLK> 0100: 80 × <ADCLK> 0101 to 1111: Reserved
3	−	R	Read as 0.
2-0	ADCLK[2:0]	R/W	Select the AD prescaler clock. 000: fc 001: fc/2 010: fc/4 011: fc/8 100 to 111: Reserved

Note 1: Specify ADCLK in range 4MHz ≤ ADCLK ≤ 40MHz. For example, when fosc = 13.5MHz and PLL = 16 multiplying, fc comes to 54MHz. In such case, set ADCLK<ADCLK[2:0]> to a value other than "000".

Note 2: Do not change the setting of <ADCLK > except when AD conversation is suspended and ADMOD1<VREFON>="0".

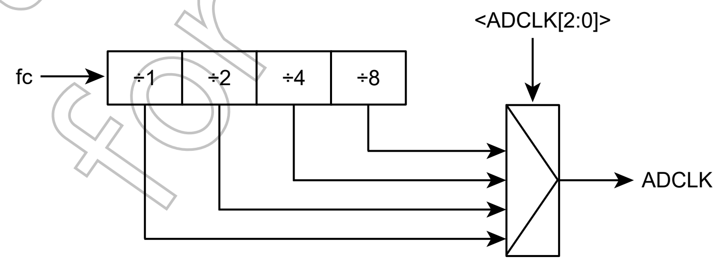


Figure 19-2 AD conversion clock (ADCLK)

A clock count required for conversion is 40 clocks at the minimum.

Examples of sample hold time and conversion time as shown as below.

<ADCLK[2:0]> Setting	<ADSH[3:0]>	Conversion time (Tconv)		
		fc=32MHz	fc=40MHz	fc=54MHz
000 (fc)	10 × ADCLK	1.25 μs	1.00 μs	−
	20 × ADCLK	1.56 μs	1.25 μs	−
	30 × ADCLK	1.88 μs	1.50 μs	−
	40 × ADCLK	2.19 μs	1.75 μs	−
	80 × ADCLK	3.44 μs	2.75 μs	−
001 (fc/2)	10 × ADCLK	2.50 μs	2.00 μs	1.48 μs
	20 × ADCLK	3.13 μs	2.50 μs	1.85 μs
	30 × ADCLK	3.75 μs	3.00 μs	2.22 μs
	40 × ADCLK	4.38 μs	3.50 μs	2.59 μs
	80 × ADCLK	6.88 μs	5.50 μs	4.07 μs
010 (fc/4)	10 × ADCLK	5.00 μs	4.00 μs	2.96 μs
	20 × ADCLK	6.25 μs	5.00 μs	3.70 μs
	30 × ADCLK	7.50 μs	6.00 μs	4.44 μs
	40 × ADCLK	8.75 μs	7.00 μs	5.19 μs
	80 × ADCLK	−	−	8.15 μs
011 (fc/8)	10 × ADCLK	10.0 μs	8.00 μs	5.93 μs
	20 × ADCLK	−	10.0 μs	7.41 μs
	30 × ADCLK	−	−	8.89 μs
	40 × ADCLK	−	−	−
	80 × ADCLK	−	−	−

Note 1: Do not change the setting of the AD conversion clock during AD conversion.

Note 2: Setting the element indicated by "−" in the above table is prohibited. Specify <ADCLK> setting in the 1μs to 10μs range.

19.3.3 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	-	-	-	-	-	-	HPADS	ADS
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	HPADS	W	Activate highest-priority AD conversion 0: Don't care 1: Start conversion "0" is always read.
0	ADS	W	Activate normal (software) AD conversion 0: Don't care 1: Start conversion "0" is always read.

Note 1: In use ADC, write "1" to ADMOD1<VREFON> first, and then start AD conversion or external trigger by setting ADMOD0<ADS> or <HPADS>.

Note 2: When both highest-priority AD conversion <HPADS> and normal AD conversion (software) are enabled and they are selected as $\overline{\text{ADTRG}}$ (external trigger input), highest-priority AD conversion is activated as a priority and normal AD conversion is not activated.

19.3.4 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	RCUT	-	HPADHWS	HPADHWE	ADHWS	ADHWE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	VREFON	R/W	VREF application control (Note1 and Note2) 0: OFF 1: ON
6	I2AD	R/W	Specify operation mode in IDLE mode 0: Stop 1: Operation
5	RCUT	R/W	Control AVREFH-AVREFL reference current 0: Apply the current only in conversion. 1: Apply the current at any time except in RESET
4	−	R	Read as 0.
3	HPADHWS	R/W	Select hardware activation source of highest-priority AD conversion 0: External trigger 1: TMRB4 capture interrupt (INTCAP40)
2	HPADHWE	R/W	Activate highest-priority AD conversion triggered by hardware factors (TMRB4 capture interrupt INTCAP40) 0: Disable 1: Enable
1	ADHWS	R/W	Select hardware activation source of normal AD conversion (Note3) 0: External trigger 1: TMRB5 capture interrupt (INTCAP50)
0	ADHWE	R/W	Activate normal AD conversion triggered by hardware factors (TMRB5 capture interrupt INTCAP50) 0: Disable 1: Enable

Note 1: In use AD conversion, write "1" to the ADMOD1<VREFON> bit, wait for 3μs during which time the internal reference voltage should stabilize, and then start AD conversion or external trigger by setting ADMOD0<ADS> or <HPADS> to "1".

Note 2: Set <VREFON> to "0" to go into standby mode upon completion of AD conversion.

Note 3: The external trigger cannot be used for H/W activation of normal AD conversion when it is used for H/W activation of highest-priority AD conversion.

Note: If it is necessary to reduce a power current with IDLE or STOP1/2 mode and if either case shown below is applicable, you must first stop the AD converter and then execute the instruction to put into low power consumption mode.

1. In the case of putting into IDLE mode with ADMOD1<I2AD> = "0".
2. In the case of putting into STOP1/STOP2 mode.

19.3.5 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	HPADCH				ADCH			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-4	HPADCH[3:0]	R/W	Select analog input channels in highest-priority AD conversion. "1111" is prohibited. (See Table 19-1.)
3-0	ADCH[3:0]	R/W	Select analog input channels in normal AD conversion. "1111" is prohibited. (See Table 19-1.)

Table 19-1 Selection of input channels in normal AD conversion or highest-priority AD conversion

<HPADCH[3:0]>	Analog input channels in highest-priority AD conversion	<ADCH[3:0]>	Analog input channels in normal AD conversion
0000	AIN00	0000	AIN00
0001	AIN01	0001	AIN01
0010	AIN02	0010	AIN02
0011	AIN03	0011	AIN03
0100	AIN04	0100	AIN04
0101	AIN05	0101	AIN05
0110	AIN06	0110	AIN06
0111	AIN07	0111	AIN07
1000	AIN08	1000	AIN08
1001	AIN09	1001	AIN09
1010	AIN10	1010	AIN10
1011	AIN11	1011	AIN11
1100	AIN12	1100	AIN12
1101	AIN13	1101	AIN13
1110	AIN14	1110	AIN14

19.3.6 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	-	ITM			-	-	REPEAT	SCAN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-4	ITM[2:0]	R/W	Specify interrupt in fixed channel repeat conversion mode. See Table 19-2.
3-2	−	R	Read as 0.
1	REPEAT	R/W	Specify repeat mode 0 : Single conversion mode 1 : Repeat conversion mode
0	SCAN	R/W	Specify scan mode 0 : Fixed channel mode 1 : Channel scan mode

Table 19-2 AD conversion interrupt specification in fixed channel repeat conversion mode

<ITM[2:0]>	Fixed channel repeat conversion mode <SCAN> = "0", <REPEAT> = "1"
000	Generate in interrupt once every single conversion.
001	Generate interrupt once every 2 conversions.
010	Generate interrupt once every 3 conversions.
011	Generate interrupt once every 4 conversions.
100	Generate interrupt once every 5 conversions.
101	Generate interrupt once every 6 conversions.
110	Generate interrupt once every 7 conversions.
111	Generate interrupt once every 8 conversions.

Note 1: <ITM[2:0]> is valid only when it's specified in the fixed channel repeat mode, <REPEAT>="1" and <SCAN>="0".

Note 2: When repeat conversion is aborted during repeat conversion (in <REPEAT>=1, fixed channel mode or channel scan mode), <REPEAT> is "0" cleared. In such case, do not change the setting except <REPEAT> bit.

19.3.7 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	SCANAREA				SCANSTA			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-4	SCANAREA [3:0]	R/W	Range of channel scan. ("1111" is prohibited.)
3-0	SCANSTA[3:0]	R/W	Select the start channel to be scanned. ("1111" is prohibited.)

To specify channel scan single mode, set ADMOD3<SCAN> to "1" and <REPEAT> to "0". And, to specify channel scan repeat mode, set ADMOD3<SCAN> to "1" and <REPEAT> to "1".

At first, select the start channel to be scanned. Then select the number of channels to be scanned, starting on the specified start channel.

For example, when ADMOD4<SCANSTA> is set to "0001"(AIN01) and <SCANAREA> is set to "0010" (3ch scan), three channels from AIN01 to AIN03 are scanned.

The following shows the range of assignable value to <SCANAREA> in relation to setting of <SCANSTA>.

Table 19-3 The range of assignable channel scan value

<SCANSTA[3:0]>	The start channel to be scanned	<SCANAREA[3:0]>	The range of assignable channel scan value
0000	(AIN00)	0000 to 1110	(1ch to 15ch)
0001	(AIN01)	0000 to 1101	(1ch to 14ch)
0010	(AIN02)	0000 to 1100	(1ch to 13ch)
0011	(AIN03)	0000 to 1011	(1ch to 12ch)
0100	(AIN04)	0000 to 1010	(1ch to 11ch)
0101	(AIN05)	0000 to 1001	(1ch to 10ch)
0110	(AIN06)	0000 to 1000	(1ch to 9ch)
0111	(AIN07)	0000 to 0111	(1ch to 8ch)
1000	(AIN08)	0000 to 0110	(1ch to 7ch)
1001	(AIN09)	0000 to 0101	(1ch to 6ch)
1010	(AIN10)	0000 to 0100	(1ch to 5ch)
1011	(AIN11)	0000 to 0011	(1chto 4ch)
1100	(AIN12)	0000 to 0010	(1ch to 3ch)
1101	(AIN13)	0000 to 0001	(1ch to 2ch)
1110	(AIN14)	0000	(1ch)

Note: In case of a setting other than listed above, AD conversion is not activated even if ADMOD0 register is set to activate AD conversion.

19.3.8 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	-	-	-	-	HPEOCF	HPADBF	EOCF	ADBF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as 0.
3	HPEOCF	R	Highest-priority AD conversion completion flag (Note1) 0: Before or during conversion 1: Completion
2	HPADBF	R	Highest-priority AD conversion BUSY flag 0: During conversion halts 1: During conversion
1	EOCF	R	Normal AD conversion completion flag (Note1) 0: Before or during conversion 1: Completion
0	ADBF	R	Normal AD conversion BUSY flag 0: During conversion halts 1: During conversion

Note 1: Both <EOCF> and <HPEOCF> flags are "0" cleared by reading the ADMOD5 register.

Note 2: If it is necessary to reduce a power current with IDLE or STOP1/2 mode and if either case shown below is applicable, you must first stop the AD converter and then execute the instruction to put into low power consumption mode.

1. In the case of putting into IDLE mode with ADMOD1<I2AD> = "0".
2. In the case of putting into STOP1/STOP2 mode.

19.3.9 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	-	-	-	-	-	-	ADRST	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	ADRST[1:0]	W	Overwriting 10 with 01 allows ADC to be software reset. A software reset initializes all the registers except for ADCLK<ADCLK>.

Note 1: When DMA transmission is executed by using AD conversion completion interrupt, software reset ADMOD6 <ADRST> first, and then operate DMAC(DMA request standby state) and configure (activate) the ADC.

Note 2: When executing the software reset, the bit of ADMOD1<VREFON> would be "1" is a valid.

Note 3: Initialization takes 3μs in case of the software reset.

19.3.10 ADMOD7 (Mode Control Register7)

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

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as 0.
3-2	−	R/W	Always write "0".
1	INTADHPDMA	R/W	Specify Highest-priority AD conversion DMA activation factor. (Triggered by INTADHP) 0 : Disable 1 : Enable
0	INTADDMA	RW	Specify normal AD conversion DMA activation factor. (Triggered by INTAD 0 : Disable 1 : Enable

19.3.11 ADCMPCR0 (Monitor Control Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	CMPCNT0			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP0EN	-	CMPCOND0	ADBIG0	AINS0			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	−	R	Read as "0".
11-8	CMPCNT0[3:0]	R/W	Number of comparison until the judgment is confirmed. An interrupt is generated when the number of the counts is achieved. 0000 : 1 time count 0110 : 7 times count 1100 : 13 times count 0001 : 2 times count 0111 : 8 times count 1101 : 14 times count 0010 : 3 times count 1000 : 9 times count 1110 : 15 times count 0011 : 4 times count 1001 : 10 times count 1111 : 16 times count 0100 : 5 times count 1010 : 11 times count 0101 : 6 times count 1011 : 12 times count
7	CMP0EN	R/W	AD monitor function 0 0: Disable 1: Enable Setting the condition <CMP0EN>="0" (disabled) clears the number of counts.
6	−	R	Read as "0".
5	CMPCOND0	R/W	Set the count condition for determination. 0: Continuous method 1: Cumulative method In the continuous method, if conditions set to <ADBIG0> goes on and the counting number reaches the number set to <CMPCNT0>, an AD monitor function interrupt (INTADM0) is generated. Even after reaching the counting number, AD monitor function interrupts (INTADM0) keep being generated as long as the condition matches. If the condition is different from that set to <ADBIG0>, the value of the counter is cleared. In the cumulative method, AD monitor function interrupts (INTADM0) are generated and clears the count numbers when the condition set to <ADBIG0> piles up and reaches the counting number set to <CMPCNT0>. Even if a value is different from that is set to <ADBIG0>, the value of the counter is held..
4	ADBIG0	R/W	Set the determination for small and large. 0: Larger than comparison register 1: Smaller than comparison register Every time when the AD conversion set to <AINS0[3:0]> is completed, compare the size of conversion results. If the result matches the settings of <ADBIG0>, the counter is incremented.
3-0	AINS0[3:0]	R/W	Set analog inputs as a target for comparison. 0000 : AIN00 0101 : AIN05 1010 : AIN10 1111 : Do not set this value 0001 : AIN01 0110 : AIN06 1011 : AIN11 0010 : AIN02 0111 : AIN07 1100 : AIN12 0011 : AIN03 1000 : AIN08 1101 : AIN13 0100 : AIN04 1001 : AIN09 1110 : AIN14

Note: AD monitor function is used the fixed repeat mode and the scan repeat mode.

19.3.12 ADCMPCR1 (AD Monitor Control Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	CMPCNT1			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP1EN	-	CMPCOND1	ADBIG1	AINS1			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	−	R	Read as 0.
11-8	CMPCNT1[3:0]	R/W	Number of comparison until the judgment is confirmed. An interrupt is generated when the number of the counts is achieved. 0000 : 1 time count 0110 : 7 times count 1100 : 13 times count 0001 : 2 times count 0111 : 8 times count 1101 : 14 times count 0010 : 3 times count 1000 : 9 times count 1110 : 15 times count 0011 : 4 times count 1001 : 10 times count 1111 : 16 times count 0100 : 5 times count 1010 : 11 times count 0101 : 6 times count 1011 : 12 times count
7	CMP1EN	R/W	AD monitor function 1 0: Disable 1: Enable Setting the condition <CMP1EN>="0" (disabled) clears the number of counts.
6	−	R	Read as 0.
5	CMPCOND1	R/W	Set the count condition for determination. 0: Continuous method 1: Cumulative method In the continuous method, if conditions set to <ADBIG1> goes on and the counting number reaches the number set to <CMPCNT1>, an AD monitor function interrupt (INTADM1) is generated. Even after reaching the counting number, AD monitor function interrupts (INTADM1) keep being generated as long as the condition matches. If the condition is different from that set to <ADBIG1>, the value of the counter is cleared. In the cumulative method, AD monitor function interrupts (INTADM1) are generated and clears the count numbers when the condition set to <ADBIG1> piles up and reaches the counting number set to <CMPCNT1>. Even if a value is different from that is set to <ADBIG1>, the value of the counter is held.
4	ADBIG1	R/W	Set the determination for small and large. 0: Larger than comparison register 1: Smaller than comparison register Every time when the AD conversion set to <AINS1[3:0]> is completed, compare the size of conversion results. If the result matches the settings of <ADBIG1>, the counter is incremented.
3-0	AINS1[3:0]	R/W	Set analog inputs as a target for comparison. 0000 : AIN00 0101 : AIN05 1010 : AIN10 1111 : Do not set this value 0001 : AIN01 0110 : AIN06 1011 : AIN11 0010 : AIN02 0111 : AIN07 1100 : AIN12 0011 : AIN03 1000 : AIN08 1101 : AIN13 0100 : AIN04 1001 : AIN09 1110 : AIN14

Note: AD monitor function is used the fixed repeat mode and the scan repeat mode.

19.3.13 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	-	-	-	-	AD0CMP			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AD0CMP							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	−	R	Read as 0.
11-0	AD0CMP[11:0]	R/W	Sets a value to be compared with the value of the conversion result register

Note: To write values into this register, the AD monitor function 0 must be disabled (ADCMPCR0<CMP0EN> ="0").

19.3.14 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	-	-	-	-	AD1CMP			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AD1CMP							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	−	R	Read as 0.
11-0	AD1CMP[11:0]	R/W	Sets a value to be compared with the value of the conversion result register

Note: To write values into this register, the AD monitor function 1 must be disabled (ADCMPCR1<CMP1EN> ="0").

19.3.15 ADREG00 to ADREG14 (Normal Conversion Result Register 00 to 14)

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

Bit	Bit Symbol	Type	Function
31-15	−	R	Read as 0.
14	ADPOSWF	R	The output switching flag of AIN port. 0: Without switching 1: With switching When PxDATA register of general input-output port which is also used as AIN changes during AD conversion, the port output switching flag, <ADPOSWF>, is set to "1". In this case, when PxCR register corresponding to the changed bit is "1", there is a possibility that the output switching during AD conversion affects conversion accuracy. This bit is "0" cleared when registers, ADREG00 through ADREG14, are read.
13	ADOVRF	R	Overrun flag 0: Not generated. 1: Generated. If the conversion result is overwritten before reading (ADREGx), this bit is set to "1". This bit is "0" cleared when ADREGx registers are read.
12	ADRF	R	AD conversion result storage flag 0: Conversion result is not stored 1: Conversion result is stored. If the conversion result is stored, this bit is set to "1". This bit is "0" cleared when the conversion result of ADREGx registers are read.
11-0	ADR[11:0]	R	12-bits AD conversion result Conversion result is stored. The previous conversion result is read while in AD conversion.

Note: Do not do the output switching during AD conversion, when other analog / input-output ports are used as output port.

19.3.16 ADREGSP (Highest-priority Conversion Result Register)

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

Bit	Bit Symbol	Type	Function
31-15	−	R	Read as 0.
14	ADPOSWFSP	R	The output switching flag of AIN port. 0: Without switching 1: With switching When PxDATA register of general input-output port which is also used as AIN changes during AD conversion, the port output switching flag, <ADPOSWFSP>, is set to "1". In this case, when PxCR register corresponding to the changed bit is "1", there is a possibility that the output switching during AD conversion affects conversion accuracy. This bit is "0" cleared when registers, ADREGx is read.
13	ADOVRFSP	R	Overrun flag 0: Not generated 1: Generated If the highest-priority AD conversion result is overwritten before reading <ADRSP>, "1" is set. This bit is "0" cleared when ADREGSP register is read.
12	ADRFSP	R	Highest-priority AD conversion result storage flag 0: Conversion result is not stored. 1: Conversion result is stored. If the highest-priority conversion result is stored, this bit is set to "1". This bit is "0" cleared when ADREGSP conversion result is read.
11-0	ADRSP[11:0]	R	Highest-priority 12-bits AD conversion result Highest-priority conversion result is stored. The previous conversion result is read while in AD conversion.

Note: Do not do the output switching during AD conversion, when other analog / input-output ports are used as output port.

19.4 Description of Operations

19.4.1 Analog Reference Voltage

The "High" level of the analog reference voltage shall be applied to the AVRFEH pin, and the "Low" shall be applied to the AVREFL pin.

To start AD conversion, make sure that you first write "1" to the <VREFON> bit, wait for 3 μs during which time the internal reference voltage should stabilize, and then write "1" to the ADMOD0<ADS> bit.

By writing "0" to the ADMOD1<RCUT> bit, a switched-on state of AVREFH − AVREFL can be turned into a switched -off state.

19.4.2 AD Conversion Mode

Two types of AD conversion are supported: normal AD conversion and highest-priority AD conversion.

19.4.2.1 Normal AD Conversion

For normal AD conversion, the following four operation modes are supported and the operation mode is selected with the ADMOD3<REPEAT, SCAN>.

- Fixed channel single conversion mode
- Channel scan single conversion mode
- Fixed channel repeat conversion mode
- Channel scan repeat conversion mode

(1) Fixed channel single conversion mode

If ADMOD3<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 by ADMOD2 <ADCH>. After AD conversion is completed, ADMOD5<EOCF> is set to "1", ADMOD5<ADBF> is cleared to "0", and the AD conversion completion interrupt request (INTAD) is generated. <EOCF> is cleared to "0" upon read.

(2) Channel scan single conversion mode

If ADMOD3 <REPEAT, SCAN> is set to "01," AD conversion is performed in the channel scan single conversion mode.

In this mode, AD conversion is performed once for the scan channel area selected by ADMOD4 <SCANAREA> from the start channel selected by ADMOD4 <SCANSTA>. After AD scan conversion is completed, ADMOD5<EOCF> is set to "1", ADMOD5<ADBF> is cleared to "0", and the conversion completion interrupt request (INTAD) is generated. <EOCF> is cleared to "0" upon read.

(3) Fixed channel repeat conversion mode

If ADMOD3<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 by ADMOD2 <ADCH>. After AD conversion is completed, ADMOD5<EOCF> is set to "1". ADMOD5<ADBF> 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 ADMOD3<ITM> to an appropriate setting. <EOCF> is set with the same timing as this interrupt INTAD is generated. <EOCF> is cleared to "0" upon read.

(4) Channel scan repeat conversion mode

If ADMOD3<REPEAT, SCAN> is set to "11", AD conversion is performed in the channel scan repeat conversion mode.

In this mode, AD conversion is performed repeatedly for the scan channel area selected by ADMOD4 <SCANAREA> from the start channel selected by ADMOD4 <SCANSTA>. Each time one AD scan conversion is completed, ADMOD5 <EOCF> is set to "1", and the conversion completion interrupt request (INTAD) is generated. ADMOD5 <ADBF> is not cleared to "0" and remains at "1". <EOCF> is cleared to "0" upon read.

19.4.2.2 Highest-priority AD conversion

By interrupting ongoing normal AD conversion, highest-priority AD conversion can be performed.

The fixed-channel single conversion is automatically selected, irrespective of the ADMOD3 <REPEAT><SCAN> setting. When conditions to start operation are met, a conversion is performed just once for a channel selected by ADMOD2<HPADCH>. When conversion is completed, the highest-priority AD conversion completion interrupt (INTADHP) is generated, and ADMOD5 <HPEOCF> showing the completion of AD conversion is set to "1". <HPADBF> returns to "0". <HPEOCF> flag is cleared to "0" upon read.

Highest-priority AD conversion activated while highest-priority AD conversion is under way is ignored.

19.4.3 AD Monitor Function

This is a function for setting the channel fixed repeat mode and the scan repeat mode.

Setting "1" to both ADCMPCR0<CMP0EN> and ADCMPCR1<CMP1EN> enables the AD monitor function. The monitor function can also be enabled for both registers at the same time.

Here is an example, taking the ADCMPCR0.

Configure the following settings: analog input as a target for comparison to the <AINS0[3:0]> of the ADCMPCR0 register, large/small determination to the <ADBIG0>, counting conditions for determination to the <CMPCOND0> and the number of counts in determination to the <CMPCNT0[3:0]>.

Once AD conversion starts, every time when one single conversion completes, large/small determination is performed. If the result of the conversion matches the settings stored in the <ADBIG0>, increment the judgment counter.

There are two conditions of judgment: continuous method and cumulative method. In the continuous method, if conditions set to the <ADBIG0> goes on and the counting number reaches the numbers set to <CMPCNT0 [3:0]>, an AD monitor interrupt (INTADM0) is generated and clears the counter value. Even if the values are different from that is set to the <ADBIG0>, the value of the counter is cleared. In the cumulative method, AD monitor function interrupts (INTADM0) are generated when conditions set to the <ADBIG0> pile up and reach the counting number set to the <CMPCNT0[3:0]>. The counter values are held even the condition is different from that is set to the <ADBIG0>. If values of the conversion result storage register set to the ADCMPCR0 are the same as the values of a register as a comparison target, the count is not incremented. AD monitor function interrupt (INTADM0) is not generated. (neither ADCMPCR1 is)

This comparison is performed every time when a result is stored to the conversion result storage register, and an interrupt (INTADM0) occurs when a condition matches. Since the storage register used as AD monitor function is not usually read using software, registers correspond to overrun flags, from ADREG00 to 14<ADOVRF>, and conversion result storage flags, from ADREG00 to 14<ADRF>, are always set to "1". Thus, do not use those flags of the conversion result storage register when you use the AD monitor function.

1. Set the AIN00 input as fixed channel repeat conversion. Compare values of the AD conversion result storage register (0x0888) to judge.
   - ADMOD3=0x0002： fixed channel repeat conversion Note: Disable the AD conversion completion interrupt (INTAD).
   - ADCMPCR0 =0x0280： Target channel for comparison: AIN00, large/small determination: larger than the comparison register, count condition for determination: continuous method, AD monitor function: enabled, counting number of large/small determination: three counts
   - ADCMP0=0x0888： AD conversion result storage register (compared value 0x0888)

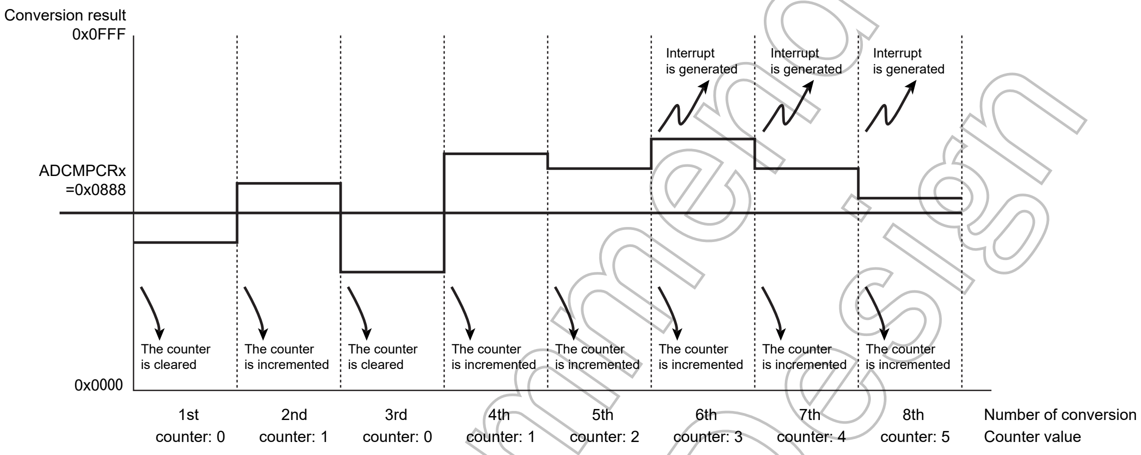


Figure 19-3 AD monitor function (Fixed channel repeat and Continuous method)

2. Set AIN00 input as fixed channel repeat conversion. Compare values of the AD conversion result comparison register (0x0888).
   - ADMOD3=0x0002： fixed channel repeat conversion Note: AD conversion completion interrupt (INTAD) is disabled.
   - ADCMPCR0 =0x02A0： target channel for comparison: AIN00, large/small determination: larger than the comparison register, count condition for determination: cumulative method, AD monitor function: enabled, counting number of large/small determination: three counts.
   - ADCMP0=0x0888: AD conversion result comparison register (comparison value 0x0888)

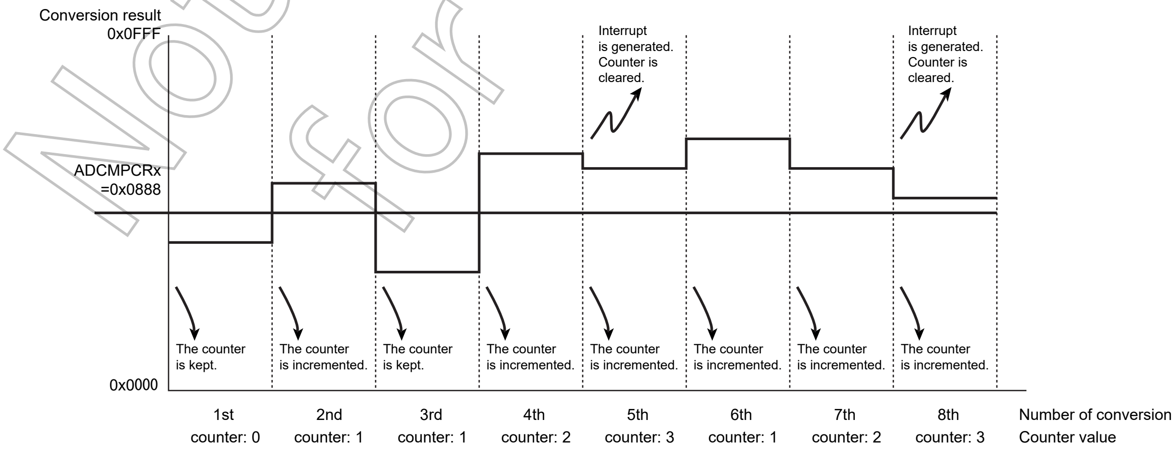


Figure 19-4 AD monitor function (Fixed channel repeat and Cumulative method)

19.4.4 Selecting the Input Channel

After reset, ADMOD3 <REPEAT, SCAN> is initialized to "00" and ADMOD2 <ADCH[3:0]> is initialized to "0000".

The channels to be converted are selected according to the operation mode of the AD converter as shown below.

1. Normal AD conversion mode
   - If the analog input channel is used in a fixed state (ADMOD3<SCAN> = "0")

     One channel is selected from analog input pins AIN00 through AIN14 by setting ADMOD2 <ADCH> to an appropriate setting

   - If the analog input channel is used in a scan state (ADMOD3<SCAN> = "1")

     The channel to be started can be specified by setting ADMOD4 <SCANSTA>. And, the number of channels to be scanned can be specified by setting ADMOD4 <SCANAREA>.

2. Highest-priority AD conversion mode

   One channel is selected from analog input pins from AIN00 through AIN14 by setting ADMOD2<HPADCH> to an appropriate setting. If highest-priority AD conversion has been activated during normal AD conversion, ongoing normal AD conversion is suspended, and restarts normal AD conversion after highest-priority AD conversion is completed.

19.4.5 AD Conversion Details

19.4.5.1 Starting AD Conversion

Two types of AD conversion are supported: normal AD conversion and top-priority AD conversion. Normal AD conversion is activated by setting ADMOD0<ADS> to "1". Highest-priority AD conversion is activated by setting ADMOD0<HPADS> 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 ADMOD3 <REPEAT><SCAN> to an appropriate setting. For highest-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 ADMOD1<ADHWS>, and highest-priority AD conversion can be activated using the HW activation source selected by ADMOD1<HPADHWS>. If bits of <ADHWS> and <HPADHWS> are "0", normal and highest-priority AD conversions are activated in response to the input of a falling edge through the $\overline{\text{ADTRG}}$ pin. If these bits are "1", normal AD conversion is activated in response to capture interrupt of TMRB5 (INTCAP50), and highest-priority AD conversion is activated in response to capture interrupt of TMRB4 (NITCAP40).

To permit H/W activation, set ADMOD1 <ADHWE> to "1" for normal AD conversion and set ADMOD1<HPADHWE> to "1" for highest-priority AD conversion.

Software activation is still valid even after H/W activation has been permitted.

Note: When an external trigger is used for the HW activation source of a highest-priority AD conversion, an external trigger cannot be set for activating normal AD conversion H/W start.

19.4.5.2 AD Conversion

When normal AD conversion starts, the AD conversion Busy flag (ADMOD5 <ADBF>) showing that AD conversion is under way is set to "1".

When highest-priority AD conversion starts, the highest-priority AD conversion Busy flag (ADMOD5 <HPADBF>) showing that AD conversion is under way is set to "1".

At that time, the value of the Busy flag ADMOD5<ADBF> for normal AD conversion before the start of highest-priority AD conversion is retained.

The value of the conversion completion flag ADMOD5 <EOCF> for normal AD conversion before the start of highest-priority AD conversion is retained.

Note: Normal AD conversion must not be activated when highest-priority AD conversion is under way. If activated when highest-priority AD conversion is under way, the highest-priority AD conversion completion flag cannot be set, and the flag for previous normal AD conversion cannot be cleared.

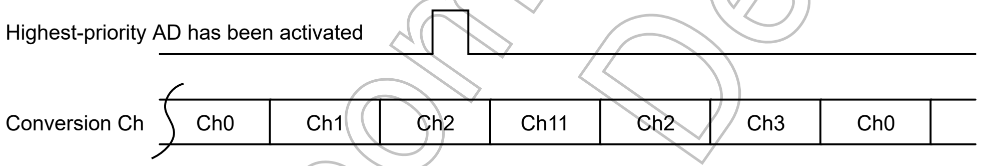
19.4.5.3 Highest-priority AD conversion requests during normal AD conversion

If highest-priority AD conversion has been activated during normal AD conversion, ongoing normal AD conversion is suspended, and restarts normal AD conversion after highest-priority AD conversion is completed.

If ADMOD0<HPADS> is set to "1" during normal AD conversion, ongoing normal AD conversion is suspended, and the highest-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 highest-priority AD conversion is stored in the storage register ADREGSP, normal AD conversion is resumed.

If H/W activation of highest-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 highest-priority AD conversion (fixed-channel single conversion) starts for a channel designated by ADMOD2<HPADCH>. After the result of this highest-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 AIN00 through AIN03 and if <HPADS> is set to "1" during AIN02 conversion, AIN02 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 AIN02.



19.4.5.4 Stopping Repeat Conversion Mode

To stop the AD conversion operation in the repeat conversion mode (fixed-channel repeat conversion mode or channel scan conversion mode), write "0" to ADMOD3<REPEAT>. When ongoing AD conversion is completed, the repeat conversion mode terminates, and ADMOD5<ADBF> is set to "0".

19.4.5.5 Reactivating normal AD conversion

If ADMOD0 <ADS> is set to "1" during normal AD conversion, normal AD conversion is reactivated. Ongoing normal AD conversion is suspended at the time that it is reactivated. At that time, the normal AD conversion Busy flag ADMOD5 <ADBF>, the normal AD conversion completion flag ADMOD5 <EOCF> and the storage result flag ADREGm <ADOVRF>, <ADRF> are cleared to "0". (m=00-14)

If H/W activation of normal 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. Ongoing normal AD conversion is suspended at the time that it is reactivated. At that time, the normal AD conversion Busy flag ADMOD5 <ADBF>, the normal AD conversion completion flag ADMOD5 <EOCF> and the storage result flag ADREGm <ADOVRF>, <ADRF> are cleared to "0". (m=00-14)

19.4.5.6 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 ADMOD5<EOCF> which indicates the completion of AD conversion and the register ADMOD5<ADBF>. The timing that interrupt request is generated and the timing that conversion result register <EOCF> <ADBF> changes vary according to a mode selected.

In mode other than fixed-channel repeat conversion mode, conversion results are stored in AD conversion result registers (ADREG00 through ADREG14) corresponding to a channel.

In fixed-channel repeat conversion mode, the conversion results are sequentially stored in storage registers ADREG00 through ADREG14. However, if interrupt setting on <ITM> is set to be generated each time one AD conversion is completed, the conversion result is stored only in ADREG00. If interrupt setting on <ITM> is set to be generated each time 8 AD conversions are completed, the conversion results are sequentially stored in ADREG00 through ADREG07.

Interrupt requests, flag changes and conversion result registers in each mode are as shown below.

- Fixed-channel single conversion mode

  After AD conversion is completed, ADMOD5 <EOCF> is set to "1", ADMOD5 <ADBF> is cleared to "0", and the interrupt request INTAD is generated.

  Conversion results are stored a conversion result register correspond to a channel.

- Channel scan single conversion mode

  After the channel scan conversion is completed, ADMOD5 <EOCF> is set to "1", ADMOD5 <ADBF> is cleared to "0", and the interrupt request INTAD is generated.

  Conversion results are stored a conversion result register correspond to a channel.

- Fixed-channel repeat conversion mode

  ADMOD5 <ADBF> is not cleared to "0". It remains at "1". The timing with which the interrupt request INTAD is generated can be selected by setting ADMOD3<ITM> to an appropriate setting. ADMOD5 <EOCF> is set with the same timing as this interrupt INTAD is generated.

  a. One conversion

     With ADMOD2 <ADCH[3:0]> set to "0000" (AIN00) and ADMOD3 <ITM[2:0]> set to "000", an interrupt request is generated each time one AD conversion is completed. In this case, the conversion results are always stored in the storage register ADREG00. After the conversion result is stored, <EOCF> is set to "1".

  b. 8 conversions

With ADMOD2 <ADCH[3:0]> set to "1110" (AIN14) and ADMOD3 <ITM[2:0]> set to "111", an interrupt request is generated each time 8 AD conversions are completed. In this case, the conversion results are sequentially stored in the storage register ADREG00 through ADREG07. After the conversion result is stored in ADREG07, <EOCF> is set to "1", and the storage of subsequent conversion results starts from ADREG00.

- Channel scan repeat conversion mode

  Each time one AD conversion is completed, ADMOD5<EOCF> is set to "1" and an interrupt request INTAD is generated. ADMOD5<ADBF> is not cleared to "0". It remains at "1".

  If ADMOD4 <SCANSTA[3:0]> is set to "0001" (AIN01) and ADMOD4 <SCANAREA [7:4]> is set to "1110" (14Ch scan), each time one AD conversion is completed, ADMOD5 <EOCF> is set to "1" and an interrupt request INTAD is generated. ADMOD5 <ADBF> is not cleared to "0" and remains at "1".

  AD conversion results are stored in a AD conversion result register corresponding to a channel.

(2) Highest-priority AD conversion completion

After the highest-priority AD conversion is completed, the highest-priority AD conversion completion interrupt (INTADHP) is generated, and ADMOD5<HPEOCF> which indicates the completion of highest-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, ADMOD5 <EOCF> 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 word access. If <ADOVRF> = "0", <ADRF> = "1" and <ADPOSWF> = "0", a correct conversion result has been obtained.

(4) DMA request

After the normal AD conversion completion interrupt (INTAD) or the highest-priority AD conversion completion interrupt (INTADHP) is generated, DMA request is issued. DMA request after any interrupt is generated can be set to "disable" or "enable" by setting ADMOD7 register to an appropriate setting. A DMA request is issued in 2 system clocks (fsys) after AD conversion completion interrupt (INTAD or INTADHP) is generated.

19.4.5.7 Interrupt generation timings and AD conversion result storage register

Table 19-4 shows a relation in the following three items: AD conversion modes, interrupt generation timings and flag operations. Table 19-5 shows a relation between analog channel inputs and AD conversion result registers.

Table 19-4 Relations in conversion modes, interrupt generation timings and flag operations

Conversion mode		Scan/repeat mode setting (ADMOD3)			Interrupt generation timing	(ADMOD5)		
		<REPEAT>	<SCAN>	<ITM[2:0]>		<EOCF>/<HPEOCF> set timing (Note1)	<ADBF> (After the interrupt is generated)	<ADBFHP> (After the interrupt is generated)
Normal conversion	Fixed-channel single conversion	0	0	−	After generation is completed.	After generation is completed.	0	−
	Fixed-channel repeat conversion	1	0	000	Each time one conversion is completed.	After one conversion is completed.	1	−
				001	Each time 2 conversion is completed.	After 2 conversions are completed.	1	−
				010	Each time 3 conversion is completed.	After 3 conversions are completed.	1	−
				011	Each time 4 conversion is completed.	After 4 conversions are completed.	1	−
				100	Each time 5 conversion is completed.	After 5 conversions are completed.	1	−
				101	Each time 6 conversion is completed.	After 6 conversions are completed.	1	−
				110	Each time 7 conversion is completed.	After 7 conversions are completed.	1	−
				111	Each time 8 conversion is completed.	After 8 conversions are completed.	1	−
	Channel scan single conversion	0	1	−	After scan conversion is completed.	After scan conversion is completed.	0	−
	Channel scan repeat conversion	1	1	−	After one scan conversion is completed.	After one scan conversion is completed.	1	−
Highest-priority conversion		−	−	−	After generation is completed.	Conversion completion	−	0

Note 1: ADMOD5 <EOCF> and <HPEOCF> are cleared upon read.

Note: In repeat mode, ADMOD5 <ADBF> is not cleared to "0" even if any interrupt is generated. To suspend the repeat operation, ADMOD5 <ADBF> is cleared to "0" after ADMOD3 <REPEAT> is written "0" and AD conversion is completed.

Table 19-5 Relations between analog channel inputs and AD conversion result registers

Fixed-channel single mode	
Channel	Storage register
AIN00	ADREG00
AIN01	ADREG01
AIN02	ADREG02
AIN03	ADREG03
AIN04	ADREG04
AIN05	ADREG05
AIN06	ADREG06
AIN07	ADREG07
AIN08	ADREG08
AIN09	ADREG09
AIN10	ADREG10
AIN11	ADREG11
AIN12	ADREG12
AIN13	ADREG13
AIN14	ADREG14

Fixed-channel repeat mode		
ADMOD3<ITM[2:0]>		Storage register
000	Interrupt by each time AD/C	ADREG00
001	Interrupt by each time 2 AD/C	ADREG00 to ADREG01
010	Interrupt by each time 3 AD/C	ADREG00 to ADREG02
011	Interrupt by each time 4 AD/C	ADREG00 to ADREG03
100	Interrupt by each time 5 AD/C	ADREG00 to ADREG04
101	Interrupt by each time 6 AD/C	ADREG00 to ADREG05
110	Interrupt by each time 7 AD/C	ADREG00 to ADREG06
111	Interrupt by each time 8 AD/C	ADREG00 to ADREG07

Channel scan single mode / repeat mode (ex. ADREG03 to depend on the scan channel range.)		
ADMOD4<SCANSTA> (Starts channel)	ADMOD4<SCANAREA> (Scan channel range)	Storage register
AIN00	15 channels	ADREG00 to ADRE14
AIN01	14 channels	ADREG01 to ADRE14
AIN02	13 channels	ADREG02 to ADRE14
AIN03	12 channels	ADREG03 to ADRE14
AIN04	11 channels	ADREG04 to ADRE14
AIN05	10 channels	ADREG05 to ADRE14
AIN06	9 channels	ADREG06 to ADRE14
AIN07	8 channels	ADREG07 to ADRE14
AIN08	7 channels	ADREG08 to ADRE14
AIN09	6channels	ADREG09 to ADRE14
AIN10	5 channels	ADREG10 to ADRE14
AIN11	4 channels	ADREG11 to ADRE14
AIN12	3 channels	ADREG12 to ADRE14
AIN13	2 channels	ADREG13 to ADRE14
AIN14	1 channels	ADREG14

Note: When the range of channel scan is set to out of the assignable value in channel scan mode, the AD conversion can not be activated even if ADMOD0 is set to activate AD conversion.

Notes on designing for AD converter inputs

<An output impedance of the external signal source which is connected with AIN pin>

An output impedance of the external signal source which is connected with AIN pin is equal or less than R_{EXAIN} shown below formula.

- Calculating formula of allowable value of output impedance of the external signal source -

The maxmum value of an output impedance connected with AIN pin : $R_{EXAIN} <$ Tscyc ÷ (ADCLK × C_{ADC} × ln (2^{14})) − R_{AIN}

MCU information	Symbol	Min	Typ	Max	Unit
ADC clock frequency	ADCLK	4	-	40	MHz
Total AIN input capacity in MCU	C_{ADC}	-	-	12.2	pF
AIN resistance in MCU	R_{AIN}	-	-	1	kΩ
Cycle number in the sample hold period	Tscyc	10	-	80	Cycle

R_{EXAIN} maximum value list (ADCLK = 40MHz)

Tcsyc	R_{EXAIN}	Unit
10	1.1	kΩ
20	3.2	kΩ
30	5.3	kΩ
40	7.5	kΩ
80	15.9	kΩ

< Addition of stabilizing capacity >

If high-speed AD conversion is required and the sample hold period cannot meet the conditions of calculating formula of allowable values of output impedance of external signal source, add stabilizing capacity to the AIN pin. The additional capacity depends on external circuitd. Although the capacity depended on the external circuit is different from the each board set, add the capacity from about 0.1μF to 1μF, appropriate amount for your circuit board.

Set the capacity to be added next to the AIN pin.

< Adjustment of sample hold period>

Generally, by setting the sample hold period long, you can make the input voltage of the comparator in the ADC circuit as same as the input voltage of the AIN pin can reduce the error of an AD conversion.

Although, in case that the sample hold period is too long, the error of an AD conversion may be increased because the voltage held in sample hold circuit is changed.

Because the suitable sample hold period is depended on the each board set, please decided the suitable sample hold period on your board set.

Notes of the use of the AD converter

The result value of AD conversion may vary depending on the fluctuation of the supply voltage, or may be affected by noise.

When using analog input pins and ports alternately, do not read and write ports during conversion because the conversion accuracy may be reduced. Also the conversion accuracy may be reduced if the output ports current fluctuate during AD conversion.

Please take counteractive measures with the program such as averaging the AD conversion results.

19.4.5.8 The way of stopping ADC in the low power consumption mode

When ADC is stopped in the low power consumption mode, it is stopped by the following order.

1. ADC is stopped in the state of ADMOD1<VREFON>="1".
2. ADCLK<ADCLK[2:0]> is set to "010" (ADCLK = fc/4) or "100" (ADCLK=fc/8).
3. Make ADC initialize by a software reset (ADMOD6<ADRST[1:0]>="10" -> "01").

4. The CPU enters in the low power consumption mode.

20. Digital/Analog Converter (DAC)

TMPM341FDXBG/FYXBG contains a digital/analog converter (DAC) with 2 channels.

20.1 Features

- A high-resolution, 10-bit D/A converter is built in.
- Built in full range buffer amplifier.
- Built in power down function.

20.2 Block Diagram

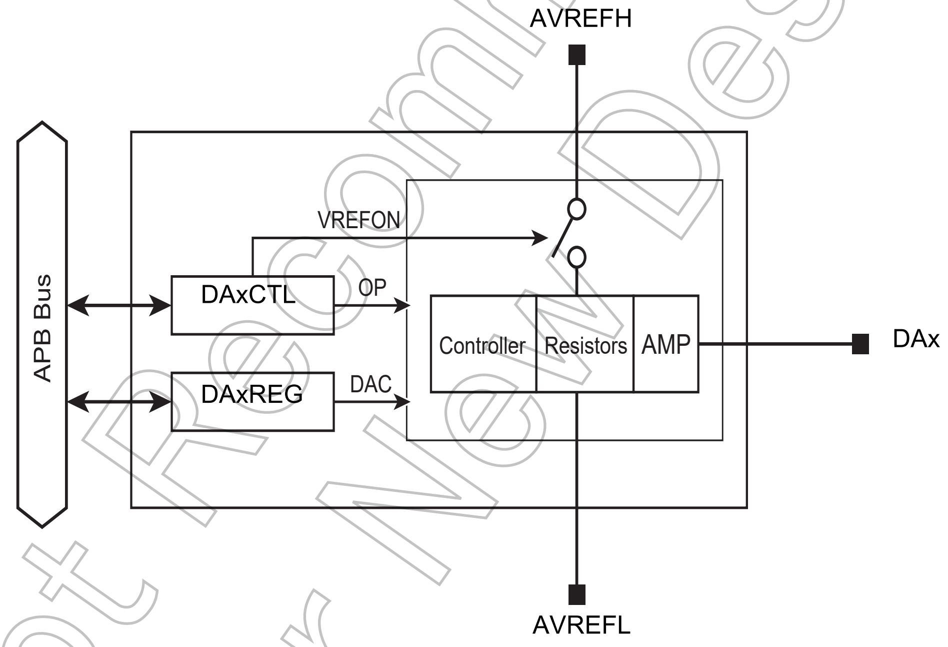


Figure 20-1 DAC Block Diagram

20.3 Registers

20.3.1 Register List

Channel x	Base Address
Channel0	0x4005_4000
Channel1	0x4005_5000

Register name (x=0~1)		Address(Base+)
Control register	DAxCTL	0x0000
Output register	DAxREG	0x0004
Reserved	−	0x0010
Reserved	−	0x0030

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

20.3.2 DAxCTL(Control register)

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

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	VREFON	R/W	VREF control 0: VREF off 1: VREF on When the setting value is "1", connect an external pin of AVREFH to D/A converter circuit.
0	OP	R/W	DAC operation control 0: Disable 1: Enable When setting value is "1", DAx pin is output the voltage that specified by the output register DAxREG. When setting value is "0", the operation stops and the output pin DAx goes into High-Z.

20.3.3 DAxREG(Output register)

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

Bit	Bit Symbol	Type	Function
31-10	−	R	Read as 0.
9-0	DAC[9:0]	R/W	Conversion value setting Configures analog voltage output level for outputting to output pin. The calculation is follow. DAx = DAC × (AVREFH - AVREFL) / 1024

20.4 Operational Description

20.4.1 Setting method

Set the control register DAxCTL<OP><VREFON> to "11" and write an output CODE to the output register DAxREG so that an output voltage specified for that CODE generates at the output pin DAx.

20.4.2 Low power consumption mode

By setting DAxCTL<VREFON> to "0", internal VREF can be cut and so VREF level equals to AVREFL and a consumption current can be reduced. In the same way, by setting DAxCTL<OP> to "0", the DA conversion stops and DAx output goes into High-Z, thus the amplifier is power-down.

When transition to low power consumption mode, set DAxCTL<OP><VREFON>="00" first and execute the instruction to enter the standby mode.

21. Flash Memory Operation

This section describes the hardware configuration and operation of the flash memory.

21.1 Flash Memory

21.1.1 Features

1. Memory capacity

   The TMPM341FDXBG/FYXBG devices contain flash memory. The memory sizes and configurations of each device are shown in the table below.

2. Write/erase time

   Writing is executed per page. The TMPM341FDXBG contains 128 words and the TMPM341FYXBG contains 64 words in a page.

   Page writing requires 1.25ms (typical) regardless of number of words.

   A block erase requires 0.1 sec. (typical).

   The following table shows write and erase time per chip.

Product Name	Memory Size	Block Configuration				# of Words	Write Time	Erase Time
		128 KB	64 KB	32 KB	16 KB			
TMPM341FDXBG	512 KB	3	1	2	−	128	1.28 sec	0.4 sec
TMPM341FYXBG	256 KB	−	3	1	2	64		

Note: The above values are theoretical values not including data transfer time.The write time per chip depends on the write method to be used by the user.

3. Programming method

   There are two types of the onboard programming mode for the user to program (rewrite) the device while it is mounted on the user's board:

   ・ The onboard programming mode

   a. User boot mode

      The user's original rewriting method can be supported.

   b. Single boot mode

      The rewriting method to use serial data transfer (Toshiba's unique method) can be supported.

4. Rewriting method

   The flash memory included in this device is generally compliant with the applicable JEDEC standards except for some specific functions. Therefore, if the user is currently using an external flash memory device, it is easy to implement the functions into this device. Furthermore, the user is not required to build his/her own programs to realize complicated write and erase functions because such functions are automatically performed using the circuits already built-in the flash memory chip.

JEDEC compliant functions	Modified, added, or deleted functions
・ Automatic programming ・ Automatic chip erase ・ Automatic block erase ・ Data polling/toggle bit	<Modified> Block protect (only software protection is supported) <Deleted> Erase resume - suspend function

5. Protect/ Security Function

This device is also implemented with a read-protect function to inhibit reading flash memory data from any external writer device. On the other hand, rewrite protection is available only through command-based software programming; any hardware setting method to apply +12VDC is not supported. See the chapter "ROM protection" for details of ROM protection and security function.

21.1.2 Block Diagram of the Flash Memory Section

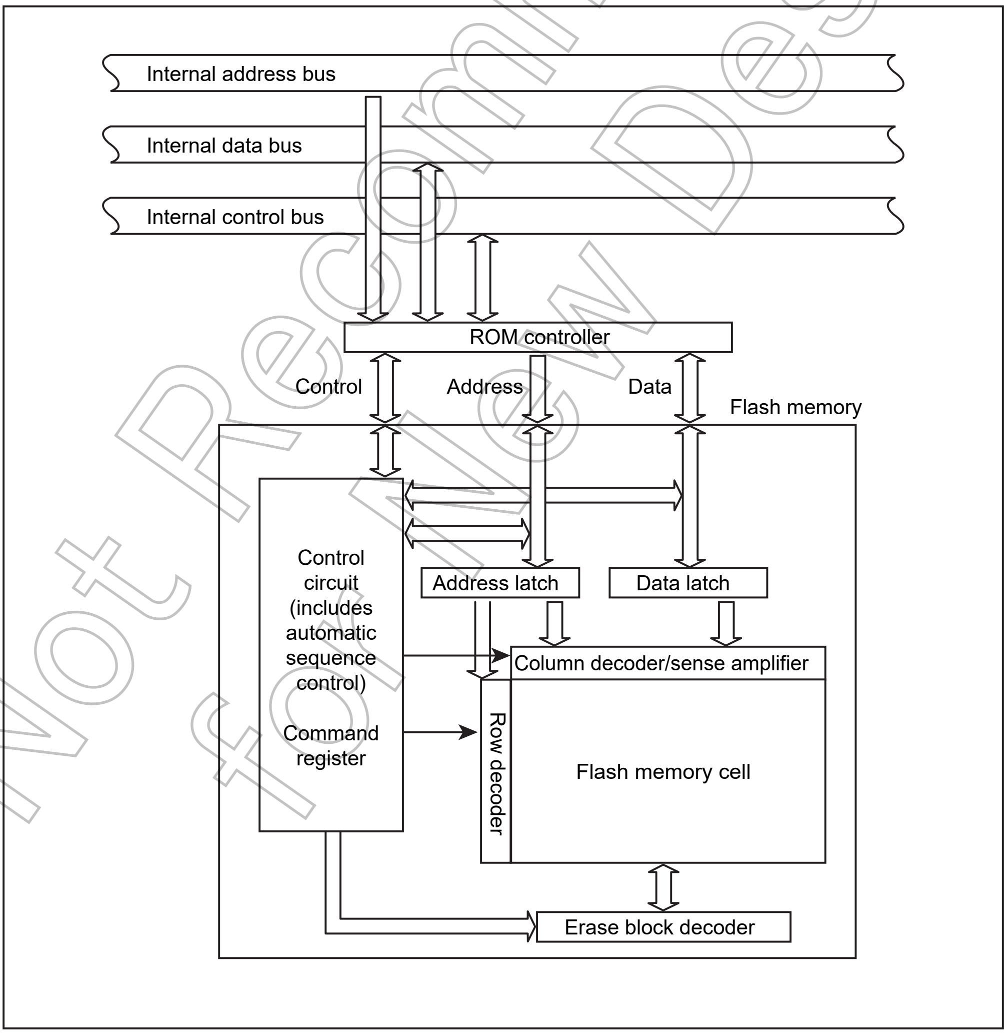


Figure 21-1 Block Diagram of the Flash Memory Section

21.2 Operation Mode

This device has three operation modes including the mode not to use the internal flash memory.

Table 21-1 Operation Modes

Operation mode	Operation details
Single chip mode	After reset is cleared, it starts up from the internal flash memory.
Normal mode	In this operation mode, two different modes, i.e., the mode to execute user application programs and the mode to rewrite the flash memory onboard the user's card, are defined. The former is referred to as "normal mode" and the latter "user boot mode".
User boot mode	The user can uniquely configure the system to switch between these two modes. For example, the user can freely design the system such that the normal mode is selected when the port "A0" is set to "1" and the user boot mode is selected when it is set to "0". The user should prepare a routine as part of the application program to make the decision on the selection of the modes.
Single boot mode	After reset is cleared, it starts up from the internal Boot ROM (Mask ROM). In the Boot ROM, an algorithm to enable flash memory rewriting on the user's set through the serial port of this device is programmed. By connecting to an external host computer through the serial port, the internal flash memory can be programmed by transferring data in accordance with predefined protocols.

Among the flash memory operation modes listed in the above table, the User Boot mode and the Single Boot mode are the programmable modes. These two modes, the User Boot mode and the Single Boot mode, are referred to as "Onboard Programming" modes where onboard rewriting of internal flash memory can be made on the user's card.

Either the Single Chip or Single Boot operation mode can be selected by externally setting the level of the $\overline{\text{BOOT}}$ (PF0) pin while the device is in reset status.

Table 21-2 Operation Mode Setting

Operation mode	Pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$ (PF0)
Single chip mode	0 → 1	1
Single boot mode	0 → 1	0

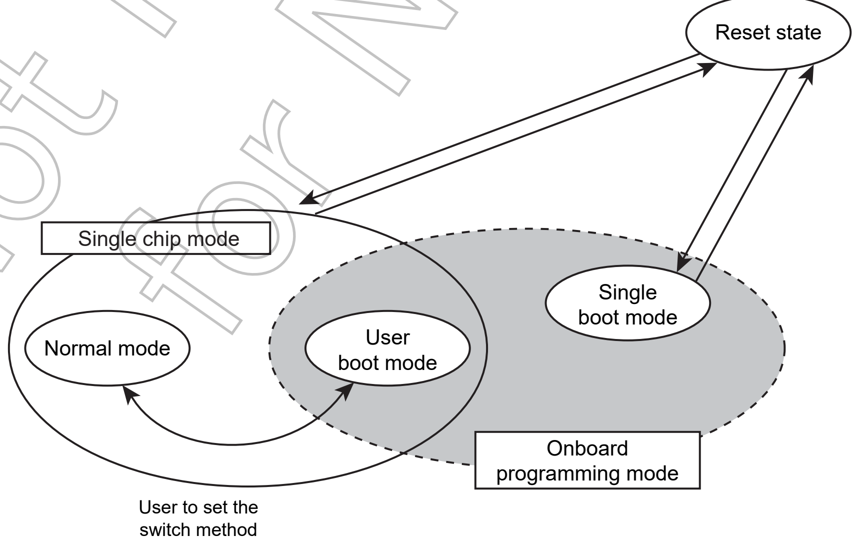


Figure 21-2 Mode Transition Diagram

21.2.1 Reset Operation

To reset the device, ensure that the power supply voltage is within the operating voltage range, that the internal oscillator has been stabilized, and that the $\overline{\text{RESET}}$ input is held at "0" for a minimum duration of 12 system clocks (0.3 µs with 40MHz operation; the "1/1" clock gear mode is applied after reset).

Note 1: **Regarding cold-reset of devices with internal flash memory; for devices with internal flash memory, it is necessary to apply "0" to the $\overline{\text{RESET}}$ inputs upon power on for a minimum duration of 1.0 milli-seconds regardless of the operating frequency.**

Note 2: **While flash automatic programming or deletion is in progress, at least 0.5 microseconds of reset period is required regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.**

21.2.2 User Boot Mode (Single chip mode)

User Boot mode is to use flash memory programming routine defined by users. It is used when the data transfer buses for flash memory program code on the old application and for serial I/O are different. It operates at the single chip mode; therefore, a switch from normal mode in which user application is activated at the single chip mode to User Boot Mode for programming flash is required. Specifically, add a mode judgment routine to a reset program in the old application.

The condition to switch the modes needs to be set by using the I/O of TMPM341FDXBG/FYXBG in conformity with the user's system setup condition. Also, flash memory programming routine that the user uniquely makes up needs to be set in the new application. This routine is used for programming after being switched to User Boot Mode. The execution of the programming routine must take place while it is stored in the area other than the flash memory since the data in the internal flash memory cannot be read out during delete/ writing mode. Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations. Be sure not to cause any exceptions including a non-maskable while User Boot Mode.

(1-A) and (1-B) are the examples of programming with routines in the internal flash memory and in the external memory. For a detailed description of the erase and program sequence, refer to "21.3 On-board Programming of Flash Memory (Rewrite/Erase)".

21.2.2.1 (1-A) Method 1: Storing a Programming Routine in the Flash Memory

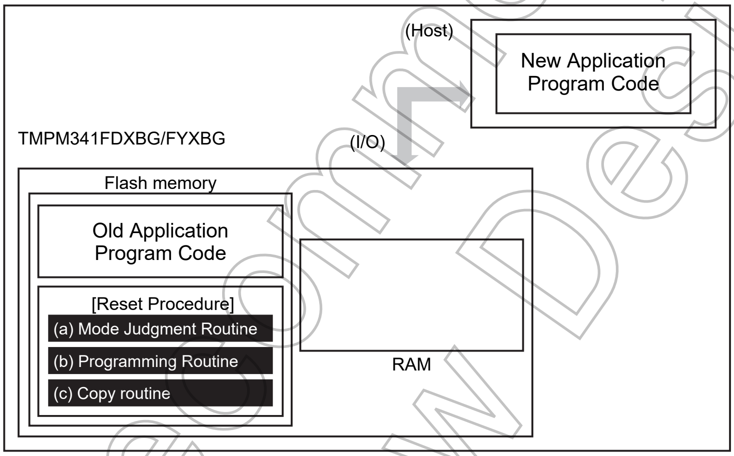
(1) Step-1

Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM341FDXBG/FYXBG on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine: Code to determine whether or not to switch to User Boot mode

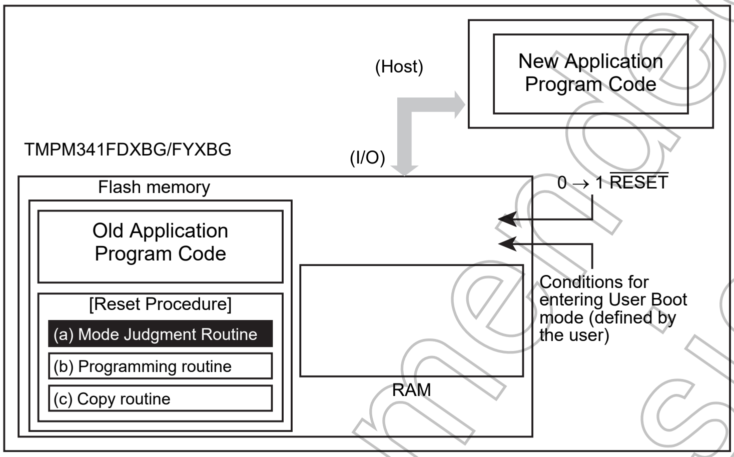
(b) Programming routine: Code to download new program code from a host controller and re-program the flash memory

(c) Copy routine: Code to copy the data described in (b) from the TMPM341FDXBG/FYXBG flash memory to either the TMPM341FDXBG/FYXBG on-chip RAM or external memory device.



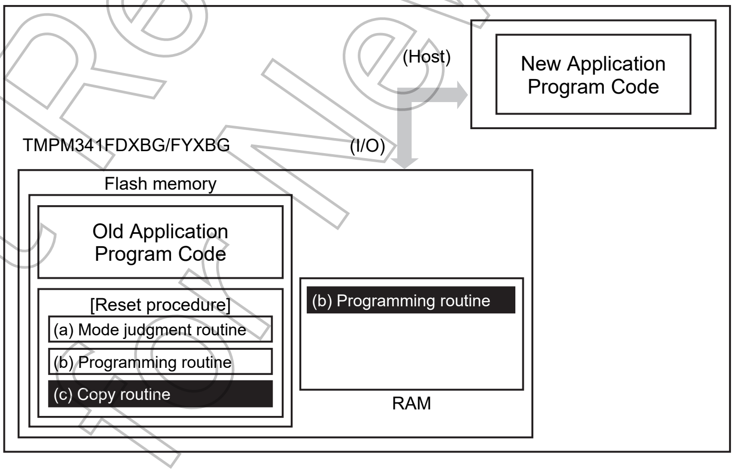
(2) Step-2

After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMPM341FDXBG/FYXBG flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be disabled while in User Boot mode.)


(Host)
New Application Program Code
TMPM341FDXBG/FYXBG
(I/O)
Flash memory
0 → 1 $\overline{\text{RESET}}$
Old Application Program Code
[Reset Procedure]
(a) Mode Judgment Routine
(b) Programming routine
(c) Copy routine
RAM
Conditions for entering User Boot mode (defined by the user)

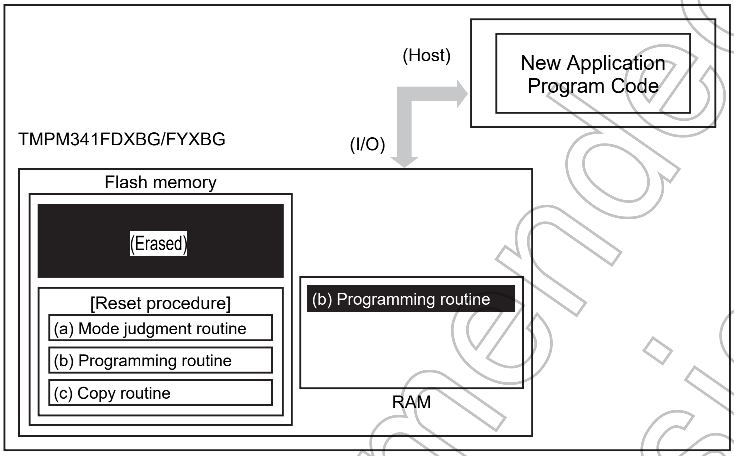
(3) Step-3

Once transition to User Boot mode is occurred, execute the copy routine (c) to copy the flash programming routine (b) to the TMPM341FDXBG/FYXBG on-chip RAM.



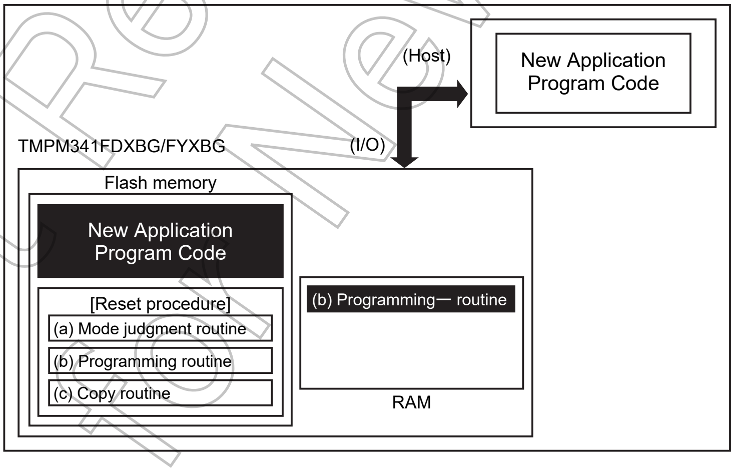
(4) Step-4

Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.



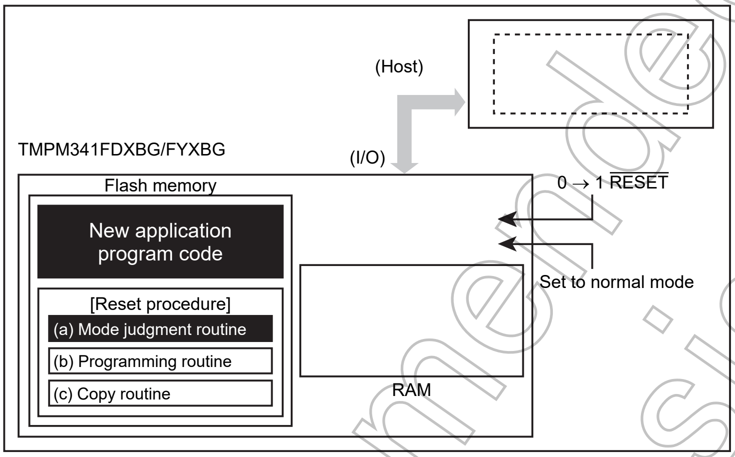
(5) Step-5

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user’s program area must be set.



(6) Step-6

Set $\overline{\text{RESET}}$ to "0" to reset the TMPM341FDXBG/FYXBG. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



21.2.2.2 (1-B) Method 2: Transferring a Programming Routine from an External Host

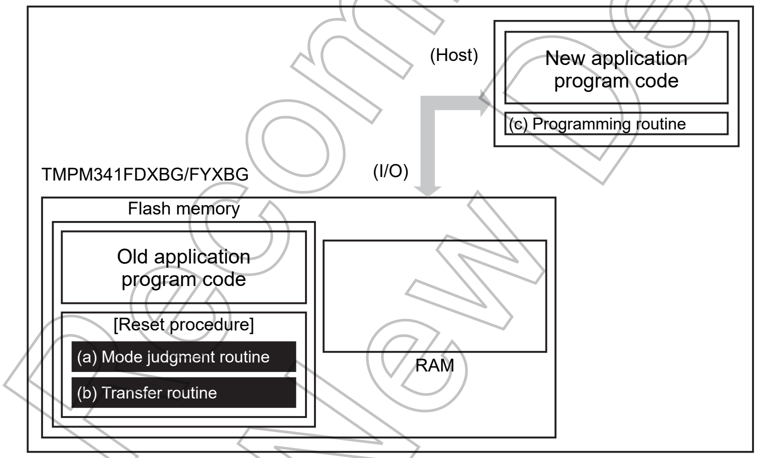
(1) Step-1

Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM341FDXBG/FYXBG on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine:	Code to determine whether or not to switch to User Boot mode
(b) Transfer routine:	Code to download new program code from a host controller

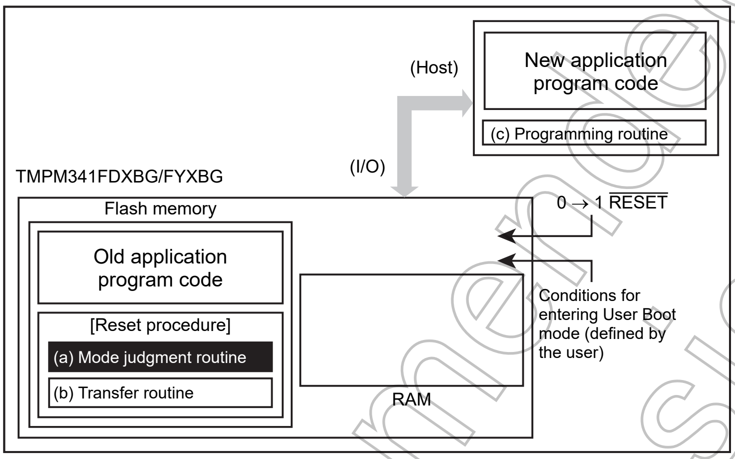
Also, prepare a programming routine shown below on the host controller:

(c) Programming routine:	Code to download new program code from an external host controller and re-program the flash memory



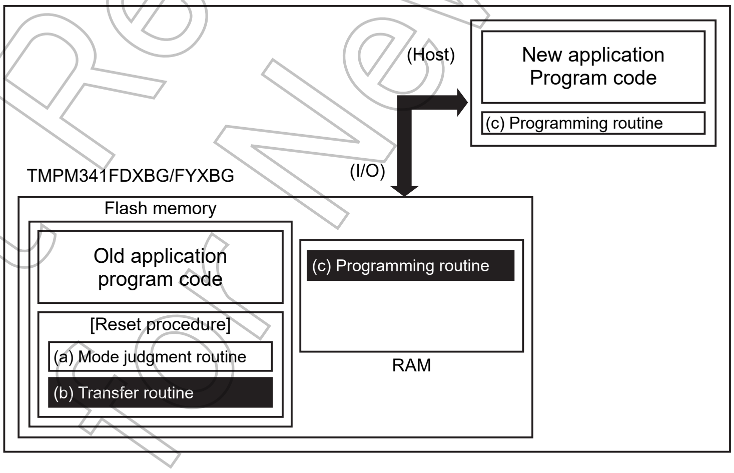
(2) Step-2

After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMPM341FDXBG/FYXBG flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be disabled while in User Boot mode).


(Host)
New application program code
(c) Programming routine
(I/O)
TMPM341FDXBG/FYXBG
Flash memory
0 → 1 $\overline{\text{RESET}}$
Old application program code
[Reset procedure]
(a) Mode judgment routine
(b) Transfer routine
RAM
Conditions for entering User Boot mode (defined by the user)

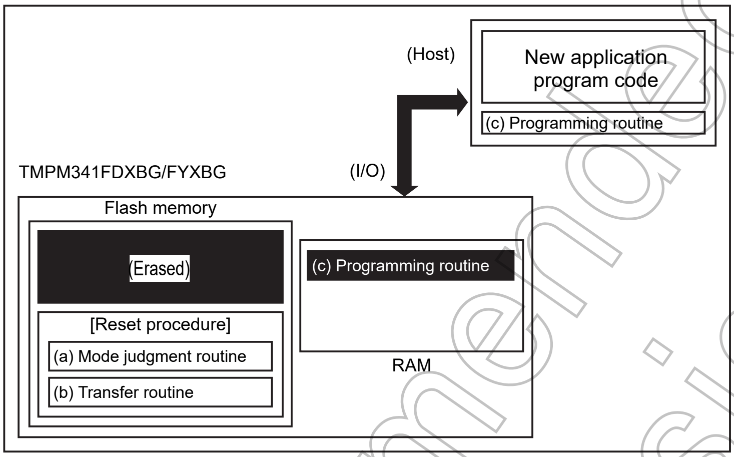
(3) Step-3

Once User Boot mode is entered, execute the transfer routine (b) to download the flash programming routine (c) from the host controller to the TMPM341FDXBG/FYXBG on-chip RAM.



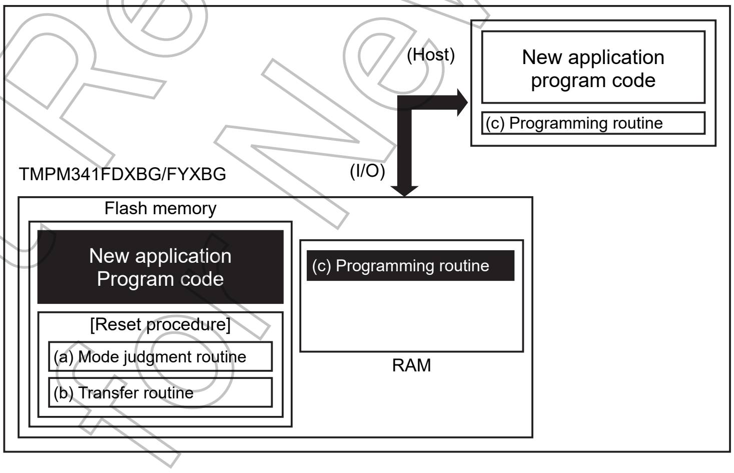
(4) Step-4

Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.



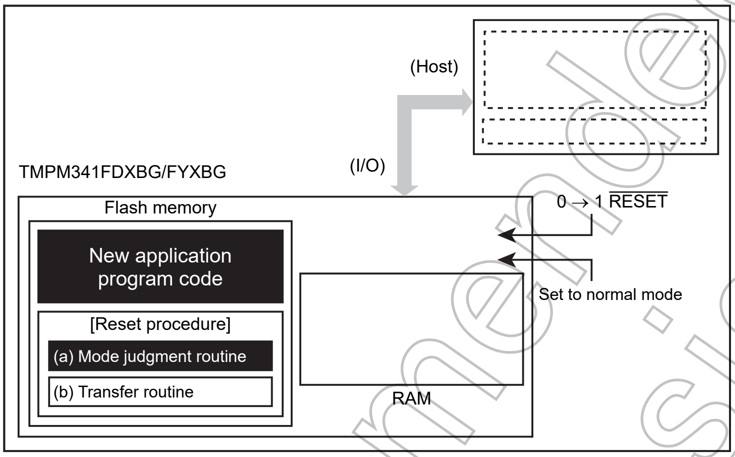
(5) Step-5

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user program area must be set.



(6) Step-6

Set $\overline{\text{RESET}}$ to "0" low to reset the TMPM341FDXBG/FYXBG. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



21.2.3 Single Boot Mode

In Single Boot mode, the flash memory can be re-programmed by using a program contained in the TMPM341FDXBG/FYXBG on-chip boot ROM. This boot ROM is a masked ROM. When Single Boot mode is selected upon reset, the boot ROM is mapped to the address region including the interrupt vector table while the flash memory is mapped to an address region different from it.

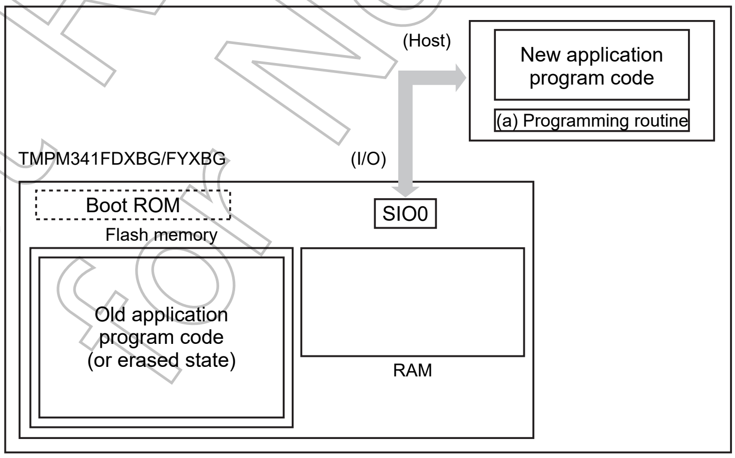
Single Boot mode allows for serial programming of the flash memory. Channel 0 of the SIO (SIO0) of the TMPM341FDXBG/FYXBG is connected to an external host controller. Via this serial link, a programming routine is downloaded from the host controller to the TMPM341FDXBG/FYXBG on-chip RAM. Then, the flash memory is re-programmed by executing the programming routine. The host sends out both commands and programming data to re-program the flash memory. Communications between the SIO0 and the host must follow the protocol described later. To secure the contents of the flash memory, the validity of the application's password is verified before a programming routine is downloaded into the on-chip RAM. If password matching fails, the transfer of a programming routine itself is aborted. As in the case of User Boot mode, all interrupts including the non-maskable interrupt (NMI) must be disabled in Single Boot mode while the flash memory is being erased or programmed. In Single Boot mode, the boot-ROM programs are executed in Normal mode.

Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations.

21.2.3.1 (2-A) Using the Program in the On-Chip Boot ROM

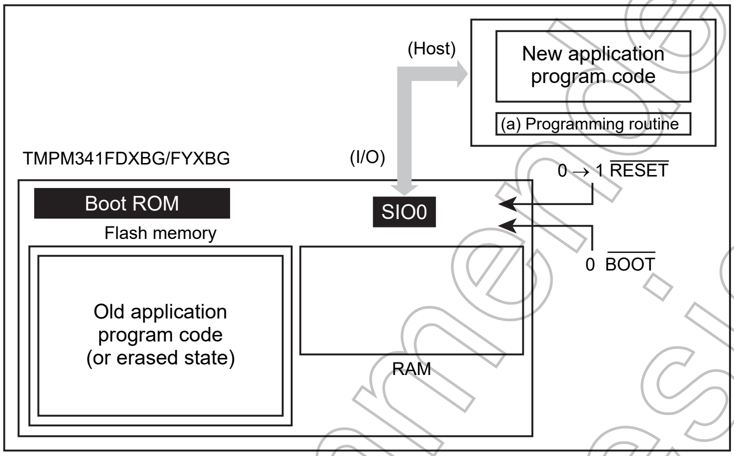
(1) Step-1

The flash block containing the older version of the program code need not be erased before executing the programming routine. Since a programming routine and programming data are transferred via the SIO (SIO0), the SIO0 must be connected to a host controller. Prepare a programming routine (a) on the host controller.



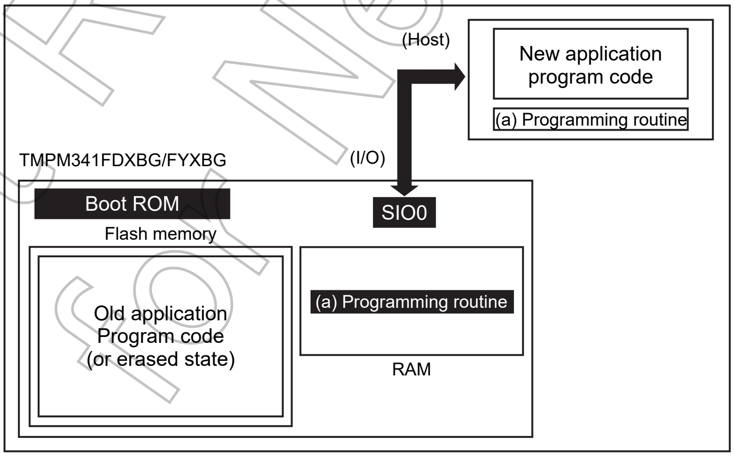
(2) Step-2

Set the $\overline{\text{RESET}}$ pin to "1" to cancel the reset of the TMPM341FDXBG/FYXBG when the $\overline{\text{BOOT}}$ pin has already been set to "0". After reset, CPU reboots from the on-chip boot ROM. The 12-byte password transferred from the host controller via SIO0 is first compared to the contents of the special flash memory locations. (If the flash block has already been erased, the password is 0xFF).



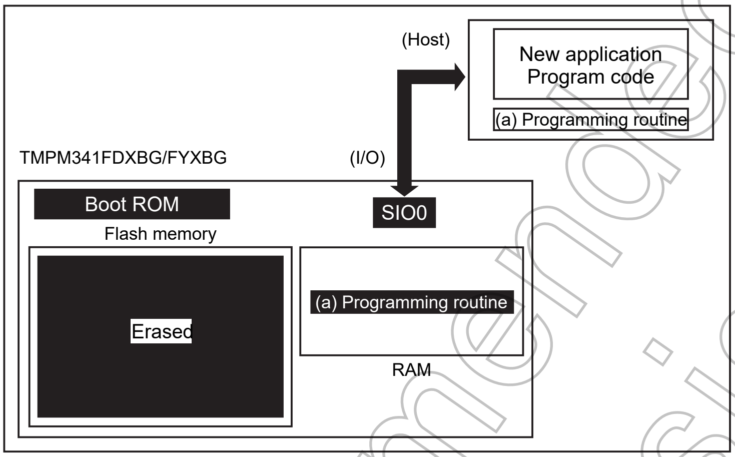
(3) Step-3

If the password was correct, the boot program downloads, via the SIO0, the programming routine (a) from the host controller into the on-chip RAM of the TMPM341FDXBG/FYXBG. The programming routine must be stored in the range from 0x2000_0400 to the end address of RAM.



(4) Step-4

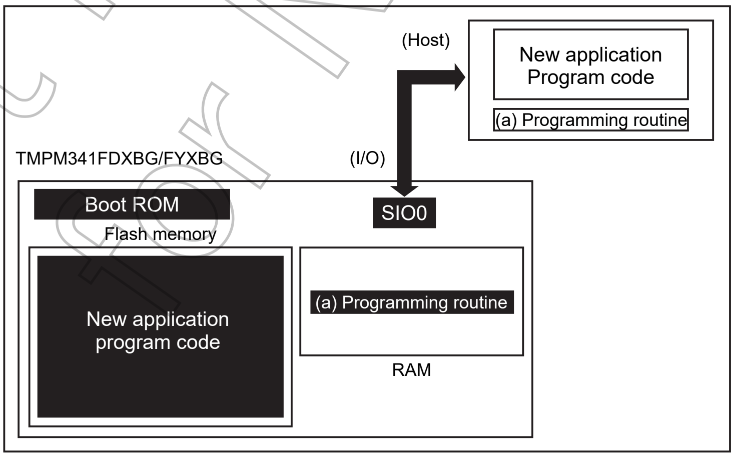
The CPU jumps to the programming routine (a) in the on-chip RAM to erase the flash block containing the old application program code. The Block Erase or Chip Erase command may be used.



(5) Step-5

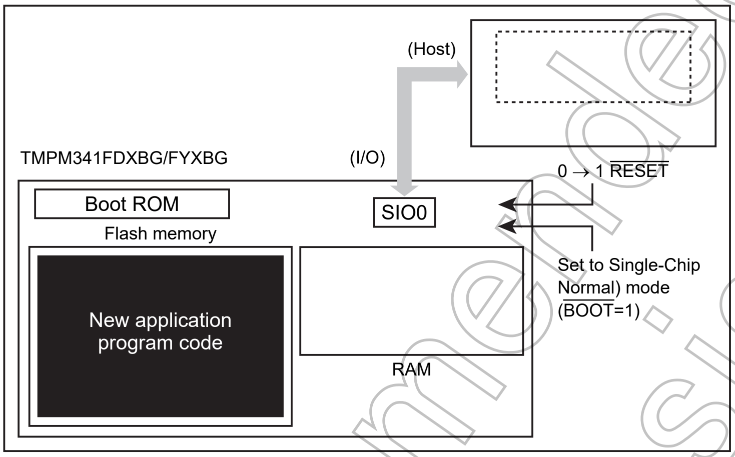
Next, the programming routine (a) downloads new application program code from the host controller and programs it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user's program area must be set.

In the example below, new program code comes from the same host controller via the same SIO0 channel as for the programming routine. However, once the programming routine has begun to execute, it is free to change the transfer path and the source of the transfer. Create board hardware and a programming routine to suit your particular needs.



(6) Step-6

When programming of the flash memory is complete, power off the board and disconnect the cable leading from the host to the target board. Turn on the power again so that the TMPM341FDXBG/FYXBG re-boots in Single-Chip (Normal) mode to execute the new program.


(Host)
TMPM341FDXBG/FYXBG
(I/O)
Boot ROM
SIO0
Flash memory
New application program code
RAM
0 → 1 $\overline{\text{RESET}}$
Set to Single-Chip Normal) mode ($\overline{\text{BOOT}}$=1)

21.2.4 Configuration for Single Boot Mode

To execute the on-board programming, boot the TMPM341FDXBG/FYXBG with Single Boot mode following the configuration shown below.

$\overline{\text{BOOT}}$(PF0) = 0
$\overline{\text{RESET}}$ = 0 → 1

Set the $\overline{\text{RESET}}$ input to "0", and set the each $\overline{\text{BOOT}}$ (PF0) pins to values shown above, and then release RESET (high).

21.2.5 Memory Map

Figure 21-3 and Figure 21-4 show a comparison of the memory maps in Normal and Single Boot modes. In Single Boot mode, the internal flash memory is mapped to 0x3F80_0000 and later addresses, and the Internal boot ROM (Mask ROM) is mapped to 0x0000_0000 through 0x0000_0FFF.

The internal flash memory and RAM addresses of each device are shown below.

Product Name	Flash Size	RAM Size	Flash Address (Single Chip/ Single Boot Mode)	RAM Address
TMPM341FDXBG	512 KB	32 KB	0x0000_0000 to 0x0007_FFFF 0x3F80_0000 to 0x3F87_FFFF	0x2000_0000 to 0x2000_7FFF
TMPM341FYXBG	256 KB		0x0000_0000 to 0x0003_FFFF 0x3F80_0000 to 0x3F83_FFFF	

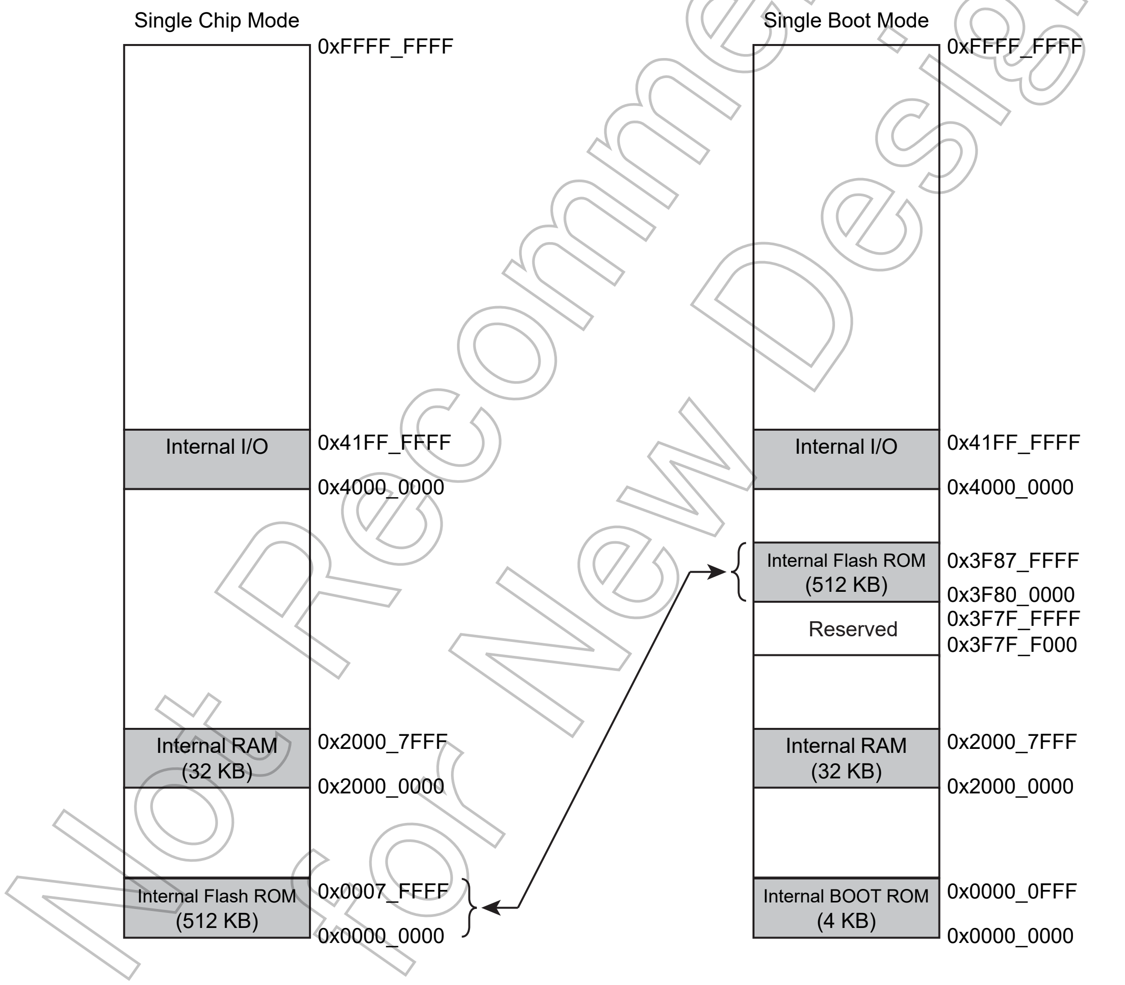


Figure 21-3 Memory Maps for TMPM341FDXBG

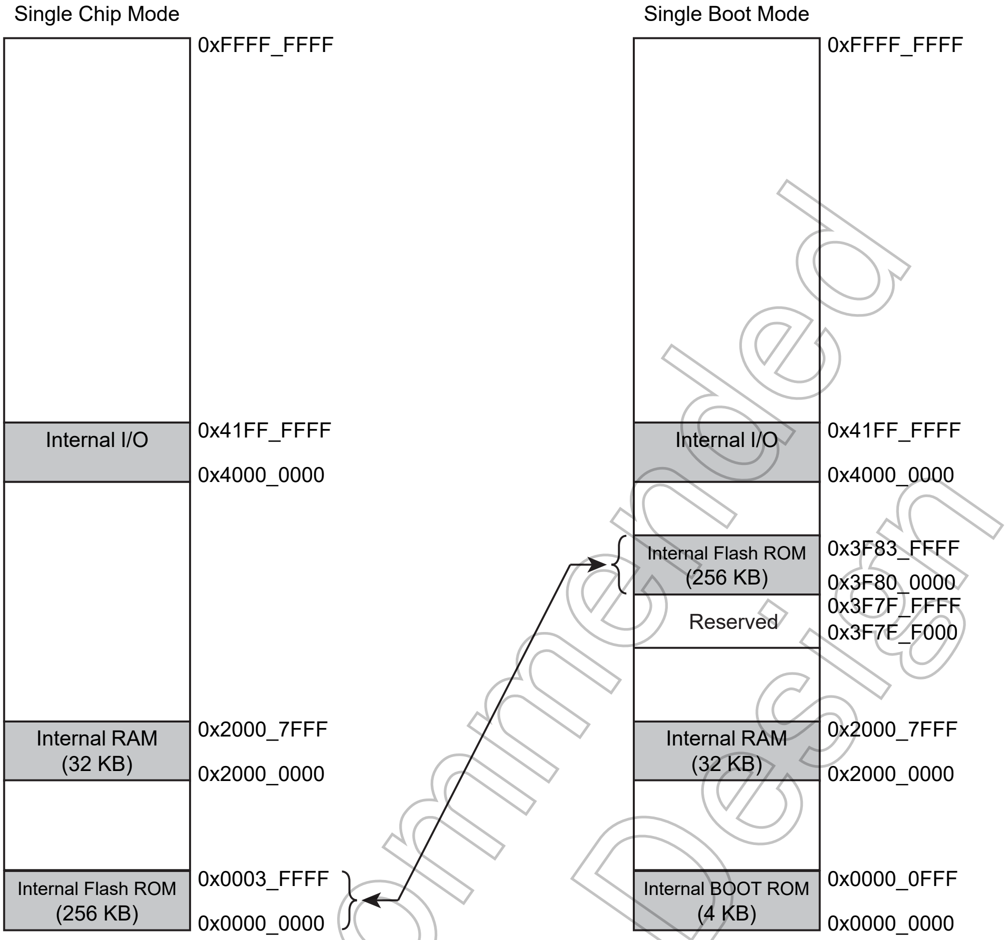


Figure 21-4 Memory Maps for TMPM341FYXBG

21.2.6 Interface specification

In Single Boot mode, an SIO channel is used for communications with a programming controller. The same configuration is applied to a communication format on a programming controller to execute the on-board programming. Both UART (asynchronous) and I/O Interface (synchronous) modes are supported. The communication formats are shown below.

- UART communication

  Communication channel : SIO channel 0

  Serial transfer mode : UART (asynchronous), half -duplex, LSB first

  Data length : 8 bit

  Parity bits : None

  STOP bits : 1 bit

  Baud rate : Arbitrary baud rate
- I/O interface mode

  Communication channel : SIO channel 0

  Serial transfer mode : I/O interface mode, full -duplex, LSB first

  Synchronization clock (SCLK0) : Input mode

  Handshaking signal : PE4 configured as an output mode

  Baud rate : Arbitrary baud rate

Table 21-3 Required Pin Connections

Pins		Interface	
		UART	I/O Interface Mode
Mode-setting pin	MODE	Connect with Pull-down resistance	
	INTLV	Connect with Pull-down resistance	
	FTEST3	fixed to open	
	BSC	Connect to GND	
	ENDIAN	Connect with Pull-down resistance	
	$\overline{\text{BOOT}}$ (PF0)	o	o
Reset pin	$\overline{\text{RESET}}$	o	o
Communication pins	TXD0 (PE0)	o	o
	RXD0 (PE1)	o	o
	SCLK0 (PE2)	×	o (Input mode)
	PE4	×	o (Output mode)

o : used × : unused

21.2.7 Data Transfer Format

Table 21-4 and Table 21-6 to Table 21-7 illustrate the operation commands and data transfer formats at each operation mode. In conjunction with this section, refer to "21.2.10 Operation of Boot Program".

Table 21-4 Single Boot Mode Commands

Code	Command
0x10	RAM transfer
0x40	Chip and protection bit erase

21.2.8 Restrictions on internal memories

Single Boot Mode places restrictions on the internal RAM and ROM as shown in Table 21-5.

Table 21-5 Restrictions in Single Boot Mode

Memory	Details
Internal RAM	A program contained in the BOOT ROM uses the area, through 0x2000_0000 to 0x2000_03FF as a work area. Store the RAM transfer program from 0x2000_0400 through the end address of RAM.
Internal ROM	The following addresses are assigned for storing software ID information and passwords. Storing program in these addresses is not recommendable. TMPM341FDXBG: 0x3F87_FFF0 to 0x3F87_FFFF TMPM341FYXBG: 0x3F83_FFF0 to 0x3F83_FFFF

21.2.9 Transfer Format for Single Boot Mode commands

The following tables shows the transfer format for each Single Boot Mode command. Use this section in conjunction with Chapter "21.2.10 Operation of Boot Program".

21.2.9.1 RAM Transfer

Table 21-6 Transfer Format for the RAM Transfer Command

	Byte	Data Transferred from the Controller to the TMPM341FDXBG/FYXBG	Baud rate	Data Transferred from the TMPM341FDXBG/ FYXBG to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode : 0x86 For I/O Interface mode : 0x30	Desired baud rate (Note 1)	−
	2 byte	−		ACK for the serial operation mode byte ・For UART mode -Normal acknowledge : 0x86 (The boot program aborts if the baud rate can not be set correctly.) ・For I/O Interface mode -Normal acknowledge :0x30
	3 byte	Command code (0x10)		−
	4 byte	−		ACK for the command code byte (Note 2) -Normal acknowledge : 0x10 -Negative acknowledge : 0xX1 -Communication error : 0xX8
	5 byte to 16 byte	Password sequence (12 bytes) TMPM341FDXBG: 0x3F87_FFF4 to 0x3F87_FFFF TMPM341FYXBG: 0x3F83_FFF4 to 0x3F83_FFFF		−
	17 byte	Check SUM value for bytes 5 - 16		−
	18 byte	−		ACK for the checksum byte (Note 2) -Normal acknowledge : 0x10 -Negative acknowledge : 0xX1 -Communication error : 0xX8
	19 byte	RAM storage start address 31 to 24		−
	20 byte	RAM storage start address 23 to 16		−
	21 byte	RAM storage start address 15 to 8		−
	22 byte	RAM storage start address 7 to 0		−
	23 byte	RAM storage byte count 15 to 8		−
	24 byte	RAM storage byte count 7 to 0		−
	25 byte	Check SUM value for bytes 19 to 24		−
	26 byte	−		ACK for the checksum byte (Note 2) -Normal acknowledge : 0x10 -Negative acknowledge : 0xX1 -Communication error : 0xX8
	27 byte to m byte	RAM storage data		−
	m + 1 byte	Checksum value for bytes 27 to m		−
	m + 2 byte	−		ACK for the checksum byte (Note 2) -Normal acknowledge : 0x10 -Negative acknowledge : 0xX1 -Communication error : 0xX8
RAM	m + 3 byte	−		Jump to RAM storage start address

Note 1: In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.

Note 2: In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.

Note 3: The 19th to 25th bytes must be within the RAM address range from 0x2000_0400 through the end address of RAM.

21.2.9.2 Chip Erase and Protect Bit Erase

Table 21-7 Transfer Format for the Chip and Protection Bit Erase Command

	Byte	Data Transferred from the Controller to the TMPM341FDXBG/FYXBG	Baud rate	Data Transferred from the TMPM341FDXBG/ FYXBG to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode : 0x86 For I/O Interface mode : 0x30	Desired baud rate (Note 1)	−
	2 byte	−		ACK for the serial operation mode byte ・For UART mode -Normal acknowledge : 0x86 ・For I/O Interface mode -Normal acknowledge : 0x30 (The boot program aborts if the baud rate can not be set correctly.)
	3 byte	Command code (0x40)		−
	4 byte	−		ACK for the command code byte (Note 2) -Normal acknowledge : 0x40 -Negative acknowledge : 0xX1 -Communication error : 0xX8
	5 byte to 16byte	TMPM341FDXBG: In case of 0x3F87_FFF0 is not 0xFF PASS WORD data (12bytes) is 0x3F87_FFF4 to 0x3F87_FFFF In case of 0x3F87_FFF0 is 0xFF DUMMY data (12bytes) is 0x3F87_FFF4 to 0x3F87_FFFF TMPM341FYXBG: In case of 0x3F83_FFF0 is not 0xFF PASS WORD data (12bytes) is 0x3F83_FFF4 to 0x3F83_FFFF In case of 0x3F83_FFF0 is 0xFF DUMMY data (12bytes) is 0x3F83_FFF4 to 0x3F83_FFFF		−
	17 byte	Checksum value for 5 byte to 16 byte		−
	18 byte	−		ACK for the checksum byte (Note 2) -Normal acknowledge : 0x40 -Negative acknowledge : 0xX1 -Communication error : 0xX8
	19 byte	Erase enable command code (0x54)		−
	20 byte	−		ACK for the command code byte (Note 2) -Normal acknowledge : 0x54 -Negative acknowledge : 0xX1 -Communication error : 0xX8
	21 byte	−		ACK for the erase command code byte -Normal acknowledge : 0x4F -Negative acknowledge : 0x4C
	22 byte	(Wait for the next command code.)		−

Note 1: In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.

Note 2: In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.

21.2.10 Operation of Boot Program

When Single Boot mode is selected, the boot program is automatically executed on startup. The boot program offers these four commands, of which the details are provided on the following subsections.

1. RAM Transfer command

   The RAM Transfer command stores program code transferred from a host controller to the on-chip RAM and executes the program once the transfer is successfully completed. The user program RAM space can be assigned to the range from 0x2000_0400 to the end address of RAM, whereas the boot program area (0x2000_0000 ~ 0x2000_03FF) is unavailable. The user program starts at the assigned RAM address.

   The RAM Transfer command can be used to download a flash programming routine of your own; this provides the ability to control on-board programming of the flash memory in a unique manner. The programming routine must utilize the flash memory command sequences described in Section 21.3. Before initiating a transfer, the RAM Transfer command verifies a password sequence coming from the controller against that stored in the flash memory.

2. Chip and Protection Bit Erase command

   This command erases the entire area of the flash memory automatically without verifying a password. All the blocks in the memory cell and their protection conditions are erased even when any of the blocks are prohibited from writing and erasing. When the command is completed, the FCSECBIT <SECBIT> bit is set to "1". This command serves to recover boot programming operation when a user forgets the password. Therefore password verification is not executed.

   This command can select whether a password is verified.

Note: If a password is set to 0xFF (erased data area), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.

21.2.10.1 RAM Transfer Command

See Table 21-6 for the transfer format of this command.

1. The 1st byte specifies which one of the two serial operation modes is used. For a detailed description of how the serial operation mode is determined, see "21.2.10.4 Determination of a Serial Operation Mode" described later. If it is determined as UART mode, the boot program then checks if the SIO0 is programmable to the baud rate at which the 1st byte was transferred. During the first-byte interval, the RXE bit in the SC0MOD register is cleared.
   - To communicate in UART mode

     Send, from the controller to the target board, 0x86 in UART data format at the desired baud rate. If the serial operation mode is determined as UART, then the boot program checks if the SIO0 can be programmed to the baud rate at which the first byte was transferred. If that baud rate is not possible, the boot program aborts, disabling any subsequent communications.
   - To communicate in I/O Interface mode

     Send, from the controller to the target board, 0x30 in I/O Interface data format at 1/16 of the desired baud rate. Also send the 2nd byte at the same baud rate. Then send all subsequent bytes at a rate equal to the desired baud rate.

     In I/O Interface mode, the CPU sees the serial receive pin as if it were a general input port in monitoring its logic transitions. If the baud rate of the incoming data is high or the chip's operating frequency is high, the CPU may not be able to keep up with the speed of logic transitions. To prevent such situations, the 1st and 2nd bytes must be transferred at 1/16 of the desired baud rate; then the boot program calculates 16 times that as the desired baud rate. When the serial operation mode is determined as I/O Interface mode, the SIO0 is configured for SCLK Input mode. Beginning with the third byte, the controller must ensure that its AC timing restrictions are satisfied at the selected baud rate. In the case of I/O Interface mode, the boot program does not check the receive error flag; thus there is no such thing as error acknowledge (bit 3, 0xX8).

2. The 2nd byte, transmitted from the target board to the controller, is an acknowledge response to the 1st byte. The boot program echoes back the first byte: 0x86 for UART mode and 0x30 for I/O Interface mode.
   - UART mode

     If the SIO0 can be programmed to the baud rate at which the 1st byte was transferred, the boot program programs the SC0BRCR and sends back 0x86 to the controller as an acknowledge. If the SIO0 is not programmable at that baud rate, the boot program simply aborts with no error indication. Following the 1st byte, the controller should allow for a time-out period of five seconds. If it does not receive 0x86 within the allowed time-out period, the controller should give up the communication. The boot program sets the RXE bit in the SC0MOD0 register to enable reception ("1") before loading the SIO transmit buffer with 0x86.
   - I/O Interface mode

     The boot program programs the SC0MOD0 and SC0CR registers to configure the SIO0 in I/O Interface mode (clocked by the rising edge of SCLK0), writes 0x30 to the SC0BUF. Then, the SIO0 waits for the SCLK0 signal to come from the controller. Following the transmission of the 1st byte, the controller should send the SCLK clock to the target board after a certain idle time (several microseconds). This must be done at 1/16 the desire baud rate. If the 2nd byte, which is from the target board to the controller, is 0x30, then the controller should take it as a go-ahead. The controller must then deliver the 3rd byte to the target board at a rate equal to the desired baud rate. The boot program sets the RXE bit in the SC0MOD register to enable reception before loading the SIO transmit buffer with 0x30.

3. The 3rd byte transmitted from the controller to the target board is a command. The code for the RAM Transfer command is 0x10.

4. The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

   If the 3rd byte is equal to any of the command codes listed in Table 21-4, the boot program echoes it back to the controller. When the RAM Transfer command was received, the boot program echoes back a value of 0x10 and then branches to the RAM Transfer routine. Once this branch is taken, password verification is done. Password verification is detailed in a later section "Password". If the 3rd byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

5. The 5th to 16th bytes transmitted from the controller to the target board, are a 12-byte password. Each byte is compared to the contents of following addresses in the flash memory. The verification is started with the 5th byte and the smallest address in the designated area. If the password verification fails, the RAM Transfer routine sets the password error flag.

Product name	Password area
TMPM341FDXBG	0x3F87_FFF4 to 0x3F87_FFFF
TMPM341FYXBG	0x3F83_FFF4 to 0x3F83_FFFF

6. The 17th byte is a checksum value for the password sequence (5th to 16th bytes). To calculate the checksum value for the 12-byte password, add the 12 bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in a later section "Checksum Calculation".

7. The 18th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th to 17th bytes. First, the RAM Transfer routine checks for a receive error in the 5th to 17th bytes. If there was a receive error, the boot program sends back 0x18 (bit 3) and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

   Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 5th to 16th bytes must result in 0x00 (with the carry dropped). If it is not 0x00, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 0x11 to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

   Finally, the RAM Transfer routine examines the result of the password verification. The following two cases are treated as a password error. In these cases, the RAM Transfer routine sends back 0x11 (bit 0) to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

   - Irrespective of the result of the password comparison, all the 12 bytes of a password in the flash memory are the same value other than 0xFF.
   - Not the entire password bytes transmitted from the controller matched those contained in the flash memory.

When all the above verification has been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

8. The 19th to 22nd bytes, transmitted from the controller the target board, indicate the start address of the RAM region where subsequent data (e.g., a flash programming routine) should be stored. The 19th byte corresponds to bits 31.24 of the address and the 22nd byte corresponds to bits 7.0 of the address. The start address of the stored RAM must be even address.

9. The 23rd and 24th bytes, transmitted from the controller to the target board, indicate the number of bytes that will be transferred from the controller to be stored in the RAM. The 23rd byte corresponds to bits 15.8 of the number of bytes to be transferred, and the 24th byte corresponds to bits 7.0 of the number of bytes.

10. The 25th byte is a checksum value for the 19th to 24th bytes. To calculate the checksum value, add all these bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in a later section "Checksum Calculation".

11. The 26th byte, transmitted from the target board to the controller, is an acknowledge response to the 19th to 25th bytes of data. First, the RAM Transfer routine checks for a receive error in the 19th to 25th bytes. If there was a receive error, the RAM Transfer routine sends back 0x18 and returns to the command wait state (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

    Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 19th to 25th bytes must result in 0x00 (with the carry dropped). If it is not 0x00, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 0x11 to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

    - The RAM storage start address must be within the range of 0x2000_0400 to the end address of RAM.

    When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

12. The 27th to mth bytes from the controller are stored in the on-chip RAM of the TMPM341FDXBG/FYXBG. Storage begins at the address specified by the 19th.22nd bytes and continues for the number of bytes specified by the 23rd.24th bytes.

13. The (m+1) th byte is a checksum value. To calculate the checksum value, add the 27th to mth bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in a later section "Checksum Calculation".

14. The (m+2) th byte is a acknowledge response to the 27th to (m+1) th bytes. First, the RAM Transfer routine checks for a receive error in the 27th to (m+1) th bytes. If there was a receive error, the RAM Transfer routine sends back 0x18 (bit 3) and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

    Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 27th to (m+1) th bytes must result in 0x00 (with the carry dropped). If it is not 0x00, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 0x11 (bit 0) to the controller and returns to the command wait state (i.e., the 3rd byte) again. When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

15. If the (m+2) th byte was a normal acknowledge response, a branch is made to the address specified by the 19th to 22nd bytes after a normal acknowledge response (0x10) is transferred.

21.2.10.2 Chip and Protection Bit Erase Command

See Table 21-7 for the transfer format of this command.

1. The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

2. From the Controller to the TMPM341FDXBG/FYXBG

   The 3rd byte, which the target board receives from the controller, is a command. The code for the Show Product Information command is 0x40.

3. From the TMPM341FDXBG/FYXBG to the Controller

   The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte.

   Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

   If the 3rd byte is equal to any of the command codes listed in Table 21-4, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was received, the boot program echoes back a value of 0x40. If the 3rd byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

4. The 5th byte to 16th byte, transmitted from the target board to the controller, differ according to the Chip Erase Enable/Disable area (TMPM341FDXBG: 0x3F87_FFF0, TMPM341FYXBG: 0x3F83_FFF0).

   If (TMPM341FDXBG: 0x3F87_FFF0, TMPM341FYXBG: 0x3F83_FFF0) is not the same value as 0xFF, a delete password is required. The 5th byte from the 16th byte becomes password data (12bytes). The verification is started with the 5th byte to compare with the addresses in the flash memory shown in the table below. If the password verification fails, the password error flag is set.

Product name	Password area
TMPM341FDXBG	0x3F87_FFF4 to 0x3F87_FFFF
TMPM341FYXBG	0x3F83_FFF4 to 0x3F83_FFFF

   If (TMPM341FDXBG: 0x3F87_FFF0, TMPM341FYXBG: 0x3F83_FFF0) is 0xFF, a password is not required. The 5th byte from 16th byte becomes dummy data.

5. The 17th byte is a checksum value. To calculate the checksum value, perform 8-bit unsigned addition of transmits data (5th byte to 16th byte), drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in a later section "Checksum Calculation".

6. The 18th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th to 17th bytes. First, the Chip and Protection bit erase routine checks for a receive error in the 5th to 17th bytes. If there was a receive error, the boot program sends back 0x48 (bit 3) and returns to the state in which it for a command (i.e., the 3rd byte) again. Since the upper four bits of the transmitted data are the same as those of the previously issued command (i.e., 4). When the SIO0 is configured for I/O Interface mode, the Chip and Protection bit erase routine does not check for a receive error.

   Next, the Chip and Protection bit erase routine performs the checksum operation to ensure data integrity. Adding the series of the 5th to 16th bytes must result in 0x00 (with the carry dropped). If it is not 0x00, one or more bytes of data have been corrupted. In case of a checksum error, the Chip and Protection bit erase routine sends back the ACK response data (0x41) to the controller and returns to the state in which it for a command (i.e., the 3rd byte) again.

   Finally, the Chip and Protection bit erase routine examines the result of the password verification. The following two cases are treated as a password error. In these cases, the Chip and Protection bit erase routine sends back the ACK response error 0x41 (bit 0) to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

   - Irrespective of the result of the password comparison (from 5th byte to 16th byte), all the 12 bytes of a password in the flash memory are the same value other than 0xFF.

   - Not the entire password bytes transmitted from the controller matched those contained in the flash memory.

   When all the above verification has been successful, the Chip and Protection bit erase routine returns a normal acknowledge response (0x40) to the controller.

7. From the Controller to the Device

   The 19th byte, transmitted from the target board to the controller, is the Chip Erase Enable command code (0x54).

8. From the Device to the Controller

   The 20th byte, transmitted from the target board to the controller, is an acknowledge response to the 19th byte.

   Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

   If the 19th byte is equal to any of the command codes to enable erasing, the boot program echoes it back to the controller (normal ACK response data). In this case, the boot program echoes back a value of 0x54 and then branches to the Chip Erase routine. If the 19th byte is not a valid command, the boot program sends back ACK response data 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the 3rd byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

9. From the Device to the Controller

   The 21st byte indicates whether the Chip Erase command is normally completed or not.

   At normal completion, completion code (0x4F) is sent.

   When an error was detected, error code (0x4C) is sent.

10. The 22nd byte of received data is the next command.

21.2.10.3 Acknowledge Responses

The boot program represents processing states with specific codes. Table 21-8 to Table 21-11 show the values of possible acknowledge responses to the received data. The upper four bits of the acknowledge response are equal to those of the command being executed. Bit 3 of the code indicates a receive error. Bit 0 indicates an invalid command error, a checksum error or a password error. Bit 1 and bit 2 are always "0". Receive error checking is not done in I/O Interface mode.

Table 21-8 ACK Response to the Serial Operation Mode Byte

Return Value	Meaning
0x86	The SIO can be configured to operate in UART mode. (See Note)
0x30	The SIO can be configured to operate in I/O Interface mode.

Note: If the serial operation mode is determined as UART, the boot program checks if the SIO can be programmed to the baud rate at which the operation mode byte was transferred. If that baud rate is not possible, the boot program aborts, without sending back any response.

Table 21-9 ACK Response to the Command Byte

Return Value	Meaning
0xX8 (See Note)	A receive error occurred while getting a command code.
0xX1 (See Note)	An undefined command code was received. (Reception was completed normally.)
0x10	The RAM Transfer command was received.
0x40	The Chip Erase command was received.

Note: The upper four bits of the ACK response are the same as those of the previous command code.

Table 21-10 ACK Response to the Checksum Byte

Return Value	Meaning
0xX8 (See Note)	A receive error occurred.
0xX1 (See Note)	A checksum or password error occurred.
0xX0 (See Note)	The checksum was correct.

Note: The upper four bits of the ACK response are the same as those of the operation command code. It is 1 (X ; RAM transfer command data [7:4]) when password error occurs.

Table 21-11 ACK Response to Chip and Protection Bit Erase Byte

Return Value	Meaning
0x54	The Erase enabling command was received.
0x4F	The Erase command was completed.
0x4C	The Erase command was abnormally completed.

21.2.10.4 Determination of a Serial Operation Mode

The first byte from the controller determines the serial operation mode. To use UART mode for communications between the controller and the target board, the controller must first send a value of 0x86 at a desired baud rate to the target board. To use I/O Interface mode, the controller must send a value of 0x30 at 1/16 the desired baud rate. Figure 21-5 shows the waveforms for the first byte.

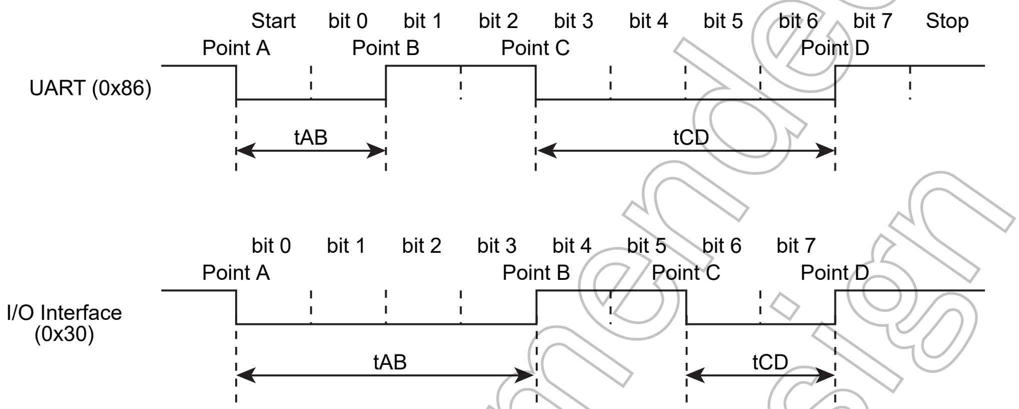


Figure 21-5 Serial Operation Mode Byte

After $\overline{\text{RESET}}$ is released, the boot program monitors the first serial byte from the controller, with the SIO reception disabled, and calculates the intervals of tAB, tAC and tAD. Figure 21-6 shows a flowchart describing the steps to determine the intervals of tAB, tAC and tAD. As shown in the flowchart, the boot program captures timer counts each time a logic transition occurs in the first serial byte. Consequently, the calculated tAB, tAC and tAD intervals are bound to have slight errors. If the transfer goes at a high baud rate, the CPU might not be able to keep up with the speed of logic transitions at the serial receive pin. In particular, I/O Interface mode is more prone to this problem since its baud rate is generally much higher than that for UART mode. To avoid such a situation, the controller should send the first serial byte at 1/16 the desired baud rate.

The flowchart in Figure 21-6 shows how the boot program distinguishes between UART and I/O Interface modes. If the length of tAB is equal to or less than the length of tCD, the serial operation mode is determined as UART mode. If the length of tAB is greater than the length of tCD, the serial operation mode is determined as I/O Interface mode. Bear in mind that if the baud rate is too high or the timer operating frequency is too low, the timer resolution will be coarse, relative to the intervals between logic transitions. This becomes a problem due to inherent errors caused by the way in which timer counts are captured by software; consequently the boot program might not be able to determine the serial operation mode correctly. To prevent this problem, reset UART mode within the programming routine.

For example, the serial operation mode may be determined to be I/O Interface mode when the intended mode is UART mode. To avoid such a situation, when UART mode is utilized, the controller should allow for a time-out period within which it expects to receive an echo-back (0x86) from the target board. The controller should give up the communication if it fails to get that echo-back within the allowed time. When I/O Interface mode is utilized, once the first serial byte has been transmitted, the controller should send the SCLK clock after a certain idle time to get an acknowledge response. If the received acknowledge response is not 0x30, the controller should give up further communications.

When the intended mode is I/O interface mode, the first byte does not have to be 0x30 as long as tAB is greater than tCD as shown above. 0x91, 0xA1 or 0xB1 can be sent as the first byte code to determine the falling edges of Point A and Point C and the rising edges of Point B and Point D. If tAB is greater than tCD and SIO is selected by the resolution of the operation mode determination, the second byte code is 0x30 even though the transmitted code on the first byte is not 0x30 (The first byte code to determine I/O interface mode is described as 0x30).

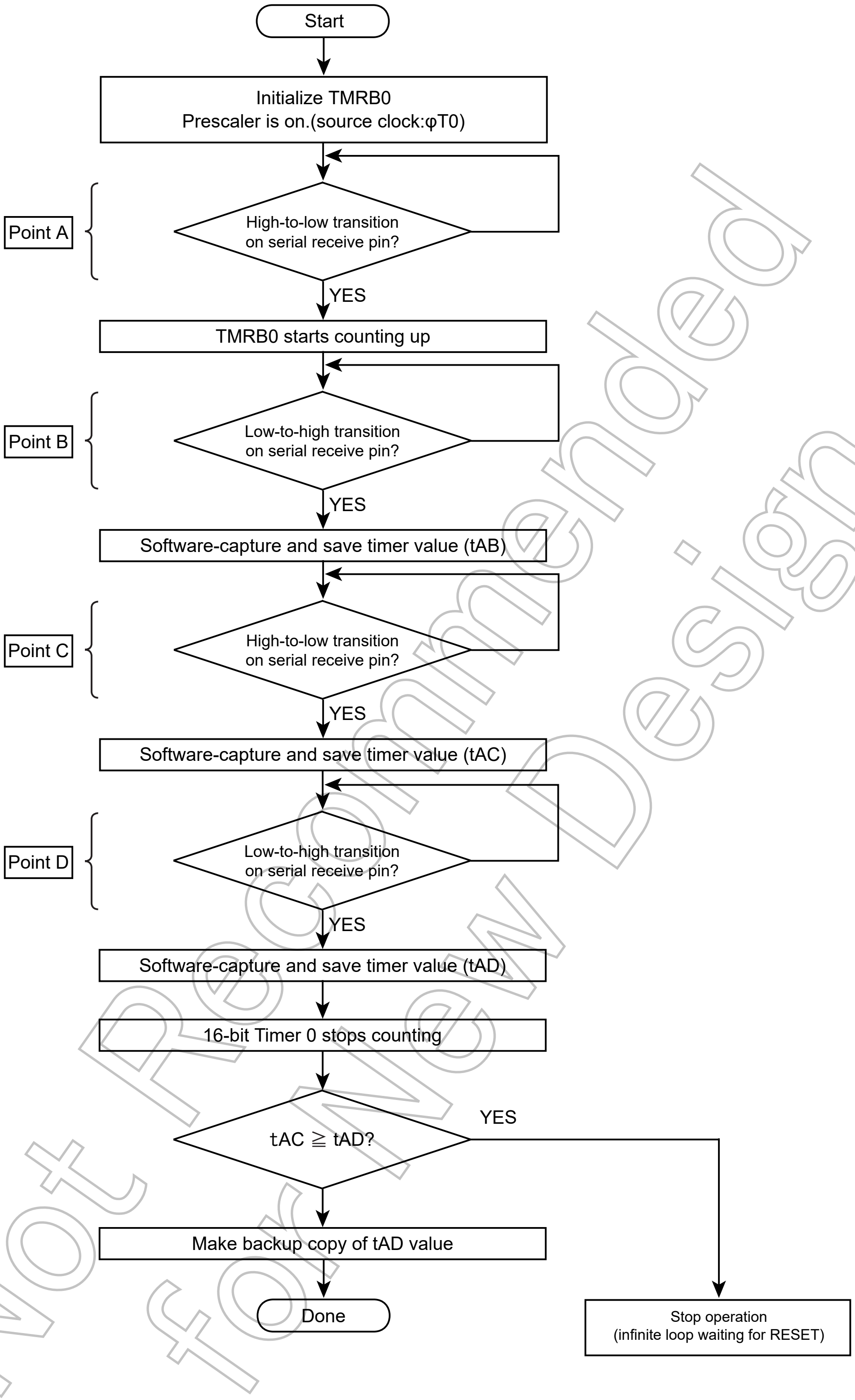


Figure 21-6 Serial Operation Mode Byte Reception Flowchart

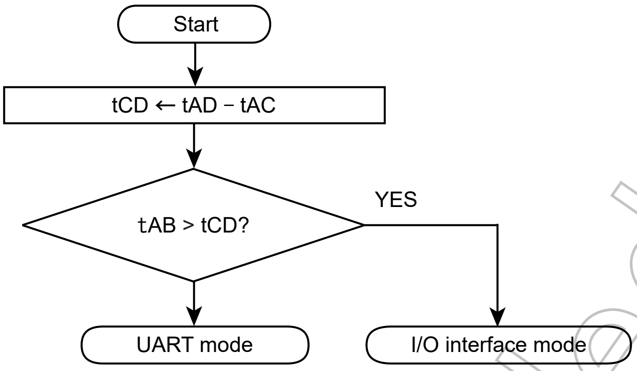


Figure 21-7 Serial Operation Mode Determination Flowchart

21.2.10.5 Password

Verification methods differ according to operation commands. The password area is common to all commands, as shown below. Password verification is performed even if the security is enabled.

Product name	Password area
TMPM341FDXBG	0x3F87_FFF4 to 0x3F87_FFFF
TMPM341FYXBG	0x3F83_FFF4 to 0x3F83_FFFF

Note: If a password is set to 0xFF (erased data area), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.

(1) RAM Transfer command

After an operation command code echoes back (0x10), the boot program verifies a data at password area (12 bytes).

If all these address locations contain the same bytes of data other than 0xFF, a password area error occurs as shown in Figure 21-8. In this case, the boot program returns an error acknowledge (0x11) in response to the checksum byte (the 17th byte), regardless of whether the password sequence sent from the controller is all 0xFFs.

The password sequence received from the controller (5th to 16th bytes) is compared to the password stored in the flash memory. All of the 12 bytes must match to pass the password verification. Otherwise, a password error occurs, which causes the boot program to reply an error acknowledge in response to the checksum byte (the 17th byte).

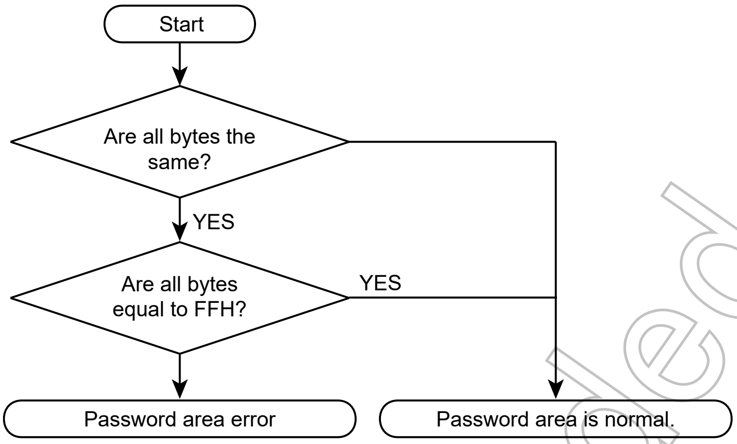


Figure 21-8 Password Area Verification Flowchart (1)

(2) Flash Memory Chip Erase and Protect Bit Erase

The Chip erase enable/disable area specifies whether password verification is performed or not. The Chip erase enable/disable area is as shown below.

Product	The Chip erase enable/disable area
TMPM341FDXBG	0x3F87_FFF0
TMPM341FYXBG	0x3F83_FFF0

As shown in the Figure 21-9, if the data contained in the Chip erase enable/disable area is not 0xFF, password verification is executed. If all data in the password area is the same, it is determined as an error. The boot program returns an error acknowledge (0x41) in response to the checksum byte (the 17th byte).

The password sequence received from the controller (5th to 16th bytes) is compared to the password stored in the flash memory. All of the 12 bytes must match to pass the password verification. Otherwise, a password error occurs, which causes the boot program to reply error acknowledges in response to the checksum byte (the 17th byte).

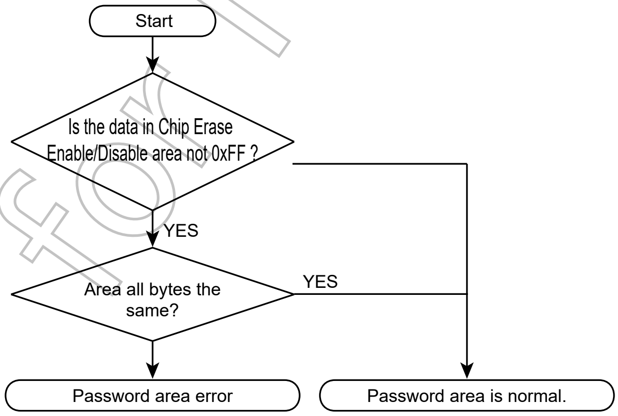


Figure 21-9 Password Area Verification Flowchart (2)

21.2.10.6 Checksum Calculation

The checksum byte for a series of bytes of data is calculated by adding the bytes together, dropping the carries, and taking the two’s complement of the total sum. The Show Flash Memory Sum command and the Show Product Information command perform the checksum calculation. The controller must perform the same checksum operation in transmitting checksum bytes.

Example) Assume the Show Flash Memory Sum command provides the upper and lower bytes of the sum as 0xE5 and 0xF6. To calculate the checksum for a series of 0xE5 and 0xF6:

Add the bytes together

0xE5 + 0xF6 = 0x1DB

Take the two’s complement of the sum, and that is the checksum byte.

0 − 0xDB = 0x25

21.2.11 General Boot Program Flowchart

Figure 21-10 shows an overall flowchart of the boot program.

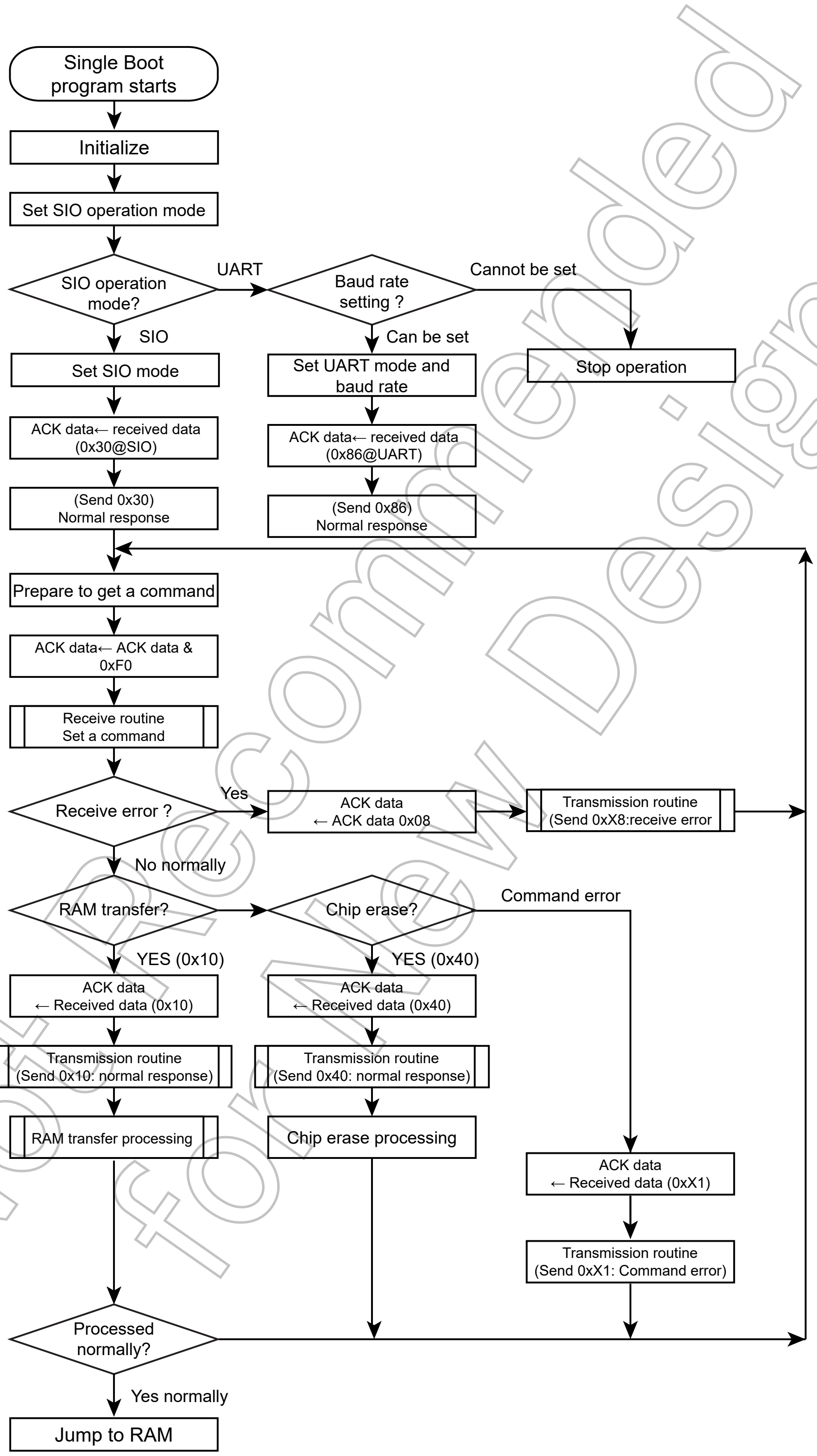

Single Boot program starts
Initialize
Set SIO operation mode
SIO operation mode?
UART
Baud rate setting ?
Cannot be set
SIO
Can be set
Set SIO mode
Set UART mode and baud rate
Stop operation
ACK data← received data (0x30@SIO)
ACK data← received data (0x86@UART)
(Send 0x30) Normal response
(Send 0x86) Normal response
Prepare to get a command
ACK data← ACK data & 0xF0
Receive routine Set a command
Receive error ?
Yes
ACK data ← ACK data 0x08
Transmission routine (Send 0xX8:receive error
No normally
RAM transfer?
Chip erase?
Command error
YES (0x10)
YES (0x40)
ACK data ← Received data (0x10)
ACK data ← Received data (0x40)
Transmission routine (Send 0x10: normal response)
Transmission routine (Send 0x40: normal response)
RAM transfer processing
Chip erase processing
ACK data ← Received data (0xX1)
Transmission routine (Send 0xX1: Command error)
Processed normally?
Yes normally
Jump to RAM

Figure 21-10 Overall Boot Program Flowchart

21.3 On-board Programming of Flash Memory (Rewrite/Erase)

In on-board programming, the CPU is to execute software commands for rewriting or erasing the flash memory. The rewrite/erase control program should be prepared by the user beforehand. Because the flash memory content cannot be read while it is being written or erased, it is necessary to run the rewrite/erase program from the internal RAM after shifting to the user boot mode.

21.3.1 Flash Memory

Except for some functions, writing and erasing flash memory data are in accordance with the standard JEDEC commands. In writing or erasing, use 32-bit data transfer command of the CPU to enter commands to the flash memory. Once the command is entered, the actual write or erase operation is automatically performed internally.

Table 21-12 Flash Memory Functions

Major functions	Description
Automatic page program	Writes data automatically per page.
Automatic chip erase	Erases the entire area of the flash memory automatically.
Automatic block erase	Erases a selected block automatically.
Protect function	The write or erase operation can be individually inhibited for each block.

21.3.1.1 Block Configuration

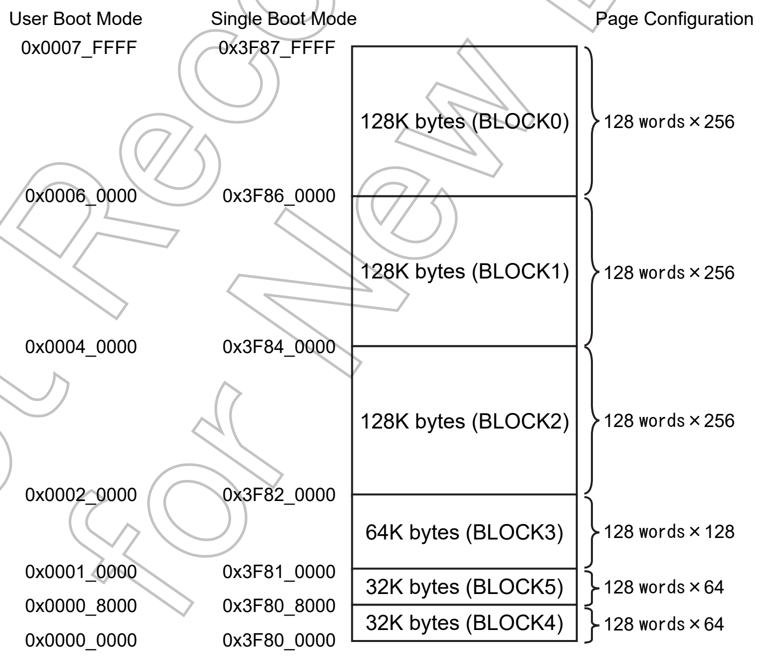


Figure 21-11 Block Configuration of Flash Memory (TMPM341FDXBG)

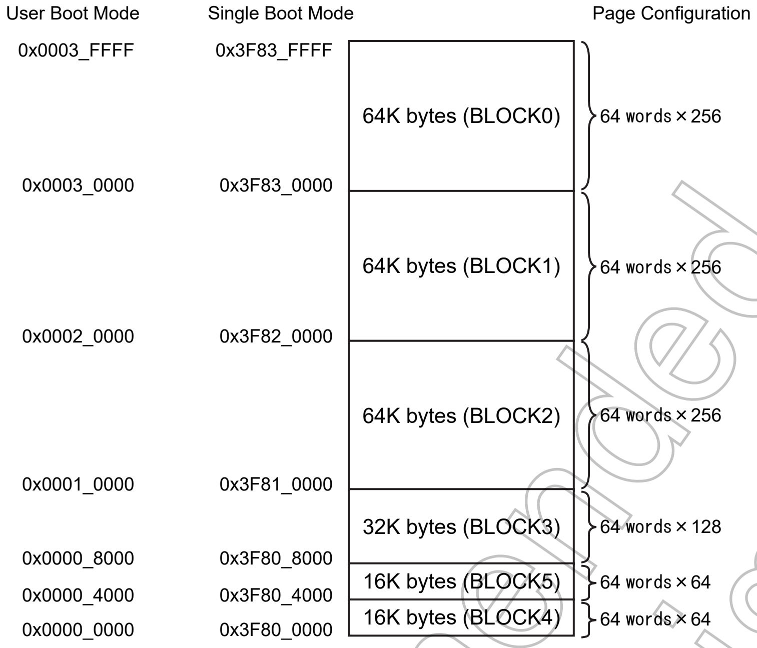


Figure 21-12 Block Configuration of Flash Memory (TMPM341FYXBG)

21.3.1.2 Basic operation

This flash memory device has the following two operation modes:

- The mode to read memory data (Read mode)
- The mode to automatically erase or rewrite memory data (Automatic operation)

Transition to the automatic mode is made by executing a command sequence while it is in the memory read mode. In the automatic operation mode, flash memory data cannot be read and any commands stored in the flash memory cannot be executed. In the automatic operation mode, any interrupt or exception generation cannot set the device to the read mode except when a hardware reset is generated. During automatic operation, be sure not to cause any exceptions other than reset and debug exceptions while a debug port is connected. Any exception generation cannot set the device to the read mode except when a hardware reset is generated.

(1) Read

When data is to be read, the flash memory must be set to the read mode. The flash memory will be set to the read mode immediately after power is applied, when CPU reset is removed, or when an automatic operation is normally terminated. In order to return to the read mode from other modes or after an automatic operation has been abnormally terminated, either the Read/reset command (a software command to be described later) or a hardware reset is used. The device must also be in the read mode when any command written on the flash memory is to be executed.

- Read/reset command and Read command (software reset)

  When a command that has not been completely written has to be canceled, the Read/reset command must be used. The Read command is used to return to the read mode after executing 32-bit data transfer command to write the data "0x0000_00F0" to an arbitrary address of the flash memory.
- With the Read/reset command, the device is returned to the read mode after completing the third bus write cycle.

(2) Command write

This flash memory uses the command control method. Commands are executed by executing a command sequence to the flash memory. The flash memory executes automatic operation commands according to the address and data combinations applied (refer to Command Sequence).

If it is desired to cancel a command write operation already in progress or when any incorrect command sequence has been entered, the Read/reset command is to be executed. Then, the flash memory will terminate the command execution and return to the read.

While commands are generally comprised of several bus cycles, the operation to apply 32-bit data transmit command to the flash memory is called "bus write cycle". The bus write cycles are to be in a specific sequential order and the flash memory will perform an automatic operation when the sequence of the bus write cycle data and address of a command write operation is in accordance with a predefined specific sequence. If any bus write cycle does not follow a predefined command write sequence, the flash memory will terminate the command execution and return to the read mode.

Note 1: Command sequences are executed from outside the flash memory area.

Note 2: Each bus write cycle must be sequentially executed by 32-bit data transmit command. While a command sequence is being executed, access to the flash memory is prohibited. Also, don't generate any interrupt (except debug exceptions when a debug port is connected).If such an operation is made, it can result in an unexpected read access to the flash memory and the command sequencer may not be able to correctly recognize the command. While it could cause an abnormal termination of the command sequence, it is also possible that the written command is incorrectly recognized.

Note 3: For the command sequencer to recognize a command, the device must be in the read mode prior to executing the command. Be sure to check before the first bus write cycle that FCFLCS <RDY/BSY> is set to "1". It is recommended to subsequently execute a Read command.

Note 4: Upon issuing a command, if any address or data is incorrectly written, be sure to perform a software reset to return to the read mode again.

21.3.1.3 Reset (Hardware reset)

A hardware reset is used to cancel the operational mode set by the command write operation when forcibly termination during automatic programming/ erasing or abnormal termination during automatic operation.

The flash memory has a reset input as the memory block and it is connected to the CPU reset signal. Therefore, when the $\overline{\text{RESET}}$ input pin of this device is set to VIL or when the CPU is reset due to any overflow of the watch dog timer, the flash memory will return to the read mode terminating any automatic operation that may be in progress. It should also be noted that applying a hardware reset during an automatic operation can result in incorrect rewriting of data. In such a case, be sure to perform the rewriting again.

Refer to Section "21.2.1 Reset Operation" for CPU reset operations. After a given reset input, the CPU will read the reset vector data from the flash memory and starts operation after the reset is removed.

21.3.1.4 Commands

(1) Automatic Page Programming

Writing to a flash memory device is to make "1" data cells to "0" data cells. Any "0" data cell cannot be changed to a "1" data cell. For making "0" data cells to "1" data cells, it is necessary to perform an erase operation.

The automatic page programming function of this device writes data of each page. The TMPM341FDXBG/FYXBG contain 128 words in a page. A 128 word block is defined by a same [31:9] address and it starts from the address [8:0] = 0x00 and ends at the address [8:0] = 0x1FF. This programming unit is hereafter referred to as a "page".

Writing to data cells is automatically performed by an internal sequencer and no external control by the CPU is required. The state of automatic page programming (whether it is in writing operation or not) can be checked by FCFLCS [0] <RDY/BSY>.

Also, any new command sequence is not accepted while it is in the automatic page programming mode. If it is desired to interrupt the automatic page programming, use the hardware reset function. If the operation is stopped by a hardware reset operation, it is necessary to once erase the page and then perform the automatic page programming again because writing to the page has not been normally terminated.

The automatic page programming operation is allowed only once for a page already erased. No programming can be performed twice or more times irrespective of the data cell value whether it is "1" or "0". Note that rewriting to a page that has been once written requires execution of the automatic block erase or automatic chip erase command before executing the automatic page programming command again. Note that an attempt to rewrite a page two or more times without erasing the content can cause damages to the device.

No automatic verify operation is performed internally to the device. So, be sure to read the data programmed to confirm that it has been correctly written.

The automatic page programming operation starts when the third bus write cycle of the command cycle is completed. On and after the fifth bus write cycle, data will be written sequentially starting from the next address of the address specified in the fourth bus write cycle (in the fourth bus write cycle, the page top address will be command written) (32 bits of data is input at a time). Be sure to use the 32-bit data transfer command in writing commands on and after the fourth bus cycle. In this, any 32-bit data transfer commands shall not be placed across word boundary. On and after the fifth bus write cycle, data is command written to the same page area. Even if it is desired to write the page only partially, it is required to perform the automatic page programming for the entire page. In this case, the address input for the fourth bus write cycle shall be set to the top address of the page. Be sure to perform command write operation with the input data set to "1" for the data cells not to be set to "0". For example, if the top address of a page is not to be written, set the input data of the fourth bus write cycle to 0xFFFFFFFF to command write the data.

Once the third bus cycle is executed, the automatic page programming is in operation. This condition can be checked by monitoring FCFLCS<RDY/BSY>. Any new command sequence is not accepted while it is in automatic page programming mode. If it is desired to stop operation, use the hardware reset function. Be careful in doing so because data cannot be written normally if the operation is interrupted. When a single page has been command written normally terminating the automatic page writing process, FCFLCS<RDY/BSY> is set to "1" and it returns to the read mode.

When multiple pages are to be written, it is necessary to execute the page programming command for each page because the number of pages to be written by a single execution of the automatic page program command is limited to only one page. It is not allowed for automatic page programming to process input data across pages.

Data cannot be written to a protected block. When automatic programming is finished, it automatically returns to the read mode. This condition can be checked by monitoring FCFLCS<RDY/BSY>. If automatic programming has failed, the flash memory is locked in the mode and will not return to the read mode. For returning to the read mode, it is necessary to execute hardware reset to reset the flash memory or the device. In this case, while writing to the address has failed, it is recommended not to use the device or not to use the block that includes the failed address.

Note: Software reset becomes ineffective in bus write cycles on and after the fourth bus write cycle of the automatic page programming command.

(2) Automatic chip erase

The automatic chip erase operation starts when the sixth bus write cycle of the command cycle is completed.

This condition can be checked by monitoring FCFLCS<RDY/BSY>. While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic chip erase operation. If it is desired to stop operation, use the hardware reset function. If the operation is forced to stop, it is necessary to perform the automatic chip erase operation again because the data erasing operation has not been normally terminated.

Also, any protected blocks cannot be erased. If all the blocks are protected, the automatic chip erase operation will not be performed and it returns to the read mode after completing the sixth bus read cycle of the command sequence. When an automatic chip erase operation is normally terminated, it automatically returns to the read mode. If an automatic chip erase operation has failed, the flash memory is locked in the mode and will not return to the read mode.

For returning to the read mode, it is necessary to execute hardware reset to reset the device. In this case, the failed block cannot be detected. It is recommended not to use the device anymore or to identify the failed block by using the block erase function for not to use the identified block anymore.

(3) Automatic block erase (for each block)

The automatic block erase operation starts when the sixth bus write cycle of the command cycle is completed.

This status of the automatic block erase operation can be checked by monitoring FCFLCS <RDY/BSY>. While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic block erase operation. If it is desired to stop operation, use the hardware reset function. In this case, it is necessary to perform the automatic block erase operation again because the data erasing operation has not been normally terminated.

Also, any protected blocks cannot be erased. If an automatic block erase operation has failed, the flash memory is locked in the mode and will not return to the read mode. In this case, execute hardware reset to reset the device.

(4) Automatic programming of protection bits (for each block)

This device is implemented with protection bits. This protection can be set for each block. See Table 21-16 for table of protection bit addresses. This device assigns 1 bit to 1 block as a protection bit. The applicable protection bit is specified by PBA in the seventh bus write cycle. By automatically programming the protection bits, write and/or erase functions can be inhibited (for protection) individually for each block. The protection status of each block can be checked by FCFLCS <BLPRO> to be described later. This status of the automatic programming operation to set protection bits can be checked by monitoring FCFLCS <RDY/BSY>. Any new command sequence is not accepted while automatic programming is in progress to program the protection bits. If it is desired to stop the programming operation, use the hardware reset function. In this case, it is necessary to perform the programming operation again because the protection bits may not have been correctly programmed. If all the protection bits have been programmed, all FCFLCS <BLPRO> are set to "1" indicating that it is in the protected state. This disables subsequent writing and erasing of all blocks.

Note: Software reset is ineffective in the seventh bus write cycle of the automatic protection bit programming command. FCFLCS <RDY/BSY> turns to "0" after entering the seventh bus write cycle.

(5) Automatic erasing of protection bits

Different results will be obtained when the automatic protection bit erase command is executed depending on the status of the protection bits and the security bits. It depends on the status of FCFLCS <BLPRO> whether all <BLPRO> are set to "1" or not if FCSECBIT<FCSECBIT> is 0x1. Be sure to check the value of FCFLCS <BLPRO> before executing the automatic protection bit erase command. See the chapter "ROM protection" for details.

- When all the FCFLCS <BLPRO> are set to "1" (all the protection bits are programmed):

  When the automatic protection bit erase command is command written, the flash memory is automatically initialized within the device. When the seventh bus write cycle is completed, the entire area of the flash memory data cells is erased and then the protection bits are erased. This operation can be checked by monitoring FCFLCS <RDY/BSY>. If the automatic operation to erase protection bits is normally terminated, FCFLCS will be set to "0x00000001".While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that it has been correctly erased. For returning to the read mode while the automatic operation after the seventh bus cycle is in progress, it is necessary to use the hardware reset to reset the device. If this is done, it is necessary to check the status of protection bits by FCFLCS <BLPRO> after retuning to the read mode and perform either the automatic protection bit erase, automatic chip erase, or automatic block erase operation, as appropriate.

- When FCFLCS <BLPRO> include "0" (not all the protection bits are programmed):

If the automatic protection bit is cleared to "0", the protection condition is canceled. With this device, protection bits can be programmed to an individual block and performed bit-erase operation in the four bits unit as shown in Table 21-17. The target bits are specified in the seventh bus write cycle.The protection status of each block can be checked by FCFLCS <BLPRO> to be described later. This status of the programming operation for automatic protection bits can be checked by monitoring FCFLCS <RDY/BSY>. When the automatic operation to erase protection bits is normally terminated, the protection bits of FCFLCS <BLPRO> selected for erasure are set to "0".

In any case, any new command sequence is not accepted while it is in an automatic operation to erase protection bits. If it is desired to stop the operation, use the hardware reset function. When the automatic operation to erase protection bits is normally terminated, it returns to the read mode.

Note: The FCFLCS <RDY/BSY> bit is "0" while in automatic operation and it turns to "1" when the automatic operation is terminated.

(6) ID-Read

Using the ID-Read command, you can obtain the type and other information on the flash memory contained in the device. The data to be loaded will be different depending on the address [15:14] of the fourth and subsequent bus write cycles (recommended input data is 0x00). On and after the fourth bus write cycle, when an arbitrary flash memory area is read, the ID value will be loaded. Once the fourth bus write cycle of an ID-Read command has passed, the device will not automatically return to the read mode. In this condition, the set of the fourth bus write cycle and ID-Read commands can be repetitively executed. For returning to the read mode, use the Read/reset command or hardware reset command.

21.3.1.5 Flash control/ status register

Base Address = 0x41FF_F000

Register name		Address (Base+)
Reserved	-	0x0000 to 0x000F
Security bit register	FCSECBIT	0x0010
Reserved	-	0x0014 to 0x001F
Flash control register	FCFLCS	0x0020
Reserved	-	0x0024 to 0x0FFF

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

(1) FCSECBIT (Security bit register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	SECBIT	R/W	Security bits 0:disabled 1:enabled

Note: **This register is initialized by cold reset or releasing STOP2 mode of standby mode.**

(2) FCFLCS (Flash control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
After reset	0	0	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	RDY/BSY
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-22	−	R	Read as 0.
21-16	BLPRO5 to BLPRO0	R	Protection for Block5 to 0 0: disabled 1: enabled Protection status bits Each of the protection bits represents the protection status of the corresponding block. When a bit is set to "1", it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.
15-1	−	R	Read as 0.
0	RDY/BSY	R	Ready/Busy (Note 1) 0: automatic operating 1:automatic operation terminated Ready/Busy flag bit The RDY/BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1".

Note 1: This command must be issued in the ready state. Issuing the command in the busy state may disable both correct command transmission and further command input. To exit from the condition, execute system reset. System reset requires at least 0.5 μs regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.

Note 2: The value varies depending on protection applied.

21.3.1.6 List of Command Sequences

Table 21-13 shows the addresses and the data of each command of flash memory.

Bus cycles are "bus write cycles" except for the second bus cycle of the Read command and the fourth bus cycle of the Read/reset command. Bus write cycles are executed by 32-bit (word) data transfer commands. (In the following table, only lower 8 bits data are shown.)

See Table 21-14 for the detail of the address bit configuration. Use a value of "Addr." in the Table 21-13 for the address [15:8] of the normal command in the Table 21-14.

Note: **Always set "0" to the address bits [1:0] in the entire bus cycle.**

Table 21-13 Flash Memory Access from the Internal CPU

Command sequence	First bus cycle	Second bus cycle	Third bus cycle	Fourth bus cycle	Fifth bus cycle	Sixth bus cycle	Seventh bus cycle
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	Data	Data	Data	Data	Data	Data	Data
Read	0xXX	–	–	–	–	–	–
	0xF0	–	–	–	–	–	–
Read/Reset	0x54XX	0xAAXX	0x54XX	RA	–	–	–
	0xAA	0x55	0xF0	RD	–	–	–
ID-Read	0x54XX	0xAAXX	0x54XX	IA	0xXX	–	–
	0xAA	0x55	0x90	0x00	ID	–	–
Automatic page programming	0x54XX	0xAAXX	0x54XX	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic chip erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	–
	0xAA	0x55	0x80	0xAA	0x55	0x10	–
Automatic block erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	BA	–
	0xAA	0x55	0x80	0xAA	0x55	0x30	–
Automatic protection bit programming	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
Automatic protection bit erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

Supplementary explanation

- RA: Read address
- RD: Read data
- IA: ID address
- ID: ID data
- PA: Program page address

  PD: Program data (32 bit data)

  After the fourth bus cycle, enter data in the order of the address for a page.
- BA: Block address
- PBA: Protection bit address

21.3.1.7 Address bit configuration for bus write cycles

Table 21-14 is used in conjunction with Table 21-13 "Flash Memory Access from the Internal CPU". Address setting can be performed according to the normal bus write cycle address configuration from the first bus cycle. "0" is recommended" in the Table 21-14 Address Bit Configuration for Bus Write Cycles can be changed as necessary.

Table 21-14 Address Bit Configuration for Bus Write Cycles

Address	Addr [31:19]	Addr [18]	Addr [17]	Addr [16]	Addr [15]	Addr [14]	Addr [13:11]	Addr [10]	Addr [9]	Addr [8]	Addr [7:0]

Normal commands	Normal bus write cycle address configuration										
	Flash area	"0" is recommended.			Command						Addr[1:0]="0" (fixed) Others:0 (recommended)
ID-READ	IA: ID address (Set the fourth bus write cycle address for ID-Read operation)										
	Flash area	"0" is recommended.			ID address		Addr[1:0]="0" (fixed), Others:0 (recommended)				

[TMPM341FDXBG]

Block erase	BA: Block address (Set the sixth bus write cycle address for block erase operation)										
	Block selection (Table 21-15)					Addr[1:0]="0" (fixed) , Others:0 (recommended)					
Automatic page programming	PA: Program page address (Set the fourth bus write cycle address for page programming operation)										
	Page address									Addr[1:0]="0" (fixed) Others:0 (recommended)	
Protection bit programming	PBA: Protection bit address (Set the seventh bus write cycle address for protection bit programming										
	Flash area	Protection bit selection (Table 21-16)		Fixed to "0".				Protection bit selection (Table 21-16)		Addr[1:0]="0" (fixed) Others:0 (recommended)	
Protection bit erase	PBA: Protection bit address (Set the seventh bus erase cycle address for protection bit erasure)										
	Flash area	Protection bit selection (Table 21-17)		Fixed to "0".				Addr[1:0]="0" (fixed) Others:0 (recommended)			

[TMPM341FYXBG]

Block erase	BA: Block address (Set the sixth bus write cycle address for block erase operation)										
	Block selection (Table 21-15)						Addr[1:0]="0" (fixed) , Others:0 (recommended)				
Automatic page programming	PA: Program page address (Set the fourth bus write cycle address for page programming operation)										
	Page address										Addr[1:0]="0" (fixed) Others:0 (recommended)
Protection bit programming	PBA: Protection bit address (Set the seventh bus write cycle address for protection bit programming										
	Flash area	Protection bit selection (Table 21-16)		Fixed to "0".					Protection bit selection (Table 21-16)		Addr[1:0]="0" (fixed) Others:0 (recommended)
Protection bit erase	PBA: Protection bit address (Set the seventh bus erase cycle address for protection bit erasure)										
	Flash area	Protection bit selection (Table 21-17)		Fixed to "0".							Addr[1:0]="0" (fixed) Others:0 (recommended)

As block address, specify any address in the block to be erased.

Table 21-15 Block Address Table

Block	Address (User boot mode)	Address (Single boot mode)	Size (Kbyte)
[TMPM341FDXBG]			
4	0x0000_0000 to 0x0000_7FFF	0x3F80_0000 to 0x3F80_7FFF	32
5	0x0000_8000 to 0x0000_FFFF	0x3F80_4000 to 0x3F80_FFFF	32
3	0x0001_0000 to 0x0001_FFFF	0x3F81_0000 to 0x3F81_FFFF	64
2	0x0002_0000 to 0x0003_FFFF	0x3F82_0000 to 0x3F83_FFFF	128
1	0x0004_0000 to 0x0005_FFFF	0x3F84_0000 to 0x3F85_FFFF	128
0	0x0006_0000 to 0x0007_FFFF	0x3F86_0000 to 0x3F87_FFFF	128
[TMPM341FYXBG]			
4	0x0000_0000 to 0x0000_3FFF	0x3F80_0000 to 0x3F80_3FFF	16
5	0x0000_4000 to 0x0000_7FFF	0x3F80_4000 to 0x3F80_7FFF	16
3	0x0000_8000 to 0x0000_FFFF	0x3F80_8000 to 0x3F80_FFFF	32
2	0x0001_0000 to 0x0001_FFFF	0x3F81_0000 to 0x3F81_FFFF	64
1	0x0002_0000 to 0x0002_FFFF	0x3F82_0000 to 0x3F82_FFFF	64
0	0x0003_0000 to 0x0003_FFFF	0x3F83_0000 to 0x3F83_FFFF	64

Note: As for the addresses from the first to the fifth bus cycles, specify the upper addresses of the blocks to be erased.

Table 21-16 Protection Bit Programming Address Table

Block	Protection bit	The seventh bus write cycle address					
		Address [18]	Address [17]	Address [16:11]	Address [10]	Address [9]	Address [8]

[TMPM341FDXBG]

Block0	<BLPRO[0]>	0	0	Fixed to "0".	0	0	"0" is recommended.
Block1	<BLPRO[1]>	0	0		0	1	
Block2	<BLPRO[2]>	0	0		1	0	
Block3	<BLPRO[3]>	0	0		1	1	
Block4	<BLPRO[4]>	0	1		0	0	
Block5	<BLPRO[5]>	0	1		0	1	

[TMPM341FYXBG]

Block0	<BLPRO[0]>	0	0	Fixed to "0".		0	0
Block1	<BLPRO[1]>	0	0			0	1
Block2	<BLPRO[2]>	0	0			1	0
Block3	<BLPRO[3]>	0	0			1	1
Block4	<BLPRO[4]>	0	1			0	0
Block5	<BLPRO[5]>	0	1			0	1

Table 21-17 Protection Bit Erase Address Table (TMPM341FD/FYXBG)

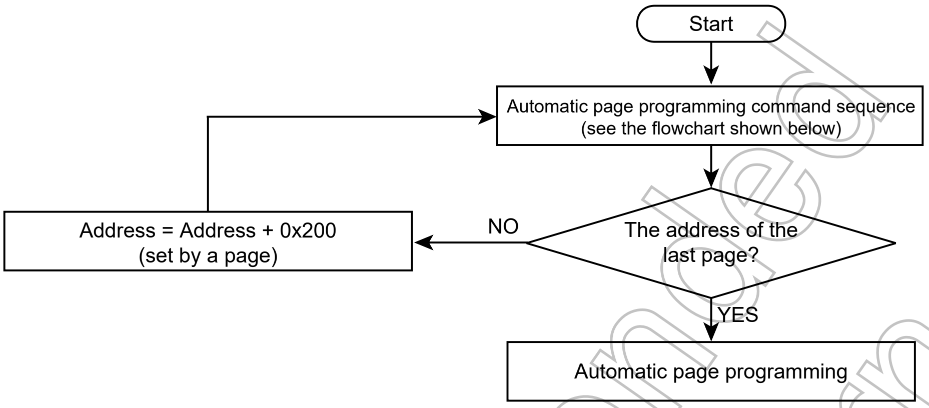
Block	Protection bit	The seventh bus write cycle address [18:17]	
		Address[18]	Address[17]
Block0 to 3	<BLPRO[0:3]>	0	0
Block4 to 5	<BLPRO[4:5]>	0	1

Note: The protection bit erase command cannot erase by individual block.

Table 21-18 The ID-Read command's fourth bus write cycle ID address (IA) and the data to be read by the following 32-bit data transfer command (ID)

IA[15:14]	ID[7:0]	Code
0y00	0x98	Manufacturer code
0y01	0x5A	Device code
0y10	Reserved	–
0y11	0x12 (TMPM341FDXBG) 0x13 (TMPM341FYXBG)	Macro code

21.3.1.8 Flowchart


Start
Automatic page programming command sequence (see the flowchart shown below)
The address of the last page?
NO
Address = Address + 0x200 (set by a page)
YES
Automatic page programming

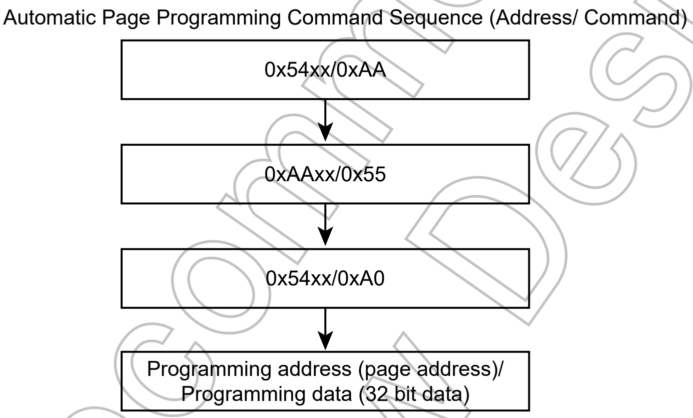

Automatic Page Programming Command Sequence (Address/ Command)
0x54xx/0xAA
0xAAxx/0x55
0x54xx/0xA0
Programming address (page address)/ Programming data (32 bit data)

Figure 21-13 Automatic Programming

Note: Command sequence is executed by 0x54xx or 0x55xx.

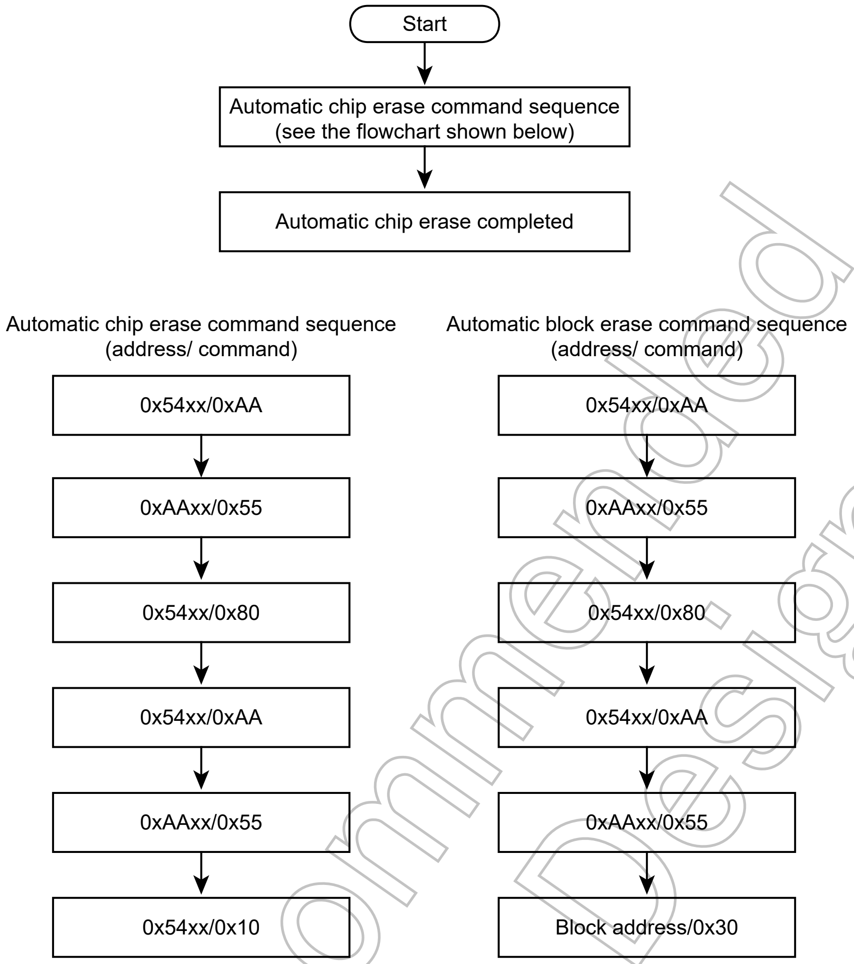


Figure 21-14 Automatic Erase

Note: Command sequence is executed by 0x54xx or 0x55xx.

22. ROM protection

22.1 Outline

The TMPM341FDXBG/FYXBG offers two kinds of ROM protection/ security functions.

One is a write/ erase-protection function for the internal flash ROM data.

The other is a security function that restricts internal flash ROM data readout and debugging.

22.2 Features

22.2.1 Write/ erase-protection function

The write/ erase-protection function enables the internal flash to prohibit the writing and erasing operation for each block.

To activate the function, write "1" to the corresponding bits to a block to protect. Writing "0" to the bits cancels the protection.

The protection settings of the bits can be monitored by the FCFLCS <BLPRO[3:0][5:0]> bit. See the chapter "Flash" for programming details.

22.2.2 Security function

The security function restricts flash ROM data readout and debugging.

This function is available under the conditions shown below.

1. The FCSECBIT <SECBIT> bit is set to"1".
2. All the protection bits (the FCFLCS<BLPRO> bits) used for the write/erase-protection function are set to "1".

Note: **The FCSECBIT <SECBIT> bit is set to "1" at a power-on reset right after power-on.**

Table 22-1 shows details of the restrictions by the security function.

Table 22-1 Restrictions by the security function

Item	Details
1) ROM data readout	Data can be read from CPU.
2) Debug port	Communication of JTAG/SW and trace are prohibited
3) Command for flash memory	Writing a command to the flash memory is prohibited. An attempt to erase the contents in the bits used for the write/ erase-protection erases all the protection bits.

22.3 Register

Base Address = 0x41FF_F000

Register name		Address(Base+)
Reserved	-	0x0000,0x0004
Security bit register	FCSECBIT	0x0010
Reserved	-	0x0014
Flash control register	FCFLCS	0x0020
Reserved	-	0x0024,0x0028
Reserved	-	0x0040,0x0044
Reserved	-	0x0050,0x0058
Reserved	-	0x0060 − 0x00B8

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

22.3.1 FCFLCS (Flash control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	BLPRO3	BLPRO2	BLPRO1	BLPRO0
After reset	0	0	0	0	(Note2)	(Note2)	(Note2)	(Note2)
bit symbol	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
After reset	0	0	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	RDY/BSY
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-20	−	R	Read as 0.
19-16	BLPRO3 to BLPRO0	R	Protection for Block3 to 0 0: disabled 1: enabled Protection status bits Each of the protection bits represents the protection status of the corresponding block. When a bit is set to "1," it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.
31-22	−	R	Read as 0.
21-16	BLPRO5 to BLPRO0	R	Protection for Block5 to 0 0: disabled 1: enabled Protection status bits Each of the protection bits represents the protection status of the corresponding block. When a bit is set to "1," it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.
15-1	−	R	Read as 0.
0	RDY/BSY	R	Ready/Busy (Note 1) 0: Auto operating 1:Auto operation terminated Ready/Busy flag bit The RDY/BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1."

Note 1: **This command must be issued in the ready state. Issuing the command in the busy state may disable both correct command transmission and further command input. To exit from the condition, execute system reset. System reset requires at least 0.5 ms regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.**

Note 2: **The value varies depending on protection applied.**

22.3.2 FCSECBIT(Security bit register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	SECBIT	R/W	Security bit 0: Disabled 1: Enabled

Note: This register is initialized by cold reset.

Note: This register is initialized by cold reset and releasing STOP2 mode of the standby mode.

22.4 Writing and erasing

Writing and erasing protection bits are available with a single chip mode, single boot mode and writer mode.

22.4.1 Protection bits

Writing to the protection bits is done on block-by-block basis.

When the settings for all the blocks are "1", erasing must be done after setting the FCSECBIT <SECBIT> bit to "0". Setting "1" at that situation erases all the protection bits. To write and erase the protection bits, command sequence is used.

See the capter "Flash" for details

22.4.2 Security bit

The FCSECBIT <SECBIT> bit that activates security function is set to "1" at a power-on reset right after power-on.

The bit is rewritten by the following procedure.

1. Write the code 0xa74a9d23 to FCSECBIT register.

2. Write data within 16 clocks from the above.1.

Note: The above procedure is enabled only when using 32-bit data transfer command.

23. Debug Interface

23.1 Specification Overview

TMPM341FDXBG/FYXBG contains the Serial Wire JTAG Debug Port (SWJ-DP) unit for interfacing with the debugging tools and the Embedded Trace Macrocell™(ETM) unit for instruction trace output.Trace data is output to the dedicated pins(TRACEDATA[3:0], SWV) for the debugging via the on-chip Trace Port Interface Unit (TPIU).

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

23.2 SWJ-DP

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

23.3 ETM

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

23.4 Pin Functions

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

The PI6 and PI5 pins are shared between the JTAG debug port function and the Serial Wire Debug Port function. The PI7 pin is shared between the JTAG debug port function and the SWV trace output function.

Table 23-1 SWJ-DP,ETM Debug Functions

SWJ-DP Pin Name	General-purpose Port Name	JTAG Debug Function		SW Debug Function	
		I / O	Explanation	I / O	Explanation
TMS / SWDIO	PI6	Input	JTAG Test Mode Selection	I / O	Serial Wire Data Input/Output
TCK / SWCLK	PI5	Input	JTAG Test Check	Input	Serial Wire Clock
TDO / SWV	PI7	Output	JTAG Test Data Output	(Output)(Note)	(Serial Wire Viewer Output)
TDI	PI4	Input	JTAG Test Data Input	-	-
$\overline{\text{TRST}}$	PI3	Input	JTAG Test $\overline{\text{RESET}}$	-	-
TRACECLK	PI2	Output	TRACE Clock Output		
TRACEDATA0	PI1	Output	TRACE DATA Output0		
TRACEDATA1	PI0	Output	TRACE DATA Output1		
TRACEDATA2	PH6	Output	TRACE DATA Output2		
TRACEDATA3	PH5	Output	TRACE DATA Output3		

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

After reset, PI3, PI4, PI5, PI6 and PI7 pins are configured as debug port function pins. The functions of other debug interface pins need to be programmed as required.

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

Note 1: If PI6 and PI7 are configured as TMS/SWDIO and TDO/SWV, output continues to be enabled even in STOP1 mode regardless of the setting of the CGSTBYCR<DRVE> bit.

Table 23-2 summarizes the debug interface pin and related port settings after reset.

Table 23-2 Debug Interface Pins and Related Port Settings after Reset

Port Name (Bit Name)	Debug Function	Value of Related port settings after reset				
		Function (PxFR)	Input (PxIE)	Output (PxCR)	Pull-up (PxPUP)	Pull-down (PxPDN)
PI6	TMS/SWDIO	1	1	1	1	−
PI5	TCK/SWCLK	1	1	0	−	1
PI7	TDO/SWV	1	0	1	0	−
PI4	TDI	1	1	0	1	−
PI3	$\overline{\text{TRST}}$	1	1	0	1	−
PI2	TRACECLK	0	0	0	0	−
PI1	TRACEDATA0	0	0	0	0	−
PI0	TRACEDATA1	0	0	0	0	−
PH6	TRACEDATA2	0	0	0	0	−
PH5	TRACEDATA3	0	0	0	0	−

− : Don't care

23.5 Peripheral Functions in Halt Mode

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

23.6 Connection with a Debug Tool

23.6.1 About connection with debug tool

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

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

Note 1: Ensure that to measure the power-consumption with debug tool connected in STOP1/STOP2 mode is prohibitted.

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

TMPM341FDXBG/FYXBG is prohibited from transmission with debug tools while reset caused by $\overline{\text{RESET}}$ pin is effective. Therefore it cannot change to the debug mode.

The PI3, PI4, PI5, PI6 and PI7 ports are the debug interface pins after reset however if these pins are changed to the general-purpose port immediately after reset, the control from the debug tools are not accepted under some circumstances.When changing the settings, please pay attention to the status of debug interface pins.

Table 23-3 Table of using debug interface pins

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

o : Enabled × : Disabled (Usable as general-purpose port)
Note: For the treatment of the pin of which the $\overline{\text{TRST}}$ function is assigned, select the $\overline{\text{TRST}}$ function with the function register and set the pin to OPEN or "High level".

24. JTAG Interface

24.1 Overview

The TMPM341FDXBG/FYXBG provides a boundary-scan interface that is compatible with Joint Test Action Group (JTAG) specifications and uses the industry-standard JTAG protocol (IEEE Standard 1149.1 ・ 1990 <Includes IEEE Standard 1449.1a・1993>).

This chapter describes the JTAG interface, with the descriptions of boundary scan and the pins and signals used by the interface.

1. JTAG standard version

   IEEE Standard 1149.1・1990 (Includes IEEE Standard 1149.1a・1993)

2. JTAG instructions

   Standard instructions (BYPASS, SAMPLE/PRELOAD, EXTEST)

   HIGHZ instruction

   CLAMP instruction

   However, the SAMPLE/RELOAD instruction doesn't function because internal circuit reset starts as for TMPM341FDXBG/FYXBG while JTAG is operating.

3. IDCODE

   Not available

4. Pins excluded from boundary scan register (BSR)
   a. Oscillator circuit pins (X1, X2)
   b. DAC pins (DA0, DA1)
   c. JTAG control pins (BSC)
   d. Power supply/GND pins (including AVREFH, AVREFL)
   e. Monitor pins (FTEST3)
   f. Control pins (RESET, MODE, INTLV)

Note: As for FP0 pin is always pull-up, the pin is output high-level while in HIGHZ instruction.

Note: Please note the input level to the analog input pins.

Note: The BSR of $\overline{\text{NMI}}$ pin has not an output function.

Note: The BSR of ENDIAN pin has an input/output function.

24.2 Signal Summary and Connection Example

The JTAG interface signals are listed below.

- TDI	JTAG serial data input
- TDO	JTAG serial data output
- TMS	JTAG test mode select
- TCK	JTAG serial clock input
- $\overline{\text{TRST}}$	JTAG test reset input
- BSC	ICE/JTAG test select input (compatible with the Enable signal) 0: ICE, 1: JTAG

The TMPM341FDXBG/FYXBG supports debugging by connecting the JTAG interface with a JTAG-compliant development tool.

For information about debugging, refer to the specification of the development tool used.

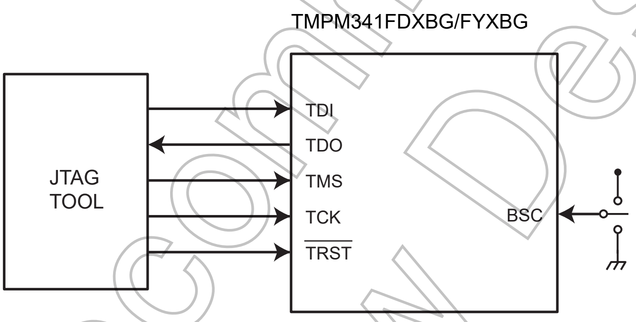


Figure 24-1 Example of connection with a JTAG development tool

Mode Setting Pin (BSC)	Operation mode
0	Set this pin to 0 except for Boundary Scan Mode. The TMPM341FDXBG/FYXBG operates as regular Debug Mode. Note: Debugging is not available if the internal BOOT is running.
1	The TMPM341FDXBG/FYXBG operates in Boundary Scan Mode.

24.3 What is Boundary Scan ?

With the evolution of ever-denser integrated circuits (ICs), surface-mounted devices, double-sided component mounting on printed-circuit boards (PCBs), and set-in recesses, in-circuit tests that depend upon physical contact like the connection of the internal board and chip has become more and more difficult to use. The more ICs have become complex, the larger and more difficult the test program became.

As one of the solutions, boundary-scan circuits started to be developed. A boundary-scan circuit is a series of shift register cells placed between the pins and the internal circuitry of the IC to which the said pins are connected. Normally, these boundary-scan cells are bypassed; when the IC enters test mode, however, the scan cells can be directed by the test program to pass data along the shift register path and perform various diagnostic tests. To accomplish this, the tests use the five signals, TCK, TMS, TDI, TDO and $\overline{\text{TRST}}$.

The JTAG boundary-scan mechanism (hereinafter referred to as JTAG mechanism in the chapter) allows testing of the connections between the processor, the printed circuit board to which it is attached, and the other components on the circuit board.

The JTAG mechanism cannot test the processor alone.

24.4 JTAG Controller and Registers

The processor contains the following JTAG controller and registers.

- Instruction register
- Boundary scan register
- Bypass register
- Device identification register
- Test Access Port (TAP) controller

JTAG basically operates to monitor the TMS input signal with the TAP controller state machine. When the monitoring starts, the TAP controller determines the test functionality to be implemented. This includes both loading the JTAG instruction register (IR) and beginning a serial data scan through a data register (DR), as shown in Table 24-1. As the data is scanned, the state of the TMS pin signals each new data word and indicates the end of the data stream. The data register is selected according to the contents of the instruction register.

24.5 Instruction Register

The JTAG instruction register includes four shift register-based cells. This register is used to select the test to be performed and/or the test data register to be accessed. As listed in Table 24-1, this instruction codes select either the boundary scan register or the bypass register.

Table 24-1 JTAG Instruction Register Bit Configuration

Instruction code (MSB to LSB)	Instruction	Selected data register
0000	EXTEST	Boundary scan register
0001	SAMPLE/PRELOAD	Boundary scan register
0100 to 1110	Reserved	Reserved
0010	HIGHZ	Bypass register
0011	CLAMP	Bypass register
1111	BYPASS	Bypass register

Figure 24-2 shows the format of the instruction register.

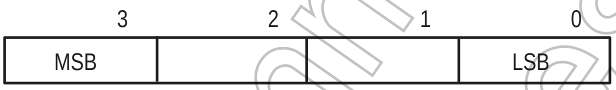


Figure 24-2 Instruction register

The instruction code is shifted out to the instruction register from the LSB.

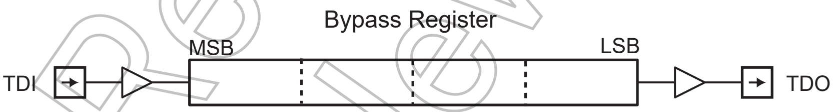


Figure 24-3 Instruction Register Shift Direction

The bypass register is 1 bit wide. When the TAP controller is in the Shift-DR (bypass) state, the data on the TDI pin is shifted into the bypass register, and the bypass register output shifts to the data out on the TDO output pin.

In essence, the bypass register is an alternative route which allows bypassing of board-level devices in the serial boundary-scan chain, which are not required for a specific test. The logical location of the bypass register in the boundary-scan chain is shown in Figure 24-4.

Use of the bypass register speeds up access to the boundary scan register in the IC that remains active in the board-level test data path.

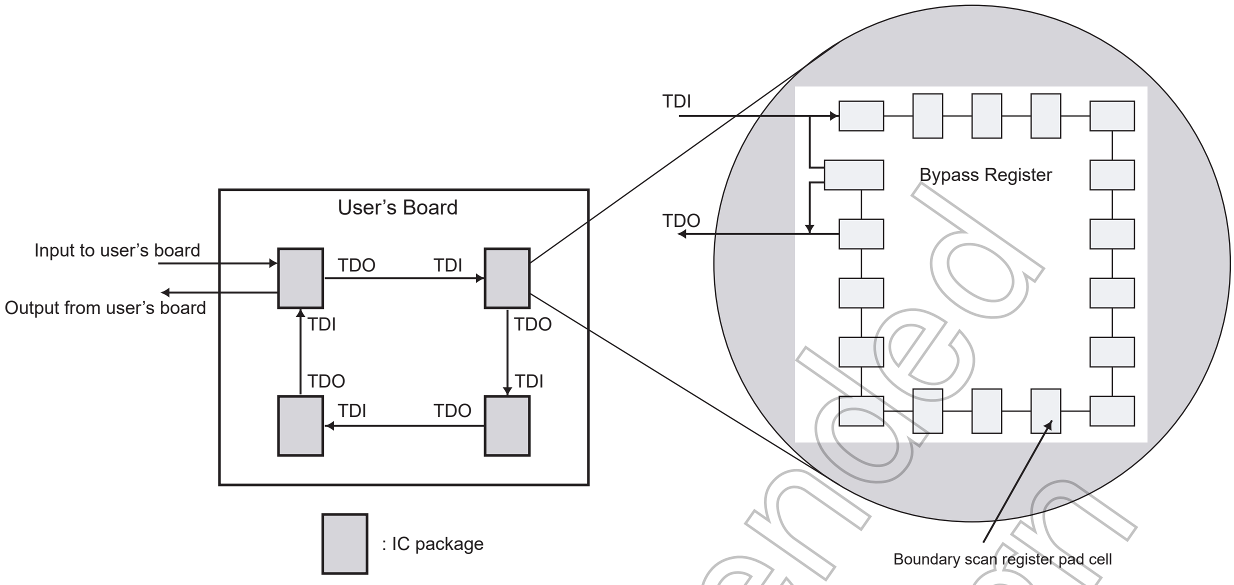


Figure 24-4 Bypass Register Operation

24.6 Boundary Scan Register

The boundary scan register provides all the inputs and outputs of the TMPM341FDXBG/FYXBG processor except some analog outputs and control signals. The pins of the TMPM341FDXBG/FYXBG allow any pattern to be driven by scanning the data into the boundary scan register in the Shift-DR state. Incoming data to the processor is examined by enabling the boundary scan register and shifting the data when the BSR is in the Capture-DR state.

The boundary scan register is a single, 231-bit-wide, shift register-based path containing cells connected to the input and output pads on the TMPM341FDXBG/FYXBG.

The TDI input is loaded to the LSB of the boundary scan register. The MSB of the boundary scan register is shifted out on the TDO output.

24.7 Test Access Port(TAP)

The Test Access Port (TAP) consists of the five signal pins: $\overline{\text{TRST}}$, TDI, TDO, TMS and TCK. These pins control a test by communicating the serial test data and instructions.

As Figure 24-5 shows, data is serially scanned into one of the three registers (instruction register, bypass register or boundary scan register) on the TDI pin, or it is scanned out from one of these three registers on the TDO pin.

The TMS input controls the state transitions of the main TAP controller state machine. The TCK input is a special test clock that allows serial JTAG data to be shifted synchronously, independent of any chip-specific or system clocks.

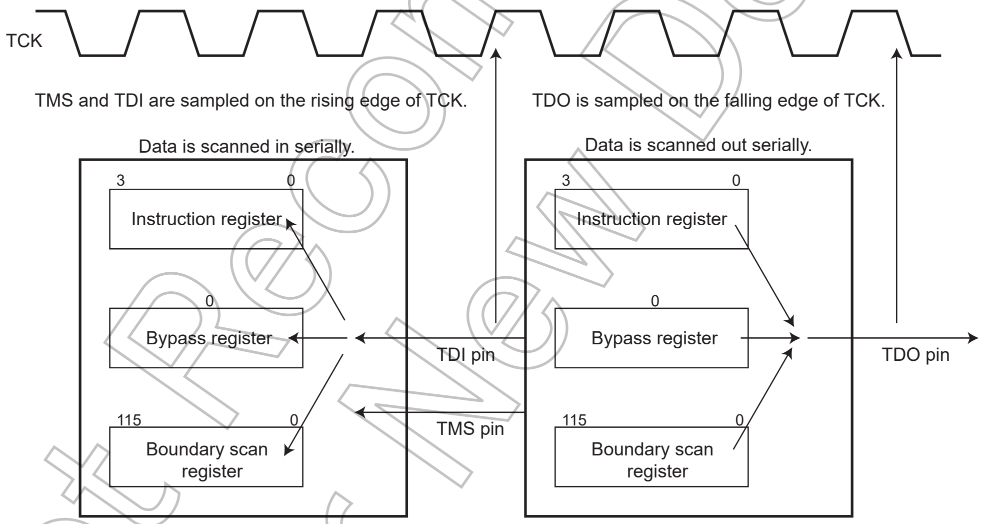


Figure 24-5 JTAG Test Access Port

Data on the TDI and TMS pins are sampled on the rising edge of the TCK input clock signal. Data on the TDO pin changes on the falling edge of the TCK clock signal.

24.8 TAP Controller

The processor incorporates the 16-state TAP controller stipulated in the IEEE JTAG specification.

24.9 Resetting the TAP Controller

The TAP controller state machine can be put into the Reset state by the following method.

Assertion of the $\overline{\text{TRST}}$ signal input (low) resets the TAP controller. After the processor reset state is released, keep the TMS input signal asserted through five consecutive rising edges of TCK input. Keeping TMS asserted maintains the Reset state.

24.10 State Transitions of the TAP Controller

The state transition diagram of the TAP controller is shown in Figure 24-6. Each arrow between states is labeled with a 1 or 0, indicating the logic value of TMS that must be set up before the rising edge of TCK to cause the transition.

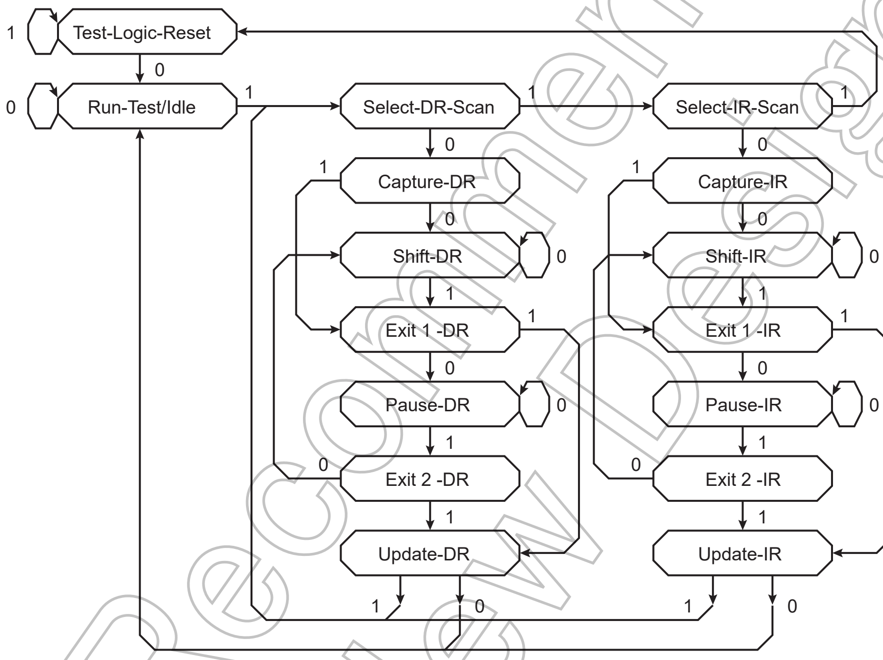


Figure 24-6 TAP Controller State Transition Diagram

The following paragraphs describe each of the controller states. The left column in Figure 24-6 is the data column, and the right column is the instruction column. The data column and instruction column reference the data register (DR) and the instruction register (IR), respectively.

- Test-Logic-Reset

  When the TAP controller is in the Reset state, the device identification register is selected by default. The MSB of the boundary scan register is cleared to 0 which disables the outputs.

  The TAP controller remains in this state while TMS is high. If TMS is held low while the TAP controller is in this state, then the controller moves to the Run-Test/Idle state.

- Run-Test/Idle

  In the Run-Test/Idle state, the IC is put in test mode only when certain instructions such as a built-in self test (BIST) instruction are present. For instructions that do not cause any activities in this state, all test data registers selected by the current instruction retain their previous states.

  The TAP controller remains in this state while TMS is held low. When TMS is held high, the controller moves to the Select-DR-Scan state.

- Select-DR-Scan

This is a temporary controller state. Here, the IC does not execute any specific functions.

If TMS is held low when the TAP controller is in this state, the controller moves to the Capture-DR state. If TMS is held high, the controller moves to the Select-IR-Scan state.

- Select-IR-Scan

This is a temporary controller state. Here, the IC does not execute any specific functions.

If TMS is held low when the TAP controller is in this state, the controller moves to the Capture-IR state. IF TMS is held high, the controller returns to the Test-Logic-Reset state.

- Capture-DR

In this state, if the test data register selected by the current instruction has parallel inputs, then data is parallel-loaded into the shift portion of the data register. If the test data register does not have parallel inputs, or if data needs not be loaded into the selected data register, then the data register retains its previous state.

If TMS is held low when the TAP controller is in this state, the controller moves to the Shift-DR state. If TMS is held high, the controller moves to the Exit 1-DR state.

- Shift-DR

In this controller state, the test data register connected between TDI and TDO shifts data out serially.

When the TAP controller is in this state, then it remains in the Shift-DR state if TMS is held low, or moves to the Exit 1-DR state if TMS is held high.

- Exit 1-DR

This is a temporary controller state.

If TMS is held low when the TAP controller is in this state, the controller moves to the Pause-DR state. If TMS is held high, the controller moves to the Update-DR state.

- Pause-DR

This state allows the shifting of the data register selected by the instruction register to be temporarily suspended. Both the instruction register and the data register retain their current states.

When the TAP controller is in this state, then it remains in the Pause-DR state if TMS is held low, or moves to the Exit 2-DR state.

- Exit 2-DR

This is a temporary controller state.

When the TAP controller is in this state, it returns to the Shift-DR state if TMS is held low, or moves on to the Update-DR state if TMS is held high.

- Update-DR

In this state, data is latched, on the rising edge of TCK, onto the parallel outputs of the data registers from the shift register path. The data held at the parallel output does not change while data is shifted in the associated shift register path.

When the TAP controller is in this state, it moves to either the Run-Test/Idle state if TMS is held low, or the Select-DR-Scan state if TMS is held high.

- Capture-IR

In this state, data is parallel-loaded into the instruction register. The data to be loaded is 0001. The Capture-IR state is used for testing the instruction register. Faults in the instruction register, if any, may be detected by shifting out the loaded data.

When the TAP controller is in this state, it moves to either the Shift-IR state if TMS is held low, or the Exit 1-IR state if TMS is high.

- Shift-IR

In this state, the instruction register is connected between TDI and TDO and shifts the captured data toward its serial output on the rising edge of TCK.

When the TAP controller is in this state, it remains in the Shift-IR state if TMS is low, or moves to the Exit 1-IR state if TMS is high.

- Exit 1-IR

This is a temporary controller state.

When the TAP controller is in this state, it moves to either the Pause-IR state if TMS is held low, or the Update-IR state if TMS is held high.

- Pause-IR

This state allows the shifting of the instruction register to be temporarily suspended. Both the instruction register and the data register retain their current states.

When the TAP controller is in this state, it remains in the Pause-IR state if TMS is held low, or moves to the Exit 2-IR state if TMS is held high.

- Exit 2-IR

This is a temporary controller state.

When the TAP controller is in this state, it moves to either the Shift-IR state if TMS is held low, or the Update-IR state if TMS is held high.

- Update-IR

This state allows the instruction previously shifted into the instruction register to be output in parallel on the rising edge of TCK. Then it becomes the current instruction, setting a new operational mode.

When the TAP controller is in this state, it moves to either the Run-Test/Idle state if TMS is low, or the Select-DR-Scan state if TMS is high.

24.11 Boundary Scan Order

The following table shows the boundary scan order with respect to the processor signals.

TDI → 1 (PI2)→ 2(PI1)→ − →82(PH5)→83(PH6)→TDO

Table 24-2 JTAG Scan Order of the TMPM341FDXBG/FYXBG Processor Pins

No.	Pin Name	No.	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
	TDI								
1	PI2	21	PE4	41	PB0	61	PK4	81	PH4
2	PI1	22	PE3	42	PA7	62	PK3	82	PH5
3	PI0	23	PE2	43	PA6	63	PK2	83	PH6
4	PG5	24	PE1	44	PA5	64	PK1		TDO
5	PG4	25	PE0	45	PA4	65	PK0		
6	PG3	26	PC7	46	PA3	66	PJ7		
7	PG2	27	PC6	47	PA2	67	PJ6		
8	PG1	28	PC5	48	PA1	68	PJ5		
9	PG0	29	PC4	49	PA0	69	PJ4		
10	PD7	30	PC3	50	PF7	70	PJ3		
11	PD6	31	PC2	51	PF6	71	PJ2		
12	PD5	32	PC1	52	PF5	72	PJ1		
13	PD4	33	PC0	53	PF4	73	PJ0		
14	PD3	34	PB7	54	PF3	74	PG7		
15	PD2	35	PB6	55	PF2	75	PG6		
16	PD1	36	PB5	56	PF1	76	PH2		
17	PD0	37	PB4	57	PF0	77	PH1		
18	PE7	38	PB3	58	ENDIAN	78	PH0		
19	PE6	39	PB2	59	PK6	79	$\overline{\text{NMI}}$		
20	PE5	40	PB1	60	PK5	80	PH3		

24.12 Instructions Supported by the JTAG Controller Cells

This section describes the instructions supported by the JTAG controller cells of the TMPM341FDXBG/FYXBG.

1. EXTEST instruction

The EXTEST instruction is used for external interconnect tests. The EXTEST instruction permits BSR cells at output pins to shift out test patterns in the Update-DR state and those at input pins to capture test results in the Capture-DR state.

Typically, before EXTEST is executed, the initialization pattern is shifted into the boundary scan register using the SAMPLE/PRELOAD instruction. If the boundary scan register is not reset, indeterminate data will be transferred in the Update-DR state and bus conflicts between ICs may occur. Figure 24-7 shows data flow when the EXTEST instruction is selected.

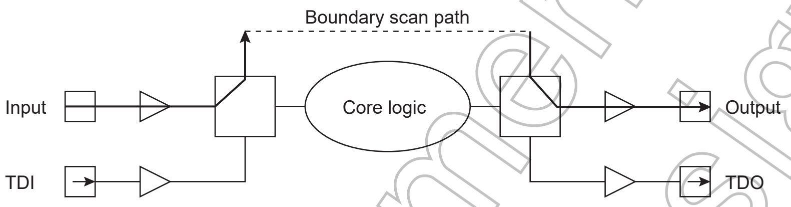


Figure 24-7 Test Data Flow when the EXTEST Instruction is Selected

The following steps describe the basic test procedure of the external interconnect test.

1. Reset the TAP controller to the Test-Logic-Reset state.
2. Load the instruction register with the SAMPLE/PRELOAD instruction. This causes the boundary scan register to be connected between TDI and TDO.
3. Reset the boundary scan register by shifting certain data in.
4. Load the test pattern into the boundary scan register.
5. Load the instruction register with the EXTEST instruction.
6. Capture the data applied to the input pin into the boundary scan register.
7. Shift out the captured data while simultaneously shifting the next test pattern in.
8. Send out the test pattern in the boundary scan register at the output on the output pin.

Repeat steps 6 to 8 for each test pattern.

2. SAMPLE/PRELOAD instruction

This instruction targets the boundary scan register between TDI and TDO. As its name implies, the SAMPLE/PRELOAD instruction provides two functions.

SAMPLE allows the input and output pads of an IC to be monitored. While it does so, it does not disconnect the system logic from the IC pins. SAMPLE is executed in the Capture-DR state. It is mainly used to capture the values of the IC's I/O pins on the rising edge of TCK during normal operation. Figure 24-8 shows the flow of data for the SAMPLE phase of the SAMPLE/PRELOAD instruction.

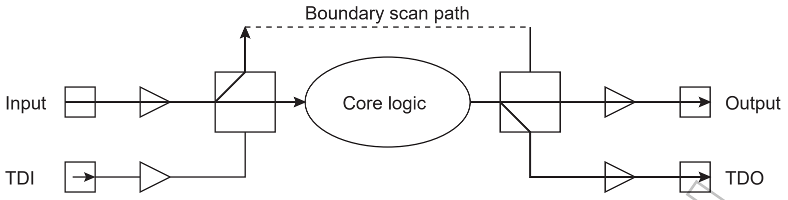


Figure 24-8 Test Data Flow while the SAMPLE is Selected

PRELOAD allows the boundary scan register to be reset before any other instruction is selected. For example, prior to selection of the EXTEST instruction, PRELOAD is used to load reset data into the boundary scan register. PRELOAD permits data shifting of the boundary scan register without interfering with the normal operation of the system logic. Figure 24-9 shows the data flow for the PRELOAD phase of the SAMPLE/PRELOAD instruction.

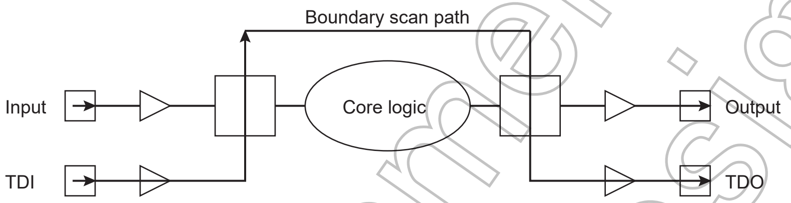


Figure 24-9 Test Data Flow while PRELOAD is Selected

3. BYPASS instruction

This instruction targets the bypass register between JTDI and JTDO. The bypass register provides the shortest serial path that bypasses the IC (between JTDI and JTDO) when the test does not require control or monitoring of the IC. The BYPASS instruction does not cause interference in the normal operation of the on-chip system logic. Figure 24-10 shows the data flow through the bypass register when the BYPASS instruction is selected.

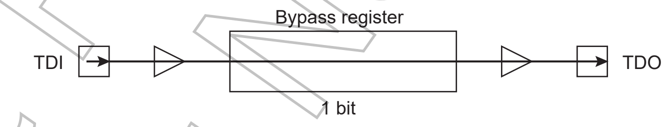


Figure 24-10 Test Data Flow when the BYPASS Instruction is Selected

4. CLAMP instruction

The CLAMP instruction outputs the value that boundary scan register is programmed according to the PRELOAD instruction, and execute Bypass operation.

The CLAMP instruction selects the bypass register between TDI and TDO.

5. HIGHZ instruction

   The HIGHZ instruction disables the output of the internal logical circuits. When the HIGHZ instruction is executed, it places the 3-state output pins in the high-impedance state.

   The HIGHZ instruction also selects the bypass register between TDI and TDO.

・ Notes

This section describes the cautions of the JTAG boundary-scan operations specific to the processor.

1. As for a PF0 pin is always pull-up, whenever HIGHZ is ordered, High is output.

2. Please note the input level to the analog input pins.

3. The JTAG circuit can be released from the reset state by either of the following two methods:

   Assert $\overline{\text{TRST}}$, initialize the JTAG circuit, and then deassertion $\overline{\text{TRST}}$.

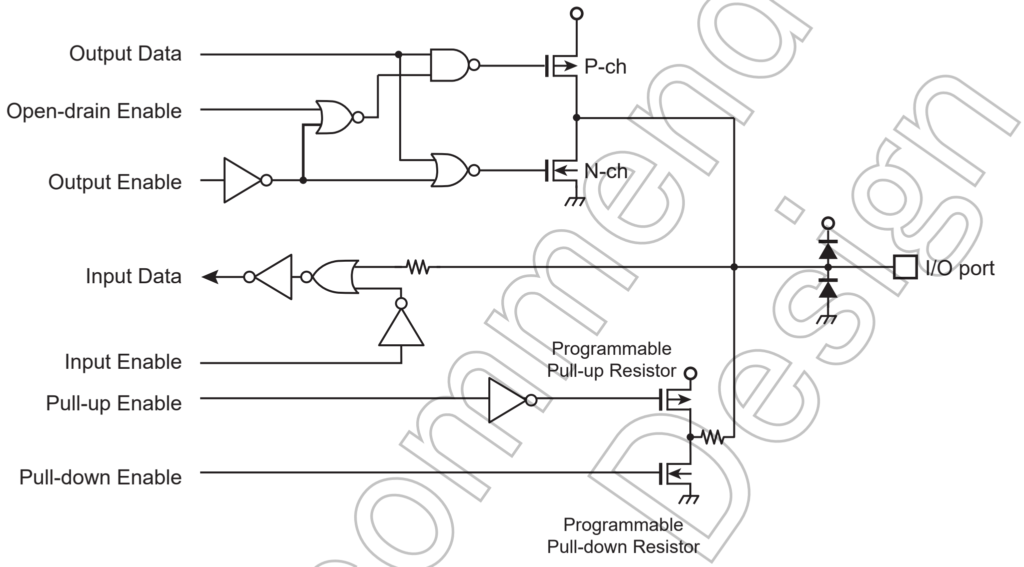
   Supply the TCK signal for 5 or more clock pulses to TCK while pulling the TMS pin High.

25. Port Section Equivalent Circuit Schematic

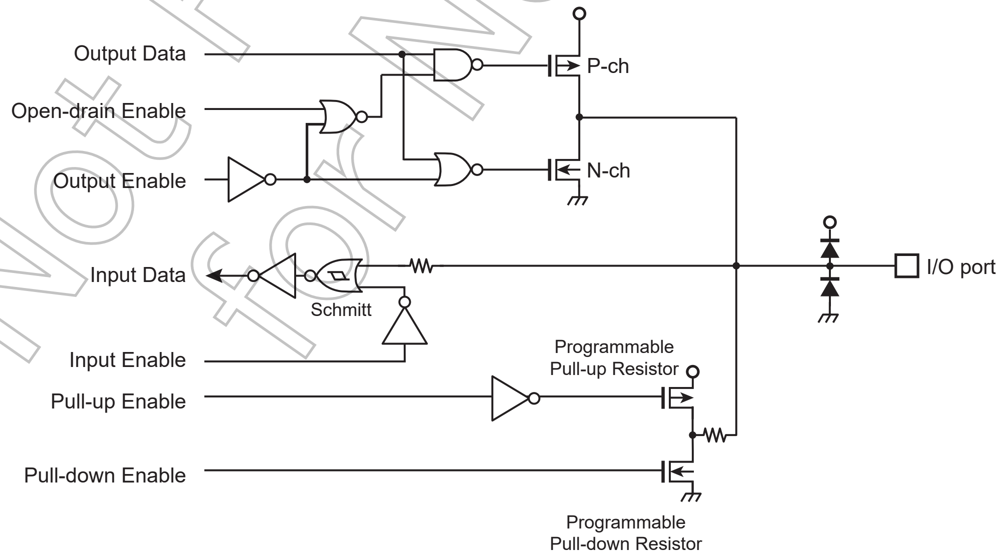
Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

The input protection resistance ranges from several tens of Ω to several hundreds of Ω.Feedback resistor and Damping resistor are shown with a typical value.

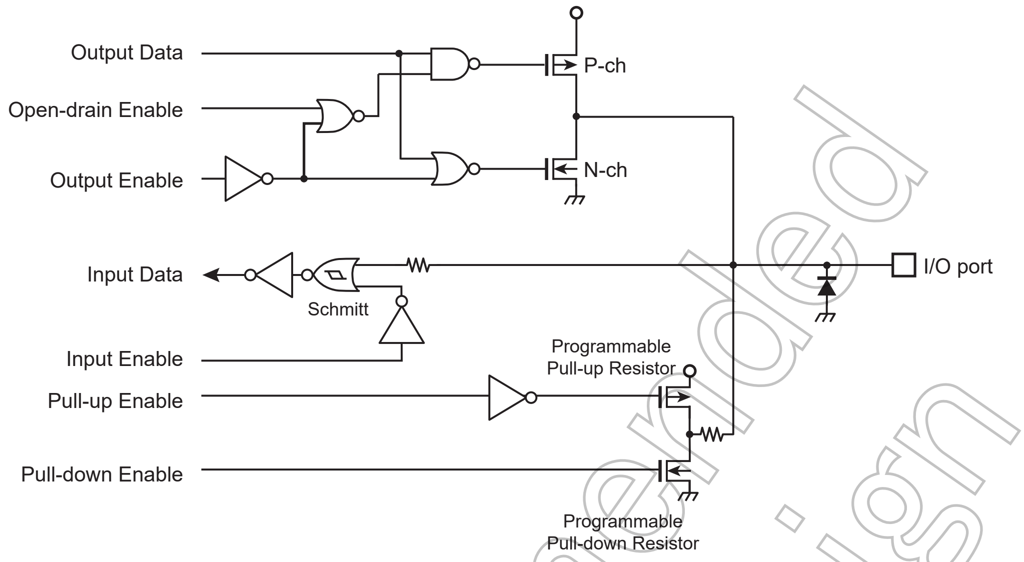
25.1 PA0 to 7,PB0 to 7



25.2 PC0 to 7,PD0 to 6,PE0 to 7,PF1 to 7,PH3 to 6,PI0 to 7

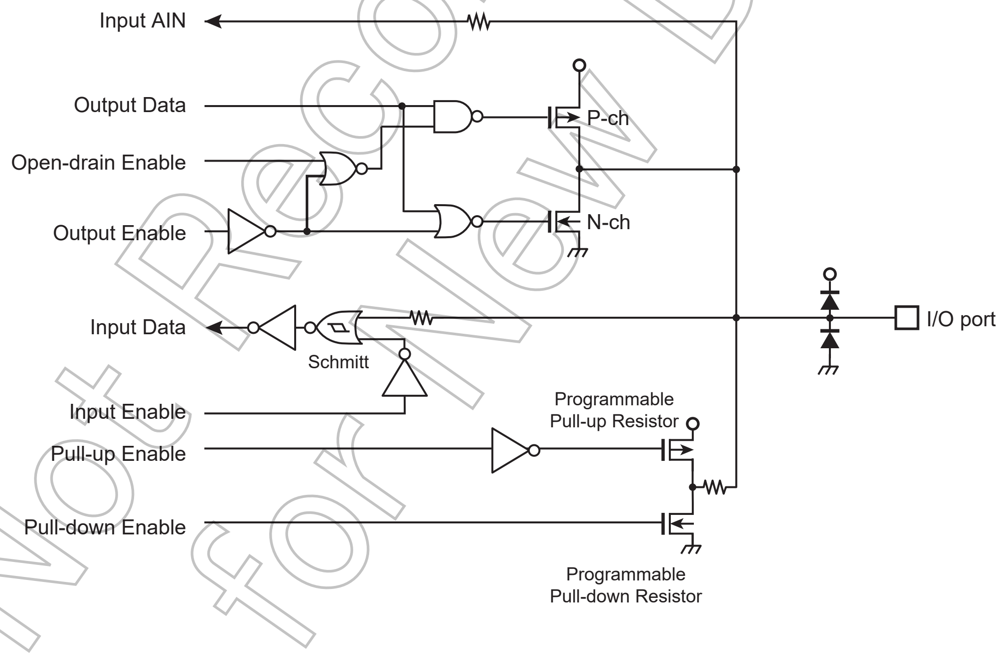


25.3 PG0 to 7,PH0 to 2

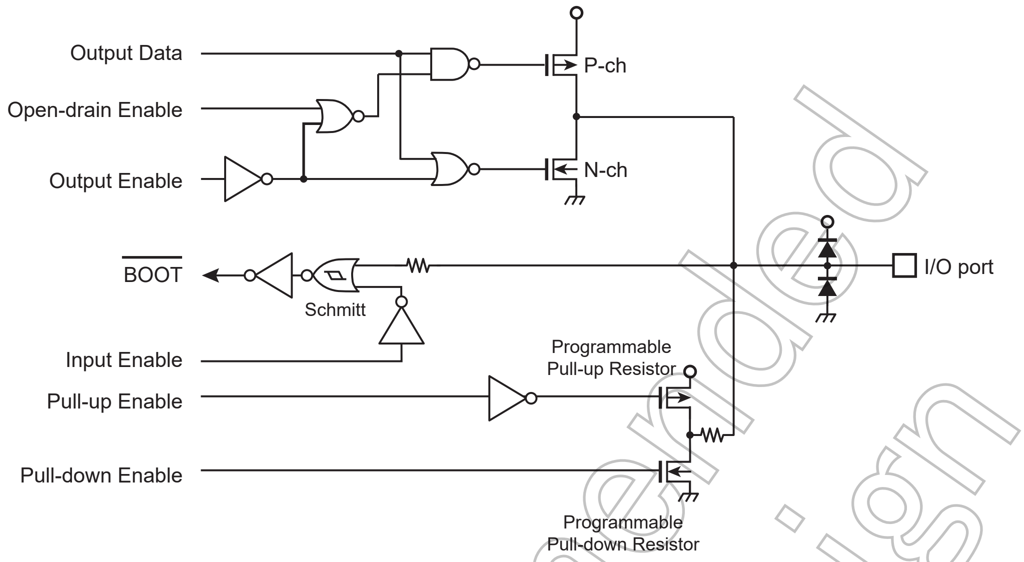


Note: Only when input is enabled, these pins tolerate 5V inputs.

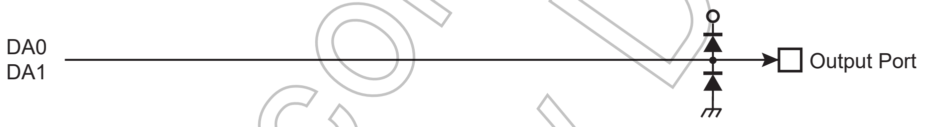
25.4 PJ0 to 7,PK0 to 7



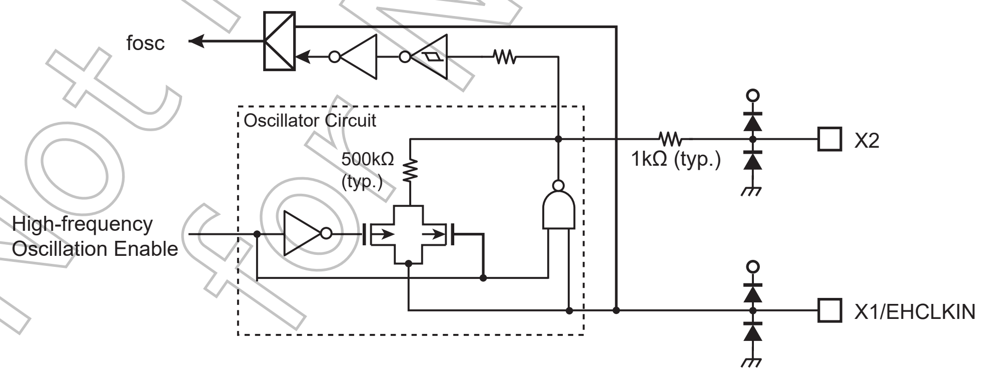
25.5 PF0



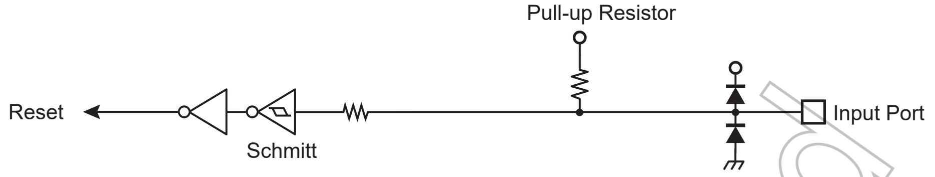
25.6 DA0,DA1



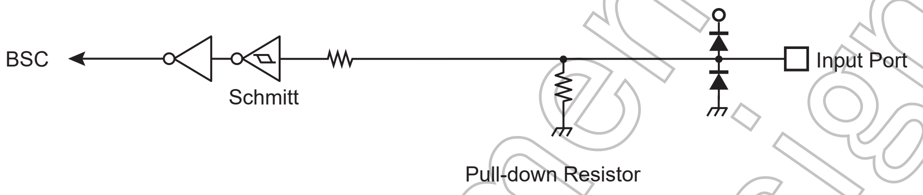
25.7 X1,X2



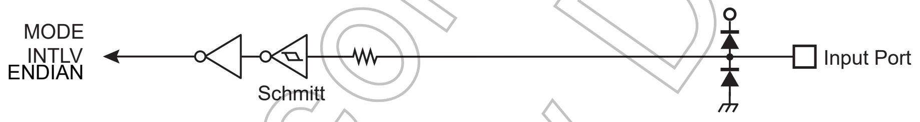
25.8 $\overline{\text{RESET}}$,$\overline{\text{NMI}}$



25.9 BSC

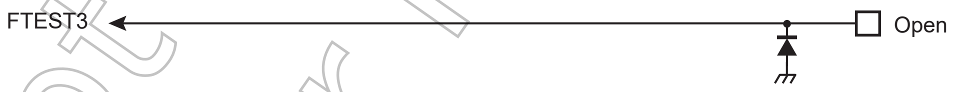


25.10 MODE,INTLV,ENDIAN



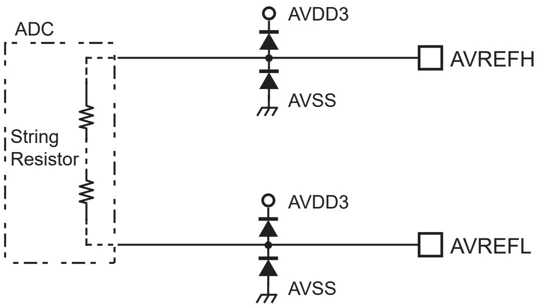
(Note)MODE pin is fixed to GND.

25.11 FTEST3



(Note)FTEST3 pin is fixed to Open.

25.12 AVREFH,AVREFL



26. Electrical Characteristics

26.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		DVDD3A	−0.3 to 3.9	V
		DVDD3B	−0.3 to 3.9	
		AVDD3	−0.3 to 3.9	
		RVDD3	−0.3 to 3.9	
Input voltage	Except below pins	V_{IN1}	−0.3 to VDD + 0.3	V
	PG7 to PG0, PH2 to PH0	V_{IN2}	−0.3 to 5.5	
Low-level output current	Per pin	I_{OL}	5	mA
	Total (Note)	ΣI_{OL}	50	
High-level output current	Per pin	I_{OH}	−5	
	Total (Note)	ΣI_{OH}	50	
Power consumption (Ta = 85 °C)		PD	600	mW
Soldering temperature(10 s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	−40 to 125	°C
Operating Temperature	Except during Flash W/E	T_{OPR}	−20 to 85	°C
	During Flash W/E		0 to 70	

Note 1: Each I/O supplied via DVDD3A and DVDD3B

Note 2: **Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no Absolute maximum rating value is exceeded with respect to current, voltage, power consumption, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning.**

26.2 DC Electrical Characteristics (1/3)

Ta = −40 ~ 85 °C

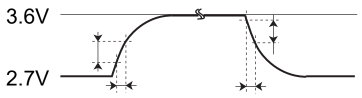
Parameter		Symbol	Condition	Min.	Typ. (Note 1)	Max.	Unit
Supply voltage	DVDD3A DVDD3B AVDD3 RVDD3 DVSSA = DVSSB = AVSS = RVSS = DVSSC = 0V	DVDD3A AVDD3 RVDD3 (Note 2)	f_{OSC} = 8 to 27 MHz fsys = 1 to 54 MHz	2.7	–	3.6	V
		DVDD3B		1.65	–	3.6	V
Low-level input voltage	PA, PB, PC, PD, PE, PF	V_{IL1}	1.65 V ≤ DVDD3B ≤ 3.6 V	−0.3	–	0.2 DVDD3B	V
	PG, PH, PI, PJ, PK	V_{IL2}	2.7 V ≤ DVDD3A ≤ 3.6 V (Include 5 V tolerant input pins)			0.2 DVDD3A	
	$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, MODE, INTLV, BSC, ENDIAN	V_{IL3}					
	X1	V_{IL4}	2.7 V ≤ RVDD3 ≤ 3.6 V			0.2 RVDD3	
High-level input voltage	PA, PB, PC, PD, PE, PF	V_{IH1}	1.65 V ≤ DVDD3B ≤ 3.6 V	0.8 DVDD3B	–	DVDD3B + 0.3	V
	PH6 to PH3, PI, PJ, PK	V_{IH2}	2.7 V ≤ DVDD3A ≤ 3.6 V	0.8 DVDD3A		DVDD3A + 0.3	
	PG7 to PG0, PH2 to PH0	V_{IH5}				5.5	
	$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, MODE, INTLV, BSC, ENDIAN	V_{IH3}	2.7 V ≤ DVDD3A ≤ 3.6 V			DVDD3A + 0.3	
	X1	V_{IH4}	2.7 V ≤ RVDD3 ≤ 3.6 V	0.8 RVDD3		RVDD3 + 0.3	
Low-level output voltage	PG, PH, PI, PJ, PK	V_{OL}	I_{OL} = 2 mA, 2.7 V ≤ DVDD3A ≤ 3.6 V	–	–	0.4	V
	PA, PB, PC, PD, PE, PF		I_{OL} = 2 mA, 2.7 V ≤ DVDD3B≤ 3.6 V			0.4	
	PA, PB, PC, PD, PE, PF		I_{OL} = 2 mA, 1.65 V ≤ DVDD3B≤ 3.6 V			0.2	
High-level output voltage	PH6 to PH3, PI, PJ, PK	V_{OH}	I_{OH} = −2 mA, 2.7 V ≤ DVDD3A≤ 3.6 V	2.4	–	DVDD3A	V
	PG7 to PG0, PH2 to PH0			2.4	–		
	PA, PB, PC, PD, PE, PF		I_{OH} = −2 mA, 2.7 V ≤ DVDD3B≤ 3.6 V	2.4	–	DVDD3B	
	PA, PB, PC, PD, PE, PF		I_{OH} = −0.5 mA, 1.65 V ≤ DVDD3B≤ 3.6 V	1.5	–	DVDD3B	
Input leakage current		I_{LI1}	0.0 ≤ V_{IN} ≤ DVDD3A, DVDD3B 0.0 ≤ V_{IN} ≤ AVDD3	–	0.02	±5	μA
Output leakage current		I_{LO}	0.2 ≤ V_{IN} ≤ DVDD3A-0.2, DVDD3B − 0.2 0.2 ≤ V_{IN} ≤ AVDD3 − 0.2	–	0.05	±10	
Pull-up resister at Reset		RRST	2.7 V ≤ DVDD3A, DVDD3B ≤ 3.6 V	–	50	150	kΩ
Hysteresis voltage		VTH1	2.7 V ≤ DVDD3A, DVDD3B ≤ 3.6 V	0.3	0.6	–	V
		VTH2	1.65 V ≤ DVDD3B ≤ 2.7 V	0.14	0.19		
Programmable pull-up/pull-down resistor		PKH	2.7 V ≤ DVDD3A, DVDD3B ≤ 3.6 V	–	50	150	kΩ
Voltage fluctuation rate in the operation range		VRS	RVDD3 = DVDD3A	–	–	23	mV/μs
		VFS		–	–	40	
Pin capacitance (Except power supply pins)		C_{IO}	fc = 1 MHz	–	–	10	pF

Note 1: Ta = 25 °C, DVDD3A = DVDD3B = RVDD3 = AVDD3 = 3.3 V, unless otherwise noted.

Note 2: The same voltage must be supplied to DVDD3A, AVDD3, and RVDD3.

Note 3: Ensure that all power supply source, including DVDD3B, is power-off and then power-on again when DVDD3A, RVDD3 and AVDD3 falls below 2.7V which is minimum operation voltage.

Note 4: The fluctuation rate of both VRS (increasing voltage) and VFS (decreasing voltage) should measure the exact point about the specification.



The fluctuation rate of both VRS(increasing voltage) and VFS(decreasing voltage) should measure the exact point about the specification.

26.3 DC Electrical Characteristics (2/3)

DVDD3A = AVDD3 = RVDD3 = 2.7 V to 3.6 V,
DVDD3B = 1.65V to 3.6V,
DVSSA = DVSSB = RVSS = DVSSC = AVSS
Ta = −40 ~ 85 °C

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Low-level output current	I_{OL1}	Per pin 2.7 V ≤ DVDD3A ≤ 3.6 V PH3 to PH6, PI0 to PI7, PJ0 to PJ7, PK0 to PK6	−	−	2	mA
	I_{OL2}	Per pin 2.7 V ≤ DVDD3A ≤ 3.6 V PG0 to PG7, PH0 to PH2	−	−	2	
	I_{OL3}	Per pin 1.65 V ≤ DVDD3B ≤ 3.6 V PA0 to PA7, PB0 to PB7, PC0 to PC7, PD0 to PD7, PE0 to PE7, PF0 to PF7	−	−	2	
	ΣI_{OL1}	Per group, Port A	−	−	10	
	ΣI_{OL2}	Per group, Port B	−	−	10	
	ΣI_{OL3}	Per group, Port C	−	−	10	
	ΣI_{OL4}	Per group, Port D	−	−	10	
	ΣI_{OL5}	Per group, Port E	−	−	10	
	ΣI_{OL6}	Per group, Port F	−	−	10	
	ΣI_{OL7}	Per group, Port G	−	−	20	
	ΣI_{OL8}	Per group, PH0 to PH2	−	−	10	
	ΣI_{OL9}	Per group, PH3 to PH6	−	−	10	
	ΣI_{OL10}	Per group, Port I	−	−	10	
	ΣI_{OL11}	Per group, Port J	−	−	10	
	ΣI_{OL12}	Per group, Port K	−	−	10	
	ΣI_{OL}	Total, all Port	−	−	35	mA

DVDD3A = AVDD3 = RVDD3 = 2.7 V to 3.6 V,
DVDD3B = 1.65V to 3.6V,
DVSSA = DVSSB = RVSS = DVSSC = AVSS
Ta = −40 ~ 85 °C

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
High-level output current	I_{OH1}	Per pin 2.7 V ≤ DVDD3A, DVDD3B ≤ 3.6 V PH3 to PH6, PI0 to PI7, PJ0 to PJ7, PK0 to PK6	–	–	-2	mA
	I_{OH2}	Per pin 2.7 V ≤ DVDD3A, DVDD3B ≤ 3.6 V PH0 to PH2, PG0 to PG7	–	–	-2	
	I_{OH3}	Per pin 1.65 V ≤ DVDD3B ≤ 3.6 V <PA0 to PA7, PB0 to PB7, PC0 to PC7, PD0 to PD7, PE0 to PE7, PF0 to PF7>	–	–	-2	
	ΣI_{OH1}	Per group, Port A	–	–	-10	
	ΣI_{OH2}	Per group, Port B	–	–	-10	
	ΣI_{OH3}	Per group, Port C	–	–	-10	
	ΣI_{OH4}	Per group, Port D	–	–	-10	
	ΣI_{OH5}	Per group, Port E	–	–	-10	
	ΣI_{OH6}	Per group, Port F	–	–	-10	
	ΣI_{OH7}	Per group, Port G	–	–	-10	
	ΣI_{OH8}	Per group, PH0 to PH2	–	–	-10	
	ΣI_{OH9}	Per group, PH3 to PH6	–	–	-10	
	ΣI_{OH10}	Per group, Port I	–	–	-10	
	ΣI_{OH11}	Per group, Port J	–	–	-10	
	ΣI_{OH12}	Per group, Port K	–	–	-10	
	ΣI_{OH}	Total, all Port	–	–	-35	mA

Note 1: The same voltage must be supplied to DVDD3A, AVDD3, and RVDD3.

Note 2: The total of port output current is accumulated each voltage supply.

26.4 DC Electrical Characteristics (3/3)

DVDD3A = DVDD3B = AVDD3 = RVDD3 = 2.7 V to 3.6 V
Ta = −40 to 85 °C

Parameter	Symbol	Condition	Min.	Typ. (Note1)	Max.	Unit
NORMAL (Note2) Gear 1/1	I_{DD}	fsys = 54 MHz	−	42	50	mA
IDLE (Note3)			−	18	25	
NORMAL (Note2) Gear 1/1		fsys = 40 MHz	−	33	42	
IDLE (Note3)			−	14	20	
STOP1		−	−	60	1000	μA
STOP2		−	−	2.5	60	

Note 1: Ta = 25 °C, DVDD3A = DVDD3B = AVDD3 = RVDD3 = 3.3 V, unless otherwise noted.

Note 2: I_{DD} NORMAL: Measurement condition

Operation program : The Dhrystone ver. 2.1 operated in FLASH.

Operation peripheral : All functions operates excluding ADC and DAC

Note 3: I_{DD} IDLE: Measurement condition

Operation peripheral : All Peripheral stopped

The currents flow through DVDD3A, DVDD3B, AVDD3 and RVDD3 are included.

26.5 12-bit ADC Electrical Characteristics

DVDD3A = DVDD3B = AVDD3 = RVDD3 = AVREFH = 2.7 V to 3.6 V
AVSS = DVSSA = DVSSB = DVSSC = AVREFL = 0V
Ta = −40 to 85 °C

Parameter		Symbol	Condition	Min.	Typ.	Max.	Unit
Analog reference voltage(+)		AVREFH	−	2.7	3.3	3.6	V
Analog input voltage		VAIN	−	AVREFL	−	AVREFH	V
Power supply current of analog reference voltage	AD conversion	IREF	DVSSA = DVSSB = DVSSC = AVSS	−	2.0	2.5	mA
	Non-AD conversion			−	−	0.1	μA
INL error		−	AIN resistance ≤ 600 Ω AIN load capacitance ≥ 30 pF Conversion time ≥ 2.0 μs	−	± 4	−	LSB
DNL error				−	± 2	−	
Zero-scale error				−	± 3	−	
Full-scale error				−	± 4	−	
Total error				−	± 6	−	
INL error		−	AIN resistance ≤ 1 kΩ AIN load capacitance ≥ 0.1μF Conversion time ≥ 1.0 μs	−	± 4	−	
DNL error				−	± 2	−	
Zero-scale error				−	± 3	−	
Full-scale error				−	± 4	−	
Total error				−	± 6	−	
INL error		−	AIN resistance ≤ 10 kΩ AIN load capacitance ≥ 0.1μF Conversion time ≥ 1.0 μs	−	± 4	−	
DNL error				−	± 2	−	
Zero-scale error				−	± 3	−	
Full-scale error				−	± 4	−	
Total error				−	± 6	−	
Conversion time		Tconv	−	1.0	−	10	μs

Note 1: 1LSB = (AVREFH − AVREFL)/4096 [V]

Note 2: Peripheral functions are disable.

Note 3: An AVREFL of analog input voltage must be applied exact same voltage level between AVSS, DVSSA, DVSSB and DVSSC.

Note 4: The total error is a combination of actual values of zero-scale error, full-size error and INL error, and is the maximum difference between the actual values and the theoretical figure (quantum error is not included).

26.6 10-bit DAC Electrical Characteristics

DVDD3A = AVDD3 = RVDD3 = AVREFH = 2.7 V to 3.6 V
DVDD3B = 1.65 V to 3.6 V
AVSS = DVSSA = DVSSB = DVSSC = AVREFL = 0V
Ta = −40 to 85 °C

Parameter		Symbol	Condition	Min.	Typ.	Max.	Unit
Analog reference voltage(+)		DAVREF	−	2.7	3.3	3.6	V
Power supply current of analog reference voltage	<VREFON>=1 ch = 0,1	IDREF	−	−	4	10	μA
	<VREFON>=0			−	0.3	1	
Settling time		Tset	−	−	−	100	μs
Output current		IDA0, IDA1	−	−	−	± 500	μA
Range of output voltage		DA0, DA1	−	AVSS + 0.3	−	AVDD3 − 0.3	V
Full-scale error		TERR	−	−	± 2	± 4	LSB

Note 1: 1LSB = (AVREFH − AVREFL)/1024 [V]

Note 2: IDREF current value is in the dual channel operation.

Note 3: No guarantee about relative accuracy in the dual channel operation.

Note 4: Load capacitance of each DAx pin is 100pF (max.).

Note 5: The total error is the output voltage error introduced in the theoretical output voltage ((AVREFH - AVREFL)/1024 × DAC).

26.7 AC Electrical Characteristics

26.7.1 AC measurement condition

The AC characteristics data of this chapter is measured under the following conditions unless otherwise noted

- Output levels: High = 0.8 × DVDD3A, 0.8 × DVDD3B
- Output levels: Low = 0.2 × DVDD3A, 0.2 × DVDD3B
- Input levels: Refer to low-level input voltage and high-level input voltage in "DC Electrical Characteristics".
- Load capacity: CL = 30pF
- Ambient Temp.: Ta = − 40 to 85 °C

26.7.2 Serial Channel (SIO/UART)

26.7.2.1 I/O Interface mode

In the table below, the letter x represents the SIO operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

(1) SCLK input mode

[Data Input]

Parameter	Symbol	Equation		40 MHz		54 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
SCLK Clock High width (input)	t_{SCH}	4x	−	100	−	74.1	−	ns
SCLK Clock Low width (input)	t_{SCL}	4x	−	100	−	74.1	−	
SCLK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	−	200	−	148	−	
Valid Data input → SCLK rise or fall (Note1)	t_{SRD}	30	−	30.0	−	30.0	−	
SCLK rise or fall (Note1) → Input Data hold	t_{HSR}	x + 30	−	55.0	−	48.5	−	

[Data Output]

Parameter	Symbol	Equation		40 MHz		54 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
SCLK Clock High width (input)	t_{SCH}	4x	−	120 (Note3)	−	101 (Note3)	−	ns
SCLK Clock Low width (input)	t_{SCL}	4x	−	120 (Note3)	−	101 (Note3)	−	
SCLK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	−	240	−	201	−	
Output Data → SCLK rise or fall (Note 1)	t_{OSS}	$t_{SCY}/2 - 3x - 45$	−	0.00 (Note2)	−	0.00 (Note2)	−	
SCLK rise or fall (Note1) → Output Data hold	t_{OHS}	$t_{SCY}/2$	−	120	−	101	−	

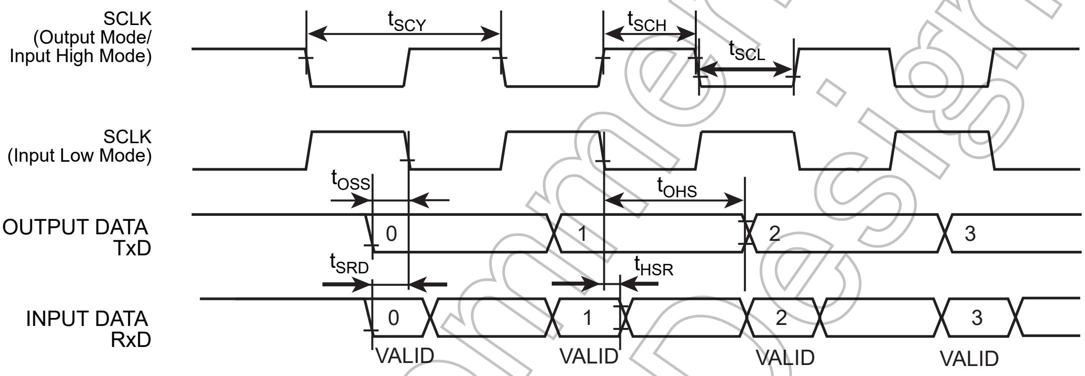
Note 1: SCLK rise/fall : SCLK rise mode uses the rise timing of SCLK. SCLK fall mode uses the fall timing of SCLK.

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 the zero or more.

(2) SCLK output mode

Parameter	Symbol	Equation		40 MHz		54 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
SCLK cycle (programmable)	t_{SCY}	4x	−	100	−	74	−	ns
Output Data ← SCLK rise	t_{OSS}	t_{SCY}/2 − 30	−	20	−	7	−	
SCLK rise → Output Data hold	t_{OHS}	t_{SCY}/2 − 30	−	20	−	7	−	
Valid Data Input → SCLK rise	t_{SRD}	45	−	45	−	45	−	
SCLK rise → Input Data hold	t_{HSR}	0	−	0	−	0	−	



26.7.3 Serial Bus Interface (I2C/SIO)

26.7.3.1 I2C Mode

In the table below, the letter x represents the I2C operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

n denotes the value of n programmed into the SCK (SCL output frequency select) field in the SBIxCR.

Parameter	Symbol	Equation		Standard Mode		Fast Mode		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
SCL clock frequency	t_{SCL}	0	−	0	100	0	400	kHz
Hold time for START condition	$t_{HD;\ STA}$	−	−	4.0	−	0.6	−	μs
SCL Low width (Input) (Note 1)	t_{LOW}	−	−	4.7	−	1.3	−	μs
SCL High width (Input) (Note 2)	t_{HIGH}	−	−	4.0	−	0.6	−	μs
Setup time for a repeated START condition	$t_{SU;\ STA}$	(Note5)	−	4.7	−	0.6	−	μs
Data hold time (Input) (Note 3, 4)	$t_{HD;\ DAT}$	−	−	0.0	−	0.0	−	μs
Data setup time	$t_{SU;\ DAT}$	−	−	250	−	100	−	ns
Setup time for a STOP condition	$t_{SU;\ STO}$	−	−	4.0	−	0.6	−	μs
Bus free time between stop condition and start condition	t_{BUF}	(Note5)	−	4.7	−	1.3	−	μs

Note 1: SCL clock Low width (output) = $(2^{n-1} + 58)/x$

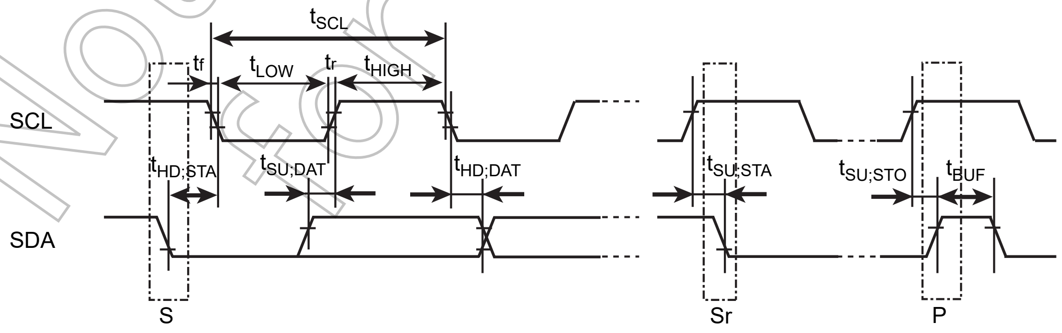
Note 2: SCL clock High width (output) = $(2^{n-1} + 14)/x$ On I2C-bus specification, Maximum Speed of Standard Mode is 100kHz, Fast mode is 400khz. Internal SCL Frequency setting should comply with Note1 & Note2 shown above.

Note 3: The output data hold time is equal to 4x of internal SCL.

Note 4: The Philips I2C-bus specification states that a device must internally provide a hold time of at least 300 ns for the SDA signal to bridge the undefined region of the falling edge of SCL. However, this SBI does not satisfy this requirement. Also, the output buffer for SCL does not incorporate slope control of the falling edges; therefore, the equipment manufacturer should design so that the input data hold time shown in the table is satisfied, including tr/tf of the SCL and SDA lines.

Note 5: Software -dependent

Note 6: The Philips I2C-bus specification instructs that if the power supply to a Fast-mode device is switched off, the SDA and SCL I/O pins must be floating so that they don't obstruct the bus lines. However, this SBI does not satisfy this requirement.



S: Start condition
Sr: Repeated start condition
P: Stop condition

26.7.3.2 Clock-Synchronous 8-Bit SIO mode

In the table below, the letter x represents the SBI operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

(1) SCK Input Mode (The electrical specifications below are for an SCK signal with a 50% duty cycle.)

[Data input]

Parameter	Symbol	Equation		40 MHz		54 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
SCK Clock High width (input)	t_{SCH}	4x	–	100	–	74	–	ns
SCK Clock Low width (input)	t_{SCL}	4x	–	100	–	74	–	
SCK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	–	200	–	148	–	
SCK rise→ Valid Data input	t_{SRD}	30 − x	–	5	–	11.5	–	
SCK rise → Input Data hold	t_{HSR}	2x + 30	–	80	–	67	–	

[Data output]

Parameter	Symbol	Equation		40 MHz		54 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
SCK Clock High width (input)	t_{SCH}	4x	–	120 (Note2)	–	100.6 (Note2)	–	ns
SCK Clock Low width (input)	t_{SCL}	4x	–	120 (Note2)	–	100.6 (Note2)	–	
SCK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	–	240	–	201	–	
SCK rise → Output Data	t_{OSS}	$t_{SCY}/2 - 3x - 45$	–	0 (Note1)	–	0 (Note1)	–	
SCK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 + x$	–	145	–	119.1	–	

Note 1: Keep this value positive by adjusting SCK cycle.

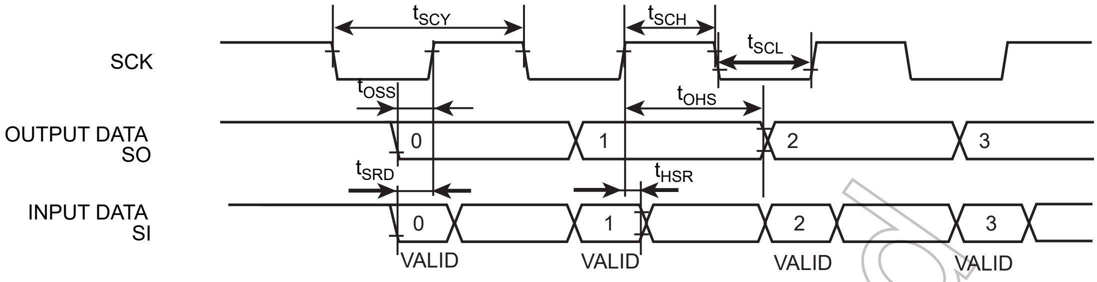
Note 2: The value indicates a minimum value that enables t_{OSS} to the zero or more.

(2) SCK Output Mode (The electrical specifications below are for an SCK signal with a 50% duty cycle.)

Parameter	Symbol	Equation		40 MHz		54 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
SCK cycle (programmable)	t_{SCY}	16x	–	400	–	296	–	ns
Output Data → SCK rise	t_{OSS}	$t_{SCY}/2 - 30$	–	170	–	118	–	
SCK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 - 30$	–	170	–	118	–	
Valid Data input → SCK rise	t_{SRD}	x + 45	–	70	–	63.5	–	
SCK rise → Input Data hold	t_{HSR}	0	–	0	–	0	–	

Note 1: SCK cycle after automatic wait becomes 14x.

Note 2: t_{OSS} after automatic wait may be t_{SCY}/2-x-20.


t_{SCY}
t_{SCH}
t_{SCL}
SCK
t_{OSS}
t_{OHS}
OUTPUT DATA
SO
0
1
2
3
t_{SRD}
t_{HSR}
INPUT DATA
SI
0
1
2
3
VALID
VALID
VALID
VALID

26.7.4 Synchronous serial Interface (SSP)

26.7.4.1 AC measurement conditions

The letter "T" used in the equations in the table represents the period of internal bus frequency (fsys).

- Output levels: High = 0.7 × DVDD3A, Low = 0.3 × DVDD3A
- Input levels: High = 0.9 × DVDD3A, Low = 0.1 × DVDD3A

Note: The "Equation" column in the table shows the specifications under the conditions DVDD3A = 2.7 V to 3.6 V.

DVDD3B = 2.7V to 3.6V (fsys ≤ 40MHz : m = 2, fsys > 40MHz : m = 6)

Parameter	Symbol	Equation		fsys=40MHz	fsys=54MHz	Unit
		Min.	Max.	(m=2, n=12)	(m=6, n=12)	
SPCLK Period (Master)	T_m	(m)T However more than 50ns	–	50	111	ns
SPCLK Period (Slave)	T_s	(n)T	–	300 (3.3MHz)	222 (4.5MHz)	
SPCLK rise up time	t_r	–	15	15	15	
SPCLK fall down time	t_f	–	15	15	15	
Master mode: SPCLK low level pulse width	t_{WLM}	(m)T/2 - 15	–	10	40.5	
Master mode: SPCLK high level pulse width	t_{WHM}	(m)T/2 - 15	–	10	40.5	
Slave mode: SPCLK low level pulse width	t_{WLS}	(n)T/2 - 15	–	135	96	
Slave mode: SPCLK high level pulse width	t_{WHS}	(n)T/2 - 15	–	135	96	
Master mode: SPCLK rise/fall → to output data valid	t_{ODSM}	–	15	15	15	
Master mode: SPCLK rise/fall → output data hold	t_{ODHM}	(m)T/2 - 15	–	10	40.5	
Master mode: SPCLK rise/fall → input data valid delay time	t_{IDSM}	–	–	10	10	
Master mode: SPCLK rise/fall → input data hold	t_{IDHM}	T/2 + 5	–	17.5	14.3	
Master mode: SPFSS valid → SPCLK rise/fall	t_{OFSM}	(m)T - 15	(m)T + 15	35 to 65	96 to 126	
Slave mode: SPCLK rise/fall → output data valid delay time	t_{ODSS}	–	(3T) + 40	115	95.5	
Slave mode: SPCLK rise/fall → output data hold	t_{ODHS}	(n)T/2 + (2T)	–	100	74	
Slave mode: SPCLK rise/fall → input data valid delay time	t_{IDSS}	–	(n)T/2 + (3T) - 10	215	156.5	
Slave mode: SPCLK rise/fall → input data hold	t_{IDHS}	(3T) + 15	–	90	70.5	
Slave mode: SPFSS valid → SPCLK rise/fall	t_{OFSS}	(n)T + 15	–	285	207	

DVDD3B = 1.65V to 3.6V (fsys ≤ 40MHz : m = 2, fsys > 40MHz : m = 6)

Parameter	Symbol	Equation		fsys=40MHz (m=2, n=12)	fsys=54MHz (m=6, n=12)	Unit
		Min.	Max.			
SPCLK Period (Master)	T_m	(m)T However more than 50ns	–	50	111	ns
SPCLK Period (Slave)	T_s	(n)T	–	300 (3.3MHz)	222 (4.5MHz)	
SPCLK rise up time	t_r	–	20	20	20	
SPCLK fall down time	t_f	–	20	20	20	
Master mode: SPCLK low level pulse width	t_{WLM}	(m)T/2 - 20	–	5	35.5	
Master mode: SPCLK high level pulse width	t_{WHM}	(m)T/2 - 20	–	5	35.5	
Slave mode: SPCLK low level pulse width	t_{WLS}	(n)T/2 - 20	–	130	91	
Slave mode: SPCLK high level pulse width	t_{WHS}	(n)T/2 - 20	–	130	91	
Master mode: SPCLK rise/fall → to output data valid	t_{ODSM}	–	20	20	20	
Master mode: SPCLK rise/fall → output data hold	t_{ODHM}	(m)T/2 - 20	–	5	35.5	
Master mode: SPCLK rise/fall → input data valid delay time	t_{IDSM}	–	–	10	10	
Master mode: SPCLK rise/fall → input data hold	t_{IDHM}	T/2 + 5	–	17.5	14.3	
Master mode: SPFSS valid → SPCLK rise/fall	t_{OFSM}	(m)T - 15	(m)T + 25	35 to 75	96 to 136	
Slave mode: SPCLK rise/fall → output data valid delay time	t_{ODSS}	–	(3T) + 55	130	110.5	
Slave mode: SPCLK rise/fall → output data hold	t_{ODHS}	(n)T/2 + (2T)	–	100	74	
Slave mode: SPCLK rise/fall → input data valid delay time	t_{IDSS}	–	(n)T/2 + (3T) - 10	215	156.5	
Slave mode: SPCLK rise/fall → input data hold	t_{IDHS}	(3T) + 15	–	90	70.5	
Slave mode: SPFSS valid → SPCLK rise/fall	t_{OFSS}	(n)T + 15	–	285	207	

Note: Baud rate clock is set under below condition

Master Mode

m=(<CPSDVSR>×(1+<SCR>))= fsys /f_{SPCLK} <CPSDVR> is set only even number.
65024≥ m ≥ 2

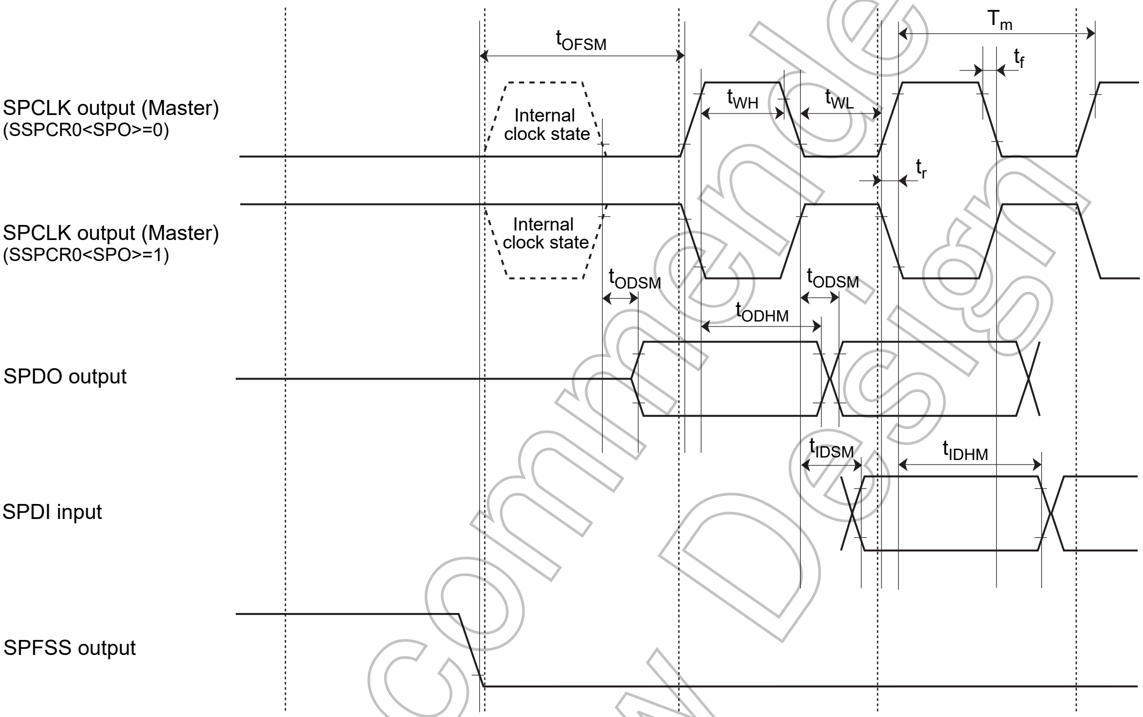
Slave mode

n=(<CPSDVSR>×(1+<SCR>))= fsys /f_{SPCLK} 65024≥ n≥ 12

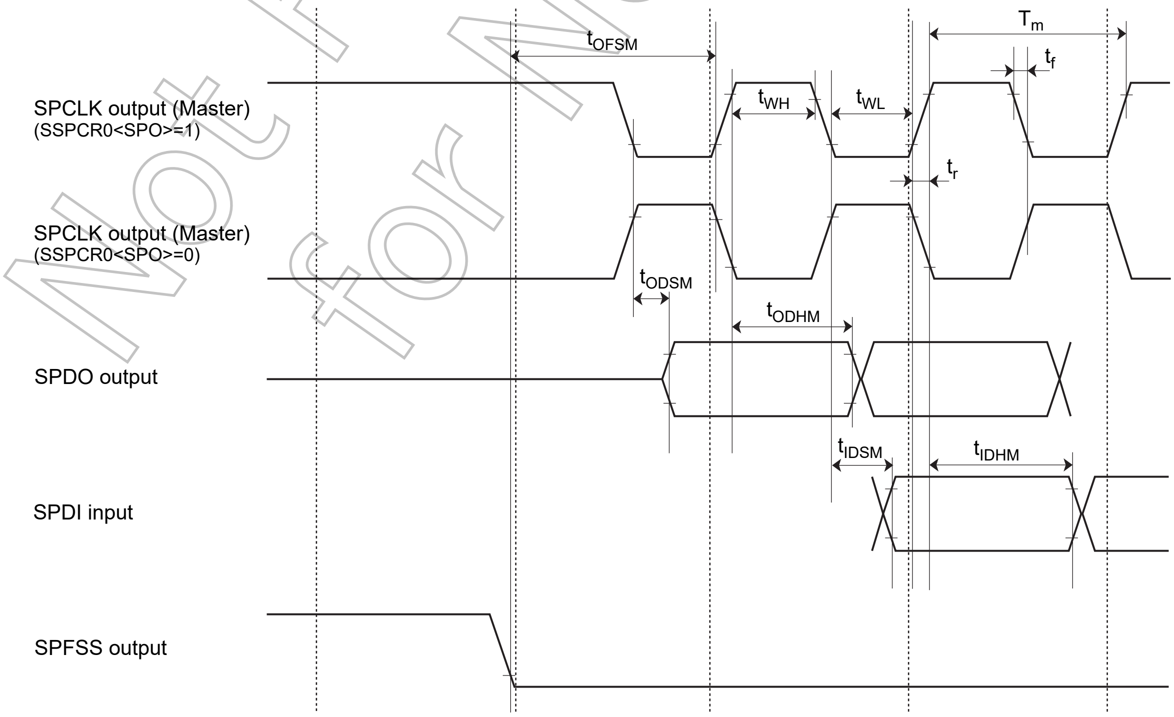
26.7.4.2 SSP SPI mode (Master)

- $fsys / 2 \geq f_{SPCLK} \geq fsys / 65024$

(1) Master SSPxCR0<SPH>=0 (Data is latched on the first edge.)



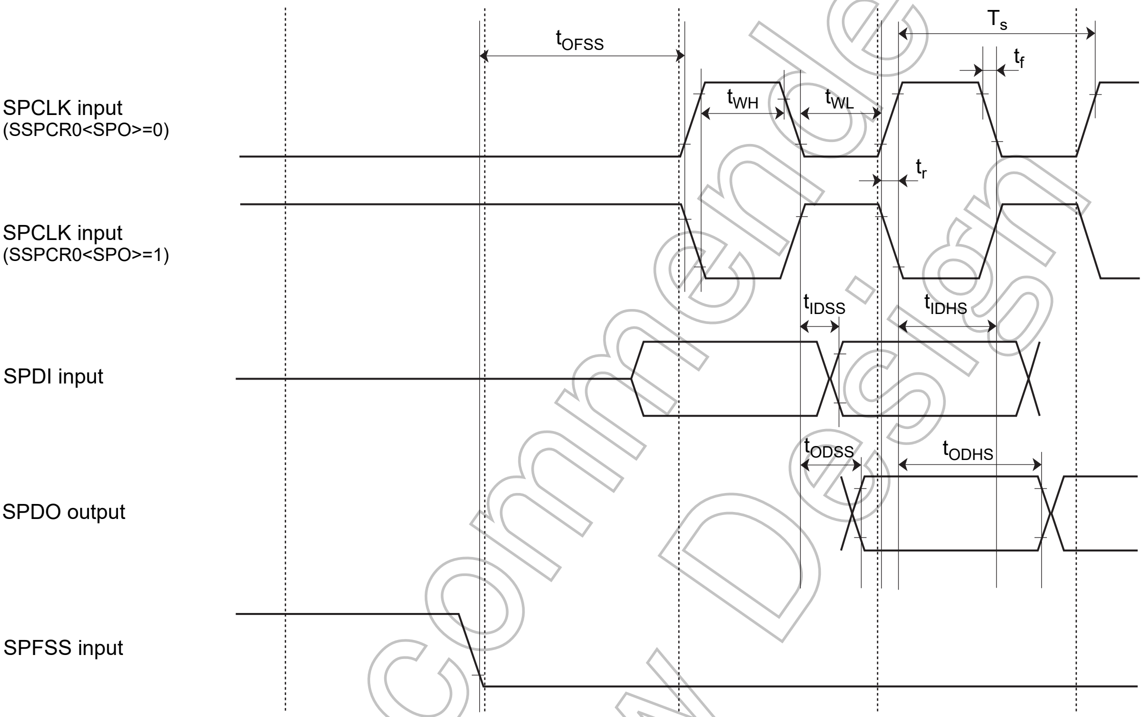
(2) Master SSPxCR0<SPH>=1 (Data is latched on the second edge.)



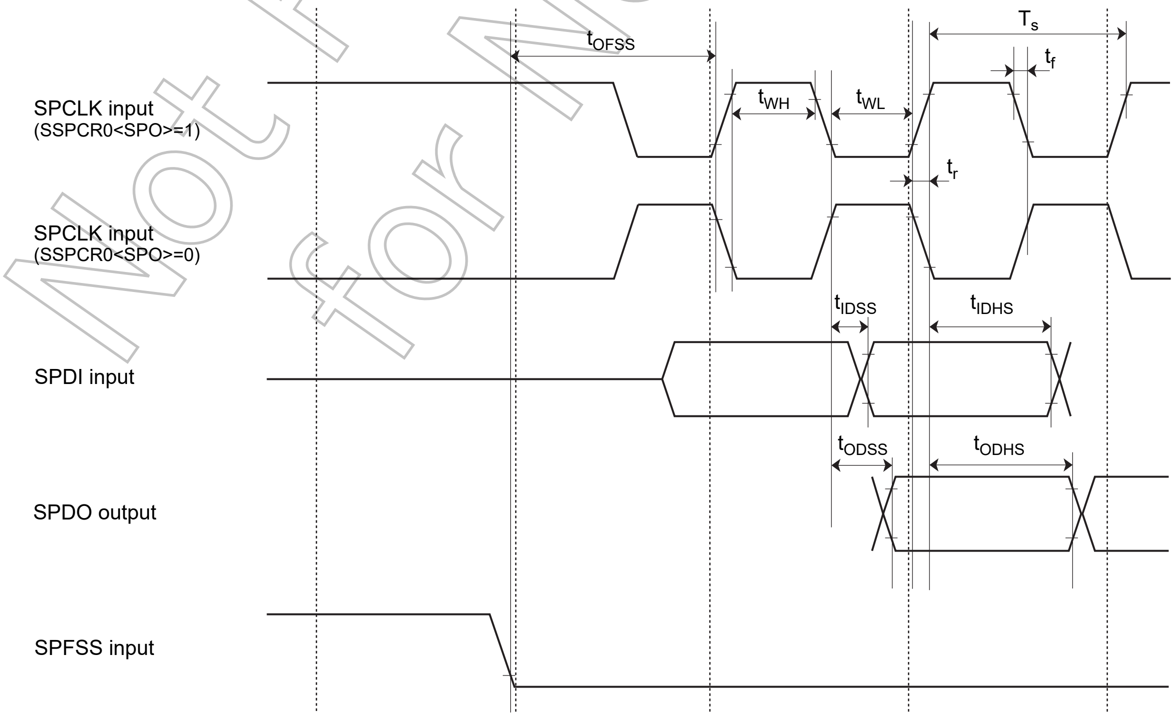
26.7.4.3 SSP SPI mode (Slave)

- fsys / 12 ≥ f_{SPCLK} ≥ fsys / 65024

(1) Slave SSPxCR0<SPH>=0 (Data is latched on the first edge.)



(2) Slave SSPCR0<SPH>=1 (Data is latched on the second edge.)



26.7.5 16-bit timer / event counter

26.7.5.1 Event Counter

In the table below, the letter x represents the 16-bit timer / event counter operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		40 MHz		54 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
Clock Low pulse width	t_{VCKL}	2x + 100	–	150	–	137	–	ns
Clock High pulse width	t_{VCKH}	2x + 100	–	150	–	137	–	ns

26.7.5.2 Capture

In the table below, the letter x represents the 16-bit timer / event counter operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		40 MHz		54 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
Low pulse width	t_{CPL}	2x + 100	–	150	–	137	–	ns
High pulse width	t_{CPH}	2x + 100	–	150	–	137	–	ns

26.7.6 External Interrupt

In the table below, the letter x represents the fsys cycle time.

1. Except STOP1 and STOP2 release interrupts

Parameter	Symbol	Equation		40 MHz		54 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
INT0 to B low level pulse width	t_{INTAL}	x + 100	–	125	–	118.5	–	ns
INT0 to B high level pulse width	t_{INTAH}	x + 100	–	125	–	118.5	–	ns

2. STOP1 release interrupts

Parameter	Symbol	Min.	Max.	Unit
INT0 to B low level pulse width	t_{INTBL}	100	–	ns
INT0 to B high level pulse width	t_{INTBH}	100	–	ns

3. STOP2 release interrupts

Parameter	Symbol	Min.	Max.	Unit
INT0 to B high level pulse width	t_{INTCH}	500	–	μs

26.7.7 $\overline{\text{NMI}}$

1. Except STOP1 and STOP2 release interrupts

Parameter	Symbol	Min.	Max.	Unit
$\overline{\text{NIM}}$ low level pulse width	t_{INTCL}	100	−	ns

2. STOP1 release interrupts

Parameter	Symbol	Min.	Max.	Unit
$\overline{\text{NMI}}$ low level pulse width	t_{INTBL}	100	−	ns

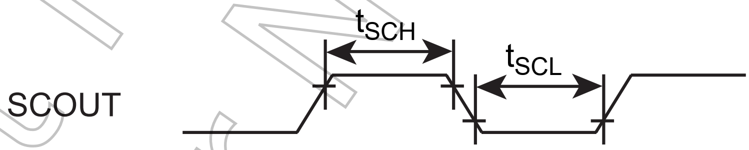
3. STOP2 release interrupts

Parameter	Symbol	Min.	Max.	Unit
$\overline{\text{NMI}}$ low level pulse width	t_{INTCL}	500	−	μs

26.7.8 SCOUT Pin AC Characteristic

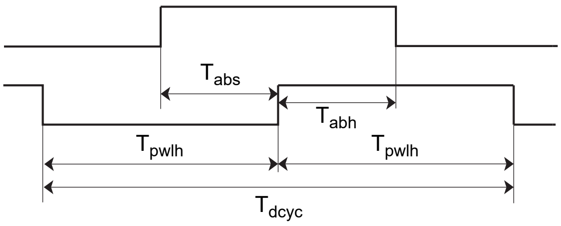
Parameter	Symbol	Equation		40 MHz		54 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
High level pulse width	t_{SCH}	0.5T − 5	−	7.5	−	4.3	−	ns
Low level pulse width	t_{SCL}	0.5T − 5	−	7.5	−	4.3	−	ns

Note: In the above table, the letter T represents the cycle time of the SCOUT output clock.



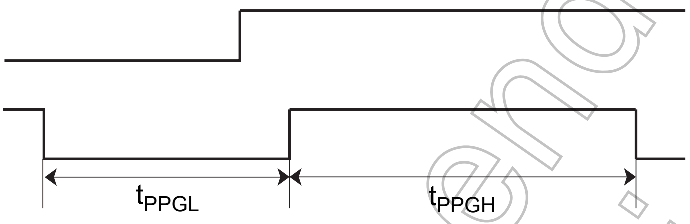
26.7.9 Two-phase pulse input counter Pin AC Characteristic

Parameter	Symbol	Min.	Max.	Unit
Two-phase pulse input cycle	T_{dcyc}	2	−	μs
Two-phase pulse input setup	T_{abs}	1	−	
Two-phase pulse input hold	T_{abh}	1	−	
Two-phase input Low/High pulse width	T_{pwlh}	1	−	


T_{abs}
T_{abh}
T_{pwlh}
T_{pwlh}
T_{dcyc}

26.7.10 High resolution 16-bit timer / PPG output Pin AC Characteristic

Parameter	Symbol	Equation		40 MHz		54 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
High level pulse width	t_{PPGH}	100	−	100	−	100	−	ns
Low level pulse width	t_{PPGL}	100	−	100	−	100	−	



26.7.11 $\overline{\text{ADTRG}}$ Trigger Input Pin AC Characteristic

In the table below, the letter x represents the fsys cycle time. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		40MHz		54MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
Low level pulse width	T_{adl}	2 x + 20	−	32.5	−	29.3	−	ns
High level pulse interval	T_{adh}	2 x + 20	−	32.5	−	29.3	−	

26.7.12 External bus interface AC Characteristic

26.7.12.1 AC measurement condition

- Output levels: High = 0.7 × DVDD3A, 0.7 × DVDD3B
- Output levels: Low = 0.3 × DVDD3A, 0.3 × DVDD3B

26.7.12.2 Separate Bus mode

Conditional variable : RWS = 1, TW = 2, RWH = 1 and CSH =1

- RWS : Number of setup cycle insertion before $\overline{\text{RD}}$, $\overline{\text{WR}}$ asserted (TW = 0, 1, 2 or 4)
- TW : Number of internal wait insertion (TW = 0 to 15)
- RWH : Number of $\overline{\text{RD}}$, $\overline{\text{WR}}$ recovery cycle insertion (RWH = 0 to 6 or 8)
- CSH : Number of $\overline{\text{CSx}}$ recovery cycle insertion (CSH = 0, 1, 2 or 4)

DVDD3A = DVDD3B = 2.7 V to 3.6 V

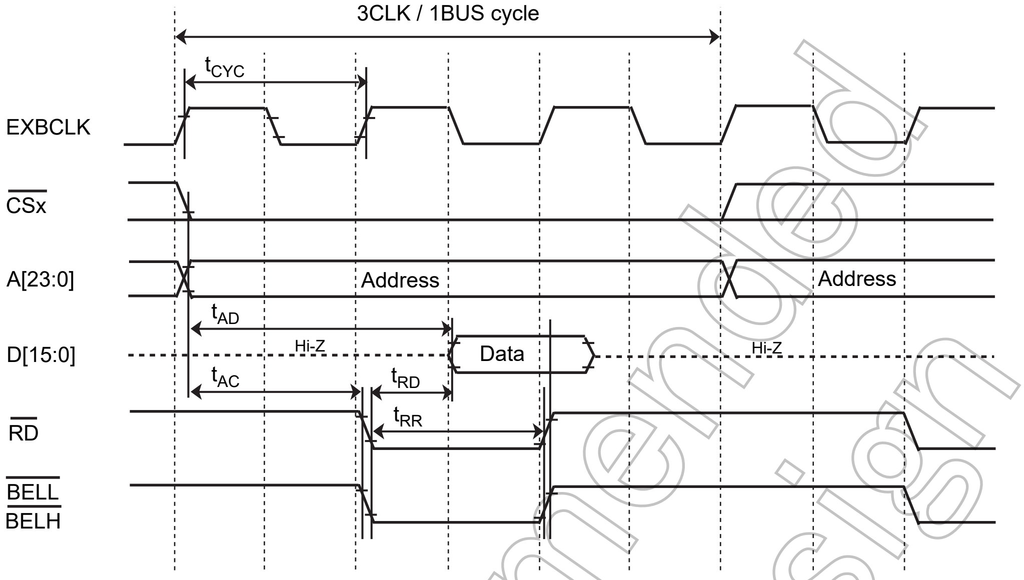
Parameter	Symbol	Equation		40MHz		54MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
System clock period (x)	t_{SYS}	x	−	25	−	18.5	−	ns
External bus clock (EXBCLK)	t_{CYC}	x	−	25	−	18.5	−	
A[23:0] valid → $\overline{RD}$, $\overline{WR}$ asserted	t_{AC}	x (1+RWS)−10	−	40	−	27	−	
$\overline{RD}$, $\overline{WR}$ negated → A[23:0] hold	t_{CAR}	x (1+RWH+CSH)−10	−	65	−	45.5	−	
A[23:0] valid → D[15:0] data input	t_{AD}	-	x (1+RWS+TW)−35	−	65	−	39	
$\overline{RD}$ asserted → D[15:0] data input	t_{RD}	-	x (1+TW)−30	−	45	−	25.5	
$\overline{RD}$ Low-level pulse width	t_{RR}	x (1+TW)−12	−	63	−	43.5	−	
$\overline{RD}$ negated → D[15:0] hold	t_{HR}	x (1+RWH)−7	−	43	−	30	−	
$\overline{RD}$ negated → A[23:0] output	t_{RAE}	x (1+RWH+CSH)−15	−	60	−	40.5	−	
$\overline{WR}$ Low-level pulse width	t_{WW}	x (1+TW)−15	−	60	−	40.5	−	
$\overline{WR}$ negated → D[15:0] valid	t_{DO}	-	−	−	20	−	20	
D[15:0] valid → $\overline{WR}$ negated	t_{DW}	x (1+TW)−15	−	60	−	40.5	−	
$\overline{WR}$ negated → D[15:0] hold	t_{WD}	x (1+RWH)−7	−	43	−	30	−	

DVDD3A = 3.6V to 2.7V, DVDD3B = 3.6V to 1.65V

Parameter	Symbol	Equation		40MHz		54MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
System clock period (x)	t_{SYS}	x	−	25	−	18.5	−	ns
External bus clock (EXBCLK)	t_{CYC}	x	−	25	−	18.5	−	
A[23:0] valid → $\overline{RD}$, $\overline{WR}$ asserted	t_{AC}	x (1+RWS)−20	−	30	−	17	−	
$\overline{RD}$, $\overline{WR}$ negated → A[23:0] hold	t_{CAR}	x (1+RWH+CSH)−25	−	50	−	30.5	−	
A[23:0] valid → D[15:0] data input	t_{AD}	-	x (1+RWS+TW)−45	−	55	−	29	
$\overline{RD}$ asserted → D[15:0] data input	t_{RD}	-	x (1+TW)−40	−	35	−	15.5	
$\overline{RD}$ Low-level pulse width	t_{RR}	x (1+TW)−20	−	55	−	35.5	−	
$\overline{RD}$ negated → D[15:0] hold	t_{HR}	x (1+RWH)−15	−	35	−	22	−	
$\overline{RD}$ negated → A[23:0] output	t_{RAE}	x (1+RWH+CSH)−20	−	55	−	35.5	−	
$\overline{WR}$ Low-level pulse width	t_{WW}	x (1+TW)−20	−	55	−	35.5	−	
$\overline{WR}$ negated → D[15:0] valid	t_{DO}	-	−	−	25	−	25	
D[15:0] valid → $\overline{WR}$ negated	t_{DW}	x (1+TW)−20	−	55	−	35.5	−	
$\overline{WR}$ negated → D[15:0] hold	t_{WD}	x (1+RWH)−15	−	35	−	22	−	

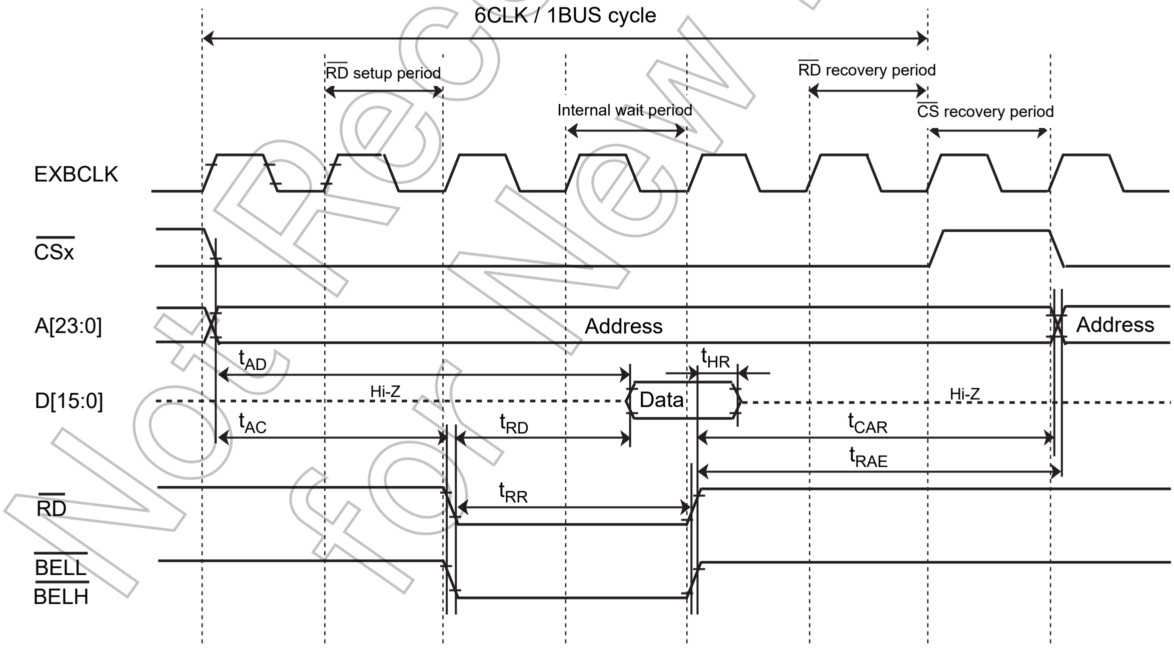
(1) Read cycle timing (minimum bus cycle)

(Neither Cycle expander, RD setup, Internal wait, CS recovery nor RD recovery function are used.)



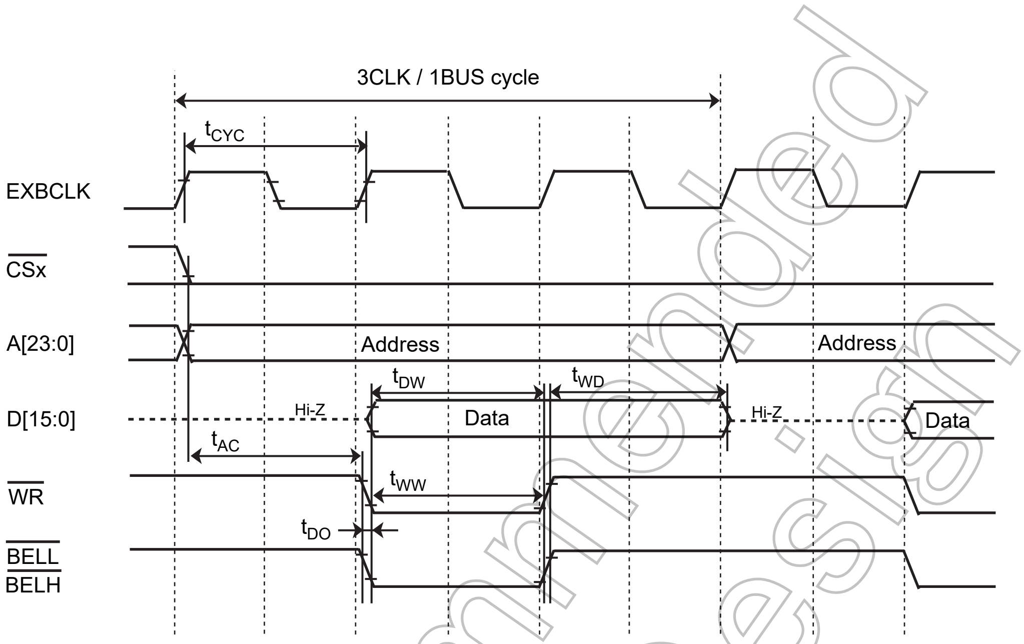
(2) Read cycle timing (1 bus cycle per 6 clock)

(RD setup, Internal wait, CS recovery and RD recovery function are set to 1 cycle though Cycle expander function is not used.)



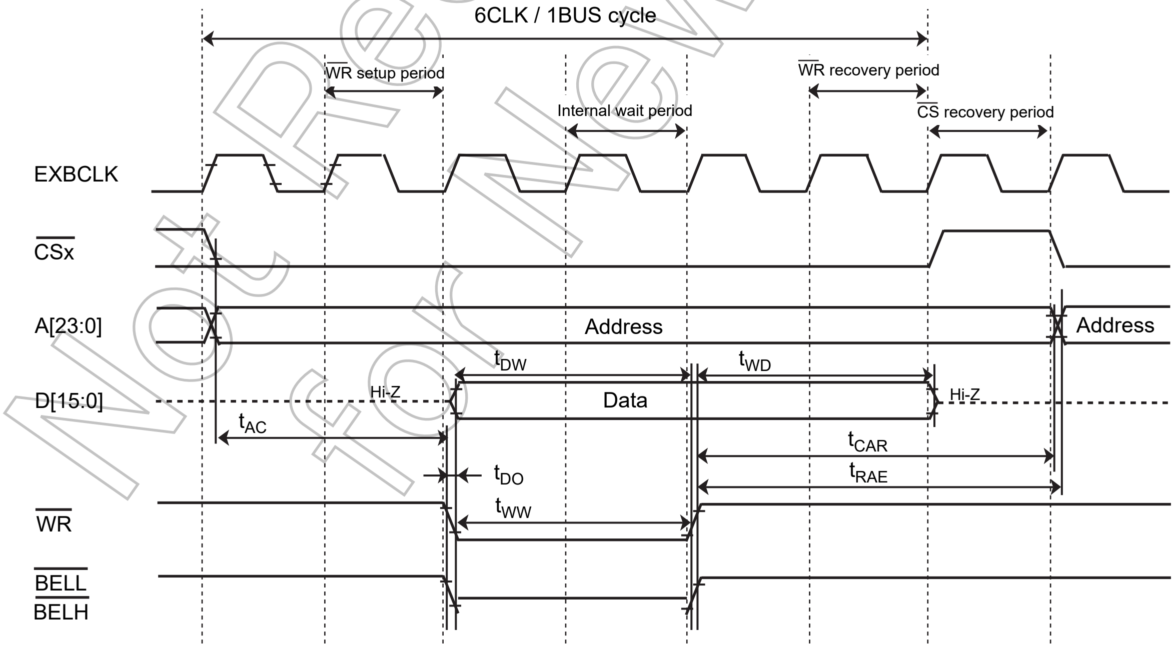
(3) Write cycle timing (minimum bus cycle)

(Neither Cycle expander, WR setup, Internal wait, CS recovery nor WR recovery function are used.)



(4) Write cycle timing (1 bus cycle per 6 clock)

(WR setup, Internal wait, CS recovery and WR recovery function are set to 1 cycle though Cycle expander function is not used.)



26.7.12.3 Multiplex Bus mode

Conditional variable : ALE = 1, RWS = 1, TW = 2, RWH = 1 and CSH =1

- ALE : Number of ALE cycle insertion (ALE = 1 + n; n = 0, 1, 2 or 4)
- RWS : Number of setup cycle insertion before $\overline{RD}$, $\overline{WR}$ asserted (TW = 0, 1, 2 or 4)
- TW : Number of internal wait insertion (TW = 0 to 15)
- RWH : Number of $\overline{RD}$, $\overline{WR}$ recovery cycle insertion (RWH = 0 to 6 or 8)
- CSH : Number of $\overline{CSx}$ recovery cycle insertion (CSH = 0, 1, 2 or 4)

DVDD3A = DVDD3B = 2.7 V to 3.6 V

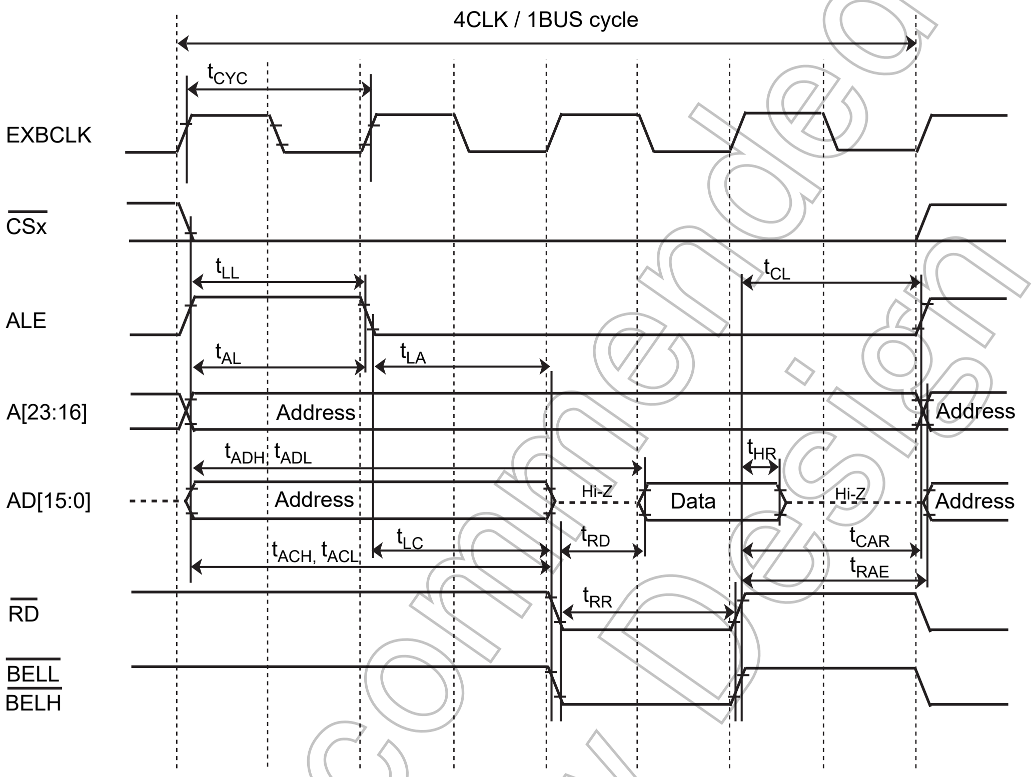
Parameter	Symbol	Equation		40MHz		54MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
System clock period (x)	t_{SYS}	x	–	25	–	18.5	–	ns
External bus clock (EXBCLK)	t_{CYC}	x	–	25	–	18.5	–	
A[23:0] valid → ALE negated	t_{AL}	x (1+ALE)-15	–	35	–	22	–	
ALE negated → A[23:0] hold	t_{LA}	x (1+RWS)-7	–	43	–	30	–	
ALE high pulse width	t_{LL}	x (1+ALE)-15	–	35	–	22	–	
ALE negated → $\overline{RD}$ or $\overline{WR}$ asserted	t_{LC}	x (1+RWS)-7	–	43	–	30	–	
$\overline{RD}$ or $\overline{WR}$ negated → ALE asserted	t_{CL}	x (1+RWH+CSH)-15	–	60	–	40.5	–	
A[15:0] valid → $\overline{RD}$ or $\overline{WR}$ asserted A[23:16] valid → $\overline{RD}$ or $\overline{WR}$ asserted	t_{ACL} t_{ACH}	x (1+ALE+RWS)-15	–	60	–	40.5	–	
$\overline{RD}$ or $\overline{WR}$ negated → A[31:16] hold	t_{CAR}	x (1+RWH+CSH)-15	–	60	–	40.5	–	
A[15:0] valid → D[15:0] input A[23:16] valid → D[15:0] input	t_{ADL} t_{ADH}	–	x (3+ALE+RWS+TW)-35	–	140	–	94.5	
$\overline{RD}$ asserted → D[15:0] input	t_{RD}	–	x (1+TW)-30	–	45	–	25.5	
$\overline{RD}$ low pulse width	t_{RR}	x (1+TW)-12	–	63	–	43.5	–	
$\overline{RD}$ negated → D[15:0] hold	t_{HR}	x (1+RWH)-7	–	43	–	30	–	
$\overline{RD}$ negated → A[23:0] output	t_{RAE}	x (1+RWH+CSH)-15	–	60	–	40.5	–	
$\overline{WR}$ low pulse width	t_{WW}	x (1+TW)-15	–	60	–	40.5	–	
D[15:0] valid → $\overline{WR}$ negated	t_{DW}	x (1+TW)-15	–	60	–	40.5	–	
$\overline{WR}$ negated → D[15:0] hold	t_{WD}	x (1+RWH)-7	–	43	–	30	–	

DVDD3A = 3.6V to 2.7V, DVDD3B = 3.6V to 1.65V

Parameter	Symbol	Equation		40MHz		54MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
System clock period (x)	t_{SYS}	x	−	25	−	18.5	−	ns
External bus clock (EXBCLK)	t_{CYC}	x	−	25	−	18.5	−	
A[23:0] valid → ALE negated	t_{AL}	x (1+ALE)-30	−	20	−	7	−	
ALE negated → A[23:0] hold	t_{LA}	x (1+RWS)-15	−	35	−	22	−	
ALE high pulse width	t_{LL}	x (1+ALE)-30	−	20	−	7	−	
ALE negated → $\overline{\text{RD}}$ or $\overline{\text{WR}}$ asserted	t_{LC}	x (1+RWS)-15	−	35	−	22	−	
$\overline{\text{RD}}$ or $\overline{\text{WR}}$ negated → ALE asserted	t_{CL}	x (1+RWH+CSH)-25	−	50	−	30.5	−	
A[15:0] valid → $\overline{\text{RD}}$ or $\overline{\text{WR}}$ asserted A[23:16] valid → $\overline{\text{RD}}$ or $\overline{\text{WR}}$ asserted	t_{ACL} t_{ACH}	x (1+ALE+RWS)-15	−	45	−	25.5	−	
$\overline{\text{RD}}$ or $\overline{\text{WR}}$ negated → A[31:16] hold	t_{CAR}	x (1+RWH+CSH)-25	−	50	−	30.5	−	
A[15:0] valid → D[15:0] input A[23:16] valid → D[15:0] input	t_{ADL} t_{ADH}	−	x (3+ALE+RWS+TW)-45	−	130	−	84.5	
$\overline{\text{RD}}$ asserted → D[15:0] input	t_{RD}	−	x (1+TW)-40	−	35	−	15.5	
$\overline{\text{RD}}$ low pulse width	t_{RR}	x (1+TW)-20	−	55	−	35.5	−	
$\overline{\text{RD}}$ negated → D[15:0] hold	t_{HR}	x (1+RWH)-15	−	35	−	22	−	
$\overline{\text{RD}}$ negated → A[23:0] output	t_{RAE}	x (1+RWH+CSH)-25	−	50	−	30.5	−	
$\overline{\text{WR}}$ low pulse width	t_{WW}	x (1+TW)-20	−	55	−	35.5	−	
D[15:0] valid → $\overline{\text{WR}}$ negated	t_{DW}	x (1+TW)-20	−	55	−	35.5	−	
$\overline{\text{WR}}$ negated → D[15:0] hold	t_{WD}	x (1+RWH)-15	−	35	−	22	−	

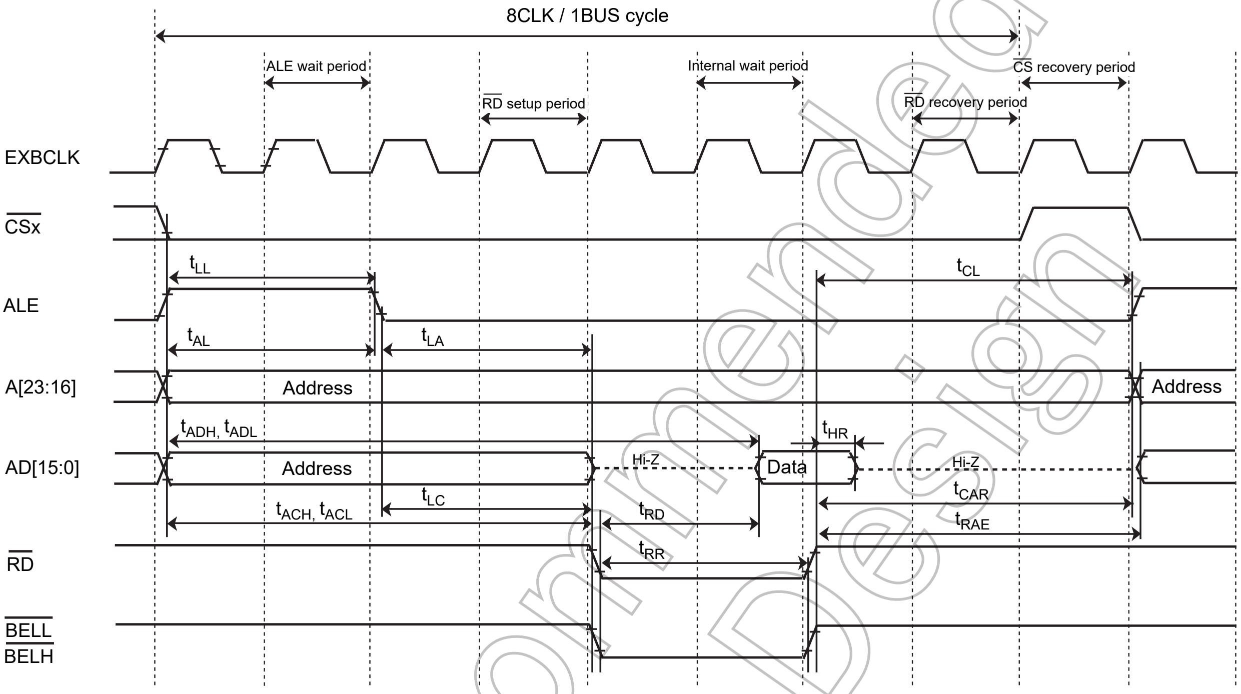
(1) Read cycle timing (minimum bus cycle)

(Neither Cycle expander, ALE wait, RD setup, Internal wait, CS recovery nor RD recovery function are used.)



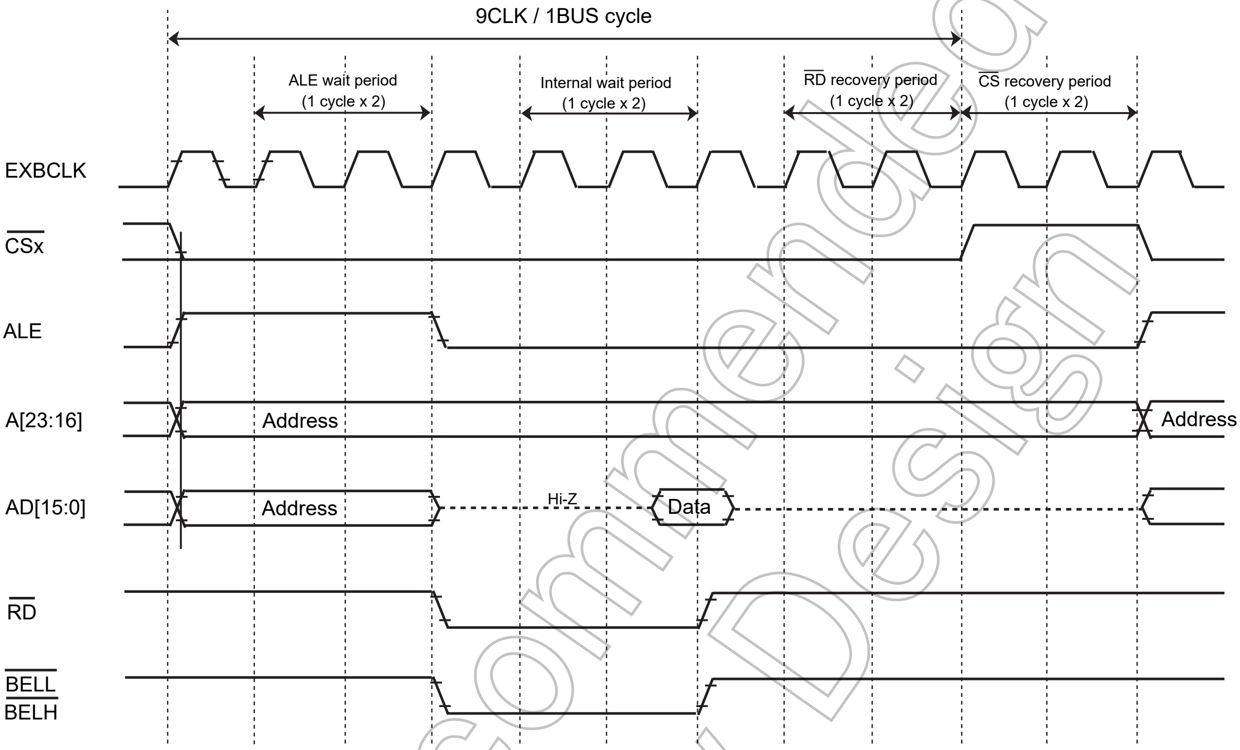
(2) Read cycle timing (1 bus cycle per 8 clock)

(ALE wait, RD setup, Internal wait, CS recovery and RD recovery function are set to 1 cycle though Cycle expander function is not used.)



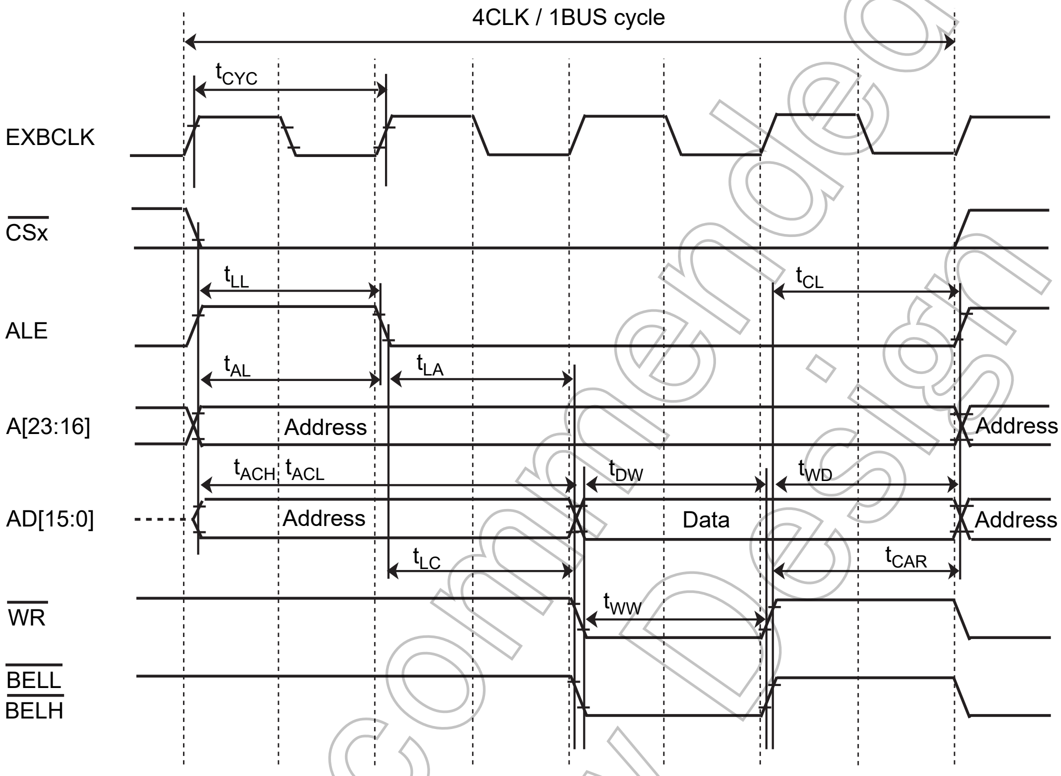
(3) Read cycle timing (1 bus cycle per 9 clock)

(ALE wait, RD setup, Internal wait, CS recovery and RD recovery function are set to 1 cycle though Cycle expander function is set double.)


9CLK / 1BUS cycle
ALE wait period (1 cycle x 2)
Internal wait period (1 cycle x 2)
RD recovery period (1 cycle x 2)
CS recovery period (1 cycle x 2)
EXBCLK
CSx
ALE
A[23:16]
Address
Address
AD[15:0]
Address
Hi-Z
Data
RD
BELL
BELH

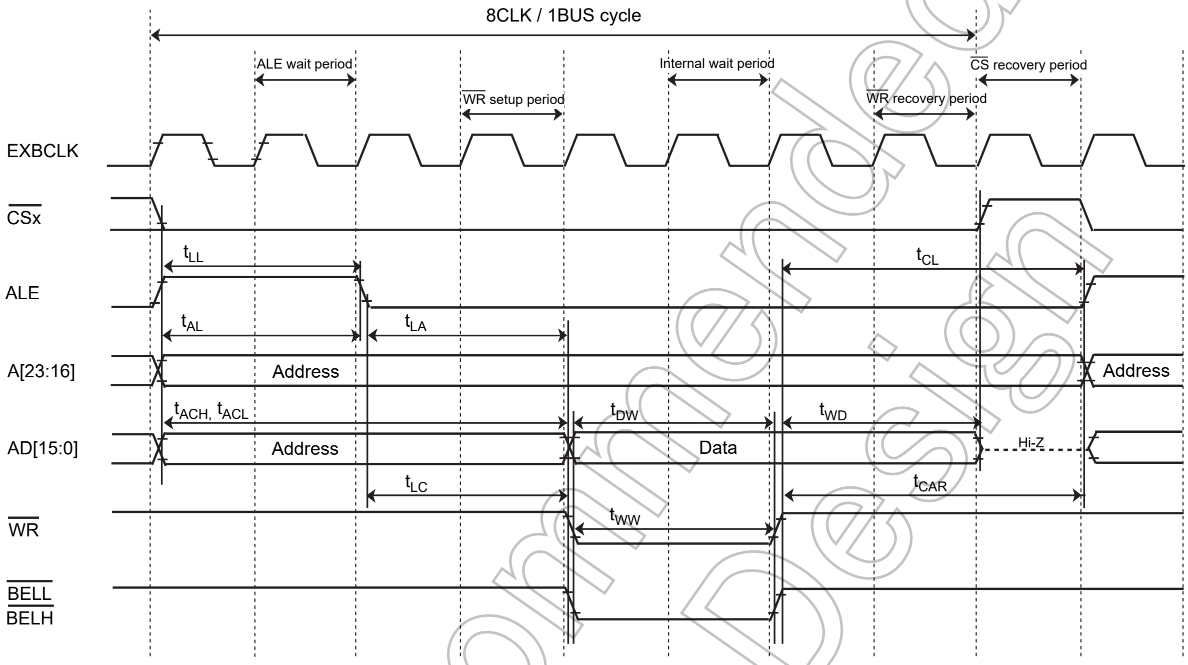
(4) Write cycle timing (minimum bus cycle)

(Neither Cycle expander, ALE wait, WR setup, Internal wait, CS recovery nor WR recovery function are used.)



(5) Write cycle timing (1 bus cycle per 8 clock)

(ALE wait, WR setup, Internal wait, CS recovery and WR recovery function are set to 1 cycle though Cycle expander function is not used.)



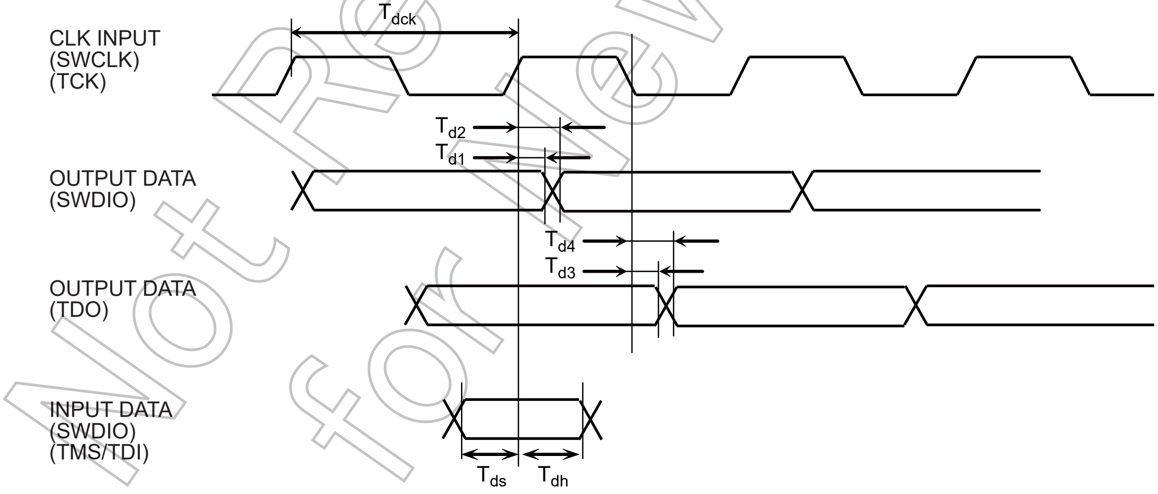
26.7.13 Debug Communication

26.7.13.1 SWD Interface

Parameter	Symbol	Min.	Max.	Unit
CLK cycle	T_{dck}	100	–	ns
CLK rise → Output data hold	T_{d1}	4	–	
CLK fall → Output data valid	T_{d2}	–	30	
Input data valid → CLK rise	T_{ds}	20	–	
CLK rise → Input data hold	T_{dh}	15	–	

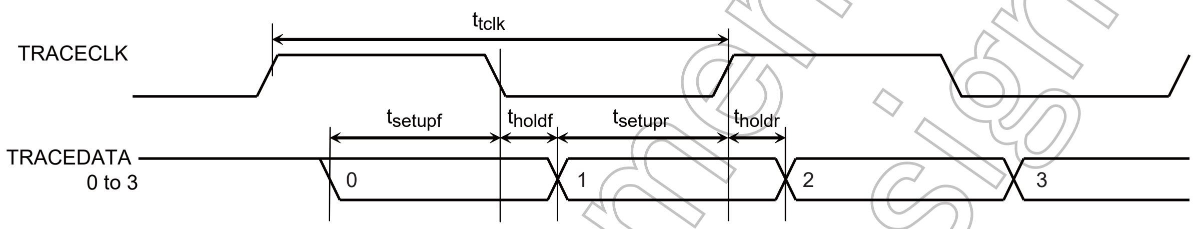
26.7.13.2 JTAG Interface

Parameter	Symbol	Min.	Max.	Unit
CLK cycle	T_{dck}	100	–	ns
CLK fall → Output data hold	T_{d3}	4	–	
CLK fall → Output data valid	T_{d4}	–	50	
Input data valid → CLK rise	T_{ds}	20	–	
CLK rise → Input data hold	T_{dh}	15	–	



26.7.14 ETM Trace

Parameter	Symbol	Min.	Max.	Unit
TRACECLK cycle	t_{tclk}	37	–	ns
TRACEDATA valid ← TRACECLK rise	t_{setupr}	2	–	
TRACECLK rise → TRACEDATA hold	t_{holdr}	1	–	
TRACEDATA valid ← TRACECLK fall	t_{setupf}	2	–	
TRACECLK fall → TRACEDATA hold	t_{holdf}	1	–	



26.7.15 On chip oscillator

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillating frequency	IHOSC	Ta = -40 to 85°C	9	10	11	MHz

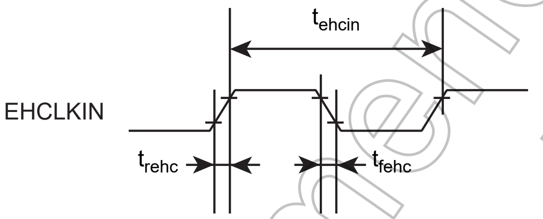
Note: The on-chip-oscillator can not be used as system clock (fsys) which is required oscillation accuracy.

26.7.16 External oscillation

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
High frequency oscillation	EHOSC	Ta = -40 to 85°C	8	–	16	MHz

26.7.17 External clock input

Parameter	Symbol	Min.	Typ.	Max.	Unit
External clock frequency	t_{ehcin}	8	–	27	MHz
External clock duty	–	45	–	55	%
External clock input rise time	t_{rehc}	–	–	10	ns
External clock input fall time	t_{fehc}	–	–	10	ns



26.7.18 Flash Characteristics

Parameter	Condition	Min.	Typ.	Max.	Unit
Guarantee on Flash-memory re-writing	DVDD3A = AVDD3 = RVDD3 = 2.7 V ~ 3.6 V, DVDD3B = 1.65 V ~ 3.6 V, Ta = 0 to 70°C	–	–	200	Times

26.7.18.1 Noise Filter Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit
Analog noise canceler width	–	15	30	60	ns

26.8 Recommended Oscillation Circuit

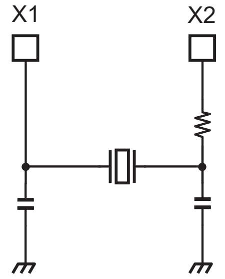


Figure 26-1 High-frequency oscillation connection

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 TMPM341FDXBG and TMPM341FYXBG have been evaluated by the oscillator vender below. Please refer this information when selecting external parts

26.8.1 Ceramic oscillator

The TMPM341FDXBG and TMPM341FYXBG recommend the high-frequency

oscillator by Murata Manufacturing Co., Ltd.

Please refer to the Murata Website for details.

26.9 Handling Precaution

26.9.1 Power-on sequence

The power-on sequence must include the time for the internal regulator, internal flash memory and internal oscillator to be stable and the reset. In the TX03, the internal circuit automatically insert the time for the internal regulator which requires the time at least 1 ms and after this, internal reset operation requires 4096 cycles on internal oscillation, therefore, A little bit of time differences occur until CPU start operate. And there are multiple independent Power supply, therefore you must be followed the procedure of Power-On.

If you use one of four power sources (DVDD3A, DVDD3B, RVDD3, AVDD3) with a different voltage, or you use only DVDD3B with a different voltage (1.65V), set the power supply to satisfy the following condition: DVDD3A=RVDD3=AVDD3 rising time (time to guaranteed operating voltage) ≤ DVDD3B rising time (time to guaranteed operating voltage).

The time required to achieve stable oscillation varies with system. At cold reset, the external reset pin must be kept "Low" for a duration of time sufficiently long enough for the internal regulator and oscillator to be stable.

Figure 26-2 shows the power-on sequence.

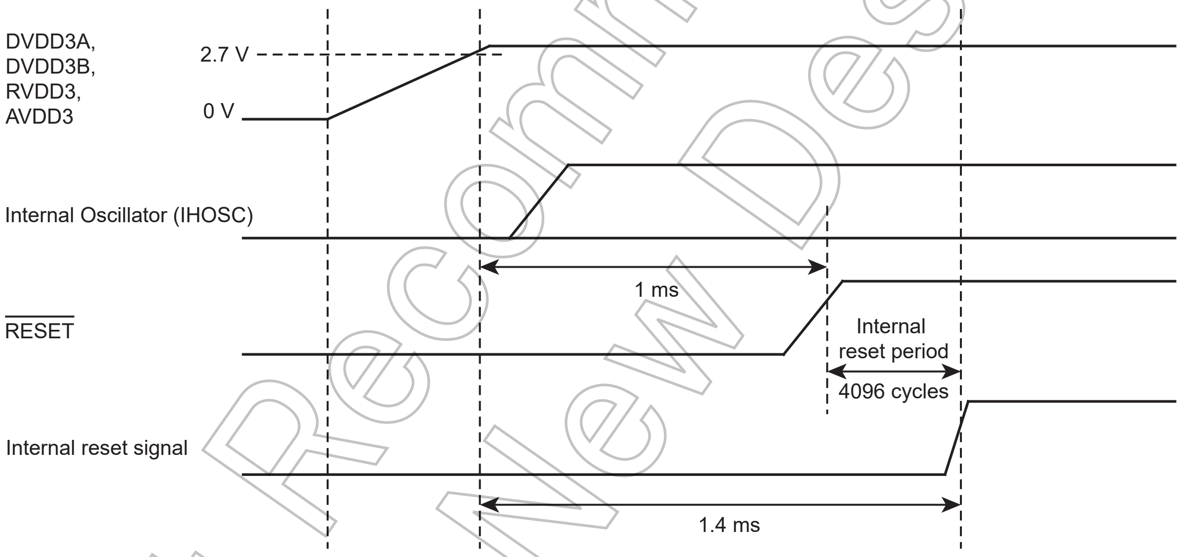
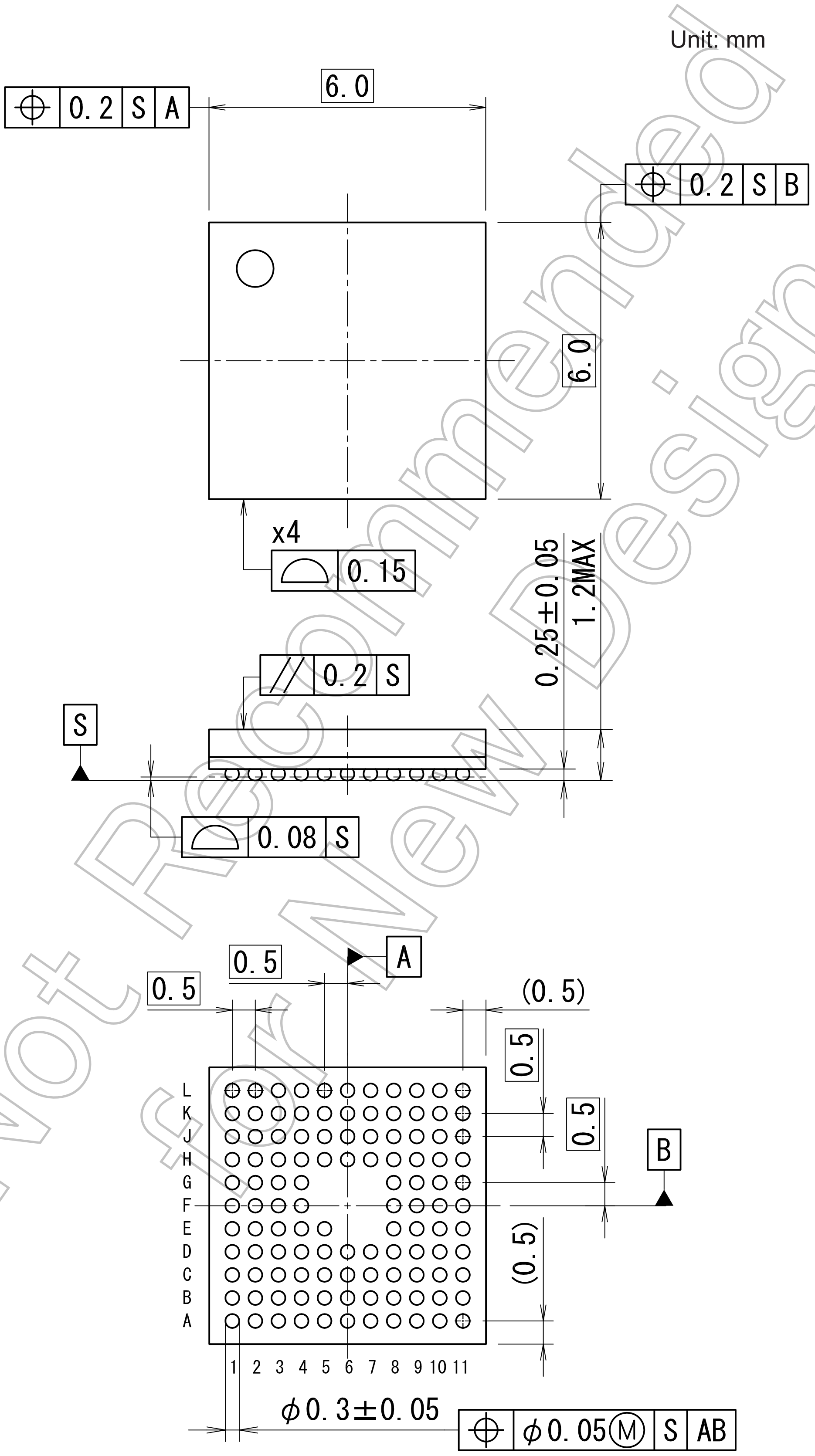


Figure 26-2 Power-on sequence

27. Package Dimensions

Type:P-TFBGA113-0606-0.50A4



• 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.**